Three Metaheuristic Algorithms for Solving the Multi-item Capacitated Lot-sizing Problem with Product Returns and Remanufacturing

Esmaeil Mehdizadeh[a,*], Amir Fatehi Kivi[b]

[a] *Assistant Professor, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[b] *MSc, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
Received 09 April, 2014; Revised 31 August, 2014; Accepted 28 October, 2014

Abstract

This paper proposes a new mixed integer programming model for multi-item capacitated lot-sizing problem with setup times, safety stock and demand shortages in closed-loop supply chains. The returned products from customers can either be disposed or be remanufactured to be sold as new ones again. Due to the complexity of problem, three meta-heuristics algorithms named simulated annealing (SA) algorithm, vibration damping optimization (VDO) algorithm and harmony search (HS) algorithm have been used to solve this model. Additionally, Taguchi method is conducted to calibrate the parameter of the meta-heuristics and select the optimal levels of the algorithm's performance influential factors. To verify and validate the efficiency of the proposed algorithms in terms of solution quality, the obtained results were compared with those obtained from Lingo 8 software for a different problem. Finally, computational results of these algorithms were compared and analyzed by producing and solving some small, medium and large-size test problems. The results confirmed the efficiency of the HS algorithm against the other methods.
Keywords: Closed-loop supply chain, Lot-sizing, Safety stocks, Vibration damping, Harmony search.

1. Introduction

The production planning problems encountered in real-life situations are generally intractable due to a number of practical constraints. The decision maker has to find a good feasible solution in a reasonable execution time rather than an optimal one. The lot-sizing problem (LSP) is a crucial step and well-known optimization problem in production planning which involves time-varying demand for set of N items over T periods. It is a class of production planning problems in which the availability amounts of the production plan are always considered as a decision variable. Two versions of the lot-sizing problems are capacitated and uncapacitated lot-sizing problem.

In industrial applications, several factors may sophisticate making the best decisions. For this reason, the capacitated lot-sizing problem and its variations have received a lot of attention from academic researchers. On the other hand, backlogging, safety stocks, limited outsourcing and returned products are four complicating constraints to reach the desired solutions in lot-sizing problem. Chen and Thizy (1990) proved that the multi-item capacitated lot-sizing problem with setup times is strongly NP-hard. There are many references dealing with the capacitated lot-sizing problem and explanation of why it is one of the most popular among exact and approximate solution methods using Lagrangian relaxation of the capacity constraint and comparing this approach with every alternate relaxation of the classical formulation of the problem.

Absi and Kedad-Sidhoum (2009) addressed a multi-item capacitated lot-sizing problem with setup times, safety stock and demand shortages. Süral et al. (2009) considered a lot-sizing problem with setup times where the objective is to minimize the total inventory carrying cost only. Wu et al. (2011) proposed two new mixed integer programming models for capacitated multi-level lot-sizing problems with backlogging. They proposed a new and effective optimization framework that achieves high quality solutions in reasonable computational time.

Kirca and kökten (1994) proposed a new heuristic approach for solving the single level multi-item capacitated dynamic lot- sizing problem. Their approach used an iterative item-by-item strategy for generating solutions to the problem. Özdamar and Barbarosoglu (2000) proposed a heuristic approach for the solution of the dynamic multi-level multi-item capacitated lot-sizing problem with general product structures. Rizk et al. (2006) studied a class of multi-item lot-sizing problems

* Corresponding author Email address: emehdi@qiau.ac.ir

with dynamic demands, as well as lower and upper bounds on a shared resource with a piecewise linear cost. The problem was formulated as a mixed-integer program. Absi et al. (2013) studied the multi-item capacitated lot-sizing problem with setup times and lost sales. Because of lost sales, demands can be partially or totally lost. They proposed a non-myopic heuristic based on a probing strategy and a refining procedure. They also proposed a metaheuristic based on the adaptive large neighborhood search principle to improve solutions. Gutierrez et al. (2013) investigated the dynamic lot-sizing problem considering multiple items and storage capacity. They proposed a heuristic procedure based on the smoothing technique.

Governmental and social pressures as well as economic opportunities have motivated many firms to become involved with the return of used products for recovery (Gungor and Gupta, 1999). The demands of a certain product for each period can be satisfied by items which have been either remanufactured from used products arriving at the beginning of every period, or have been newly manufactured. Golany et al. (2001) studied a production planning problem with remanufacturing. They proved the problem is NP-complete and obtained an $O(T^3)$ algorithm for solving the problem. Teunter and Pelin Bayındır (2006) addressed the dynamic lot-sizing problem for systems with product returns. They presented an exact, polynomial time dynamic programming algorithm. Li et al. (2007) analyzed a version of the capacitated dynamic lot-sizing problem with substitutions and return products. They first applied a genetic algorithm to determine all periods requiring setups for batch manufacturing and batch remanufacturing, and then developed a dynamic programming approach to provide the optimal solution to determine how many new products are manufactured or return products are remanufactured in each of these periods. Pan et al. (2009) addressed the capacitated dynamic lot-sizing problem arising in closed-loop supply chain where returned products are collected from customers. They assumed that the capacities of production, disposal and remanufacturing are limited, and backlogging is not allowed. Moreover, they proposed a pseudo-polynomial algorithm for solving the problem with both capacitated disposal and remanufacturing. Pin˜eyro and Viera (2010) investigated a lot-sizing problem with different demand streams for new and remanufactured items, in which the demand for remanufactured items can also be satisfied by new products, but not vice versa. They provided a mathematical model for the problem and demonstrated that it is NP-hard, even under particular cost structures. Zhang et al. (2012) investigated the capacitated lot-sizing problem in closed-loop supply chain considering setup costs, product returns, and remanufacturing. They formulated the problem as a mixed integer program and proposed a Lagrangian relaxation-based solution approach.

Returned products are collected from customers. These returned products can either be disposed or be remanufactured to be sold as new ones again; hence, the market demands can be satisfied by either newly produced products or remanufactured ones. Due to the variety of products in the current manner under review, each product might be produced through different manners, and the costs of each unit and the value of resources used depend on the selected manner of production. In most wide industrial implications, one of the most important questions is to identify the best value of production. In this research, an integer linear programming model was developed for the multi-item capacitated lot-size by taking into consideration many industrial limitations. The goal is to maximize the profit against costs of production, inventory costs, shortage costs, safety stock deficit costs, setup costs, out-sourcing costs, disposing returned products costs, and remanufacturing returned products cost.

