CIRRELT

CIRRELT-2025-11

An Integrated Lot Sizing and Maintenance Planning Problem for a Single Machine

Harcènage Dansou
Matthieu Gruson
François Lamothe
Julien Legavre

April 2025

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

An Integrated Lot Sizing and Maintenance Planning Problem for a Single Machine

Harcènage Dansou[1,2], Matthieu Gruson[1,2,*], François Lamothe[2,3], Julien Legavre[2,4]

1. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
2. Département d'analytique, opérations et technologies de l'information (AOTI), UQAM
3. LAAS-CNRS, Toulouse, France
4. Ecole Nationale de l'Aviation Civile, Toulouse, France

Abstract. This study proposes new mathematical formulations as mixed-integer linear programs for an integrated Capacitated Lot Sizing and Preventive Maintenance problem. We reformulate the maintenance decision variables that are incorporated into the classical and facility location lot sizing formulations. This reformulation of maintenance decisions is called extended maintenance formulation in our study. Also, we apply a Dantzig-Wolfe (DW) decomposition to these models by using maintenance patterns. We demonstrate theoretically that the extended maintenance formulation gives a relaxation equivalent to the relaxation of the DW decomposition model. Subsequently, we use a relax-and-fix heuristic combined with fix-and-optimize to solve the original problem. Relax-and-fix is used to build an initial solution and fix-and-optimize allows us to improve the built solution. Computational experiments evaluate the performance of the heuristics and the models solved with Gurobi on the generated instances. The results show that the solutions obtained by the heuristics are close to those obtained by the models in terms of gap and in a reduced time.

Keywords: lot sizing, preventive maintenance, relax-and-fix, fix-and-optimize, mathematical modelling, Dantzig-Wolfe decomposition

Acknowledgements. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Discovery Grants 2021-03327 and 329202. This support is gratefully acknowledged.

---

* Corresponding author: gruson.matthieu@uqam.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2025

1. Introduction

Production planning is a very important activity for manufacturing companies. Thus, they invest numerous efforts into effective production planning. Operations researchers have proposed several models and solution approaches, discussing the efficiency of production planning in manufacturing companies. One of the most studied problem related to production planning is the Capacitated Lot Sizing Problem (CLSP). The CLSP consists in determining the production timing and level over a discrete and finite time horizon while minimizing operational costs. The operational costs are production costs, setup costs, and holding costs.

According to Aghezzaf et al. [3], production planning assumes the system will function at its maximum performance during the planning horizon. But, in practice, it happens that there is a sudden interruption of the production system, which stops production. In this case, maintenance operations are carried out to restore the broken machine. It is therefore necessary to plan preventive maintenance operations which help to avoid sudden interruptions Kojchen and Monchy [21]. Preventive maintenance includes preplanned and scheduled adjustments, major overhauls, inspections, and lubrications, to maintain equipments and facilities in a condition where breakdowns and the need for emergency repairs are minimized Ashayeri et al. [8].

Often, maintenance planning is carried out before production planning. This leads to suboptimal production plans, in terms of cost, since maintenance decisions are constraints for production planning (e.g., the capacity is impacted by maintenance decisions). It is therefore of interest to make a synchronization between both plannings to decrease the costs generated by the production interruptions. According to Ashayeri et al. [8], this approach has not receive enough management attention. Their hypothesis for the lack of management attention is the belief that maintenance costs cannot be controlled. In the last years, few studies have addressed combined maintenance and lot sizing problems despite the interest for companies and the potential benefits. For instance, Schreiber et al. [31] show that integrated production and maintenance planning significantly reduces machine downtime and increases the production pace.

The purpose of this paper is to address this area of research by combining maintenance planning and lot sizing problem. We take as starting point the study of Shamsaei and Van Vyve [32] and include maintenance decisions into the CLSP. Unlike the classical CLSP, we consider that the capacity available in each time period is not a parameter, but instead a decision variable linked to the maintenance decisions: when a maintenance is carried out, the capacity goes back to a defined maximum value, whereas capacity decreases from one period to the next one if no maintenance is carried out. The decrease of capacity follows a degradation function which is an input in our study. We further consider non-cyclical preventive maintenance, which is the most studied for planning maintenance operations Fitouhi and Nourelfath [15], Aghezzaf et al. [3], Shamsaei and Van Vyve [32]. Shamsaei and Van Vyve [32] state that non-cyclical maintenance reduces the total costs compared to cyclical maintenance in their study. Cyclic maintenance is scheduled periodically, unlike non-cyclical maintenance, which allows maintenance to be carried out at any time. We call our resulting problem the integrated capacitated lot sizing and maintenance planning problem (CLSPM).

The contribution of our study is threefold. First, we propose a new way of defining maintenance decision variables compared to the literature. The advantage of our new variables is to keep information about the last

maintenance performed over the time horizon. This allows us to consider several degradation functions for the machine. On the modeling side, we further consider that the degradation may happen if and only if production occurs. Second, we develop a relax-and-fix with a fix-and-optimize approach to solve this problem. The CLSP itself is an NP-hard problem and integrating maintenance decisions makes the problem harder. We therefore tackle the problem heuristically. We build an initial solution using a relax-and-fix heuristic. We then use a fix-and-optimize approach to improve the solution obtained after the relax-and-fix process. Third, we perform a thorough numerical campaign. This campaign is divided into two main sets of experiments. The first one, on MIP models, to analyze the impact of the different degradation functions and to highlight the possibilities offered by our new maintenance variables. The second one, on the heuristics, to assess its performance compared to the MIP models.

The remainder of this paper is organized as follows. In Section 2, we discuss the major studies addressing synchronization between production planning and maintenance planning. In Section 3, we present in detail the mathematical formulations for our problem, including a new way to define the maintenance decision variables. Section 4 shows the relax-and-fix with fix-and-optimize algorithms designed to heuristically solve the problem. Computational results are presented in Section 5. Finally, concluding remarks and future work are given in Section 6.

2. Literature review

This section reviews the literature related to our study. We start by reviewing the literature on the integration of maintenance into lot sizing problems. We focus on the solution method used as well as on the way machine degradation is taken into account. We then review the use of relax-and-fix and fix-and-optimize approaches to solve lot sizing problems.

2.1. Maintenance and lot sizing

Maintenance operations are a tactical decision that affect production planning and, in turn, the fulfillment of demand. This involves determining the appropriate strategy to keep production machinery in good condition, so as production can take place. Consequently, research has been conducted in this area, illustrating several ways to integrate maintenance operations into lot-sizing problems. In this section we take a look at different maintenance features that have been addressed in lot sizing problems, along with solution methods used and the way machine degradation is taken into account. Note that this section is not intended to be exhaustive. The interested reader is referred to the reviews of Garg and Deshmukh [18] and De Jonge and Scarf [12].

2.1.1. Maintenance features

Several main features of maintenance have been taken into account in the literature. The first one is cyclic and non-cyclic maintenance strategies. In a cyclic strategy, maintenance follows a specific pattern. On the contrary, a non-cyclic strategy allows maintenance operations to occur at any time. Non-cyclic strategies give more freedom and therefore lead to harder problems to solve. The study of such strategies has been addressed in Aghezzaf

et al. [3], Aghezzaf and Najid [2], Fitouhi and Nourelfath [15, 16], Aghezzaf et al. [4] and Shamsaei and Van Vyve [32].

Another studied feature of maintenance is the case of imperfect maintenance operations. An imperfect preventive maintenance operation fails in putting back the system in an as good as new state. Imperfect maintenance operations have been considered in Levitin and Lisnianski [25], El-Ferik [14], Le Tam et al. [24] and Wang et al. [36].

Other features include the consideration of multi-level production systems (Aghezzaf and Najid [2]), parallel production lines or multiple machines (Aramon Bajestani et al. [6], Zhu [40]), maximum time before a maintenance action must occur (Lu et al. [27]), the study of specific maintenance policies (Gao et al. [17]), maintenance time-windows (Najid et al. [28], Alaoui-Selsouli et al. [5]), or the moment when maintenance can be performed during a time period (Absi et al. [1]).

2.1.2. Machine degradation

An important aspect of the links between maintenance and lot sizing problems lie in the way capacity is affected by both production and maintenance decisions. In that respect, different machine degradation modeling have been used in the literature. It includes an exponential degradation, where the capacity available at a given period is equal to a proportion of the capacity available in the previous period, unless a maintenance activity has been performed. Such modeling is used in Aghezzaf et al. [3], Shamsaei and Van Vyve [32]

Another used method to model machine degradation is the use of failure rates. In such cases, it is assumed that a specific probability failure density function is known in advance. The distribution used are usually Weibull or Gamma distributions, to obtain an analytical expression of the capacity reduction due to failure of some components. Weibull distribution is the most used distribution in reliability engineering because it provides a good fit with data in many applications and arises naturally in theory Arts [7]. Failure rates have been used in Aghezzaf et al. [3], Aghezzaf and Najid [2], Najid et al. [28], Fitouhi and Nourelfath [15], Alaoui-Selsouli et al. [5], Lu et al. [27], Aghezzaf et al. [4], Le Tam et al. [24], Wang et al. [36], Gao et al. [17], and Zhu [40]. Any failure that happens reduces the capacity of the machine.

