RAIRO-Oper. Res. 49 (2015) 123–142
DOI: 10.1051/ro/2014044

RAIRO Operations Research
www.rairo-ro.org

LOT-SIZING FOR PRODUCTION PLANNING IN A RECOVERY SYSTEM WITH RETURNS

Tarik Zouadi[1], Alice Yalaoui[2], Mohamed Reghioui[1] and Kamal Eddin El Kadiri[1]

Abstract. This paper deals with the production planning and control problem of a single product involving combined manufacturing and remanufacturing operations. We investigate here a lot-sizing problem in which the demand for items can be satisfied by both the new and the remanufactured products. We assume that produced and recovered items are of the same quality. Two types of inventories are involved in this problem. The produced items are stored in the first inventory. The returned products are collected in the second inventory and then remanufactured. The objective of this study is to propose a manufacturing/remanufacturing policy that would minimize the holding, the set up and preparation costs. The decision variables are the manufacturing and the remanufacturing rates. The paper proposes an extension of the reverse Wagner/Whitin dynamic production planning and inventory control model, a Memetic Algorithm (MA) and a Hybrid Algorithm (HA). The HA was improved with a post-optimization procedure using Path Relinking. Numerical experiments were conducted on a set of 300 instances with up to 48 periods. The results show that both methods give high-quality solutions in moderate computational time.

Keywords. Lot-sizing, remanufacturing, genetic algorithm, GRASP, path relinking.

Mathematics Subject Classification. 90B30, 90C27, 90C59.

Received July 17, 2013. Accepted May 22, 2014.

[1] National School of Applied Sciences of Tetouan Mhannech II, BP 2121 Tetouan, Morocco. tarik.zouadi@gmail.com

[2] LOSI (UMR CNRS ICD 6281), University of Technology of Troyes, 12 Rue Marie Curie CS 42060, 10004 Troyes Cedex, France.

Article published by EDP Sciences

1. Introduction

Reverse logistics is the process of planning, implementing, and controlling the efficient, cost effective flow of raw materials, in-process inventory, finished goods and related information from the point of consumption to the point of origin for the purpose of recapturing value or proper disposal (as defined by Rogers and Tibben-Lembke [29]). Reverse logistics has interested many researchers in the last decade due to legislative, environmental, and economic considerations. Indeed, many countries impose strict measures to protect the environment. This context generates favorable conditions to technological advancement in processes such as remanufacturing, recycling, refurbishing, and repair and more generally in reverse logistics and green supply chain management. Furthermore, nowadays, customers are more attracted by environment friendly products (Krikke *et al.* [17]).

Fleischmann *et al.* [9] and De Brito *et al.* [7] discussed four main steps in a reverse logistic process. The first step refers to the collect of the used products from secondary market. The second step is reserved to the inspection and the sorting of the products according to their quality characteristics. These steps are followed by a third step of re-processing or "direct recovery" of used items. The fourth and the final step close the cycle with the redistribution of recovered items to markets (customers). The recovery of products may take any of the following ways: repair, refurbishing, remanufacturing, cannibalization and recycling (see Fig. 1). Repair refers to the fact of bringing the defective products to a working status. The refurbishing restores the quality of used products and extends their service-life. Remanufacturing restores a used product to as-good-as new state. Cannibalization recovers a limited set of reusable parts from a used product, while recycling extracts materials and components from used products for reuse. This paper focuses on used products, which are either remanufactured or refurbished. Figure 2 shows the reverse logistics system considered [33].

The paper is organized as follows. In Section 2 a literature review is presented. The problem definition is detailed in Section 3. The mathematical model is given in Section 4. Then, the developed heuristics and metaheuristics are explained respectively in Sections 5, 6 and 7. Finally results are discussed in Section 8 and some conclusions and perspectives are presented in the last section.

2. Literature review

Production planning and control activities in remanufacturing differ greatly from those in traditional manufacturing according to Guide [12]. Junior *et al.* [15] review the literature on production planning and control for remanufacturing until 2008. This review determines the gaps identified by Guide [12] and proposes a classification for different problems related to production planning and control.

Lot sizing problems received recently particular attention from researchers, Brahimi *et al.* [6] present a state-of-the-art of the single item lot sizing problem, in its uncapacitated and capacitated versions. Suwondo *et al.* [32] review the

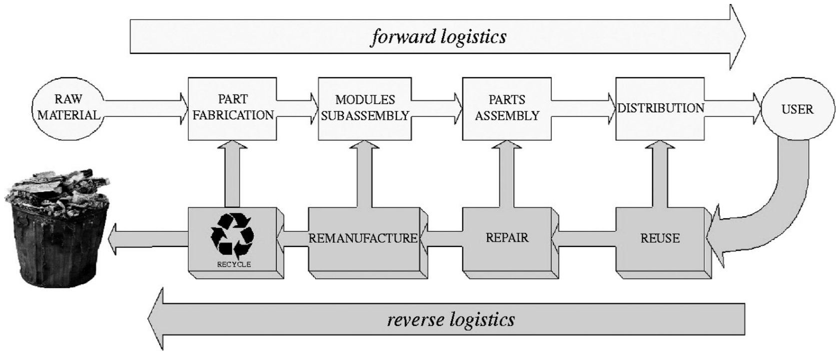

FIGURE 1. Reverse logistics strategies for end-of-life products [13].

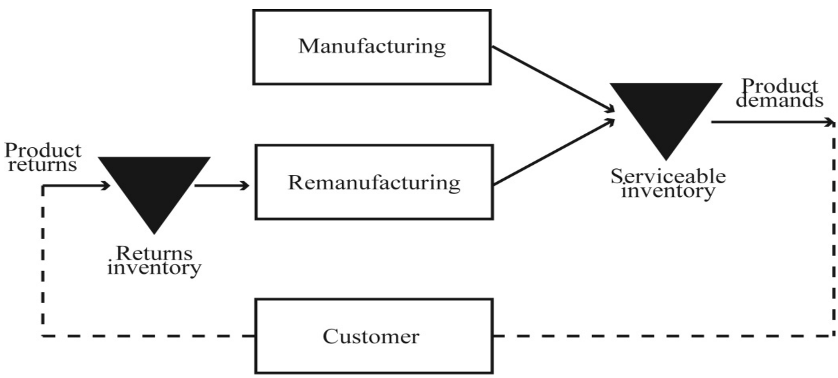

FIGURE 2. Inventory system with remanufacturing [33].

models and the proposed algorithms for the dynamic lot-sizing problems. Boissière *et al.* [5] focus on an N-stage serial production-distribution system with limited production capacity at the first stage. The objective in such system is to find inventory management policies that minimize the global logistic cost, including transportation and holding costs.

