ENAC Civil Servant Graduate Engineer
Major in Air Operations and Safety (OPS)

Julien LEGAVRE

IENAC 20

End of Studies Project Memoir

August 2023

Mixed-integer linear programming modelling of an integrated maintenance and capacitated lot-sizing problem

2023

END OF STUDIES MEMORY

French Civil Aviation University

Graduate Civil Servant Engineer

AND

University of Toulouse III - Paul Sabatier

Master of Operational Research

Mixed-integer linear programming modelling of an integrated maintenance and capacitated lot-sizing problem

---

Julien LEGAVRE

August 2023

Promotion:

IENAC20 OPS

Supervised at CIRRELT by:

Matthieu GRUSON and François LAMOTHE

Supervised at the French Civil Aviation University by:

Andrija VIDOSAVLJEVIC

CENTRE INTERUNIVERSITAIRE DE RECHERCHE
SUR LES RÉSEAUX D'ENTREPRISE
LA LOGISTIQUE ET LE TRANSPORT

Résumé

Les entreprises dans l'industrie cherchant à être toujours plus productives et efficaces dans leurs opérations, la planification de la maintenance et celle de la production se développent de plus en plus. Elles consistent principalement à établir pour la première les créneaux de maintenance, et pour la seconde, les quantités à produire pour chaque période de temps (semaines ou mois). Ces planifications sont généralement optimisées séparément. Le problème de planification de production (problème de lotissement) a été largement étudié par le biais de modèles de Programmation Linéaire en Nombres Entiers (PLNE) le décrivant. Il est toujours compliqué de résoudre avec une garantie d'optimalité et dans un temps raisonnable de grandes instances sans relâcher certaines contraintes comme l'obligation de satisfaire les demandes à chaque période (possibilités d'être en retard dans la production). C'est pourquoi, beaucoup d'études récentes portent sur des méthodes approchées pour résoudre ce problème, afin d'avoir une solution correcte (écart avec la solution optimale faible) dans un temps raisonnable pour ces instances. Le problème de la planification de la maintenance a quant à lui été étudié majoritairement comme un problème distinct de la planification de production où l'on cherche à maximiser la disponibilité d'une machine (Garg et al. [1]). La difficulté d'inclure des actions maintenance dans un modèle de planification de production en PLNE est aussi un frein à l'établissement d'une optimisation conjointe.

L'objectif de ce rapport est donc de proposer un modèle PLNE de problème de lotissement intégrant des actions de maintenance en ayant les meilleures performances possibles en termes de temps de résolution et d'optimalité.

Après une introduction et un état de l'art sur les problèmes de planification de production et de maintenance, un focus sera fait sur une variante de ces problèmes, le problème de gestion de planification de production avec une capacité dépendante des actions de maintenance. Pour la résolution de ces problèmes, les solveurs CPLEX et Gurobi seront utilisés avec leur algorithme de séparation et de coupe.

Mots-clés: Planification de production, Problème de lotissement avec capacité, Maintenance intégrée, Optimisation

Abstract

With the growing interest of companies to be more productive and efficient in their process, optimization in production and maintenance planning is becoming more and more developed. They mainly consist in establishing for the first one maintenance slots and, for the second one, the quantities to produce at each time step (weeks or months). These plannings are usually optimized separately. The LSP has been widely studied with Mixed Integer Linear Programming (MILP) models describing it. It is still difficult to solve to optimality in a reasonable time large LSP instances without relaxing some constraints such as the obligation to satisfy demands at each period (possibilities to have a delay in the production). This is why many recent studies use heuristic methods to solve it, in order to have a 'good' solution (small gap with the optimal solution) in a reasonable time for these instances. The maintenance planning problem has for its part been mainly studied as a problem distinct from the production planning where it is tried to maximize machine availability (Garg et al. [1]). The difficulty to integrate maintenance actions in the production planning MILP model is also an obstacle to a joint optimization.

The objective of this report is to propose a MILP model of the Lot-Sizing Problem with integrated maintenance actions while having the best possible performances in terms of computation time and bound.

After an introduction and review on LSP and maintenance planning, we will focus on a variant of LSP, the Capacitated Lot-Sizing Problem (CLSP) with a capacity dependant of maintenance actions released in a finite period of time. As for the practical resolution of this problem, we will use the solvers Gurobi and CPLEX with their Branch-and-Cut (BC) algorithm.

Keywords: Production Planning, Capacitated Lot Sizing, Integrated Maintenance, Optimization

Acknowledgements

First of all, I would like to warmly thank my supervisors, Matthieu Gruson and François Lamothe who gave me the chance to live and work on this project in Montreal during six months. Moreover, they offer me a great support along my journey here, such technically with their expertise that for project sidelines.

I would also like to thank all the members of the CIRRELT laboratory for welcoming me and enabling me to work in a pleasant atmosphere. I particularly thank the Tarot league and Happy Wednesday members for all their actions to keep this atmosphere.

I thank my office mate Harcènage Dansou for his advices and his support along this research project as well as students doing their internship in CIRRELT and GERAD.

Finally, I want to express my gratitude to Marcel Mongeau for his visit and his constructive suggestions during this internship as well as Andrija Vidosavljevic for his feedbacks on my work.

Contents

List of Figures and Tables

Acronyms

BC Branch-and-Cut

CLSP Capacitated Lot-Sizing Problem

CENTOR CEntre de Recherche sur les Technologies de l'Organisation de Réseaux

CIRRELT Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport

CSLP Continuous Setup Lot Sizing Problem

CRT Centre de Recherche sur les Transports

DLSP Discrete Lot Sizing and Scheduling Problem

GLSP General Lot Sizing and Scheduling Problem

GPU Graphics Processing Unit

HEC École des Hautes Études Commerciales

LSP Lot-Sizing Problem

MILP Mixed Integer Linear Programming

PLNE Programmation Linéaire en Nombres Entiers

PLSP Proportional Lot Sizing and Scheduling Problem

PRP Production Routing Problem

ULSP Uncapacitated Lot Sizing Problem

UdeM Université de Montréal

UQAM Université du Québec à Montréal

WW Wagner-Whitin

Introduction

The aim of this research project is primarily to obtain MILP deterministic models including maintenance and production decisions. The objective is also to study the performance of each model. In addition, it should solve, thanks to general purpose solvers, instances with a number of periods and products sufficient for companies in a reasonable time.

Context

The Capacitated Lot-Sizing Problem (CLSP) is a classical problem in the literature of Operational Research which aims to optimize the production quantities of a factory in a given industrial field. In the CLSP, decisions have to be made on the quantity of items to produce at each period of a finite time horizon. These quantities are often limited by the production capacity of the factory. This capacity is generally defined as constant in time but there exists also another thought which defines it as decreasing depending on the number of items produced due to, for instance, the deterioration of machines. A mean to increase this capacity is to plan preventive maintenance operations which would be executed on machines. On the other side, maintenance planning has its own constraints. The problem consists in finding the periods where maintenance actions will be performed that minimize the maintenance costs or maximize the machine availability. The maintenance planning is a problem with a sufficient flexibility to consider a joint resolution with the production planning. Indeed, maintenance planning independently of the production - which is currently the classical way of modelling things in the literature - allows to minimize that induced maintenance impact without taking into account its impact on the production. Well, decisions on maintenance heavily influence the production capacity with potentially economic losses. A joint optimization would therefore allow companies to recover these losses. A parallel can be made between this problem of CLSP with integrated maintenance and the Production Routing Problem (PRP). In a PRP, the production and the delivery are optimized simultaneously. This also applies to our problem with the production and the maintenance instead of the delivery.

Objectives

The initial subject of this internship was to develop optimization models integrating production and maintenance decisions. This project was handled jointly with my colleague Harcènage Dansou. The project was therefore separated into two aspects, the modelling of the problem, and its resolution. On my side, I worked on the modelling aspects. Nevertheless, given that my work is linked with the one of Harcènage Dansou, the decisions made by me and my tutors on the directions this internship should take are not only due to my results but also to Harcènage Dansou's one.

As it will be further explained in chapter 1, a standard version of the Capacitated Lot-Sizing Problem (CLSP) with integrated maintenance was first introduced in 1990s (Lee [4]), while several variants were proposed in the following years (Aghezzaf, Jamali and Ait-Kadi [5], Aghezzaf, Sitompul and Najid [6], Aghezzaf, Khatab and Tam[7]). Therefore, our initial ambition was to develop a model able to provide the optimal solution for CLSP with integrated maintenance. By providing a compact model of CLSP with integrated maintenance, companies could be more interested to use it. As developed in chapter 1, the CLSP is widely studied, but its version with integrated maintenance is recent. The starting point for this internship heavily relies on a previous work of Van Vyve and Shamsaei [8], who have developed a CLSP model integrating maintenance with assumptions that are near the ones we use. The authors have put emphasis on the modelling aspects of different maintenance points of view without particularly studying the model strength. We have therefore chosen to study deeper the modelling of production planning starting from the models developed on the Lot-Sizing Problem in the literature. In the end, we have developed other modellings on the maintenance part.

The first step of the internship was consequently to perform a literature review in order to understand all the classical hypotheses made in the initial definition of a Lot-Sizing Problem (LSP). It also included the analysis of the existing variants of CLSP and the search on ways to integrate maintenance in this problem. Then, a closer look was given to the model of Van Vyve and Shamsaei [8]. Finally, alternative models were developed, from the definition of variables to the search of valid inequalities.

Report structure

The rest of the report is organized as follows. The first page describes the work environment of my internship. The literature review of previous related work is then detailed in Chapter 1.
Chapter 2 exposes the mathematical models used to address the CLSP with integrated maintenance and the rationale which led to consider them. It also introduces the notations used and important assumptions made.
Chapter 3 details a new maintenance formulation. A proof of the strength of this formulation is given as well as an example to illustrate it.
Chapter 4 presents another mean to improve our model with valid inequalities.
Experimental results are presented in chapter 5. The instances and parameters used are exposed before an analysis of the results obtained. Tables with all the results can be found in appendix.
Finally, conclusions are drawn, the difficulties encountered are outlined as well as the perspectives of future works.

Presentation of the work environment

This chapter describes my work environment during this internship. It took place in the research laboratory Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT) at Université de Montréal (UdeM).