Machine degradation can be also taken into account not using capacity degradation, but age reduction. In this case, the machine or the components of the machine ages and it increases a given failure probability, unless preventive maintenance is performed. Age reduction has been tackled in Levitin and Lisnianski [25], El-Ferik [14] and Absi et al. [1].

Another widely used way of modeling machine degradation is to take into account different machine states. This can be included in a Markov scheme, such as in Sloan and Shanthikumar [33] and Aramon Bajestani et al. [6], or with regular probabilities as in Fitouhi and Nourelfath [16].

Note that the integration of maintenance planning into scheduling problems has also received quite some attention in the operation research literature. The interested reader is referred to Sortrakul et al. [34], Yang et al. [38], Chaabane et al. [10] and Babaeimorad et al. [9] for specific problems, and to Geurtsen et al. [19] who provide a detailed review of both the integration of maintenance and resources within a production process.

2.1.3. Solution methods

Solving integrated maintenance and lot sizing problems is not an easy task. Several solution methods have been used. Those include modeling the problems as MILP and using solver to solve them. This is the case is Sloan and Shanthikumar [33], Aghezzaf et al. [3], Najid et al. [28], Aghezzaf et al. [4], Shamsaei and Van Vyve [32] and Zhu [40]. MILP solvers are also used in an iterative way, for instance in Fitouhi and Nourelfath [15]. Note that mathematical modeling is also exploited to obtain optimality conditions on the solution. This is the case in Sloan and Shanthikumar [33], El-Ferik [14] and Gao et al. [17]. Sloan and Shanthikumar [33] further embed the optimality conditions into a simulation framework.

Such integrated problems have also been tackled using heuristics and decomposition methods. In particular, Aghezzaf and Najid [2] and Lu et al. [27] use lagrangian relaxation, Aghezzaf et al. [4] and Alaoui-Selsouli et al. [5] use a fix-and-optimize heuristic, and Le Tam et al. [24] use a combined relax-and-fix and fix-and-optimize approach.

Other methods have also been used, but more rarely. We can mention here the use of genetic algorithm by Levitin and Lisnianski [25] and Wang et al. [36], the use of simulated annealing by Fitouhi and Nourelfath [16], the use of a Markov decision process by Aramon Bajestani et al. [6], or more recently the proposition of dynamic programming approaches by Absi et al. [1].

Table 1 summarizes the works mentioned in this section, focusing on maintenance features, solution methods, and machine degradation. When machine degradation is considered through failure rates, we specify the distribution used in the numerical experiments.

2.2. Relax-and-Fix/Fix-and-Optimize and lot sizing

Relax-and-fix (RF) and fix-and-optimize (FO) are successful approaches developed to tackle the complexity of the capacitated lot-sizing problem [30]. Instead of solving the whole formulation, the RF heuristic reduces the complexity of the problem by solving a series of partially relaxed mixed-integer programming models (Roshani et al. [29]). The fix-and-optimize (FO) heuristic uses the same principle as relax-and-fix but without any relaxed variables.

As mentioned in Sahling et al. [30], one important feature of both RF and FO are the ways subsets of variables are built. Several strategies have been used in the literature. The most natural one is to use temporal subsets, based on the periods (see, e.g., Helber and Sahling [20], Lang and Shen [23], Roshani et al. [29] and You et al. [39]). It is also possible to build sets based on the different items (Sahling et al. [30], Lang and Shen [23] and You et al. [39]), or on the possibility to substitute items (Lang and Shen [23]). More recently, a value-wise approach has been proposed in Toledo et al. [35], based on the values of the relaxed variables. Other building possibilities are more problem specific. For instance, the sets may be built based on the different families of items, on the production processes (Sahling et al. [30] and Helber and Sahling [20]), on the resources used (Sahling et al. [30] and Helber and Sahling [20]), or a combination of those (Chen [11] and Li et al. [26]). Note that for the temporal and item-based approaches, the selection of periods or items can be done according to the data input, or can be random as in You et al. [39]. In our case, we tried several strategies but just kept the temporal one, based on preliminary experiments.

Table 1: Summary of papers related to integrated maintenance and lot sizing problems

Reference	Maintenance feature	Solution method	Machine degradation
Aghezzaf et al. [3]	Cyclic and non-cyclic	MILP	Failure rate
Aghezzaf and Najid [2]	Multi-level system	Lagrange relaxation	Failure rate
Fitouhi and Nourelfath [15]	Non-cyclic	MILP	Failure rate (Weibull)
Shamsaei and Van Vyve [32]	Cyclic and non-cyclic	MILP	Exponential capacity reduction
Levitin and Lisnianski [25]	Imperfect maintenance	Genetic algorithm	Age reduction
El-Ferik [14]	Imperfect maintenance	Simulation	Age reduction (Weibull)
Aghezzaf et al. [4]	Imperfect maintenance	MILP and fix-and-optimize	Failure rate (Gamma)
Le Tam et al. [24]	Imperfect maintenance	relax-and-fix and fix-and-optimize	Failure rate (Gamma)
Sloan and Shanthikumar [33]	-	MILP	Machine state (Markov)
Najid et al. [28]	Time-window	MILP	Failure rate
Alaoui-Selsouli et al. [5]	-	Lagrangian	Failure rate (Weibull)
Lu et al. [27]	Max time before maintenance	Lagragian heuristic	Failure rate (Weibull)
Aramon Bajestani et al. [6]	Multi-machine	Markov decision process and MILP	Machine state (Markov)
Fitouhi and Nourelfath [16]	Non-cyclic	Simulated annealing	Machine state
Wang et al. [36]	Imperfect maintenance	Genetic algorithm	Failure rate (exponential)
Gao et al. [17]	Maintenance policies	Mathematical modeling	Failure rate (exponential and Weibull)
Absi et al. [1]	Maintenance timing	Dynamic programming	Age reduction
Zhu [40]	Parallel lines	MILP	failure rate (Weibull)
Our study	Non-cyclic	MILP and Heuristic	Diverse capacity reductions

Both RF and FO have been applied successfully to different lot sizing problems. The RF heuristic has been applied to a CLSP with remanufacturing (Roshani et al. [29]). The FO heuristic has been applied to the multi-level CLSP (Toledo et al. [35]), possibly with positive lead times (Helber and Sahling [20] and Chen [11]), with setup carry-overs (Sahling et al. [30]) and even in a stochastic context (Li et al. [26]), or in a time-varying environment (You et al. [39]). In a time-varying environment, costs and production capacity are dynamic through time. The FO heuristic has also been applied to the CLSP with sequence-dependent setups and substitutions (Lang and Shen [23]), or to a variant with parallel machines, backlogging and time windows (Xiao et al. [37]).

The RF and FO heuristics have also been applied jointly in a two-stage manner, as in Lang and Shen [23] and Toledo et al. [35]. They can further be used alone, several times, or in combination with other approaches. When used several times, it is possible to use different set building strategies as in Sahling et al. [30]. When used in a combination with other approaches, it is often used with variable neighborhood search methods, as in Toledo et al. [35], Chen [11] and Li et al. [26].

Xiao et al. [37] introduce MIP-based fix-and-optimize algorithms for the parallel machine capacitated lot-sizing and scheduling problem. The authors introduce a local search phase and the neighborhood modification phase in the FO heuristic. They also show that the performance of the proposed algorithms is satisfactory for real-world instances.

3. The integrated capacitated lot sizing and maintenance problem

In this section, we first formally describe the integrated capacitated lot sizing and maintenance problem (CLSPM). Next, we present the mathematical models of lot sizing and how maintenance operations are integrated into these models. In this respect, we introduce the new models by reformulating the maintenance decision variables to keep more information about maintenance operations.

3.1. Problem description

We consider a set I of items that need to be produced over a planning horizon. The planning horizon T is finite and divided into several periods. Production is carried out with a single machine which is limited by capacity. At each period t, a demand d_{it} for each item i must be satisfied. This demand is dynamic and deterministic, i.e., it varies from one period to the next one but they are known at the start of the planning horizon.

The production process implies the degradation of the machine that we model through a reduction of machine capacity. We consider different degradation function that allows us to define the capacity available in each time period, depending on the maintenance decisions, setup decisions, and the capacity available in the previous period.

The production of each item i in period t requires setup on the production machine which generates fixed setup costs sc_{it}. The production of each item generates a variable holding cost per item-unit and per period, h_{it}. Finally, carrying out a maintenance operation in period t incurs a fixed cost mc_t. The objective of the problem is to minimize the sum of all these setup, holding and maintenance costs.

In each time period, we must decide on the production quantities for each item. If we produce, this implies a setup for production. We also decide on the inventory that we carry from one period to the next one. Finally, we decide on the maintenance operations.

3.2. Mathematical formulation

In this section we will show two mathematical formulations for the problem. The first one differs from the work of Shamsaei and Van Vyve [32] in two ways. First, they consider binary decision variables for the maintenance decisions, indicating if there is a maintenance operation performed in period t. We introduce a new way of modeling maintenance decisions. We define z_{kt} as a binary variables that indicates if, in period t, the last time a maintenance was performed was in period k. Such definition allows us to track maintenance decisions, and consider different degradation function. Second, for the lot sizing part, Shamsaei and Van Vyve [32] use the classical lot sizing variables. We instead use a transportation reformulation, which is shown in the literature to provide a stronger LP relaxation than the classical formulation (Krarup and Bilde [22]). The second formulation we will show is a Dantzig-Wolfe reformulation of the first one.