The lot sizing problem is extensively studied in the literature under several constraints. Massonnet *et al.* [22] propose a review of approximation algorithms for deterministic lot sizing problems with time-varying demand constraint. Absi *et al.* [2] studied mixed integer programming based heuristics for the lot sizing problem with multi item products. In [3], the authors propose a model with more constraints, like production time windows, early productions, backlogs and lost sales. In a recent contribution, Absi *et al.* [1] investigated also the lot sizing problem with carbon emission constraints. These environmental constraints aim at limiting the carbon emission per unit of product supplied with different modes.

Dynamic lot sizing planning for manufacturing/production orders over a finite number of future periods in which demand is dynamic and deterministic, is one of the most extensively studied problem in production and inventory control. Lu *et al.* [19] discussed the joint replenishment problem with multiple items and propose a state-of-the-art of the dynamic models.

Teunter *et al.* [33] suggest two models for production systems with returns. In their first model, they assume that on a production line, both operations manufacturing and remanufacturing, own one common setup cost (line preparation cost), while in the second model, they consider that each operation has its own setup cost. The authors propose an adaptation of the well-known heuristics Silver Meal and Part period balancing. Schulz [30] proposes a generalization of the adaptation of the Silver and Meal heuristic introduced by Teunter *et al.* [33] for the separate setup cost setting. Furthermore, a simple improvement heuristic is applied to the solution obtained to enhance the heuristic performance. In [31], Schulz *et al.* propose a flexibly structured heuristic that allows for differently sized remanufacturing batches, and shows that the proposed approach outperforms other existing ones.

The quality of returns and remanufactured products is one of the main problems in these types of systems that integrate the remanufacturing options. Many studies about the remanufacturing systems try to show the importance of the disposal option, when a rate of returns could not be remanufactured. Pineyro *et al.* [25] provide a model with a disposal option and one way substitution constraint. The model considers two types of demand, a demand for new and remanufactured products where the demand of the remanufactured products could be satisfied with the new products (substitution). Hasanov *et al.* [14] consider a model where the remanufactured products are incompatible. They are not perceived by customers to have the same quality characteristics. This leads to the conclusion that the remanufacturing products do not have the same quality, and should be oriented to the secondary market.

Furthermore, many proposed models try to involve procurement cycle, which means that instead of producing, the firm prefers to purchase the rest of the needs to meet the client demand or to combine the purchasing and production. These models involve a procurement cost. Wang *et al.* [34] study a model with an outsourcing option. The model presented shows that the client's needs are satisfied by new items manufacturing, remanufactured items, or outsourcing, which explains the introduction of the outsourcing costs (transport, handling) to the problem. Also, Konstantaras *et al.* [16] consider a remanufacturing model in which the demand is satisfied by remanufacturing and outside procurement. The model considers an inspection phase in which some recovered items do not qualify to be classified as "remanufactured" and are perceived by customers to be of secondary quality.

Many approaches were used to solve dynamic lot sizing problems, Raf and Zeger [27] review the literature on the meta-heuristics used to solve the lot sizing problem and provide a comparison between the solution approaches, when Goren *et al.* [11] present a literature review on the use of the Genetic Algorithm to solve the dynamic lot sizing problem.

Hybrid Algorithms are used by many researchers to solve lot sizing problems. Fernandes and Lourenço [8] furnish a literature review on Hybrid Algorithms combining local search heuristics with exact algorithms. Christian Almeder [4] use the hybrid optimization approach for the multi multi-level capacitated lot-sizing problems, and Flindt *et al.* [10] use a hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times.

3. Problem definition

The objective of this study is to propose a manufacturing/remanufacturing policy such that minimizes the holding, set up and preparation costs for manufactured and remanufactured products.

Lot-Sizing is one of the most important research topics in production and inventory management. Since we are dealing with reverse logistics, we focus on a lot-sizing problem in a system that supports and remanufacture returns to be as good as new products.

The system studied is a single production line on which we can accomplish both the classical production and remanufacturing, which means that two different set up costs are considered. However, when the production or the remanufacturing is started we will need to prepare the production chain, which also leads to a common cost of preparation.

This realistic configuration gives a more generic model compared to the one of Teunter *et al.* [33]. Furthermore, we consider that demands and returns rates are deterministic on a finite planning horizon and returns cannot be eliminated since they are all remanufactured. They will be of the same quality as new products. We also assume that the initial stocks are equal to zero.

The objective is to find a production planning along the finite horizon which minimizes the sum of preparation, setup cost, and holding cost, when the demand of these periods is assumed to be deterministic.

4. The mathematical model

Many researchers propose mathematical models for the lot sizing problem, Wu *et al.* [35] review mathematical models for capacitated multi-level production planning problems with linked lot sizes.

In this section, the model of Teunter *et al.* [33] proposed in 2006 is extended to the case with the preparation cost. A generic model is proposed considering three costs, one setup cost for each operation and a common preparation cost for both operations.

