# Robust Formulations for Economic Lot-Sizing Problem with Remanufacturing

Öykü Naz Attila*[1], Agostinho Agra[2], Kerem Akartunalı[1], and Ashwin Arulselvan[1]

[1]Dept. of Management Science, University of Strathclyde, Glasgow, UK
[2]Department of Mathematics and CIDMA, University of Aveiro, Aveiro, Portugal

**Abstract**

In this paper, we consider a lot-sizing problem with the remanufacturing option under parameter uncertainties imposed on demands and returns. Remanufacturing has recently been a fast growing area of interest for many researchers due to increasing awareness on reducing waste in production environments, and in particular studies involving remanufacturing and parameter uncertainties simultaneously are very scarce in the literature. We first present a min-max decomposition approach for this problem, where decision maker's problem and adversarial problem are treated iteratively. Then, we propose two novel extended reformulations for the decision maker's problem, addressing some of the computational challenges. An original aspect of the reformulations is that they are applied only to the latest scenario added to the decision maker's problem. Then, we present an extensive computational analysis, which provides a detailed comparison of the three formulations and evaluates the impact of key problem parameters. We conclude that the proposed extended reformulations outperform the standard formulation for a majority of the instances. We also provide insights on the impact of the problem parameters on the computational performance.

***Keywords:*** Integer Programming, Lot-Sizing, Robust Optimization, Extended Reformulations, Decomposition

*Email addresses: oyku.attila@strath.ac.uk, aagra@ua.pt, kerem.akartunali@strath.ac.uk, ashwin.arulselvan@strath.ac.uk

# 1 Introduction

Lot-sizing has been an active area of research in the manufacturing sector over the last six decades due to its significant cost savings. Primarily, the lot-sizing problem aims to find the most cost efficient production plan for given demands/orders over a specified planning horizon, which consists of decisions such as when and how much to produce/stock/backlog products and when to set up machines for manufacture of specific products, under the natural limitations of a manufacturing system, such as capacities and satisfying orders on time. As [30] demonstrate in their excellent review of mathematical models developed for lot-sizing problems, the body of research devoted to the topic is extensive, covering a rich set of tools including: i) exact methods such as valid inequalities [8, 1], extended reformulations [25, 17], and decompositions [23] often exploiting simple sub-problems, and ii) non-exact methods such as heuristics [33, 4, 38] customized for use with real-world instances inherent in manufacturing systems. More recent theoretical achievements worth remarking include [5] and [26], and we refer the interested reader to [15] for a thorough review of single-item problems, and to [16] for a recent overview on complex multi-item lot-sizing problems.

Remanufacturing is simply the process of recovering used products by repairing and replacing worn out components so that a product is created at least at the same quality level as a newly manufactured product, providing not only an environmentally sustainable alternative to classical manufacturing, saving tonnes of landfill every year, but also offering many industries from car engines to office copiers the potential for significant savings through the exploitation of used product inventories and many precious raw materials that are becoming scarcer ([22]). Remanufacturing can be operated either under a dedicated system (i.e., remanufacturing only) or a hybrid system (remanufacturing combined with manufacturing), and most remanufacturing operations in European countries, as noted by [27], employ a hybrid system. In the context of lot-sizing, hybrid models also vary between different industries and also different products, some of which allow production setups to occur jointly for manufacturing and remanufacturing, referred to as “joint setup” systems, and others requiring separate production setups, referred to as “separate setup” system. In this paper, we investigate a hybrid system with joint setups, where backlogging is also allowed.

The lot-sizing literature for remanufacturing has been growing at an increasing rate over the past two decades. The key difference from a mathematical modelling point of view in this setting is that it has at least two levels of operations, where the first level handles the returned products and hence the associated remanufacturing operations, whereas the second level of operations incorporates both remanufactured products from the first level and products to be newly manufactured in order to meet demands. The earlier works of [18] discussed the implications of the emerging reuse efforts with a review of the mathematical models proposed in the literature, and [19] studied a lot-sizing problem with remanufacturing, formulating it as a network flow model and solving by dynamic programming. [35] have shown the problem to be polynomially solvable when both remanufacturing and manufacturing processes have a joint setup cost and if they are time invariant, and the problem was shown to be $\mathcal{NP}-$hard for most general configurations, including separate setups with time invariant costs, in [31]. Further complexity results were provided by [39] for the case of concave costs, by [28] for constant capacities, by [3] for special cost structures and by [29] for the case with a disposal option. We also remark some recent and effective heuristics for practical size problems as presented in [7, 32], the recent polyhedral study of [34] comparing various reformulations and valid inequalities, and the recent work of [24] showing optimality properties and how these can be used to decompose the problem.

Although there is extensive literature on lot-sizing problems, the main focus has been on deterministic problems, where problem parameters such as demands have been assumed to be known a priori. When uncertainties are present and they cannot be sufficiently described using probability distributions (e.g., due to shortage of reliable data, or data not properly fitting into any distribution), robust optimization offers a solution that will be feasible for any realization taken in the so-called uncertainty set, i.e., the collection of all possible realizations. There has been many advances in the field of robust optimization over the last 15 years since the seminal papers of [10] describing an ellipsoid uncertainty set and [12] proposing budgeted polytopes to handle problems with discrete variables more effectively, and we refer the interested reader to the detailed reviews of [11] and [20]. The uncertainties are in particular critical in the remanufacturing setting, where both returns (numbers and product qualities) and demands are unknown, affecting both levels of the operations and their interactions in between. Although, robust approaches have been considered for classical lot-sizing problems e.g. [9, 13, 14], we are only aware of the work of [37] in this area, who consider a lot-sizing problem with uncertain demands and returns, and use a robust linear programming formulation. However, the robust formulation is based on the static approach introduced for inventory problems by [13] which is known to produce very conservative solutions for some instances, as noticed in [14]. Our recent preliminary study presented in [6] proposed an exact approach, and therefore, less conservative than the one from [37], which is known as adversarial approach (see [20]) and was introduced by [14]. This is a decomposition approach where a robust optimization min-max problem is decomposed into a master (minimization) subproblem and an adversarial (maximization) subproblem. By exploring the properties of the adversarial problem for budgeted polytopes, [2] proposed a general robust optimization dynamic programming framework that is shown to work effectively in lot-sizing problems. However, for many practical lotsizing problems, as the one considered here, the master subproblem is computationally harder to solve than the adversarial subproblem and, to the best of our knowledge, there is no work in this area studying the underlying mathematical structures for the master subproblems, such as the polyhedral characteristics or extended reformulations, which are essential ingredients in many lot-sizing algorithms developed in deterministic problems. On the other hand, it is worth to remark that stochastic programming has been used for such understanding, albeit in a very limited sense, e.g., see [21]. Thus, there is clear potential to gain invaluable insights in the remanufacturing problems by studying their mathematical properties using robust optimization.

In this paper, we primarily aim to extend our previous study that proposed a decomposition approach for a two-stage robust lot-sizing problem with remanufacturing and backlogging. We propose two extended reformulations for the computationally challenging master problem of the decomposition, where we propose first aggregating the separate production variables but extend them in the classical facility location formulation fashion, and then we propose an approximate extended reformulation. We discuss some key aspects of these reformulations (also in comparison to the basic formulation), and then present an extensive computational analysis in order to identify specific strengths and weaknesses of different formulations, as well as to support our theoretical claims. More specifically, we are extending the ideas of [36] in order to solve the robust version of the lot sizing problem with returns and remanufacturing option. In order to effectively handle the size of the master model which increases with the number of scenarios, we make the key observation on the formulation that the flow conservation only on the last scenario's demand is sufficient, which significantly reduces the size of the formulations. To the best of our knowledge, this is the first use of an extended formulation technique for multiple scenarios under a robust setting. In addition, we provide a thorough study of the structure associated with the returns and, in particular, we observe the equivalence between the uncertain sets with positive and negative deviations from the nominal values for the case of returns. Finally, from a computational perspective, we present comprehensive numerical results on the tightness of the extended formulations by providing a threshold value for the parameter $P$ for various input classes, when the lower bound improvement tails off.

In the next section, we present first a deterministic formulation of the problem with a detailed explanation of the practical setting, and then propose the robust version of this. Then, in Section 3, we propose two extended reformulations of the robust problem, and also remark the theoretical strength of using these in comparison to the basic formulation. We then present the results of thorough computational experiments in Section 4, which evaluates the proposed reformulations from a number of perspectives, including computational times, lower bounds and sensitivity to input parameters. Finally, we conclude with some key remarks and potential future directions in Section 5.

## 2 Problem Definition

In this section, we first introduce the deterministic lot-sizing problem with remanufacturing (LSR) option, where we introduce our notations and specify our assumptions on cost structures, inventories and handling of returned items. We then introduce the uncertainty involved in returned items and demands and formally define the robust LSR (RLSR) problem. We then provide a formulation using a min-max approach to obtain a robust production plan. A schematic

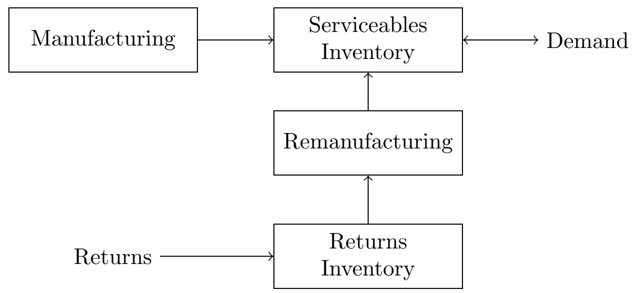


Figure 1: The production process with returns and remanufacturing

diagram providing an overview of the production process can be seen in Figure 1.

## 2.1 Deterministic LSR Formulation

We now consider LSR presented in the preliminary study of [6]. In this problem, we consider a time horizon $T$, a set of deterministic demands, $\mathcal{D} = \{D_1, D_2, \ldots, D_T\}$, and a set of deterministic returns, $\mathcal{R} = \{R_1, R_2, \ldots, R_T\}$, over the time horizon. Demands are to be satisfied by items that are produced, which can be achieved either by manufacturing an item from scratch or remanufacturing a returned item. We consider the setting where the costs involved are time invariant. At every time period, we incur a variable cost of $m$ (resp. $r$) per item manufactured (resp. remanufactured) and a fixed joint set up cost of $K$ if an item was produced in that period. Note that all costs are positive. We assume that both manufactured and remanufactured items, referred to as "serviceable items", achieve the minimum quality level necessary for satisfying the demand. In a given time period, the serviceable items at hand can be in excess or short of the demand at that period. Excess serviceable items are carried over as serviceable inventory at a cost of $h^s$ per item to the subsequent time period, and can be used to satisfy future demands. Unsatisfied demands are backlogged at a cost of $b$ per item from the subsequent time period, and have to be satisfied by serviceable items that are produced at a future time period. At each time period, we have the option to remanufacture the returned items at hand or carry them over to the next time period as unprocessed returns in the "return inventory" at a cost of $h^r$ per item, or dispose them at a cost of $f$ per item.

The output of LSR comprises of a production plan that satisfies the demand at every time period and minimizes the overall costs involved. A production plan needs to specify the amount of manufactured and remanufactured items, return and serviceable inventory levels, amount of backlogged and disposed items for every time period over the planning horizon. Table 1 presents a complete list of the decision variables.