Following the problem description, Table 2 summarizes all the sets, parameters and decision variables used in the mathematical models.

Table 2: Notation used in the models

Sets	Definition
T	Set of periods in the planning horizon, indexed by t
I	Set of items, indexed by i
Parameters	Definition
sc_{it}	Fixed production setup cost for item $i \in I$ in period $t \in T$
hc_{it}	Holding cost per unit of item $i \in I$ hold in the inventory at the end of period $t \in T$
mc_t	Fixed maintenance cost if a maintenance is performed in period $t \in T$
d_{it}	Demand for item $i \in I$ in period $t \in T$
C_{max}	Maximum (nominal) capacity of the machine
Variables	Definition
y_{it}	Binary variable equal to one if item $i \in I$ is produced in the period $t \in T$, and zero otherwise
z_{kt}	Binary variable equal to one if the last maintenance before period $t \in T$ was performed in period $k \leq t$, and zero otherwise
x_{ikt}	Production quantity of item $i \in I$ produced in period $k \in T$ to satisfy the demand of period $t \in T$, with $k \leq t$
c_t	Available capacity of the machine in period $t \in T$

The CLSPM can then be modelled as follows:

$$(FL-EM) \min_{x,y,c,z} \sum_{i\in I}\sum_{t\in T}\left(\sum_{k=1}^{t}\left(\sum_{j=k}^{t-1} hc_{ij}\right)x_{ikt}\right) + \sum_{i\in P}\sum_{t\in T} sc_{it}y_{it} + \sum_{t\in T} mc_t z_{tt} \tag{1}$$

$$\text{s.t.} \sum_{k=1}^{t} x_{ikt} = d_{it} \qquad \forall\, i \in I,\, t \in T \tag{2}$$

$$x_{ikt} \leq d_{it} y_{ik} \qquad \forall\, i \in I,\ k \in T,\ t \in T,\ t \geq k \tag{3}$$

$$\sum_{i \in I} \sum_{k=t}^{|T|} x_{itk} \leq c_t \qquad \forall\, t \in T \tag{4}$$

$$c_t = f(z) \qquad \forall\, t \in T \tag{5}$$

$$c_1 \leq C_{max} z_{11} \tag{6}$$

$$z_{kt} \leq z_{kk} \qquad \forall t \in T, k \leq t \tag{7}$$

$$\sum_{k=1}^{t} z_{kt} \leq 1 \qquad \forall t \in T \tag{8}$$

$$z_{kt} \in \{0,1\} \qquad \forall\, k \in T,\ t \in T, k \leq t \tag{9}$$

$$c_t \geq 0 \qquad \forall\, t \in T \tag{10}$$

$$x_{ikt} \geq 0 \qquad \forall\, i \in I,\ k \in T,\ t \in T,\ t \geq k \tag{11}$$

$$y_{it} \in \{0,1\} \qquad \forall\, i \in P,\ t \in T. \tag{12}$$

The objective function (1) minimizes the total costs including setup, inventory and maintenance costs. Based on the defined maintenance variables, the overall maintenance cost is expressed by $\sum_{t \in T} mC_t z_{tt}$, since if z_{tt} is equal to one, the maintenance operation is performed at period t. Constraints (2) express the satisfaction of the demand for each item at each period. Constraints (3) are the setup forcing constraints. Constraints (4) define the available capacity of the machine for each time period. Constraints (5) define the evolution of the available capacity depending on the degradation function f. Constraint (6) links the capacity and maintenance actions for the first period, so as we pay a maintenance cost to initiate production. Constraints (7) ensure that if the last maintenance was done in period $k, k < t$, then there must be a maintenance action in period k. This enables linking all the z_{kt} variables to the maintenance cost since only the z_{kk} variables appear in the objective function. Constraints (8) state that for each period $t \in T$ there is only one period for which the last maintenance has been performed before. Finally, constraints (9)-(12) define the domains of the decision variables. Note that the mathematical model presented above considers machine degradation over time regardless of if production occurs.

3.3. Capacity degradation functions

We consider three possible capacity degradation functions: exponential, linear, and dirac. Those three possibilities are described in the following subsections.

3.3.1. Exponential degradation

We consider that the capacity follows an exponential degradation. We define the capacity reduction coefficient α between 0 and 1 that allows us to represent the level of the machine degradation. In other words, without any maintenance operation carried out, the capacity in period $t+1$ is equal to α times the capacity in period t. A high capacity reduction factor means that the level of machine degradation is low, whereas a low value indicates higher degradation. In this case, the degradation function is $f(z) = C_{max} \sum_{k=1}^{t} \alpha^{t-k} z_{kt}$.

3.3.2. Linear degradation

We consider that the capacity follows a linear degradation. We define a capacity reduction value β that is a proportion of the maximum available capacity C_{max}. In this case, the degradation function is $f(z) = C_{max} - \beta C_{max} \sum_{k=1}^{t}(t-k)z_{kt}$.

3.3.3. Dirac degradation

We consider here that the capacity stays the same as long as the last maintenance action occurred no more than γ periods before the current period. If the last maintenance occurred strictly more than γ periods before the current period, then the available capacity is set to a low value, that we denote as C_{low}. In this case, the degradation function is $f(z) = C_{max} \sum_{k=t-\gamma}^{t} z_{kt} + C_{low} \sum_{k=1}^{t-\gamma-1} z_{kt}$. Such kind of degradation function highlights the benefits of having maintenance variables that track the last maintenance period.

3.4. Dantzig-Wolfe reformulation

We can apply a Dantzig-Wolfe reformulation on (1)-(12). The Dantzig-Wolfe (DW) [13] decomposition allows a mathematical program to be expressed as a set of independent sub-problems that are easier to solve. It also identifies a set of constraints as "coupling" constraints. We propose a DW decomposition based on the previous model by defining the maintenance decisions as maintenance plans over the planning horizon. A maintenance plan represents a set of maintenance decisions that define the periods during which maintenance is performed or not. We define Ω as the set of all possible maintenance plans. Let M_j be the overall cost of maintenance plan $j \in \Omega$ and let C_t^j be the available capacity during period $t \in T$ in maintenance plan $j \in \Omega$. Note that for the sake of simplicity, we do not index the parameter C_t^j with the degradation function f, even if the degradation function has a direct impact on the value of those parameters. Let w_j be a binary variable that takes the value 1 iff maintenance plan $j \in \Omega$ is chosen, and 0 otherwise. The DW reformulation is as follows:

$$(FL - M_{DW}) \quad \min_{x,y,c,w} \sum_{i\in P}\sum_{t\in T}\left(\sum_{k=1}^{t}\left(\sum_{j=k}^{t-1} hc_{ij}\right)x_{ikt}\right) + \sum_{i\in P}\sum_{t\in T} f_{it}y_{it} + \sum_{j\in\Omega} M_j w_j \tag{13}$$

$$\text{s.t. } (2)-(4), (10)-(12)$$

$$\sum_{j\in\Omega} w_j = 1 \tag{14}$$

$$c_t = \sum_{j\in\Omega} C_t^j w_j \quad \forall\, t \in T \tag{15}$$

$$w_j \in \{0,1\} \quad \forall\, j \in \Omega. \tag{16}$$

Constraint (14) forces the selection of a single maintenance plan. Machine's capacity at each period is expressed with constraints (15). This is equivalent to constraints (5). Constraints (16) represent the domain of the variables. The literature indicates that a Dantzig-Wolfe can only improve the linear relaxation. This is however not the case here.

Proposition 1. *Let V_1 be the value of the linear relaxation of $(FL-EM)$ formulation and V_2 be the value of the linear relaxation of $(FL-M_{DW})$ formulation, then $V_1 = V_2$.*

Proof. See Appendix A □

Note that despite this theoretical result we tried to solve the linear relaxation of $FL-M_{DW}$ via column generation. This however did not prove successful, in terms of CPU time, compared to a direct resolution of $FL-EM$.

3.5. Impact of production on capacity degradation

One natural extension of the CLSPM is to consider that capacity decreases iff there is production that takes place. We call this variant CLSPM-P. To that end, let Y_t be a variable that takes the value one if any production occurs in period $t \in T$, and zero otherwise. The resulting formulation is as follows:

$$(FL-EM) \quad \min_{x,y,c,z,Y} \sum_{i\in I}\sum_{t\in T}\left(\sum_{k=1}^{t}\left(\sum_{j=k}^{t-1} hc_{ij}\right)x_{ikt}\right) + \sum_{i\in P}\sum_{t\in T} sc_{it}y_{it} + \sum_{t\in T} mc_t z_{tt} \tag{17}$$

s.t. $(2)-(4), (10)-(12)$

$$c_{t+1} \leq f(c_t, z) + C_{max}z_{t+1,t+1} + C_{max}(1-Y_t) \qquad \forall\, t \in T \tag{18}$$

$$c_{t+1} \leq c_t + C_{max}z_{t+1,t+1} + C_{max}Y_t \qquad \forall\, t \in T \tag{19}$$

$$c_t \leq C_{max} \qquad \forall\, t \in T \tag{20}$$

$$y_{it} \leq Y_t \qquad \forall\, i \in P,\ t \in T \tag{21}$$

$$Y_t \leq \sum_{i\in P} y_{it} \qquad \forall\, i \in P,\ t \in T \tag{22}$$

$$Y_t \in \{0;1\} \qquad \forall\, t \in T. \tag{23}$$

Constraints (18)-(19) define the evolution of the available capacity depending on the degradation function f, on the maintenance action, and on the setup decisions Y. Constraints (21) indicate if any production took place in a given period. Constraints (22) forces the Y variables to be equal to 0 if no production occurs. Such constraints are necessary, compared to similar ones for the setup variables, since the Y variables do not appear in the objective function. Finally, constraints (23) define the domains of the decision variables. In such case, the degradation function f is as follows:

$$f(c_t, z) = \begin{cases} \alpha c_t & \text{if the degradation is exponential} \\ c_t - \beta & \text{if the degradation is linear} \\ C_{low} + (C_{max} - C_{low}) \sum_{k=t+1-\gamma}^{t} z_{k,t+1} & \text{if the degradation is dirac} \end{cases}$$

4. Solving the CLSPM via a relax-and-fix and fix-and-optimize heuristic

In this section we present a heuristic we design to solve the CLSPM. The heuristic decomposes into two heuristics: relax-and-fix (RF) and fix-and-optimize (FO). The relax-and-fix heuristic is used to build an initial solution while the fix-and-optimize heuristic is used to improve the solution obtained by using RF.