4.1. Notations

T:	Planning horizon.
R_t:	Number of returns received at the beginning of period t.
D:	The sum of demands for the remaining periods.
D_t:	Number of products demanded in period t.
K:	(Joint Set-up cost) Preparation Cost.
Kr:	(separate) Setup cost for remanufacturing.
Km:	(separate) Setup cost for manufacturing.
hr:	Unit holding cost for returns per period.
hs:	Unit holding cost of end-products (serviceable) per period.
Xr_t:	Number of products remanufactured in period t.
Xm_t:	Number of products manufactured in period t.
Ir_t:	Inventory level of returns at the end of period t.
Is_t:	Inventory level of serviceable at the end of period t.
δr_t:	0-1 Indicator variable for remanufacturing set-up in period t.
δm_t:	0-1 Indicator variable for manufacturing set-up in period t.
δ_t:	0-1 Indicator variable for preparation cost in period t.
M:	Large integer.

4.2. The mathematical model

Minimize:

$$FO = \sum_{(t=1)}^{T} (K\delta_t + Kr\ \delta r_t + Km\ \delta m_t + hr\ Ir_t + hs\ Is_t) \tag{4.1}$$

Subject to:
For $t = 1, 2, 3, \ldots T$.

$$Ir_t + R_t - Xr_t = Ir_{(t+1)} \tag{4.2}$$

$$Is_t + Xr_t + Xm_t - D_t = Is_{(t+1)} \tag{4.3}$$

$$Xm_t \leq M \times \delta m_t \tag{4.4}$$

$$Xr_t \leq M \times \delta r_t \tag{4.5}$$

$$\delta_t \geq \delta m_t \tag{4.6}$$

$$\delta_t \geq \delta r_t \tag{4.7}$$

$$\delta_t \leq \delta r_t + \delta m_t \tag{4.8}$$

$$\delta_t, \delta r_t, \delta m_t \in \{0, 1\} \tag{4.9}$$

$$M = D \tag{4.10}$$

$$Ir_t \geq 0,\ Is_t \geq 0. \tag{4.11}$$

The objective function (4.1) including set up, preparation and inventory costs computes the solution fitness. Constraints (4.2) and (4.3) are the inventory flow conservation equations for returned products and serviceable products, respectively. Constraints (4.4) and (4.5) ensure the cancellation of production quantities for the periods without setup cost. Constraints (4.6), (4.7) and (4.8) guarantee the adding of the preparation cost when one of the two operations manufacturing or remanufacturing occurs. This model was used to find solutions with CPLEX. The results are given in Section 6.

5. Heuristics

Two of the most known heuristics for lot sizing are the Silver Meal heuristic (SM) and the Part Period Balancing (PPB). The logic followed by both heuristics satisfies the well known zero-inventory property, which means that on each period with a manufacturing/remanufacturing decision any solution that satisfies the following property should have zero returns at the end of this period. This leads to conclude that the logic followed by these heuristics favors to remanufacture first.

The inconvenience of these heuristics is that they are shortsighted, which means that they are unmindful of future period cost consequences.

In the sequel, we propose an adaptation of both heuristics to the case of separate costs and the joint preparation cost. As mentioned before, they favor returns production first, which means that when we decide to produce on the period t, we start by remanufacturing all the returns and if the demand is greater than the quantity of returns, we complete by manufacturing new items.

5.1. Part period balancing (PPB)

If we produce in period s to meet the demand of periods $s, s+1, s+2 \dots, s+t$ the total cost incurred has the following form:

$$Cost(s, s+t) = preparation\ cost + setup\ costs + holding\ cost(s, s+t).$$

Since the goal is to achieve a compromise between the setup cost and the holding cost, it seems reasonable to choose t (for a given value of s) so that the setup cost is approximately equal to the holding cost. The total cost CT associated to the heuristic in the interval [l,k] is given by the following expression:

$$CT_{(l,k)} = K + Km + Kr + \sum\nolimits_{(t=l)}^{k} ((hs \times Is_t) + (hr \times Ir_t)). \tag{5.1}$$

5.2. Silver meal heuristic (SM)

To introduce this approach, we consider again the cost *C (s, s + t)* defined above.

Typically, when t increases, the average cost per period $C(s,\ s+t)/(s+t)$ is first decreasing (since the fixed set up costs are amortized over several periods) and then it starts to increase since the holding costs become more important.

The Silver Meal heuristic strategy is to produce in order to meet the demands of periods s to $s+t$, where $s+t$ is the last period for which the average cost per period is decreasing. When the period t is identified, the cost is calculated as it was described in the PPB heuristic.

Table 1. Example of returns rate and demand.

Period	1	2	3	4	5	6	7	8
Returns	5	5	5	5	5	5	5	5
Demand	15	15	15	15	15	15	15	15

6. Memetic Algorithm (MA)

Genetic Algorithms are population-based metaheuristics. They have been successfully applied to many optimization problems [24]. Many researches use the Genetic Algorithms to solve lot-sizing problems. However, the premature convergence is an inherent characteristic of such classical Genetic Algorithms that makes them unable of searching numerous solutions of the problem domain.

The Memetic Algorithm is an extension of the traditional Genetic Algorithm (Labadi *et al.* [18]). It uses a local search technique to reduce the likelihood of the premature convergence. Guner *et al.* [11] review the literature on the use of the Memetic Algorithm to solve the dynamic lot sizing problem.

Memetic Algorithms therefore rely on three features:

The selection allows to foster individuals who have a better fitness. For our problem, the fitness is the sum of the preparation cost, the setup cost, and the holding cost.

The crossover combines two parents to form one or two children (offspring) while trying to keep the good features of parents.

A local search allows making moves from solution to another one in the space of candidate solutions by applying local changes, until a solution deemed the best is found or a time bound is elapsed.

6.1. Solution representation

The representation of solutions (encoding) is a critical point for the efficiency of a Memetic Algorithm as it must be adapted to the problem and to the solution evaluation process.

The zero-inventory property [33] requires that in each period with a production decision, the stock of returns at the beginning of the period should be equal to zero and the returned products should be all used.

Teunter *et al.* [33] show that any optimal solution in the case of joint setup cost should satisfy this property, but in the case of separate set up costs, they demonstrate that the property is no longer verified which prevents the use of a binary coding. For this reason, we proposed a presentation of the solution as a set of quantities. For this reason, the chromosome is presented as a list of quantities which need to be produced in each period for both manufacturing and remanufacturing.