We define the vectors $x^m := (x^m_1, x^m_2, \ldots, x^m_T)$, $x^r := (x^r_1, x^r_2, \ldots, x^r_T)$, $d := (d_1, d_2, \ldots, d_T)$, $y := (y_1, y_2, \ldots, y_T)$, and $s^r := (s^r_1, s^r_2, \ldots, s^r_T)$ associated with

| | |
|---|---|
| $x_t^m$ | Number of items manufactured in period $t$. |
| $x_t^r$ | Number of items remanufactured in period $t$. |
| $H_t^s$ | Serviceables inventory cost incurred in period $t$. |
| $H_t^r$ | Returns inventory cost incurred in period $t$. |
| $y_t$ | 1 if setup occurred in period $t$, and 0 otherwise. |
| $d_t$ | Number of returns disposed in period $t$. |

Table 1: Decision variables for the deterministic LSR problem.

production (manufacturing and remanufacturing), disposal, setup and returns inventory variables, respectively. Let $\mathbf{x} := (x^m, x^r, d, y)$. Then, the objective is to minimize the total operational cost defined by:

$$\theta^{D,R}(\mathbf{x}) + \sum_{t=1}^{T}(H_t^s + H_t^r),$$

where

$$\theta^{D,R}(\mathbf{x}) = \sum_{t=1}^{T}(Ky_t + mx_t^m + rx_t^r + fd_t)$$

W.l.o.g., we assume initial inventory levels are zero. Then, a mixed integer program (MIP) formulation for LSR is as follows:

$$\min \quad \theta^{D,R}(\mathbf{x}) + \sum_{t=1}^{T}(H_t^s + H_t^r) \tag{LSR-D}$$

st.

$$H_t^s \geq h^s \sum_{i=1}^{t}(x_i^m + x_i^r - D_i) \qquad \forall t = 1, \ldots, T \tag{1}$$

$$H_t^s \geq -b \sum_{i=1}^{t}(x_i^m + x_i^r - D_i) \qquad \forall t = 1, \ldots, T \tag{2}$$

$$H_t^r \geq h^r \sum_{i=1}^{t}(R_i - x_i^r - d_i) \qquad \forall t = 1, \ldots, T \tag{3}$$

$$M_t y_t \geq x_t^m + x_t^r \qquad \forall t = 1, \ldots, T \tag{4}$$

$$x^m, x^r, d \geq 0, \tag{5}$$

$$y \in \{0,1\}^T \tag{6}$$

The above formulation implies that demand is satisfiable through manufacturing ($x^m$) and/or remanufacturing ($x^r$), whose sum is referred to as "serviceables". Constraints (1) and (2) determine the total holding and backlogging cost for serviceables in period $t$. Note that, for a given time period $t$, at most one of these two constraints' right hand side can be non-negative, making the other constraint redundant. Returns inventory cost is determined by constraint (3). Constraint (4) ensures a joint set up when manufacturing and/or remanufacturing takes place in a given time period $t$, with an appropriate choice of $M_t$. Finally, constraints (5) and (6) enforce nonnegativity and integrality restrictions on the variables. For a fixed $y$, it is easy to observe that the problem reduces to a network flow problem.

## 2.2 Robust LSR Formulation

Determining accurate values for the input parameters of the problem is often a challenging task in practice. In addition, many input parameters in realistic applications are naturally uncertain (e.g., any quantities in the future), and there is often a danger that an optimal solution may become severely infeasible or expensive even when small changes occur in input parameters (see [10]). Although stochastic optimization is very effective in some cases, it makes the critical assumption that the uncertainty has a probabilistic description, which is not realistic in many applications. In such cases, robust optimization provides a suitable framework for handling parameter uncertainties by defining them as parts of predefined uncertainty sets. We refer the interested reader to [10, 11] for a detailed discussion on general motivations for choosing a robust optimization approach for tackling input uncertainty. As noted by [14], complexities in manufacturing systems varying from long production leadtimes to complex supply chains result in significant inadequacy of demand data, which often dictates the use of uncertainty sets for most effective treatment of uncertainties. Moreover, return of items for remanufacturing purposes entail further complications such as customer behavior or variation in the levels of use of the products, further motivating the case for a robust optimization framework.

A robust solution is defined as a solution that remains feasible over the entire uncertainty set. Such solution assumes that the production quantities represent "here and now" variables, corresponding to decisions taken before the uncertain parameters are revealed while the inventory levels are adjustable to the materialized parameters. Although such solutions provide immunity to all eventualities, considering an exhaustive number of cases may lead to solutions which may be poor for most of the reasonable scenarios. Such a solution, in order to retain feasibility, has to potentially accommodate extreme case scenarios that have a negligible chance of realisation. In order to avoid this, we employ the method introduced by [12], which controls the number of scenarios in the uncertainty set using budgeted polytopes, as discussed below. Moreover, uncertainty sets of this type are known to be tractable and computationally easier to handle.

Following the approach of [6], we model the uncertainty in demands and returns as budgeted polytopes. We assume the uncertainty in demands and returns are independent of each other. For each time period, $t = 1, \ldots, T$, we are provided with the nominal demands (resp. returns) $\bar{D}_t$ (resp. $\bar{R}_t$) and the maximum possible deviation from the nominal value $\hat{D}_t$ (resp. $\hat{R}_t$). In other words, the uncertain demand (resp. return) $D_t$ (resp. $R_t$) in time $t$ takes a value in the interval $[\bar{D}_t, \bar{D}_t + \hat{D}_t]$ (resp. $[\bar{R}_t, \bar{R}_t + \hat{R}_t]$). For each time period, $t$, we introduce the variables $z_t^D \in [0,1]$ (resp. $z_t^R \in [0,1]$), in order to model the proportion of deviation we have from the nominal demand (resp. return), namely $D_t = \bar{D}_t + \hat{D}_t z_t^D$ (resp. $R_t = \bar{R}_t + \hat{R}_t z_t^R$). Only positive deviations are considered as this corresponds to the worst case for demands (where the production is lower than expected leading to backlogged demands), and for returns we can show that feasibility implies that only the expected minimum number of items can be used (see further discussion in Section 2.3), which allows to conclude that the case of positive deviations is equivalent to the case of negative deviations. In order to avoid over-conservative estimation of the parameters, we introduce the parameters $\Gamma_t^D$ (resp. $\Gamma_t^R$) in order to constrain $z_t^D$ (resp. $z_t^R$):

$$Z^D(\Gamma^D) := \{z^D \in [0,1]^T : \sum_{i=1}^{t} z_i^D \leq \Gamma_t^D,\ \forall t = 1, \dots, T\} \tag{7}$$

$$Z^R(\Gamma^R) := \{z^R \in [0,1]^T : \sum_{i=1}^{t} z_i^R \leq \Gamma_t^R,\ \forall t = 1, \dots, T\} \tag{8}$$

Then, the independent uncertainty sets for demands and returns, respectively, can be defined as follows:

$$U^D(\Gamma^D) := \{D \in \mathbb{R}_+^T : D_t = \bar{D}_t + \hat{D}_t z_t^D, z^D \in Z^D(\Gamma^D)\} \tag{9}$$

$$U^R(\Gamma^R) := \{R \in \mathbb{R}_+^T : R_t = \bar{R}_t + \hat{R}_t z_t^R, z^R \in Z^R(\Gamma^R)\} \tag{10}$$

We will sometimes refer to the uncertainty set given in (9) and (10) as $U^{D+}(\Gamma^D)$ and $U^{R+}(\Gamma^R)$ to indicate we are considering positive deviations from the nominal value.

Recall returns inventory variables $s^r := (s_1^r, s_2^r, \dots, s_T^r)$. For the number of returns, we can use the implicit balance constraints

$$s_{i-1}^r + R_i = d_i + x_i^r + s_i^r,\ \forall i = 1, \dots, T$$

to derive the following (since $s_0^r = 0$)

$$s_t^r = \sum_{i=1}^{t} (R_i - d_i - x_i^r),\ \forall t = 1, \dots, T.$$

Non-negativity of $s_t^r$ implies

$$\sum_{i=1}^{t} (R_i - d_i - x_i^r) \geq 0,\ \forall t = 1, \dots, T. \tag{11}$$

Under the robust setting, if the returns $R$ belong to a given uncertainty set $U$ then, for all $t = 1, \dots, T$, (11) becomes

$$\sum_{i=1}^{t} (d_i + x_i^r) \leq \min\{\sum_{i=1}^{t} R_i | R \in U\}. \tag{12}$$

The following proposition establishes that there is no loss of generality in considering positive deviations. Consider the uncertainty set with negative deviations:

$$U^{R-}(\Gamma^R) := \{R \in \mathbb{R}^T_+ : R_t = \bar{R}_t - \hat{R}_t z^R_t, z^R \in Z^R(\Gamma^R)\} \tag{13}$$

**Proposition 1.** *Let* $A_t = \max\{\sum_{i=1}^{t} \hat{R}_i z^R_i | z^R \in Z^R(\Gamma^R)\}$ *for* $t = 1, \ldots, T$, *let* $\bar{S}_1 = \bar{R}_1 - A_1$, $\bar{S}_t = \sum_{i=1}^t \bar{R}_i - A_t - \sum_{i=1}^{t-1} \bar{S}_i$ *for* $t = 2, \ldots, T$ *and* $\hat{S}_t = \hat{R}_t$ *for* $t = 1, \ldots, T$, *where* $\bar{S}_t$ *signifies the smallest possible return realization in period* $t$ *under the return deviations* $\hat{S}$. *Then, for* $t = 1, \ldots, T$, *the following equalities hold:*

*a)* $\min_{R \in U^{R-}(\Gamma^R)} \sum_{i=1}^t R_i = \min_{S \in U^{R+}(\Gamma^R)} \sum_{i=1}^t S_i$.

*b)* $\max_{R \in U^{R-}(\Gamma^R)} \sum_{i=1}^t R_i = \max_{S \in U^{R+}(\Gamma^R)} \sum_{i=1}^t S_i$.

*where* $S_t$ *indicates a single scenario in the corresponding uncertainty sets.*

*Proof.* Since the proofs of (a) and (b) are almost identical, we only prove (a).

$$\min_{R \in U^{R-}(\Gamma^R)} \sum_{i=1}^{t} R_i = \min_{z^R \in Z^R(\Gamma^R)} \sum_{i=1}^{t} (\bar{R}_i - \hat{R}_i z^R_i)$$

$$= \sum_{i=1}^{t} \bar{R}_i + \min_{z^R \in Z^R(\Gamma^R)} \sum_{i=1}^{t} - \hat{R}_i z^R_i = \sum_{i=1}^{t} \bar{R}_i - \max_{z^R \in Z^R(\Gamma^R)} \sum_{i=1}^{t} \hat{R}_i z^R_i = \sum_{i=1}^{t} \bar{R}_i - A_t$$

$$= \sum_{i=1}^{t} \bar{S}_i = \min_{z^R \in Z^R(\Gamma^R)} \sum_{i=1}^{t} (\bar{S}_i + \hat{S}_i z^R_i) = \min_{S \in U^{S+}(\Gamma^R)} \sum_{i=1}^{t} S_i$$

□

**Example 2.1.** *Consider* $T = 3$ *and* $\Gamma_i = 1$, $\forall i = 1, ..., T$. *Let* $\bar{R} = (5, 5, 5)$ $\hat{R} = (1, 2, 3)$. *Here we provide a numerical example for case a) from Proposition 1, where:*

$$A_1 = \max_{z^R \in [0,1]^1} \{z^R_1 | z^R_1 \leq 1\} = 1$$

$$A_2 = \max_{z^R \in [0,1]^2} \{z^R_1 + 2z^R_2 | z^R_1 + z^R_2 \leq 1\} = 2$$

$$A_3 = \max_{z^R \in [0,1]^3} \{z^R_1 + 2z^R_2 + 3z^R_3 | z^R_1 + z^R_2 + z^R_3 \leq 1\} = 3$$

*In order to represent the minimum of a returns uncertainty set with negative deviations as one with positive deviations, we need to compute* $S$. *From Proposition 1, we have the following:*

$$\bar{S}_1 = \bar{R}_1 - A_1 = 5 - 1 = 4$$

$$\bar{S}_2 = \sum_{i=1}^{2} \bar{R}_i - A_2 - \bar{S}_1 = (5+5) - 2 - 4 = 4$$

$$\bar{S}_3 = \sum_{i=1}^{3} \bar{R}_i - A_3 - \sum_{i=1}^{2} \bar{S}_i = (5+5+5) - 3 - (4+4) = 4$$

*In this case, the minimum number of cumulative returns for the uncertainty set* $S \in U^{R+}(\Gamma^R) \sum_{i=1}^{3}(\bar{S}_i + \hat{S}_i z_i^R)$ *can be found as* $\sum_{i=1}^{3} \bar{S}_i = 12$*, in which case* $z^R = (0, 0, 0)$*. This is equivalent to the number of minimum cumulative returns derived from the uncertainty set where* $R \in U^{R-}(\Gamma^R)$*, which can be calculated as* $\sum_{i=1}^{3} \bar{R}_i - A_3 = (5+5+5) - 3 = 12$.

Note that Proposition 1 does not imply the equivalence of sets $S$ and $R$ in terms of optimality. Instead, we show that one set can be written as the other to maintain feasibility properties.

Hence we focus only on positive deviations, that is, when $U = U^{R+}(\Gamma^R)$. In this case inequalities (12) can be written as follows

$$\sum_{i=1}^{t}(\bar{R}_i - d_i - x_i^r) \geq 0, \ \forall t = 1, \ldots, T. \tag{14}$$

A favourable aspect of considering only positive deviations is that the decision maker has one absolute optimal production plan (since production variables are scenario independent). This is also favourable from a computational point of view, since the number of variables considered when we only have positive deviations is significantly less compared to the case where both negative and positive deviations are considered. On the other hand, from a practical perspective, positive deviations can be observed in many production settings. For example, in a make-to-order production environment, confirmed orders essentially provide a lower bound on demand, while any potential upcoming orders can be represented as a positive deviation in the uncertain quantity of demand. Similarly, remanufacturers operating under contracts guaranteeing minimal returns can deviate any surplus in returns with positive deviations in this setting.