The rest of this paper is organized as follows: Section 2 describes an MIP (mixed integer programing) formulation of the multi-item capacitated lot-sizing problem with safety stocks in closed-loop supply chain. The solution approach for solving the proposed model is introduced in Section 3. The Taguchi method for tuning the parameters and computational experiments are presented in Section 4. The conclusions and suggestions for future studies are included in Section 5.

2. Mathematical Formulation

Mehdizadeh and Fatehi kivi (2014) proposed an MIP model for Single-item Capacitated Lot-sizing Problem. In this section, we present an MIP formulation of the problem based on the last model.

In order to close the gap between the conditions of the problem and the real world conditions in this research, the multi-item lot-size problem has been studied with considerations of production line equilibrium limitation and capacity limitation. Not only has there been a consideration of different production manners for products, but also the model has been designed in the conditions of having safety stock and shortage being allowed. Also the factory is responsible for processing used products returned from customers. Two options are available for these returned products: remanufacturing and disposal. Remanufactured products can be sold as new ones with the same quality commitment. The main goal is to present a mathematical model to optimize production, inventory, outsourcing, shortage, remanufactured and disposal quantities as well as to determine the best production manner.

2.1. Assumptions

Before the formulation is considered, the following assumptions are made on the problem:

- The demand is considered deterministic.
- The amount of the returned products is regarded deterministic over the planning horizon.
- Shortage is backlogged.
- Shortage and inventory costs must be taken into consideration at the end.
- Raw material resource with given capacities are considered.
- The quantity of inventory and shortage at the beginning of the planning horizon is zero.
- The quantity of inventory and shortage at the end of the planning horizon is zero.

2.2. Parameters

T: Number of periods, indexed from 1 to T, involved in the planning horizon.

N: Number of products, i = 1, …, N.

J: Number of production manner, j = 1,…, J.

d_{it}: The demand for product i in the period t.

L_{it}: The quantity of the safety stock of product i in the period t.

r_{it}: The selling price per unit of product i in the period t.

C_{ijt}: The production cost of each unit of product i in the period t through the manner j.

A_{ijt}: The setup cost of the production of product i in the period t through the manner j.

h_{it}^{+} : The unit holding cost of product i in the period t.

h_{it}^{-} : Unitary safety stock deficit cost of product i in period t.

∂_{it} : Unitary shortage cost of product i in period t.

B_{kt}: The capacity of the K source at hand in the period t

α_{ik}: The quantity of the K source used by each unit of the product i.

f_{ijk}: The quantity of wasted K source for product i produced through the manner j.

γ_{it} : Unit out-sourcing cost of each unit of product i in the period t.

M_i : A large number.

τ_{ik} : The K source consumption for repair of item i.

v_i : Space needs for per unit of product i.

φ_t : The total available space in period t.

F_{it}: The cost of disposing returned products for each unit of product i in period t.

g_{it}: The cost of remanufacturing returned products for each unit of product i in period t.

θ_{it} : The unit holding cost of product i of returned products in period t.

C_{it}^{d} : The maximum number of returned products of product i that could be disposed in period t.

C_{it}^{r} : The maximum number of returned products of product i that could be remanufactured in period t.

R_{it}: the number of returned products of product i in period t.

2.3. Decision Variables

X_{ijt}: Production quantity for product i in the period t through the manner j.

X_{it}^{f} : The number of returned products of product i that remanufactured in period t.

X_{it}^{s} : The number of returned products of product i that disposed in period t.

y_{ijt}: Binary variable; 1 if the product i is produced in the period t through the manner j, otherwise y_{ijt}=0.

U_{it}: Out-sourcing level of product i in the period t.

I_{it}^{r} : The number of returned products of product i held that in inventory at the end of period t.

I_{it}^{-} : The quantity of shortage of product i in the period t.

S_{it}^{+} : The quantity of overstock deficit of product i in the period t.

S_{it}^{-} : The quantity of safety stock deficit of product i in the period t.

The objective function (1) shows the difference between selling price with the shortage costs, inventory costs, disposing costs, remanufacturing costs, production costs, safety stock costs and outsourcing costs. Constraints (2) are the inventory flow conservation equations through the planning horizon. Constraints (3) and (4) define, respectively, the demand shortage and the safety stock deficit for item *i* at the end period is zero. Constraints (5) are the inventory flow conservation equations for returned products. Constraints (6) are the capacity constraints; the overall consumption must remain lower than or equal to the available capacity. If we produce an item *i* at period *t*, then constraints (7) impose that the quantity produced must not exceed a maximum production level M_{it}. M_{it} could beset to the minimum between the total demand requirements for item *i* on section [*t*, *T*] of the horizon and the highest quantity of item *i* that could be produced regarding the capacity constraints, M_{it} is then equal Eq. (17) to:

2.4. The proposed Model

$$Max Z = \sum_{t=1}^{T}\left(\sum_{i=1}^{N}\left(r_{it}\left(\sum_{j=1}^{J}X_{ijt}+X_{it}^{s}+U_{it}\right)-\sum_{j=1}^{J}\left(C_{ijt}X_{ijt}+A_{ijt}y_{ijt}\right)-\partial_{it}I_{it}^{-}-h_{it}^{+}S_{it}^{+}-h_{it}^{-}S_{it}^{-}-\gamma_{it}U_{it}-F_{it}^{f}X_{it}-g_{it}^{s}X_{it}-\theta_{it}I_{it}\right)\right) \quad (1)$$

s.t:

$$S_{i,t-1}^{+}-S_{i,t-1}^{-}-I_{i,t-1}^{-}+I_{i,t}^{-}+\sum_{j=1}^{J}X_{ijt}+X_{it}^{s}+U_{it}=S_{it}^{+}-S_{it}^{-}+d_{it}+L_{it}-L_{i,t-1}\ \forall\ i=1,2,....,N, t=1,2,....,T \quad (2)$$

