Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Simulation optimization of an inventory control model for a reverse logistics system

Hanane Rachih[a*], Fatima Zahra Mhada[a] and Raddaoune Chiheb[a]

[a]*Rabat IT Center, ENSIAS, Mohammed V University, Rabat, Morocco*

CHRONICLE

Article history:
Received May 5, 2021
Received in revised format:
June 1, 2021
Accepted September 16 2021
Available online
September 16, 2021

Keywords:
Inventory Control
Stochastic Optimization
Reverse Logistics
Simulation
Metaheuristics
Design of Experiments

ABSTRACT

Nowadays, companies are recognizing their primordial roles and responsibilities towards the protection of the environment and save the natural resources. They are focusing on some contemporary activities such as Reverse Logistics which is economically and environmentally viable. However, the integration of such an initiative needs flows restructuring and supply chain management in order to increase sustainability and maximize profits. Under this background, this paper addresses an inventory control model for a reverse logistics system that deals with two separated types of demand, for new products and remanufactured products, with different selling prices. The model consists of a single shared machine between production and remanufacturing operations, while the machine is subject to random failures and repairs. Three stock points respectively for returns, new products and remanufactured products are investigated. Meanwhile, in this paper, a modeling of the problem with Discrete-Event simulation using Arena® was conducted. Regarding the purpose of finding, a near-optimal inventory control policy that minimizes the total cost, an optimization of the model based on Tabu Search and Genetic Algorithms was established. Computational examples and sensitivity analysis were performed in order to compare the results and the robustness of each proposed algorithm. Then the results of the two methods were compared with those of OptQuest® optimization tool.

1. Introduction

Supply Chain Management (SCM) encompasses all the resources, means, methods, tools and techniques that drive all processes and flows to transform raw materials into final products deliverable to customers. The SCM is considered a cooperation system that assumes the active streamlining of the supply-side business's activities in order to gain the customers satisfaction and maximize profits. Across the SCM, the coordination and integrating decisions of the important functions such as production planning, procurement, inventory control and distribution management may lead to obtaining an optimal strategy that minimizes total costs for the entire company for a given service level (Simchi-Levi et al., 2004). It is a matter of anticipating needs and being able to deliver the right product, where it is necessary, when necessary, while ensuring control of quantity and quality. A product is never made by the same company from the start to the end, many suppliers and third-party logistics are involved in different stages of the product's manufacturing. When these companies do not communicate with each other, supply can be out of order at any time, and production flow can be slowed or even interrupted. In addition, to prevent the inevitable production riskiness's (machine breakdowns, transportation problems, stock-out) there are few solutions to build up. As one of these solutions, inventory management consists of an important core component of supply chain management. Inventory management remains in maintaining stock levels, controlling and supervising purchases from suppliers and customers, controlling the quantity of products for sales and customer demand fulfillment. In terms of cost, the highest single cost in a supply chain is that of inventory as it represents about 50% of the total logistics costs (Lancioni, 2000). Therefore, it's necessary to focus on inventory management and production planning in order to minimize the total cost. Nowadays, supply chain management is becoming more complicated, while the modern industries are focusing on some

* Corresponding author. Tel.: +212655616201
E-mail address: rachih.hanane@gmail.com (H. Rachih)

doi: 10.5267/j.dsl.2021.9.001

contemporary activities which are economically and environmentally feasible such as Reverse Logistics. Regarding sustainable supply chain management, Reverse Logistics (RL) is considered as a business strategy in which recovery activities serve to increase sustainability (Ayvaz et al., 2015). RL addresses significant environmental problems and creates competitive opportunities in the market. It includes the reuse of returned and used products from the customers and then try to give a new life for the End-of-Life (EOL) products through different recovery processes (Dev et al., 2019). These recovery processes can occur on repair, refurbishing, remanufacturing, cannibalization, recycling and incineration (De Brito & Dekker, 2003).

The inventory is used to regularize the production flow in order to satisfy the customer demands and to avoid stock-out or overproduction (Mhada et al., 2013). But integration of recovery processes into the supply chain complicates more the production planning and the inventory management. This complexity is due to uncertainties associated with the returns that include the timing of returns, the quantity of returns, and the quality of returns (Akçali & Çetinkaya, 2011). Thus, keeping an eye on the inventory and managing the stock levels in a reverse supply chain is an issue that interests both industrial managers and scientific researchers. Many articles address the problem of production planning and inventory control for reverse logistics systems, one can refer to some literature reviews for mathematical models in (Moritz et al., 1997) and recently in (Bazan et al., 2016). A first deterministic model of inventory management was proposed early by Schrady (1967). This model deals with a constant rate for returns and demands, fixed lead-times and setup cost for external orders and recovery, and linear holding cost for new and recovered products. Schrady (1967) presented a policy where each procurement order is succeeded by ***n*** identical recovery batches with aims to obtain the optimal manufacturing and recovery quantities. Schrady's model was considered in several research papers, such as in (Mabini et al., 1992) where a variant of the previous model with a constraint for stock-out service level is proposed. Richter (1994) studied the relation of the cost function with the return rate for a Schrady's model with one recovery batch. The last was extended in (Richter, 1996) and (Richter, 1997) for multiple consecutive procurement and recovery batches. Teunter (2001) extended Schrady's work by considering different holding costs for new products and recovered products for a model with multiple production and repair cycles. Other variants of the model were studied later by Teunter and Vlachos (2002) where stochastic demands and return rates and discounted cost were considered. Jaber and El Saadany (2009) differentiated between the quality of new products and remanufactured products and assumed two types of demand corresponding to each type of products, also a backlog of demands and substitution of a new product with a remanufactured are considered in the model.