CIRRELT presentation

The Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT) has been created in 2006 resulting of the merge of Centre de Recherche sur les Transports (CRT) and CEntre de Recherche sur les Technologies de l'Organisation de Réseaux (CENTOR). This research laboratory gathers 8 Montreal universities (ETS, Polytechnique Montréal, École des Hautes Études Commerciales (HEC) de Montréal, Université du Québec à Montréal (UQAM), UdeM, Mc Gill University, Laval University and Concordia University) allowing them to share their knowledge and competence in applied mathematics, mathematical engineering, computer sciences and industrial engineering. CIRRELT aims mainly at designing resilient and sustainable networks, planning large-scale networks under uncertainty, supporting multi-attribute and multi-objective decision-making and supporting network integration. [9]

The work environment

I had two supervisors at CIRRELT: Matthieu Gruson who is a researcher as well as a teacher at ESG at UQAM and François Lamothe who is a post-doc fellow at CIRRELT. They both followed my progress during this research project. Matthieu Gruson is an expert in Lot-Sizing Problem when François Lamothe adds his expertise on discrete optimization. I can also rely on him for specific questions on mathematical aspects or on coding.

I share my office with Harcènage Dansou who works on the same topic as me to wit CLSP with integrated maintenance. It is located at the 3^{rd} floor of the Pavillon Aisenstadt building in the campus of Université de Montréal (UdeM). I work on my laptop with a second screen landed by Université du Québec à Montréal (UQAM). I am also connected to the internal network so as to launch instances on CIRRELT's Graphics Processing Unit (GPU) with CPLEX and Gurobi licences.

1 State of the art

1.1 Introduction

The Lot-Sizing Problem (LSP) has been formulated for the first time by the economist Harris in 1913 with his *"Economic Order Quantity"* which aims to minimize the total costs engendered by setup and inventory on an infinite horizon time with a given annual deterministic demand. It has been widely studied since 1958 with the development of a lot-sizing model solved by dynamic programming by Wagner and Whitin [10]. Many variants have been created, considering one or several levels, a finite or infinite planning horizon, a production capacity or not, a deterministic or stochastic demand, an inventory shortage or not, a setup structure or not, etc. To start this literature review, classical assumptions made for LSP are developed.

1.2 Lot-Sizing Problems

According to Karimi et al. [11], the production planning can be broken down in three levels: long-term which is strategic, medium-term which is the tactical part and short-term which is the operational one. In long-term planning, we define a strategy to take the most benefits as possible of the predicted evolution of markets. The strategy is principally based on how the investments are managed and divided between equipment, process amelioration, item choices, etc. In medium-term, the required material is defined and production quantities (or lot sizes) are established. Finally, the short-term is daily decisions and adaptations to sudden problems which can occur. A great part of lot-sizing models concern the medium-term because it is mainly at this moment that the question of production planning is raised. In short-term, scheduling will be preferred to lot-sizing because the decisions on quantities to produce have already been taken. The classic LSP aims to determine the production-inventory strategy such that the sum of holding, production, and setup costs is minimized. In more details, it decides on the quantities produced at each period of a given finite and discrete time horizon. The Capacitated Lot-Sizing Problem (CLSP) is a variant of LSP where production is restricted by a given capacity that may vary during the time horizon which is a $\mathcal{NP}$-hard problem (Florian et al. [12]). It is first primordial to have a good comprehension of the particularities of the LSP in order to understand the CLSP we will study along this report. The following section develops the main specific features of LSP.

1.2.1 Lot-Sizing Problem hypotheses

Single or multi levels Equipment uses raw material to produce desired items. However, it can also need a good previously produced (component) by the same company as entry to produce the final item (for instance doors have to be produced to produce a car). In this case, there are therefore two levels of production, one for the component and one for the final item.

Single or multi-items The single item production system consists in planning the production of only one type of item (Dauzere-Peres [13]). On the contrary, in the multi-item case, different types of item are produced, possibly in the same period. The number of different items produced affects the LSP in several means. Indeed, the definition of decisions variables and parameters is impacted as well as the complexity of the problem (Karimi et al. [11]).

Planning horizon The planning horizon is the time interval on which the production plan extends into the future. It can be finite or infinite. A finite-planning horizon is usually accompanied by dynamic demand and an infinite planning horizon by stationary demand (Karimi et al. [11]). Depending on the time period considered, lot sizing problems will be called either big bucket or small bucket problems. In the big-bucket problems, it is possible to produce several items within the same period when in the small-bucket problems, only one item can be produced at each period.

Demand Demand type is considered as an input to the model of the problem. The demand can be static which means that its value does not change over time (also called stationary or constant demand). Demand can also be dynamic and in this case the demand for an item may differ from one period to another one. Nevertheless, in both cases, demand is known in advance and so they are deterministic. In other cases, demands can be stochastic, following a certain law of probability. This engenders more complex LSP compared to deterministic demand problems (Aloulou et al. [14], Akartunali et al. [15]).

The number of levels is also important for the demand type. If the problem considered has a single level, then demand will be independent given that requirements of an item do not depend on decisions regarding another item's lot size. In multi-level lot sizing, there is a parent–component relationship among the items. So the demand at one level depends on the demand for their parents and this demand is named dependent.

Inventory An inventory is always defined in a LSP. This inventory can have a maximum capacity and has usually an associated cost. This cost can represent the rental of a warehouse thefts, insurance, good depreciation, heating, etc. Other costs more hidden can be the loss of a part of the stocked production due to its nature (perishable items which can not support a long period in a warehouse), the conditions that have to be maintain to keep these items saleable (for instance a cold chain that should be maintained in the warehouse) (Önal et al. [16]) or to transport itself (delicate items) (Sarker et al. [17]).

Production capacity A LSP can be constrained by a production capacity. In this case, it is called a Capacitated Lot-Sizing Problem (CLSP) otherwise it is a Uncapacitated Lot Sizing Problem (ULSP). This constraint can be explained by machines used in the production line which have a limit in the number of units they can produce or in their available time. This capacity is the link between the production of each item type since they are produced by the same machines and so the capacity is given for a number of items whatever their type. The capacity can be define as a constant or it can be dynamically updated depending on the time the machine is used, the quantity produced by this machine or by the time passed since its purchase. Capacity restrictions has an important impact on the problem complexity (Karimi et al. [11]).

Setup structure When different items are produced by an equipment, some costs occur such as the time to adapt the equipment to desired item, the test of the initial output or the cost of changing a tool. A setup structure models this phenomenon by different ways. It is possible to penalize the number of different produced items with a setup cost in the objective in order to reduce the number of times production is changed (Jans et al. [18]). It is also possible to define a setup time which will reduce the total available production time. In case of a Capacitated Lot-Sizing Problem (CLSP), a setup loss of capacity can be defined as well in order to minimize the number of setup changes.

After this introduction on LSP possible assumptions, we will have a closer look on a variant which interests us more, the Capacitated Lot-Sizing Problem (CLSP).

1.2.2 Variants of CLSP

Regarding all the possible assumptions available for a LSP, there are several variants of the LSP with a capacitated resources constraint studied in the literature [19]. Ramya et al. [2] show these variants on figure 1.1. The most common ones are presented in this subsection.

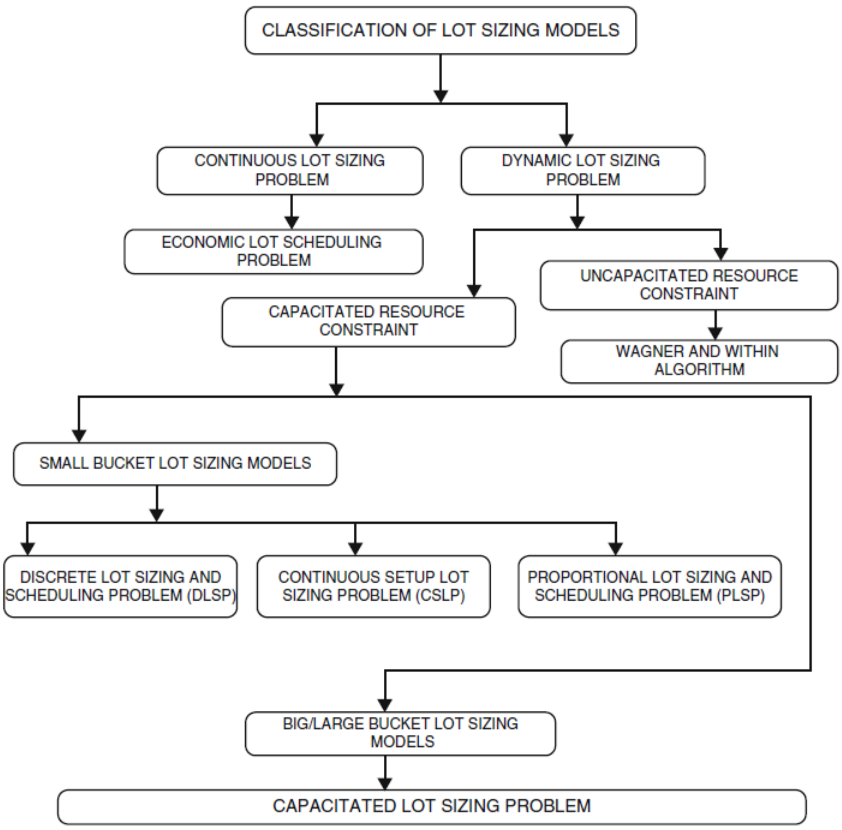

Figure 1.1: Classification of lot-sizing models [2]

1.2.2.1 Discrete Lot Sizing and Scheduling Problem (DLSP)

The multi-item single-resource discrete lot-sizing and scheduling problem or DLSP is a variant of CLSP where the periods are divided into several smaller periods. The LSP part is dedicated to find the right amount to produce at each period when the scheduling is developed to find when and how each item should be produced (Dauzere-Peres [13]). The DLSP relies on a fundamental discrete production policy: it assumes that at most one item can be produced at each period and that the resource either processes one item at full capacity or is completely idle (Gicquel et al. [20]). The costs to be minimized are the inventory holding costs and the changeover costs. A MILP model of the DLSP is proposed in Brüggemann and Jahnke [21].

1.2.2.2 Continuous Setup Lot Sizing Problem (CSLP)

The CSLP is close to DLSP since it takes the same idea. Its difference lies in the relaxation of the assumption forcing the production to be at full capacity or to be stopped. Drexl and Kimms [22] illustrate the differences by presenting a classical modelling of each of the problems. In other words, when the setup of an item is activated, this item can be produced without consuming all the capacity, which is not the case in the DLSP where all the capacity must be used. However, it is still possible to produce at most only one item per period.