$$S_{iT}^{+}=0 \quad (3)$$

$$I_{i,T}^{-}=0 \quad (4)$$

$$I_{it}^{r}=I_{i,t-1}^{r}-X_{it}^{f}-X_{it}^{s}+R_{it}\ \forall\ t=1,2,....,T,\quad i=1,2,....,N \quad (5)$$

$$\sum_{i=1}^{N}\left(\sum_{j=1}^{J}(\alpha_{ik}X_{ijt}+f_{ijk}y_{ijt})+\tau_{ik}X_{it}^{s}\right)\le B_{kt}\ \forall\ k=1,2,....,K,\ t=1,2,....,T \quad (6)$$

$$X_{ijt}\le My_{ijt}\ \forall\ j=1,2,....,J,\ i=1,2,....,N,\ t=1,2,....,T \quad (7)$$

$$I_{it}^{-}\le d_{it}\ \forall\ i=1,2,....,N\ ,\ t=1,2,....,T-1 \quad (8)$$

$$S_{it}^{-}\le L_{it}\ \forall\ i=1,2,....,N\ ,\ t=1,2,....,T \quad (9)$$

$$0\le U_{it}\le I_{i,t-1}^{-}+S_{i,t-1}^{-}+d_{it}+L_{it}\ \forall\ i=1,2,....,N\ ,\ t=1,2,....,T \quad (10)$$

$$X_{it}^{f}\le C_{it}^{d}\ \forall\ i=1,2,....,N\ ,\ t=1,2,....,T \quad (11)$$

$$X_{it}^{s}\le C_{it}^{r}\ \forall\ i=1,2,....,N\ ,\ t=1,2,....,T \quad (12)$$

$$\sum_{i=1}^{N}v_{i}\left(\sum_{j=1}^{J}X_{ijt}+X_{it}^{S}+U_{it}\right)\le\varphi_{t}\ \forall\ t=1,2,....,T \quad (13)$$

$$y_{ijt}\in\{0,1\}\ \forall\ i=1,2,....,N,\ j=1,2,....,J,\ t=1,2,....,T \quad (14)$$

$$X_{ijt},X_{it}^{f},X_{it}^{s},I_{it}^{r},I_{it}^{-},S_{it}^{-},S_{it}^{+}\ge 0\ \forall\ i=1,2,....,N\ ,j=1,2,....,J\ ,t=1,2,....,T \quad (15)$$

$$X_{ijt},X_{it}^{f},X_{it}^{s},I_{it}^{r},I_{it}^{-},S_{it}^{-},S_{it}^{+}\ \text{integer} \quad (16)$$

$$M_{it}=Min\left(\frac{B_{kt}-f_{ijk}}{\alpha_{ijk}},\sum_{t=1}^{T}d_{it}\right) \quad (17)$$

Constraints (8) and (9) define upper bounds on, respectively, the demand shortage and the safety stock deficit for item *i* in period *t*. Constraints (10) ensure that outsourcing level U_{it} at period t is nonnegative and cannot exceed the sum of the demand, safety stock of period t and the quantity backlogged, safety stock deficit from previous periods. Constraints (11) and (12) are the capacity constraints of disposal, remanufacturing. Constraints (13) are the Maximum space available for storage of items in excess. Constraints (14) and (15) characterize y_{ijt} is a binary variable and the variable's domains: $X_{ijt},X_{it}^{f},X_{it}^{s},I_{it}^{r},I_{it}^{-},S_{it}^{-},S_{it}^{+}$ are non-negative and integer for $i\in N, j\in J$ and $t\in T$.

3. Solution Approaches

3.1. Simulated Annealing Algorithm

Simulated annealing (SA) was presented by Kirkpatrick et al. (1983). The SA methodology draws its analogy from the annealing process of solids. In the annealing process, a solid is heated to a high temperature and gradually cooled to a low temperature to be crystallized. As the heating process allows the atoms to move randomly, if the cooling is done too rapidly, it gives the atoms enough time to align themselves in order to reach a minimum energy state that is named stability or equipment. This analogy can be used in combinatorial optimization in which the state of solid corresponds to the feasible solution; the energy at each state corresponds to the improvement in the

objective function and the minimum energy state will be the optimal solution.
The steps of SA algorithm are shown below:
Step 1: Generating feasible initial solution. $X_{best} = X_0$
Step 2: Initializing the algorithm parameters which consist of initial temperatures (T_0), rate of the current temperature decreases (α), max of iteration at each temperature (L), freezing temperature (T_f), in this paper $T_f = 0$.
Step 3: Calculating the objective value $C(X_0)$ for initial solution.
Step 4: Initializing the internal loop
In this step, the internal loop is carried out for S =1 and repeated while S < L.
Step 5: Neighborhood generation
Step 6: Accepting the new solution
Set $\Delta C = C(X_n) - C(X)$ Now, if $\Delta C \le 0$, accept the new solution, else if $\Delta C > 0$ generate a random number r between (0, 1);
If $r < 1 - e^{\left(\frac{-\Delta C}{T_0}\right)}$, then accept a new solution; otherwise, reject the new solution and accept the previous solution.
If S ≥ L, go to step 7; otherwise S +1 $\rightarrow$ S and go back to step 5.
Step 7: Adjusting the temperature
In this step, $T_0 = T \times \alpha$ is used for reducing temperature at each iteration of the outer cycle of the algorithm. If $T_0 = T_f$ return to step 8; otherwise, go back to step 4.
Step 8: Stopping criteria.
Two important issues that need to be defined when adopting this general algorithm to a specific problem are the procedures to generate both initial solution and neighboring solutions.

3.1.1. Representation schema

To design simulated annealing optimization algorithm for the mentioned problem, a suitable representation scheme that shows the solution characteristics is required. In this paper, the general structure of the solution representation performed for running the simulated annealing for four periods with two production manners is shown in Figure 1.
Number of bets in Part 1= Product number × Number of periods × Number of production manner
Number of bets in Part 2= Product number × Number of periods

Y_{111}	Y_{112}	Y_{113}	Y_{114}	Y_{211}	Y_{212}	Y_{213}	Y_{214}
0	1	1	0	0	1	1	0

X^s_{11}	X^s_{12}	X^s_{13}	X^s_{14}	X^s_{21}	X^s_{22}	X^s_{23}	X^s_{24}
2	2	0	1	3	0	1	2

Fig. 1. Solution representation

3.1.2. Neighborhood scheme

At each temperature level a search process is applied to explore the neighborhoods of the current solution. In this paper we use mutation scheme for produce neighborhood solution. Figure 2 illustrates this operation where there are four periods, two products and one production manner. In each iteration, we produce two new solutions from two old solutions. At first for new product (Part A), a chromosome is selected, one bit was randomly selected and the number of bit was changed to zero if it was one and was changed to one if it was zero, then, for remanufacturing products (Part B), one bit is selected randomly and its quantity changed across the authority range.