Kenne et al. (2012) proposed a production planning policy for a reverse logistics system which takes as decision and state variables the production rates and the stock levels for optimizing the cost of the system. Cobb (2016) studied an inventory control model for Returnable Transport Items (RTI) where production requirements are satisfied by a combination of the new and repaired RTI. The performance of the model is studied regarding the effect of the purchase, inspection, repair, and transportation of RTI on the costs of distributing products to customers. The optimal solution is obtained obligatory when inspection and repair runs begin simultaneously. Recently, Ahiska et al. (2017) treats an inventory control model for a hybrid manufacturing-remanufacturing system with product substitution under stochastic demand and returns. They assume that remanufactured and new products have different costs and different selling prices and they propose three heuristic policies and develop a two-phase local search algorithm to find the optimal values of the parameters for these policies.

We remark that inventory management for reverse logistics systems has received great interest in the literature and the majority of these studies assume that recovered products have the same quality as of new products and that the demands can be fulfilled by both types of products. On the other hand, only a few studies, where it is considered that the quality is different of the two types and they can be sold separately, can be found. In the light of the previous works, the present paper analyzes an inventory model that considers arrival of different types of demands for new or for remanufactured products. We propose a control policy which differs from the existing models because our model deals with a single shared machine for production that is subject to random failures and repairs. Three stocks respectively for returns, new products and remanufactured products are investigated. Demands and returns are stochastic, and the backorders are permitted. The main purpose is to find a near-optimal control policy which determines the optimal levels for the three stocks for a minimum total cost using metaheuristics methods for optimization and comparison of the results and the performance of each method.

The study of a problem involves some specific steps in order to successfully conduct the study and achieve the objectives. The following briefly describes the basic steps in the research methodology adopted. The first step concerns the problem definition and the model formulation (Notations, assumptions, decision variables and objective function). In the second step, a simulation model has been built, verified and validated. The first attempt to optimize this model is by using the OptQuest® tool, then developing and conducting the design of experiments, regression analysis was used to validate the outputs of the simulation. In parallel, an optimization of the model with a developed Tabu Search algorithm and then with a Genetic Algorithm was conducted to get a nearly optimal and useful solution for the research problem. Finally, we compare and evaluate the performance of those previously adopted approaches.

The remainder of the paper is organized as follows. Firstly, we describe the problem and define the proposed model, assumptions and conditions considered for the optimization of the model in section 2. In section 3, we define the method used to model the problem and the optimization approaches adopted to find a near-optimal solution with the implementation of the two metaheuristics to the inventory problem. The evaluations and validation of the performance of the proposed methods are illustrated in section 4. A sensitivity analysis on the GA results and numerical examples are presented in section 5. Finally, conclusions and suggestions for future research are presented in section 6.

2. Problem description and formulation

In this section we will present the framework, assumptions and the notations used for the proposed inventory control model for the hybrid manufacturing remanufacturing system.

2.1 Model framework

In this paper, we study an inventory control model for a production system including remanufacturing of returned products. The remanufacturing operation begins after the acquisition of returned products from a collection center. The production system faces two types of demands: new product demands which are satisfied with manufacturing of raw materials and demands of recovered products that are fulfilled with remanufacturing of returns. The considered model is of a stochastic nature, because the demands and return arrivals are stochastic. We assume that a single machine denoted M is flexible to perform only one of the operations at the same time: Manufacturing or Remanufacturing or can be in stop mode. Furthermore, machine M is subject to random breakdowns and repairs. The state of the system is described by the continuous variables $x_0(t)$, $x_1(t)$ and $x_2(t)$ corresponding respectively to the levels of the stocks of returns, new products and remanufactured products. In the operational mode, when the level of returns stock $x_0(t)$ reached a threshold z_0, the machine M switch to the remanufacturing mode until the stock of remanufactured products $x_2(t)$ reached z_2. Else the machine performs the manufacturing operations until $x_1(t)$ reached the threshold z_1. Otherwise, the machine will be in the stop mode.

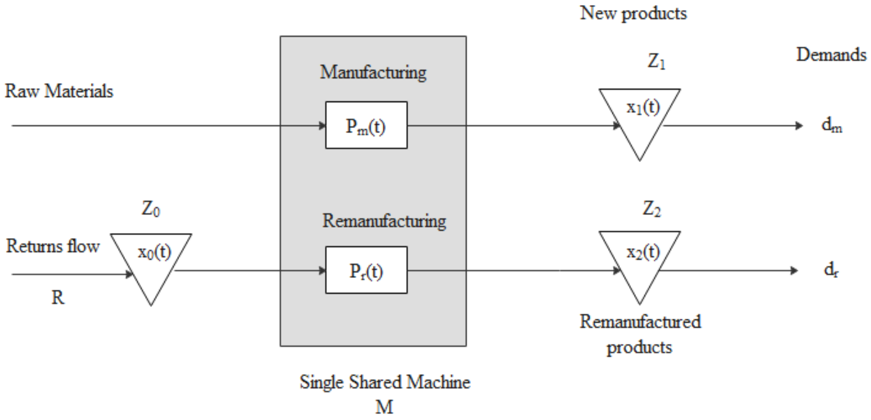

Fig. 1. Production system with recovery option

2.2 Model Notations

In the following, we introduce the notations used throughout this article.

$\boldsymbol{x_0(t)}$: the level of the stock of returns.
$\boldsymbol{x_1(t)}$: the level of the stock of new products.
$\boldsymbol{x_2(t)}$: the level of the stock of remanufactured products.