An alternative characterisation for these uncertainty sets can be provided in terms of the convex hull of its extreme points as follows, where $J_D$ (resp. $J_R$) indicates the number of extreme points for demands (resp. returns):

$$Z^D(\Gamma^D) := Conv(\{z^{D^1}, z^{D^2}, \ldots, z^{D^{J_D}}\})$$

$$Z^R(\Gamma^R) := Conv(\{z^{R^1}, z^{R^2}, \ldots, z^{R^{J_R}}\})$$

and

$$U^D(\Gamma^D) := Conv(\{D^1, D^2, \ldots, D^{J_D}\})$$
$$U^R(\Gamma^R) := Conv(\{R^1, R^2, \ldots, R^{J_R}\})$$

The $t^{th}$ component of vector $D^j$ (resp. $R^j$) is given by $D_t^j = \bar{D}_t + \hat{D}_t z_t^{D^j}$ and $R_t^j = \bar{R}_t + \hat{R}_t z_t^{R^j}$, for all $j = 1, \ldots, J_D$ (resp. $j = 1, \ldots, J_R$).

Under this setting, the (RLSR) problem seeks a solution that is feasible for any demand $D \in U^D(\Gamma^D)$ and return $R \in U^R(\Gamma^R)$. As constraints (1)–(3) are affected by parameter uncertainty in LSR-D, we rewrite these constraints, so that a solution would be feasible for all $\tilde{D} \in U^D(\Gamma^D), \tilde{R} \in U^R(\Gamma^R)$ resulting in the following robust formulation for the (RLSR) problem:

$$\min \quad \theta^{D,R}(\mathbf{x}) + \pi \tag{RLSR}$$

s.t.

$$\pi \geq \sum_{t=1}^{T} (H_t^{sj} + H_t^{ri}) \qquad \forall j = 1, \ldots, J_D, \ \forall i = 1, \ldots, J_R \tag{15}$$

$$H_t^{sj} \geq h^s \sum_{i=1}^{t} (x_i^m + x_i^r - D_i^j) \qquad \forall t = 1, \ldots, T, \ \forall j = 1, \ldots, J_D \tag{16}$$

$$H_t^{sj} \geq -b \sum_{i=1}^{t} (x_i^m + x_i^r - D_i^j) \qquad \forall t = 1, \ldots, T, \ \forall j = 1, \ldots, J_D \tag{17}$$

$$H_t^{rj} \geq h^r \sum_{i=1}^{t} (R_i^j - x_i^r - d_i) \qquad \forall t = 1, \ldots, T, \ \forall j = 1, \ldots, J_R \tag{18}$$

$$\sum_{i=1}^{t} (\bar{R}_i - d_i - x_i^r) \geq 0 \qquad \forall t = 1, \ldots, T \tag{19}$$

$$(4) - (6)$$

Here, the variables $H_t^{sj}$ (resp. $H_t^{rj}$) correspond to the cost of serviceable inventory or backlogging (resp. return inventory) incurred at time $t$ for the demand $D^j$ (resp. return $R^j$). The variable $\pi$ stores the highest cost of inventory and backlogging incurred by any demand or return. Constraint (19) is elaborated on earlier as it is equivalent to (14), and enforces feasibility of a production plan for all possible realisations of returns, ensuring we do not remanufacture or dispose more than the nominal return levels. Constraints (16) - (18)are defined for all demand and return vectors corresponding to extreme points of the budgeted uncertainty sets $U^D(\Gamma^D)$ and $U^R(\Gamma^R)$. Hence, we have exponentially many constraints in our formulation. We handle this using a decomposition approach, in a similar fashion to the approach of [14], to obtain robust solutions to (RLSR), as discussed in the next section.

An important observation in (RLSR) is that the worst costs for returns can be generated in advance of solving the robust problem. In order to find the worst-case scenario for returns, we introduce the variable $H_t^{rw}$, which denotes the worst total cost associated with returns inventory at time t ($t = 1, \ldots, T$). The justification for this replacement follows from the following proposition.

**Proposition 2.** *Let, for all* $t = 1, \ldots, T$,

$$H_t^{rw} = \max_{j=1,\ldots,J_R} h^r \sum_{i=1}^{t} (R_i^j - x_i^r - d_i)$$

*and suppose* $z^{Rw}$ *is the optimal solution to*

$$\max_{z^R \in Z^R(\Gamma^R)} \sum_{t=1}^{T} (T - t + 1)\hat{R}_t z_t^R. \tag{20}$$

*Then,* $\sum_{t=1}^{T} H_t^{rw} = \sum_{t=1}^{T} h^r \sum_{i=1}^{t} (\bar{R}_i + \hat{R}_i z_i^{Rw} - d_i - x_i^r)$

*Proof.*

$$\sum_{t=1}^{T} H_t^{rw} = \sum_{t=1}^{T} \max_{j=1,\ldots,J_R} h^r \sum_{i=1}^{t} (R_i^j - x_i^r - d_i)$$

$$= h^r \sum_{t=1}^{T} \max_{z^R \in Z^R(\Gamma^R)} \sum_{i=1}^{t} (\bar{R}_i + \hat{R}_i z_i^R - x_i^r - d_i)$$

$$= h^r \sum_{t=1}^{T} \sum_{i=1}^{t} (\bar{R}_i - x_i^r - d_i) + h^r \max_{z^R \in Z^R(\Gamma^R)} \sum_{t=1}^{T} \sum_{i=1}^{t} \hat{R}_i z_i^R$$

Observing that

$$\max_{z^R \in Z^R(\Gamma^R)} \sum_{t=1}^{T} \sum_{i=1}^{t} \hat{R}_i z_i^R = \max_{z^R \in Z^R(\Gamma^R)} \sum_{t=1}^{T} (T - t + 1)\hat{R}_t z_t^R$$

which obtains the maximum when $z^R = z^{Rw}$ (from (20)) we obtain

$$h^r \sum_{t=1}^{T} \sum_{i=1}^{t} (\bar{R}_i - x_i^r - d_i) + h^r \max_{z^R \in Z^R(\Gamma^R)} \sum_{t=1}^{T} \sum_{i=1}^{t} \hat{R}_i z_i^R$$

$$= h^r \sum_{t=1}^{T} \sum_{i=1}^{t} (\bar{R}_i - x_i^r - d_i) + h^r \sum_{t=1}^{T} \sum_{i=1}^{t} \hat{R}_i z_i^{Rw}$$

$$= h^r \sum_{t=1}^{T} \sum_{i=1}^{t} (\bar{R}_i + \hat{R}_i z_i^{Rw} - d_i - x_i^r)$$

□

## 2.3 Min-Max Decomposition Approach

Our min-max approach involves iteratively solving a restricted version of (RLSR), which is referred to as the "Decision Maker's Problem" (DMP), where only a subset of extreme points, denoted by $\tilde{U}^D \subseteq U^D(\Gamma^D)$ are considered. Note that, since worst-case returns can be precomputed, these amounts are not considered. A new demand point is added to the subset $\tilde{U}^D$ at every iteration by solving a certain maximization problem that we refer to as the "Adversarial Problem" (AP). Given an optimal production plan, AP seeks the demand $D \in U^D(\Gamma^D)$ vector with the highest inventory and backlogging cost for this specific production plan. This demand vector is then used to update $\tilde{U}^D$, see also [6, 14, 40].

Let $\tilde{J}_D$ be the number of extreme points in the set $\tilde{U}^D$. Then, (DMP) can be stated as follows:

$$\min\ \theta^{D,R}(\mathbf{x}) + \pi \tag{DMP}$$

$$\text{s.t. } \pi \geq \sum_{t=1}^{T}(H_t^{sj} + H_t^{rw}) \qquad \forall j = 1,\ldots,\tilde{J}_D \tag{21}$$

$$H_t^{sj} \geq h^s \sum_{i=1}^{t}(x_i^m + x_i^r - (\bar{D}_i + \hat{D}_i z_i^{Dj})) \qquad \forall t = 1,\ldots,T,\ \forall j = 1,\ldots,\tilde{J}_D \tag{22}$$

$$H_t^{sj} \geq -b \sum_{i=1}^{t}(x_i^m + x_i^r - (\bar{D}_i + \hat{D}_i z_i^{Dj})) \qquad \forall t = 1,\ldots,T,\ \forall j = 1,\ldots,\tilde{J}_D \tag{23}$$

$$H_t^{rw} = h^r \sum_{i=1}^{t}(\bar{R}_i + \hat{R}_i z_i^{Rw} - d_i - x_i^r) \qquad \forall t = 1,\ldots,T \tag{24}$$

$$(4) - (6), (19)$$

Here, the main difference between inventory and backlogging cost constraints in (RLSR) and (DMP) formulations is that constraints (22) – (24) are written for a subset of demand and return points, rather than the complete uncertainty set. Also note that the entire constraint set (18) was replaced by the single constraint (24) for each time period.

In the following discussion, we will omit the subscript $D$ from the parameter $J_D$, as we are now only enumerating the extreme points of the uncertain demand set in our formulation. Also, we let $D_i^j := \bar{D}_i + \hat{D}_i z_i^{D_j}$, for all $i = 1,\ldots,T, j = 1,\ldots,J$.

Next, we define the Adversarial Problem (AP). Here, the aim is to find a specific demand vector that implies a higher total inventory and backlogging cost for a given production plan. As such a maximum is given by (20) for returns, AP only seeks a new demand vector. Under this setting, the optimal production plan $u^* = (x^{m*}, x^{r*}, d^*, y^*)$ of (DMP) is the input to AP. For notational simplicity, let $X_t^* = \sum_{i=1}^{t}(x_i^{m*} + x_i^{r*})$ for $t = 1,\ldots,T$. Then, AP can be written as:

$$\max \pi \tag{AP}$$

$$\text{s.t. } \pi \leq \sum_{t=1}^{T} H_t^s \tag{25}$$

$$H_t^s = \max\Big\{ h^s(X_t^* - \sum_{i=1}^{t}(\bar{D}_i + \hat{D}_i z_i^D)), \\ - b(X_t^* - \sum_{i=1}^{t}(\bar{D}_i + \hat{D}_i z_i^D)) \Big\} \qquad \forall t = 1, \ldots, T \tag{26}$$

$$\sum_{i=1}^{t} z_i^D \leq \Gamma_t^D \qquad \forall t = 1, \ldots, T \tag{27}$$

$$0 \leq z_t^D \leq 1 \qquad \forall t = 1, \ldots, T \tag{28}$$

The optimal $\pi$ value indicates the worst total inventory and backlogging costs in the uncertainty set (9), as enforced by constraints (27) and (28). Note that the true total worst cost can be computed as $\pi + \sum_{t=1}^{T}(H_t^s + H_t^{rw})$. Since $H_t^{rw}$ is a constant in (AP), we do not include this term in constraint (25). To linearize constraint (26), a new binary variable $s_t \in \{0, 1\}$, $\forall t = 1, \ldots, T$ is introduced, which is 1 if $H_t^s$ represents the inventory cost, and 0 in case of backlogging. Then, the following constraints are added to AP, where $D_t^{max} = \sum_{i=1}^{t}(\bar{D}_t + \hat{D}_t)$ and $D_t^{min} = \sum_{i=1}^{t} \bar{D}_t$ :

$$H_t^s \leq h^s(X_t^* - \sum_{i=1}^{t}(\bar{D}_i + \hat{D}_i z_i^D)) + M_{1t}(1 - s_t) \qquad \forall t = 1, \ldots, T \tag{29}$$

$$H_t^s \leq -b(X_t^* - \sum_{i=1}^{t}(\bar{D}_i + \hat{D}_i z_i^D)) + M_{2t} s_t \qquad \forall t = 1, \ldots, T \tag{30}$$

$$X_t^* - \sum_{i=1}^{t}(\bar{D}_i + \hat{D}_i z_i^D) \leq s_t(X_t^* - D_t^{min}) \qquad \forall t = 1, \ldots, T \tag{31}$$

$$- X_t^* + \sum_{i=1}^{t}(\bar{D}_i + \hat{D}_i z_i^D) \leq (s_t - 1)(X_t^* - D_t^{max}) \qquad \forall t = 1, \ldots, T \tag{32}$$

Constraints (31) and (32) ensure the correct setting of the $s_t$ variables, and then either constraint (29) or (30) is dominated, incurring either serviceables holding or backlogging cost, respectively. We note that $M_{1t}$ and $M_{2t}$ can be defined as follows:

$$M_{1t} = -b(X_t^* - D_t^{max}) - h^s(X_t^* - D_t^{max}) \qquad \forall t = 1, \ldots, T \tag{33}$$

$$M_{2t} = h^s(X_t^* - D_t^{min}) + b(X_t^* - D_t^{min}) \qquad \forall t = 1, \ldots, T \tag{34}$$

Finally, we remark that convergence is ensured through constraint (21) in (DMP), which ensures that a production plan with higher total serviceables and returns inventory cost is obtained in each iteration. In such setting, the optimal value for the objective function for (DMP) determines the global lower bound ($GLB$). On the other hand, the optimal objective value for (AP) provides a local upper bound in each iteration. In order to find the global upper bound on a given iteration $\tilde{J}_D$, we determine $GUB = \min_{j \in \{1,\ldots,\tilde{J}_D\}} \{\pi^{*j} + \theta^{*j,D,R}(u^*)\}$, where $\pi^{*j}$ indicates the optimal value for $\pi$ in (AP) (for iteration $j$) and $\theta^{*j,D,R}(u^*)$ is the optimal production and disposal costs of (AP) solved in iteration $j$. We define $\epsilon = \frac{GUB-GLB}{GLB}$ to represent the magnitude of convergence.

