



HAL
open science

New valid inequalities for a multi-echelon multi-item lot-sizing problem with returns and lost sales

Franco Quezada, Céline Gicquel, Safia Kedad-Sidhoum

► **To cite this version:**

Franco Quezada, Céline Gicquel, Safia Kedad-Sidhoum. New valid inequalities for a multi-echelon multi-item lot-sizing problem with returns and lost sales. ICCL 2021: International Conference on Computational Logistics, Sep 2021, Enschede (online), Netherlands. pp.192-207, 10.1007/978-3-030-87672-2_13 . hal-03351045

HAL Id: hal-03351045

<https://hal.science/hal-03351045v1>

Submitted on 21 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

New valid inequalities for a multi-echelon multi-item lot-sizing problem with returns and lost sales ^{*}

Franco Quezada^{1,2}, Céline Gicquel³, and Safia Kedad-Sidhoum⁴

¹ Sorbonne Université, LIP6, 75005 Paris, France

² Universidad de Santiago de Chile, LDSPS, Santiago, Chile
`franco.quezada@usach.cl`

³ Université Paris Saclay, LISN, 91190 Gif-sur-Yvette, France
`celine.gicquel@lri.fr`

⁴ CNAM, CEDRIC, 75003 Paris, France
`safia.kedad_sidhoum@cnam.fr`

Abstract. This work studies a multi-echelon multi-item lot-sizing problem with remanufacturing and lost sales. The problem is formulated as a mixed-integer linear program. A new family of valid inequalities taking advantage of the problem structure is introduced and used in a customized branch-and-cut algorithm. The provided numerical results show that the proposed algorithm outperforms both the generic branch-and-cut algorithm embedded in a standard-alone mathematical solver and a previously published customized branch-and-cut algorithm.

Keywords: Production planning · Lot-sizing · Remanufacturing · Mixed-integer linear programming · Valid inequalities.

1 Introduction

Industrial companies face an increasing pressure from customers and governments to become more environmentally responsible and mitigate the environmental impact of their products. One way of achieving this objective is to remanufacture the products once they have reached their end of life. Remanufacturing is defined as a set of processes transforming used products into like-new products, mainly by rehabilitating damaged components. By reusing the materials and components embedded in used products, remanufacturing both contributes in reducing pollution emissions and natural resource consumption, making production processes more environment-friendly.

The present work considers a remanufacturing system involving three production echelons: disassembly of used products brought back by customers, refurbishing of the recovered parts and reassembly into like-new finished products. We

^{*} This work was partially funded by the National Agency for Research and Development (ANID) / Scholarship Program / DOCTORADO BECAS CHILE/2018 - 72190160.

aim at optimizing the production planning for the corresponding three-echelon system over a multi-period horizon. Within a remanufacturing context, production planning includes making decisions on the used products returned by customers, such as how much and when used products should be disassembled, refurbished or reassembled in order to build new or like-new products. The main objective is to meet customers' demand for the remanufactured products in the most cost-effective way.

We consider the case where the production on a machine requires setup operations such as machine calibration and incurs fixed setup costs. As a naive perception, to reduce these setup costs, production should be run using large lot sizes. However, this generates desynchronized patterns between the customers' demand and the production plan leading to costly high levels of inventory. Lot-sizing models thus aim at reaching the best possible trade-off between minimizing the setup costs and minimizing the inventory holding costs, taking into account both the customers' demand satisfaction and the practical limitations of the system. In the present work, we investigate the problem of reaching the best possible trade-off between setup and inventory holding costs within a remanufacturing environment and introduce an additional lost sales cost to be paid when the customers' demand is not satisfied on time. We thus study a 3-echelon lot-sizing problem with returns and lost sales.

Only a few works have addressed such multi-echelon production systems through exact solution approaches. A first attempt at tackling this difficulty can be found in [12]. Quezada et al. [12] considered the problem in a stochastic setting, taking into account uncertainties on the problem input parameters. They proposed a multi-stage stochastic approach based on the use of scenario trees. The problem was formulated as a MILP and solved through a new customized branch-and-cut algorithm. This algorithm relied on valid inequalities focused on strengthening the formulation of the single-echelon uncapacitated lot-sizing sub-problems embedded in the main problem. Although this approach was successful at providing near optimal solutions for small to medium size instances, some numerical difficulties were encountered to solve the larger instances. Intuitively, this difficulty might be partly due to the fact that the valid inequalities used to strengthen the formulation considered uncapacitated single-echelon sub-problems. They did not take into account the fact that, even if the production resources are assumed uncapacitated, the amount of products that can be processed on a resource at a given time period is limited among others by the amount of available used products returned up to this time period and by the yield of the disassembly process, i.e. by the proportion of disassembled parts that are in a sufficiently good state to be refurbished and reused in a remanufactured product. Hence, using valid inequalities taking into account this aspect of the problem might contribute in further strengthening its MILP formulation and decrease the computational effort needed to solve large-size instances.

To the best of our knowledge, the formulation of valid inequalities that explicitly take into account the impact of a limited returns quantity on the production plan has not yet been studied for a multi-echelon remanufacturing system.

The present work aims at partially closing this gap by proposing new valid inequalities for this problem. However, in view of the theoretical and numerical difficulties encountered when developing new valid inequalities, we focus on the deterministic variant of the problem. Our contributions are thus twofold. First, we propose a new family of valid inequalities for the problem under study. These valid inequalities can be seen as an extension of valid inequalities previously known for the uncapacitated single-echelon lot-sizing problem with lost sales (the (k, U) inequalities first introduced in [8]) to take into account, at each echelon of the studied multi-echelon production system, the constraints on the production plan coming from the limited availability of the returns. We prove that these new valid inequalities are at least as strong as the previously known inequalities. Second, we develop a branch-and-cut algorithm based on the newly proposed valid inequalities and seek to assess its computational performance by comparing it with the one of a stand-alone mathematical programming solver and the one of a branch-and-cut algorithm based on previously known valid inequalities. The numerical results show the usefulness of the proposed inequalities at solving the problem under study.