In part A, new products are shown; in this part one bit was randomly selected and the number of bit was changed. In part B remanufacturing products are shown; in this section all bits in chromosome are changing numbers, for per bit randomly selected from Authority range for each remanufacturing product.

Part A:

Y_{111}	Y_{112}	Y_{113}	Y_{114}	Y_{211}	Y_{212}	Y_{213}	Y_{214}
0	1	1	0	0	1	1	0
⇩							
Y_{111}	Y_{112}	Y_{113}	Y_{114}	Y_{211}	Y_{212}	Y_{213}	Y_{214}
0	1	1	0	0	1	1	0

Part B:

X^s_{11}	X^s_{12}	X^s_{13}	X^s_{14}	X^s_{21}	X^s_{22}	X^s_{23}	X^s_{24}
2	2	0	1	3	0	1	2
⇩	⇩	⇩	⇩	⇩	⇩	⇩	⇩
X^s_{11}	X^s_{12}	X^s_{13}	X^s_{14}	X^s_{21}	X^s_{22}	X^s_{23}	X^s_{24}
1	1	3	2	0	1	2	0

Fig. 2. An example of the neighborhood structure

3.1.3. Cooling schedule scheme

The temperature is another basic characteristic of the SA which is gradually decreased when the algorithm progressed. Initially, T is set to a high value T_i, and it can be reduced with some patterns at each step of algorithm. The cooling schedule with $T_i = \alpha \times T_{i-1}$ (where α is the cooling factor constant and belong to (0 1)) is considered as cooling pattern for this research.

3.2. Vibration Damping Optimization

Recently, a new heuristic optimization technique based on the concept of the vibration damping in mechanical vibration was introduced by Mehdizadeh and Tavakkoli-

Moghaddam (2009), named vibration damping optimization (VDO) algorithm. They already utilized the algorithm to solve parallel machine scheduling problem. The VDO algorithm is illustrated in the following steps:

Step 1: Generating feasible initial solution.

Step 2: Initializing the algorithm parameters which consist of: initial amplitude (A_0), max of iteration at each amplitude (l_{max}), damping coefficient (γ), and standard deviation (σ). Finally, parameter t is set in one (t=1)

Step 3: Calculating the objective value U_0 for initial solution.

Step 4: Initializing the internal loop

In this step, the internal loop is carried out for l =1 and repeated while $l < l_{max}$.

Step 5: Neighborhood generation.

Step 6: Accepting the new solution

Set $\Delta = U - U_0$ Now, if $\Delta < 0$, accept the new solution, else if $\Delta > 0$ generate a random number *r* between (0, 1);

If $r < 1 - \exp\left(\frac{-A_t^2}{2\sigma^2}\right)$, then accept a new solution; otherwise, reject the new solution and accept the previous solution.

If $l > l_{max}$, then $t+1 \rightarrow t$ and go to step 7; otherwise $l+1 \rightarrow l$ and go back to step 5.

Step 7: Adjusting the amplitude

In this step, $A_t = A_0 \exp(\frac{-\gamma t}{2})$ is used for reducing amplitude at each iteration of the outer cycle of the algorithm. If $A_t = 0$ return to step 8; otherwise, go back to step 4.

Step 8: Stopping criteria

In this step, the proposed algorithm will be stopped after the predetermined number of iterations. At the end, the best solution is obtained.

3.2.1. Representation schema

To design vibration damping optimization algorithm for the mentioned problem, a suitable representation scheme that shows the solution characteristics is needed. In this paper, the general structure of the solution representation performed for running the vibration damping for two products, four periods with one production manner is shown in Fig 1.

3.2.2. Neighborhood scheme

In this paper we use mutation scheme, Figure 2 illustrates this operation on the two products, four periods with one production manner.

3.3. Harmony Search algorithms

Harmony search (HS), proposed by Geem et al. (2001), is a new heuristic method that mimics the improvisation of music players. Inspiration was drawn from musical performance processes that occur when a musician searches for a better state of harmony, improvising the instrument pitches towards a better aesthetic outcome. The HS algorithm imposes fewer mathematical requirements and does not require specific initial value settings of the decision variables (Yadav et al, 2012) (2012). Because the HS algorithm is based on stochastic random searches, the derivative information is also not necessary. In the HS algorithm, musicians search for a perfect state of harmony determined by aesthetic estimation, as the optimisation algorithms search for the best state (i.e., global optimum) determined by an objective function. Each musician corresponds to a decision variable; a musical instrument's pitch range corresponds to a range of values for the decision variables; musical harmony at a certain time corresponds to a solution vector certain iteration; and an audience's aesthetics correspond to the objective function. Just as musical harmony is incrementally improved, a solution vector is also improved iteration by iteration. To understand the design principle of the HS algorithm, let us first idealize the improvisation process adopted by a skilled musician. When a musician is improvising, he or she has three possible choices: (1) playing any famous tune exactly from his or her memory, (2) playing something similar to the aforementioned tune (thus adjusting the pitch slightly) or (3) composing new or random notes. In this section, various steps of the HS algorithm and a description of how the HS is designed and applied are presented.

Step 1: Initialize the optimisation problem and algorithm parameters to apply HS, in the first step, the optimization problem are specified as follows:

Minimize (or Maximize) f(x) (17)

Subject to $x_i \in X_i$, $i = 1, 2, ..., N$

where f (x) is an objective function to be optimized, x is a solution vector composed of decision variables, x_i e X_i is the set of possible range of values for each decision variable x_i (continuous decision variable), that is ${}_Lx_i \leq Xi \leq {}_Ux_i$, where ${}_Lx_i$ and ${}_Ux_i$ are the lower and upper bounds for each decision variable, respectively, and N is the number of decision variables. Furthermore, the control parameters of HS are specified in this step. These parameters are the harmony memory size (HMS), harmony memory consideration rate (HMCR), and pitch adjusting rate (PAR).

