Computers & Operations Research 43 (2014) 225–236

Contents lists available at ScienceDirect

Computers & Operations Research

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

A heuristic solution procedure for the dynamic lot sizing problem with remanufacturing and product recovery

M. Fazle Baki [a,*], Ben A. Chaouch [a], Walid Abdul-Kader [b]

[a] Odette School of Business, University of Windsor, 401 Sunset Avenue, Windsor, Ontario, Canada N9B 3P4
[b] Industrial and Manufacturing Systems Engineering, University of Windsor, 401 Sunset Avenue, Windsor, Ontario, Canada N9B 3P4

ARTICLE INFO

Available online 16 October 2013

Keywords:
Lot-sizing
Product returns
Remanufacturing
Heuristic
Dynamic programming

ABSTRACT

Here we discuss the lot sizing problem of product returns and remanufacturing. Let us consider a forecast of demands and product returns over a finite planning horizon — the problem is to determine an optimal production plan. This consists of either manufacturing new products or remanufacturing returned units, and in this way meets both *demands* at minimum costs. The costs of course are the fixed set-up expenses associated with manufacturing and/or remanufacturing lots and also the inventory holding costs of stocks kept on hand.

In addition to showing that a general instance of this problem is NP-Hard, we develop an alternative mixed-integer model formulation for this problem and contrast it to the formulation commonly used in the literature. We show that when integrality constraints are relaxed, our formulation obtains better bounds. Our formulation incorporates the fact that every optimal solution can be decomposed into a series of well-structured blocks with distinct patterns in the way in which set-ups for manufacturing and remanufacturing occur. We then construct a dynamic programming based heuristic that exploits the block structure of the optimal solution. We also propose some improvement schemes as well. Finally, our numerical testing shows that the heuristic performs very well as intended.

1. Introduction

Today, the terms "reuse" and "remanufacture" are no longer considered alien terms by many manufacturers and retailers alike [12]. In fact, a growing number of companies are beginning to see the business value of having a green reputation, actively demonstrating to their customers that they care about the environment [26]. For instance, "Cradle-to-cradle" manufacturing has been embraced as the new style of manufacturing that provides recycling and remanufacturing of all components. What this means is, from the first use of one product to the reuse of parts in other products [12–15].

The literature on remanufacturing and product recovery has rapidly grown in the past 15 years and now encompasses a large number of contributions [12,14,15,26]. This healthy growth in research contributions has been spurred by two major trends. First, recent environmental laws and take-back regulations have forced both manufacturers and retailers to become more environmentally conscious. Second, the growing concern of the general public in the rapid deterioration of the environment (dwindling natural resources, escalating pollution levels and so on) has also forced companies to improve their corporate image by managing their businesses for the good of the environment [26].

As a result, companies are now seeking better ways to manage and optimize their reverse logistics systems. This trend has led to the development of a number of quantitative models dealing with various aspects of these systems from designing an efficient reverse logistics network [9] to better managing stocks of returned products [13] and lastly in choosing optimal lot sizes in production planning and control for remanufacturing [28, 29]. Useful survey papers that discuss both the strategic and tactical issues of managing product returns for remanufacturing can be found in Fleischmann et al. [8], Guide et al. [12], Guide and Van Wassenhove [14, 15], Blackburn et al. [5] and Srivastava [26].

In this paper, we focus on one important tactical aspect of manufacturing for reuse. Namely, we consider production-scheduling applications to an environment with fluctuating demand requirements for an item that can be met either by manufacturing new items or by remanufacturing returned products. Returns are often referred to as cores or virgin items in the literature. The number of core units available for remanufacture also varies with time. A recently acquired core can be remanufactured immediately into a like-new item or carried in inventory for future reuse. A production schedule will specify how many new units of product to make and how many cores to remanufacture during each period over the planning horizon. After all, there are costs of holding the resultant

* Corresponding author. Tel.: +1 519 253 3000; fax: +1 519 973 7073.
E-mail addresses: fbaki@uwindsor.ca (M. Fazle Baki), chaouch@uwindsor.ca (B.A. Chaouch), kader@uwindsor.ca (W. Abdul-Kader).

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.10.001

inventories of cores as well as the finished items. Not to mention, there are also fixed set-up costs associated with batch production. The objective is to find a schedule that minimizes the total set-up and holding costs while satisfying all demand requirements in a timely way.

Having thus stated our purpose, we have organized the paper as follows: in Section 2, we discuss a commonly used mixed-integer linear programming (MILP) formulation (since a number of authors alluded to the fact that the problem is NP-hard, we give proof to show that a general instance of this problem is indeed NP-hard); in Section 3, we review the literature related to the MILP model; in Section 4, we develop an alternative MILP formulation of this problem and discuss its advantages over the original formulation; in Section 5, we propose a computationally efficient heuristic method that can be used to solve the problem; in Section 6, we provide ample numerical testing to illustrate the performance of the alternative model and heuristic; and to end off, in Section 7, we conclude with some remarks.

2. Model statement

As stated above, a number of operations researchers have examined the impact of remanufacturing and product recovery on the lot-sizing problem with deterministic time-varying demands (see [21,22,11,4,32,29,23,1617]). To better relate our contribution to this literature, we begin by stating a typical mixed-integer model formulation of the problem. Table 1 summarizes the notation that will be used throughout the paper.

Let us consider a forecast of demands D_i and returns R_i over the planning horizon N. The basic problem is to choose Q_i^S and Q_i^R, namely, the quantities to be manufactured and remanufactured in period i, so as to satisfy all demands at minimal costs. Demand is met either from newly manufactured products or from the remanufacturing of some returns or both. The total costs include the fixed set-up costs of manufacturing and remanufacturing and the inventory holding costs for serviceables and returns. Without loss of generality, let the initial inventories $I_0^S=0$ and $I_0^R=0$ and define

$$y_i^S=\begin{cases}1 & \text{if new products are manufactured in period } i\\ 0 & \text{Otherwise}\end{cases}$$

$$y_i^R=\begin{cases}1 & \text{if returns are remanufactured in period } i\\ 0 & \text{Otherwise}\end{cases}$$

The complete model (P) is then

$$\text{Min} \sum_{i=1}^{N}\{h^S I_i^S+h^R I_i^R+K^S y_i^S+K^R y_i^R\} \tag{1}$$

s.t.

$$I_i^S=I_{i-1}^S+Q_i^S+Q_i^R-D_i \quad \forall i=1,2,\ldots,N \tag{2}$$

$$I_i^R=I_{i-1}^R+R_i-Q_i^R \quad \forall\ i=1,2,\ldots,N \tag{3}$$

$$Q_i^S\le\left(\sum_{j=i}^{N}D_j\right)y_i^S \quad \forall\ i=1,2,\ldots,N \tag{4}$$

$$Q_i^R\le\left(\sum_{j=i}^{N}D_j\right)y_i^R \quad \forall\ i=1,2,\ldots,N \tag{5}$$

$$y_i^S,y_i^R\in\{0,1\},Q_i^S,Q_i^R,I_i^S,I_i^R\ge 0 \quad \forall\ i=1,2,\ldots,N$$

Constraints (2) and (3) in (P) are inventory balance equations for serviceables and returns, respectively. Constraints (4) and (5) in (P) insure that if a set-up is not performed in a period, then the quantity made in that period is zero, but if a set-up is undertaken in period i, the bound $\sum_{j=i}^{N}D_j$ on the quantity produced is appropriately chosen so that the two constraints are redundant. The objective function (1) is to minimize the total set-up and carrying costs over the planning horizon N.

The formulation shown in (P) presumes the following sequence of events. In each period we first observe returns and then we decide how much to manufacture or remanufacture. Then the demand is observed and satisfied and holding costs are assessed based on the remaining stocks of serviceables and returns at the end of the period.

Table 1
Notation.

General	
N	Planning horizon
i	Index for periods in the planning horizon, $i=1,\ldots,N$
D_i	Number of products demanded in period i
R_i	Number of products returned at the beginning of period i
K^S	Set-up cost to manufacture new units (or serviceables)
K^R	Set-up cost to remanufacture a returned unit
h^S	Holding cost to carry a unit of serviceable inventory from period i to period $i+1$
h^R	Holding cost to carry a returned unit in inventory from period $i$to period $i+1$
Mixed integer linear programming (MILP) formulations	
d_{ij}	Cumulative demand from period $i$to period j
r_{ij}	Cumulative returns from period $i$to period j
Q_i^S	The quantity of products manufactured in period i
Q_i^R	The quantity of products remanufactured in period i
I_i^S	Inventory in units of serviceables left over at the end of period i
I_i^R	Inventory in units of returns left over at the end of period i
y_i^S	Binary variable $=1$ if new products are manufactured in period$i$; 0 otherwise
y_i^R	Binary variable $=1$ if returns are remanufactured in period$i$; 0 otherwise
x_{ij}^S	Binary variable $=1$ if manufacturing occurs in period$i$and next in period$(j+1)$; 0 otherwise
x_{ij}^R	Binary variable $=1$ if remanufacturing occurs in period$i$and next in period $(j+1)$; 0 otherwise
Constructive heuristic	
D_i^S	The part of demand $D_i$in period $i$satisfied by manufacturing new products
D_i^R	The part of demand $D_i$in period $i$satisfied by remanufacturing returned products
γ_i	The target end-of-period inventory of returned products for period i
$e'_{ij}(s,t)$	Total manufacturing cost in period i to meet demands D_k^S for $k=i,i+1,\ldots,j$
$f_j(s,t)$	The minimum total set-up and holding costs of meeting the demands $D_i^S$by manufacturing only from period s through$j$
$e''_{ij}(s,t)$	Total remanufacturing cost in period i to meet demands D_k^R for $k=i,i+1,\ldots,j$
$g_j(s,t)$	The minimum total set-up and holding costs of meeting the demands D_i^R by remanufacturing only from period s through j
$c(s,t)$	Optimal total set-up and holding costs of satisfying the demands D_i from period s through t

3. Review of the literature related to model (P)

Richter and Sombrutzki [21] study model (P) (presented in Section 2) with the additional restriction that enough product returns are available at the start of the planning period to cover demands over the entire horizon. This assumption makes it possible to transform model (P) to a problem instance that preserves the all-important “zero-inventory property.” This property states that replenishments are to take place in a given period only if the starting inventory in that period is zero. As a result of this, the authors show that optimal solutions can be calculated using a Wagner–Whitin algorithm. Richter and Weber [22] extend the analysis given in Richter and Sombrutzki [20] by adding variable manufacturing and remanufacturing costs to the objective function (1).