The remainder of this paper is organized as follows. We first provide a brief overview of the related literature in Section 2. The problem description, together with its MILP formulation, are provided in Section 3. We then present the proposed new family of valid inequalities in Section 4. Computational results are summarized in Section 5. Finally, Section 6 gives a conclusion and some research perspectives.

2 Related works

Throughout the last decade, several works sought to strengthen the MILP formulation of single-echelon lot-sizing problems involving remanufacturing, either through extended reformulations or through valid inequalities.

Helmrich et al. [13] discussed several MILP formulations of the uncapacitated single-item single-echelon lot-sizing problem with remanufacturing and introduced new valid inequalities by adapting the previously known (l, S, WW) proposed by [10] to their problem. The inequalities developed in [13] are based on the assumption that all returned products are either processed or kept in stock and do not consider the possibility that some of the returns may be discarded in case of an unbalance between returns quantity and demand. They can therefore not be directly used for the problem under study here. Similarly, [5] proposed a multi-commodity reformulation and a new set of valid inequalities for this problem. In particular, they further strengthened the (l, S, WW) inequalities presented in [13] by considering that the amount of finished products remanufactured in a given period t is limited by the cumulative quantity of returned products brought back up to t . Ali et al. [2] enriched the previous works by highlighting a theoretical property with regards to the equivalence of the shortest path and facility location reformulations. They also carried out a polyhedral analysis of a related sub-problem based on the single node fixed-charge network

problem, proving the validity of several flow cover inequalities and their facet-defining conditions as well. Akartunali and Arulsevan [1] studied both the uncapacitated and capacitated variants of this single-item single-echelon lot-sizing problem. They showed that the uncapacitated problem cannot have a fully polynomial time approximation scheme (FPTAS) and provided a pseudo-polynomial algorithm to solve the problem. They also provided valid inequalities based on the flow-cover inequalities for the problem with a limited production capacity. Finally, we refer the reader to [4] for a recent survey on single-item lot-sizing problems with remanufacturing.

We note that all the above mentioned works focus on single-item single-echelon problems and do not consider the fact that remanufacturing may involve several processing steps, i.e. several production echelons, in order to transform the returned used products into like-new products. Moreover, these works consider hybrid manufacturing/remanufacturing systems and assume that, thanks to the presence of an uncapacitated manufacturing system, it will always be possible to satisfy the demand for finished products on time. In contrast, we investigate a pure remanufacturing system and consider that the demand will be lost in case the quantity and/or the quality of the returned products are insufficient to meet it on time. This means among others that the inequalities introduced in [13], [5] and [2] are not valid for our problem and that new valid inequalities taking into account its specific features are needed.

3 Problem description and modeling

3.1 Production system

We consider a remanufacturing system comprising three main production echelons: disassembly, refurbishing and reassembly. We seek to plan the production activities in this system over a horizon comprising a discrete set $\mathcal{T} = \{1, \dots, T\}$ of periods. The system involves a set \mathcal{I} of items. Among these ones, item $i = 0$ represents the used products returned by customers in limited quantities at each period. A used product is composed of I parts. Let α_i be the number of parts i embedded in a used product. The returned products are first disassembled to obtain a set $\mathcal{I}_r = \{1, \dots, I\}$ of recoverable parts. Due to the usage state of the used products, some of the parts obtained during disassembly have to be discarded. In order to reflect the variations in the quality of the used products, i.e. the yield of the disassembly process, we let π_i^t denote the proportion of parts which will be recoverable at each time period t for each item $i = \{1, \dots, I\}$. The recoverable parts are then refurbished on dedicated refurbishing processes. The set of $\mathcal{I}_s = \{I + 1, \dots, 2I\}$ of serviceable parts obtained after refurbishing are reassembled into remanufactured products which have the same bill-of-material as the used products. These remanufactured products, indexed by $i = 2I + 1$, are used to satisfy the dynamic demand of customers.

The system comprises a set $\mathcal{P} = \{0, \dots, I + 1\}$ of production processes: $p = 0$ corresponds to the disassembly process, $p \in \{1, \dots, I\}$ corresponds to the process refurbishing the recoverable part indexed by p into the serviceable part indexed

by $p+I$ and $p = I+1$ corresponds to the reassembly process. All these processes are assumed to be uncapacitated. However, the system might not be able to satisfy the customer demand on time due to part shortages if there are not enough used products returned by customers or if their quality is low. In this situation, the corresponding demand is lost incurring a high penalty cost to account for the loss of customer goodwill. Moreover, some used products and recoverable parts are allowed to be discarded. This option might be useful in case more used products are returned than what is needed to satisfy the demand for remanufactured products and in case there is a strong unbalance between the part-dependent disassembly yields leading to an unnecessary accumulation in inventory of the easy-to-recover parts.

All input parameters of the problem are time-dependent: r^t denotes the quantity of collected used products, d^t the customers' demand and π_i^t the proportion of recoverable parts $i \in \mathcal{I}_r$ obtained by disassembling one unit of returned product at period t . As for the costs, for each period t , we have the setup cost f_p^t for process $p \in \mathcal{P}$, the unit inventory cost h_i^t for part $i \in \mathcal{I}$, the unit lost-sales penalty cost l^t , the unit cost q_i^t for discarding item $i \in \mathcal{I}_r \cup \{0\}$ and the unit cost g^t for discarding the unrecoverable parts obtained while disassembling one unit of returned product.