4.1. Relax-and-fix

The relax-and-fix heuristic (see Section 2.2) is an approach that has proven effective in addressing diverse lot sizing problems. It is an iterative construction heuristic that allows us to obtain a solution to a given problem by solving several small mixed-integer linear problems (MILP). In each iteration, the set of binary variables is divided into three disjoint sets: variables fixed, variables to be optimized, and relaxed variables. We define those sets through the time periods where the binary variables are fixed, optimize or relaxed. Those sets are denoted by $\mathcal{F}, \mathcal{O}$ and $\mathcal{R}$, respectively.

Two key elements for the success of the relax-and-fix approach are the size of the sets $\mathcal{F}, \mathcal{O}$ and $\mathcal{R}$, and the strategy used to build the sets. In our case, we used a temporal approach, meaning that we follow the time horizon to build the sets $\mathcal{F}, \mathcal{O}$ and $\mathcal{R}$. Such approach has proven successful in the past (see, e.g., Helber and Sahling [20]). In other words, let t^- and t^+ denote two time periods, with $t^- < t^+$. We set $\mathcal{F} = \{t | t \leq t^-\}$, $\mathcal{O} = \{t | t^- < t \leq t^+\}$, and $\mathcal{R} = \{t | t > t^+\}$.

We further define $\kappa = t^+ - t^-$ as the size of the interval, in terms of time periods, of variables to be optimized and Δ as the size of the interval, in terms of time periods, for which the binary variables are fixed to a certain value after a specific iteration. Coincidentally, Δ is the number of additional periods whose corresponding variables are fixed at the subsequent iterations. The relax-and-fix heuristic is described in Algorithm 1.

Algorithm 1 Relax-and-fix heuristic (RF)

1: Initialization: $t^- = 0$, $t^+ = \kappa$
2: Build sets $\mathcal{F}, \mathcal{O}$ and $\mathcal{R}$
3: **while** all variables are not binary **do**
4: Solve the resulting mixed-integer problem taking into account $\mathcal{F}, \mathcal{O}$ and $\mathcal{R}$
5: $t^- = t^- + \Delta$, $t^+ = \min\{|T|; t^+ + \Delta\}$
6: Update sets $\mathcal{F}, \mathcal{O}$ and $\mathcal{R}$
7: **end while**

Note that in our case, the RF heuristic did not provide infeasible solutions.

4.2. Fix-and-optimize

The feasible solution built by the RF heuristic will be used as initial solution for the FO heuristic. The FO heuristic (see, e.g., Toledo et al. [35]) is similar to RF, where several MILP problems are solved iteratively. The main difference is that no variables are relaxed: we re-optimize variables whose values are already integer. We then use the same set $\mathcal{O}$ and $\mathcal{F}$, except that $\mathcal{F}$ is defined as follows: $\mathcal{F} = \{t | t \leq t^- \text{ and } t > t^+\}$. The parameters

κ and Δ are defined in the same way as for RF. In each iteration, we move several integer variables from $\mathcal{F}$ to $\mathcal{O}$. The main aim of the FO is to improve the solution obtained after RF. The FO heuristic is detailed in Algorithm 2. We denote by RFFO the global heuristic that executes the FO heuristic from the solution given by the RF heuristic.

Algorithm 2 Fix-and-optimize heuristic (FO)

1: Input: a feasible solution
2: Initialization: $t^- = 0$, $t^+ = \kappa$
3: Build sets $\mathcal{F}, \mathcal{O}$
4: **while** all variables are not binary **do**
5: Solve the resulting mixed-integer problem taking into account $\mathcal{F}$ and $\mathcal{O}$
6: $t^- = t^- + \Delta$, $t^+ = \min\{|T|; t^+ + \Delta\}$
7: Update sets $\mathcal{F}, \mathcal{O}$
8: **end while**

5. Computational experiments

This section presents the results obtained in our numerical campaign. We first describe the instances used and give the details of the experiments. Then, we analyze the results obtained from the MIP models, to analyze the impact of the degradation function and of the presence or absence of the Y variables. Finally, we analyze the performance of the RF-FO heuristic compared to the MIP approach.

5.1. Experimental setting

This section presents the instances used in the experiments with both solvers and RF-FO heuristic. The models are implemented in Julia with the library JuMP using Gurobi 11.0.0. The relax-and-fix and fix-and-optimize are also implemented in Julia. All the computational tests are carried out using a computer with an Intel Xeon Gold 6258R CPU @ 2.70GHz. We use only one thread in all the experiments. We keep all other Gurobi parameters at their default values. The tested instances and numerical experiments are discussed as follows.

Given the variety of settings encountered in the literature regarding maintenance features, we decide to generate our own instances. The instances we use are inspired from the work of Shamsaei and Van Vyve [32]. We set the number of items $|P|$ to $\{5, 20, 50, 100\}$ and the number of periods $|T|$ to $\{5, 10, 25\}$. The demand for each item in each time period is generated from $U[0; 50]$. The nominal capacity of the machine is generated from $U[40|P|; 50|P|]$. The setup costs are generated from $U[500; 1000]$. The unit holding costs are generated from $U[5; 10]$. We finally consider three ways to generate the maintenance costs: (i) from $U[500; 1000]$, i.e., similar to a single setup cost, (ii) we take the value of (i) that we multiply by $|P|$, to have maintenance costs closer to setup costs, and (iii), we take the value of (i) that we multiply by $\frac{|P|}{2}$, to have an value in between the one from (i) and the one from (ii). Those different ways of generating the maintenance costs also allows us to see the impact

of the repartition of costs on the difficulty of the instances. For each combination of number of items, number of periods, maintenance costs, we generate 10 different instances.

For each instance, we further test different values for the parameters of the degradation function. For the exponential degradation, the parameter α is set to $\{0.7; 0.8; 0.9\}$. For the linear degradation, the parameter β is set to $\{0.05; 0.1; 0.15\}$. For the dirac degradation, the parameter γ is set to $\{2; 3; 5\}$, and the parameter δ is set to $\{0.1; 0.2; 0.5\}$. It results in a total of 10,800 instances to be solved.

5.2. Analysis of the MIP solutions

In this section, we report the results obtained when using the solver only. The objective is to analyze the impact of the different instances settings. In that regards, we look at the LP relaxation value, the time taken to solve the LP relaxation, the value of the MIP solution, the time taken to solve the problem, the gap between the best lower and upper bounds at the end of the time limit, the number of nodes explored, the number of optimal solutions obtained, the integrality gap, the proportion of maintenance, setup and inventory holding costs. Table 3 reports the results obtained. In Table 3, the different metrics measured are displayed in columns LP, LP time, UB, UB time, Gap, Nodes, Opt, I-Gap, M cost, S cost and H cost, respectively. We also run each instance with classical maintenance variables z_t, i.e., variables that indicate if maintenance takes place in period t. We refer the reader to Shamsaei and Van Vyve [32] for the full model. We give the solver a time limit of two hours to solve each instance.

We can draw several conclusions from Table 3. For the sake of readability, we just highlight the main findings from the results. Regarding the type of the degradation function, one can see that the degradation function has no impact on the repartition of costs. We however see that the linear degradation generally leads to better performances in terms of time to solve the MIP, final gap, number of nodes explored, and integrality gap. On the contrary, the dirac degradation function has a negative impact on the value of the linear relaxation.

Regarding the values chosen for the different parameters in the maintenance degradation functions, the changes in the different metrics are not always linear with the evolution of the parameters values. The evolution of the values of α, β, δ and γ has expected impacts on the metrics. For instance, lower capacity reduction leads to easier instances to be solved.

The value chosen for maintenance costs has an impact on the difficulty of the instances. Having low maintenance costs leads to easier instances, as stated by the time to solve the MIP, the number of optimal solutions obtained, the final gap and the integrality gap. We also note that the proportion of inventory holding costs is not sensitive to the choice of maintenance degradation function.

In terms of modeling, our newly proposed maintenance variables have limited benefits. The number of nodes explored is reduced by around 12%, and the time taken to solve the MIP is around 3% lower with those new variables. This is aligned with the improved LP relaxation obtained by using the new maintenance variables.