$\boldsymbol{d_m(t)}$: new products demand. It follows Poisson distribution with the mean $\boldsymbol{d_m^{av}}$, and time between two arrivals follows Exponential distribution with a mean equal to 2 hours.

$\boldsymbol{d_r(t)}$: remanufactured products demand. It follows Poisson distribution with the mean $\boldsymbol{d_r^{av}}$, and time between two arrivals follows Exponential distribution with a mean equal to 3 hours.

$\boldsymbol{R(t)}$: the return flow is made up by the arrival of a constant quantity R and a time between two arrivals follows Exponential distribution with a mean equal to 20 hours.

$\boldsymbol{hc_i}$: holding cost for stock i, with $\boldsymbol{i} \in \{\boldsymbol{0, 1, 2}\}$ (\$/product/unit of time).

$\boldsymbol{bc_i}$: backlog cost for new and remanufactured products, with $\boldsymbol{i} \in \{\boldsymbol{1, 2}\}$ (\$/product/unit of time).

$\boldsymbol{pm(t)}$: production rate of the manufacturing system.

$\boldsymbol{pr(t)}$: production rate of the remanufacturing system.

$\boldsymbol{pm_{max}}$: maximal production rate for the manufacturing system.

$\boldsymbol{pr_{max}}$: maximal production rate for the remanufacturing system.

$\boldsymbol{z_i}$: threshold level for $\boldsymbol{x_i}$, with $\boldsymbol{i} \in \{\boldsymbol{0}, \boldsymbol{1}, \boldsymbol{2}\}$.

2.3 Model Assumptions

In order to facilitate the modeling of the system, some assumptions are made through this study by focusing on the most important factors:

- The model uses a continuous time horizon.
- All types of customer demands must be satisfied, otherwise they are backlogged and a shortage cost bc_i is applied.
- The switching times and costs for the machine are neglected.
- All types of products are considered compliant.
- The backlog cost bc_i is significantly higher than the holding cost hc_i, for stock $i \in \{1,2\}$.

The system's stability is ensured by assuming:

$$\frac{d_m^{av}}{pm_{max}} + \frac{d_r^{av}}{pr_{max}} \prec 1 \qquad (1)$$

The state of the considered system is described by a continuous Markov process $\xi(t)$ with two states for the machine, $On = 1$ for operational mode and $Off = 0$ for stop mode.

$$\xi(t) = \begin{cases} 1, \; if\ M\ is\ operational \\ 0, \quad if\ M\ is\ stopped \end{cases} \qquad (2)$$

The dynamics of the machine are illustrated using the states transition diagram in Fig. 2. Such that the transition rates from operational mode to stop mode are denoted:

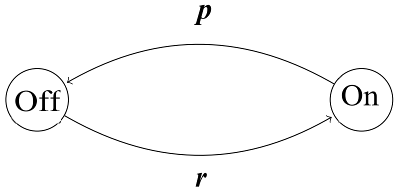

Fig. 2. States transition diagram of the machine

p: Failure rate;

r: Repair rate;

The state of the system is defined also by the continuous components x_0, x_1 and x_2. the following differential equations describes the dynamics of the system:

$$\begin{cases} \dfrac{dx_0(t)}{dt} = R(t) - pr(t) \\ \dfrac{dx_1(t)}{dt} = pm(t) - d_m(t) \\ \dfrac{dx_2(t)}{dt} = pr(t) - d_r(t) \end{cases} \qquad (3)$$

With

$$pm(t) = \begin{cases} pm_{max}, if\ (x_0 < z_0\ and\ x_1 \leq z_1)\ or\ (x_0 \geq z_0\ and\ x_1 \leq z_1\ and\ x_2 \geq z_2) \\ 0, \quad if(x_1 > z_1) \\ 0, \quad if\ \ (M\ is\ failed) \end{cases} \tag{4}$$

$$pr(t) = \begin{cases} pr_{max}, \quad if\ (x_0 \geq z_0\ and\ x_2 \leq z_2) \\ 0, \quad if\ (x_2 > z_2) \\ 0, \quad if\ (\ M\ is\ failed\ or\ x_0 = 0) \end{cases} \tag{5}$$

The control problem studied here is about to find the optimal triplet $(z_0{}^*, z_1{}^*, z_2{}^*)$ which minimizes the long-term average total cost of storage and production shortages $J_T\big(x_0(t), x_1(t), x_2(t)\big)$. More precisely, the cost to be minimized over stock levels parameters z_i is given in Eq. (6):

where $x^+(t) = max(x(t), 0)$ and $I(:)$ is the indicator function.

$$J_T\big(x_0(t), x_1(t), x_2(t)\big) = \lim_{T\to\infty} \frac{1}{T} \left(\sum_{i=0}^{2} E\left[\int_0^T h\, c_i\, x_i^+(t)dt \right] + \sum_{i=1}^{2} E\left[\int_0^T b\, c_i\, I(x_i(t) = 0)dt \right] \right) \tag{6}$$

3. Simulation & Optimization

3.1 Simulation

In this section, the previously described model for reverse logistics system has been implemented in a simulation model using Discrete-Event Simulation (DES). DES is considered one of the important tools for complex systems modeling and experiments performing (Rossetti, 2015). It is known as a very effective way to estimate and evaluate the performance of a planned flow line. Several software packages with graphical user interfaces allow the planner to easily model a system at an arbitrary level of detail (Dias et al., 2016). A great degree of modeling flexibility in terms of probability distributions and other details about the line's mode of operations is offered by DES. However, the modeling of a flow line is easier than a systematic optimization of this model design (Helber et al., 2011). To facilitate the simulation modeling, several software exists to offer the advantage of tracing complex system processes (considering any distribution functions and the complex relationship between the stations, elements, components, and sections) providing timely information on operating characteristics. Between those tools, Arena® (Version 16.00.00002; System 64 bit; RAM 4,00 GB) has been adopted in this research to develop the simulation model.