Golany et al. [11] consider a more general formulation of model (P) that allows for concave cost functions and disposal of surplus inventory in each period. They show that the problem can be reformulated and stated as a concave flow network and demonstrate that it is NP-hard. For linear-cost case, they show that the model can be viewed as a transportation problem, which can be solved using an algorithm with $O(N^3)$ time. Interestingly, Yang et al. [32] continue the work done in Golany et al. [11], but in their model the costs involved in the objective function are represented by concave functions. They observe that optimal solutions of a problem are the extreme points of the feasible region. These extreme points possess a special spanning-tree property with distinct flow patterns. Therefore, utilizing this property, they develop a polynomial-time heuristic for solving the problem. As well, Teunter et al. [29] examine model (P) in greater detail. First, they consider the situation in which there is a joint set-up cost for manufacturing and remanufacturing. This is a special case of model (P). The authors construct a dynamic programming algorithm that runs in $O(N^4)$ time for solving this particular problem. Next, they consider the more general case where the set-up costs for the two manufacturing activities are different, that is, model (P) given in (1)–(5). Since the zero-inventory property does not hold for this model instance, the authors are led to the conjecture that the problem is NP-hard. As a result, they propose heuristic methods by adapting some popular lot-sizing techniques, such as the Silver–Meal (SM) heuristic [24] to cope with the problem. Schulz [23] offers a different solution; he modifies the SM heuristic discussed in Teunter et al. [24], by proposing additional improvements that reduce the error gap noticeably.

It is also important that we now turn our attention on the dynamic lot sizing problems that incorporate both product returns and disposal of excess inventory which is what Beltran and Krass [4] do in one instance. Their model, though, does not include the possibility of remanufacturing returns as in model (P). In other words, each return is considered to arrive as good as new and is immediately added to serviceable inventory. They show that a generalized version of the zero-inventory property still holds for this problem. Keeping this in mind, the authors construct a dynamic programming algorithm with $O(N^3)$ complexity to solve the problem. In a companion paper, Li et al. [17] devise a Tabu search algorithm based on the alternative MILP model formulation discussed in Section 4. The algorithm solves the problem of several small linear programs, each one corresponding to a sub-problem of the original model. It is sophisticated in its construction, and the method produces excellent results. Other researchers present different formulations on this same issue. Helmrich et al. [16] analyze the complexity of the problem under a more general cost structure.

There are many other quantitative models besides model (P) that examine various product recovery options including remanufacturing, repairing, refurbishing, and recycling (see [15,19,20] for recent surveys). Some of these works make use of EOQ models (see [28, 27, 2], for example), while others are in the context of stochastic models (see [30, 256] for example). Before we present an alternative formulation for model (P) and contrast it, we must then next show that model (P) is indeed NP-hard as was conjectured in Teunter et al. [29].

Theorem 1. *The optimization problem stated in (P) is NP-hard.*

Proof. See Appendix.■

4. An alternative MILP model formulation

We propose an alternative MILP model for model (P) defined by Eqs. (1)–(5). In the same spirit as in the classical economic lot sizing problem, the quantity manufactured (or remanufactured) in a given period is set to meet part or all the requirements of an integer number of periods into the future. With the option of remanufacturing now available, the manufacturing quantity in period i, Q_i^S, may only be used to satisfy part of the total requirements of period i to some future period j. Similarly, if we also remanufacture Q_i^R units in period i, this quantity may also be used to meet part of the total demand of period i to some future period j'.

For $1 \le i \le j \le N$, we let $d_{ij} = \sum_{k=i}^{j} D_k$ and $r_{ij} = \sum_{k=i}^{j} R_k$. For notational convenience, we also add two fictitious periods 0 and $(N+1)$, in which demands are zero and so no manufacturing activity whatsoever happens. We further define the following:

$$x_{ij}^S = \begin{cases} 1 & \text{if manufacturing occurs in period } i \text{ and next in period } (j+1) \\ 0 & \text{Otherwise} \end{cases} \quad 0 \le i \le j \le N$$

and

$$x_{ij}^R = \begin{cases} 1 & \text{if remanufacturing occurs in period } i \text{ and next in period } (j+1) \\ 0 & \text{Otherwise} \end{cases} \quad 0 \le i \le j \le N$$

To be more clear, x_{ii}^S (x_{ii}^R)=1 implies that new (core) products are successively manufactured (remanufactured) in periods i and $i+1$. To explain further, $x_{0j}^S(x_{0j}^R)=1$ means that no manufacturing (remanufacturing) occurs in the first j periods and the first manufacturing (remanufacturing) occurs in period $(j+1)$. Not to mention, $x_{00}^S(x_{00}^R)=1$ implies that a manufacturing (remanufacturing) occurs in period 1. Finally, $x_{0N}^S(x_{0N}^R)=1$ means that no manufacturing (remanufacturing) occurs from beginning to end.

The MILP problem (P') can be given by

$$\text{Min } h^s \sum_{i=1}^{N} I_i^s + h^R \sum_{i=1}^{N} I_i^R + K^s \sum_{i=1}^{N} \sum_{j=i}^{N} x_{ij}^S + K^R \sum_{i=1}^{N} \sum_{j=i}^{N} x_{ij}^R \tag{6}$$

$$I_i^S = I_{i-1}^S + Q_i^S + Q_i^R - D_i \quad \forall i = 1, 2, \dots, N \tag{7}$$

$$I_i^R = I_{i-1}^R + R_i - Q_i^R \quad \forall i = 1, 2, \dots, N \tag{8}$$

$$Q_i^S \le \sum_{j=i}^{N} d_{ij} x_{ij}^S \quad \forall i = 1, 2, \dots, N \tag{9}$$

$$Q_i^R \le \sum_{j=i}^{N} d_{ij} x_{ij}^R \quad \forall i = 1, 2, \dots, N \tag{10}$$

$$\sum_{j=0}^{N} x_{0j}^S = 1 \tag{11}$$

$$\sum_{i=0}^{j} x_{ij}^S = \sum_{k=j+1}^{N} x_{(j+1)k}^S \quad \forall j = 1, 2, \dots, N-1 \tag{12}$$

$$\sum_{i=0}^{N} x_{iN}^S = 1 \tag{13}$$

$$\sum_{j=0}^{N} x_{0j}^R = 1 \tag{14}$$

$$\sum_{i=0}^{j} x_{ij}^R = \sum_{k=j+1}^{N} x_{(j+1)k}^R \quad \forall j = 1, 2, \dots, N-1 \tag{15}$$

$$\sum_{j=0}^{N} x_{jN}^R = 1 \tag{16}$$

$$x_{ij}^S, x_{ij}^R \in \{0,1\}, Q_i^S, Q_i^R, I_i^S, I_i^R \geq 0, \quad \forall i = 0,1,2,\ldots,N$$
$$\text{and} \quad j = 0,1,2,\ldots,N$$

Eqs. (7) and (8) are inventory balance equations. Eq. (9) states that if new products are manufactured in period i and no manufacturing is done in periods $(i+1),(i+2),\ldots,j$, then the quantity of new products manufactured must be less than or equal to the demand in periods $i,(i+1),\ldots,j$. Similarly, Eq. (10) states that if returned products are remanufactured in period i and no remanufacturing is done in periods $(i+1),(i+2),\ldots,j$, then the quantity remanufactured must be less than or equal to the total demand in periods $i,(i+1),\ldots,j$.

Eqs. (11)–(13) determine when to *manufacture*. Eq. (11) states that either the first manufacturing occurs in period j, $1 \leq j \leq N$, so that $x_{0(j-1)}^S = 1$ or there is no manufacturing at all and $x_{0N}^S = 1$. Eq. (12) states that if (and only if) a manufacturing lot covers demands for a set of periods ending at period j, the next lot covers the demand for period $(j+1)$. Eq. (13) states that either there is no manufacturing, so that $x_{0N}^S = 1$ or a manufacturing lot covers some periods ending in period N.

Conversely, Eqs. (14)–(16) determine when to *remanufacture*. Eq. (14) states that either the first remanufacturing occurs in period j, $1 \leq j \leq N$, so that $x_{0(j-1)}^R = 1$ or there is no remanufacturing at all and $x_{0N}^R = 1$. Eq. (15) states that if (and only if) a remanufacturing lot covers some periods ending in period j, the next lot covers period $(j+1)$. Eq. (16) states that either there is no remanufacturing, so that $x_{0N}^R = 1$ or a remanufacturing lot covers some periods ending in period N.

In Theorem 2, we show that this alternative model (P') has one important advantage over the earlier model (P) discussed in Section 2. The lower bounds obtained when the integrality constraints are relaxed are, in general, much better.