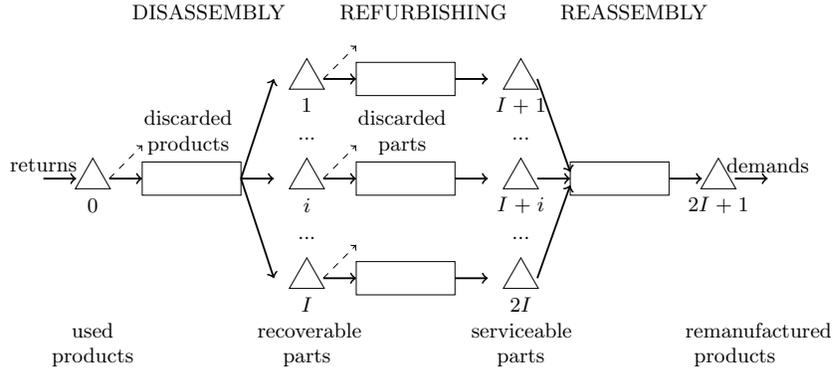


Fig. 1: Studied remanufacturing system

3.2 Natural formulation

In order to build a mathematical model for the problem, we introduce the following decision variables at time period $t \in \mathcal{T}$: X_p^t the quantity of parts processed on process $p \in \mathcal{P}$, $Y_p^t \in \{0, 1\}$ the setup variable for process $p \in \mathcal{P}$, S_i^t the inventory level of part $i \in \mathcal{I}$, Q_i^t the quantity of part $i \in \mathcal{I}_r \cup \{0\}$ discarded and L^t the lost sales of remanufactured products. This leads to the following MILP model.

$$\min \sum_{t \in \mathcal{T}} \left(\sum_{p \in \mathcal{J}} f_p^t Y_p^t + \sum_{i \in \mathcal{I}} h_i^t S_i^t + l^t L^t + \sum_{i \in \mathcal{I}_r \cup \{0\}} q_i^t Q_i^t + g^t X_0^t \right) \quad (1)$$

$$X_p^t \leq M_p^t Y_p^t \quad \forall p \in \mathcal{J}, \forall t \in \mathcal{T} \quad (2)$$

$$S_0^t = S_0^{t-1} + r^t - X_0^t - Q_0^t \quad \forall t \in \mathcal{T} \quad (3)$$

$$S_i^t = S_i^{t-1} + \pi_i^t \alpha_i X_0^t - X_i^t - Q_i^t \quad \forall i \in \mathcal{I}_r, \forall t \in \mathcal{T} \quad (4)$$

$$S_i^t = S_i^{t-1} + X_{i-I}^t - \alpha_i X_{I+1}^t \quad \forall i \in \mathcal{I}_s, \forall t \in \mathcal{T} \quad (5)$$

$$S_{2I+1}^t = S_{2I+1}^{t-1} + X_{I+1}^t - d^t + L^t \quad \forall t \in \mathcal{T} \quad (6)$$

$$S_i^0 = 0 \quad \forall i \in \mathcal{I} \quad (7)$$

$$S_i^t \geq 0 \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \quad (8)$$

$$X_p^t \geq 0, Y_p^t \in \{0, 1\} \quad \forall p \in \mathcal{J}, \forall t \in \mathcal{T} \quad (9)$$

The objective function (1) aims at minimizing the total remanufacturing cost over the whole planning horizon, i.e., the sum of the setup, inventory holding, lost sales and disposal costs. Constraints (2) link the production quantity variables to the setup variables. Constraints (3)-(6) are the inventory balance constraints. More specifically, Constraints (3) ensure that any returned product is either disassembled, discarded, or kept in stock. Constraints (4) guarantee that any item obtained from the disassembly process is either refurbished, discarded, or kept in stock. Constraints (5) ensure that any refurbished item is either used in the reassembly process or kept in stock. Constraints (6) ensure that any remanufactured/finished product is either used to satisfy the demand or kept in stock and that, if there is not enough remanufactured products to satisfy the demand, the unsatisfied demand is lost. Without loss of generality, we assume that the initial inventory, S_i^0 , is set to 0 for each item $i \in \mathcal{I}$ (see Constraints (7)). Finally, Constraints (8)-(9) provide the domain of the decision variables.

Note that the value of each constant M_p^t can be set by using an upper bound on the quantity that can be processed on process p at each time period t . This quantity is limited by two elements: the availability of the used products already returned by customers and the future demand for remanufactured products. We thus have, for each period t :

$$\begin{aligned} - M_0^t &= \min \left\{ \sum_{1 \leq \kappa \leq t} r^\kappa, \frac{\sum_{t \leq \kappa \leq T} d^\kappa}{\min_{i=1, \dots, I} \pi_i^t} \right\} \\ - M_p^t &= \alpha_p \min \left\{ \sum_{1 \leq \kappa \leq t} r^\kappa \hat{\pi}_p^{\kappa, t}, \sum_{1 \leq \kappa \leq T} d^\kappa \right\}, \text{ for } p \in \mathcal{I}_r. \\ - M_{I+1}^t &= \min \left\{ \min_{p \in \mathcal{I}_r} \left\{ \sum_{1 \leq \kappa \leq t} r^\kappa \hat{\pi}_p^{\kappa, t} \right\}, \sum_{t \leq \kappa \leq T} d^\kappa \right\} \end{aligned}$$

where $\hat{\pi}_p^{\kappa, t} = \operatorname{argmax}\{\pi_p^\theta, \theta = \kappa, \dots, t\}$ denotes the maximum disassembly yield that can be obtained for recoverable item p over the time interval $[\kappa, t]$.