Step 2: In the second step, each component of each vector in the parental population (harmony memory) is initialized with a uniformly distributed random number

between the upper and lower bounds $[_{L}x_i, {}_{U}x_i]$, Where 1 < i < N. The ith component of the jth solution vector is given by

x_i^j = Rand [possible range of values for x_i]

Where j = 1, 2, 3, . . ., HMS. Each row consists of a randomly generated solution vector for the formulated optimization problem, and the objective function value for the jth solution vector is denoted by $f(x^j)$. The matrix formed is governed by

HM (j, 1: N) = x^j

HM (j, N + 1) = f (x^j)

The HM with the size of HMS × (N + 1) can be represented by a matrix, as:

$$HM=\begin{bmatrix} X^1 \\ X^2 \\ \vdots \\ \vdots \\ X^{HMS} \end{bmatrix}=\begin{bmatrix} X_1^1 & \cdots & X_N^1 & f(x^1) \\ X_1^2 & \cdots & X_N^2 & f(x^2) \\ \vdots & \cdots & \vdots & \vdots \\ X_1^{HMS} & \cdots & X_N^{HMS} & f(x^{HMS}) \end{bmatrix} \quad (18)$$

Step 3: Improvise a new harmony from the HM

after defining the HM as shown in Equation 15; for the optimization problem, the improvisation of the HM is performed by generating a new harmony vector $x' = (x_1', x_2', \ldots\ldots\ldots, x_N')$. Each component of the new harmony vector is generated using

$$x_i' \leftarrow \begin{cases} x_i' \in HM(i) & \text{With probability HMCR} \\ x_i' \in X_i & \text{With probability (1 - HMCR)} \end{cases} \quad (19)$$

Where HM (i) is the ith column of the HM, HMCR is defined as the probability of selecting a component from the HM members, and (1 - HMCR) is, therefore, the probability of generating a component randomly from the possible range of values. If x_i' is generated from the HM, then it is further modified or mutated according to PAR. PAR determines the probability of a candidate from the HM mutating, and (1 - PAR) is the probability of no mutation. Here the pitch adjustment for the selected x_i' is given by

$$x_i' \leftarrow \begin{cases} x_i' = \text{Rand}\left\{x_i' \in X_i\right\} & \text{With probability PAR} \\ x_i' & \text{With probability (1 - PAR)} \end{cases} \quad (20)$$

Step 4: Update the HM

The newly generated harmony vector (x_0) is evaluated in terms of the objective function value. If the objective function value for the new harmony vector is better than the objective function value for the worst harmony in the HM, then the new harmony is included in the HM and the existing worst harmony are excluded from the HM.

Step 5: If the stopping criterion (maximum number of improvisations) is satisfied, computation is terminated. Otherwise, steps 3 and 4 are repeated.

4. Results

In this paper, all tests are conducted on a notebook with Intel Core i5 Processor 2.53 GHz and 4 GB of RAM and the proposed algorithms; namely, SA, VDO and HS are coded in Visual Basic 2000.

4.1. Parameter calibration

Appropriate design of parameters has a significant impact on efficiency of meta-heuristics. In this paper, Taguchi method (Taguchi, 2000) was applied to calibrate the parameters of the proposed methods; namely, SA, VDO and HS algorithms. This method is based on maximizing performance measures called signal-to-noise ratios in order to find the optimized levels of the effective factors in the experiments. The S/N ratio refers to the mean-square deviation of the objective function that minimizes the mean and variance of quality characteristics to make them closer to the expected values. For the factors that have a significant impact on S/N ratio, the highest S/N ratio provides the optimum level for that factor. As pointed out earlier, the purpose of Taguchi method is to maximize the S/N ratio. In this subsection, the parameters for experimental analysis are determined.

Table 1 lists different levels of the factors for SA, VDO and HS. In this paper, according to the levels and the number of the factors, respectively, the Taguchi method L_9 is used for the adjustment of the parameters for the SA and HS and L_{27} is used for the VDO.

Table 1
Factors and their level

Factor	Algorithm	Notation	Level	Value
Initial temperature	SA	T_0	3	800, 1000, 1200
Number of iteration at each temperature		L	3	40, 60, 80
Rate cooling		α	3	0.9, 0.95, 0.99
Initial amplitude	VDO	A_0	3	6, 8, 10
Max of iteration at each amplitude		l_{max}	3	20, 30, 40
Damping coefficient		γ	3	0.05, 0. 1, 0.5
Stopping criteria		t_{max}	3	400, 600, 800
standard deviation		σ	3	0.5, 1, 1.5
Pitch-adjusting rate	HS	PAR	3	0.1, 0.3, 0.5
Harmony memory considering rate		HMCR	3	0.1, 0.7, 0.9
harmony memory size		HMS	3	20, 30, 40
Stopping criteria		**STOP**	**3**	**50, 100, 150**