Theorem 2. *Let $\overline{P}$ and $\overline{P}'$ be the LP relaxation of P and P', respectively. Every feasible solution to $\overline{P}'$ gives a feasible solution to $\overline{P}$. Accordingly, the objective function value (OFV), associated with $\overline{P}'$ yields a lower bound that is, at least, as large as that for the corresponding OFV of $\overline{P}$.*

Proof. See Appendix.■

5. A constructive heuristic solution procedure for solving model (P')

This section discusses a heuristic solution procedure that can be used to solve model (P'). The idea behind our heuristic stems from the following realization. Each feasible solution to model (P') can be decomposed into a series of well-structured blocks with distinct patterns in the set-ups for manufacturing and remanufacturing. Each block consists of a set of consecutive periods. A block may contain a string of manufacturing set-ups, a string of remanufacturing set-ups or both: manufacturing set-ups followed by remanufacturing set-ups. More precisely, a block (s,t) is defined as a set of consecutive periods $s, s+1, \ldots, t$ such that it contains at least one period in which a manufacturing or remanufacturing set-up occurs and if there exists a manufacturing set-up in period i, $s \leq i \leq t$ and a remanufacturing set-up in period j, $s \leq j \leq t$, then $i \leq j$. Table 2 illustrates examples of block patterns.

In example A, two manufacturing set-ups in periods 1 and 3 are followed by a remanufacturing set-up in period 4. Therefore, periods 1–6 constitute a block. The second example B shows a block with periods 1–4 where both manufacturing and remanufacturing are done in period 3. In example C, periods 1–5 do not constitute a block because the remanufacturing set-up in period 4 comes before the manufacturing set-up in period 5. However, periods 1–4 constitute a block, while period 5 is a block by itself.

Table 2
Examples of block patterns.

A							B					C					
i	1	2	3	4	5	6	i	1	2	3	4	i	1	2	3	4	5
y_i^S	1	0	1	0	0	0	y_i^S	1	0	1	0	y_i^S	1	0	0	0	1
y_i^R	0	0	0	1	0	0	y_i^R	0	0	1	1	y_i^R	0	0	0	1	0

Alternatively, for this example, we may also consider three separate blocks, periods 1–3 as one block, period 4 as another, and period 5 as the third block. This block structure forms the backbone of our heuristic. Since every feasible solution comprises a chain of blocks, our problem boils down to the construction of an optimal chain of blocks. Further details are given next.

5.1. Heuristic construction

For each period i demand D_i can be viewed from 2 separate sources, one that manufactures new products only, and another one that remanufactures returned units only. The demand in period i, D_i can then be partitioned into 2 demand components D_i^S and D_i^R, where $D_i = D_i^S + D_i^R$, $\quad i = 1, 2, \ldots, N$. Thus, for $1 \leq i \leq N$, we let D_i^S be the portion of demand in period i allocated to the manufacturing of new products and D_i^R be the portion of demand in period i allocated to the remanufacturing of returned products.

With this interpretation, for any given D_i^S values $\forall i$, the manufacturing quantities Q_i^S $\forall i$ can be optimally computed in polynomial time using a shortest-path type dynamic program. Similarly, given D_i^R values such that $D_i^R = D_i - D_i^S$ $\forall i$, the optimal remanufacturing quantities Q_i^R $\forall i$ can also be calculated using a polynomial-time algorithm of a similar nature.

Consider a network with $(N+1)$ nodes numbered $i = 0, 1, 2, \ldots, N$. Each node $i > 0$ represents the end of period i and node 0 represents the beginning of period 1. For $1 \leq i \leq j \leq N$, an arc from node i to node j represents that production of new units occurs in period $(i+1)$, does not occur in any of the following periods $(i+2)$ to j, and then occurs again in period $(j+1)$. Hence, the lot size in period $(i+1)$ is given by $Q_{i+1}^S = D_{i+1}^S + D_{i+2}^S + \cdots + D_j^S$. Every path from node 0 to node N in this network corresponds to a production schedule and vice versa. However, a minimum cost path from node 0 to node N gives an *optimal production schedule*. In the same fashion, we can represent the remanufacturing problem of meeting the D_i^R demands at minimum cost with a similar network.

Let us focus on the problem of satisfying all the demands from periods S to t, where $1 \leq s \leq t \leq N$. In solving this problem, we restrict our attention to solutions that will exhibit the block format described above. We say that this problem is associated with block (s,t) that begins and ends with the following inventories: $I_{s-1}^S = 0$, $I_t^S = 0$, $I_{s-1}^R = \gamma_{s-1}$, and $I_t^R = \gamma_t$. The symbol γ_i represents an appropriately chosen target inventory of returned products for period i whenever a block ends in period i. In our implementation, we set $\gamma_0 = I_0^R$ and then recursively compute γ_i, $\forall i = 1, 2, \ldots, N$, as $\gamma_i = \max(0, \gamma_{i-1} + R_i - D_i)$.

Notice that for the important special case in which $D_i \geq R_i \forall i$, the γ_i values will be equal to zero for each i. In other words, if demand is greater than returns in each period, then each block begins and ends with zero inventories of new and returned products. However, if $D_i < R_i$ for some i, then it may not be feasible to end a block with zero inventory of returned products. In such a case, we set positive target inventories of $I_i^S = 0$, and $I_i^R = \gamma_i$ whenever a block ends in period i. This choice of target inventories for each block is rationalized by the following theorem.

Theorem 3. *(1) For every feasible solution to (P), $I_i^S + I_i^R \geq \gamma_i$ $\forall i$. (2) There exists a solution to (P) for which $I_i^S = I_0^S$ and $I_i^R = \gamma_i$ $\forall i$.*

Proof. See the appendix.■

It is always feasible to construct a block ending in period i with target inventories equal to $I_i^S = 0$, and $I_i^R = \gamma_i$. Such target inventories may not be optimal for the block in question. However, this approach assures simplicity in that it side-steps the need to introduce an additional state variable to keep track of ending inventories in the dynamic programming recursion explained below. Clearly, the addition of another variable will render the computations and the proposed heuristic much more complicated.

Let $c(s,t)$ be the optimal total set-up and holding costs associated with block (s,t). Unlike the approach taken in Li et al. [17], which solves the problem associated with block (s,t) as a linear program in parts of the heuristic, we shall use a different approach here and show how to compute $c(s,t)$ by using two separate Wagner–Whitin dynamic programs.

Towards this end, let $f_u(s,t)$ be the minimum total set-up plus carrying costs of satisfying the demand D_i^S through manufacturing only from period S to some period u, $s \leq u \leq t$. In the same way, let $g_v(s,t)$ be the corresponding costs of meeting the demands D_i^R through remanufacturing only from some period v to period t, $s \leq u \leq v \leq t$. To compute $f_u(s,t)$ and $g_v(s,t)$, we need to show how the $D_i^S(s,t)$ and $D_i^R(s,t)$ values for block (s,t) are constructed and at the same time explain how periods u and v are determined.

First, if $D_i = 0$ $\forall s \leq i \leq t$, we set $Q_i^S = Q_i^R = 0$ $\forall s \leq i \leq t$. The cost of block (s,t) will be $h^R(\gamma_{s-1}(t-s+1) + \sum_{i=s}^t R_i(t-i+1))$. Otherwise, if $D_i \neq 0$ for some i, $s \leq i \leq t, D_i^S(s,t)$ and $D_i^R(s,t)$ $\forall s \leq i \leq t$ are calculated as shown below in steps (a)–(c). We then apply dynamic programming recursions to compute the Q_i^S and Q_i^R quantities, and the resulting cost of block (s,t).

If $\gamma_{s-1} + \sum_{i=s}^t R_i = 0$, then block (s,t) is a manufacture-only block. In this case, we set $D_i^S(s,t) = D_i \forall i$, $D_i^R(s,t) = 0 \forall i$ and $u = t$.

If $\gamma_{s-1} + \sum_{i=s}^t R_i > 0$, let δ_i be the shortage of returns corresponding to period i. We initialize $\delta_{s-1} = -\gamma_{s-1}$ and recursively compute $\delta_i = \delta_{i-1} + D_i - R_i \forall s \leq i \leq t$. If $\max_i(\delta_i) \leq 0$, there is no shortage of returns and block (s,t) is a remanufacture-only block. As a result, we set $D_i^S(s,t) = 0$ $\forall i, D_i^R(s,t) = D_i \forall i$ and $v = s$.

If $\max_i(\delta_i) > 0$, block (s,t) is a mixed block and the total manufacturing and remanufacturing quantities are set equal to $\max_i(\delta_i)$ and $\sum_{i=s}^t D_i - \max_i(\delta_i)$, respectively. We let v be the largest integer such that

$$\Delta(s,t) = \sum_{i=v}^{t} D_i - \sum_{i=s}^{t} D_i + \max_i(\delta_i) \geq 0.$$

If $\Delta(s,t) = 0$, we let $u = v-1$; otherwise we let $u = v$. In this case, $D_i^S(s,t)$ and $D_i^R(s,t)$ are given by

$$D_i^S(s,t) = \begin{cases} D_i, & s \leq i < v \\ \Delta(s,t), & i = v \\ 0, & v < i \leq t \end{cases}$$

$$D_i^R(s,t) = D_i - D_i^S(s,t)\,, \quad s \leq i \leq t,$$

with the understanding that if $v = s$ or $v = t$ then $\{s \leq i < v\}$ and $\{v < i \leq t\}$ are empty sets.

We observe that both $f_u(s,t)$ and $g_v(s,t)$ can be computed by means of Wagner–Whitin dynamic programming (DP) recursions. The procedures for calculating $f_u(s,t)$ and $g_v(s,t)$are shown as Recursions 1 and 2. Moreover, the total cost corresponding to block(s, t) is given by $c(s,t) = f_u(s,t) + g_v(s,t)$.

Recursion 1. Let $e'_{i,j}(s,t) = K^S + h^S \sum_{k=i+1}^j (k-i-1) D_k^S (s-1) \leq i < j \leq u$ and $f_{s-1}(s,t) = 0$. Then $f_j(s,t) = \min_i(f_i(s,t) + e'_{i,j}(s,t) \;/\; i = s-1, s, \ldots, j-1)$ for $j = s, s+1, \ldots, u$.