In the example of Table 1, each period has a demand of fifteen products and five products are returned. Based on this data, Figure 3 shows the coding of a feasible solution.

FIGURE 3. Presentation of a solution.

In the first period, we manufactured 30 items to meet the client's needs in the first and in the second period. The need of the third period is satisfied by remanufacturing 15 items. The demands of the period 4, 5 and 6 are satisfied by manufacturing 45 items and 30 items are remanufactured to meet the demands of the last periods 7 and 8.

6.2. Generating initial population

The random generation of solutions can lead to infeasible solutions, or solutions that do not meet customer needs or which lead to stock outs. To avoid generating such solutions, we have developed a population generation procedure.

Algorithm 6.1 Population generating procedure

for each period:
    if $I_t^m + I_t^r \leq D_t$ **then**
    $x_t^r \leftarrow Generator(0, I_t^r)$ **and** $x_t^m \leftarrow Generator(D_t - x_t^r, D)$
    Update the parameters:
        $D \leftarrow D - x_t^r - x_t^m$
        $I_{(t+1)}^m \leftarrow I_t^m + x_t^m + x_t^r - D_t$
        $I_{(t+1)}^r \leftarrow I_t^r + R_{(t+1)} - x_t^r$
    endif
    if $I_t^m + I_t^r \geq D_t$ **then**
        if $I_t^m \geq D_t$ **then**
        $x_t^m \leftarrow 0$, $x_t^r \leftarrow 0$ **and** $D \leftarrow D - D_t$
        endif
        if $I_t^m < D_t$ **then**
        $x_t^r \leftarrow MAX(Generator(0, I_t^r), D_t - I_t^m)$ *and* $x_t^m \leftarrow 0$
        Update the parameters:
            $D \leftarrow D - x_t^r - x_t^m$
            $I_{(t+1)}^m \leftarrow I_t^m + x_t^m + x_t^r - D_t$
            $I_{(t+1)}^r \leftarrow I_t^r + R_{(t+1)} - x_t^r$
        endif
    endif
endfor

We consider a function called generator (a, b) that returns a random value between a and b. where a and b are two numerical values.

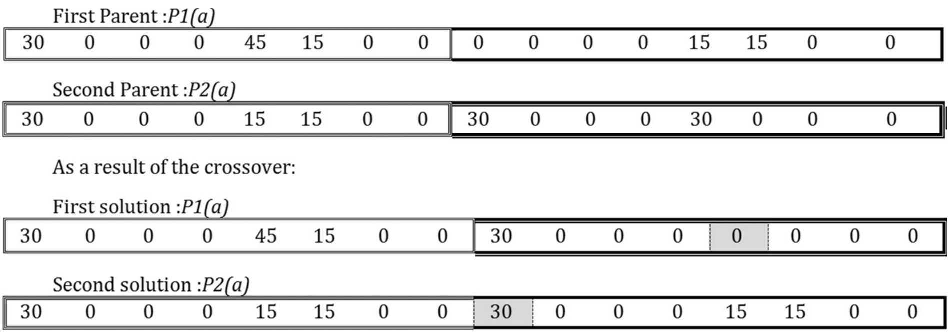

FIGURE 4. Crossover example.

This procedure detailed in Algorithm 6.1 consists of launching a random production quantity when we have a lack of products to meet the client demand. For each period the algorithm determine the stock quantity of produced and returned products, when the stock of new products and returns is less than the demand of the current period ($Is_t + Ir_t \leq D_t$), we launch a random quantity of returns to be remanufactured ($Xr_t \leftarrow \mathit{Generator}(0, Ir_t)$) and we complete the demand with a random quantity of new products ($Xm_t \leftarrow \mathit{Generator}(D_t - Xr_t, D)$). If the returns quantity is enough to satisfy the clients' needs ($Is_t + Ir_t \geq D_t$ & $Is_t < D_t$), then we only launch a random quantity of returns to be remanufactured ($Xr_t \leftarrow \mathit{MAX(Generator}(0, Ir_t), D_t - Is_t)$ *and* $Xm_t \leftarrow 0$). But if the quantity of the new products is greater than the current demand and sufficient to meet the clients' needs ($Is_t + Ir_t \geq D_t$ & $Is_t \geq D_t$), we do not launch production ($Xm_t \leftarrow 0$, $Xr_t \leftarrow 0$).At each iteration the algorithm update respectively the demand rate, stock quantity of the produced items and the quantity of returns ($D \leftarrow D - Xr_t - Xm_t$, $Is_{(t+1)} \leftarrow Is_t + Xm_t + Xr_t - D_t$ & $Ir_{(t+1)} \leftarrow Ir_t + R_{(t+1)} - Xr_t$).

6.3. CROSSOVER

The crossover operator used in this Memetic Algorithm is based on one cutting point (LOX). LOX principle is to copy the beginning of the first parent P1 until a position P1(a) then the parent P2 is swept completely from left to right and the solution is completed with periods not already present in the offspring. Since the first half of the chromosome is feasible, we have only to coordinate data of the first half with the second. Coordination between the first and the second half of the offspring consists of adding in the beginning of the second half a quantity of products when the sum of the quantities is unable to meet the client needs and to subtract a quantity from the end when there is a product excess in the second half of the offspring.

The second offspring solution is generated in the same way while considering parent P2 before parent P1, as illustrated by the example described at Figure 4.

Initial solution:

15	0	15	0	60	0	0	0	0	0	15	15	0	0	0	0

Move 1 applied to period 6.

15	0	15	0	60	0	0	0	0	0	30	0	0	0	0	0

Move 2 applied to period 6.

15	0	15	0	60	0	0	0	0	0	0	30	0	0	0	0

FIGURE 5. Local search example.

In this example, in order to make the offspring solutions feasible, the production rate of some periods was changed. For the first offspring, on the second half of the chromosome, there is 60 items that should be produced, while the need is just of 30 items, to meet the customer's demand. For this reason, we subtract the surplus from the end (period 7). For the second offspring, the produced quantity is not enough to satisfy the needs, so the remaining quantity is produced in the first period of the second half of the offspring (period 5).