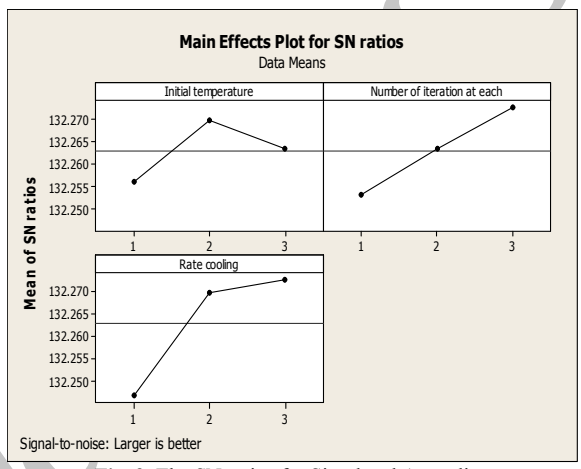

Fig. 3. The SN ratios for Simulated Annealing

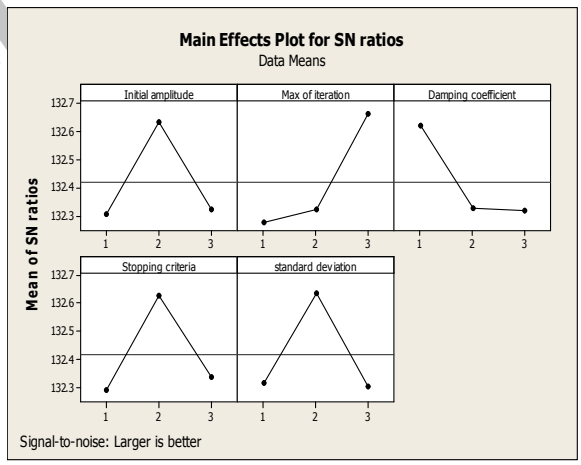

Fig. 4. The SN ratios for Vibration Damping

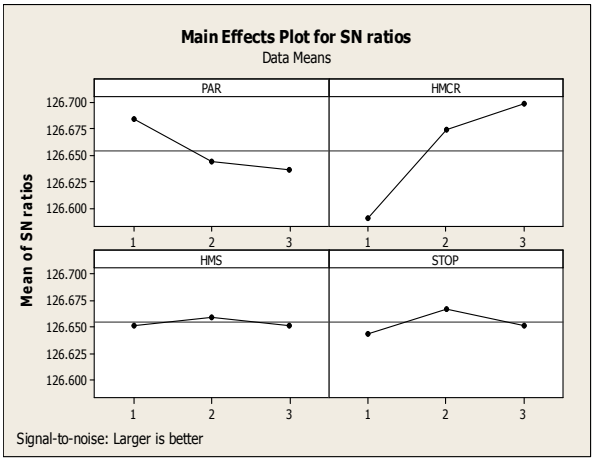

Fig. 5. The SN ratios for Harmony Search

Figures 3, 4 and 5 show S/N ratios. According to these figures, 1000, 80, 0.99, 8, 40, 0.05, 600, 1, 0.1, 0.9, 30 and 100 are the optimal level of the factors T_0, L, α, A_0, l_{max}, γ, t_{max}, σ, PAR, HMCR, HMS and STOP.

4.2. Computational results

Computational experiments were conducted to validate and verify the behavior and the performance of the meta-heuristic algorithms employed to solve the considered multi-item capacitated lot-sizing problem with safety stocks in closed-loop supply chain. We tried to test the performance of the SA, VDO and HS in finding good quality solutions in reasonable time for the problem. For this purpose, 30 problems with different sizes are generated. These test problems are classified into three classes: small size, medium size and large size.

The number of manners, products and periods have the most impact on problem hardness. The proposed model coded with Lingo (ver.8) software using for solving the instances. The approaches are implemented to solve each instance in five times to obtain more reliable data. The best results are recorded as a measure for the related problem. Table 2 shows details of computational results obtained by solution methods for all test problems.

The results of running SA, VDO and HS are compared with the optimal solution of the instances, obtained from Lingo software, in rows 1 to 10 of Table 2. The presented statistical analyses (the variance analysis outcome) were reported for problems with small, medium, and large dimensions, in Tables 3, 4, and 5. According to the values of the survey (or *P-Value*), we can conclude that the HS showed its usefulness in different problems as compared to the SA and VDO, and statistical results also are significantly different for problems with medium and large dimensions. To clarify the matter further, confidence distances for different sizes are illustrated in Figures 6, 7 and 8.

In addition, Figure 9 depicts the comparison between solution quality of the SA and HS of the instances. Figure 10 depicts the comparison between solution quality of the VDO and HS of the instances. Figure 11 depicts the comparison between solution quality of the SA and VDO of the instances. A general review of the results illustrated in Tables 3, 4 and 5, and Figures 6, 7 and 8 reveal that:

- ✓ The SA, VDO and HS can solve all test problems.
- ✓ The computational time required to solve problems with SA is smaller than HS.
- ✓ The SA, VDO and HS can find good quality solutions for small size problems.
- ✓ The objective values obtained by HS are also better than SA and VDO results.
- ✓ For small size test problems, VDO algorithm has to find good quality solutions. However, its results will be worse when the problem size increases.
- ✓ The objective values obtained by VDO and HS are closer to each other and are also better than SA results.
- ✓ The computational time required to solve problems with SA is smaller than VDO.
- ✓ The objective values obtained by HS are better than VDO and SA results when the problem size increases.

Table 2
Details of computational results for all test problems

No	Class	P.M.T	Objective Function Value (OFV)							
			Lingo	T	SA	T	VDO	T	HS	T
۱		2.2.3	1366849	0	1373991	9	1366849	13	1366849	7
۲		2.2.5	2170470	0	2170470	11	2170470	22	2170470	15
3		2.3.5	2180288	0	1983830	11	2165478	21	2150365	17
4		5.3.5	4692306	0	3726211	27	4136639	47	3959751	18
5	Small	5.2.6	5497372	0	4052009	32	4921411	51	4845100	27
6		5.3.6	5621934	42	4386065	31	4836306	51	4613509	26
7		5.3.8	7696358	55	7696358	41	7696358	61	7696358	31
9		6.4.11	14610520	90	13980417	68	13868797	80	14030655	81
11		6.4.12	15545650	508	14879737	72	14746709	85	14841150	80
10		6.4.15	19599720	1134	19599720	88	19599720	118	19599720	91
11		6.4.20	----------	----	24370398	114	25422376	127	25924788	126
12		5.7.20	----------	----	23131730	95	23616685	106	24137117	129
13		5.5.27	----------	----	21976304	120	30252478	160	31345533	154
14		6.5.18	----------	----	22144555	106	25028195	143	25635024	157
15	Medium	5.5.20	----------	----	21159175	90	23028121	119	23556950	159
16		7.6.18	----------	----	24256977	135	28400108	154	29415116	165
17		5.5.24	----------	----	22477120	106	27500560	151	28637867	168
18		6.5.17	----------	----	21473228	101	23464365	134	24299040	172
19		6.4.17	----------	----	21470110	101	23775301	135	24260212	176
20		5.5.30	----------	----	26205622	139	34801172	174	36133719	178
21		8.6.18	----------	----	30261061	159	33088207	171	33718440	192
22		6.8.30	----------	----	26402966	174	41647320	220	43398938	205
23		7.5.25	----------	----	26722266	186	39377083	191	40581142	228
24		8.6.20	----------	----	31393122	172	36621457	198	37354221	243
25	Large	6.4.24	----------	----	27781380	135	33130214	285	34223254	250
26		8.6.24	----------	----	28169230	208	43837475	220	45056702	237
27		7.6.20	----------	----	28965910	141	31786688	167	32914558	241
28		6.4.27	----------	----	30352647	153	362572206	188	37497164	249
29		7.6.24	----------	----	28441443	169	38667467	206	39612415	283
۳۰		8.6.30	----------	----	51742301	260	55903827	263	58126552	406

— Means that a feasible solution has not been found after 3600 s of computing time.