# 3 Extended Reformulations

Although the min-max approach is an effective method for obtaining robust optimal solutions, its computational efficiency is heavily dependent on the (DMP), as initially observed in the preliminary test of [6] and also further discussed in Section 4. Therefore, in this section, we present two extended reformulations to (DMP): "Aggregated Extended Formulation" (DMP-EFAG) and "Approximate Extended Formulation" (DMP-EFAP). We provide a detailed explanation on the structure of both formulations, while discussing their strengths and limitations. We will empirically support our claims in the Section 4 and provide a detailed account of our computational experience.

## 3.1 Extended Aggregated Reformulation

We consider a facility location reformulation for DMP, which was originally proposed by [25]. For this purpose, we introduce the following set of decision variables:

$$\mathbf{x}^{EF} := \{\tilde{x} \in \mathbb{R}_+^{(T+1)\times(T+1)} : \sum_{t=1}^{T+1} \tilde{x}_{it} = x_i^m + x_i^r, \quad \forall i = 1, \ldots, T\}, \tag{35}$$

where the new variables, $\tilde{x}_{it}$, indicate the total amount of items that have been manufactured and remanufactured in time period $i$, in order to satisfy the demand in period $t$. Throughout the paper, we refer to this quantity as the "aggregated production" quantity. This results in $(T + 1)$ new variables for each time period $t$, where the aggregated production in the $(T+1)^{th}$ period indicates the amount manufactured and remanufactured after the planning period and backlogged to satisfy the demand in a period inside the planning horizon. More specifically, we only need variables $\tilde{x}_{(T+1)i}$, for all $i$, in order to account for such backlogging. Note that we use the same objective function as in DMP, along with the original production variables $x_t^m$ and $x_t^r$.

In order to keep the formulation size reasonable and effective, we consider to apply our extended variables only to one demand scenario, namely, the one introduced in the most recent iteration $J$. For any production plan, the idea is to create an aggregated production plan corresponding to the $J^{th}$ scenario, while tightening the constraint (4) by using the aggregated decision variables. We also define separate $\tilde{H}^s_t$ and $\tilde{B}_t$ variables in order to account for the holding and backlogging costs of serviceables in the $J^{th}$ scenario. For all other iterations $j = 1, \dots, J-1$, we preserve the structure from DMP, with serviceables inventory cost defined through the original variables. Next, we state the extended reformulation formally.

$$\min \theta^{D,R}(\mathbf{x}) + \pi \qquad \text{(DMP-EFAG)}$$

$$\text{s.t. } \pi \geq \sum_{t=1}^{T} (H^{sj}_t + H^{rw}_t) \qquad \forall j = 1, \dots, J \qquad (36)$$

$$H^{sj}_t \geq h^s \sum_{i=1}^{t} (x^m_i + x^r_i - D^j_i) \qquad \forall t = 1, \dots, T, \ \forall j = 1, \dots, J-1 \qquad (37)$$

$$H^{sj}_t \geq -b \sum_{i=1}^{t} (x^m_i + x^r_i - D^j_i) \qquad \forall t = 1, \dots, T, \ \forall j = 1, \dots, J-1 \qquad (38)$$

$$\tilde{H}^s_t = h^s \sum_{i=1}^{t} \sum_{k=t+1}^{T+1} \tilde{x}_{ik} \qquad \forall t = 1, \dots, T \qquad (39)$$

$$\tilde{B}_t = b \sum_{i=1}^{t} \sum_{k=t+1}^{T+1} \tilde{x}_{ki} \qquad \forall t = 1, \dots, T \qquad (40)$$

$$H^{sJ}_t \geq \tilde{H}^s_t + \tilde{B}_t \qquad \forall t = 1, \dots, T \qquad (41)$$

$$\sum_{i=1}^{T+1} \tilde{x}_{it} = D^J_t \qquad \forall t = 1, \dots, T \qquad (42)$$

$$\sum_{i=1}^{T+1} \tilde{x}_{ti} = x^m_t + x^r_t \qquad \forall t = 1, \dots, T \qquad (43)$$

$$\tilde{x}_{tk} \leq D^J_k y_t \qquad \forall t = 1, \dots, T, \ \forall k = 1, \dots, T \qquad (44)$$

$$H^{sj}_t, x^m_t, x^r_t, d_t \geq 0 \qquad \forall t = 1, \dots, T, \ \forall j = 1, \dots, J \qquad (45)$$

$$\tilde{x}_{it} \geq 0 \qquad \forall i, t = 1, \dots, T+1 \qquad (46)$$

$$y \in \{0, 1\} \qquad (47)$$

$$(19), (24)$$

In the formulation above, constraints (37) and (38) indicate the serviceables holding and backlogging costs for scenarios where $j \neq J$. For iteration $J$, we define constraints (39) and (40), which indicate the total serviceables holding and backlogging cost for period $t$, respectively, and these costs are then linked to the variable $H_t^{sJ}$ through constraint (41). We ensure that the demand in a given period $t$ is satisfied through the sum of items manufactured and remanufactured (including backlogs from beyond the planning horizon) in constraint (42). Constraint (43) is used to link the original manufacturing and remanufacturing variables with the aggregated production variable $\tilde{x}_{it}$, and finally setup periods are determined through constraint (44). The observation we make here is that any production plan is a feasible production plan, because we allow backlogging to the final period. Therefore, which scenario's demand is assigned to the $\tilde{x}$-variables is insignificant, as this can be realised as a different production plan for another scenario and consequently, its corresponding cost calculated.

We refer to the polytope corresponding to the LP relaxation of (DMP) (resp. (DMP-EFAG)) as $\mathcal{P}_J^{DMP}$ (resp. $\mathcal{P}_J^{DMP-EFAG}$), where the subset of extreme points of the uncertainty, indexed by set $J$, have been considered in the formulation of (DMP) (resp. (DMP-EFAG)). We slightly abuse the notation here by referring both to the index set and the index of the last scenario in the index set by $J$, but we could distinguish them by context easily. We let $H^s$ to denote the vector $(H_1^{s1}, \ldots, H_T^{sJ})$. For a polytope $P := \{(u, x) \in \mathcal{U} \times \mathcal{X}\}$, where $\mathcal{U}$ and $\mathcal{X}$ are vector spaces, we define the projection of polytope $P$ onto the $x$-space (or onto $\mathcal{X}$) as

$$proj_x(P) := \{x \in \mathcal{X} : \exists u \in \mathcal{U} : (u, x) \in P\}.$$

**Proposition 3.** *For any index set $J$, $proj_{\mathbf{x},d,y,H^s,\pi}(\mathcal{P}_J^{DMP-EFAG}) \subset \mathcal{P}_J^{DMP}$*

*Proof.* It is easy to show that for any feasible solution to $\mathcal{P}^{DMP-EFAG}$, projecting out the $\mathbf{x}^{EF}$ variables results in a feasible solution to $\mathcal{P}^{DMP}$. The first thing to note is that the only difference in the two formulations is with respect to scenario $J$. Thus, we need to show that any solution $(\mathbf{x}, d, y, H^s, \pi)$ $\in proj_{\mathbf{x},d,y,H^s,\pi}(\mathcal{P}_J^{DMP-EFAG})$ satisfies constraints (22) and (23), implying $(\mathbf{x}, d, y, H^s, \pi) \in \mathcal{P}_J^{DMP}$. First, note that, for a specific $t$ and in an extreme point solution, either $\tilde{H}_t^s$ or $\tilde{B}_t$ in constraints (39) and (40) will be zero (as this could be perceived as a flow along a negative cost cycle and hence cancelled by sending flow in the opposite direction). Hence, for a given $t$, let $\tilde{B}_t = 0$. Then,

$$\begin{aligned}
H_t^{sJ} &\geq \tilde{H}_t^s + \tilde{B}_t \\
&= h^s \sum_{i=1}^{t} \sum_{k=t+1}^{T} \tilde{x}_{ik} = h^s \sum_{i=1}^{t} (\sum_{k=1}^{T} \tilde{x}_{ik} - \sum_{k=1}^{t} \tilde{x}_{ik}) \\
&= h^s \sum_{i=1}^{t} \left( (x_i^m + x_i^r) - \sum_{k=1}^{t} \tilde{x}_{ik} \right) \quad \text{(due to constraint (43))} \\
&= h^s \left( \sum_{i=1}^{t} (x_i^m + x_i^r) - \sum_{k=1}^{t} \sum_{i=1}^{t} \tilde{x}_{ik} \right) \quad \text{(rearranged terms)} \\
&\geq h^s \left( \sum_{i=1}^{t} (x_i^m + x_i^r) - \sum_{k=1}^{t} D_k^J \right) \quad \text{(due to constraint (42))}
\end{aligned}$$

$$= h^s \sum_{i=1}^{t} \left((x_t^m + x_t^r) - D_i^J\right)$$

The argument is analogous in the case when $\tilde{H}_t^s = 0$. Next, consider constraint (44). For a given $t$, summing up the constraint over all $k$, we obtain

$$x_t^m + x_t^r = \sum_{k=1}^{T+1} \tilde{x}_{tk} \leq \sum_{k=1}^{T} D_k^J y_t \leq M_t y_t$$

The first equality follows from constraint (43) and the last inequality follows from the fact that $M_t$ needs to be chosen so the formulation is feasible for any scenario. This concludes the proof for $proj_{\mathbf{x},d,y,H^s,\pi}(\mathcal{P}_J^{DMP-EFAG}) \subseteq \mathcal{P}_J^{DMP}$.

In order to show that it is a proper subset, let us consider a specific feasible solution to (DMP) with $H_t^{sJ} = 0$, $d_t = \bar{R}_t$, i.e., we have no inventory and backlogging of serviceables at any time period for scenario $J$, and all nominal returns are immediately disposed. For the sake of simplicity, assume that all nominal demands are strictly positive. Then, $y_t > 0$ holds for all $t = 1 \ldots T$ to maintain feasibility. Then, the following condition will hold for any solution to $\mathbf{x}$ in (DMP):

$$D_t^J \leq x_t^m + x_t^r \leq \sum_{i=t}^{T} (\bar{D}_i + \hat{D}_i) y_t \quad \forall t = 1, \ldots, T$$

A feasible solution satisfying this condition is when $D_t^J = x_t^m + x_t^r = \sum\limits_{i=t}^{T} (\bar{D}_i + \hat{D}_i) y_t$, which implies $y_t = D_t^J / \sum\limits_{i=t}^{T} (\bar{D}_i + \hat{D}_i)$. This produces the feasible solution $\big(\mathrm{x^m} = (D_1^J, \ldots, D_T^J), \mathrm{x^r} = (0, \ldots, 0), H^{sJ} = (0, \ldots, 0), y = (D_1^J / \sum\limits_{i=1}^{T} (\bar{D}_i + \hat{D}_i), \ldots, D_T^J / (\bar{D_T} + \hat{D_T}))\big)$ to (DMP).