3.2 Optimization Methods

According to previous analysis and studies, Tabu Search and Genetic Algorithm are considered one of the most flexible metaheuristics for Reverse Logistics problems (Rachih et al., 2019). Thus, to tackle with the inventory control problem, we proposed two approaches based on the Tabu Search algorithm and Genetic Algorithm in order to find the best combination of the z_i for a minimal total cost and to compare the results of the two methods.

3.2.1 Tabu Search

Tabu search (TS) is one of the wide spread single-solution based meta-heuristics that was introduced by Glover (1989) and developed in detail by Glover and Laguna (1998) as a local based search strategy that accepts non-improving moves to escape from local optima. As the local search is based on moves to explore the neighborhood of solutions, TS generates a neighborhood solution from the current solution and accepts the best solution even if it is worse than the current solution in order to explore a larger fraction of the search space. The particular feature of the TS is the use of a short-term memory called tabu list to prevent the cyclic movements. The tabu list maintains information about the recently visited solution to avoid some solutions being revisited. According to an aspiration criterion, if a solution is present in the tabu list and can decrease the objective function below the minimal level, thus, it can be revisited and accepted.

3.2.1.1 Algorithm description

The variables used for the TS algorithm are introduced as below:

S_0: initial solution
S_c: current solution
S_{Nbest}: best neighbor solution
C_{Nbest}: best neighbor cost
S_{best}: best solution
C_{best}: best cost
$List_N$: list of neighbor candidates
$List_{tabu}$: tabu list of visited candidates

The solutions of the inventory problem are presented by the triplet composed of the stock levels (z_0, z_1, z_2). The initial solution in the TS is given randomly. Starting from the initial solution S_0, the set of the neighbors N_c of each current solution is examined in each iteration because the number of neighbors is known and not large. The list of the neighbors $List_N$ is provided by addition and subtraction of a random signed number to each z_i of the current solution as illustrated in the example below:

If the current solution is $S_c = (z_0, z_1, z_2)$. Then the list of neighbors will be

$$\begin{aligned} List_N = \{&N_1(z_0 + r, z_1, z_2); N_2(z_0 - r, z_1, z_2); N_3(z_0, z_1 + r, z_2); \\ &N_4(z_0, z_1 - r, z_2); N_5(z_0, z_1, z_2 + r); N_6(z_0, z_1, z_2 - r)\} \end{aligned}$$

with r = random signed number.

To examine the list of neighbors, TS calls the simulation model to calculate the fitness value which is the cost corresponding to each neighbor of the current solution recorded in a simulation list. The neighbor with the minimum cost is chosen as the new current solution even if it is a non improving solution. After all, neighbor candidates are stored on the tabu list $List_{tabu}$. The best neighbor solution is compared to the best global solution. The termination of the TS algorithm is equal to the maximum of the number of the iterations (replications).

The pseudo-code of the implemented Tabu Search algorithm is given in the following:

```
**Algorithm 1:** Tabu Search Algorithm
Choice of the initial solution $S_0$
Initialization of the variables
$List_{tabu} \leftarrow \varnothing$
$S_c \leftarrow S_0$
$S_{best} \leftarrow S_0$
       **While** ($i < Max$ of iterations) **do**
           Generation of the list of neighbors $List_N$ for the current solution $S_c$
      **Repeat**
           Simulation of the Neighbor candidate in Arena®
           Computation of the total cost
      **Until** All neighbors are examined
           Selection of the best Neighbor $S_{Nbest}$
          Update of the tabu list $List_{tabu} \leftarrow List_{tabu} + List_N$
                **if** ( $f(S_{Nbest}) < f(S_{best})$ ) **then**
                 $S_{best} \leftarrow S_{Nbest}$
                 $C_{best} \leftarrow C_{Nbest}$
                **end if**
      $S_c \leftarrow S_{Nbest}$
      $i \leftarrow i + 1$
      **end while**
**Return** $S_{best}$
```

3.2.1.2 Implementation of TS

By using Arena, we investigated all the possible combinations of the factor levels, (z_0, z_1, z_2) which are generated by the Tabu Search algorithm to simulate the total cost. The design adopted for this experiment consists of the steps as below:

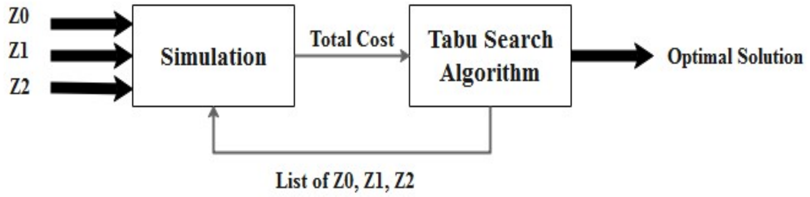

Fig. 3. Factors and response

- Definition of the response variables for simulation: the results of the total cost function given in the Eq. (6).
- Selection of the response variables for Tabu Search algorithm: the best cost and the neighbors of the current solution
- Choice of factors: we consider the triplet of the independent factors that influence the performance of our system, z_0, z_1, z_2. A set of six combinations of the three factors is generated by Tabu Search after simulation and valuation of the previous set. The initial set is given randomly.
- Number of replications: 200 combinations of the three factors were simulated in this study and each combination was simulated 5 times.
- Replication Length: 150.000 hrs.