6.4. Local search

The Memetic Algorithm uses two neighborhood structures (local searches). They are based on a first improvement strategy and use respectively the two following moves illustrated at Figure 5:

Move1: it consists of moving manufactured quantities to remanufacturing over the same period. Then we test whether the solution is feasible and if it is improved. The purpose is to avoid the set up cost and use returns.

Move2: it is the contrary of Move 1. The quantity to remanufacture is moved to manufacturing on the same period, and then we test if the solution is feasible and improved. The objective is to avoid setup and holding costs.

Algorithm 6.2 Local search

$S \leftarrow$ Initial Solution
$S^* \leftarrow S$
$S1 \leftarrow$ Local_search_1(S)
if (Cost(S1)<Cost(S*))
  $S^* \leftarrow S1$
$S2 \leftarrow$ Local_search_2(S*)
if (Cost(S2)<Cost(S*))
  $S^* \leftarrow S2$

Algorithm 6.2 summarizes the structure of this improvement strategy. The objective of this implementation is to change the neighborhood during the search. The first local search (Local_search_1()) is called and when it converges to local optima, the best solution found (S1) is used as a starting solution for the second local search (Local_search_2()). The improvement procedure stops when the second local search converges to local optima. S is the selected solution, S* the starting solution, S1 is the best solution found by the first local search.

S2 is the best solution found by the second local search, Local_search_1() is the first local search using move1, Local_search_2() is the first local search using move2.

In the local search procedure, the chromosome is scanned from left to right to test for each period if the move gives improvement or not. When an improvement is performed, the local search restarts from the first period.

7. Hybrid Algorithm

Hybrid Algorithms are used by many researchers to solve Lot-Sizing problems. Puchinger *et al.* [26] review the literature on the algorithms combining metaheuristics and exact algorithms in combinatorial optimization.

In this section, we propose a resolution approach based on a Hybrid Algorithm inspired by the Greedy Randomized Adaptive Search Procedure (GRASP).

The GRASP [28] is a multi-start or iterative process, in which an iteration consists of two phases: a construction phase, in which a feasible solution is produced with a randomized heuristic, and a local search phase, in which a local optimum in the neighborhood of the constructed solution is sought. The best overall solution is kept as the result.

The proposed implementation consists on a Hybrid Algorithm with one restart. A randomized version of SM heuristic is used in the initial run. PPB heuristic was randomized and used in the restart. Each solution resulting from a randomized heuristic is used to set the values of decision variables δr_t, δm_t, and δ_t in the model. These variables indicate if manufacturing or remanufacturing is launched. Then, CPLEX is called to seek the best quantities to produce. This second step replaces the local search phase of the GRASP in the implemented hybrid method.

7.1. Greedy randomized heuristics

As introduced in Section 4 , the SM heuristic logic is to produce in a period s to cover the client's needs until the period $s{+}t$ where the period $s{+}t$ is the last period for which the average cost per period is decreasing. The randomization consists of choosing a random period in the neighborhood of the period $s{+}t$ determined by the SM heurisitic. The choice of production type to launch on this period is also randomized. For each period with a production, a real number r is generated randomly between 0 and 1 and the type of production to launch is chosen as follows:

- If $r < 0.33$: manufacturing;
- If $0.33 < r < 0.66$: remanufacturing;
- If $0.66 < r$: manufacturing and remanufacturing.

The same strategy is followed for the Part Period Balancing heuristic (PPB). A random period in the nieghberhood of the period $s + t$ is selected.

7.2. Exact method to determine the produced quantities

At each iteration of the Hybrid Algorithm, the randomized heuristic proposes the set of periods on which the manufacturing and/or the remanufacturing could be launched. The production propositions generated by the randomized heuristic as a binary coding are integrated in the mathematical model as decision variables which is then solved by CPLEX.

The solution returned by CPLEX defines the optimal quantities according to the production periods proposed by the randomized heuristics.

7.3. Path Relinking

Path Relinking is a local search procedure proposed as an intensification and diversification approach [21]. Nascimento *et al.* [23] propose an implementation of the GRASP heuristic with path-relinking for the multi-plant capacitated lot sizing problem.

This procedure is based on generating new solutions by exploring new trajectories which link high quality solutions. This is done by starting from an initial solution and by generating a trajectory in the neighborhood which guide the solutions search. Marti *et al.* [20] used a Path Relinking combines with a GRASP as an intensification procedure. The relinking in this context consists in finding a path between a solution found with the GRASP and a chosen elite solution. Therefore, the relinking concept has a different interpretation within the GRASP since the solutions found in different GRASP iterations are not linked by a sequence of moves. Numerous examples of GRASP with Path Relinking were presented by Marti *et al.* [21].

The Path Relinking procedure was combined with the Hybrid Algorithm. For the lot-sizing problem the Path Relinking works on the binary encoding of the solutions which shows when the production can be launched.

Each move of the Path Relinking procedure consists of removing or adding a production decision in a special periode. A maximum of two moves is needed at each period to change the production decision for manufacturing or remanufacturing. In the worst case, the number of moves is equal to $(2 \times T)$ when the two solutions are completely different.

8. Numerical experiments

We summarize in this section the numerical experiments performed with the developed algorithms. The obtained results by the Memetic Algorithm (MA), the Hybrid Algorithm (HA), and the Hybrid Algorithm combined with Path Relinking (HAPR) were compared with those of the Silver Meal heuristic and the ones

Table 2. Parameters of the memetic Algorithm.

β :	the number of crossing;
μ :	number of individuals in the initial population;
α :	the probability of crossover;
ξ :	the maximum number of non-improving iterations;
σ :	local search order.

obtained by CPLEX. Then, we prove the performance of the MA, HA, and HAPR by testing on the Teunter *et al.* [33] instances and comparing with their results.

Many tests have been done to tune the parameters required by the methods. The best configuration of each algorithm was found by varying the following parameters:

- number of iterations;
- population size;
- the maximum number of non-improving iterations;
- local search type (first improvement/best improvement);
- number of restarts.