3.3 Echelon stock reformulation

We now provide a reformulation of the problem using the echelon stock concept [11]. The echelon stock of a product in a multi-echelon production system corresponds to the total quantity of the product held in inventory, either as such

or as a component within its successors in the bill-of-material. We thus denote by E_i^t the echelon stock level of item $i \in \mathcal{I} \setminus \{0\}$ at the end of period t . Replacing variables S_i^t by variables E_i^t in Problem (1)-(9) leads to the following MILP formulation:

$$\min \sum_{t \in \mathcal{T}} \left(\sum_{p \in \mathcal{J}} f_p^t Y_p^t + h_0^t S_0^t + \sum_{i \in \mathcal{I} \setminus \{0\}} e h_i^t E_i^t + \sum_{i \in \mathcal{I}_r \cup \{0\}} q_i^t Q_i^t + g_0^t X_0^t \right) \quad (10)$$

$$X_p^t \leq M_p^t Y_p^t \quad \forall p \in \mathcal{J}, \forall t \in \mathcal{T} \quad (11)$$

$$S_0^t = S_0^{t-1} + r^t - X_0^t - Q_0^t \quad \forall t \in \mathcal{T} \quad (12)$$

$$E_i^t = E_i^{t-1} + \pi_i^t \alpha_i X_0^t - \alpha_i (d^t - L^t) - Q_i^t \quad \forall i \in \mathcal{I}_r, \forall t \in \mathcal{T} \quad (13)$$

$$E_i^t = E_i^{t-1} + X_{i-I}^t - \alpha_i (d^t - L^t) \quad \forall i \in \mathcal{I}_s, \forall t \in \mathcal{T} \quad (14)$$

$$E_{2I+1}^t = E_{2I+1}^{t-1} + X_{I+1}^t - d^t + L^t \quad \forall t \in \mathcal{T} \quad (15)$$

$$S_0^0 = 0 \quad (16)$$

$$E_i^0 = 0 \quad \forall i \in \mathcal{I} \setminus \{0\} \quad (17)$$

$$E_i^t - E_{I+i}^t \geq 0 \quad \forall i \in \mathcal{I}_r, \forall n \in \mathcal{T} \quad (18)$$

$$E_i^t - \alpha_i E_{2I+1}^t \geq 0 \quad \forall i \in \mathcal{I}_s, \forall n \in \mathcal{T} \quad (19)$$

$$E_i^t \geq 0 \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T} \quad (20)$$

$$S_0^t, L^t \geq 0 \quad \forall t \in \mathcal{T} \quad (21)$$

$$X_p^t \geq 0, Y_p^t \in \{0, 1\} \quad \forall p \in \mathcal{J}, \forall t \in \mathcal{T} \quad (22)$$

The objective function (10) aims at minimizing the total cost over the whole planning horizon. Constraints (11) link the production quantity variables to the setup variables. Constraints (12)-(15) are the inventory balance constraints. Constraints (12) use the classical inventory variables, whereas Constraints (13)-(15) make use of the echelon inventory variables. Constraints (16)-(17) translate the fact that the initial inventory of each item is assumed to be equal to 0. Constraints (18)-(19) ensure consistency between the echelon inventory at the different echelons of the bill-of-material and guarantee that the physical inventory of each product, S_i^t , remains non-negative for all $i \in \mathcal{I}$. Finally, Constraints (20)-(22) define the domain of the decision variables.

The use of the echelon stock reformulation (11)-(22) enables us to decompose the initial problem into a series of single-echelon sub-problems by relaxing the linking constraints (18)-(19). Each of these sub-problems is an uncapacitated single-echelon single-item lot-sizing problem with lost sales, whose formulation can be strengthened by the (k, U) valid inequalities proposed in [8]. We refer the reader to [12] for a detailed description of each subproblem and the single-echelon (k, U) inequalities applied to each subproblem. Nonetheless, this decomposition into single-echelon uncapacitated sub-problems overlooks the fact that the production on each process at a given period is limited by the amount of used

products returned up to this period. Therefore, in what follows, we investigate a class of valid inequalities in which these aspects of the problem are explicitly considered.

4 Single-echelon (ℓ, k, U) inequalities

We now seek to strengthen the single-echelon (k, U) inequalities investigated in [8] and [12] by considering the limited quantity of returned products available at each time period in the system. The (ℓ, k, U) inequalities are defined as follows:

Proposition 1. *Let $0 \leq \ell \leq k \leq T$ be two periods of the planning horizon.*

Let $U \subseteq \{k+1, \dots, T\}$ be a subset of periods and $t^ = \max\{\tau : \tau \in U\}$ be the last time period belonging to U .*

The following inequalities are valid for Problem (11)-(22):

$$S_0^\ell \hat{\pi}_i^{\ell, t^*} + \alpha_i^{-1} E_i^k + \sum_{k < t \leq t^*} \phi_i^t Y_0^t \geq \sum_{t \in U} (d^t - L^t) \quad \forall i \in \mathcal{I}_r \quad (23)$$

$$S_0^\ell \hat{\pi}_i^{\ell, t^*} + \alpha_i^{-1} (E_i^\ell - E_{i+1}^\ell) + \alpha_i^{-1} E_{i+1}^k + \sum_{k < t \leq t^*} \phi_i^t Y_i^t \geq \sum_{t \in U} (d^t - L^t) \quad \forall i \in \mathcal{I}_r \quad (24)$$

$$S_0^\ell \hat{\pi}_i^{\ell, t^*} + (\alpha_i^{-1} E_i^\ell - E_{2I+1}^\ell) + E_{2I+1}^k + \sum_{k < t \leq t^*} \phi_i^t Y_{I+1}^t \geq \sum_{t \in U} (d^t - L^t) \quad \forall i \in \mathcal{I}_r \quad (25)$$

$$\text{with } \phi_i^t = \min \left\{ \sum_{\ell < \nu \leq t} r^\nu \hat{\pi}_i^{\nu, t}, \sum_{\nu \in U: t \leq \nu} d^\nu \right\}$$

Proof. Let (X, Y, S, E, L, Q) be a feasible solution of Problem (10)-(22). We show that this solution complies with inequalities (23) for any pair of periods (ℓ, k) , any subset $U \subset \{k+1, \dots, T\}$ and any recoverable item $i \in \mathcal{I}_r$.

Let $\tau \in [k+1, t^*]$ be the first production period in which $\phi_i^\tau = \sum_{\nu \in U: \tau \leq \nu} d^\nu$. By

convention, $\tau = t^* + 1$ if there is no such period.

$$\text{We have } \phi_i^\tau Y_0^\tau = \sum_{t \in U: \tau \leq t} d^t \geq \sum_{t \in U: \tau \leq t} (d^t - L^t).$$

We consider two cases.