1.2.2.3 Proportional Lot Sizing and Scheduling Problem (PLSP)

The basic idea behind the PLSP is to use the remaining capacity in CSLP for scheduling a second item in the particular period, if the capacity of a period is not used in full. The underlying assumption of the PLSP is that the setup state of the machine can be changed at most once per period. Following this, at most two items may be produced per period. A detailed model is given in Drexl and Kimms [22].

1.2.2.4 General Lot Sizing and Scheduling Problem (GLSP)

The GLSP starts from the same point as DLSP. The underlying idea comes this time from lot sizing with stationary demand, where each lot is uniquely assigned to a position number in order to define a sequence (Zipkin [23]). In the GLSP, the main assumption is that a parameter defines a maximum number of lots that can be produced per period. Fleischmann and Meyr [24] define this problem and propose a model for it. It is also proved that the GLSP with non-zero minimum lot-sizes is $\mathcal{NP}$-complete.

1.3 Maintenance

In this section, the main interest is the other part of our problem: the maintenance. Maintenance planning has also been widely studied because of costs it requires as much as savings it can allow. It has been particularly the case in refineries and petrochemical complexes, motivated by the cost driven optimization of the existing large base of older plants (Goel et al. [25]). The problem of maintenance optimization is to identify the maintenance policy that optimizes the balance between maintenance benefits and costs. Goel et al.[25] represent this mechanism on Figure 1.2.

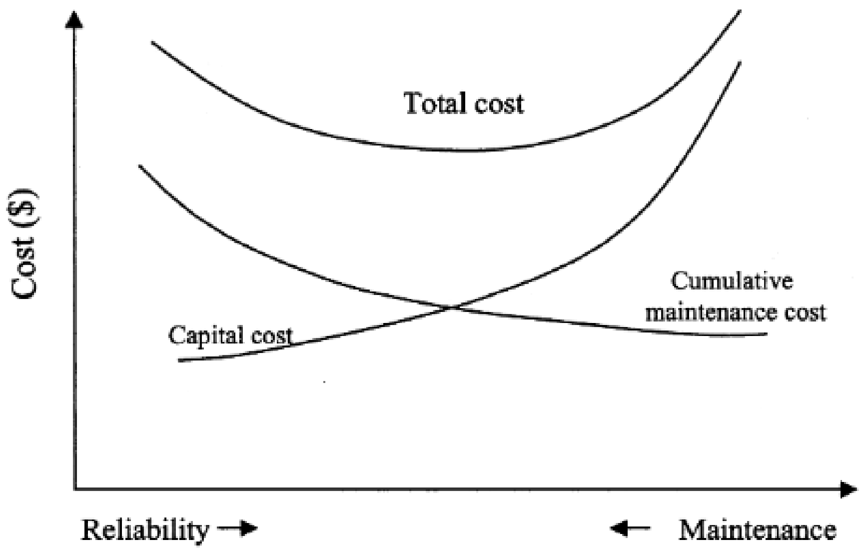

Figure 1.2: Impact of maintenance on the total cost

In the review done by Garg and Deshmukh [1], maintenance models built between the early 1990s to 2003 are presented. They present a classification of models depending on the types of maintenance considered and on how the maintenance is integrated in models. Critical observations are added in order to justify why one model fits more than an other in a particular context. Maintenance scheduling can be divided in five main parts (Garg et al. [1]): Condition-Based Maintenance (CBM), Corrective Maintenance (CM), Preventive Maintenance (PvM), Predictive Maintenance (PdM) and Palliative Maintenance (PaM). CBM focuses on sensors which give equipment state and allow to know when maintenance actions have to be performed. CM is the simplest maintenance policy, it consists in waiting until a failure occurs to maintain equipment. CM can also be performed when an unexpected failure occurs although other maintenance policy are applied. In PvM, we seek out and repair more minor issues principally by observation in order to decrease the occurrence of major repairs. PdM consists in analyzing how equipment deteriorates in order to define the best maintenance schedule (so a planning which avoids failures and lack of productivity without an excessive amount of maintenance actions). Finally, PaM allows to repair momentarily an equipment with available resources in order to continue to produce despite the downgrade of the equipment due to a bad reparation. It gives time to go back to the nominal production without completely stopping a machine potentially essential for the production.

PvM is the privileged policy in maintenance modelling because of its "proactive" side compared to CM and CBM. It is also easier to model than PdM due to the difficulty to model the state of machines and to define a maintenance periodic cycle from it (Chu et al. [26]), and than PaM which has not been modelled in any maintenance planning problem to the best of our knowledge. The impact of machine breakdowns on production time and the consideration of CM has first been considered by Gronevelt et al. [27] but it has quickly been abandoned in aid of PvM. The question raised by this last model is how often a maintenance should be performed. Many works have considered a cyclic maintenance (detailed in subsection 1.4.1) in their model in order to get a maintenance planning easier to follow in practice and because it can be applied until a very long time horizon (Grigoriev et al. [28]). Aghezzaf et al. [5] presented, after having studied different failure rate distributions, a model which introduces a CLSP with a decreasing production capacity over the time if no maintenance actions are performed.

1.4 Integrated maintenance in CLSP

In this section we will discuss the integration of maintenance into the CLSP. According to Gao et al. [29], the joint optimization of maintenance and production planning allows to reduce the possibility of system's future failures. It is also a way to reduce maintenance costs which represent between 15% and 40% depending on the fields (Chu et al. [26]). By combining them instead of optimizing production planning and maintenance planning separately, Schreiber et al. showed that benefits can be rather important [30]. Considering a joint optimization with an impact of maintenance on the production time, they obtained that the average setup and maintenance time is reduced by about 18% from 5 hours to 4.1 hours. The following sections present a review on how maintenance is integrated in the joint production and maintenance planning problem. Two cases are considered: the cyclic maintenance models and the non-cyclic ones.

1.4.1 Cyclic maintenance in CLSP

Aghezzaf et al. [5] worked on a multi-item Capacitated Lot-Sizing Problem (CLSP) in which the system is maintained periodically with preventive and corrective maintenance actions. The two main objectives of their model are determining the quantities of units to produce for each item and determining the optimal step k which defines the periodic maintenance. In order to realize this, the authors have proposed a Mixed Integer Linear Programming (MILP) model in which the deterioration of the system is modeled by a reduction in capacity based on the time passed since the last maintenance action was performed (preventive maintenance), and a rate of failure which follows a distribution of probability (corrective maintenance). They also propose an iterative algorithm based on the MILP model (Aghezzaf et al. [5]). Indeed, by deciding the fixed maintenance step k, the capacities of each period and the cost of maintenance are easily determined with a classic solver since the problem consists now in solving classic CLSP (CPLEX here). In the end, the value of k giving the best possible solution is selected. It is worth recalling that the CLSP is $\mathcal{NP}$-hard so the complexity of the iterative algorithm depends on the complexity of the CLSP.

Aghezzaf, Sitompul and Najid [6] investigate the optimization of both maintenance and production planning within a multi-line production system. Their work builds upon prior research by Aghezzaf, Jamali and Ait-Kadi [5], which already considered cyclic preventive maintenance. However, Aghezzaf, Sitompul and Najid extend this by incorporating machine-to-task assignments for production quantities. To tackle this issue, they present a model formulated as a Mixed Integer Linear Programming (MILP). This model accounts for preventive maintenance cycles initiated at the beginning of planning periods, with the stipulation that maintenance actions must be concluded within the same period. The authors propose two distinct solution approaches: an exact algorithm and a Lagrangian relaxation-based algorithm. The exact algorithm is iterative and systematically explores all possible maintenance plans for each machine. This involves calculating available capacities throughout the planning horizon for each cyclic maintenance pattern. Subsequently, a Capacitated Lot-Sizing Problem (CLSP) is solved to determine feasible solutions. Notably, the number of potential combinations of preventive maintenance cycle sizes grows exponentially, with a complexity of order T^m, where T is the number of periods and m is the number of machines. Consequently, the complexity of this procedure depends on the complexity of the algorithm used for the underlying CLSP. In the solution procedure based on Lagrangian relaxation, a new variable is used to reformulate the initial model and to allow a Lagrangian relaxation on capacity constraints. The problem is therefore divided into two groups of sub-problems. The first group is a set of production planning problems with integrated cyclic preventive maintenance on each single machine. The second part aims to determine the best assignment of the demand d_{it} to machines, for each item i at each period t. In their numerical experiments, Aghezzaf, Sitompul and Najid [6] use the exact algorithm to solve the single-line first group of sub-problem. A set of 720 problems are tested with the proposed Lagrangian-based solution and they obtained a very good upper bound on the optimal value. The gap is also calculated, varying from 2% for loose capacity problems to 16% when capacity is tight.

Van Vyve and Shamsaei [8] show how to formulate production planning problem with an integrated maintenance for a single machine as a Mixed Integer Linear Programming (MILP). They model initially the deterioration as a linear reduction in the nominal capacity of the machine at each period with a reduction coefficient of $0 \leq \alpha \leq 1$. They do not take corrective maintenance into account in contrary to Aghezzaf, Jamali and Ait-Kadi [5] and Aghezzaf, Sitompul and Najid [6]. However, they assume that the capacity can be reduced by a term proportional to the quantity of produced items so that capacity depends on the age of machine since its last maintenance action as well as its production pace. They also introduce a decrease of capacity modelling so that when a machine starts a production batch, or when a machine switches from one item to another the capacity is a bit consumed. The deteriorations that can appear are also included in this decrease of capacity. With all these details taken into account, they tried to have a model as close as possible to the industrial realities where, for example, starting an engine can reduce its production capacity since this capacity is often based on a number of starts. The authors compare in this article a cyclic maintenance program approach (CMP) and a non-cyclic maintenance program approach (NCMP). As with Aghezzaf et al. [5], the CMP approach is a question of finding a step k optimal for the periodic maintenance by adding constraints to the model. In the NCMP approach, maintenance actions are more flexible but it makes the model harder to solve compared to the CMP approach. According to Van Vyve and Shamsaei [8], it can allow substantial benefits in terms of costs (savings up to 41%) but the limits are reached with instances containing 10 items and 24 periods. This difference is also explained by Fitouhi et al. [3] on Figure 1.3. In the case of the cyclic preventive maintenance, the demand is taken into account at maintenance planning stage and so maintenance is planned considering the future demands.