To explain further, in Recursion 1, $e'_{i,j}(s,t)$ is the total cost of manufacturing in period i to meet the demands D_k^S for periods $k = i, i+1, \ldots, j$, whereas $f_j(s,t)$ is the minimum total cost for periods $s, s+1, \ldots, j$. In Recursion 2, $e''_{i,j}(s,t)$ and $g_j(s,t)$ are analogously defined except that remanufacturing is used to meet the demands D_k^R. Note that $e''_{i,j}(s,t)$ assumes a finite value only if enough returns are available to satisfy the D_k^R values for periods $k = v, v+1, \ldots, j$. In addition, in recursion 2, $g_{v-1}(s,t)$ represents the total holding costs of keeping in storage all the returns for periods $s, s+1, \ldots, v-1$ for possible use in periods v to t.

Recursion 2. For i and j such that $(v-1) \leq i < j \leq t$, let

$$e''_{i,j}(s,t) = \begin{cases} 0 & \text{if } \gamma_{s-1} + \sum_{k=s}^{i+1} R_k = 0 \\ K^R + (h^S - h^R) \sum_{k=i+1}^{j} (k-i-1) D_k^R & \text{if } \gamma_{s-1} + \sum_{k=s}^{i+1} R_k \geq \sum_{k=v}^{j} D_k^R > 0 \\ \infty & \text{Otherwise} \end{cases}$$

and

$$g_{v-1}(s,t) = h^R \sum_{l=s}^{t} \left(\gamma_{s-1} + \sum_{k=s}^{l} R_k - \sum_{k=s}^{l} D_k^R \right).$$

Then

$$g_j(s,t) = \min_i(g_i(s,t) + e''_{i,j}(s,t) \;/\; i = v-1, v, \ldots, j-1) \quad \text{for}$$
$$j = v, v+1, \ldots, t$$

Finally we find the optimal blocks by using the following shortest-path algorithm.

Recursion 3. Set $h(0) = 0$ and recursively compute $h(j) = \min_i(h(i) + c(i+1,j) \mid i = 0, 1, \ldots, j-1)$ for $j = 1, 2, 3, \ldots, N$. The optimal blocks can be found by backtracking.

5.2. Heuristic improvement

In what follows, we propose some improvement steps that can be used individually or in combination to further reduce the cost difference between the heuristic solution and the optimal solution.

Inter-block trapezoid improvement: If 2 remanufacturing set-ups occur between 2 manufacturing set-ups, then there can be an improvement opportunity. More specifically, if remanufacturing occurs in periods j and k and manufacturing occurs in periods i and l with $i \leq j < k < l$ and $w = Q_j^R < \min(Q_i^S, Q_k^R, Q_l^S)$, then an improved schedule is obtained by moving the manufacturing of w units from period l to i and by further moving the remanufacturing of w units from period j to k if

$$K^R + w(h^S - h^R)(k-j) - wh^S(l-i) > 0$$

Elimination of remanufacturing set-up: For each period j with a remanufacturing set-up, an improved schedule can be obtained by eliminating a remanufacturing set-up from period j and manufacturing Q_j^R units in period $i \leq j$ if savings

$$K^R - K^S(1 - y_i^S) - Q_j^R \{h^R(N-j+1) + h^S(j-i)\} > 0$$

We search for j backwards and choose period i which gives a maximum positive savings.

Mutual optimal solution: For a given remanufacturing plan, the manufacturing schedule can be optimized. We allocate demand to manufacturing as follows: let $I_0^S = 0$ and do the following for

Table 3
Results of the heuristic procedure applied to the numerical example given in Section 5.3.

Period, i	1	2	3	4	5	Cost of block
γ_i	17	14	0	5	0	
Block	$D_i^S, D_i^R, Q_i^S, Q_i^R$					
1,1	0,23,0,23					30.2
1,2	0,23,0,37	0,14,0,0				44.2
1,3	4,19,4,33	0,14,0,0	0,25,0,25			109
1,4	4,19,4,33	0,14,0,0	0,25,0,25	0,0,0,0		112
1,5	23,0,54,0	14,0,0,0	17,8,0,8	0,0,0,0	0,72,0,72	245.6
2,2		0,14,0,14				28.6
2,3		4,10,4,10	0,25,0,25			90.8
2,4		4,10,4,10	0,25,0,25	0,0,0,0		93.8
2,5		14,0,39,0	25,0,0,0	0,0,0,0	15,57,15,57	186.8
3,3			4,21,4,21			60
3,4			4,21,4,21	0,0,0,0		63
3,5			25,0,25,0	0,0,0,0	29,43,29,43	128.2
4,4				0,0,0,0		3
4,5				0,0,0,0	50,22,50,22	63
5,5					50,22,50,22	60

$i = 1, 2, \ldots, N$

$I_i^S = \max(0, I_{i-1}^S + Q_i^R - D_i)$
$D_i^S = \max(0, D_i - I_{i-1}^S - Q_i^R)$

Given these D_i^S values $\forall i$, the Q_i^S values $\forall i$ can be computed using Recursion 1 with $s = 1, u = t = N$.

Similarly, for a given manufacturing plan, the remanufacturing schedule can be optimized. Let $I_0^S = 0$ and do the following for $i = 1, 2, \ldots, N$,

$I_i^S = \max(0, I_{i-1}^S + Q_i^S - D_i)$
$D_i^R = \max(0, D_i - I_{i-1}^S - Q_i^S)$

Given these D_i^Rvalues $\forall i$, the Q_i^R values $\forall i$ can be computed using Recursion 2 with $s = v = 1$, $t = N, \gamma_0 = 0$, $\gamma_N = \sum_{i=1}^N Q_i^S + \sum_{i=1}^N R_i - \sum_{i=1}^N D_i$.

5.3. Illustrative numerical example

We illustrate our constructive heuristic by applying it to a 5-period problem with the following parameter values. We let $N = 5$, $D_1 = 23$, $R_1 = 40$, $D_2 = 14$, $R_2 = 11$, $D_3 = 25$, $R_3 = 7$, $D_4 = 0$, $R_4 = 5$, $D_5 = 72$, $R_5 = 17$, $K^S = \$40$, $K^R = \$20$, $h^S = \$1.00$, and $h^R = \$0.60$.

The results of the procedure are presented in a table format and are summarized in Table 3.

A shortest path in Fig 3 is 0–2–4–5 with a cost of \$167.2. Hence, the corresponding blocks are [1,2], [3,4] and [5,5].

The corresponding production quantities are given by $Q_1^S = 0$, $Q_2^S = 0$, $Q_3^S = 4$, $Q_4^S = 0$, $Q_5^S = 50$, $Q_1^R = 37$, $Q_2^R = 0$, $Q_3^R = 21$, $Q_4^R = 0$, and $Q_5^R = 22$.

Furthermore, after applying improvement step 2, the optimal solution to the problem is given by $Q_1^S = 0$, $Q_2^S = 0$, $Q_3^S = 4$, $Q_4^S = 0$, $Q_5^S = 72$, $Q_1^R = 37$, $Q_2^R = 0$, $Q_3^R = 21$, $Q_4^R = 0$, and $Q_5^R = 0$ yielding a minimum cost of \$160.40.

6. Numerical experience

We test our heuristic solution procedure described in Section 5 for 23,760 test problems with 12-period planning horizon. These tests have been conducted on a HP Z400 64-bit computer equipped with an Intel Xeon CPU with speed of 2.67 GHz and 6 GiB DDR3 memory. The computing environment includes Gentoo Linux operating system and compiler GCC version 4.7.1. Lastly, the heuristic was coded in C++ programming language.

Moreover, to allow the reader to make meaningful comparison of the results reported here to those discussed in Teunter et al. [29] and Schulz [23], we use the same numerical experiment given in Teunter et al. [29] to test the proposed heuristic. As in Teunter et al. [29], we generate demand and return flows over the 12 periods of the planning horizon as follows: $D_i = \mu + \tau(i-1) + a \sin[(2\pi i/c) + d(\pi/2)] + \varepsilon_i$ for $i = 1, \ldots, N$, where μ is the starting level of pattern, τ is the trend level, a is the amplitude of the cycle, c is the cycle length, d represents the peak of the cycle and $\varepsilon_i (i = 1, \ldots, N)$ are independent normally distributed random variables with standard deviation σ. This model allows one to generate a wide range of demand and return patterns (stationary, linearly increasing and/or decreasing, and seasonal) that are likely to be encountered in practice.

The set-up costs for manufacturing and remanufacturing are given the values 200, 500, and 2000. The holding cost for serviceable is normalized to 1 and the holding cost for cores is varied as 0.2, 0.5, and 0.8. In addition, as in the Teunter et al. [29] study, we generate and test 4 separate realizations for each demand and return patterns. Altogether, our experiment includes a total of 23,760 test problems – 10 types of demand patterns, 22 types of return patterns, 3 levels for each of the cost parameters K^S, K^R and h^R, and 4 replicates for each treatment combination. The data set and cost parameters used in our experiment are reproduced here in Table 4 for reader's convenience.

To provide insight, we conduct our numerical experiment in 3 phases. In the first phase, we consider the special, but important, case in which the demand is greater than the amount returned in each period of the planning horizon, i.e. $D_i \geq R_i \forall i$. This situation is likely to be the case for products experiencing rapid growth, or that are in the mature stage of the life cycle. Furthermore, for this situation, we restrict our attention to policies that will reuse all the acquired cores by the end of the planning horizon, i.e. $I_N^R = 0$. Such a policy is likely to have practical appeal to firms for which the acquired cores have no value beyond the planning horizon and also since disposing of them is considered environmentally unfriendly. The Breeze-Eastern company can be cited as an example for this situation [18]. Breeze-Eastern makes, among other products, cargo hooks, as well as rescue hoist systems for helicopters. As part of its sustainability strategy, Breeze-Eastern collects and remanufactures used cargo hooks. Once remanufactured, these hooks are as good as new. Since the demand rate is much higher than the return rate of used cargo hooks, Breeze-Eastern remanufactures the entire inventory of collected hooks over the planning period.