Solutions of this type cannot be obtained by projection of any feasible solution for (DMP-EFAG) because constraints (42) and (44) cannot be satisfied simultaneously. From constraint (42), we have

$$\sum_{i=1}^{T+1} \tilde{x}_{it} = \tilde{x}_{tt} = D_t^J$$

The first equality is due to the fact that we have no backlogging or serviceable inventory for scenario $J$. From constraint (44), we have

$$\tilde{x}_{tt} \leq D_t^J y_t = D_t^J \frac{D_t^J}{\sum\limits_{i=t}^{T} (\bar{D}_i + \hat{D}_i)} < D_t^J$$

□

### 3.2 Approximate Extended Reformulation

Even though we are able to obtain tighter lower bounds using DMP-EFAG, the excessive number of variables can deteriorate computational performance. For this reason, preserving a relatively tight lower bound while introducing a smaller number of variables is crucial for improvement in computational times. For DMP-EFAG, one way of achieving this is to eliminate aggregated production variables that are likely to take a value of zero in the optimal solution. This is mostly the case for $\tilde{x}_{it}$ when $|i-t|$ is too high. Thus, we implement a partial formulation for DMP-EFAG, where a predefined parameter $P$ is used to define the intervals for which $\tilde{x}_{it}$ is introduced in a similar fashion to [36]. Ideally, we would like to choose $P$ such that it represents an estimation for the number of periods between consecutive setup periods.

In our study, we exploit the iterative procedure involved in the min-max approach, where we derive $P$ by tuning its value according to the structure of the optimal solutions from previous iterations. More specifically, for iteration $j$, let $\mathcal{T} = \{t^1, t^2, ..., t^s\}$ be an ordered set of increasing indices with active setup periods in the solution of iteration $j-1$, i.e., $y_t = 1$, $\forall t \in \mathcal{T}$ and $t^i < t^{i+1}$, $\forall i = 1 \dots s-1$. Then, we set $P = \max\{t^2 - t^1, t^3 - t^2, ..., t^s - t^{s-1}\}$ for the current iteration $j$. Note that for the first iteration, $P$ is chosen arbitrarily as $P = 3$, motivated by the results given in Section 4.

Once $P$ is determined, $\tilde{x}_{ti}$ is introduced for a subset of time periods, where $\forall t \in \{1, \dots, T+1\}$, and $i \in S_t^P$ such that $S_t^P = \big\{ \max\{1, t-P\}, ..., \min\{t+P, T\}\big\}$. Under these assumptions, demands in periods $i \notin S_t^P$ are not allowed to be satisfied through the aggregated production variables. Note that we will also introduce non-extended variables to allow that $D_i^j : i \notin S_t^P$ may be satisfied through production in period $t$, as follows:

- $v_t^{1s}$ : Number of items produced in period $t$ to satisfy demand in any period in the interval $[t+P+1, \dots, T]$, through keeping serviceables inventory.
- $v_t^{1b}$ : Number of items produced in period $t$ to satisfy demand in any period in the interval $[1, \dots, t-P-1]$, through backlogging.
- $v_t^{2s}$ : Amount of demand in period $t$ satisfied through $v_i^{1s} : i = [1, \dots, t-P-1]$ variables.
- $v_t^{2b}$ : Amount of demand in period $t$ satisfied through $v_i^{1b} : i = [t+P+1, \dots, T]$ variables.

In order to account for the inventory decisions taken through these variables, we also introduce further variables. First, variable $w_t^s$ (resp. $w_t^b$) represents the amount that is kept in serviceables inventory (resp. backlogged) in period $t$ through the use of $v^{1s}, v^{2s}$ (resp. $v^{1b}, v^{2b}$). The total serviceables holding and backlogging cost associated with these amounts is given as $H_t^{sPJ}$. Then, we present the formulation formally as follows, which we discuss in detail next.

$$\min \theta^{D,R}(\mathbf{x}) + \pi \qquad \text{(DMP-EFAP)}$$

$$\text{s.t.} \quad \pi \geq \sum_{t=1}^{T}(H_t^{sJ} + H_t^{sPJ} + H_t^{rw}) \tag{48}$$

$$\pi \geq \sum_{t=1}^{T}(H_t^{sj} + H_t^{rw}) \qquad \forall j = 1, \ldots, J\text{-}1 \tag{49}$$

$$\tilde{H}_t^s = h^s \sum_{i=a_t}^{t} \sum_{j=t+1}^{b_t} \tilde{x}_{ij} \qquad \forall t = 1, \ldots, T \tag{50}$$

$$\tilde{B}_t = b \sum_{i=a_t}^{t} \sum_{j=t+1}^{b_t} \tilde{x}_{ji} \qquad \forall t = 1, \ldots, T \tag{51}$$

$$H_t^{sPJ} = \tilde{H}_t^{sP} + \tilde{B}_t^P \qquad \forall t = 1, \ldots, T \tag{52}$$

$$\tilde{H}_t^{sP} = h^s(w_t^s + \sum_{i=s_t^{min}}^{s_t^{max}} v_{i+P+1}^{2s}) \qquad \forall t = 1, \ldots, T \tag{53}$$

$$\tilde{B}_t^P = b(w_t^b + \sum_{i=b_t^{min}}^{b_t^{max}} v_i^{2b}) \qquad \forall t = 1, \ldots, T \tag{54}$$

$$w_{t-1}^s + v_t^{1s} = w_t^s + v_{t+P+1}^{2s} \qquad \forall t = 1, \ldots, T\text{-}P\text{-}1 \tag{55}$$

$$w_{t-1}^s = w_t^s \qquad \forall t = T\text{-}P, \ldots, T \tag{56}$$

$$w_{t-1}^b + v_{t-P-1}^{2b} = w_t^b + v_t^{1b} \qquad \forall t = P\text{+}2, \ldots, T \tag{57}$$

$$D_t^J = \begin{cases} v_t^{2b} + \sum_{i=a_t}^{b_t} x_{it}, & \forall t = 1, \ldots, v_{min}^2 \\ v_t^{2s} + v_t^{2b} + \sum_{i=a_t}^{b_t} x_{it}, & \forall t = P\text{+}2, \ldots, T\text{-}P \\ v_t^{2s} + \sum_{i=a_t}^{b_t} x_{it}, & \forall t = v_{max}^2, \ldots, T \\ \sum_{i=a_t}^{b_t} x_{it}, & \text{if } P\text{+}2 > T\text{-}P, \ \forall t = v_{min}^2\text{+}1, \ldots, v_{max}^2\text{-}1 \end{cases} \tag{58}$$

$$x_t^m + x_t^r = \begin{cases} v_t^{1s} + \sum_{i=a_t}^{b_t} x_{ti} & \forall t = 1, \ldots, v_{min}^1 \\ v_t^{1b} + \sum_{i=a_t}^{b_t} x_{ti} & \forall t = v_{max}^1, \ldots, T \\ \sum_{i=a_t}^{b_t} x_{ti} & \text{if } P\text{+}2 > T\text{-}P\text{-}1 \ \forall t = v_{min}^1\text{+}1, \ldots, v_{max}^1\text{-}1 \\ v_t^{1s} + v_t^{1b} + \sum_{i=a_t}^{b_t} x_{ti} & \forall t = P\text{+}2, \ldots, T\text{-}P\text{-}1 \end{cases} \tag{59}$$

$$\tilde{x}_{it} \leq (\bar{D}_t + \hat{D}_t)y_i \qquad \forall i = 1, \ldots, T \ \ \forall t \in S_i^P \tag{60}$$

$$x_t^m + x_t^r \leq M_t y_t \qquad \forall t = 1, \ldots, T \tag{61}$$

$$v_t^{1s} \leq \sum_{i=t+P+1}^{T} (\bar{D}_i + \hat{D}_i) y_t \qquad \forall t = 1, \ldots, T\text{-}P\text{-}1 \tag{62}$$

$$v_t^{1b} \leq \sum_{i=1}^{t-P-1} (\bar{D}_i + \hat{D}_i) y_t \qquad \forall t = P\text{+}2, \ldots, T \tag{63}$$

$$H_t^{rj}, x_t^m, x_t^r, x_{it}, d_t \geq 0 \tag{64}$$

$$y_t, \text{binary} \tag{65}$$

$$(19), (37) - (41)$$

Here, we have three different types of inventory costs. The first set arise from the original variables in (DMP) and hence presented again by constraints (37) and (38), through which we decide the inventory levels for demand scenarios $j = 1, ..., J-1$, and therefore, the total serviceables inventory cost as $H^{sj}$.

Secondly, there is inventory cost incurred through variables $\tilde{x}_{it}$ and constraints (50) and (51), by which the inventory levels for the last scenario $J$ are decided. These individual costs are linked to the variable $H_t^{sj}$ by constraint (41).

Thirdly, we consider the inventory costs incurred through the non-extended variables, where the total inventory cost is given by $H_t^{sPJ}$ in constraint (52). Constraints (53) and (54) indicate the independent serviceables inventory and backlogging costs, respectively, and we define the following measures to determine the specific variables that contribute to these costs, based on the values of $t$ and $P$:

- $[s_t^{min}, s_t^{max}]$ where $s_t^{min} = \max\{1,\ t-P\}$ and $s_t^{max} = \min\{t,\ T-P-1\}$: Determines the interval, in which serviceables holding cost is incurred through $v_t^{2s}$ for period $t$.
- $[b_t^{min}, b_t^{max}]$ where $b_t^{min} = \max\{1,\ t-P\}$ and $b_t^{max} = \min\{t,\ T-P\}$: Determines the interval, in which backlogging cost is incurred through $v_t^{2b}$ for period $t$.

Constraint (48) is used to determine the total serviceables inventory and backlogging cost for the last scenario $J$, where all three types of inventory costs are summed. As the remaining demand points are handled through the constraints in DMP-EFAG, we indicate the total inventory costs for scenarios $j = 1, ..., J-1$ through constraint (49).

Flow conservation of non-extended variables are achieved through constraints (55)-(57). The set of constraints in (58) ensure that demand is satisfied in each time period. Here, we have four different cases, depending on the specific values of $t$ and $P$ as aggregated production variables are only introduced for the interval $[t-P, t+P]$. Under this setting, demand can either be satisfied by both, none or only one of the approximate and aggregated production variables. In order to determine the exact intervals for each of these cases, we introduce the following:

- $v^2_{min} = \min\{T-P,\ P+1\}$: Determines the period until which demand can be satisfied through $v^{2b}_t$ and approximate extended production variables.
- $v^2_{max} = \max\{T-P+1,\ P+2\}$: Determines the period from which demand can be satisfied through $v^{2s}_t$ and approximate extended production variables.

Similarly, the flow conservation constraints given in (59) for production variables vary for specific combinations of $t$, $P$ and $T$. In this case, the value of the original production variables $x^m_t$ and $x^r_t$ is either equal to the sum of the extended production variables, $\tilde{x}_{it}$, or slip between the sum and $v^{1s}_t$ and/or $v^{1b}_t$. The sum of extended production variables is given as $\sum_{i=a_t}^{b_t} x_{ti}$, where $a_t = \max\{1,\ t-P\}$ and $b_t = \min\{t+P,\ T+1\}$. Here, $a_t$ and $b_t$ are used to ensure that the approximate variables at the beginning and end of the planning horizon remain in the set $S^P_i$. The intervals for each case is determined according to the following values:

- $v^1_{min} = \min\{T-P-1,\ P+1\}$: Determines the period until which the total production (through original production variables $x^m$ and $x^r$) is distributed between approximate extended production variables ($\tilde{x}_{it}$) and $v^{1s}_t$.
- $v^1_{max} = \max\{T-P,\ P+2\}$: Determines the period from which the total production is distributed between approximate extended production variables and $v^{1b}$ (until the last period $T$).

Figure 2 illustrates an example for the use of approximate variables, where $T=5$ and $P=1$. In contrast to DMP-EFAG, the value of the original manufacturing and remanufacturing variables are not only distributed between the aggregated production variables, but also $v^{1s}$ and $v^{1b}$, where applicable. Note that Figure 2 only illustrates the decisions taken on the serviceables level, as decisions related to returns and remanufacturing remain unchanged.

Finally, we ensure that a joint setup cost is incurred when production takes place in constraints (60) - (63). As $v^{1s}_t$ is only used to satisfy demand points in time periods $[t+P+1,\ldots,T]$, we set $M_t = \sum_{i=t+P+1}^{T} (\bar{D}_i + \hat{D}_i)$ for constraint (62). Similarly, for constraint (63), as demands that are backlogged through the non-extended variables are defined as $v^{1b}_t$ are in the interval $[1,\ldots,t-P-1]$, we define $M_t = \sum_{i=1}^{t-P-1} (\bar{D}_i + \hat{D}_i)$.