The Memetic Algorithm employs a set of parameters that requires fine tuning. In Table 2, these parameters are listed and explained. Based on a large number of runs, the following set of parameters was finally selected $(\beta; \mu; \alpha; \xi; \sigma) = (2000; 40;$ 0,7; 50; 1-2).

For the HA, the number of iterations of the first phase is the only parameter that requires tuning. After a large number of runs, the number of iterations that gives the best tradeoff between computational time and solution quality is 150 for each restart (which gives a total of 300 iterations). The decision related to the choice of the next period on which the production will be launched is based on a random picking of one period in an interval defined around the period found following the SM strategy. The size of this interval is dynamic and chosen according to the planification horizon length. The Silver Meal and part period balancing solutions were injected in the population of the Memetic Algorithm to add some chromosomes of good quality to the initial population. The obtained results of all the algorithms are compared following two different instances classifications:

– Grouped according to the number of periods of the planning horizon.

– Grouped according to the type of demand.

The instances are derived from the article of Teunter *et al.* [33]. Five different types of demand and return patterns are considered: stationary, linearly increasing, linearly decreasing, seasonal with peak in the middel, seasonal with valley in the middel. Tests are performed on 300 instances grouped in 12 sets, each set containing 25 instances.

TABLE 3. The Memetic Algorithm results.

Period	CPLEX		SM	MA		
	AS	AT (s)	AS	AS	GMC	AT (s)
4	3732.8	0.21	3810	3740	0.22%	0.71
8	7189.4	0.96	8040	7231	0.52%	1.17
12	10 463.5	5	10 970	10 723	2.38%	1.62
16	13 823.2	26	14 669	14 376	3.84%	1.73
20	17 385	43	19 036	18 254	4.74%	1.75
24	21 715.3	181	24 205	22 654	3.96%	1.91
28	26 557.6	412	30 467	27 930	4.59%	2.11
32	32 034	623	37 065	33 664	6.57%	2.17
36	–	–	44 376	40 278	–	2.34
40	–	–	52 909	47 681	–	2.93
44	–	–	62 311	56 603	–	3.07
48	–	–	72 929	66 682	–	3.61
Average	–	–	31 732.2	29 151.3	3.35%	2.09

8.1. Grouped according to the number of periods of the planning horizon

The instances have a planification horizon up to 48 periods. The obtained results are given in Tables 3–5. The notations used are listed below:

- AS: average solution;
- AT (s): average time;
- GMC: gap between MA and CPLEX;
- GHAC: gap between Hybrid Algorithm and CPLEX;
- GHAPRC: gap between Hybrid Algorithm with PR and CPLEX;
- GSC: gap between Silver Meal heuristic and CPLEX.

For all the instances, the proposed approches are compared with the optimal value returned by CPLEX, but when CPLEX can't prove optimality, we compare with the lower bound.

For each Table (3, 4, and 5) the first column shows the number of periods in each set. Columns 2, 3 and 4 present respectively the average solution found by CPLEX, its average computational time and the average solution cost of SM for the instances of each set. The next three columns give respectively the average solution cost obtained by the method for each set, the percentage gap between the results of the method and CPLEX and the average computational time.

As far as the results obtained by CPLEX are concerned, the optimality is no more achieved when the number of periods exceeds 32.

Compared to CPLEX solutions, when the optimum is achieved, the HA (2.89%) gives near optimal solutions while the MA (3.35%) is less efficient. On the other hand, the computational time of the HA is greater than the MA.

TABLE 4. The Hybrid Algorithm results.

Period	CPLEX		SM	HA		
	AS	AT (s)	AS	AS	GHAC	AT (s)
4	37 32.8	0.21	3810	3732.8	0%	14
8	7189.4	0.96	8040	7189.4	0%	14
12	10 463.5	5	10 970	10 648	1.73%	16
16	13 823.2	26	14 669	14 348	3.78%	17
20	17 385	43	19 036	18 142	4.16%	17
24	21 715.3	181	24 205	22 577	3.98%	18
28	26 557.6	412	30 467	27 614	3.83%	18
32	32 034	623	37 065	33 165	5.66%	19
36	–	–	44 376	39 776	–	20
40	–	–	52 909	46 986	–	20
44	–	–	62 311	55 049	–	23
48	–	–	72 929	63 851	–	24
Average	–	–	31 732.2	28 589.7	2.89%	18.33

TABLE 5. The Hybrid Algorithm with Path Relinking results.

Period	CPLEX		SM	HAPR		
	AS	AT (s)	AS	AS	GHAPRC	AT (s)
4	3732.8	0.21	3810	3732.8	0%	24
8	7189.4	0.96	8040	7189.4	0%	29
12	10 463.5	5	10 970	10 550	0.83%	35
16	13 823.2	26	14 669	14 178	2.57%	39
20	17 385	43	19 036	17 856	2.71%	51
24	21 715.3	181	24 205	22 367	2.54%	63
28	26 557.6	412	30 467	27 125	2.14%	74
32	32 034	623	37 065	32 965	2.78%	92
36	–	–	44 376	39 486	–	94
40	–	–	52 909	43 970	–	120
44	–	–	62 311	53 005	–	133
48	–	–	72 929	61 001	–	144
Average	–	–	31 732.2	27 764.93	1.70%	74.833

For the HA with Path Relinking, Table 5 shows that this later gives better results than the simple HA mainly thanks to the intensification by the Path Relinking. However the computational time increases significantly.

The quality of solutions depends on the length of the planning horizon. For small horizons (4 to 12), both methods give near optimal solutions, while for long horizons the solution is pretty far from being close to optimal.

Regarding the computational time, CPLEX is rather efficient to prove optimality for small size instances, but on larger ones the computational time becomes more important and exceeds half an hour on many instances without finding an optimal solution.

TABLE 6. Average Gap between (MA), the HA and HA with Path Relinking and CPLEX.