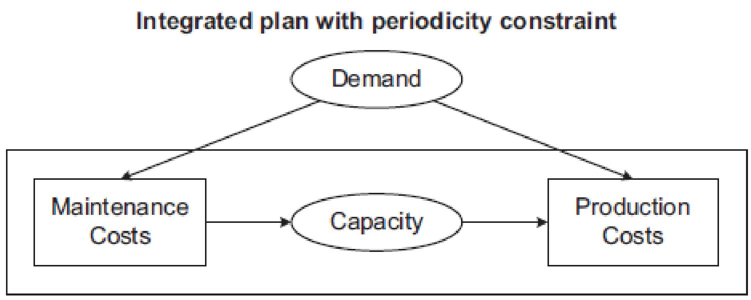

Figure 1.3: Demand influence on the production plans with cyclic preventive maintenance [3]

Nevertheless, it does not take into account the possible available capacities at the end of each period due to the cycle. This adaptation is made while the maintenance can be performed at any period and in particular when capacity is not sufficient for the upcoming demands after it has been fully used in previous periods. This non-cyclic maintenance will therefore be developed in the following subsection.

1.4.2 Non-cyclic maintenance in CLSP

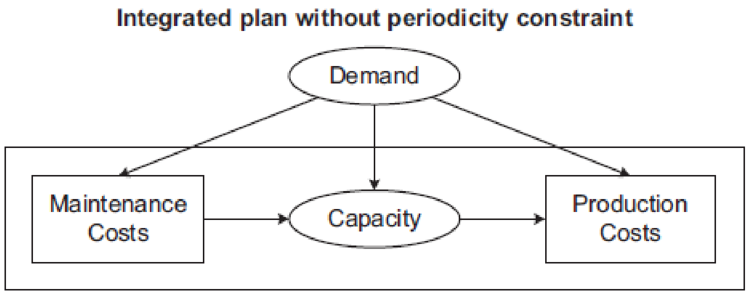

Figure 1.4: Demand influence on the production plans with non-cyclic preventive maintenance [3]

Fitouhi et al. [3] developed a model integrating non cyclical preventive maintenance and tactical production planning for a single machine. They assume that backlogging is allowed and so demands can be satisfied later than they should be in exchange of an additional fee. The maintenance policy suggests possible preventive replacements at the beginning of each production planning period, and minimal repair at machine failure as in Aghezzaf, Jamali, Ait-Kadi [5] and Aghezzaf, Sitompul, Najid [6]. The deterioration of the capacity is modeled with a Weibull distribution and the CLSP model is an Integer Linear Program instead of a MILP. The authors developed an algorithm which enumerates the possible maintenance plans and proceeds to the solving of Capacitated Lot-Sizing Problems until all combinations have been tested (2^{T-1}, where T is the number of periods).

Zhu developed an integrated data-driven model that coordinates maintenance planning decisions with production scheduling decisions to solve the problem of scheduling and maintenance planning for a parallel-series production line [31]. He introduced a minor repair, a replacement and a preventive maintenance which can happen depending on a failure rate for the first two, and depending on accumulative processing time for the last one. The proposed model is a MILP which considers several machines and a capacity limited in maximum production time instead of a maximum number of units. This allows to define a decrease of the capacity which depends on the real total production time since the last maintenance action instead of defining the decrease of capacity at each period of time independent of the number of the produced units. The author mainly shows that a good maintenance plan, integrated to production plan can lead to substantial savings. To demonstrate it, an instance of 8 items and 30 periods is solved with an optimization of the production planning and the maintenance planning separately as well as a joint one.

Sloan et al. developed a Markov decision process based-model of a single-machine production system with multiple items and with multiple machine states [32]. The problem is to determine a production and maintenance schedule that maximizes long-run expected average profit. The condition, or state, of the machine deteriorates over time, and this condition affects the probability of successfully producing the various items. In the context of the paper, i.e., a semiconductor industry, maintenance is reduced to a cleaning but it is more the time to disrupt production and to open the machine chambers that is taken into account in the maintenance cost. Here, the deterioration of machines is modeled with states of a Markovian process. State matrices are defined for each machine and as the machine deteriorates, rates to switch to the worst states increase. In this article, demands are assumed to be random, independent and identically distributed. A constrained Markov decision process model is then obtained and divided in two sub-problems. The first is a maintenance planning problem and the second is a production planning problem. Three methods are proposed to solve the two sub-problems in a slightly different way. The first one is to first solve the maintenance planning problem and then the production planning one with the rule that when a decision to produce is made, we chose which item to produce on a first-come, first-serve basis. This is the traditional approach. The second method consists in the same idea with, instead of having a first-come, first-serve basis, a defined probability to obtain each item by the fact that a certain proportion of total production must consist of a particular item at each period in average. The last method considers the same production decision policy as the method 2 but problems are simultaneously solved this time. In their experimental results, they obtain on a set of 2,970 test problems that the third method is on average 25% better than method 1 and 5% better than method 2.

Another solution to model maintenance in production planning is to look at machine age instead of modelling a decreasing capacity in order to schedule a PvM. Absi et al. [33] [34] study an integrated lot-sizing and maintenance planning problem on a single machine. Their version defines a maintenance that can be executed only when a minimum age of the machine is reached and before a maximum age, production setups that generate fixed age setups for the machines, and a maintenance that can be performed at any point in time. Although they production capacity is unlimited, the following constraint is implicitly given because the machine age is given in terms of produced units: a maximum number of items is set between two maintenance operations. It can be seen as a production capacity but more flexible than the one defined by Aghezzaf et al. [5]. A predefined maximum allowed age activates a maintenance when it is reached and a minimum one prevents to maintain equipment to early. The authors showed that their problem is a generalization of a capacitated lot-sizing problem in the single-item case which can be solved in polynomial time by dynamic programming when certain parameters are time-invariant in the single-item case and becomes strongly $\mathcal{NP}$-hard with multiple items.

1.5 Conclusion

After this literature review on Capacitated Lot-Sizing Problem (CLSP), maintenance modelling aspects and on the joint optimization of production planning and maintenance planning, all the required knowledge to understand the models of CLSP with integrated maintenance has been given. Different mathematical models are discussed in the next chapter. They are mainly based on the existing literature on CLSP. It will allow us to go deeper in the modelling of CLSP with integrated maintenance.

2 Mathematical models

For our study, we consider an integrated multi-item Capacitated Lot-Sizing Problem (CLSP) with preventive maintenance for a single machine.

In this chapter, three different models are presented with some particularities in their formulation but modelling the same CLSP. After having studied the model proposed by Shamsaei and Van Vyve [8], the plant location formulation of Krarup and Bilde [35] is developed to finally present the Eppen and Martin's one [36] (shortest path formulation).

First, let us define $T = \{0, ..., n\}$, the planning horizon where n is the end of the period horizon considered. As the focus is made on medium-term decisions, T will vary between 6 months and a year which is the average vision for a company concerning tactical planning. It will be discretised by week as it is often the case in industrial production, shorter period steps resulting more of scheduling problems (see section 1.2.2.1). Considering these choices, we want the solvers to be able to solve instances where n reaches at least 25 and ideally around 50. We also define the set of the different items produced as $P = \{1, ..., m\}$ where m is the number of different goods produced. Notations are used together with the following assumptions:

- Demand d_{it} is a time-varying but deterministic demand for item $i \in P$ in period $t \in T$.
- Production costs p_{it} depend on items $i \in P$ and periods $t \in T$.
- We consider a setup cost f_{it} which applies if at least one unit of a item $i \in P$ is produced in period $t \in T$. It represents the cost of adapting the machine to the production of this item.
- Backlogging is not allowed meaning that demand must be satisfied at the right time.
- There is only a single level of production.
- The capacity decreases with a rate of α from one period to another. It represents the equipment degradation due to a nominal use. Moreover, each item $i \in P$ has a coefficient β_i associated in order to model the degradation this specific item engenders on the capacity. This last degradation is linear with the number of units produced and it can be explained by an excessive use of a part of the machine or to constraints applied on the machine that are near its limits.

It is now possible to define the parameters which will be used along this report:

- d_{it}: Demand of units of item $i \in P$ in period $t \in T$

- f_{it}: Setup cost of item $i \in P$ in period $t \in T$
- p_{it}: Variable production cost by unit of item $i \in P$ in period $t \in T$
- h_{it}: Holding cost by unit of item $i \in P$ at the end of period $t \in T$
- m_t: Maintenance cost in period $t \in T$
- α: Coefficient of capacity deterioration
- β_i: Deterioration of production capacity due to production $i \in P$
- c_{max}: Maximum capacity allowed by the machine

2.1 Initial model

Shamsaei and Van Vyve[8] model has been the starting point for our work. Their model is given below. This model is called **(ICLSP)** in the rest of the report.

2.1.1 Model

The following decision variables are defined:

- x_{it}: number of units of item i produced in period t
- y_{it}: $\begin{cases} 1 & \text{, if at least one item } i \text{ is produced in period } t \text{ (if } x_{it} > 0) \\ 0 & \text{, otherwise} \end{cases}$
- I_{it}: Inventory of units of item i at the end of period t
- z_t: $\begin{cases} 1 & \text{, if there is a maintenance in period } t \\ 0 & \text{, otherwise} \end{cases}$
- c_t: production capacity in period t

So it is now possible to describe this CLSP model:

$$\textbf{(ICLSP)} \quad \min \quad \sum_{t \in T} \sum_{i \in P} (f_{it} y_{it} + p_{it} x_{it} + h_{it} I_{it}) + m_t z_t \tag{2.1a}$$

$$\text{s.t.}$$

$$x_{it} + I_{i(t-1)} = d_{it} + I_{it} \qquad i \in P, t \in T, \tag{2.1b}$$

$$x_{it} \leq (\sum_{s=t}^{T} d_{is}) y_{it} \qquad i \in P, t \in T, \tag{2.1c}$$

$$\sum_{i \in P} x_{it} \leq c_t \qquad t \in T, \tag{2.1d}$$

$$c_t = max(\alpha c_{t-1} - \sum_{i \in P} \beta_i x_{i,t-1}, c_{max} z_t) \qquad t \in T, t \geq 1, \tag{2.1e}$$

$$x_{it}, I_{it} \in \mathbb{R}^+ \qquad i \in P, t \in T, \tag{2.1f}$$

$$c_t \geq 0 \qquad t \in T, \tag{2.1g}$$

$$y_{it} \in \{0, 1\} \qquad i \in P, t \in T, \tag{2.1h}$$

$$z_t \in \{0, 1\} \qquad t \in T. \tag{2.1i}$$

The objective (2.1a) minimizes the sum of setup costs, production costs, holding costs and maintenance costs. Constraint 2.1b forces the production to satisfy the demand for each item

$i \in P$ and each period $t \in T$. Constraint (2.1c) is a big-M constraint that forces the setup variable $y_{it}, i \in P, t \in T$ to take the value one if there is a strictly positive production, i.e., if $x_{it} > 0$ in order to satisfy the constraint. Otherwise, when there is no production, the constraint will always be satisfied and, because of the minimization of the objective, y_{it} will take the value 0. The best big-M value is the smallest one so $M = \sum_{s=t}^{T} d_{is}$ because the production will never exceed the remaining demands until the end of the planning horizon.