If we consider that the capacity decreases if and only if production occurs, i.e., when variables Y are present in the model, we obtain more difficult instances. This can be seen in particular through the final gap, the number of nodes explored, and the time taken to solve the MIP. We however observe that the time limit of two hours

Table 3: Analysis of the instance settings

Parameter	Value	LP	LP time (s)	UB	UB time (s)	Gap (%)	Nodes	Opt	I-Gap (%)	M cost (%)	S cost (%)	H cost (%)
Maintenance type	Dirac	234578	0.1	242880	962	0.3	92876	9	2.9	12	30	58
	Exponential	241015	0.1	248568	999	0.1	77614	9	2.9	14	28	58
	Linear	236847	0.1	241568	590	0	57393	9	2	12	29	59
α	0.7	250056	0.1	260275	1206	0.2	89895	9	3.9	17	28	55
	0.8	240884	0.1	248928	1119	0	87344	9	3.1	14	28	58
	0.9	232104	0.1	236500	672	0	55602	10	1.7	11	29	60
β	0.05	230312	0.1	232721	114	0	9306	10	1.2	11	29	61
	0.1	236596	0.1	241905	879	0	116622	9	2.2	12	30	58
	0.15	243633	0.1	250077	776	0	46252	9	2.5	15	28	57
δ	0.1	240967	0.1	251329	683	0.2	58366	9	3.3	14	30	56
	0.3	234769	0.1	245581	1157	0.7	156523	9	3.5	13	30	58
	0.5	227999	0.1	231732	1046	0	63738	9	1.8	10	29	60
γ	2	237824	0.1	247949	1165	0.5	117325	8	3.5	13	30	57
	3	234444	0.1	242395	1127	0.4	117265	9	2.7	12	29	58
	5	231466	0.1	238297	593	0.1	44037	9	2.4	11	30	59
Maintenance costs	sc	212251	0	213022	15	0	1367	10	0.9	2	31	66
	$sc * \|P\|$	259939	0.1	274027	1537	0.5	154299	8	4.5	22	27	51
	$sc * \frac{\|P\|}{2}$	236768	0.1	244217	1133	0.1	92515	9	2.6	13	29	58
Model	z_t	234551	0.1	243759	909	0.2	87488	9	3.4	13	29	58
	z_{kt}	238088	0.1	243752	881	0.2	77966	9	1.9	13	29	58
Y	no	236319	0.1	243777	822	0	89517	9	2.7	13	29	58
	yes	236319	0.1	243734	967	0.4	75937	9	2.6	13	29	58
$\|T\|$	5	101789	0	103003	0	0	24	10	1.5	16	23	61
	10	181701	0	186485	311	0.1	94664	10	2.7	12	31	58
	25	425467	0.3	441778	2373	0.5	153493	7	3.9	11	34	56
$\|I\|$	5	27751	0	29153	41	0	7893	10	4	14	28	58
	20	107394	0	110943	663	0.2	101197	9	2.4	12	29	58
	50	269248	0.1	277478	1305	0.3	127245	9	2.2	12	30	58
	100	540884	0.3	557448	1570	0.3	94572	8	2.2	12	30	58

	RF		FO	
$\|T\|$	κ	Δ	κ	Δ
5	2	1	3	1
10	3	1	5	2
25	6	3	10	4

Table 4: Heuristic setting

Method	Obj	CPU time (s)	gap MIP (%)	Best (%)
MIP	243998	869	0	95
RF	245245	30	0.4	32
RFFO	244410	80	0.2	49

Table 5: Heuristic performance

is sufficient to obtain a similar number of optimal solutions obtained as when capacity decreases regardless of if production occurs.

Finally, we see that the number of periods has a much stronger influence on the difficulty of the instances compared to the number of items. This can be easily explained by the increased number of maintenance variables.

5.3. Heuristic performance

In this Section, we analyze the performance of the RFFO heuristic in terms of CPU time taken and gap compared to the solution given by the FL-EM model. Note that we only kept this model in the analyses since it gives the best results (see Table 3 when comparing the parameters z_t and z_{kt}). The values of κ and Δ are determined based on $|T|$ for each heuristic. They are presented in Table 4 and were set after several preliminary tests on the instances. The execution of heuristics and the resolution of MILP subproblems with Gurobi are performed on a single core. For relax-and-fix, each MILP subproblem is limited to 180 seconds, while for fix-and-optimize, the limit is 300 seconds. Additionally, the maximum execution time allocated to each heuristic is set to 3600 seconds.

We display the results obtained in Table 5. In Table 5, for a specific method, i.e., a specific line, the columns Obj, CPU time, gap MIP and Best report the average solution value, average CPU time taken by the method, average gap compared to the solution given by the MIP model, and proportion of instances where the method obtained the best objective, respectively. The gap is computed as $\frac{obj_H - obj_{MIP}}{obj_{MIP}}$ where obj_H and obj_{MIP} denote the objective function obtained from heuristic H and from the MIP model, respectively.

In terms of quality of the objective function, the RF heuristic has on average solutions that are 0.4 % more costly than the best solution found by the solver. This slightly higher cost comes, however, at the benefit of a much shorter average CPU time taken. Indeed, the RF heuristic takes on average close to 30 times less time than the use of Gurobi. This objective value is further improved by the use of the FO heuristic, to reach an average gap of 0.2 % compared to the solution given by Gurobi. This additional gain comes this time, however,

at a higher CPU time. The CPU time taken by the RFFO is yet, on average, 11 times less than that taken by Gurobi. We can therefore see the gain given by the use of the RFFO heuristic.

Still related to the quality of the objective function, the number of best solutions obtained by RFFO is 49%. We are even able to obtain a better solution, in terms of objective function, compared to Gurobi, for 5% of the instances (see the following subsection).

In the following subsections, we analyze in more detail the quality of the objective function and the CPU time gains.

5.3.1. Quality of the solution

We display in Table 6 the detailed results obtained by the RF and RFFO heuristics. In Table 6, we report the solution value obtained from Gurobi, the solution value obtained from the RF and RFFO heuristics, the gap between the solution of the heuristic and Gurobi, the proportion of instances where the RFFO heuristic obtains the best solution quality, and the proportion of instances where the RFFO strictly obtains the best solution quality. Those values are reported in columns UB MIP, RF Obj, RFFO Obj, RF gap, RFFO gap, RFFO Better, and RFFO Strictly better, respectively.

In Table 6, one can see that the performance of both the RF and RFFO heuristics are quite stable in terms of gap compared to the solution obtained by Gurobi. This illustrates the robustness of the approaches, which is a main strength. We also see that the RFFO heuristic obtains a quite reasonable proportion of solutions with the best objective, always more than 50%. The proportion of instances where the RFFO strictly obtains the best solution is however quite low, close to 5% on average. In Table 6, one can see that for instances with 5 time periods, the proportion of best solution obtained is quite high. This is in line with the performance of Gurobi on such instances, as shown in Table 3. We finally see that for harder instances, i.e., instances with 100 items, the RFFO heuristic has the strictly best solution value for 12% of the instances, more than twice the average performance on this metric.

5.3.2. CPU time gains

In terms of CPU times, we note that the CPU gains vary depending on the instance setting. Table 7 displays the distribution of CPU time gains compared to the use of Gurobi, and the repartition of the CPU time between RF and FO for the RFFO heuristic. The CPU time gains are measured as the ratio between the CPU time taken by Gurobi, and the CPU time taken by the heuristic. We report in this table the minimum, maximum, median, first and third quartiles, and average values for such metrics.

One can see in Table 7 that the CPU time gains for RF heuristic ranges between 0.02 times faster and 4077 times faster, with an average of 34.5 times faster. The CPU time gains for the RFFO heuristic ranges between 0.02 time faster and 3217 times faster, with an average of 13.9 times faster. Finally, the repartition of time between RF and FO ranges between 0.6% and 96%, with an average of 53%. This illustrates that the instance setting has a strong impact on the performance of the heuristic, in terms of CPU time. The distribution of CPU time gains also indicates that a large number of instances do not offer strong gains, as observed with the median

Table 6: Detailed analysis of quality of the solution for the heuristics

Parameter	Value	UB MIP	RF Obj	RF gap (%)	RFFO Obj	RFFO gap (%)	RFFO better (%)	RFFO Strictly better (%)
Maintenance type	Dirac	243288	244571	0.4	243656	0.1	56	5
	Exponential	248572	249723	0.5	248977	0.2	49	5
	Linear	241554	242791	0.4	242105	0.2	60	4
α	0.7	260283	261562	0.6	260578	0.3	38	7
	0.8	248932	250028	0.5	249341	0.2	50	4
	0.9	236501	237578	0.3	237012	0.1	58	6
β	0.05	232721	233117	0.1	232907	0.1	78	6
	0.1	241906	243875	0.8	242878	0.4	50	4
	0.15	250035	251381	0.4	250531	0.1	52	3
δ	0.1	252551	254067	0.6	253042	0.2	54	5
	0.3	245581	246731	0.4	245829	0.1	54	7
	0.5	231732	232915	0.3	232096	0.1	59	5
γ	2	247937	249455	0.6	248359	0.2	51	4
	3	243624	244822	0.4	243904	0.1	57	7
	5	238303	239437	0.3	238705	0.1	59	5
Maintenance costs	sc	213022	213110	0.1	213065	0.1	63	5
	$sc * \|P\|$	274026	276319	0.7	274680	0.2	52	7
	$sc * \frac{\|P\|}{2}$	244946	246307	0.5	245485	0.2	51	4
Y	no	244264	245330	0.4	244681	0.2	54	4
	yes	243732	245161	0.5	244139	0.2	56	6
$\|T\|$	5	103003	103132	0.2	103042	0.1	81	6
	10	186658	187664	0.6	187040	0.2	45	3
	25	442333	444941	0.6	443148	0.2	39	7
$\|I\|$	5	29172	29362	0.6	29246	0.2	69	0
	20	111039	111658	0.4	111277	0.2	59	3
	50	277746	278979	0.3	278176	0.1	46	5
	100	558035	560983	0.4	558941	0.1	46	12
Average		243998	245245	0.4	244410	0.2	55	5