# 4 Computational Results

The computational experiments presented in this section are conducted with datasets that have been generated and used in the study presented in [6]. Moreover, we follow the same Benders' framework presented in [6]. In this framework,

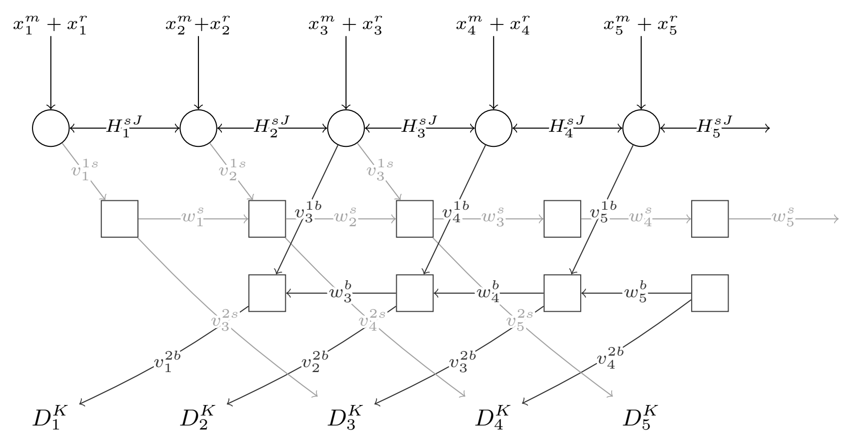

Figure 2: (DMP-EFAP) with $P = 1$ and $T = 5$

the initial upper bound (UB) and lower bound (LB) are set to $\infty$ and 0, respectively. Then, LB is updated at every iteration after the (DMP) is re-solved with a new scenario, while UB is updated only if the AP improves UB, where the minimum of the current UB and the cost corresponding to the new scenario will be taken as the new UB.

For all datasets, the nominal demand is generated within the interval $\bar{D} = [D_{min}, D_{max}]$, where $D_{min} = 50$ and $D_{max} = 100$, and the serviceables holding cost is generated in the interval $h^s = [5, 10]$. The manufacturing and remanufacturing cost are defined as $m = m^f * h^s$ and $r = 2 * h^r$, where we refer to $m^f$ as the *manufacturing factor*, which is set as $m^f = 2$ for all datasets used in the computational tests given below (Tables 2, 3 and 4). We set the backlogging cost as $b = 4 * h^s$. Note that we may set the backlogging cost for the last time period higher than $b$ in order to account for the setup and production costs outside the planning horizon, where $b_T = kb$, such that $k > 1$. Although this accounts for the costs incurred beyond the last time period $T$, it is worth noting that $b_T$ is only an estimated measure for costs associated with decisions that are taken beyond the planning horizon. As a result of this, the exact realization of costs for period $T + 1$ may differ from what is incurred in an optimal solution. Furthermore, we identify the following key parameters and their variations in order to obtain a broad variety of problem characteristics:

- Very high, high, medium and low levels of the setup cost, $K^V = 200 * h^s * D_{max}$ , $K^H = 5 * h^s * D_{max}$, $K^M = 2 * h^s * (\frac{D_{max}+D_{min}}{2})$, $K^L = 0.1 * h^s * D_{min}$, respectively.

- High, medium and low levels of nominal returns, $\bar{R}^H \in [0.7 * D_{min}, 0.7 * D_{max}]$, $\bar{R}^M \in [0.5 * D_{min}, 0.5 * D_{max}]$, $\bar{R}^L \in [0.3 * D_{min}, 0.3 * D_{max}]$, respectively.

| | $K^V$ | | | $K^H$ | | |
|---|---|---|---|---|---|---|
| $p, d$ | $\bar{R}^H$ | $\bar{R}^M$ | $\bar{R}^L$ | $\bar{R}^H$ | $\bar{R}^M$ | $\bar{R}^L$ |
| H,G | 12.0, *2.0* | 11.4, *2.0* | 9.1, *2.0* | – | – | – |
| H,L | 11.3, *2.0* | 8.9, *2.0* | 10.9, *2.0* | 9724.4, *3.0* | – | – |
| M,G | 11.7, *2.0* | 11.8, *2.0* | 9.1, *2.0* | – | – | – |
| M,L | 10.8, *2.0* | 9.8, *2.0* | 10.3, *2.0* | 9362.9, *3.0* | – | – |
| L,G | 14.4, *2.0* | 12.7, *2.0* | 10.9, *2.0* | 9144.7, *3.0* | 6143.3, *3.2* | 8725.9, *3.0* |
| L,L | 11.9, *2.0* | 10.3, *2.0* | 9.7, *2.0* | 5899.9, *3.2* | 8034.6, *3.3* | 8959.2, *3.0* |
| **Mean** | 12.0, *2.0* | 10.8, *2.0* | 10.0, *2.0* | 9024.2, *3.1* | 9030.7, *3.3* | 9615.3, *3.0* |
| | $K^M$ | | | $K^L$ | | |
| $p, d$ | $\bar{R}^H$ | $\bar{R}^M$ | $\bar{R}^L$ | $\bar{R}^H$ | $\bar{R}^M$ | $\bar{R}^L$ |
| H,G | 4808.6, *4.8* | 3302.3, *5.3* | 2948.0, *4.6* | 45.2, *10.6* | 12.2, *9.0* | 12.5, *9.2* |
| H,L | 1631.0, *4.6* | 6378.9, *5.0* | 4909.3, *5.0* | 11.4, *8.4* | 17.9, *10.6* | 17.9, *10.2* |
| M,G | 2888.4, *3.5* | 1235.8, *5.0* | 2578.7, *5.0* | 56.4, *9.0* | 15.6, *8.8* | 12.3, *7.4* |
| M,L | 3999.2, *4.3* | 1407.9, *4.8* | 3794.7, *5.2* | 15.4, *10.4* | 25.3, *10.6* | 21.0, *10.4* |
| L,G | 2707.4, *3.8* | 1270.4, *4.8* | 166.0, *3.8* | 92.7, *10.4* | 20.8, *10.0* | 18.9, *9.4* |
| L,L | 973.8, *5.0* | 442.2, *4.6* | 1789.0, *4.6* | 16.1, *8.6* | 22.1, *9.2* | 17.0, *8.2* |
| **Mean** | 2834.7, *4.3* | 2339.6, *4.9* | 2697.6, *4.7* | 39.5, *9.6* | 19.0, *9.7* | 16.6, *9.1* |

"–" indicates that time limit was reached for these instances before reaching the desired optimality gap.

Table 2: Average computational time (in sec.) and average number of iterations required to reach convergence (given in italic, excluding instances where the time limit is reached) for DMP with $T = 50$ for all datasets.

- High, medium and low probability of constraint violation caused by $\Gamma_t$, $p^H = 0.1$, $p^M = 0.05$, $p^L = 0.01$, respectively, where the probability measures are calculated in a similar fashion as proposed in [12], where we use the approximation $p = 1 - \Phi(\frac{\Gamma_t - 1}{\sqrt{t}})$ to obtain the values for $\Gamma_t^D$ and $\Gamma_t^R$ according to desired levels of probability.

- Disposal cost, either less or greater than the remanufacturing cost, set as $f^L = \frac{r}{2}$, $f^G = 2 * r$, respectively.

This experimental design resulted in 72 different combinations and hence 72 datasets were generated, with five instances in each dataset. We also note that the parameter deviations for a given period $t$, i.e., $\hat{D}_t$ and $\hat{R}_t$, are set as $0.1 * \bar{D}_t$ and $0.1 * \bar{R}_t$, respectively, for all datasets.

All instances were solved as MIPs using Java API for CPLEX 12.7 on an Intel Core i5, 3.30 GHz CPU, 3.29 GHz, 8 GB RAM machine. The terminating condition is met when either the time limit of 10,000s is reached, or a robust optimal solution is found, where $\epsilon = 0.01$. In order to tackle the excessive time requirements while solving the DMP, the MIP gap tolerance for earlier iterations were kept higher, while the final iteration has a relative MIP gap optimality tolerance of 1%. As the last iteration cannot be determined in advance, the MIP gap tolerance is reverted to 1% when $\epsilon \approx 0.01$ is achieved, and kept unchanged until a robust optimal solution is found.

We begin presenting the computational results for DMP, through which we highlight the strengths and weaknesses of the extended reformulations (DMP-

EFAG) and (DMP-EFAP). Before discussing detailed results, we note that a common observation for all three formulations is that an optimal solution to the adversarial problem is achieved under a maximum of 20 seconds for all instances and datasets. On the other hand, for certain instances and datasets, a disproportionate amount of time is required to solve the decision maker's problem. For this reason, we assume that the total time requirements for the decomposition algorithm is representative of the time requirements for solving the decision maker's problem.

As the results in Table 2 indicate, there exists a significant difference in the computational times when setup costs vary. We first observe that the instances with very high ($K \in K^V$) and low ($K \in K^L$) setup costs are solved very quickly, whereas the computational times are significantly higher for instances with medium setup costs ($K \in K^M$) and the majority of instances with high setup costs ($K \in K^H$) even exhaust the time limit of 10,000s. When setup costs are decreased towards zero, one would naturally expect the problem to become much easier to solve, since the binary decisions become almost obsolete as one may set all or almost all of them to 1. On the other hand, increasing setup costs from very low up to a certain level naturally complicates the solution procedure, as the combinatorial nature of setup decisions becomes much more dominating as a result of the competition between such decisions. However, once setup costs are significantly increased, then the problem would again become naturally easy to solve, as setups become prohibitive and hence almost all setup variables will be set to 0. We also observe that the computational times in general decrease as the probability of constraint violation decreases from high ($p^H$) to low ($p^L$). This is not unexpected, as lower probability of constraint violation would naturally ease the search process for feasible solutions. Finally, although we observe a significant variation in times when nominal return levels vary between $\bar{R} \in \bar{R}^H$, $\bar{R}^M$ and $\bar{R}^L$ or disposal costs vary between $f^L$ and $f^G$, we can not observe a clear tendency as to when the computational times would increase or decrease. However, this does not mean that we should exclude their impact, because it may be possible to observe a pattern after controlling other factors.

Next, in the same fashion, we present the computational times for the extended aggregate reformulation (DMP-EFAG) and approximate extended reformulation (DMP-EFAP) in Tables 3 and 4, respectively. In comparison to previous results, (DMP-EFAG) has a vast improvement on the overall time performances for datasets with $K \in K^H$, where we are now able to solve all instances except one within the time limit. Although the average time requirements remain similar for $K \in K^M$, some datasets such as those with the probability $p^L$ are solved much more efficiently, within only 243 seconds. Similar to previous results, datasets with low values of setup costs can still be solved very fast.

As the results in Table 4 indicate, (DMP-EFAP) has even further improved the computational times for datasets with $K \in K^H$ in comparison to (DMP-EFAG), where the average time requirement across datasets is now 11.2 seconds for $\bar{R} \in \bar{R}^H$, contrary to the time performance of 347.8 seconds for (DMP-

| | $K^V$ | | | $K^H$ | | |
|---|---|---|---|---|---|---|
| $p, d$ | $\bar{R}^H$ | $\bar{R}^M$ | $\bar{R}^L$ | $\bar{R}^H$ | $\bar{R}^M$ | $\bar{R}^L$ |
| H,G | 1.6, *2.0* | 0.8, *2.0* | 0.8, *2.0* | 854.2, *2.8* | 472.9, *2.4* | 167.8, *2.4* |
| H,L | 1.6, *2.0* | 0.9, *2.0* | 0.8, *2.0* | 295.9, *2.2* | 2.0, *2.2* | 2.8, *2.2* |
| M,G | 1.6, *2.0* | 1.2, *2.0* | 0.7, *2.0* | 923.9, *2.4* | 1.7, *2.0* | 3.2, *2.4* |
| M,L | 1.3, *2.0* | 0.8, *2.0* | 0.7, *2.0* | 3.6, *2.4* | 3.2, *2.2* | 651.1, *2.6* |
| L,G | 1.0, *2.0* | 0.9, *2.0* | 0.7, *2.0* | 5.4, *2.4* | 4.3, *2.2* | 3372.1, *2.5* |
| L,L | 0.9, *2.0* | 0.8, *2.0* | 0.8, *2.0* | 3.7 , *2.2* | 3.4, *2.2* | 5.2, *2.6* |
| **Mean** | 1.3, *2.0* | 0.9, *2.0* | 0.8, *2.0* | 347.8, *2.4* | 81.3, *2.2* | 700.4, *2.5* |
| | $K^M$ | | | $K^L$ | | |
| $p, d$ | $\bar{R}^H$ | $\bar{R}^M$ | $\bar{R}^L$ | $\bar{R}^H$ | $\bar{R}^M$ | $\bar{R}^L$ |
| H,G | 3140.5, *4.3* | 1122.9, *3.8* | 152.6, *3.2* | 12.3, *6.2* | 12.9, *6.4* | 13.6, *6.4* |
| H,L | 1603.3, *3.4* | 620.1, *3.2* | 1770.7, *3.8* | 18.4, *6.4* | 16.8, *7.2* | 12.7, *6.0* |
| M,G | 2646.4, *3.3* | 2036.4, *3.3* | 3516.4, *3.5* | 15.1, *6.2* | 12.9, *5.8* | 29.8, *6.8* |
| M,L | 3566.6, *3.3* | 459.3, *4.2* | 894.8, *3.4* | 12.0, *5.2* | 12.3, *6.0* | 13.5, *6.4* |
| L,G | 17.4, *3.0* | 25.8, *3.2* | 4.8, *3.2* | 12.0, *4.8* | 11.3, *5.4* | 16.2, *6.8* |
| L,L | 70.9, *3.8* | 8.8, *3.4* | 27.4, *3.4* | 13.6, *6.0* | 18.2, *7.0* | 22.9, *7.6* |
| **Mean** | 1840.8, *3.5* | 712.2, *3.5* | 1061.1, *3.4* | 13.9, *5.8* | 14.1, *6.3* | 18.1, *6.7* |