Constraints (2.1d) and (2.1e) define the production capacity. (2.1d) ensures that the total production over all items does not exceed the production capacity for each period $t \in T$ whatever the type of item it is. (2.1e) is linearized and explained in the following subsection.

Finally, constraints (2.1f), (2.1g), (2.1h) and (2.1i) define the domain of definition of each variables.

2.1.2 Linearization

Constraint (2.1e) can be linearized as follows:

$$c_t \leq \alpha c_{t-1} - \sum_{i \in P} \beta_i x_{i,t-1} + c_{max} z_t \quad t \in T, t \geq 1 \tag{2.2}$$

$$c_t \leq c_{max} \quad t \in T \tag{2.3}$$

Constraints (2.2) and (2.3) replace constraint (2.1e). When there is a maintenance, $z_t = 1$ and c_t is limited to c_{max} with (2.3) but nothing will guarantee that it will reach it since $\alpha c_{t-1} - \sum_{i \in P} \beta_i x_{i,t-1}$ can be negative. We already know that $c_{t-1} \geq \sum_{i \in P} x_{i,t-1}, \forall t \in T, t > 1$. Along this report, $\beta_i, i \in P$ will have a 'small' value compared to the one of α in order to have $\alpha \geq \frac{\sum_{i \in P} \beta_i x_{i,t-1}}{\sum_{i \in P} x_{i,t-1}}, \forall t \in T, t > 1$.

Expression (2.2) can be separated into two different parts. The term $\alpha c_{t-1} + c_{max} z_t$ represents a decrease of the capacity until the next maintenance of the equipment. It decreases with time and so it is based on the age of the machine. The other term $\sum_{i \in P} \beta_i x_{i,t-1}$ represents the linear decrease of capacity associated to the produced items.

2.2 Plant location model

Starting from the previous model, we wonder how the model could be improved. The first idea was to change the definition of the production variables x. Krarup and Bilde [35] formulated an extended formulation for their plant location problem which can be applied to our problem. The idea is to define x with the information of the period for which a item is produced. Even though it increases the total number of variables and of constraints, it also enables the deletion of the variables I_{it} since x is sufficient to obtain all information required. Various theoretical and computational results concerning such reformulations (this one and the shortest path formulation of the next section) have been published in the literature (Fiorotto et al. [37]). The linear relaxations of these alternative formulations are stronger than of the initial formulation presented in section 2.1.

x is therefore defined as follows:

- x_{ikt}: number of units of item i produced in period k to satisfy the demand of period t

The other variables sued in this model were defined in section 2.1. Objective and constraints (2.1b), (2.1c), (2.1d), (2.1f) and (2.2) are thus transformed as follows:

$$\textbf{(PLCLSP)} \min \quad \sum_{t\in T}(\sum_{i\in P}(\sum_{k=0}^{t}(\sum_{j=k}^{t} h_{ij} + p_{ik})x_{ikt}) + f_{it}y_{it}) + m_t z_t \tag{2.4a}$$

s.t.

$$\sum_{k=0}^{t} x_{ikt} = d_{it} \qquad i \in P, t \in T, \tag{2.4b}$$

$$x_{ikt} \leq d_{it} y_{ik} \qquad i \in P, k \in T, t \in T, t \geq k, \tag{2.4c}$$

$$\sum_{i\in P}\sum_{j=t}^{n} x_{itj} \leq c_t \qquad \forall t \in T, \tag{2.4d}$$

$$c_t \leq \alpha c_{t-1} - \sum_{i\in P}\beta_i(\sum_{j=t-1}^{n} x_{i,t-1,j}) + c_{max} z_t \qquad , t \in T, t \geq 1, \tag{2.4e}$$

$$x_{ikt} \in \mathbb{R}^+ \qquad i \in P, k \in T, t \in T, t \geq k, \tag{2.4f}$$

$$c_{max} \geq c_t \geq 0 \qquad t \in T, \tag{2.4g}$$

$$y_{it} \in \{0, 1\} \qquad i \in P, t \in T, \tag{2.4h}$$

$$z_t \in \{0, 1\} \qquad t \in T. \tag{2.4i}$$

In (2.4a), since the inventory value is not accessible directly from a variable, each production variable x_{ikt} is penalized with its associated variable production cost as well as its inventory cost which is known (holding cost between periods k and t to apply) which is equivalent to the initial penalization.

2.3 Shortest Path model

This model is inspired from the shortest path formulation of Eppen and Martin [36]. It is based on the principle of flow conservation, detailed in the following subsection.

2.3.1 Flow conservation principle

The principle of flow conservation essentially means that a node's incoming flows are equal to its outgoing flows. This notion is widespread in science. Figure 2.1 represents this principle.

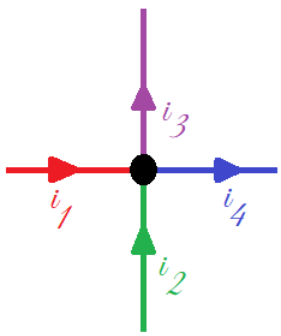

Figure 2.1: Representation of the flow conservation principle

On this figure, $i_1 + i_2 = i_3 + i_4$ according to the flow conservation principle. This notion of flow without any loss across nodes it goes through will be used for our model. The following subsection explains better how we model this flow and what it represents.

2.3.2 Model

Let ϕ be defined as:

- ϕ_{ikt} the percentage of accumulated demands of item i between periods k and t produced in period k

To have a better comprehension of ϕ, Figure 2.2 shows from which demands each ϕ takes its value with $P = \{1, 2\}$ and $T = \{0, 1, 2, 3\}$. Each color is associated to a given ϕ_{ikt}. Here, only the time step $t = 3$ is represented but note that the same principle can be applied for $t = 0, t = 1$ and $t = 2$ to have the whole example.

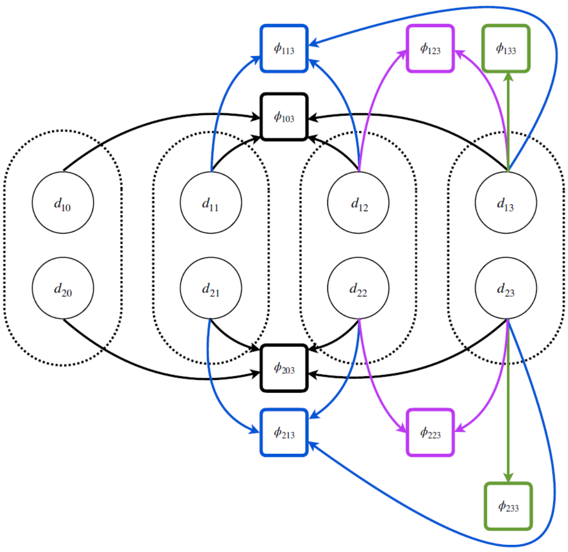

Figure 2.2: Incoming demands of ϕ_{ikt}, $i \in \{1, 2\}, k \in \{0, 1, 2, 3\}, t = 3$

In this model, objective and constraints (2.1b), (2.1c) and (2.1f) are thus transformed as follows:

$$
\begin{aligned}
(\mathbf{SPCLSP})\ \min \quad & \sum_{t \in T}\Big(\sum_{i \in P} f_{it} y_{it} + \big(\sum_{j=t}^{n} h_{ij} + p_{it}\big) x_{it} - h_{it} \sum_{j=0}^{t} d_{ij}\Big) + m_t z_t && && (2.5a)\\
\text{s.t.} \quad & && && \\
& \sum_{k=0}^{n} \phi_{i0k} = 1 && i \in P, && (2.5b)\\
& \sum_{k=0}^{t-1} \phi_{ikt} - \sum_{j=t}^{n} \phi_{itj} = 0 && i \in P, t \in T, t \geq 1, && (2.5c)\\
& \sum_{\substack{\tau=k \\ \sum_{j=k}^{\tau} d_{ij} > 0}}^{n} \phi_{ik\tau} \leq y_{ik} && i \in P, k \in T, && (2.5d)\\
& \sum_{i \in P} x_{it} \leq c_t && t \in T, && (2.5e)\\
& c_t \leq \alpha c_{t-1} - \sum_{i \in P} \beta_i x_{i,t-1} + c_{max} z_t && t \in T, t \geq 1, && (2.5f)\\
& \sum_{k \in T, k \geq t} \phi_{itk}\big(\sum_{j=t}^{k} d_{ij}\big) = x_{it} && i \in P, t \in T, && (2.5g)\\
& 1 \geq \phi_{ikt} \geq 0 && i \in P, k \in T, t \in T, t \geq k, && (2.5h)\\
& x_{it} \in \mathbb{R}^{+} && i \in P, t \in T, && (2.5i)\\
& c_{max} \geq c_t \geq 0 && t \in T, && (2.5j)\\
& y_{it} \in \{0, 1\} && i \in P, t \in T, && (2.5k)\\
& z_t \in \{0, 1\} && t \in T. && (2.5l)
\end{aligned}
$$

First, because of the missing information with x_{it}, objective can only be written by replacing the inventory by the current total production minus the current total demand which leads to the expression (2.5a). Constraints (2.5c) ensures that the entering flow (100% of demand) will be equal to the outgoing one and constraints (2.5b) is the initial conditions for this flow. Constraint (2.5d) has the same idea than (2.1c) but it does not require a big-M since $\phi \in [0, 1]$. However, this constraint requires the cumulative demands between the considered period k and t to be strictly positive. The considered period is the first setup period for the item i. If the condition is not satisfied, invalid inequalities would be created since in previous periods, ϕ_{ikt} equals 0 and so the initial flow of 1 would have been lost and reduced to 0 at the period k. Equation (2.5g) relates the new variable ϕ_{ikt} to the original production variable x_{it} of the initial model **(ICLSP)**$^{2.1}$ in order to make the model easier to understand. Then, constraints (2.5e), (2.5f), (2.5j), (2.5k), (2.5l) stay respectively the same as (2.1d), (2.2), (2.1g) and (2.3), (2.1h), (2.1i). Finally, (2.5h) and (2.5i) define the domain of definition of ϕ and x.