Table 7: Detailed analysis of CPU time gains for the heuristics

	CPU time RF gain	CPU time RFFO gain	Prop RF time (%)
Min	0.02	0.02	0.6
Max	4077	3217	96
1st quartile	0.4	0.2	40
Median	0.9	0.5	52
3rd quartile	5.2	2.9	67
Average	34.5	13.9	53

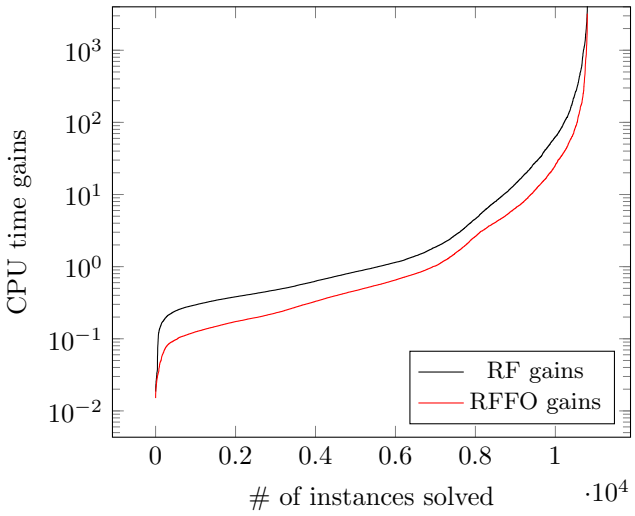

Figure 1: Profile curves on CPU time gains

value. A closer look at the detailed results indicates that instances with 5 time periods, i.e., easy instances, make the use of the heuristic useless.

To better illustrate the CPU time gains, we report in Figure 1 the profile curve for the different CPU time gains on a logarithmic scale. The x-axis indicates the number of instances solved, while the y-axis indicates, for a given x, the CPU time gains in increasing order.

One can see on Figure 1 that the RF heuristic has gains after 5568 instances, i.e., roughly half of the instances, and the RFFO heuristic has gains after 6939 instances, i.e., roughly two third of the instances solved. We further see that the CPU time gains tend to explode after instance 9000, showing the usefulness of our approach for harder instances. Similar to Table 7, we report in Table 8 the CPU time gains obtained by the heuristics depending on the instance settings.

In Table 8, one can see that the cost configuration has a strong impact. Indeed, when the maintenance costs follow the setup costs (first cost configuration), the CPU time gains are much lower than on average, with the RF heuristic being 2.9 times faster on average, and the RFFO heuristic being just 1.3 time faster on average. On the contrary, difficult instances, i.e., instances with higher maintenance costs, come with higher CPU time gains.

In terms of capacity degradation function, the dirac degradation brings higher CPU time gains, the RF heuristic being on average 37.8 times faster, and the RFFO one being on average 15.6 times faster.

The instances with a higher number of items (50 or 100), come also with higher CPU time gains. With 100 items, the RF heuristic is even 48.1 times faster than Gurobi on average.

We also note that the consideration or not of production decisions on capacity degradation has no impact on the CPU time gain for the RF heuristic, but its presence has a positive impact on the gains for the RFFO heuristic.

We finally note that for instances with a low number of periods (5 periods), the heuristics are not competitive. This contrasts with instance with 10 periods where the RFFO has the highest CPU time gains.

Table 8: Detailed analysis of CPU time gains for the heuristics

Parameter	Value	CPU time RF gains	CPU time RFFO gains	Prop RF time (%)
Maintenance type	Dirac	37.8	15.6	53
	Exponential	26	12.6	53
	Linear	33.2	10.5	50
α	0.7	31	12.8	54
	0.8	25.3	13.2	54
	0.9	21.7	11.7	52
β	0.05	4.8	1.9	48
	0.1	74.2	19.3	48
	0.15	20.7	10.3	54
δ	0.1	12.7	7.1	59
	0.3	76.1	33.1	53
	0.5	24.7	6.4	48
γ	2	51.5	24.6	54
	3	49.9	18	53
	5	12.1	4.1	53
Maintenance costs	sc	2.9	1.3	51
	$sc * \|P\|$	64.2	24.7	53
	$sc * \frac{\|P\|}{2}$	36.5	15.8	54
Y	no	34.2	10.6	51
	yes	34.9	17.3	54
$\|T\|$	5	0.7	0.3	45
	10	61.9	23.1	62
	25	41.1	18.4	51
$\|I\|$	5	9.3	4.8	61
	20	36.5	18.2	57
	50	44.2	20	51
	100	48.1	12.8	42
Average		34.5	13.9	53

6. Conclusion

In this paper, we address an integrated lot sizing and maintenance problem. We consider a single machine whose capacity decreases with time. We test different capacity degradation functions: an exponential one, a linear one, and a dirac one, to cover a broader spectrum of potential degradations. We further consider that the capacity decreases even if no production occurs. We model the problem using newly introduced maintenance variables compared to the literature. Those new variables are inspired from the transportation formulation for the lot sizing problem. We further develop a relax-and-fix fix-and-optimize heuristic to solve the problem, and compare the results of using a solver compared to the heuristic approach. The results indicate that the heuristic is very robust in terms of quality of the solution obtained, and is even able to obtain better solutions than the solver for difficult instances (5% of the instances). The heuristic obtains CPU time gains for numerous instances, and the CPU time gains can be quite large. On average, the RF heuristic is 34 times faster than the use of Gurobi, and the RFFO is 14 times faster.

In future research we would like to explore the possibility of having a decrease of capacity based on the production quantities. We would also like to explore different strategies for the RFFO approach in terms of variables to be relaxed and variables to be fixed.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Discovery Grant 2021-03327 and by the Fonds de Recherche Nature et Technologies of Québec (FRQNT) under grant 329202. This support is gratefully acknowledged.

Appendix A. Proof of Proposition 1

In this section we present the proof of Proposition 1

Proposition. *Let V_1 be the value of the linear relaxation of $(FL-EM)$ formulation and V_2 be the value of the linear relaxation obtained with the $(FL-M_{DW})$ formulation, then $V_1 = V_2$.*

Proof. Let R be the polyhedron induced by constraints (2)-(4), (11)-(12) modeling the facility location constraints, and let Q_1 be the polyhedron modeling the maintenance:

$$Q_1 = \{z_{kt} \in [0,1], c_t \in \mathbb{R}^+, \text{ satisfying constraints (5)-(10)}\} \tag{A.1}$$

The solution space of the linear relaxation of the (FL-EM) model is $R \cap Q_1$. Meanwhile, the $(FL-M_{DW})$ model is obtained by applying a Dantzig-Wolfe decomposition to the polyhedron Q_1. Thus, the solution space of its linear relaxation is $R \cap Q_2$ where Q_2 is the convex envelope of points of Q_1 corresponding to integer values of z_{kt}. In order to show that the value of both linear relaxations is the same, we will show that $Q_1 = Q_2$ and thus that the solution space of both formulations is the same.

The inclusion $Q_2 \subseteq Q_1$ is implied by the fact Q_2 is the convex envelope of integer points of Q_1 and that Q_1 as a polyhedron is convex. Thus, let us now show $Q_1 \subseteq Q_2$.

To that end, consider an assignment of the variables z_{kt} satisfying constraints (5)-(10). We will show that there exists an assignment of the variables w_j of the $(FL-M_{DW})$ model inducing the same maintenance actions, the same capacity and the same cost. This will mean any point of Q_1 can be expressed as a point of Q_2; *i.e.* that $Q_1 \subseteq Q_2$. We now explain Algorithm 3 which computes such an assignment of the variables w_j from an assignment of the variables z_{kt}.

Note that each variable w_j is associated with a maintenance plan and that an assignment of the variables w_j is a set of coefficients in a convex combination of maintenance plans. In what follows, we will call partial plan, a boolean vector corresponding to the beginning a maintenance plan (*i.e.* indicating for the first time-steps if a maintenance action is taken). The operator $\oplus$ will be used to add a maintenance action to a partial plan; *e.g.* $[1,1,0] \oplus 1 = [1,1,0,1]$. Let us denote π_t the set of partial plans created at iteration $t \in T$ by the algorithm together with their coefficients in the convex combination, $\pi_t = [(P_i, w_{P_i})]_{i \in I_t}$, where P_i is a partial plan and w_{P_i} its associated coefficient. Let also π_t^k be the subset of π_t which contains plans that have no maintenance since period k. For π a set of partial plans created by the algorithm, let $W(\pi)$ be the sum of the coefficients associated to these partial plans.