Type of demand	MA	HA	HAPR
	GMAC	GHAC	GHAPRC
Stationnaire	1.89%	2.21%	1.21%
Positive Trend	2.65%	3.09%	1.89%
Negative Trend	4.21%	2.84%	1.54%
Seasonal (Peak in middle)	4.14%	2.58%	1.92%
Seasonal (Valley in middle)	3.72%	3.23%	1.93%

8.2. Grouped according to the type of demand

In this part, the results are grouped according to the type of demand: stationary, positive trend, negative trend, seasonal (peak in middle and valley in middle). Table 6 shows the average gap between the MA, the Hybrid Algorithm, the Hybrid Algorithm with Path Relinking , and the optimal solutions (when they are reached) for each set.

This analysis shows that the MA performs better when the demand is constant, seasonal peak in middle and seasonal valley in middle, however when the demand is positive or negative trend, the MA is less efficient.

For the HA and the HAPR, the variation is less important, but we can notice a little improvement on the stationary demand.

8.3. Numerical experiment on Teunter models

In Tables 7 and 8, the first column shows the number of periods in each set. Column 2 presents the percentage gap between the results of CPLEX (mathematical model) and columns 3 and 4 give respectively the percentage gap between the results of the MA and CPLEX, and the average computational time. The last four columns present the same information for the Hybrid Algorithm and Hybrid Algorithm with Path Relinking.

Teunter *et al.* [33] show in their contribution that the average gap found by the SM heuristic for the case with a commun set up is 3% and 8.4% for the second case with separate set up costs.

Table 7 shows the obtained results on Teunter *et al.* [33] instances. For the first model with a commun setup cost the gap of the MA is 0.83%, 0.67% for the HA, and 0.38% for the HAPR.

For the model with separate setup costs, Table 8 shows the results found with the proposed methods, the average gap in this case is 3.06% for the MA, 2.44% for the HA, and 1.54% for the HAPR.

The results show clearly that the proposed algorithm instances gives better solutions in moderate computational time in comparison with Tuenter *et al.* [33].

TABLE 7. Results for instances with common set up cost.

Period	SM	MA		HA		HAPR	
	GSC	GMAC	AT (s)	GHAC	AT (s)	GHAPRC	AT (s)
4	1.20%	0%	0.24	0%	0.19	0%	1
8	4.72%	0%	0.92	0%	0.81	0%	2.4
12	3.82%	0%	1.32	0%	1.4	0%	3.4
16	2.76%	0%	2	0%	2.2	0%	5
20	4%	0.06%	3	1.30%	2.9	0.29%	7
24	2.43%	0.48%	5	0.93%	4	0.62%	8
28	2.71%	0.85%	9	1.09%	6	0.68%	9
32	2.90%	1.76%	14	1.08%	9	0.83%	15
36	2.23%	1.59%	16	0.89%	13	0.61%	21
40	2.15%	1.41%	19	0.89%	15	0.33%	25
44	2.34%	1.97%	22	0.87%	17	0.45%	29
48	2.32%	1.78%	24	0.98%	21	0.73%	31
Average	2.80%	0.83%	9.7	0.67%	7.7	0.38%	13.1

TABLE 8. Results for instances model with separate set up cost.

Period	SM	MA		HA		HAPR (PR)	
	GSC	GMAC	AT (s)	GHAC	AT (s)	GHAPRC	AT (s)
4	9.31%	0%	1	0%	1	0%	1
8	7.90%	3.11%	2	0%	3	0%	3
12	11.38%	3.64%	4	1.34%	9	0%	11
16	12.41%	3.86%	10	2.48%	10	3.21%	15
20	8.08%	4,48%	11	2.79%	16	2.68%	23
24	9.29%	4.77%	13	2.94%	21	2.10%	41
28	10.31%	3.44%	17	3.24%	29	2.66%	56
32	8.19%	3.45%	19	3.26%	37	1.90%	68
36	9.26%	3.42%	24	3.46%	46	1.94%	79
40	12.04%	2.06%	31	3.44%	57	1.37%	95
44	11.26%	2.20%	39	3.36%	63	1.52%	121
48	9.11%	1.80%	58	2.93%	87	1.12%	144
Average	9.93%	3.03%	19	2.44%	32.58	1.54%	54.75

9. CONCLUSION AND PERSPECTIVES

In this paper, a new formulation for a multi-stage lot-sizing problem for flow lines with returns is presented. The advantage of this new formulation is that it considers both setup and preparation costs. Two methods were developed to solve the problem, a memetic algorithm and a Hybrid Algorithm (HA). This later uses an exact resolution based on the proposed mathematical model to enhance solutions obtained by randomized heuristics. A second implementation of the Hybrid Algorithm uses a post-optimization based on a Path Relinking procedure. Numerical results have shown that both solution approaches generate high-quality solutions, especially the HAPR.

The proposed approches were adapted to the cases proposed by Tuenter *et al.* [33], and the results show clearly that both methods perform perfectly and give better solutions in a moderate computational time.

This work can provide a basis for initiating several researches by considering constraints such as the remanufactured products are not of the same quality as the new ones or a variable capacity in each period.

References

[1] N. Absi and S. Kedad-Sidhoum, MIP-based heuristics for multi-item capacitated lot-sizing problem with setup times and shortage costs. *RAIRO–Oper. Res.* **41** (2007) 171–192.

[2] N. Absi, S. Dauzère-Pérès, S. Kedad-Sidhoum, B. Penz and C. Rapine, Lot sizing with carbon emission constraints. *Eur. J. Oper. Res.* **227** (2013) 5561.

[3] N. Absi, S. Kedad-Sidhoum and S. Dauzère-Pérès, Uncapacitated lot-sizing problem with production time windows, early productions, backlogs and lost sales, *Int. J. Prod. Res.* Special Issue: Lot sizing and scheduling: new models and solution approaches to address industrial extensions **49** (2011) 2551–2566.

[4] C. Almeder, A hybrid optimization approach for multi-level capacitated lot-sizing problems. *Eur. J. Oper. Res.* **200** (2010) 599–606.

[5] J. Boissière, Y. Frein and C. Rapine, Lot-sizing in a serial distribution system with capacitated in-system production flow. *Int. J. Prod. Econ.* **112** (2008) 483–494.