3 A new formulation for maintenance

In order to improve our model, we focused our work on the improvement of the linear relaxation of our model. Harcènage Dansou worked on a Dantzig-Wolfe decomposition of the maintenance part and I tried a formulation of maintenance with more variables in the trend of the plant location formulation (we will call it extended maintenance formulation). We finally proved that the linear relaxations of these two formulations lead to the same quality of linear relaxation. Given that the Dantzig-Wolfe decomposition gives already the best possible linear relaxation on the maintenance part, the extended maintenance formulation gives the best possible linear relaxation too. In this chapter, the initial model **(ICLSP)** defined in section 2.1 is considered. It is also admitted that $\beta_i = 0, \forall i \in P$ in this model and that a maintenance action is always performed at $t = 0$ in order to have a full capacity at $t = 1$.

3.1 Maintenance definition

Another idea to obtain a model with a better linear relaxation is to create a model with more maintenance variables. This allows us to embed more information in the variables which leads to a tighter relaxation:

- $z_{kt} = \begin{cases} 1 & \text{, if the last maintenance before period } t \text{ was performed in period } k \\ 0 & \text{, otherwise} \end{cases}$

Objective (2.1a) and constraint (2.1e) can therefore be transformed in the initial model as:

$$
\begin{aligned}
\textbf{(ICLSPM)}\ \min\quad & \sum_{t\in T}\sum_{i\in P}(f_{it}y_{it}+p_{it}x_{it}+h_{it}I_{it})+m_t z_{tt} && && \text{(3.1a)}\\
\text{s.t.}\quad & && &&\\
& x_{it}+I_{i(t-1)}=d_{it}+I_{it} && i\in P, t\in T, && \text{(3.1b)}\\
& x_{it}\le(\sum_{s=t}^{T} d_{is})y_{it} && i\in P, t\in T, && \text{(3.1c)}\\
& \sum_{i\in P} x_{it}\le c_t && t\in T, && \text{(3.1d)}\\
& c_t=c_{max}\sum_{k=0}^{t}\alpha^{t-k}z_{kt} && t\in T, && \text{(3.1e)}\\
& z_{kt}\le z_{kk} && k\in T, t\in T, t\ge k+1, && \text{(3.1f)}\\
& \sum_{k=0}^{t} z_{kt}\le 1 && t\in T, && \text{(3.1g)}\\
& x_{it}, I_{it}\in\mathbb{R}^+ && i\in P, t\in T, && \text{(3.1h)}\\
& c_t\ge 0 && t\in T, && \text{(3.1i)}\\
& y_{it}\in\{0,1\} && i\in P, t\in T, && \text{(3.1j)}\\
& z_t\in\{0,1\} && t\in T && \text{(3.1k)}
\end{aligned}
$$

The objective (3.1a) has $z_{tt}, t\in T$ as a costly maintenance action since other $z_{kt}, (k,t)\in T, k<t$ only indicates that the previous maintenance action was performed at period k. Constraint (3.1e) ensures the same capacity constraint as constraint (2.1e) but an equality constraint can be defined since the period where a maintenance action has been done is known thanks to the information added by this formulation. Constraint (3.1f) ensures that if period t considers that the last maintenance was done in period k then there must be a maintenance action in period k. This enables to link all the z_{kt} variables to the maintenance cost since only the z_{kk} variables appear in the objective function. Finally, constraint (3.1g) authorizes z_{kt} to be equal to 1 only once for each period t since there is only one period at which the last maintenance has been performed before t.

3.2 Strength of the extended maintenance formulation

Introduction Along these paragraphs, it will be demonstrated that the extended maintenance formulation gives a relaxation equivalent to the column generation one. It will first be shown that the polyhedron of the column generation formulation is included in the extended maintenance formulation polyhedron, then it will be proven that a point in the extended maintenance formulation can always be expressed by a combination of points resulting from column-generation formulation. The pseudo-code associated to this decomposition will be given, as well as the demonstration of the equality of the costs and capacities given by both formulations. Finally, an example will be detailed in order to have a better comprehension of the algorithm explained in the second and third paragraphs.

Assumptions. In this chapter, the initial model defined in (2.1) is considered. It is also admitted that $\beta_i=0, \forall i\in P$ in this model and that a maintenance action is always performed at $t=0$ in order to have a full capacity at $t=1$. Note that the column-generation formulation is in annex B.

Proposition 1. *Let V_1 be the value of the linear relaxation of the extended maintenance formulation and V_2 be the value of the linear relaxation obtained with the column-generation formulation, then $V_1 = V_2$.*

Proof. Let R be the polyhedron induced by constraints (2.1b), (2.1c), (2.1d), 2.1f, (2.1g), (2.1h) and let Q_1 be the polyhedron:

$$Q_1 = \{z_{kt} \in [0,1], \text{ satisfying constraints (3.1e), (3.1f) and (3.1g)}\} \tag{3.2}$$

The solution space of the linear relaxation of the extended maintenance formulation is $R \cap Q_1$. Meanwhile, the maintenance pattern formulation 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 integer points of Q_1. In order to show that the value of both linear relaxations is the same, we will show that $Q_1 = Q_2$.

The inclusion $Q_2 \subseteq Q_1$ is implied by the fact Q_2 is the result of the Dantzig-Wolfe decomposition of Q_1. Thus, let us now show $Q_1 \subseteq Q_2$.

To that end, consider an assignment of the variables z_{kt} satisfying constraints (3.1e), (3.1f) and (3.1g). We will show that there exists an assignment of the variables z_p of the maintenance pattern formulation inducing the same maintenance actions, the same capacity and cost. We now present an algorithm to compute such an assignment.

In what follows, let us denote the set of partial plans created at iteration $t \in T$ by the algorithm together with their coefficients in the convex combination, $\pi_t = [(\lambda_{P_i}, P_i)]_{i \in I_t}$, where P_i is the partial plan and λ_{P_i} its associated coefficient. Let us the partial plans denote the partial plans with a list of booleans $[b_1, ..., b_p]$, *e.g.* $[1,]$ denotes the partial plan containing only a maintenance action at the first time-step but no information for the other time-step. Finally, the notation $\oplus$ defines an addition of a maintenance action to a partial plan, *e.g.* $[1,1,0] \oplus 1 = [1,1,0,1]$.

Algorithm outline. The algorithm constructs a set of maintenance plan as well as their coefficients in the convex combination. First, Π is initialized with the partial plan $\pi_0 = [((1), 1)]$ for the first time-step. Then at iteration $t \in T$, plans in π_{t-1} are extended with an other maintenance action or not depending on the value of z_{tt} for the second time step, and so on. We can note that a plan can be partially extended with a maintenance action. In this case, the sum of coefficients of the part with a maintenance action and the one which is extended without it should be equal to the previous coefficient of the plan. This construction has two underlying objectives. First, we want that capacities defined by (3.1e) for the extended maintenance formulation and by (2.1d) for the maintenance pattern formulation are equivalent, so that the model still give the best available production capacity whatever formulation is used. Second, we extend plans with the minimum possible cost. An example of one execution of the algorithm 1 is given in the section 3.2.1 below.

Pseudo-code

Algorithm 1 Algorithm of decomposition

Require: $z = \{z_{tt} | t \in T\}$ $\triangleright$ $z_{11}, z_{22}, ... \in [0, 1]$ are the part of maintenance actions realized.
Require: n : number of considered periods
Initialize Π with π_0
for $t \in \{1, ..., n\}$ **do**
    Sort π_{t-1} by a decreasing number of consecutive *0* in the plans
    $temp \leftarrow z_{tt}$
    $i \leftarrow 0$ $\triangleright$ We consider that the starting index for π_t is 0.
    while $temp \geq \lambda_{P_i}, P_i \in \pi_{t-1}$ **do**
        $\pi_t \leftarrow \pi_t \cup (P_i \oplus 1, \lambda_{P_i})$
        $temp \leftarrow temp - \lambda_{P_i}$
        $i \leftarrow i + 1$
    end while
    if $temp > 0$ **then**
        $\pi_t \leftarrow \pi_t \cup (P_i \oplus 1, temp)$
        $\pi_t \leftarrow \pi_t \cup (P_i \oplus 0, \lambda_{P_i} - temp)$
        $i \leftarrow i + 1$
    end if
    for maintenance plans $P_j, j > i$ of the previous period that have not been extended **do**
        $\pi_t \leftarrow \pi_t \cup (P_j \oplus 0, \lambda_{P_j})$
    end for
end for
return π_n

Let us introduce some notations. Let π_t be the set of partial plan created at iteration t by the algorithm and π_t^k be the subset of π_t which contains plans that have no maintenance since $t - k$ periods. For π a set of partial plans created by the algorithm, let $\Lambda(\pi)$ be the sum of the coefficients associated to these partial plans. Finally, let us the partial plans denote the partial plans with a list of booleans $[b_1, ..., b_p]$, *e.g.* $[1,]$ denotes the partial plan containing only a maintenance action at the first time-step but no information for the other time-step.

We now show that the solution given by Algorithm 1 has the same cost and capacity as the optimal solution of the extended maintenance formulation. To that end, we will show that the two solutions use the same maintenance action, *i.e.* $\Lambda(\pi_t^k) = z_{kt}$ for all $k, t \in T$ with $k \leq t$.