For this special case, we have $\gamma_i = 0 \; \forall i$. To find the different components defining block (s, t), we let $\delta_i = \sum_{j=s}^{i} (D_j - R_j)$, and let v be the largest integer such that: $\Delta(s,t) = \sum_{i=v}^{t} D_i - \sum_{i=s}^{t} R_i$. Since $\max_i(\delta_i) = \sum_{j=s}^{t} (D_j - R_j) \geq 0$, the case with remanufacture-only blocks never arises. The blocks are either empty $(D_i = 0)$ or manufacture-only $(R_i = 0)$ or mixed. In addition, improvement steps 1 and 3 can be helpful in this situation. We note that this special case of model (P) with the additional constraints that $D_i \geq R_i \forall i$ and $I_N^S = I_N^R = 0$ is also an NP-hard optimization problem [3].

In the second phase of our numerical study, we put our attention on the general case, i.e. problem (P). Table 5 summarizes the results of applying our heuristic algorithm and improvement scheme for the 23,760 test problems. The results corresponding to the special case $D_i \geq R_i \forall i$ and $I_N^S = I_N^R = 0$ are shown in parentheses in the tables and the comments we make below.

Table 5 shows the average percentage cost errors (increase in total costs when compared to the cost of optimal solutions) and the standard deviation of errors for the different treatment combinations. As can be seen from this table, our heuristic generates solutions

Table 4
Data set used in the computational study.

Demand pattern						Return pattern						
μ	σ	τ	a	c	d	μ	σ	τ	a	c	d	
Stationary						Stationary						
100	10	0	0	n.a.	n.a.	(1)	30	3	0	0	n.a.	n.a.
100	20	0	0	n.a.	n.a.	(2)	30	6	0	0	n.a.	n.a.
						(3)	50	5	0	0	n.a.	n.a.
						(4)	50	10	0	0	n.a.	n.a.
						(5)	70	7	0	0	n.a.	n.a.
						(6)	70	14	0	0	n.a.	n.a.
Positive trend						Positive trend						
100	10	10	0	n.a.	n.a.	(1)	30	3	3	0	n.a.	n.a.
100	10	20	0	n.a.	n.a.	(2)	30	3	6	0	n.a.	n.a.
						(3)	70	7	7	0	n.a.	n.a.
						(4)	70	7	14	0	n.a.	n.a.
Negative trend						Negative trend						
210	10	−10	0	n.a.	n.a.	(1)	63	3	−3	0	n.a.	n.a.
320	10	−20	0	n.a.	n.a.	(2)	96	3	−6	0	n.a.	n.a.
						(3)	147	7	−7	0	n.a.	n.a.
						(4)	224	7	−14	0	n.a.	n.a.
Seasonal 1 (peak in middle)						Seasonal 1 (peak in the middle)						
100	10	0	20	12	1	(1)	30	3	0	6	12	1
100	10	0	40	12	1	(2)	30	3	0	12	12	1
						(3)	70	7	0	14	12	1
						(4)	70	7	0	28	12	1
Seasonal 2 (valley in middle)						Seasonal 2 (valley in middle)						
100	10	0	20	12	3	(1)	30	3	0	6	12	3
100	10	0	40	12	3	(2)	30	3	0	12	12	3
						(3)	70	7	0	14	12	3
						(4)	70	7	0	28	12	3
Cost parameters												
Parameter	Values											
K^S, K^R	200, 500, 2000											
h^R	0.2, 0.5, 0.8											
h^S	1											

within 4.28% (2.24%) of optimality over all instances. That is, the difference in total costs by applying this solution technique to these sample problems averaged only 4.28% (2.24%) above the cost for optimal solutions found by solving models (P) or (P') using CPLEX.

As is clearly shown, these results compare quite favorably against those of the Silver–Meal (SM) heuristic, which was used and tested in Teunter et al. [29] and Schulz [23]. The heuristic in Teunter et al. resulted in solutions within 8.3% of optimal solution on average. Teunter et al. tested other lot-sizing heuristics as well such as Least Unit Cost (LUC) and Part Period Balancing (PPB). Amazingly, they found that SM was the best performing heuristic in the group in terms of cost error. By incorporating additional improvement options to the Silver–Meal heuristic, Schulz [23] was able to reduce the average error gap to less than 3%. This is quite the feat. However, contrary to the more extensive experiment conducted in Teunter et al. [29] and also in our study, Schulz' results were limited only to the case with stationary demand and return patterns.

On top of all this, we compare the lower bounds obtained by solving the LP relaxations $\overline{P}$ and $\overline{P}'$ to problems P and P', respectively, for a total of 23,760 problem settings. We found that the minimum objective function value (OFV) associated with $\overline{P}$ was on the average about 55.9% (76%) below the minimum OFV to the integer program produced by CPLEX. On the other hand, the minimum OFV associated with $\overline{P}'$ was on the average 48.9% (53%) below the corresponding benchmark value. In other words, the MILP formulation P' shown in (6)–(16), in general, yields much better bounds when the 0–1 restrictions on the variables are relaxed. The improved bounds can be used when selecting a bounding strategy in searching for an optimal solution to the original problem (see [17]).

In the third phase of our numerical testing, we want to analyze the variability of the percentage cost error (PCE) through DOE (design of experiments) to see how some of the key parameters of the model affect this quantity. In our case, lower numbers of PCE are more desirable than higher numbers. In order to keep the number of factors that need to be investigated to a reasonable size, and to make the interpretation of the results somewhat easier, we consider the following smaller experimental design. Each cost factor is given 2 levels: low and high, except for h^S which is fixed to 1. In particular, K^S is given the values 200 and 2000; K^R is set to 200 and 500; h^R is varied as 0.2 and 0.8. In regards to the demand factor, we also consider 2 levels, D_1 and D_2. Levels D_1 and D_2 describe the situation in which the demand in each period of the 12-period planning horizon is drawn from a normal distribution with mean equal to 100 and standard deviations equal to 10 and 20, respectively. Likewise, the return factor is allowed to take on 4 levels, R_1, R_2, R_3 and R_4. Levels R_1 and R_2 (R_3 and R_4) represent the scenario in which the amount returned in each period of the 12-period planning horizon comes from a normal distribution with mean equal to 30 (70), and standard deviations equal to 3 and 6 (7 and 14), respectively. The purpose of this selection is to find out whether both the variance in returns and the return ratio (the average return rate relative to the mean demand rate) are key influencers of PCE. The return ratio changes from a low of 30% to a high of 70%.

Furthermore, the experiment is designed and conducted by replicating each of the 64 treatment combinations 5 times for a

Table 5
Performance of the heuristic method based on the data shown in Table 4.

Percentage cost error		
	Average	Standard deviation
All instances	4.28 (2.24) 8.46 (2.94)	
Demand pattern		
Stationary		
Small variance	4.10 (1.76)	6.93 (2.39)
Large variance	4.09 (1.95)	6.80 (2.41)
Positive trend		
Small trend	4.00 (3.00)	8.27 (3.40)
Large trend	3.42 (3.61)	5.79 (4.18)
Negative trend		
Small trend	5.19 (2.48)	11.86 (3.33)
Large trend	4.74 (1.29)	13.22 (2.49)
Peak in the middle		
Small amplitude	4.49 (2.14)	7.54 (2.49)
Large amplitude	4.79 (2.60)	8.43 (2.83)
Valley in the middle		
Small amplitude	4.05 (1.80)	6.12 (2.30)
Large amplitude	3.86 (1.79)	5.84 (2.15)
Return pattern		
Stationary		
(1)	3.09 (1.73)	4.96 (2.74)
(2)	3.43 (1.82)	5.26 (2.96)
(3)	5.27 (2.45)	11.49 (2.74)
(4)	4.87 (2.61)	12.94 (2.85)
(5)	4.65 (2.46)	8.52 (2.99)
(6)	4.81 (2.59)	8.70 (3.03)
Positive trend		
(1)	3.09 (2.70)	4.96 (2.93)
(2)	3.43 (2.34)	5.26 (2.34)
(3)	5.27 (1.53)	11.49 (2.68)
(4)	4.87 (2.01)	12.94 (2.63)
Negative trend		
(1)	4.94 (2.69)	7.39 (3.09)
(2)	7.22 (2.63)	13.74 (3.43)
(3)	10.50 (1.29)	17.50 (2.10)
(4)	1.26 (2.48)	2.40 (2.84)
Peak in middle		
(1)	2.47 (1.62)	4.42 (2.59)
(2)	2.46 (1.62)	4.22 (2.47)
(3)	4.86 (2.55)	8.46 (3.03)
(4)	4.80 (2.36)	8.66 (2.91)
Valley in middle		
(1)	2.93 (2.17)	5.06 (3.34)
(2)	3.03 (2.31)	5.26 (3.39)
(3)	4.98 (2.61)	9.09 (3.13)
(4)	5.76 (2.74)	9.46 (3.29)
Manufacturing set-up cost K^S		
200	3.33 (3.65)	5.93 (3.50)
500	4.11 (2.30)	8.67 (2.63)
2000	5.47 (0.78)	10.21 (1.59)
Remanufacturing set-up cost K^R		
200	1.91 (1.10)	4.32 (1.83)
500	6.53 (2.66)	10.29 (2.89)
2000	4.36 (2.97)	8.85 (3.49)
Returns holding cost h^R		
0.2	4.74 (2.13)	8.19 (2.54)
0.5	5.07 (2.32)	7.94 (2.96)
0.8	3.05 (2.28)	9.06 (3.26)

The values in parentheses correspond to the special case with $D_i \geq R_i \forall i$ and $I_N^S = I_N^R = 0$.

total of 320 experimental runs. To reduce the experimental error, the 320 runs are performed in completely random order. We conduct the analysis of variance (ANOVA) of PCE for this full factorial design using the statistical software package Minitab 15. Table 6 shows the ANOVA output from Minitab, while Figs. 4 and 5 display the main effect and interaction plots for PCE.