- Case 1: there is no production on $p = 0$ over the interval $[k+1; \tau-1]$

In this case, $Y_0^t = 0$ and $X_0^t = 0$ for all periods t in $[k+1, \tau-1]$. As no disassembly occurs over $[k+1, \tau-1]$, all the recoverable items needed to satisfy the demand over this time interval, and in particular needed to satisfy $\sum_{t \in U: t \leq \tau-1} (d^t - L^t)$,

should already have been disassembled previously and be in stock at the end of period k . This gives $\alpha_i^{-1} E_i^k \geq \sum_{t \in U: t \leq \tau-1} (d^t - L^t)$. We thus have:

$$S_0^\ell \hat{\pi}_i^{\ell, t^*} + \alpha_i^{-1} E_i^k + \sum_{k < t \leq t^*} \phi_i^t Y_0^t \geq \alpha_i^{-1} E_i^k + \phi_i^\tau Y_0^\tau \geq \sum_{t \in U} (d^t - L^t)$$

Inequality (23) is thus valid in this case.

- Case 2: there is at least one production period on $p = 0$ over interval $[k+1; \tau-1]$

Let θ be the last period of production on $p = 0$ over the interval $[k+1; \tau-1]$. By definition of θ , we have: $\phi_i^\theta = \sum_{\ell < \nu \leq \theta} (r^\nu \hat{\pi}_i^{\nu, \theta})$.

By summing up the inventory balance constraints (13) over periods $k+1, \dots, \tau-1$ and using the fact that variables E_i^k and $Q_i^t, \forall t = k+1, \dots, \tau-1$, are non-negative, we have:

$$\alpha_i^{-1} E_i^k + \sum_{t=k+1}^{\tau-1} \pi_i^t X_0^t \geq \sum_{t=k+1}^{\tau-1} (d_t - L_t) \quad (26)$$

By definition of τ , θ and ℓ , we have:

$$\sum_{t=k+1}^{\tau-1} \pi_i^t X_0^t = \sum_{t=k+1}^{\theta} \pi_i^t X_0^t \leq \sum_{t=\ell+1}^{\theta} \pi_i^t X_0^t \quad (27)$$

This gives:

$$\alpha_i^{-1} E_i^k + \sum_{t=\ell+1}^{\theta} \pi_i^t X_0^t \alpha_i^{-1} \geq E_i^k + \sum_{t=k+1}^{\tau-1} \pi_i^t X_0^t \quad (28)$$

$$\geq \sum_{t=k+1}^{\tau-1} (d_t - L_t) \quad (29)$$

$$\geq \sum_{t \in U: t \leq \tau-1} (d_t - L_t) \quad (30)$$

We now compute an upper bound of $\sum_{t=\ell+1}^{\theta} \pi_i^t X_0^t$. This one is obtained by first computing the linear combination $\sum_{t=\ell+1}^{\theta} (\hat{\pi}_i^{t, \theta}) \times (12)_t$. This gives:

$$\sum_{t=\ell+1}^{\theta} (\hat{\pi}_i^{t, \theta}) S_0^t = \sum_{t=\ell+1}^{\theta} (\hat{\pi}_i^{t, \theta}) [S_0^{t-1} + r^t - X_0^t - Q_0^t] \quad (31)$$

By the non-negativity of variables Q_0^t and S_0^t and the fact that $\hat{\pi}_i^{t, \theta} \geq \hat{\pi}_i^{t+1, \theta}$, we have:

$$\sum_{t=\ell+1}^{\theta} \pi_i^t X_0^t \leq \sum_{t=\ell}^{\theta} \hat{\pi}_i^{t, \theta} X_0^t \quad (32)$$

$$\leq \sum_{t=\ell+1}^{\theta} (\hat{\pi}_i^{t, \theta}) S_0^{t-1} - \sum_{t=\ell+1}^{\theta} (\hat{\pi}_i^{t, \theta}) S_0^t + \sum_{t=\ell+1}^{\theta} (\hat{\pi}_i^{t, \theta}) r^t \quad (33)$$

$$\leq \left(\hat{\pi}_i^{\ell, \theta}\right) S_0^\ell + \sum_{t=\ell+1}^{\theta} \left(\hat{\pi}_i^{t, \theta}\right) r^t \quad (34)$$

$$\leq \left(\hat{\pi}_i^{\ell, t^*}\right) S_0^\ell + \phi_i^\theta Y_0^\theta \quad (35)$$

Replacing $\sum_{t=\ell}^{\theta} \pi_i^t X_0^t$ in Inequalities (30) by its upper bound provided by (35), we have:

$$\alpha_i^{-1} E_i^k + \left(\hat{\pi}_i^{\ell, t^*}\right) S_0^\ell + \phi_i^\theta Y_0^\theta \geq \sum_{t \in U: t \leq \tau-1} (d_t - L_t) \quad (36)$$

Finally, we have:

$$\begin{aligned} S_0^\ell \hat{\pi}_i^{\ell, t^*} + \alpha_i^{-1} E_i^k + \sum_{k < t \leq t^*} \phi_i^t Y_0^t &\geq S_0^\ell \hat{\pi}_i^{\ell, t^*} + \alpha_i^{-1} E_i^k + \phi_i^\theta Y_0^\theta + \phi_i^\tau Y_0^\tau \\ &\geq \sum_{t \in U: t \leq \tau-1} (d_t - L_t) + \sum_{t \in U: t \geq \tau} (d_t - L_t) \\ &\geq \sum_{t \in U} (d_t - L_t) \end{aligned}$$

This concludes the proof of validity for Inequality (23). The same arguments can be used to prove the validity of Inequalities (24) and (25). \square

It is worth mentioning that the (k, U) inequalities used in [12] to strengthen the formulation (10)-(22) can be seen as a particular case of the more general family of (ℓ, k, U) inequalities (23)-(25) proposed in this work. Namely, by setting ℓ to 0 and by computing the value of ϕ^t without taking the returns into account (i.e. by setting ϕ^t to $\sum_{\nu \in U: t \leq \nu} d^\nu$), each (ℓ, k, U) inequality (23)-(25) becomes a (k, U) inequality.