For all the numerical scenarios, we set $5 \leq z_i \leq 50$ and the values of the constant parameters as illustrated on Table 1.

Table 1
The constant parameters value

Parameters	hc_i	bc_i	d_m^{av}	d_r^{av}	pm_{max}	pr_{max}	R	r	p
Value	2	20	5	3	14	10	5	0.5	0.05

After simulation of all replications, the combination of Tabu Search algorithm and simulation gives a best total cost equal to **82.53** for the values of stock levels $z_0^* = 5,\ z_1^* = 12,\ z_2^* = 23$.

3.2.2 Genetic Algorithm

Genetic Algorithm (GA) is considered one of the most powerful optimization methods for combinatorial problems. It was proposed by John Holland (1975), and then developed by Goldberg (1989). The GA is a metaheuristics method based on the principles of the natural selection and it belongs to the class of Evolutionary Algorithms (EA). An EA approach is considered generally as an iterative, population-based approach that uses selection and random variation to generate new solutions (Ghosh et al., 2011).

The basic process of GA is based on three operations inspired by biological evolution: selection, reproduction and mutation. To begin with, an initial random population is generated and composed of the individual solutions to the research problem. This population evolves from one generation to another by repeating the three operations: selection, crossover, mutation. Selection of the best individuals by their fitness function. Then the chromosomes of the selected parents are combined to create new child using the crossover operator. And to expand the search space, the mutation operator is used to modify a random gene of the child individuals.

3.2.2.1 Algorithm description

In this section, we describe the basic characteristics and components of the GA employed in this research.

Generally, GA starts by generating randomly an initial population composed of Pop_{size} individuals (z_0, z_1, z_2). The fitness (cost function) of every individual is calculated and evaluated to select the best ones with the roulette wheel selection for recombination. These selected individuals will be recombined by a crossover operator who chooses them according to a probability P_{cross}. The generated child can be mutated by a mutation operator with a probability of P_{mut} in order to maintain the diversity from generation to another. After the selection and recombination (crossover and mutation), a new population is generated and composed of individuals who have good chances to be stronger than those of the previous generation. From generation to generation, these steps are repeated as many times until the stopping criterion is satisfied.

The different steps of the algorithm are presented in detail in the following and resumed in the Fig. 4.

- Encoding individual: a population is a set of individuals and each individual is defined by a chromosome. The efficiency of the Genetic Algorithm will depend on the choice of the coding of a chromosome. In this research, a direct coding is used because the chromosome will present the solution (z_0, z_1, z_2) of the problem and the genes of the chromosome are the levels of the three stocks z_i

$\boldsymbol{z_0}$	$\boldsymbol{z_1}$	$\boldsymbol{z_2}$

- Initial population: the used mechanisms for generating the initial population are a uniform function that generates randomly the combination of (z_0, z_1, z_2) until the population size. A condition on the z_i is integrated in the function to avoid duplicated individuals in the population and to guarantee a diversity in the initial population.

- Evaluation fitness: an evaluation function is used to measure the performance of each individual. This function allows to evaluate the capacity of an individual to survive by assigning a weight called fitness to it. While the objective function of our problem is the cost function defined previously in Eq. (6), then we choose it to evaluate the strength of the individuals.

- Selection method: selection is used to identify individuals that can be recombined in a population. There are several selection techniques among them Roulette Wheel selection. This consists of creating a biased lottery wheel for which each individual in the population occupies a selection on the wheel proportional to its fitness value. If we consider max_{cost} as the maximum value of the cost function of the individuals, the $cost_{indiv}$ as the cost of each individual and the sum_{cost} as the sum of all the costs values of the individuals in the population. The probability of selection $prob_{indiv}$ is then calculated as:

$$prob_{indiv} = (max_{cost} - cost_{indiv})/sum_{cost} \tag{7}$$

- Crossover operator: crossover allows to enrich the population by manipulating the components of the chromosomes. The crossover considered two parents and generates one or two children and it is applied with a probability P_{cross}. After selecting two individuals (P_1, P_2) and a random number $r \in [0,1]$, we test if $r \leq P_{cross}$ then the arithmetic (barycentric) crossover operator is applied on the parents in the position (i) and two children are generated, as below:

$$\begin{aligned} Child_1(i) &= \alpha \times P_1(i) + (1-\alpha) \times P_2(i) \\ Child_2(i) &= (1-\alpha) \times P_1(i) + \alpha \times P_2(i) \end{aligned} \tag{8}$$

with $\alpha \in [0.5,1]$

- Mutation operator: mutation operator is considered in the process of GA with a low probability, in order to introduce small modifications to some solutions in each population and to guarantee the diversity from generation to other. The mutation operator in the proposed GA is applied with a probability P_{mut} on two different positions in the generated child.

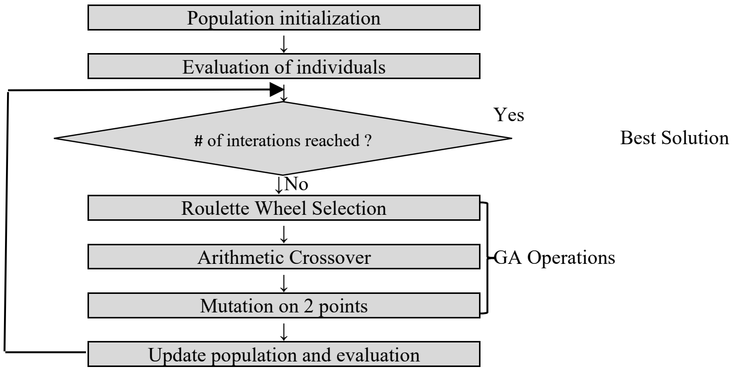
Population initialization
Evaluation of individuals
# of interations reached ?
Yes
Best Solution
No
Roulette Wheel Selection
Arithmetic Crossover
Mutation on 2 points
GA Operations
Update population and evaluation