[6] N. Brahimi, S. Dauzere-Peres, N.M. Najid and A. Nordli, Single item lot sizing problems. *Eur. J. Oper. Res.* **168** (2006) 1–16.

[7] M.P. de Brito, R. Dekker and S.D.P Flapper, *Reverse logistics: A review of case studies. Lecture Notes in Economics and Mathematical Systems* **544** (2005) 243–281.

[8] S. Fernandes and H.R. Lourenço, Hybrids combining local search heuristics with exact algorithms, in *Proceeding of the V Congreso Español sobre Metaheurísticas. Algoritmos Evolutivos y Bioinspirados, MAEB'2007*, February 14-16 (2007) 269–274.

[9] M. Fleischmann, B. Gräve and J.A.E.E. van Nunen, Integrating Closed-Loop Supply Chains and Spare-Parts Management at IBM. *Interfaces* **33** (2003) 44–56.

[10] L. Flindt Muller, S. Spoorendonk and D. Pisinger, A hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times. *Eur. J. Oper Res.* **218** (2012) 614–623.

[11] G.H. Goren and S. Tunali, R. Jans. A review of applications of genetic algorithms in lot sizing. *J. Intelligent Manufacturing* 21 (2010) 575–590.

[12] V.D.R. Jr. Guide, Production planning and control for remanufacturing: Industry practice and research needs. *J. Oper. Manag.* **18** (2000) 467–483.

[13] J. Hanafi, S. Kara and H. Kaebernick, Reverse logistics strategies for end-of-life products. *Int. J. Logistics Manage.* **19** (2008) 367–388.

[14] P. Hasanov, M.y. Jaber and S. Zolfaghari, production, remanufacturing and waste disposal models for the cases of pure and partial backordering. *Appl. Math. Model.* **36** (2012) 5249–5261.

[15] M.L. Junior and M.G. Filho, Production planning and control for remanufacturing: literature review and analysis. Production planning & control. *Manage. Oper.* **23** (2012) 419–435.

[16] I. Konstantaras, K. Skouri and M.Y. Jaber, Lot sizing for a recoverable product with inspection and sorting. *Comput. Indust. Eng.* **58** (2010) 452–462.

[17] H.R. Krikke, H.M. le Blanc and S. van de Velde, The impact of product life cycle management on circular supply chains - and reverse. *Icfaian J. Manage. Res.* **3** (2004) 7–31.

[18] N. Labadi, C. Prins and M. Reghioui, A memetic algorithm for the vehicle routing problem with time windows. *RAIRO–Oper. Res.* **42** (2008) 415–431.

[19] L. Lu and X. Qi, Dynamic lot sizing for multiple products with a new joint replenishment model. *Eur. J. Oper. Res.* **212** (2011) 74–80.

[20] L. Marti and R. Marti, GRASP and Path Relinking for 2-Layer Straight Line Crossing Minimization. *INFORMS J. Comput.* **11** (1999) 44–52.

[21] R. Martí, M.G.C. Resende and C.C. Ribeiro, Special issue of Computers and Operations Research: GRASP with Path Relinking, Developments and applications. *Comput. Oper. Res.* **40** (2013) 3080.

[22] G. Massonnet, J.-P. Gayon, C. Rapine, Approximation algorithms for deterministic continuous-review inventory lot-sizing problems with time-varying demand. *Eur. J. Oper. Res.* 234 (2014) 641–649.

[23] M.C.V. Nascimento, M.G.C. Resende and F.M.B. Toledo, GRASP heuristic with path-relinking for the multi-plant capacitated lot sizing problem. *Eur. J. Oper. Res.* **200** (2010) 747–754.

[24] F. Neri and C. Cotta, Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation* **2** (2012) 114.

[25] P. Pineyro and O. Viera, The economic lot-sizing problem with remanufacturing and one-way substitution. *Int. J. Prod. Econ.* 124 (2012) 482–488.

[26] J. Puchinger and G.R. Raidl, Combining metaheuristics and exact algorithms in combinatorial optimization: survey and classification, in *Proceedings of the first international work-conference on the interplay between Natural and artificial computation*, edited by J. Mira and J.R. Alvarez, vol. 3562 (2005) 41–53.

[27] J. Raf and D. Zeger, Meta-Heuristics for Dynamic Lot Sizing: A Review and Comparison of Solution Approaches (24 (2004) 6,). ERIM Report Series Reference No. ERS-2004-042-LIS. *J. Oper. Res.* **234** (2014) 641–649.

[28] M.G.C. Resende and C.C. Ribeiro, A GRASP with path-relinking for private virtual circuit routing. *Networks* **41** (2003) 104–114.

[29] D.S. Rogers and R.S. Tibben-Lembke, Going backwards: reverse logistics trends and practices. University of Nevada. Reno Center for Logistics Management. *Reverse Logistics Executive Council* (1998) 101–115.

[30] T. Schulz, A new Silver-Meal based heuristic for the single-item dynamic lot sizing problem with returns and remanufacturing. *Int. J. Prod. Res.* **49** (2011) 2519–2533.

[31] T. Schulz and G. Voigt, A flexibly structured lot sizing heuristic for a static remanufacturing system. *Omega* **44** (2014) 21–31.

[32] E. Suwondo and H. Yuliando, Dynamic lot-sizing problems: A Review on Model and Efficient Algorithm. *Agroindust. J.* **1** (2012) 36–49.

[33] R.H. Teunter, Z.P. Bayindir and W. van den Heuvel, Dynamic lot sizing with product returns and remanufacturing. *Int. J. Prod. Res.* **44** (2006) 4377–4400.

[34] N. Wang, Z. He, J. Sun, H. Xie and W. Shi, A single-item incapacitated lot-sizing problem with remanufacturing and outsourcing. *Procedia Engineer.* (2011) 5170–5178.

[35] T. Wu and L. Shi, Mathematical models for capacitated multi-level production planning problems with linked lot sizes. *Int. J. Prod. Res.* **49** (2011) 6227–6247.