Proposition 2. *The linear relaxation of formulation (10)-(22) strengthened by valid inequalities (23)-(25) is at least as tight as the linear relaxation strengthened by the (k, U) valid inequalities used in [12].*

Proof. Let P_{LR} be the linear relaxation of polyhedron given by inequalities (11)-(22), (23)-(25) and \tilde{P}_{LR} be the linear relaxation of polyhedron given by inequalities (11)-(22) and the (k, U) inequalities. As any (k, U) inequality is a valid inequality (23)-(25) with $\phi^t = \sum_{\nu \in U: t \leq \nu} d^\nu$ and $\ell = 0$, we have $P_{LR} \subseteq \tilde{P}_{LR}$. \square

The main implication of Proposition 2 is that the lower bound obtained by strengthening the formulation (10)-(22) with the (ℓ, k, U) inequalities is at least as tight as the lower bound obtained while using the single-echelon (k, U) inequalities.

We now briefly discuss the resolution of the separation problem for the (ℓ, k, U) valid inequalities. Recall that this problem consists in finding an inequality (23)-(25) violated by a given solution $(\tilde{X}, \tilde{Y}, \tilde{S}, \tilde{E}, \tilde{L}, \tilde{Q})$ of the linear

relaxation of Problem (11)-(22) or prove that no such inequality exists. In the present case, in order to find the most violated inequality among e.g. inequalities (23), we should find, for each period $k = 1, \dots, T$, the set U of time periods and the period ℓ that maximize the difference between the right-hand and the left-hand side of the inequality. This is not trivial, in particular because the value of each coefficient ϕ_i^t simultaneously depends on U and ℓ . We thus consider a heuristic separation algorithm in our computational experiments. This one can be summarized as follows. For a given process p and time period k :

1. For each period $t = k + 1, \dots, T$, add t to U if $d^t(1 - \sum_{\tau=k+1}^t \tilde{Y}_p^\tau) - \tilde{L}^t > 0$.
2. For each period $\ell = 0, \dots, k$,
 - compute the value of each coefficient $\phi_i^t = \min \left\{ \sum_{\ell < \nu \leq t} r^\nu \hat{\pi}_i^{\nu,t}, \sum_{\nu \in U: t \leq \nu} d^\nu \right\}$
 - compute the left-hand side of the inequality (23) (resp. (24) and (25)).
3. Set ℓ to the period index which minimizes this left-hand side value.

This algorithm has a time complexity of $\mathcal{O}(T^2)$ as the computation of set U in step 1 and of coefficients ϕ_i in step 2 both require $\mathcal{O}(T^2)$ operations.

5 Computational Experiments

In this section, we focus on assessing the performance of the proposed valid inequalities when used within a customized branch-and-cut algorithm. We compare the performance of this algorithm with the one of the generic branch-and-cut algorithm embedded in a mathematical programming solver and the one of a branch-and-cut algorithm using single-echelon (k, U) inequalities.

5.1 Instance Generation

We considered two sets of instances: Set 1 instances involve $T = 25$ periods and $I = 10$ parts whereas Set 2 instances involve $T = 35$ periods and $I = 10$ parts. Within each set, the instances were randomly generated by adapting the procedure presented in [6]. More precisely, we considered four values of the setup-holding cost ratio $f/h \in \{600, 1200, 1800, 2400\}$, two values for the production-holding cost ratio $g/h \in \{2, 4\}$ and three values of the returns-demand quantity ratio $r/d \in \{1, 2, 3\}$. For each set and each possible combination of f/h , g/h , r/d , ten random instances were generated, resulting in a total of 480 instances.

- For each instance, the value of each problem parameter was set as follows.
- Demand d^t was uniformly distributed in the interval $[0, 100]$ and the returns quantity r^t was uniformly distributed in the interval $[0.8(r/d)\bar{d}, 1.2(r/d)\bar{d}]$, where $\bar{d} = \frac{\sum d^t}{T}$ is the average demand per period.
 - The proportion of recoverable parts π_i^t was uniformly distributed in the interval $[0.4, 0.6]$.
 - The bill-of-materials coefficients $\alpha_i = \alpha_{i+I}, i \in \mathcal{I}_r$, were randomly generated following a discrete uniform distribution over $[1; 6]$ and we set $\alpha_0 = \alpha_{2I+1} = 1$.
 - The holding cost h_0^t for the returned product $i = 0$ was fixed to 1. The holding

cost h_i^t for each recoverable item $i \in \mathcal{I}_r$ was randomly generated following a discrete uniform distribution over interval $[2, 7]$. Similarly, the holding cost h_i^t for each serviceable item $i \in \mathcal{I}_s$ was randomly generated following a discrete uniform distribution over interval $[7, 12]$. Finally, in order to ensure non negative echelon costs, we set the value of the inventory holding cost for the remanufactured product, h_{2I+1}^t , to $\sum_{i=1}^I \alpha_i h_{I+i}^t + \epsilon$, where ϵ follows a discrete uniform distribution over interval $[80, 100]$.