Fig. 4. Steps of Genetic Algorithm

3.2.3 *Tuning of GA Parameters*

One of the difficulties of using a Genetic Algorithm consists in the choice of many parameters that control it: size of the population, number of generations or stopping criterion, crossover and mutation probabilities. It is worth tuning those parameters to find the best settings for the problem studied and to increase the efficiency of the GA. To this end, we applied a factorial design (FD) in Minitab software for the parametrization of the GA and to determine the factors which have the largest effect on the GA performance. We conducted many experiments of the 2^kFD, where k is the number of the factors and *2* means two levels (Low, High) of each factor. While *4* factors (Pop_{size}, Num_{sim}, P_{cross}, P_{mut}) are studied for the GA, thus $2^4 = 16$ trials are simulated in each experiment. The simulation of all the performed experiments and the comparison of the regression statistics of the models generated in Minitab software, leads to find a best experiment with $S = 0.0725$; $RSquare = 99.96\%$; $RSquare(Adj) = 99.38\%$; $RSquare(Pred) = 89.38\%$, which means that this model fits well the data and we can use it to determine the good parameters setting for the present problem. The low and high levels of the GA parameters and the 16 combinations of the best experiment with their corresponding responses, iteration number and run times needed to find the minimal cost are presented respectively in Table 2 and Table 3. We completed the column of the responses in Minitab software after simulating all the combinations in Arena and analyzed the FD and optimized it. The key outputs of the analysis, the Pareto chart (Fig. 5) and the main effect plot for the cost function (Fig. 6), are generated and illustrated below:

Table 1
The levels of GA parameters

Parameter	Low level	High level
Pop_{size}	40	60
Num_{sim}	200	400
Pcross	0.8	0.85
Pmut	0.03	0.05

Table 2
The simulated trials and responses

Trial	Pop_{size}	Num_{sim}	*Pcross*	*Pmut*	Response (Cost)	Iteration	Run time (s)
1	60	200	0.8	0.05	83.70	197	6895
2	40	400	0.8	0.05	83.60	265	9275
3	60	200	0.8	0.03	84.47	197	6895
4	60	200	0.85	0.03	86.05	40	1400
5	40	400	0.8	0.03	82.54	360	12 600
6	40	200	0.8	0.03	84.71	39	1365
7	60	200	0.85	0.05	85.06	146	5110
8	40	400	0.85	0.05	84.71	39	1365
9	60	400	0.85	0.03	84.46	268	9380
10	40	200	0.85	0.05	84.71	39	1365
11	40	200	0.85	0.03	84.71	39	1365
12	40	400	0.85	0.03	84.56	228	7980
13	60	400	0.85	0.05	83.45	387	13 545
14	60	400	0.8	0.03	83.06	266	9310
15	40	200	0.8	0.05	84.71	39	1365
16	60	400	0.8	0.05	82.89	379	13 265

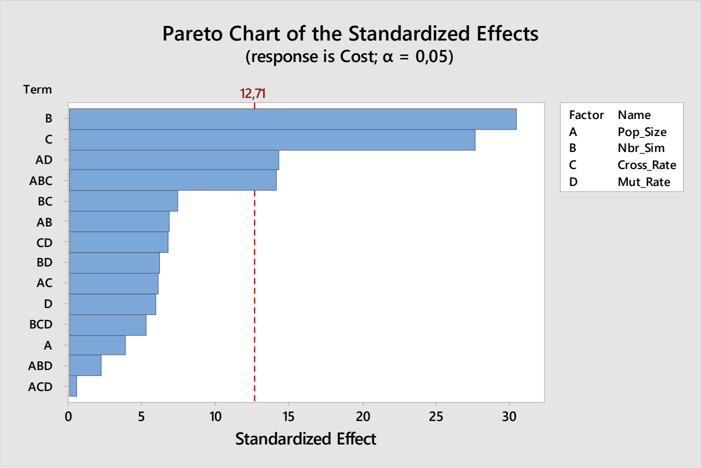

Fig. 5. Pareto Chart

From the effect plots and the Pareto chart, we can conclude that the simulation number and the crossover probability have the largest effect on the GA performance. Thus, a higher simulation number and a crossover probability smaller than 0.85 lead to a good solution. The variation of the mutation probability and the population size have a small effect on the solution. We found an optimal setting with Minitab of the GA parameters located at $Pop_{size}^{*} = 40$, $Num_{sim}^{*} = 400$, $P_{cross}^{*} = 0.8$, $P_{mut}^{*} = 0.03$. With this setting we get a minimum cost equal to **82.54** for $z_0^* = 5, z_1^* = 11, z_2^* = 15$.

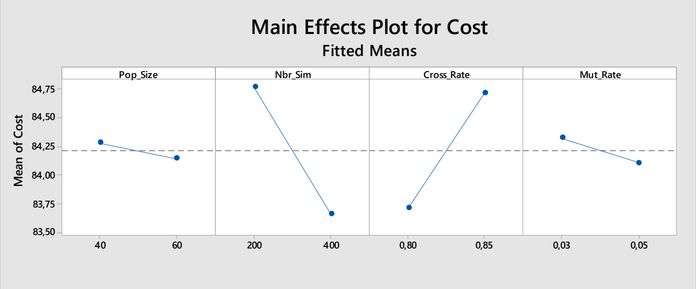

Fig. 6. Main effect plot for Cost function

4. Comparison of the optimization methods

In this section, we compare the performance of the three optimization methods, TS algorithm, GA algorithm and OptQuest® for Arena®.