Algorithm outline. The algorithm constructs a set of maintenance plan as well as their coefficients in the convex combination. To that end, a set of partial plans containing only the action for the first time-step $\pi_0 = [((1), 1)]$ is initialized. Note that a maintenance action is required at the first time-step. Then at iteration $t \in T \setminus \{0\}$, the partial plans in π_{t-1} are extended with a maintenance action for time-step t. Note that one plan P can be extended with two maintenance actions. In this case, two plans $P \oplus 0$ and $P \oplus 1$ are created and their coefficients in the convex combination must satisfy $w_{P\oplus 0} + w_{P\oplus 1} = w_P$. This construction has two underlying objectives. First, we want in the end that the plans in convex combination decide the same maintenance actions as the variables z_{kt} (this will imply the same maintenance cost). If denote P_t^j the boolean maintenance action taken by plan j at time-step t then we precisely want for all $t \in T$, $z_{tt} = \sum_j P_t^j w_j$. This means the proportion of maintenance plans extended with a one at each iteration is decided by z_{tt}. We also want that the maintenance plans created to induce the same capacity as the variables z_{kt}; the highest capacity possible with these maintenance actions. In particular, this induces that we will always extend with a 1 the plan whose last maintenance action is the earliest because this resets to C_{max} the capacity of the plan with the lowest current capacity.

Algorithm 3 Decomposition algorithm

Require: $z = \{z_{tt} | t \in T\}$ ▷ $z_{11}, z_{22}, ... \in [0, 1]$ are the maintenance actions realized.

Output : $\pi_{|T|}$ a set of maintenance plans with a coefficient in the convex combination
$\pi_0 = [((1), 1)]$
for $t \in T \setminus \{0\}$ **do**
  $temp = z_{tt}$
  while $temp > 0$ **do**
    Set P to be the unextended plan in π_{t-1} whose last maintenance action is the earliest
    if $temp \geq w_P$ **then**
      $\pi_t = \pi_t \cup (P \oplus 1, w_P)$
      $temp = temp - w_P$
    else
      $\pi_t = \pi_t \cup (P \oplus 1, temp)$
      $\pi_t = \pi_t \cup (P \oplus 0, w_P - temp)$
      $temp = 0$
    end if
  end while
  for all unextended maintenance plans $P \in \pi_{t-1}$ **do**
    $\pi_t = \pi_t \cup (P \oplus 0, w_P)$
  end for
end for
return $\pi_{|T|}$

We now show that the solution given by Algorithm 3 has the same cost and capacity as the optimal solution of the (FL-EM) model. To that end, we will show that the two solutions use the same maintenance action, *i.e.* $W(\pi_t^k) = z_{kt}$ for all $k, t \in T$ with $k \leq t$. Indeed, in this case the costs are the same:

$$\sum_j M_j w_j = \sum_j \sum_t mc_t P_t^j w_j \tag{A.2}$$

$$= \sum_t mc_t \sum_j P_t^j w_j \tag{A.3}$$

$$= \sum_t mc_t W(\pi_t^t) \tag{A.4}$$

$$= \sum_t mc_t z_{tt} \tag{A.5}$$

and — denoting δ_{kt}^j the boolean indicating if the last maintenance before time-step t in plan j is at time-step k — the capacities are the same:

$$\sum_j c_t^j w_j = \sum_j C_{max} \sum_{k=0}^{t} \alpha^{t-k} \delta_{kt}^j w_j \tag{A.6}$$

$$= C_{max} \sum_{k=0}^{t} \alpha^{t-k} \sum_{j} \delta_{kt}^{j} w_j \tag{A.7}$$

$$= C_{max} \sum_{k=0}^{t} \alpha^{t-k} W(\pi_t^k) \tag{A.8}$$

$$= C_{max} \sum_{k=0}^{t} \alpha^{t-k} z_{kt} \tag{A.9}$$

But first, let us show that, there is an optimal solution of the (FL-EM) model for which the variables z_{kt} for $k < t$ will take the value $\min\{z_{kk}, 1 - \sum_{i=k+1}^{t} z_{it}\}$. Let us consider a specific pair $(k,t) \in T^2$ with $k < t$. If constraints (7) and (8) are satisfied, z_{kt} must be lower than $\min\{z_{kk}, 1 - \sum_{i=k+1}^{t} z_{it}\}$. Moreover, if z_{kt} is strictly lower than $\min(z_{kk}, 1 - \sum_{i=k+1}^{t} z_{it})$, then we can construct an optimal solution of the (C-EM) model with a higher z_{kt} as follows. If $\sum_{i=1}^{t} z_{it} < 1$ then one can just increase z_{kt} by a small amount without breaking any constraint. Otherwise ($\sum_{i=1}^{t} z_{it} = 1$), having $z_{kt} < 1 - \sum_{i=k+1}^{t} z_{it}$ means that $\sum_{i=k}^{t} z_{it} < 1$ which means at least one of the variables $z_{k't}$ for $k' < k$ is higher than zero. Let $\epsilon \in \mathbb{R}$ be small enough. One can modify the solution by adding ϵ to z_{kt} and subtracting ϵ to $z_{k't}$. This remains an optimal solution for a small enough ϵ since $z_{kt} < z_{kk}$ and this process just increases the capacity c_t (since $\alpha \leq 1$). In other words, if we do not have $z_{kt} = \min\{z_{kk}, 1 - \sum_{i=k+1}^{t} z_{it}\}$, we can always build an optimal solution where we do have $z_{kt} = \min\{z_{kk}, 1 - \sum_{i=k+1}^{t} z_{it}\}$.

In line 7 and 12, Algorithm 3 extends maintenance plans with a 1 until the sum of the coefficients of the extended plans equals z_{tt}. This exactly means $W(\pi_t^t) = z_{tt}$.

We will now show that for every $k, t \in T, k < t$, we have $W(\pi_t^k) = z_{kt}$. To that end, we will show $W(\pi_t^k) = \min\{z_{kk}, 1 - \sum_{i=k+1}^{t} z_{it}\}$. This will be shown by induction on the pairs (k,t) with $k \leq t$ in the following order :

$$(0,1) \to (1,2) \to (0,2) \to (2,3) \to (1,3) \to (0,3) \to (3,4) \to (2,4) \to (1,4) \to (0,4) \to (4,5)...$$

When considering the induction step of a pair, the hypothesis will be that $W(\pi_t^k) = z_{kt} = \min(z_{kk}, 1 - \sum_{i=k+1}^{t} z_{it})$ for all the previous pairs.

Base Case. $W(\pi_1^0) = w_{[1,0]} = 1 - w_{[1,1]} = 1 - z_{11}$ and $w_{[1,0]} \leq w_{[1,]} = z_{00}$. Thus $W_{\pi_1^0} = \min(z_{00}, 1 - z_{11})$.

Inductive step. Let us now prove that $W(\pi_t^k) = z_{kt}$ for a specific $(k,t) \in T^2$ with $k < t$. The partial plans in π_t^k are the partial plans of π_{t-1}^k that have been extended by the algorithm with a zero. Let us now do the following case disjunction: $\sum_{k'<k} W(\pi_{t-1}^{k'}) \leq z_{tt}$ or $\sum_{k'<k} W(\pi_{t-1}^{k'}) > z_{tt}$. This case disjunction decides whether some of the partial plans in π_{t-1}^k are extended with a one or not. This is because the algorithm prioritizes extending with a one the partial plans whose last maintenance action is the earliest.

Case $\sum_{k'<k} W(\pi_{t-1}^{k'}) \leq z_{tt}$: In this case, all the partial plan of π_{t-1} finishing with more than $t-1-k$ zeros are extended with a one at time-step t. Thus, none of the plans in π_t finish with more than $t-k$ zeros. Thus, $\sum_{k'=k}^{t} W(\pi_t^{k'}) = \sum_{k'=0}^{t} W(\pi_t^{k'}) = 1$. This means that $W(\pi_t^k) = 1 - \sum_{k'=k+1}^{t} W(\pi_t^{k'}) = 1 - \sum_{k'=k+1}^{t} z_t^{k'}$ because of the induction hypothesis. Moreover, $W(\pi_t^k) \leq W(\pi_{t-1}^k)$ since the partial plans in π_t^k are the partial plans of π_{t-1}^k that have been extended by the algorithm with a zero. This leads to by the induction hypothesis to

$W(\pi_t^k) \leq z_{k,t-1} = \min(z_{kk}, 1-\sum_{i=k+1}^{t} z_{i,t-1}) \leq z_{kk}$. This finishes to show that $W(\pi_t^k) = \min(z_{kk}, 1-\sum_{i=k+1}^{t} z_{it})$ in this case.