But first, let us show that, given a solution with its values of z_{kk} fixed (because they decide the maintenance cost of the solution), the variables z_{kt} for $k < t$ will be equal to $\min(z_{kk}, 1 - \sum_{i=k+1}^{t} z_{it})$. This will help us show that $\Lambda(\pi_t^k) = z_{kt}$. Let us consider a specific pair $(k, t) \in T^2$ with $k < t$. If z_{kt} is higher than z_{kk} or $1 - \sum_{i=k+1}^{t} z_{it}$ than the solution does not satisfy one of the constraints (3.1f) or (3.1g):

$$z_{kt} \leq z_{kk}, \quad \forall k, t \in T, k < t$$

$$\sum_{i=0}^{t} z_{it} \leq 1, \quad \forall k, t \in T, k < t$$

Thus z_{kt} is 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 another valid solution of the extended maintenance formulation inducing the same maintenance cost and a capacities as follows. Let $\epsilon \in \mathbb{R}$ be small enough. If the solution does not satisfy constraint (3.1g) to equality, *i.e.* $\sum_{i=0}^{t} z_{it} < 1$, then one can modify the solution by adding ϵ to the variable z_{kt}. Otherwise, we have $\sum_{i=0}^{t} z_{it} = 1$ and $\sum_{i=k}^{t} z_{it} < 1$. Thus, at least one of the variables $z_{k't}$ for $k' < k$ is higher than zero. In that case, one can modify the solution by adding ϵ to the variable z_{kt} and subtracting ϵ to the variable z_{kt}. Overall, if the solution of the extended maintenance formulation does not satisfy for every $k, t \in T$ with $k < t$:

$$z_{kt} = \min(z_{kk}, 1 - \sum_{i=k+1}^{t} z_{it}),$$

it is either invalid or one can create a better solution of the extended maintenance formulation. This contradicts the hypothesis that the considered solution was the optimal solution of the extended maintenance formulation. Thus, it does satisfy the above equation.

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

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

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

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

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

Inductive step. Let us now prove that $\Lambda(\pi_t^k) = z_{kt}$ 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: $\sum_{k'<k} \Lambda(\pi_{t-1}^{k'}) \leq z_{tt}$ or $\sum_{k'<k} \Lambda(\pi_{t-1}^{k'}) > z_{tt}$. 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 finishing with the most numbers of zeros.

Case $\sum_{k'<k} \Lambda(\pi_{t-1}^{k'}) \leq z_{tt}$: 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, $\sum_{k'=k}^{t} \Lambda(\pi_t^{k'}) = \sum_{k'=0}^{t} \Lambda(\pi_t^{k'}) = 1$. This means that $\Lambda(\pi_t^k) = 1 - \sum_{k'=k+1}^{t} \Lambda(\pi_t^{k'}) = 1 - \sum_{k'=k+1}^{t} z_t^{k'}$ because of the induction hypothesis. Moreover, $\Lambda(\pi_t^k) \leq \Lambda(\pi_{t-1}^k)$ 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 $\Lambda(\pi_t^k) \leq z_{k,t-1} = \min(z_{kk}, 1 - \sum_{i=k+1}^{t} z_{i,t-1}) \leq z_{kk}$. This finishes to show that $\Lambda(\pi_t^k) = \min(z_{kk}, 1 - \sum_{i=k+1}^{t} z_{it})$ in this case.

Case $\sum_{k'<k} \Lambda(\pi_{t-1}^{k'}) > z_{tt}$: 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, $\sum_{k'=k}^{t} \Lambda(\pi_t^{k'}) < 1$ which means $\Lambda(\pi_t^k) < 1 - \sum_{k'=k+1}^{t} \Lambda(\pi_t^{k'}) = 1 - \sum_{k'=k+1}^{t} z_t^{k'}$ 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, $\Lambda(\pi_t^k) = \Lambda(\pi_{t-1}^k) = z_{k,t-1} = \min(z_{kk}, 1 - \sum_{i=k+1}^{t-1} z_{i,t-1})$. Moreover, by the induction hypothesis we have $\sum_{k'=0}^{k-1} z_{k',t-1} = \sum_{k'=0}^{k-1} \Lambda(\pi_{t-1}^{k'}) > z_{tt} \geq 0$. This means in conjunction with constraints (3.1g) 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 $\Lambda(\pi_t^k) = z_{kk}$. This finishes proving that $\Lambda(\pi_t^k) = \min(z_{kk}, 1 - \sum_{i=k+1}^{t} z_{it})$ in this case.

The maintenance cost associated to the solution of the extended maintenance formulation is $\sum_{t \in T} m_t z_{tt}$ while the one of algorithm is $\sum_{t \in T} m_t \Lambda(\pi_t^t)$. The capacity at timestep t associated to the solution of the extended maintenance formulation is $c_{max} \sum_{k=0}^{t} \alpha^{t-k} z_{kt}$ while the one of the algorithm is $c_{max} \sum_{k=0}^{t} \alpha^{t-k} \Lambda(\pi_t^k)$. Since we have for all $k, t \in T$ with $k \leq t$ that $\Lambda(\pi_t^k) = z_{kt}$, this means that with the above algorithm, any assignment of the variables z_{kt} can be translated into a convex combination of maintenance plan inducing the same maintenance costs and the same capacity. This means that $Q_1 \subseteq Q_2$. Thus, the linear relaxation of the maintenance pattern formulation is as strong as the linear relaxation of the column-generation formulation. Thus, we have finally shown that the two linear relaxations are equivalent. □

3.2.1 Example

Consider an instance with 5 time periods, assuming that a maintenance is performed at the first period (it is the case in our model). The following table shows the different steps by which Algorithm 1 will have to go through.

We remind that maintenance should always be shared between the first period and the current period and so $\sum_{k=1}^{t} z_{kt} = 1, \forall t \in \{1, 2, 3, 4, 5\}$. Plans with an associated coefficient equal to 0 are not represented in tables as well as z_{kt} values equal to 0 to make the explanation clearer.

t	0	1	2	3	4
	$z_{11} = 1$	$z_{22} = 0.2$	$z_{33} = 1$	$z_{44} = 0.3$	$z_{55} = 0.6$
		$z_{12} = 0.8$	$z_{23} = 0$	$z_{34} = 0.7$	$z_{45} = 0.3$
			$z_{13} = 0$	$z_{24} = 0$	$z_{35} = 0.1$
				$z_{14} = 0$	$z_{25} = 0$
					$z_{15} = 0$

(a) Example: values for variables z_{kt} in the relaxed model ICLSP

Coefficient	z_p
1	*(1)*

(b) Example: Initial table of combinations of P. It is assumed that a maintenance is always performed at the first period.

Coefficient	z_p
1	*(1)*
0.2	*(1,1)*
0.8	*(1,0)*

(c) $t = 1$: plans are extended while respecting table (3.1a). 20% of maintenance actions should be performed on the part of the capacity which has been maintained the longest time ago. Here, there was only one maintenance before so 20% of its plan should be extended with a *1* and the rest with a *0*.

Coefficient	z_p
1	*(1)*
0.2	*(1,1)*
0.8	*(1,0)*
0.2	*(1,1,1)*
0.8	*(1,0,1)*

(d) $t = 2$: all plans are extended by a *1* since 100% of maintenance actions should be performed. The same coefficients are thus kept in this step.

Coefficient	z_p
1	*(1)*
0.2	*(1,1)*
0.8	*(1,0)*
0.2	*(1,1,1)*
0.8	*(1,0,1)*
0.3	*(1,0,1,1)*
0.2	*(1,1,1,0)*
0.5	*(1,0,1,0)*

(e) $t = 3$: 30% of maintenance actions is done. There is not one "oldest" part of the capacity since all plans have been updated with a *1*. By convention, plans with the most *0* before this *1* are extended with a *1* before the others. So 30 of the 80 % of plan*(1,0,1)* are extended by a *1*. The rest of this plan and plan *(1,1,1)* are extended with a *0*.

Coefficient	z_p
1	*(1)*
0.2	*(1,1)*
0.8	*(1,0)*
0.2	*(1,1,1)*
0.8	*(1,0,1)*
0.3	*(1,0,1,1)*
0.2	*(1,1,1,0)*
0.5	*(1,0,1,0)*
0.1	*(1,1,1,0,1)*
0.5	*(1,0,1,0,1)*
0.3	*(1,0,1,1,0)*
0.1	*(1,1,1,0,0)*

(f) $t = 4$: 60% of maintenance actions should be performed. All the maintenance plan *(1,0,1,0)* is first extended with a *1* and there is still $0.6 - 0.5 = 0.1$ of maintenance actions to perform. Plan *(1,1,1,0)* is therefore divided, 0.1 is extended with a *1* and 0.1 with a 0. The rest of the plans are extended with a *0*.

Table 3.1: Example of execution of the algorithm used in the proof

In the end:

$$\Pi = \{[((\mathit{1}), 1)], [((\mathit{1,1}), 0.2), ((\mathit{1,0}), 0.8)], [((\mathit{1,1,1}), 0.2), ((\mathit{1,0,1}), 0.8)],$$
$$[((\mathit{1,0,1,1}), 0.3), ((\mathit{1,1,1,0}), 0.2), ((\mathit{1,0,1,0}), 0.5)],$$
$$[((\mathit{1,1,1,0,1}), 0.1), ((\mathit{1,0,1,0,1}), 0.5), ((\mathit{1,0,1,1,0}), 0.3), ((\mathit{1,1,1,0,0}), 0.1)]\}$$

and we obtain the following values for z_{kt} for the last period:

- $z_{44} = 0.6$
- $z_{34} = 0.3$
- $z_{24} = 0.1$
- $z_{14} = z_{04} = 0$

t	Extended formulation			z_p combinations		
	Value	Capacity	Cost	Value	Capacity	Cost
0	$z_{00} = 1$	c_{max}	m_0	$1 \times (\mathit{1})$	c_{max}	m_0
1	$z_{11} = 0.2$ $z_{01} = 0.8$	$0.2 \times c_{max}$ $+$ $0.8 \times \alpha c_{max}$	$0.2m_1$	$0.2 \times (\mathit{1,1})$ $+$ $0.8 \times (\mathit{1,0})$	$0.2 \times c_{max}$ $+$ $0.8 \times \alpha c_{max}$	$0.2m_1$
2	$z_{22} = 1$	c_{max}	m_2	$0.2 \times (\mathit{1,1,1})$ $+$ $0.8 \times (\mathit{1,0,1})$	$0.2 \times c_{max}$ $+$ $0.8 \times c_{max}$	$0.2m_2$ $+$ $0.8m_2$
3	$z_{33} = 0.3$ $z_{23} = 0.7$	$0.3 \times c_{max}$ $+$ $0.7 \times \alpha c_{max}$	$0.3m_3$	$0.3 \times (\mathit{1,0,1,1})$ $+$ $0.2 \times (\mathit{1,1,1,0})$ $+$ $0.5 \times (\mathit{1,0,1,0})$	$0.3 \times c_{max}$ $+$ $0.2 \times \alpha c_{max}$ $+$ $0.5 \times \alpha c_{max}$	$0.3m_3$
4	$z_{44} = 0.6$ $z_{34} = 0.3$ $z_{24} = 0.1$	$0.6 \times c_{max}$ $+$ $0.3 \times \alpha c_{max}$ $+$ $0.1 \times \alpha^2 c_{max}$	$0.6m_4$	$0.1 \times (\mathit{1,1,1,0,1})$ $+$ $0.5 \times (\mathit{1,0,1,0,1})$ $+$ $0.3 \times (\mathit{1,0,1,1,0})$ $+$ $0.1 \times (\mathit{1,1,1,0,0})$	$0.1 \times c_{max}$ $+$ $0.5 \times c_{max}$ $+$ $0.3 \times \alpha c_{max}$ $+$ $0.1 \times \alpha^2 c_{max}$	$0.1m_4$ $+$ $0.5m_4$

Table 3.2: Comparison of the capacities and the maintenance costs obtained with the algorithm of the proof and with the extended formulation.