- The production cost g^t was uniformly distributed in the interval $[0.8(g/h)\bar{h}, 1.2(g/h)\bar{h}]$, where $\bar{h} = \frac{\sum h_{2I+1}^t}{T}$ is the average holding cost.
- The setup cost f^t was uniformly distributed in the interval $[0.8(f/h)\bar{h}, 1.2(f/h)\bar{h}]$.
- Discarding costs were set to $q_i^t = 0.8\bar{h}_i^t$, where $\bar{h}_i^t = \frac{1}{T} \sum_{\kappa=t}^T h_i^\kappa$. The unit penalty cost for lost sales, l^n , was fixed to 10000 per

5.2 Results

We carried out extensive numerical experiments in order to assess the computational performance of the proposed valid inequalities. This was achieved by solving each considered instance using three alternatives branch-and-cut algorithms:

- *CPX*: the generic branch-and-cut algorithm embedded in CPLEX 12.8 using the echelon-stock formulation (11)-(22).
- (k, U) : a customized branch-and-cut algorithm using the (k, U) inequalities to strengthen the echelon-stock formulation (11)-(22) similarly to what was done in [12].
- (ℓ, k, U) : a customized branch-and-cut algorithm using the newly introduced (ℓ, k, U) inequalities to strengthen the echelon-stock formulation (11)-(22). This algorithm is based on the solver CPLEX 12.8. It generates inequalities of type (23)-(25) through a cutting-plane generation algorithm at the root node and at intermediate nodes of the branch-and-bound search tree using the UserConstraints callbacks provided by the solver.

All related linear programs and mixed-integer linear programs were solved using CPLEX 12.8 with the solver default settings. The algorithms were implemented in C++ using the Concert Technology environment. All tests were run on the computing infrastructure of the Laboratoire d'Informatique de Paris VI (LIP6), which consists of a cluster of Intel Xeon Processors X5690. We set the cluster to use two 3.46GHz cores and 12GB RAM to solve each instance. We imposed a time limit of 3600 seconds.

The corresponding results are displayed in Tables 1 for Set 1 instances and 2 for Set 2 instances. Column **Method** indicates the branch-and-cut algorithm used to solve the instances. Column **R.LP_{gap}** reports the gap between the value of the linear relaxation strengthened by the corresponding valid inequalities and the best feasible solution found through the branch-and-bound search. For the CPX method, it reports the gap between the value of the initial linear relaxation and the best feasible solution found through the branch-and-bound search. Column **R.MIP_{gap}** reports the gap between the lower bound at the root node (after the

Table 1: Performance of CPLEX and branch-and-cut methods over instance in Set 1.

r/d	g/h	Method	R.LP _{gap}	R.MIP _{gap}	MIP _{gap}	C.Time	R.Time	T.Time
1	2	<i>CPX</i>	8.26	4.10	0.06	0.04	1.16	898.00
		(k, U)	6.08	3.66	0.05	0.92	2.93	1,011.79
		(ℓ, k, U)	4.17	3.55	0.03	0.97	2.35	792.64
	4	<i>CPX</i>	5.38	2.49	0.06	0.03	0.97	1,360.48
		(k, U)	3.95	2.33	0.05	0.50	1.47	1,088.78
		(ℓ, k, U)	2.84	2.22	0.04	0.85	1.98	1,056.29
2	2	<i>CPX</i>	40.86	11.28	0.97	0.03	1.81	2,570.62
		(k, U)	19.36	9.17	0.46	0.35	2.74	1,883.71
		(ℓ, k, U)	11.88	8.68	0.22	0.74	2.67	1,558.09
	4	<i>CPX</i>	37.79	10.71	2.29	0.03	1.59	3,280.23
		(k, U)	18.39	9.25	1.06	0.19	1.47	2,856.09
		(ℓ, k, U)	11.89	8.48	0.94	0.62	2.27	2,883.50
3	2	<i>CPX</i>	41.86	13.47	0.04	0.03	1.38	545.15
		(k, U)	20.60	9.68	0.02	0.29	1.96	296.65
		(ℓ, k, U)	14.16	9.11	0.02	0.48	1.91	250.52
	4	<i>CPX</i>	38.29	12.44	0.09	0.02	0.98	817.93
		(k, U)	20.81	9.71	0.02	0.15	1.00	458.41
		(ℓ, k, U)	15.36	8.99	0.06	0.41	1.57	522.70

generation of CPLEX generic cutting planes) and the best feasible solution found through the branch-and-bound search. Column MIP_{gap} reports the gap between the best lower and the best feasible solution found through the branch-and-bound search. The average CPU time for the cutting-plane generation of each method is reported in column **C.Time**, the CPU time spent at the root node in Column **R.Time** and the average total CPU time in Column **T.Time**. Note that each line corresponds to the average value of the corresponding 40 instances.

In general, we observe that the customized branch-and-cut algorithms based either on the (k, U) or on the (ℓ, k, U) inequalities outperform method CPX, providing solutions of better quality within shorter computation times. Specifically, the total computation time is reduced on average by 20% when using the branch-and-cut algorithm based on the (k, U) and by 26% when using the branch-and-cut algorithm based on the (ℓ, k, U) inequalities.

Regarding the relative performance of the (k, U) and (ℓ, k, U) inequalities, we note that the branch-and-cut algorithm based on the (ℓ, k, U) outperforms the algorithm based on (k, U) when the value of the demand-returns ratio is small, ie. when $r/d \in \{1, 2\}$. Thus, over the 160 instances corresponding to a value of r/d equal to 1, the total average computation time is reduced from 1732s when using (k, U) inequalities to 1482s when using (ℓ, k, U) inequalities. Similarly, over the 160 instances corresponding to a value of r/d equal to 2, the average MIP gap is reduced from 2.72% when using (k, U) inequalities to 2.38% when using (ℓ, k, U) inequalities. We note however that the relative performance of the proposed (ℓ, k, U) inequalities deteriorates for the instances corresponding to the largest considered value of the demand-returns ratio. Namely, when r/d is set to 3,

Table 2: Performance of CPLEX and branch-and-cut methods over instance in Set 2.