As can be seen from Table 6, there is a strong evidence, as judged by the p-values < 0.0005 and the main effects plots, to suggest that K^S, K^R, h^R, and the return-ratio level have a significant influence on PCE. For example, PCE increases as K^S and K^R are separately increased. This can be the case because our heuristic procedure, as it is designed, may at times include undue set-ups that can be avoided. On the contrary, PCE shows a moderate decrease when h^R increases. This can happen when solving the remanufacturing problem associated with each block as our heuristic may sometimes add extraneous remanufacturing set-ups that can also be eliminated when h^R is low.

To continue then, we observe that an increase in the return ratio from 30% to 70% has a clear influence on PCE, which increases from about 2% to roughly 8%. This is in agreement with the results of Table 5. Notice that the R_1 and R_2 levels (low number of returns in relation to the demand rate) correspond to the case in which $D_i \geq R_i$ for $i = 1, 2, \ldots, N$ in most test examples. As demonstrated in Table 5, our heuristic performs very well in this setting. When the return ratio is high (as in R_3 and R_4), our heuristic may occasionally add some unnecessary manufacturing set-ups that can lead to higher costs especially when K^S is high (see Fig. 5). Interestingly, the results of Table 6 and Figs. 4 and 5 suggest that an increase in the variability of demand per period, or amount returned per period has no significant effect on PCE.

There is strong evidence to suggest the presence of important interaction effects on PCE between some key parameters. As shown in Table 6 and Fig. 5, the most influential parameters on PCE are K^S, K^R, h^R, and the return ratio judging by their significant 2-way interactions (K^S and h^R, K^R and h^R, and K^S and return ratio) and 3-way interactions between K^S(or K^R), h^R and return ratio. For example, Fig. 5 shows that when K^S is low, PCE increases slightly when h^R changes from low to high. However, the reverse effect occurs when K^S is high. Comparatively, when K^S is high, Fig. 5 reveals that PCE experiences a steep increase as the return ratio goes from low to high. These results are not only *insightful*, but also *useful* in that they may be used to devise other improvement schemes that will reduce the error gap even further.

In terms of solution speed and PCE, our heuristic finds a near-optimal solution very efficiently compared to the time taken by CPLEX to solve the MILP model for large planning horizon. Table 7 shows the run times (in milliseconds) and PCE side by side as the number of periods N is increased. The results of Table 7 are based on the following problem setting: demand is assumed to be normally distributed with mean equal to 100 and standard deviation equal to 10; the amount returned is assumed to be normally distributed with mean equal to 30 and standard deviation equal to 3; the cost parameters are given the values $K^S = K^R = 200$, $h^S = 1$, and $h^R = 0.2$. For each size of the planning horizon N, we generate 10 random problem instances, and measure the PCE and the times taken by CPLEX, our heuristic, and CPLEX when started with the heuristic solution. Afterwards, we average the PCE and the resulting times of these 10 replications.

As can be seen from Table 7, our heuristic handles even larger problems very efficiently and effectively. While the run times of the heuristic and CPLEX started at the heuristic solution are much smaller than the run times of CPLEX, the PCE ranges from 1.05% to 4.20%. It is important to comment here that our heuristic solution procedure presented in Section 5 makes use of 3 separate Wagner–Whitin dynamic programs to find a solution to model (P) or (P'). The run times reported in Table 7 are based on a straightforward programming and implementation of the WW algorithm which requires $O(N^2)$ time, where N is the number of periods of the problem instance. Recursions 1 and 2 in our heuristic algorithm draw upon the WW algorithm to evaluate at

Table 6
Analysis of variance output from Minitab for percentage cost error.

Factor		Type levels				Values
K^S		Fixed 2				Low, high
K^R		Fixed 2				Low, high
h^R		Fixed 2				Low, high
D		Fixed 2				D_1, D_2
R		Fixed 4				R_1, R_2, R_3, R_4
Analysis of variance for error gap, using adjusted SS for tests						
Source	DF	Seq *SS*	Adj *SS*	Adj *MS*	*F*	*P*
K^S	1	477.73	477.73	477.73	44.99	0.000
K^R	1	790.12	790.12	790.12	74.41	0.000
h^R	1	216.24	216.24	216.24	20.36	0.000
D	1	0.74	0.74	0.74	0.07	0.792
R		33,599.00	3599.00	1199.67	112.98	0.000
$K^S \times K^R$	1	1.88	1.88	1.88	0.18	0.674
$K^S \times h^R$	1	435.97	435.97	435.97	41.06	0.000
$K^S \times D$	1	1.26	1.26	1.26	0.12	0.731
$K^S \times R$	3	733.34	733.34	244.45	23.02	0.000
$K^R \times h^R$	1	68.87	68.87	68.87	6.49	0.011
$K^R \times D$	1	0.55	0.55	0.55	0.05	0.820
$K^R \times R$	3	71.46	71.46	23.82	2.24	0.084
$h^R \times D$	1	0.03	0.03	0.03	0.00	0.961
$h^R \times R$	3	34.91	34.91	11.64	1.10	0.351
$D \times R$	3	50.61	50.61	16.87	1.59	0.192
$K^S \times K^R \times h^R$	1	10.34	10.34	10.34	0.97	0.325
$K^S \times K^R \times D$	1	3.04	3.04	3.04	0.29	0.593
$K^S \times K^R \times R$	3	50.29	50.29	16.76	1.58	0.195
$K^S \times h^R \times D$	1	0.05	0.05	0.05	0.00	0.946
$K^S \times h^R \times R$	3	761.28	761.28	253.76	23.90	0.000
$K^S \times D \times R$	3	28.69	28.69	9.56	0.90	0.441
$K^R \times h^R \times D$	1	6.42	6.42	6.42	0.60	0.437
$K^R \times h^R \times R$	3	107.43	107.43	35.81	3.37	0.019
Error	278	2952.00	2952.00	10.62		
Total	319	10,402.24				
$S=3.25863$	$R^2 = 71.62\%$	$R^2(\text{adj}) = 67.44\%$				

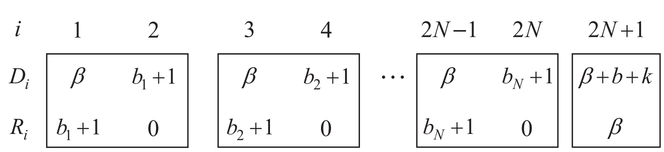

Fig. 1. Reduction scheme.

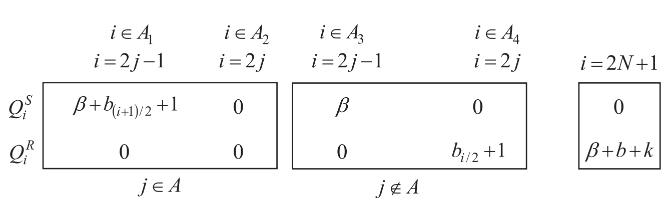

Fig. 2. Construction of a lot sizing solution given a partition.

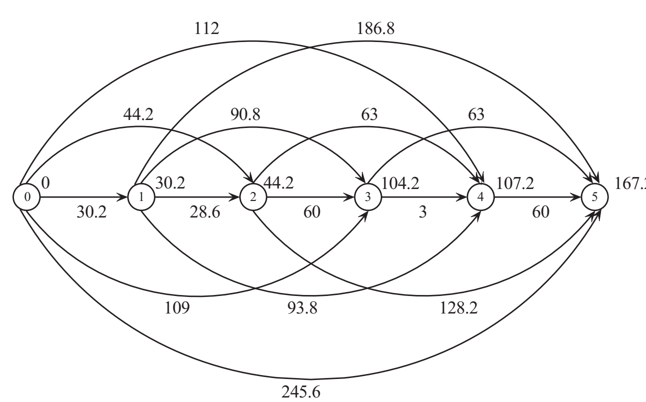

Fig. 3. Network representation of the illustrative numerical example.

most $N(N+1)/2$ blocks to find the optimal manufacturing and remanufacturing quantities. The capability of our algorithm can be said to require no more than $O(N^4)$ time in general. We note, however, that researchers have developed much faster solution procedures for the WW algorithm requiring only $O(N)$ time for solving problem instances for a cost structure similar to ours [7, 31,1]). Consequently, our heuristic algorithm can be implemented to require at most $O(N^2)$ time. This will lead to substantially lower solution times than the ones displayed in Table 7.

7. Conclusion

In today's production environments, remanufacturing is an established practice. This paper studies the dynamic lot size problem of product returns and remanufacturing. This problem is difficult to solve in the sense that there is no known algorithm that can find the optimal solution in polynomial time. To remedy this, a need for a heuristic method to solve this problem effectively is of paramount importance. In this paper, we propose such a heuristic that exploits and mimics the structure of optimal solutions. The heuristic is based on the observation that a feasible solution to this problem can be split into a sequence of blocks that shows a distinct structure in the way in which both manufacturing and remanufacturing set-ups occur. This has led us to construct an efficient heuristic that makes use of dynamic programming and the Wagner–Whithin algorithm to solve the problem.

Our extensive numerical testing shows that the procedure performs extremely well in terms of percentage cost error, which averages 4.85% in general, and 2.24% in the special case of $D_i \geq R_i \forall i$ and $I_N^S = I_N^R = 0$ in a total of 23,760 sample problems. The heuristic's performance compares favorably vis-à-vis with other competing heuristics such as the Silver–Meal that have been proposed in the literature as a solution procedure for this problem. Furthermore, since our heuristic is fast and produces high-quality solutions, it can be embedded within CPLEX to speed up the optimization process as shown in Table 7.

We also propose a totally different MILP formulation for the problem that has not been studied before. The main advantage of this formulation over the model formulation suggested in the literature is that the LP relaxation of the model produces much tighter bounds on solutions to the integer program. These bounds can be used when selecting a bounding strategy.