OptQuest® for Arena® is an Arena's inbuilt optimization tool that allows you to search for an optimal solution within the simulation models built in Arena. OptQuest uses a combination of some powerful heuristics in a one algorithm that is capable of finding an optimal or near optimal solution for complex problems. As the objective of this study is to find the optimal solution of inventory control variables (z_0, z_1, z_2), we used the OptQuest tool to evaluate the performance of the two proposed methods (Tabu search and Genetic Algorithm). We exploited the simulation model developed in Arena for the Optquest tool. OptQuest is simple to use, after establishing the model in Arena, the OptQuest takes over the execution of the Arena model. Once the control variables (inputs) are selected and the objective function and constraints are defined for the optimization problem, a different set of control values is called by Arena to be evaluated in order to find the optimal value of the cost. As shown in Table 4, a comparison between the Tabu Search algorithm, Genetic Algorithm and the OptQuest was conducted based on the outcomes and the total of simulations. We can remark that the gap between the best total cost obtained by OptQuest and the two metaheuristics TS and GA solutions is about 0.30 while the total of simulations needed to obtain a good solution for OptQuest and GA is greater than that of TS algorithm. But in term of stock levels or number of iterations, the GA solution remain the best because it gave low levels of stocks and it found the solution in less iterations.

Table 3
Comparison of Tabu search & Genetic Algorithm & OptQuest results

Optimization method	Tabu Search	Genetic Algorithm	OptQuest
Best cost value	82.53	82.54	82.83
Best control $(\boldsymbol{z_0^*}, \boldsymbol{z_1^*}, \boldsymbol{z_2^*})$	(5, 12, 23)	(5, 11, 15)	(5, 11, 29)
Total simulations	200	400	446
Number of iterations	33	10	-

5. Sensitivity Analysis

In this section, we performed a sensitivity analysis on the fixed values for the simulation runs presented previously in Table 1. We changed only the holding cost $\boldsymbol{hc_i}$ and the backlog cost $\boldsymbol{bc_i}$ and kept the other parameters constant. We explored all the variations and compared them to the basic case founded in the previous section, as illustrated in Table 5.

Table 4
The sensitivity analysis table

$\boldsymbol{hc_i}$	$\boldsymbol{bc_i}$	$\boldsymbol{z_0^*}$	$\boldsymbol{z_1^*}$	$\boldsymbol{z_2^*}$	Cost	Remark
2	20	5	11	15	82.54	Basic case
2	30	6	12	50	110.07	z_2^* increase
2	10	6	9	39	57.26	z_1^* decrease, z_2^* increase
3	20	9	10	42	108.94	z_1^* decrease, z_2^* increase
3	30	5	12	28	124.05	$z^*{}_2$ increase
3	10	7	11	46	77.45	z_0^* increase, z_2^* increase
1	20	6	12	50	67.72	z_2^* increase
1	30	6	18	18	94.47	$z^*{}_1$ increase, z_2^* increase
1	10	6	11	49	42.30	z_2^* increase

Through the above Table, it appears that results make sense which demonstrate the robustness of the proposed method. The levels of the stocks are adapted with the variation the holding hc_i and backlog cost bc_i. Thus, we can distinguish four control policies depending on the variation of the costs:

- If $\boldsymbol{bc_i}$ increases: when the backlog cost increases, it is clear from the numerical examples that the levels of the stocks increase, especially the stock of the remanufactured products z_2 in order to avoid backlogs and to ensure availability of products for future demands.

- If $\boldsymbol{bc_i}$ decreases: in this case the stock level of new products decreases to avoid the overstocks and further holding costs.

- If $\boldsymbol{hc_i}$ increases: when the holding cost rise, the results show that the levels of stocks decrease to confine further holding costs.

- If $\boldsymbol{hc_i}$ decreases: same as the first case, it appears that the stock levels increase to make products available and to avoid backlog costs.

6. Conclusion

This paper investigated a reverse logistics system that deals with two types of customer demands, one for new products and the second for remanufactured products with different selling prices. This system consists of a single shared machine between production and remanufacturing operations, while the machine is subject to random failures and repairs. Demands and returns are considered stochastic and the backlog of demands is permitted. In the presented work, the decision variables studied are the levels of the three respectively stocks (x_0, x_1, x_2) for returned products, new products and the remanufactured products. The purpose of the study was to find an optimal policy which gives a near optimal combination of the levels of the three stocks in order to minimize the total cost of the system. Then to compare the performance of a metaheuristics method to find the best policy. Therefore, we used the Discrete Event Simulation (DES) to model the presented problem and to perform the experiments in Arena®. Near optimal solutions were obtained using two approaches based on Tabu Search algorithm and Genetic Algorithm. The results of the two metaheuristics are compared to the solution of another optimization tool (OptQuest®) and then a sensitivity analysis was performed to present the robustness of the GA. This study points out that the management of the inventory levels plays a significant role in minimization of the total cost in the reverse logistics systems. In this paper, the optimal control policy consists of minimal levels for stocks of returns and new products and higher levels for remanufactured products while the demand of new products is higher than demand of remanufactured products and the backlog cost is also higher than the holding cost. Regarding sensitivity analysis, the optimal policy varies with the holding and backlogs costs variations. There are several ways to extend this study. While the set-up cost and the possibility to dispose of the returns are not considered in this paper, it will be interesting to include these constraints into the model. The consideration of separate holding cost and backlog cost for new products and remanufactured products can make the model more realistic. In addition, trying other types of the encoding of the individuals and the tuning of the Genetic Algorithm parameters can help in improving the performance of the algorithm. The analysis of different scenarios and numerical tests for the model can lead to other optimal policies. The optimization of the model can be conducted using other metaheuristics, or hybridization of multiple metaheuristics and other optimization methods.