Case $\sum_{k'<k} W(\pi_{t-1}^{k'}) > z_{tt}$*:* In this case, at least one partial plan of π_{t-1} finishing with more than $t-1-k$ zeros is extended with a zero at time-step t. Thus, at least one plan in π_t finishes with more than $t-k$ zeros. Thus, $\sum_{k'=k}^{t} W(\pi_t^{k'}) < 1$ which means $W(\pi_t^k) < 1-\sum_{k'=k+1}^{t} W(\pi_t^{k'}) = 1-\sum_{k'=k+1}^{t} z_t^{k'}$ because of the induction hypothesis. Moreover, since only partial plans with more than $t-1-k$ zeros were extended with a one, all the plans finishing with $t-k$ zeros were extended with a zero by the algorithm. Thus, $W(\pi_t^k) = W(\pi_{t-1}^k) = z_{k,t-1} = \min\{z_{kk}, 1-\sum_{i=k+1}^{t-1} z_{i,t-1}\}$. Moreover, by the induction hypothesis we have $\sum_{k'=0}^{k-1} z_{k',t-1} = \sum_{k'=0}^{k-1} W(\pi_{t-1}^{k'}) > z_{tt} \geq 0$. This means in conjunction with constraints (8) that $z_{k,t-1} < 1-\sum_{i=k+1}^{t-1} z_{i,t-1}$ which means $z_{k,t-1} = z_{kk}$ and in turn $W(\pi_t^k) = z_{kk}$. This finishes proving that $W(\pi_t^k) = \min\{z_{kk}, 1-\sum_{i=k+1}^{t} z_{it}\}$ in this case. □

References

[1] N. Absi, W. van den Heuvel, and S. Dauzére-Pérès. Complexity analysis of integrated dynamic lot sizing and maintenance planning problems. *European Journal of Operational Research*, 318(1):100–109, 2024. doi: 10.1016/j.ejor.2024.04.025.

[2] E.-H. Aghezzaf and N. M. Najid. Integrated production planning and preventive maintenance in deteriorating production systems. *Information Sciences*, 178(17):3382–3392, 2008.

[3] E. H. Aghezzaf, M. A. Jamali, and D. Ait-Kadi. An integrated production and preventive maintenance planning model. *European journal of operational research*, 181(2):679–685, 2007.

[4] E.-H. Aghezzaf, A. Khatab, and P. Le Tam. Optimizing production and imperfect preventive maintenance planning's integration in failure-prone manufacturing systems. *Reliability Engineering & System Safety*, 145: 190–198, 2016.

[5] M. Alaoui-Selsouli, A. Mohafid, and N. M. Najid. Lagrangian relaxation based heuristic for an integrated production and maintenance planning problem. *International Journal of Production Research*, 50(13): 3630–3642, 2012.

[6] M. Aramon Bajestani, D. Banjevic, and J. C. Beck. Integrated maintenance planning and production scheduling with markovian deteriorating machine conditions. *International Journal of Production Research*, 52(24):7377–7400, 2014.

[7] J. Arts. Maintenance modeling and optimization. 2017.

[8] J. Ashayeri, A. Teelen, and W. Selenj. A production and maintenance planning model for the process industry. *International journal of production research*, 34(12):3311–3326, 1996.

[9] S. Babaeimorad, P. Fatthi, and H. Fazlollahtabar. A joint optimization model for production scheduling and preventive maintenance interval. *International Journal of Engineering*, 34(11):2508–2516, 2021.

[10] K. Chaabane, A. Khatab, C. Diallo, E.-H. Aghezzaf, and U. Venkatadri. Integrated imperfect multimission selective maintenance and repairpersons assignment problem. *Reliability Engineering & System Safety*, 199: 106895, 2020.

[11] H. Chen. Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. *Omega*, 56:25–36, 2015.

[12] B. De Jonge and P. A. Scarf. A review on maintenance optimization. *European journal of operational research*, 285(3):805–824, 2020.

[13] J. Desrosiers, M. Lübbecke, G. Desaulniers, and J. B. Gauthier. Branch-and-price. Les Cahiers du GERAD G-2024-36, Groupe d'études et de recherche en analyse des décisions, GERAD, Montréal QC H3T 2A7, Canada, June 2024. URL `https://www.gerad.ca/en/papers/G-2024-36`.

[14] S. El-Ferik. Economic production lot-sizing for an unreliable machine under imperfect age-based maintenance policy. *European Journal of Operational Research*, 186(1):150–163, 2008.

[15] M.-C. Fitouhi and M. Nourelfath. Integrating noncyclical preventive maintenance scheduling and production planning for a single machine. *International Journal of Production Economics*, 136(2):344–351, 2012.

[16] M.-C. Fitouhi and M. Nourelfath. Integrating noncyclical preventive maintenance scheduling and production planning for multi-state systems. *Reliability Engineering & System Safety*, 121:175–186, 2014.

[17] K. Gao, R. Peng, L. Qu, and S. Wu. Jointly optimizing lot sizing and maintenance policy for a production system with two failure modes. *Reliability Engineering & System Safety*, 202:106996, 2020.

[18] A. Garg and S. Deshmukh. Maintenance management: literature review and directions. *Journal of quality in maintenance engineering*, 12(3):205–238, 2006.

[19] M. Geurtsen, J. B. Didden, J. Adan, Z. Atan, and I. Adan. Production, maintenance and resource scheduling: A review. *European Journal of Operational Research*, 305(2):501–529, 2023.

[20] S. Helber and F. Sahling. A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, 123(2):247–256, 2010.

[21] C. Kojchen and F. Monchy. *Maintenance : outils, méthodes et organisations efficientes, 5e Edition*. Dunod, 2019.

[22] K. Krarup and O. Bilde. *Plant location, set covering and economic lot-sizes: An $O(mn)$ algorithm for structured problems*. L.Collatz (Editor), Birkhauser Verlag, Bassel, 1977.

[23] J. C. Lang and Z.-J. M. Shen. Fix-and-optimize heuristics for capacitated lot-sizing with sequence-dependent setups and substitutions. *European Journal of Operational Research*, 214(3):595–605, 2011.

[24] P. Le Tam, E.-H. Aghezzaf, A. Khatab, and C. H. Le. Integrated production and imperfect preventive maintenance planning - an effective milp-based relax-and-fix/fix-and-optimize method. In *International Conference on Operations Research and Enterprise Systems*, volume 2, pages 483–490. SCITEPRESS, 2017.

[25] G. Levitin and A. Lisnianski. Optimization of imperfect preventive maintenance for multi-state systems. *Reliability Engineering & System Safety*, 67(2):193–203, 2000.

[26] L. Li, S. Song, C. Wu, and R. Wang. Fix-and-optimize and variable neighborhood search approaches for stochastic multi-item capacitated lot-sizing problems. *Mathematical Problems in Engineering*, 2017(1): 7209303, 2017.

[27] Z. Lu, Y. Zhang, and X. Han. Integrating run-based preventive maintenance into the capacitated lot sizing problem with reliability constraint. *International Journal of Production Research*, 51(5):1379–1391, 2013.

[28] N. M. Najid, M. Alaoui-Selsouli, and A. Mohafid. An integrated production and maintenance planning model with time windows and shortage cost. *International Journal of Production Research*, 49(8):2265–2283, 2010.

[29] A. Roshani, D. Giglio, and M. Paolucci. A relax-and-fix heuristic approach for the capacitated dynamic lot sizing problem in integrated manufacturing/remanufacturing systems. *IFAC-PapersOnLine*, 50(1):9008–9013, 2017.

[30] F. Sahling, L. Buschkühl, H. Tempelmeier, and S. Helber. Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research*, 36 (9):2546–2553, 2009.

[31] M. Schreiber, K. Vernickel, C. Richter, and G. Reinhart. Integrated production and maintenance planning in cyber-physical production systems. *Procedia CIRP*, 79:534–539, 2019.

[32] F. Shamsaei and M. Van Vyve. Solving integrated production and condition-based maintenance planning problems by mip modeling. *Flexible Services and Manufacturing Journal*, 29:184–202, 2017.

[33] T. W. Sloan and J. G. Shanthikumar. Combined production and maintenance scheduling for a multiple-product, single-machine production system. *Production and Operations Management*, 9(4):379–399, 2000.

[34] N. Sortrakul, H. L. Nachtmann, and C. R. Cassady. Genetic algorithms for integrated preventive maintenance planning and production scheduling for a single machine. *Computers in Industry*, 56(2):161–168, 2005.

[35] C. F. M. Toledo, M. da Silva Arantes, M. Y. B. Hossomi, P. M. França, and K. Akartunalı. A relax-and-fix with fix-and-optimize heuristic applied to multi-level lot-sizing problems. *Journal of heuristics*, 21:687–717, 2015.

[36] L. Wang, Z. Lu, and X. Han. Joint optimisation of production, maintenance and quality for batch production system subject to varying operational conditions. *International Journal of Production Research*, 57(24): 7552–7566, 2019.

[37] J. Xiao, C. Zhang, L. Zheng, and J. N. Gupta. Mip-based fix-and-optimise algorithms for the parallel machine capacitated lot-sizing and scheduling problem. *International Journal of Production Research*, 51 (16):5011–5028, 2013.

[38] S.-l. Yang, Y. Ma, D.-l. Xu, and J.-b. Yang. Minimizing total completion time on a single machine with a flexible maintenance activity. *Computers & Operations Research*, 38(4):755–770, 2011.

[39] M. You, Y. Xiao, S. Zhang, S. Zhou, P. Yang, and X. Pan. Modeling the capacitated multi-level lot-sizing problem under time-varying environments and a fix-and-optimize solution approach. *Entropy*, 21(4):377, 2019.

[40] K. Zhu. A joint optimization model of production scheduling and maintenance based on data driven for a parallel-series production line. *Journal of Mathematics*, 2021:1–11, 2021.