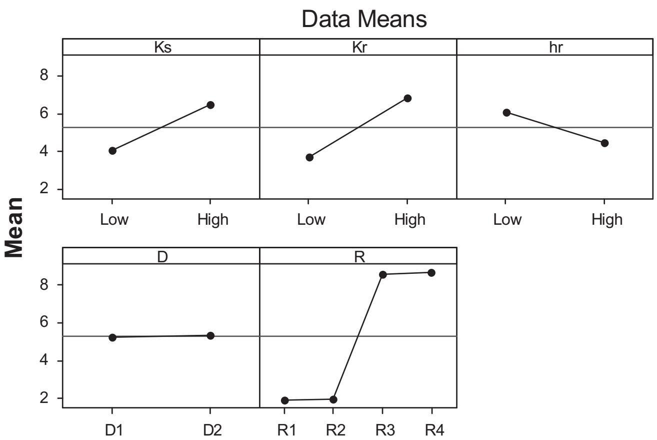

Fig. 4. Main effects plots for percentage cost error.

Acknowledgment

We would like to thank 2 anonymous referees for their constructive and helpful comments that have led to this much improved manuscript. The first and third authors would like to acknowledge the financial support from Natural Sciences and Engineering Research Council (NSERC). We would like to thank Bryan St.-Amour for writing computer programs. Thanks to Rosemary Scott for her careful reading of the manuscript and for her comments and suggestions on writing.

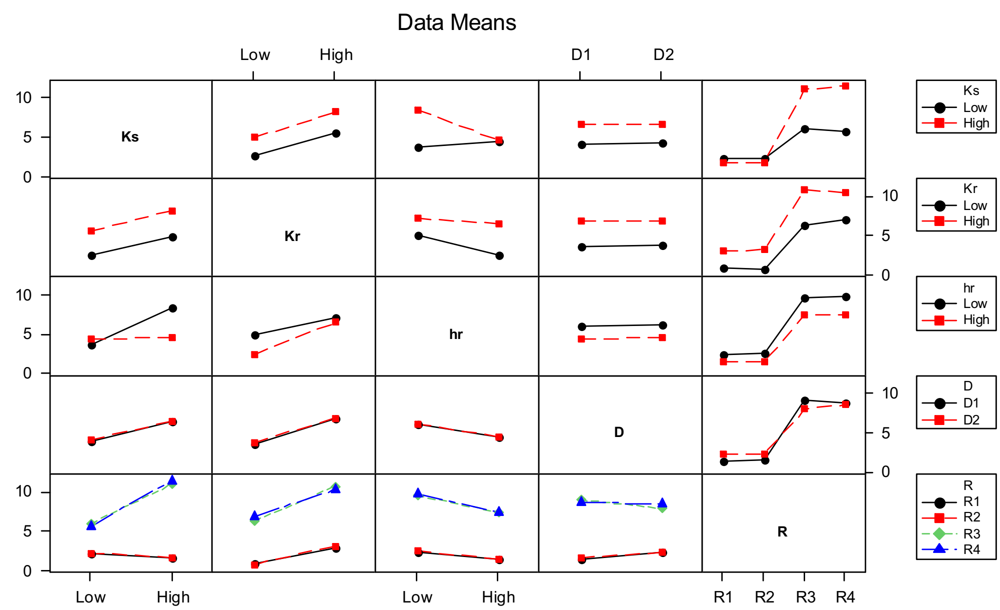

Fig. 5. Interaction plots for percentage cost error.

Table 7
Run times in milliseconds and percentage cost error as the number of periods, N, increases.

N	CPLEX	Heuristic	CPLEX started at heuristic solution	Percentage cost error
10	1957.9	0.0	34.4	4.1958
15	2926.4	0.0	75.7	1.0498
20	20,187.5	1.0	196.3	1.5608
25	25,690.3	2.0	710.8	2.3327
30	274,227.5	5.0	1281.3	1.1181
35	352,138.9	8.5	2250.8	1.5113
40	766,845.4	14.0	7210.9	1.9312
45	3,299,862.9	21.3	17,706.2	1.3706
50	5,036,067.1	31.6	36,575.3	1.2307
55	25,435,167.1	44.8	144,465.0	1.5356
60	109,118,058.4	62.2	441,652.9	1.3450

Appendix: Proofs

Proof of Theorem 1

We use a reduction from the following variation of the *Partition* problem, which is NP-hard: Given integers $b_1, b_2, b_3, \ldots, b_N$ with $\sum_{i=1}^{N} b_i = 2b$, is there a subset $A \subseteq \{1, 2, 3, \ldots, N\}$ with $|A| = k$ such that $\sum_{i \in A} b_i = b$? The proof strategy is as follows: we give a polynomial reduction scheme that builds an instance of problem (P) and a target cost. We clearly show that problem (P) has a solution with the target cost or better, if and only if, the Partition problem has a solution with $|A| = k$ such that $\sum_{i \in A} b_i = b$. This implies that if there exists a polynomial algorithm for problem (P), then it can be used to solve the Partition problem. For further details of this strategy, see Garey and Johnson [10].

Reduction scheme: Now we shall define an instance of problem (P) with $(2N+1)$ periods as follows. Let $K^S = 2kb$, $h^S = 1/(b+k)$, $K^R = h^R = 0$, $\alpha = (2Nkb+1)$ and $\beta = (\alpha + 2b + 1)\,(b+k)$

$$D_i = \begin{cases} \beta & \text{for } i = 1,3,5,\ldots,(2N-1) \\ b_{i/2}+1 & \text{for } i = 2,4,6,\ldots,2N \\ \beta+b+k & \text{for } i = 2N+1 \end{cases}$$

$$R_i = \begin{cases} b_{(i+1)/2}+1 & \text{for } i = 1,3,5,\ldots,(2N-1) \\ 0 & \text{for } i = 2,4,6,\ldots,2N \\ \beta & \text{for } i = 2N+1 \end{cases}$$

Fig. 1 is used to show that there exists a solution to problem (P) with total cost $\le \alpha$, if and only if, a subset A satisfies the Partition condition.

The if part: Suppose there exists a subset $A \subseteq \{1, 2, 3, \ldots, N\}$ with $|A| = k$ such that $\sum_{i \in A} b_i = b$. Define.

$A_1 = \{2j-1 \mid 1 \le j \le N, j \in A\}$
$A_2 = \{2j \mid 1 \le j \le N, j \in A\}$
$A_3 = \{2j-1 \mid 1 \le j \le N, j \notin A\}$
$A_4 = \{2j \mid 1 \le j \le N, j \notin A\}$

and let the manufacturing and remanufacturing quantities be, respectively, given by

$$Q_i^S = \begin{cases} \beta + b_{(i+1)/2}+1 & \text{for } i \in A_1 \\ \beta & \text{for } i \in A_3 \\ 0 & \text{Otherwise} \end{cases}$$

$$Q_i^R = \begin{cases} b_{i/2}+1 & \text{for } i \in A_4 \\ \beta+b+k & \text{for } i = 2N+1 \\ 0 & \text{Otherwise} \end{cases}$$

In addition, let $I_0^R = 0$. The lot-sizing policy stated above yields the following ending inventories for serviceables and cores, respectively,

$$I_i^S = \begin{cases} b_{(i+1)/2}+1 & \text{for } i \in A_1 \\ 0 & \text{for } i \in A_3 \\ 0 & \text{Otherwise} \end{cases}$$

$$I_i^R = \begin{cases} I_{i-1}^R + b_{(i+1)/2} + 1 & \text{for } i \in A_1 \cup A_3 \\ I_{i-1}^R - b_{i/2} - 1 & \text{for } i \in A_4 \\ I_{i-1}^R & \text{for } i \in A_2 \\ 0 & \text{for } i = 2N+1 \end{cases}$$

This policy requires N manufacturing set-ups yielding a total set-up cost of $NK^S = 2Nkb$. Furthermore, the associated total cost of holding serviceable inventory is equal to $h^S \sum_{i=1}^{2N+1} I_i^S = h^S \sum_{i \in A_1} (b_{(i+1)/2}+1) = h^S(b+k) = 1$ by our supposition. Therefore, the total cost is then equal to $2Nkb+1 = \alpha$.

(i) *The only if part*: Now suppose that there exists a solution to problem (P) with total cost less than or equal to α. Consider for a moment, then, from all good feasible solutions, what is an optimal solution. Then, in this case, the following statements can be made.

(i) This highly favorable solution must include a manufacturing or remanufacturing set-up in each odd-numbered period i, $i = 1,3,5,\ldots,(2N+1)$. Otherwise, the demand of period i, $i = 1,3,5,\ldots,(2N+1)$, must be met by carrying at least β units of serviceable inventory from an earlier period at a cost of at least $\beta h^S \ge \alpha + 1$.
(ii) It follows after (i) that the first N set-ups in periods $i = 1,3,5,\ldots,(2N-1)$ must involve a manufacturing set-up, because the cumulated returns up to each one of these periods are not sufficient to meet the demand requirement by remanufacturing only (Fig. 2).
(iii) Hence, the total set-up cost is at least $NK^S = 2Nkb$.
(iv) It then follows from (iii) and our supposition that the total cost of holding serviceable inventory is less than $\alpha - NK^S = 1$.
(v) Another consequence of (iii) is that a set-up in period $(2N+1)$ must involve a remanufacturing set-up only, or else the total cost would exceed α.
(vi) Now we claim that $I_{2N}^S = 0$. Otherwise, if it is not so, but rather $I_{2N}^S > 0$, we can find a cheaper solution by decreasing the quantity of production by at least 1 unit from Q_{2N}^R if $Q_{2N}^R > 0$ or Q_{2N-1}^S if $Q_{2N}^R = 0$. The resulting solution requires less holding cost with no increase in the set-up costs. This cheaper solution would contradict the optimality assumption. Hence, $I_{2N}^S = 0$.
(vii) From (v) and (vi), it follows that $I_{2N}^R \ge D_{(2N+1)} - R_{(2N+1)} = b+k$.
(viii) Now, we shall show that the inequality shown in (vii) actually holds as an equality. If (vii) does not hold as an equality, i.e. $I_{2N}^R > b+k$, then $I_{2N}^R = \sum_{i=1}^{2N} R_i - \sum_{i=1}^{2N} Q_i^R > b+k \Rightarrow 2b+N-\sum_{i=1}^{2N} Q_i^R > b+k \Rightarrow \sum_{i=1}^{2N} Q_i^R < b+N-k$. Since $\sum_{i=1,3,5,\ldots,(2N-1)} I_i^S \ge \sum_{i=2,4,6,\ldots,2N} D_i - \sum_{i=2,4,6,\ldots,2N} Q_i^R$, we also have $\sum_{i=1,3,5,\ldots,(2N-1)} I_i^S \ge 2b+N-\sum_{i=1}^{2N} Q_i^R > b+k$. Thus, the total cost of holding serviceables is $h^S \sum_{i=1}^{2N+1} I_i^S > h^S(b+k) = 1$, which contradicts the statement made in (iv).
(ix) Hence, $I_{2N}^R = b+k$ and there must exist a subset $A \subseteq \{1,2,3,\ldots,N\}$ with $|A| = k$ such that $\sum_{i \in A} b_i = b$.