r/d	g/h	Method	R.LP _{gap}	R.MIP _{gap}	MIP _{gap}	C.Time	R.Time	T.Time
1	2	<i>CPX</i>	7.48	3.65	0.25	0.06	2.26	2,544.42
		(k, U)	5.09	3.08	0.25	8.08	13.32	2,300.22
		(ℓ, k, U)	3.49	2.95	0.11	7.59	11.07	1,764.24
4	<i>CPX</i>		4.96	2.33	0.19	0.06	1.91	2,930.72
		(k, U)	3.31	2.00	0.11	3.66	6.58	2,529.28
		(ℓ, k, U)	2.40	1.91	0.10	5.09	7.95	2,319.77
2	2	<i>CPX</i>	44.73	12.89	6.76	0.06	4.43	3,599.07
		(k, U)	20.10	10.00	4.34	0.94	5.71	3,508.81
		(ℓ, k, U)	12.85	9.54	3.77	2.09	6.08	3,362.24
4	<i>CPX</i>		43.15	12.54	7.29	0.06	3.57	3,599.07
		(k, U)	20.02	10.40	5.03	0.53	3.08	3,599.54
		(ℓ, k, U)	13.14	9.49	4.61	2.42	6.01	3,599.44
3	2	<i>CPX</i>	43.25	14.11	2.67	0.04	2.93	3,044.18
		(k, U)	18.97	9.48	0.55	0.87	4.26	1,855.28
		(ℓ, k, U)	13.38	9.11	0.64	1.40	4.30	1,614.35
4	<i>CPX</i>		37.56	11.83	1.60	0.04	2.55	2,802.69
		(k, U)	16.82	8.30	0.19	0.49	2.30	1,035.84
		(ℓ, k, U)	12.67	7.86	0.20	1.21	3.67	1,029.84

the branch-and-cut algorithm based on the (k, U) inequalities provides a smaller MIP gap and/or smaller computation times. This might be explained by the fact that the corresponding instances involve a large amount of returned products so that the quantity processed on a resource at a given period is not (or at least to a lesser extent) limited by the availability of the returned products. This means that the proposed refinements in the expression of the valid inequalities are less relevant in this case.

6 Conclusion and perspectives

We considered a lot-sizing problem aiming at planning production for a multi-echelon remanufacturing system. This problem can be formulated as a mixed-integer linear program. We focused on strengthening this formulation in order to be able to provide optimal or near-optimal solutions of this problem. Our main contribution is the development of a new set of valid inequalities which take into account, at each production echelon, the limitations on the produced quantities coming from the limited availability of the returned products. The results of our computational experiments show that a branch-and-cut algorithm based on these new valid inequalities performs well as compared to the generic branch-and-cut algorithm of CPLEX solver and to a previously published branch-and-cut algorithm based on less general valid inequalities.

A first possible research direction could be to develop an exact separation algorithm for the (ℓ, k, U) inequalities in the presented branch-and-cut algorithm

as this may further improve their performance when used in a branch-and-cut algorithm. Moreover, it would also be worth studying whether valid inequalities previously proposed for capacitated lot-sizing problems (see e.g. [3], [7], [9]) might be useful to help solving our problem. Additional computational experiments are also needed to assess the size of the largest instances that may be solved with the proposed exact solution approach.

On a longer perspective, we could seek to extend the proposed valid inequalities to lot-sizing problems with returns involving complicating features such as a limited capacity, backloging, safety stocks or minimum production levels. Finally, extending the proposed valid inequalities and the cutting-plane generation to solve the stochastic version of the problem studied in [12] is also worth investigating.

References

1. Akartunalı, K., Arulsevan, A.: Economic lot-sizing problem with remanufacturing option: complexity and algorithms. In: International Workshop on Machine Learning, Optimization, and Big Data. pp. 132–143. Springer (2016)
2. Ali, S.A.S., Doostmohammadi, M., Akartunalı, K., van der Meer, R.: A theoretical and computational analysis of lot-sizing in remanufacturing with separate setups. *International Journal of Production Economics* **203**, 276–285 (2018)
3. Bansal, M.: Facets for single module and multi-module capacitated lot-sizing problems without backloging. *Discrete Applied Mathematics* **255**, 117–141 (2019)
4. Brahimi, N., Absi, N., Dauzère-Pérès, S., Nordli, A.: Single-item dynamic lot-sizing problems: An updated survey. *European Journal of Operational Research* **263**(3), 838–863 (2017)
5. Cunha, J.O., Konstantaras, I., Melo, R.A., Sifaleras, A.: On multi-item economic lot-sizing with remanufacturing and uncapacitated production. *Applied Mathematical Modelling* **50**, 772–780 (2017)
6. Guan, Y., Ahmed, S., Nemhauser, G.L.: Cutting planes for multistage stochastic integer programs. *Operations Research* **57**(2), 287–298 (2009)
7. Leung, J.M., Magnanti, T.L., Vachani, R.: Facets and algorithms for capacitated lot sizing. *Mathematical programming* **45**(1), 331–359 (1989)
8. Loparic, M., Pochet, Y., Wolsey, L.A.: The uncapacitated lot-sizing problem with sales and safety stocks. *Mathematical Programming* **89**(3), 487–504 (2001)
9. Miller, A.J., Nemhauser, G.L., Savelsbergh, M.W.: On the capacitated lot-sizing and continuous 0–1 knapsack polyhedra. *European Journal of Operational Research* **125**(2), 298–315 (2000)
10. Pochet, Y., Wolsey, L.A.: Polyhedra for lot-sizing with wagner—whitin costs. *Mathematical Programming* **67**(1), 297–323 (1994)
11. Pochet, Y., Wolsey, L.A.: *Production planning by mixed integer programming*. Springer Science & Business Media (2006)
12. Quezada, F., Gicquel, C., Kedad-Sidhoum, S., Vu, D.Q.: A multi-stage stochastic integer programming approach for a multi-echelon lot-sizing problem with returns and lost sales. *Computers & Operations Research* **116**, 104865 (2020)
13. Retel Helmrich, M.J., Jans, R., van den Heuvel, W., Wagelmans, A.P.: Economic lot-sizing with remanufacturing: complexity and efficient formulations. *IIE Transactions* **46**(1), 67–86 (2014)