$C_{jt} \in [60,85]$; $A_{jt} \in [20000,30000]$; $d_t \in [1,10]$; $r_t \in [45000,70000]$; $L_t \in [1,4]$; $\partial_t \in [14,19]$; $h_t^- \in [12,16]$; $h_t^+ \in [7,10]$; $\gamma_t \in [30000,45000]$; $V = 1$; $\varphi_t = [20,100]$; $B_{kt} = [12,300]$

Table 3
Analysis of variance for small size problem

Source	DF	SS	MS	F	P
Small Size	2	1.61382E+11	80691069755	0.00	0.998
Error	27	1.09951E+15	4.07224E+13		
Total	29	1.09967E+15			

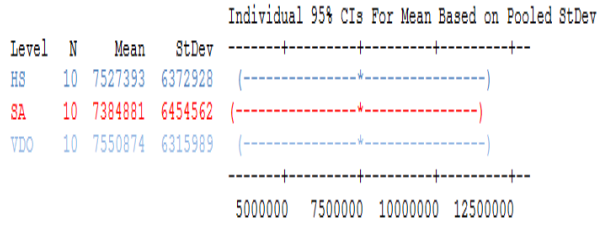

Fig. 6. The output of analysis of variance for small size test problem

Table 4
Analysis of variance for medium size problem

Source	DF	SS	MS	F	P
Medium Size	2	1.13418E+14	5.67090E+13	5.12	0.013
Error	27	2.99191E+14	1.10811E+13		
Total	29	4.12609E+14			

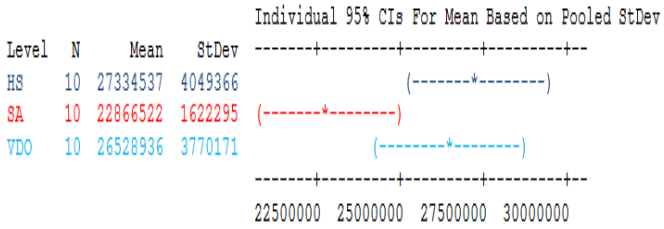

Fig. 7. The output of analysis of variance for medium size test problem

Table 5
Analysis of variance for large size problem

Source	DF	SS	MS	F	P
Large Size	2	5.02440E+14	2.51220E+14	4.67	0.018
Error	27	1.45281E+15	5.38078E+13		
Total	29	1.95525E+15			