This completes the proof. ■

Proof of Theorem 2. ■

Let x_{ij}^S and x_{ij}^R, for $i = 0,1,\ldots,N$ and $j = 0,1,\ldots,N$, be a feasible solution to $\overline{P}'$. We define y_i^S and y_i^R for $i = 0,1,\ldots,N$ as follows: $y_i^S = \sum_{j=i}^{N} x_{ij}^S$ and $y_i^R = \sum_{j=i}^{N} x_{ij}^R$. First, we note that $y_N^S = x_{NN}^S \le 1$ and that $y_0^S = \sum_{j=0}^{N} x_{0j}^S = 1$ given constraint (11) in P'. If we add the corresponding sides of constraint (12) in P' for $j = 1,2,\ldots,(i-1)$ and then cancel out the common terms, we get that $y_i^S \le \sum_{j=1}^{i-1} x_{1j}^S$ and so $0 \le y_i^S \le 1$ for $i = 2,\ldots,N-1$.

Note that constraints (2) and (3) in $P$are similar to constraints (7) and (8) in P'. To see if constraint (4) holds, consider 2 cases. In the first case, let $\sum_{j=i}^{N} x_{ij}^S = 0$. Clearly, $x_{ij}^S = 0$ for $j = i, \ldots, N$ and $y_i^S = 0$. Constraint (9) in P' implies $Q_i^S \le \sum_{j=i}^{N} d_{ij} x_{ij}^S = 0$ and thus, $Q_i^S = 0$. In the second case, let $0 < \sum_{j=i}^{N} x_{ij}^S \le 1$. Since $Q_i^S \le \sum_{j=i}^{N} d_{ij} x_{ij}^S$, $d_{ij} = \sum_{k=i}^{j} D_k$, and $\sum_{j=k}^{N} x_{ij}^S \le y_i^S$ for $k \ge i$, we have $Q_i^S \le \sum_{j=i}^{N} \left(\sum_{k=i}^{j} D_k\right) x_{ij}^S = \sum_{k=i}^{N} D_k \left(\sum_{j=k}^{N} x_{ij}^S\right) \le \left(\sum_{j=i}^{N} D_j\right) y_i^S$.

Thus, constraint (4) holds in both cases. Using a similar argument it can be shown that constraint (5) holds too. Thus, every constraint of $\overline{P}$ is satisfied. It follows, then, that $\overline{P}'$ gives better lower bounds on solutions to the original integer program. ■

Proof of Theorem 3. If $\gamma_i = 0$, statement (1) of the theorem holds due to non-negativity. Suppose $\gamma_i > 0$. In this case, $\gamma_i = I^R_{i-1} + R_i - D_i$. By adding Eqs. (1) and (2) and canceling out the common terms, $I^S_i + I^R_i = I^S_{i-1} + I^R_{i-1} + R_i - D_i + Q^S_i = I^S_{i-1} + \gamma_i + Q^S_i \geq \gamma_i$. Hence statement (1) holds. The solution mentioned in statement (2) follows from the lot for lot policy with priority given to remanufacturing, i.e. $Q^R_i = \min(I^R_{i-1} + R_i, D_i)$ and $Q^S_i = D_i - Q^R_i$. From Eq. (1), $I^S_i = I^S_{i-1} \forall i$. Hence, $I^S_i = I^S_0 \forall i$. From Eq. (2), $I^R_i = I^R_{i-1} + R_i - \min\left(I^R_{i-1} + R_i, D_i\right) = \gamma_i$. ■

References

[1] Aggarwal A, Park J. Improved algorithms for economic lot-size problems. Oper Res 1993;41:549–71.
[2] Atasu A, Cetinkaya S. Lot sizing for optimal collection and use of remanufacturable returns over a finite life cycle. Prod Oper Manage 2006;15:473–87.
[3] Baki MF, Chaouch AB, Abdul-Kader W. On the NP-completeness of the dynamic lot sizing problem with product returns and remanufacturing. Working Paper, University of Windsor; 2010.
[4] Beltran JL, Krass D. Dynamic lots sizing with returning items and disposals. IIE Trans 2002;34:437–48.
[5] Blackburn JD, Guide Jr VDR, Souza GC, Van Wassenhove LN. Reverse supply chains for commercial returns. Calif Manage Rev 2004;46(2):6–22.
[6] Chaouch ABA. Replenishment control system with uncertain returns and random opportunities for disposal. Int J Inventory Res 2011;3/4:221–47.
[7] Federgruen A, Tzur M. A simple forward algorithm to solve general dynamic lot sizing models with n periods in $O(n\log n)$ or $O(n)$ time. Manage Sci 1991;37:909–25.
[8] Fleischmann M, Bloemhof-Runwaard JM, Dekker R, van der Laan E, van Nunen JAEE, van Waasenhove LN. Quantitative models for reverse logistics: a review. Eur J Oper Res 1997;103:1–17.
[9] Fleischmann M, Beullens P, Bloemhof-Ruwaard M, Van Wassenhove LN. The impact of product recovery on logistics network design. Prod Oper Manage 2001;10(2):156–73.
[10] Garey MR, Johnson DS. Computers and intractability. New York: Freeman; 1998.
[11] Golany B, Yang J, Yu G. Economic lot-sizing with remanufacturing options. IIE Trans 2001;33:995–1003.
[12] Guide Jr VDR, Jayaraman V, Srivastava R, Benton WC. Supply-chain management for recoverable manufacturing systems. Interfaces 2000;30:125–42.
[13] Guide Jr VDR, Jayaraman V, Srivastava R. Production planning and control for remanufacturing: a state-of-the-art survey. Robotics Comput Integrated Manuf 1999;15(3):221–30.
[14] Guide Jr VDR, Van Wassenhove LN. Managing product returns for remanufacturing. Prod Oper Manage 2001;10(2):142–55.
[15] Guide Jr VDR, Van Wassenhove LN. The evolution of closed-loop supply chain research. Oper Res 2009;57(1):10–8.
[16] Helmrich MJR, Jans R, van den Heuvel W. Wagelmans APM. Economic lot-sizing with remanufacturing: complexity and efficient formulations. Economic Institute Report EI 2010-71. Rotterdam, The Netherlands: Erasmus University; 2010.
[17] Li X, Baki MF, Peng T, Chaouch AB. Robust block-chain tabu-search for dynamic lot sizing with product returns and remanufacturing. Omega 2014;42:75–87.
[18] Mahadevan B, Pyke DF, Fleischmann M. Periodic review, push inventory policies for remanufacturing. Eur J Oper Res 2003;151(3):536–51.
[19] Pineyro P, Viera O. Inventory policies for the economic lot-sizing problem with remanufacturing and final disposal options. J Ind Manage Optim 2009;5:217–38.
[20] Pineyro P, Viera O. The economic lot-sizing problem with remanufacturing and one-way substitution. Int J Prod Econ 2010;124:482–8.
[21] Richter K, Sombrutzki M. Remanufacturing planning for the reverse Wagner/Whitin models. Eur J Oper Res 2000;121:304–15.
[22] Richter K, Weber J. The reverse Wagner/Whitin model with variable manufacturing and remanufacturing cost. Int J Prod Econ 2001;71:447–56.
[23] Schulz TA. New Silver–Meal based heuristic for the single-item dynamic lot sizing problem with returns and remanufacturing. Int J Prod Res 2011;49(9):2519–33.
[24] Silver EA, Pyke DF, Peterson R. Inventory management and production planning and scheduling. New York: Wiley; 1998.
[25] Simpson VP. Optimal solution structure for a repairable inventory problem. Oper Res 1978;26(2):270–81.
[26] Srivastava SK. Green supply-chain management: a state-of-the-art literature review. Int J Manage Rev 2007;9(1):53–80.
[27] Tang O, Teunter RH. Economic lot scheduling problem with returns. Prod Oper Manag 2006;15:488–97.
[28] Teunter RH. A reverse logistics valuation method for inventory control. Int J Prod Res 2001;39(9):2023–35.
[29] Teunter RH, Bayindir ZP, van der Heuvel W. Dynamic lot sizing with product returns and remanufacturing. Int J Prod Res 2006;44:4377–400.
[30] Van der Laan E, Salomon M, Dekker R, Van Wassenhove LN. Inventory control in hybrid systems with remanufacturing. Manage Sci 1999;45(5):733–47.
[31] Wagelmans A, Van Hoesel SV, Kolen A. Economic lot sizing: an $O(n \log n)$ algorithm that runs in linear time in the Wagner–Whitin case. Oper Res 1992;40(1):S145–56.
[32] Yang J, Golany B, Yu G. A concave-cost production planning with remanufacturing options. Nav Res Logistics 2005;52:443–58.