Table 3: Average computational time (in sec.) and average number of iterations required to reach convergence (given in italic, excluding instances where the time limit is reached) for DMP-EFAG with T=50 for all datasets.

| | $K^V$ | | | $K^H$ | | |
|---|---|---|---|---|---|---|
| $p, d$ | $\bar{R}^H$ | $\bar{R}^M$ | $\bar{R}^L$ | $\bar{R}^H$ | $\bar{R}^M$ | $\bar{R}^L$ |
| H,G | 3.2, *2.0* | 2.4, *2.0* | 2.3, *2.0* | 17.4, *2.6* | 1099.2, *3.0* | 145.7, *2.4* |
| H,L | 2.9, *2.0* | 2.5, *2.0* | 2.3, *2.0* | 33.9, *2.6* | 2.7, *2.4* | 95.0, *2.6* |
| M,G | 2.4, *2.0* | 2.4, *2.0* | 2.2, *2.0* | 2.3, *2.0* | 2.8, *2.6* | 2.5, *2.4* |
| M,L | 2.5, *2.0* | 2.6, *2.0* | 2.4, *2.0* | 4.8, *2.8* | 830.9, *2.4* | 2001.7, *2.3* |
| L,G | 2.7, *2.0* | 2.5, *2.0* | 2.3, *2.0* | 4.3, *2.4* | 2.4, *2.0* | 2.5, *2.2* |
| L,L | 2.5, *2.0* | 2.4, *2.0* | 2.3, *2.0* | 4.3, *2.4* | 2.4, *2.0* | 2.4, *2.0* |
| **Mean** | 2.7, *2.0* | 2.4, *2.0* | 2.3, *2.0* | 11.2, *2.5* | 323.4, *2.4* | 375.0, *2.3* |
| | $K^M$ | | | $K^L$ | | |
| $p, d$ | $\bar{R}^H$ | $\bar{R}^M$ | $\bar{R}^L$ | $\bar{R}^H$ | $\bar{R}^M$ | $\bar{R}^L$ |
| H,G | 6073.2, *4.5* | 2407.5, *3.5* | 248.7, *3.0* | 11.5, *7.2* | 14.7, *7.2* | 25.2, *8.8* |
| H,L | 2649.5, *4.2* | 2409.3, *3.8* | 8002.5, *2.0* | 11.2, *7.0* | 18.8, *8.2* | 14.3, *7.0* |
| M,G | 2069.1, *3.3* | 1645.5, *3.2* | 2069.6, *4.5* | 10.7, *5.8* | 18.3, *6.8* | 22.7, *8.0* |
| M,L | 2016.3, *2.8* | 43.7, *3.4* | 4032.7, *3.0* | 11.7, *6.2* | 16.0, *7.8* | 19.5, *7.6* |
| L,G | 19.6, *2.8* | 12.8, *3.0* | 4.2, *2.8* | 11.8, *6.2* | 17.4, *6.4* | 12.8, *6.8* |
| L,L | 21.1, *2.8* | 14.8, *3.2* | 4.8, *3.0* | 13.6, *6.6* | 24.7, *8.4* | 15.2, *6.2* |
| **Mean** | 2141.5, *3.4* | 1088.9, *3.3* | 2393.8, *3.1* | 11.8, *6.5* | 18.3, *7.5* | 18.3, *7.4* |

Table 4: Average computational time (in sec.) and average number of iterations required to reach convergence (given in italic, excluding instances where the time limit is reached) for DMP-EFAP with T=50 for all datasets, using the maximum interval approach to determine $P$.

EFAG). One possible reason for this can be observed in Figure 7, where we observe that for low values of the manufacturing factor $m^f$, low values of $P$ are sufficient to improve the lower bound so that (DMP-EFAG) does not have an advantage over the approximate extended formulation (DMP-EFAP). Although the approximate extended reformulation has achieved worse times in the set $K \in K^M$ when the high probability $p^H$ of infeasibility parameter is applied, it has a better or similar time performance in comparison to (DMP-EFAG) for $p^M$ and $p^L$. The inferential observation from Tables 3 and 4 is that the variation in computational times with respect to varying levels of setup costs and returns is similar to (DMP). Although the computational times for instances with high setup costs ($K \in K^H$) are significantly reduced, instances with medium level setup costs ($K \in K^M$) are the most challenging for extended reformulations.

Another interesting aspect to remark here is the number of scenarios needed to reach convergence. We observe that the total number of iterations mainly varies for different levels of the setup cost. As the results in Tables 2, 3 and Table 4 suggest, lower levels of setup costs tend to increase the number of scenarios required to reach convergence for all three formulations. All instances for $K \in K^V$ have managed to reach a robust optimal solution in 2 iterations, whereas this number is much higher for $K \in K^L$. More specifically, instances with lower setup costs require on average $\approx 9.5$ iterations to converge in DMP, whereas this decreases to $\approx 6.3$ for (DMP-EFAG) and to $\approx 7.1$ for (DMP-EFAP). This behavior is also observed for setup costs where $K \in K^H, K^M$. However, the difference between reformulations is less significant for these classes of datasets, where (DMP) requires on average $\approx 3.9$ iterations, while this amount is $\approx 2.9$ for (DMP-EFAG) and $\approx 2.8$ for (DMP-EFAP).

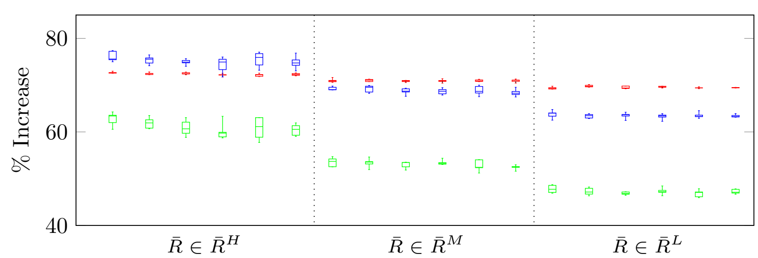


Figure 3: Percentage increase in the lower bound for DMP-EFAG with respect to DMP for $\tilde{J}_D = 1$, for different levels of returns ($\bar{R} \in \bar{R}^H, \bar{R}^M, \bar{R}^L$) when $K \in K^V$ (red), $K \in K^H$ (blue) and $K \in K^M$ (green).

Another important consideration is with respect to the improvement in lower bounds when extended reformulations are applied. Figure 3 indicates the percentage improvement of the lower bounds at the root node in (DMP-EFAG) with respect to (DMP). We can observe that the most significant gains are achieved for datasets with high setup costs, where we are able to obtain an improvement of $\approx 75\%$ over varying levels of returns. On the other hand, the gains in lower bounds are slightly less when the setup cost levels are medium, though they are

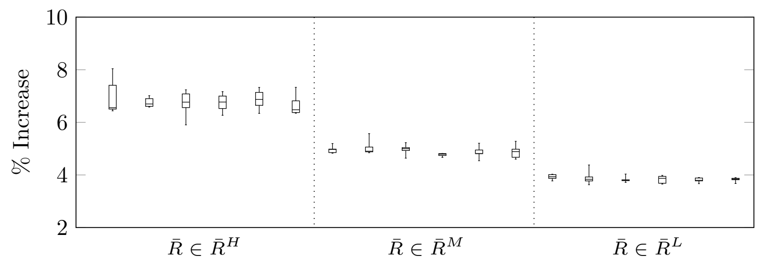


Figure 4: Percentage increase in the lower bound for DMP-EFAG with respect to DMP, for different levels of returns: $\bar{R} \in \bar{R}^H, \bar{R}^M, \bar{R}^L$ when $K \in K^L$

still very effective with an improvement of $\approx 60\%$ over varying levels of returns.

In a similar fashion, we present the improvements in lower bounds for low setup cost datasets in Figure 4. The tendency of increasing improvements as returns level move towards high can also be observed in this case. However, in comparison with previous results, low setup costs result in less significant gains, with the improvements achieving at most a maximum of 8.0%. This is likely to occur as the fractionality in setup variables in the LP relaxation of (DMP) would not result in significant cost improvement compared with the integer solution since the setup cost associated with these variables are themselves low.

Another interesting aspect for comparison is the computational behaviour when the extended reformulations are applied, as demonstrated in Figure 5. With the tolerance for the MIP gap set to 1%, we are able to obtain optimal solutions for (DMP) for 71.9% of the instances among all datasets, whereas this percentage shows a considerable increase to 97.8% and 94.4% for (DMP-EFAG) and (DMP-EFAP), respectively. In addition, we observe that only 34.8% of the instances were solved under 100 seconds for (DMP), majority of which are those with $K \in K^L$ (as seen in Table 2 before), as they constitute 33.3% of the total number of instances (excluding instances where $K \in K^V$). On the other hand, for (DMP-EFAG) and (DMP-EFAP), we observe a vast increase in the number of instances solved under 100 seconds, with 80.4% and 84.4% of the instances, respectively. This clearly implies a strong improvement in the overall computational performance for both reformulations. Another point to note here is that the variance among the computational time requirements for instances that are solved quickly (under 100 seconds) is very small for all formulations.

Although the computational time requirements for both (DMP-EFAG) and (DMP-EFAP) have shown a significant improvement, we observe that as computational time increases, (DMP-EFAG) becomes the more effective method, achieving a higher percentage of instances solved in comparison to (DMP-EFAP), which is observed when the time requirement surpasses 2079 seconds. On the other hand, for cases requiring less computational time, (DMP-EFAP) is the method of choice, achieving a higher percentage of instances solved. In addition, we remark that the choice of $P$ plays a crucial role in the computational time performance for (DMP-EFAP), and thus has an impact on the resulting

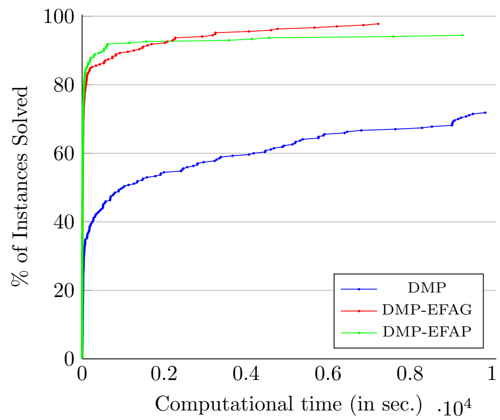


Figure 5: Percentage of instances (over datasets with $K \in K^H, K^M, K^L$) that are solved to optimality, where the MIP gap tolerance is set as 1%.

computational time performance.

Next, we analyze the impact of our choice of $m^f$ and $P$ in (DMP-EFAP) on the optimal objective value obtained from its LP relaxation. As we increase $P$, this results in an increase in the number of extended variables and constraints (60), which enable us to obtain tighter relaxations. However, as $P$ increases, the LP relaxation value of (DMP-EFAP) increases towards the LP relaxation value of (DMP-EFAG). Hence, we define $P^s$ as the $P$ value for which this increase becomes negligible. We classify the increase as negligible when the difference in the optimal objective function value between two LP relaxations is below 0.001. The manufacturing factor $m^f$ plays a crucial role in the value of $P^s$. As $m^f$ increases, remanufacturing and backlogging naturally become more favorable. This would result in a greater number of extended variables becoming active, and hence a higher value of $P^s$ is needed.

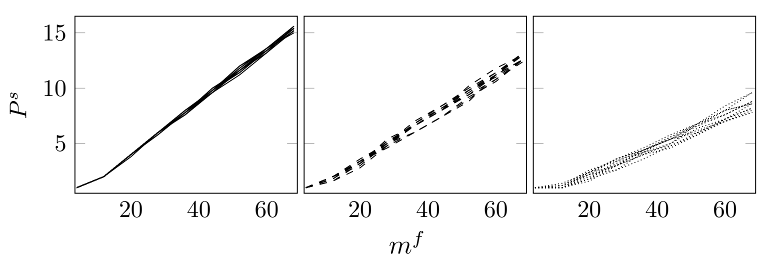


Figure 6: Average $P^s$ for various manufacturing factors, where $K \in K^M, K^L$ and $\bar{R} \in R^H$ (dotted), $R^M$ (dashed), $R^L$ (straight).