References

Ahiska, S. S., Gocer, F., & King, R. E. (2017). Heuristic inventory policies for a hybrid manufacturing/remanufacturing system with product substitution. *Computers \& Industrial Engineering*, *114*, 206–222.

Akçali, E., & Çetinkaya, S. (2011). Quantitative models for inventory and production planning in closed-loop supply chains. *International Journal of Production Research*, *49*(8), 2373–2407.

Ayvaz, B., Bolat, B., & Aydın, N. (2015). Stochastic reverse logistics network design for waste of electrical and electronic equipment. *Resources, Conservation and Recycling*, *104*, 391–404. https://doi.org/10.1016/j.resconrec.2015.07.006

Bazan, E., Jaber, M. Y., & Zanoni, S. (2016). A review of mathematical inventory models for reverse logistics and the future of its modeling: An environmental perspective. *Applied Mathematical Modelling*, *40*(5–6), 4151–4178. https://doi.org/10.1016/j.apm.2015.11.027

Cobb, B. R. (2016). Inventory control for returnable transport items in a closed-loop supply chain. *Transportation Research Part E: Logistics and Transportation Review*, *86*, 53–68.

De Brito, M. P., & Dekker, R. (2003). A framework for reverse logistics. ERIM report series research in Management. *Erasmus Research Institute of Management*.

Dev, N. K., Shankar, R., & Swami, S. (2019). Diffusion of green products in industry 4.0: Reverse logistics issues during design of inventory and production planning system. *International Journal of Production Economics*, *223*, 107–519.

Dias, L. M. S., Vieira, A. A. C., Pereira, G. A. B., & Oliveira, J. A. (2016). Discrete simulation software ranking; A top list of the worldwide most popular and used tools. *2016 Winter Simulation Conference (WSC)*, 1060–1071.

https://doi.org/10.1109/WSC.2016.7822165

Ghosh, T., Sengupta, S., Chattopadhyay, M., & Dan, P. (2011). Meta-heuristics in cellular manufacturing: A state-of-the-art review. *International Journal of Industrial Engineering Computations*, *2*(1), 87–122.

Glover, F. (1989). Tabu search—part I. *ORSA Journal on Computing*, *1*(3), 190–206.

Glover, F., & Laguna, M. (1998). Tabu search. In *Handbook of combinatorial optimization* (pp. 2093–2229). Springer.

Helber, S., Schimmelpfeng, K., Stolletz, R., & Lagershausen, S. (2011). Using linear programming to analyze and optimize stochastic flow lines. *Annals of Operations Research*, *182*(1), 193–211.

Jaber, M. Y., & El Saadany, A. M. A. (2009). The production, remanufacture and waste disposal model with lost sales. *International Journal of Production Economics*, *120*(1), 115–124.

Kenne, J.-P., Dejax, P., & Gharbi, A. (2012). Production planning of a hybrid manufacturing--remanufacturing system under uncertainty within a closed-loop supply chain. *International Journal of Production Economics*, *135*(1), 81–93.

Lancioni, R. A. (2000). New Developments in Supply Chain Management for the Millennium. *Industrial Marketing Management*, *29*(1), 1–6. https://doi.org/https://doi.org/10.1016/S0019-8501(99)00106-6

Mabini, M. C., Pintelon, L. M., & Gelders, L. F. (1992). EOQ type formulations for controlling repairable inventories. *International Journal of Production Economics*, *28*(1), 21–33.

Mhada, F. Z., Malhamé, R. P., & Pellerin, R. (2013). Joint assignment of buffer sizes and inspection points in unreliable transfer lines with scrapping of defective parts. *Production \& Manufacturing Research*, *1*(1), 79–101.

Moritz, F., Bloemhof-Ruwaard, J. M., der Laan, V., Van Nunen, E., Jo, A. E. E., & Van Wassenhove, L. N. (1997). Quantitative models for reverse logistics: A review. *European Journal of Operational Research*, *103*(1), 1–17.

Rachih, H., Mhada, F. Z., & Chiheb, R. (2019). Meta-heuristics for reverse logistics: A literature review and perspectives. *Computers & Industrial Engineering*. https://doi.org/10.1016/j.cie.2018.11.058

Richter, K. (1996). The EOQ repair and waste disposal model with variable setup numbers. *European Journal of Operational Research*, *95*(2), 313–324.

Richter, K. (1997). Pure and mixed strategies for the EOQ repair and waste disposal problem. *Operations-Research-Spektrum*, *19*(2), 123–129.

Richter, K. (1994). An EOQ repair and waste disposal mode. *Eighth International Working Seminar on Production Economics*, 83–91.

Rossetti, M. D. (2015). *Simulation modeling and Arena*. John Wiley \& Sons.

Schrady, D. A. (1967). A deterministic inventory model for reparable items. *Naval Research Logistics Quarterly*, *14*(3), 391–398.

Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (2004). *Managing The Supply Chain: Definitive Guide*. Tata McGraw-Hill Education.

Teunter, R. H. (2001). Economic ordering quantities for recoverable item inventory systems. *Naval Research Logistics (NRL)*, *48*(6), 484–495.

Teunter, R. H., & Vlachos, D. (2002). On the necessity of a disposal option for returned items that can be remanufactured. *International Journal of Production Economics*, *75*(3), 257–266.