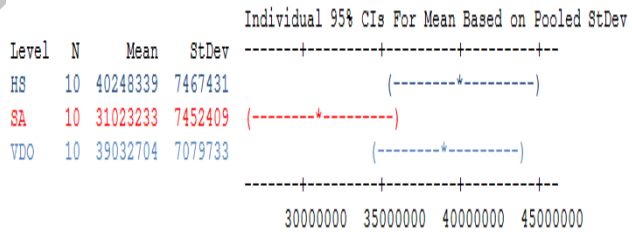

Fig. 8. The output of analysis of variance for large size test problem

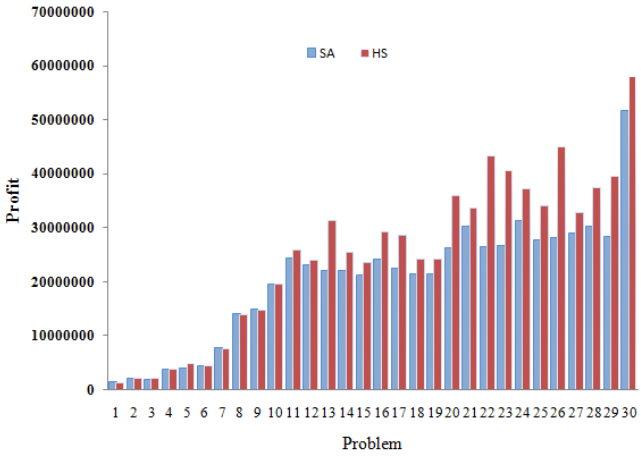

Fig. 9. Comparison between solution quality of the HS and SA

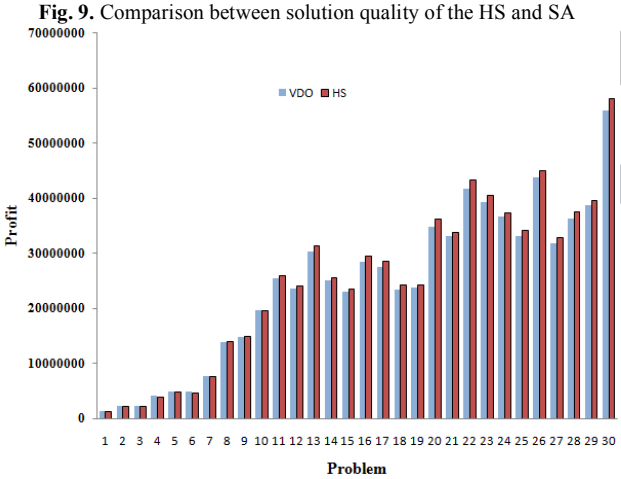

Fig. 10. Comparison between solution quality of the HS and VDO

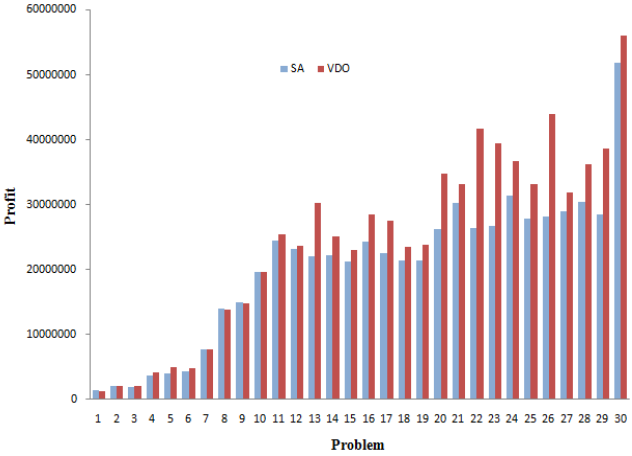

Fig. 11. Comparison between solution quality of the SA and VDO

5. Conclusion

In this paper, we proposed a mathematical formulation for a new multi-item capacitated lot-sizing problem with setup times in closed-loop supply chain. This formulation takes into account several industrial constraints such as shortage costs, safety stock deficit costs, limited outsourcing and return products. Due to the complexity of the problem, three meta-heuristic algorithms named simulated annealing (SA) algorithm, vibration damping optimization (VDO) algorithm and harmony search (HS) algorithm were used to solve problem instances. Additionally, an extensive parameter setting with performing Taguchi method was conducted for selecting the optimal levels of the factors that affect algorithm's performance. For showing the performance of the proposed algorithms, 30 problems with different sizes were generated. These test problems were classified into

three classes: small size, medium size and large size. The comparison between the proposed algorithms and results obtained from Lingo software showed the efficiency of the algorithms. Also, the three algorithms were statically compared. The objective values obtained by HS are better than VDO and SA results when the problem size increases. One straightforward opportunity for future research is extending the assumption of the proposed model for including real conditions of production systems such as limited inventory, fuzzy demands, etc. Also, developing a new heuristic or meta-heuristic to construct better feasible solutions is recommended.

References

[1] Absi, N., Detienne, B., Dauzère-Pérès, S. (2013). Heuristics for the multi-item cpacitated lot-sizing problem with lost sales, Computers & Operations Research, 40, 264-272.

[2] Absi, N., Kedad-Sidhoum, S. (2009). The multi-item capacitated lot-sizing problem with safety stocks and demand shortage costs. Computer and Operations Research, 36, 2926–2936.

[3] Chen, WH., Thizy, JM. (1990). Analysis of relaxations for the multi-item capacitated lot-sizing problem. Annals of Operations Research, 26, 29–72.

[4] Geem, Z.W., Kim, J.H., Loganathan, G. (2001). A new heuristic optimization algorithm: harmony search. Simulation, 76, 60–8.

[5] Golany, B., Yang, J., Yu, G. (2001). Economic lot-sizing with remanufacturing options. IIE Transactions, 33, 995-1003.

[6] Gungor, A., Gupta, S.M., (1999). Issues in environmentally conscious manufacturing and product recovery: a survey. Computers and Industrial Engineering 36, 81–853.

[7] Gutierrez, J. Colebrook, M. Abdul-Jalbar, B. Sicilia, J. (2013). Effective replenishment policies for the multi-item dynamic lot-sizing problem with storage capacities. Computer & Operations Research. 40, 2844-2851.

[8] Kirca, ö. Kökten, M. (1994). A new heuristic approach for the multi-item dynamic lot sizing problem. European Journal of Operational Research. 75. 332-341.

[9] Mehdizadeh, E., Fatehi Kivi, A. (2014). Three Meta-heuristic Algorithms for the Single-item Capacitated Lot-sizing Problem, International Journal of Engineering, 27,1223-1232.

[10] Mehdizadeh, E., Tavakkoli-Moghaddam, R (2009). Vibration damping optimization algorithm for an identical parallel machine scheduling problem. Proceeding of the 2nd International Conference of Iranian Operations Research Society, Babolsar, Iran. May 20- 22.

[11] Özdamar, L., Barbarosoglu, G. (2000). An integrated Lagrangean relaxation-simulated annealing approach to the multi-level multi-item capacitated lot sizing problem. International Journal of Production Economics. 68, 319-331.

[12] Pan, Z., Tang , J. Liu, O. (2009).Capacitated dynamic lot sizing problems in closed-loop supply chainv. European Journal of Operational Research, 198, 810-821.

[13] Pin˜eyro, P., Viera, O. (2010). The economic lot-sizing problem with remanufacturing and one-way substitution. IIE Transactions, 124, 482–488.

[14] Rizk, N., Martel, A., Ramudhin, A. (2006). A lagrangean relaxation algorithm for multi-item lot-sizing problems with joint piecewise linear resource costs. International Journal of Production Economics, 102, 344-357.

[15] S. Kirkpatrick, C. Gelatt and M. Vecchi. (1983). Optimization by simulated annealing, Science, 220, 671-680.

[16] Süral, H., Denizel, M., Van Wassenhove, LN. (2009). Lagrangean relaxation based heuristics for lot-sizing with setup times. European Journal of Operational Research, 194, 51–63.

[17] Taguchi, G, Chowdhury, S. (2000). Taguchi, Robust Engineering. McGraw-Hill, New York.

[18] Teunter, R. H., Pelin Bayındır, Z., van den Heuvel, W. (2006). Dynamic lot sizing with product returns and remanufacturing. International Journal of Production Research, 44, 4377-4400.

[19] Wu,T., Shi, L., Geunes, J., . Akartunal, K. (2011). An optimization framework for solving capacitated multi-level lot-sizing problems with backlogging. European Journal of Operational Research, 214, 428–441.

[20] Y. Li, Chen, J., Cai, X. (2007). Heuristic genetic algorithm for capacitated production planning problems with batch processing and remanufacturing. Int. J. Production Economics, 105, 301–317.

[21] Yadav, P., Kumar, R., Panda, S.K., Chang, C.S. (2012). An Intelligent Tuned Harmony Search algorithm for optimisation. Information Sciences, 196, 47–72.

[22] Zhang, Z.H., Jiang, H., Pan, X. (2012). A Lagrangian relaxation based approach for the capacitated lot sizing problem in closed-loop supply chain. Int. J. Production Economics, 140, 249-255.