From Figures 6 and 7, we can observe that the rate of increase in $P^s$ varies with respect to the levels of returns and setup costs, as expected from our discussion above. As seen in Figure 7 for high setup costs, $P^s$ remains fairly

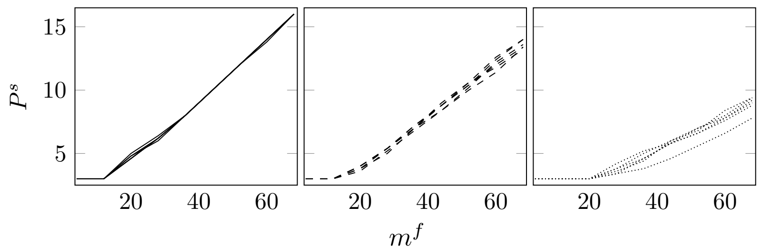


Figure 7: Average $P^s$ for various manufacturing factors, where $K \in K^H$, and $\bar{R} \in R^H$ (dotted), $R^M$ (dashed), $R^L$ (straight).

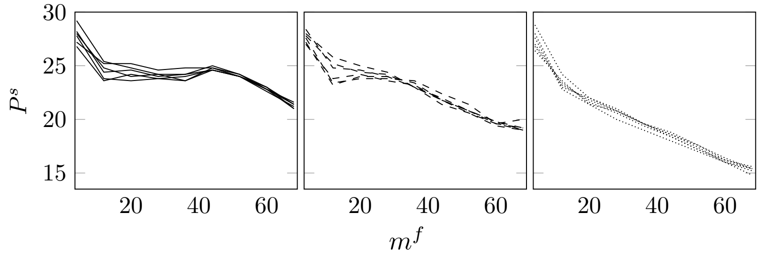


Figure 8: Average $P^s$ for various manufacturing factors for $T = 50$, where $K \in K^V$, and $\bar{R} \in R^H$ (dotted), $R^M$ (dashed), $R^L$ (straight).

low for low values of $m^f$ (till around an average value of 18.6) and it starts increasing with $m^f$. This average value of $m^f$ till which $P^s$ remains low drops to 10.6 for medium and low setup costs, as seen in Figure 6. On the other hand, since an increase in returns allows higher rates of remanufacturing rather than backlogging (and hence we can expect a decrease in extended production variables), we would expect a slower rate of increase for $P^s$ with higher returns. This behavior can be observed in Figures 6 and 7, where it is easy to see that $P^s$ starts increasing with $m^f$ but with a gentler slope for datasets with $R \in R^H$ (dotted), in comparison to the those with $R \in R^M, R^L$ (dashed and straight lines, respectively). On the other hand, as seen in Figure 8, datasets with $K \in K^V$ have a much greater overall $P^s$ value due to the significant increase in the setup cost. An interesting behaviour we observe here, unlike the previous cases, is the decrease in the value of $P^s$ as $m^f$ increases. As a result of very expensive setup costs, only a very limited number of setups is expected in the optimal solution, and we observe this often with a single setup taking place in the optimal solutions of these instances. As $m^f$ increases, we observe that the setup periods start to split the horizon more equally in order to balance remanufacturing and backlogging costs, which in turn decreases the $P^s$ value. We also observe that occasionally a significantly higher $m^f$ value results in an additional setup period, again contributing to the decrease in $P^s$.

# 5 Conclusions

In this paper, we study a lot-sizing problem with the remanufacturing option, where uncertainties exist simultaneously for demand and return parameters. Following the setting of our previous work ([6]), we define parameter uncertainties in the form of polyhedral uncertainties. After a discussion of deterministic problem formulation, we present in detail a min-max decomposition approach. The framework iteratively solves a decision maker's problem that evaluates a limited number of scenarios to generate a production plan, and an adversarial problem that generates a scenario that has not yet been considered by the decision maker using the proposed production plan. As the computational challenge of this framework primarily lies in the decision maker's problem, we then investigate this problem further in order to improve computational performance. In particular, we propose a novel approach for formulating the robust lot sizing problem with remanufacturing, which employs two different reformulations. As detailed computational results demonstrate, these extended reformulations are capable of improving the computational performance immensely, in particular the case where setup costs are high. We also present a thorough understanding on the impact of a range of problem parameters, which we believe are invaluable to researchers not only in the area of lot-sizing but also in the broader community of robust optimization. In near future, we would like to address the complexity issues of the adversarial problem. We would like to address a few cost structures that we have not considered in this work. For instance we would like to introduce a variable costs component for our manufacturing costs and make the costs time variant.

# Acknowledgements

The second author was supported by the *Center for Research and Development in Mathematics and Applications (CIDMA)* through the Portuguese Foundation for Science and Technology (FCT - Fundação para a Ciência e a Tecnologia), references UIDB/04106/2020 and UIDP/04106/2020. The third and fourth authors were supported by US Air Force Office of Scientific Research (Grant number FA9550-17-1-0105).

# References

[1] A. Agra and M. Constantino. Lotsizing with backlogging and start-ups: The case of wagner-whitin costs. *Operations Research Letters*, 25(2):81–88, 1999.

[2] Agostinho Agra, Marcio Costa Santos, Dritan Nace, and Michael Poss. A dynamic programming approach for a class of robust optimization problems. *SIAM Journal on Optimization*, 26(3):1799–1823, 2016.

[3] Kerem Akartunalı and Ashwin Arulselvan. Economic lot-sizing problem with remanufacturing option: complexity and algorithms. In *International Workshop on Machine Learning, Optimization, and Big Data*, pages 132–143. Springer, 2016.

[4] Kerem Akartunalı and Andrew J Miller. A heuristic approach for big bucket multi-level production planning problems. *European Journal of Operational Research*, 193(2):396–411, 2009.

[5] Ayse Akbalik and Yves Pochet. Valid inequalities for the single-item capacitated lot sizing problem with step-wise costs. *European Journal of Operational Research*, 198(2):412–434, 2009.

[6] Öykü Naz Attila, Agostinho Agra, Kerem Akartunalı, and Ashwin Arulselvan. A decomposition algorithm for robust lot sizing problem with remanufacturing option. In *International Conference on Computational Science and Its Applications*, pages 684–695. Springer, 2017.

[7] M Fazle Baki, Ben A Chaouch, and Walid Abdul-Kader. A heuristic solution procedure for the dynamic lot sizing problem with remanufacturing and product recovery. *Computers & Operations Research*, 43:225 – 236, 2014.

[8] Imre Barany, Tony J Van Roy, and Laurence A Wolsey. Strong formulations for multi-item capacitated lot sizing. *Management Science*, 30(10):1255–1261, 1984.

[9] Aharon Ben-Tal, Boaz Golany, Arkadi Nemirovski, and Jean Philippe Vial. Retailer-supplier flexible commitments contracts: A robust optimization approach. *Manufacturing & Service Operations Management*, 7(3):248–271, 2005.

[10] Aharon Ben-Tal and Arkadi Nemirovski. Robust optimization — methodology and applications. *Mathematical Programming*, 92:453—-480, 2002.

[11] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[12] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[13] Dimitris Bertsimas and Aurélie Thiele. A robust optimization approach to inventory theory. *Operations Research*, 54(1):150–168, 2006.

[14] Daniel Bienstock and Nuri Özbay. Computing robust basestock levels. *Discrete Optimization*, 5(2):389–414, 2008.

[15] N. Brahimi, N. Absi, S. Dauzère-Pérès, and A. Nordli. Single-item dynamic lot-sizing problems: An updated survey. *European Journal of Operational Research*, 263(3):838–863, 2017.

[16] Mahdi Doostmohammadi and Kerem Akartunalı. Valid inequalities for two-period relaxations of big-bucket lot-sizing problems: Zero setup case. *European Journal of Operational Research*, 267(1):86 – 95, 2018.

[17] Gary D Eppen and R Kipp Martin. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*, 35(6):832–848, 1987.

[18] Moritz Fleischmann, Jacqueline M Bloemhof-Ruwaard, Rommert Dekker, Erwin Van der Laan, Jo AEE Van Nunen, and Luk N Van Wassenhove. Quantitative models for reverse logistics: A review. *European Journal of Operational Research*, 103(1):1–17, 1997.

[19] Boaz Golany, Jian Yang, and Gang Yu. Economic lot-sizing with remanufacturing options. *IIE Transactions*, 33(11):995–1004, 2001.

[20] Bram L Gorissen, İhsan Yanıkoğlu, and Dick den Hertog. A practical guide to robust optimization. *Omega*, 53:124 – 137, 2015.

[21] Yongpei Guan and Andrew J Miller. Polynomial-time algorithms for stochastic uncapacitated lot-sizing problems. *Operations Research*, 56:1172–1183, 2008.

[22] Winifred L Ijomah. Addressing decision making for remanufacturing operations and design-for-remanufacture. *International Journal of Sustainable Engineering*, 2(2):91–102, 2009.

[23] R. Jans and Z. Degraeve. Improved lower bounds for the capacitated lot sizing problem with setup times. *Operations Research Letters*, 32:185–195, 2004.

[24] Onur A Kilic and Wilco van den Heuvel. Economic lot sizing with remanufacturing: structural properties and polynomial-time heuristics. *IISE Transactions*, pages 1–14, 2019.

[25] Jakob Krarup and Ole Bilde. Plant location, set covering and economic lot size: An 0 (mn)-algorithm for structured problems. In *Numerische Methoden bei Optimierungsaufgaben Band 3*, pages 155–180. Springer, 1977.

[26] Simge Küçükyavuz and Yves Pochet. Uncapacitated lot sizing with backlogging: the convex hull. *Mathematical Programming*, 118(1):151–175, 2009.

[27] Jianzhi Li, Miguel González, and Yun Zhu. A hybrid simulation optimization method for production planning of dedicated remanufacturing. *International Journal of Production Economics*, 117(2):286–301, 2009.

[28] Zhendong Pan, Jiafu Tang, and Ou Liu. Capacitated dynamic lot sizing problems in closed-loop supply chain. *European Journal of Operational Research*, 198(3):810–821, 2009.

[29] Pedro Pineyro and Omar Viera. The economic lot-sizing problem with remanufacturing and one-way substitution. *International Journal of Production Economics*, 124:482–488, 2010.

[30] Yves Pochet and Laurence A Wolsey. *Production planning by mixed integer programming*. Springer, 2006.

[31] Mathijn J Retel Helmrich, Raf Jans, Wilco van den Heuvel, and Albert PM Wagelmans. Economic lot-sizing with remanufacturing: complexity and efficient formulations. *IIE Transactions*, 46(1):67–86, 2014.

[32] A. Sifaleras and I. Konstantaras. Variable neighborhood descent heuristic for solving reverse logistics multi-item dynamic lot-sizing problems. *Computers & Operations Research*, 78:385 – 392, 2017.

[33] H. Stadtler. Multilevel lot sizing with setup times and multiple constrained resources: Internally rolling schedules with lot-sizing windows. *Operations Research*, 51(3):487–502, 2003.

[34] Sharifah Aishah Syed Ali, Mahdi Doostmohammadi, Kerem Akartunalı, and Robert van der Meer. A theoretical and computational analysis of lot-sizing in remanufacturing with separate setups. *International Journal of Production Economics*, 203:276 – 285, 2018.

[35] Ruud H Teunter, Z Pelin Bayindir, and Wilco Van Den Heuvel. Dynamic lot sizing with product returns and remanufacturing. *International Journal of Production Research*, 44(20):4377–4400, 2006.

[36] Mathieu Van Vyve and Laurence A. Wolsey. Approximate extended formulations. *Mathematical Programming*, 105(2-3):501–522, 2006.

[37] C. Wei, Y. Li, and X. Cai. Robust optimal policies of production and inventory with uncertain returns and demand. *International Journal of Production Economics*, 134(2):357—-367, 2011.

[38] Tao Wu, Zhe Liang, and Canrong Zhang. Analytics branching and selection for the capacitated multi-item lot sizing problem with nonidentical machines. *INFORMS Journal on Computing*, 30(2):236–258, 2018.

[39] Jian Yang, Boaz Golany, and Gang Yu. A concave-cost production planning problem with remanufacturing options. *Naval Research Logistics*, 52(5):443–458, 2005.

[40] B. Zeng and L. Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461, 2013.