With Table 3.1f, we can now check if the capacity and the maintenance cost at each period are the same for the formulation with z_{kt} and the one with z_p. Table 3.2 compares them. On one side we can obtain the capacity linked to a z_{kt} with the formula $c_{kt} = \alpha^{t-k} c_{max}$ and we can add them to obtain the capacity at a given period ($c_t = \sum_{k=1}^{t} c_{kt}$). Maintenance costs are determined with the value of z_{tt} times the maintenance cost of the period t because z_{tt} is the only costly part. On the other side, we compute the capacity associated to a plan by searching the latest maintenance performed in this plan (the last 1). Depending on how the number of periods since no maintenance have been realized, the capacity is known. For instance, in the plan *10*, the last maintenance was on period 1 so a capacity of αc_{max} is available. Concerning the maintenance costs, plans ending by a *1* are the one which are costly so their associated coefficient multiplied by the maintenance cost of the considered period are added to know the final maintenance costs.

4 Valid inequalities

In addition to the search of new formulation of the problem which would allow to have a faster resolution of the problem, the search of new valid inequalities is also a mean to tighten the search space of the problem and to accelerate its solving by a solver. The first section presents how valid inequalities work and the second shows a valid inequality for our model.

4.1 Valid inequality explanation

A possibility to tighten the bounds of the LP relaxation is to generate valid inequalities. They reduce the size of the solution space by cutting off irrelevant parts, and by introducing constraints that reduce the feasible space without removing integer solutions. In other words, valid inequalities approximate feasible space to the integer convex hull. To better view this, let us consider the case of $\mathbb{R}^2$ as in figure 4.1.

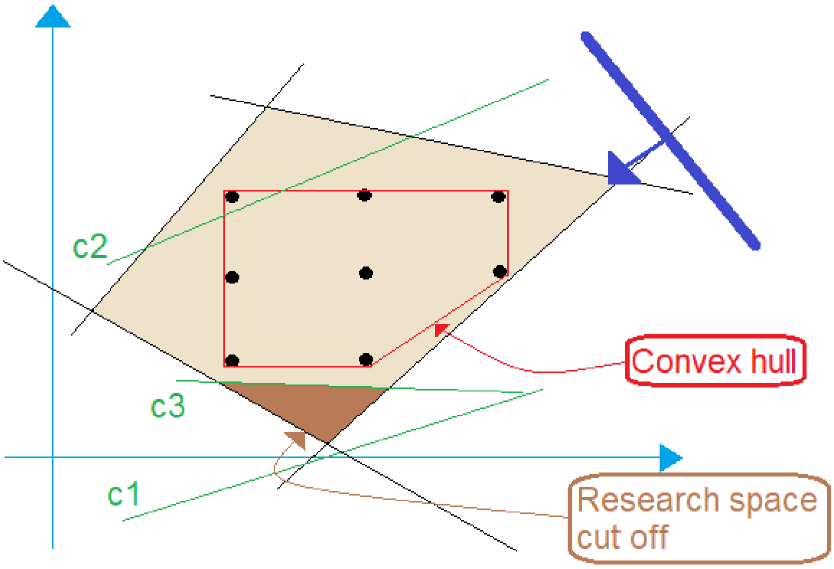

Figure 4.1: Three inequalities in the case of a minimization in $\mathbb{R}^2$

Feasible integer solutions are represented by the dark dots we are seeking in the set (colored in beige). The minimization is represented by the blue line which is following the arrow in the same color. c_1 is a valid inequality since all feasible points are still valid after adding it, but it does not help the search for the optimal solution because it is out of our research set. c_2 cuts off this set, but it is not valid since it removes a feasible solution. c_3 is a valid inequality since all feasible points are still valid after adding the cut, and it also contributes to reduce the search space (the brown part is removed).

4.2 Wagner-Whitin inequality

The first inequality which occurs often in the literature [8] [38] is the Wagner-Whitin (WW) inequality. It is assumed that $p_{i,t-1} + h_{it} \geq p_{it}, i \in P, t \in T \backslash \{0\}$. It means that the maximum production costs will never exceed the addition of the cost to produce a product and to stock it during a period which is often the case for most of the product in reality (the variation of production costs are often minor compared to the cost of holding items).

The (WW) expression for the model of Van Vyve and Shamsaei [8] is written below:

$$I_{i(t-1)} \geq \sum_{j=t}^{l} d_{ij}(1 - \sum_{k=t}^{j} y_{ik}) \quad i \in P, t \in T, l \in T, l \geq t. \tag{4.1}$$

This inequality forbids the stock to contain less than the future demands unsatisfied by another production for item i. This inequality tightens the problem without removing potential solution since demands should be satisfied at each period (backlogging is not allowed).

This inequality can be adapted to each formulation. For the plant location model (2.4), it then becomes:

$$\sum_{k=0}^{t-1} \sum_{j=t}^{n} x_{ikj} \geq \sum_{j=t}^{l} d_{ij}(1 - \sum_{k=t}^{j} y_{ik}) \quad i \in P, t \in T, l \in T, l \geq t \tag{4.2}$$

Inventory is replaced by all items already produced for period $j \in \{t, ..., n\}$ at all periods k until the period $t-1$.

Valid inequality (4.1) becomes for the shortest path model (2.5):

$$\sum_{j=0}^{t-1} (x_{ij} - d_{ij}) \geq \sum_{j=t}^{l} d_{ij}(1 - \sum_{k=t}^{j} y_{ik}) \quad i \in P, t \in T, l \in T, l \geq t \tag{4.3}$$

Here, the inventory is simply replaced by the difference between the production and the demand at each period.

5 Experimental results

The fifth chapter details the performance results obtained with the six models and their formulations described in chapters 2 and 3 using CPLEX and Gurobi solvers: the initial model using x_{it} variables, the plant location formulation using x_{ikt} variables and the shortest path formulation with ϕ_{ikt} variables, each combined with respectively the extended maintenance variable z_{kt} and the classic maintenance variable z_t. The section 5.1 introduces the instances and the definition of the parameters while the section 5.2 provides an analysis of the results.

5.1 Presentation of the test campaign

The models presented previously have been tested on instances that we generate. Some instances already exist for the CLSP such as instances of Trigeiro et al. [39] and are often used (Aghezzaf, Jamili, Ait-Kadi [5] and Aghezzaf, Khatab, Tam [7]). However, our model requires a maintenance cost for each product at each period. This cost is not defined in Trigeiro's instances. In the following subsection, the way our instances were created is presented.

5.1.1 Creation of instances

Each instance used in our results has been set with the following parameters for each period $t \in T$. It is reminded that m is the number of items considered and n the length of the horizon.

- a demand d_{it} which varies randomly between 0 and 50 units for each product $i \in P$
- The setup cost $f_{it} \in \{500, ..., 1000\}, i \in P$
- The production cost $p_{it} \in \{10, 11, 12, 13, 14\}, i \in P$
- The holding cost $h_{it} \in \{5, 6, 7, 8, 9, 10\}, i \in P$ so that the condition imposed to use WW inequality is satisfied $(p_{i,t-1} + h_{it} \geq p_{it}, i \in P, t \in T \backslash \{0\})$
- The maintenance cost $m_t \in \{1000 + 35 \times m \times n, 5000 + 35 \times m \times n\}$

The maximum capacity c_{max} is fixed inside each instance but varies between each of them. Its variation is defined as m multiplied by a random integer varying between 40 and 50. Note that this choice of parameter setting allows us to have feasible instances since the capacity is in general high enough in the first period to cover the demand. The values of α and $\beta_i, i \in P$ are defined in each sections, their values depending on the part of the models we look at. These choices have been made regarding the value used in the literature (Aghezzaf, Jamali, Ait-Kadi [5], Van Vyve and Shamsaei [8], Aghezzaf, Khatab, Tam [7]). Demands, setup cost, production cost, holding cost and the maximum capacity have been chosen to vary around the same ranges. Maintenance cost bounds have been chosen in order to have a final maintenance cost which represents around 15% of the total cost. This allows to be in concordance with Chu et al. [26] (see section 1.4).

Using the previous parameters, 200 instances were generated. They are divided into 5 groups of 40 instances, 4 groups consider $\beta_i = 0, i \in P$ and α equals to $0.6, 0.7, 0.8$ and 0.9, respectively. The fifth group of instances considers $\alpha \in \{0.6, 0.9\}$ and $\beta_i \in \{0.2, 0.4\}$. Each group is subdivided into 4 sizes of instances: $(m = 10, n = 10)$, $(m = 25, n = 10)$, $(m = 25, n = 15)$, $(m = 25, n = 20)$.

5.1.2 Conditions of tests

Because of difficulties to use CIRRELT's GPU, the computer used is my personal laptop. Its characteristics are: CPU model Intel Core I5-2410M 2.30GHz with 2 physical cores and 4 logical processors. Solvers used are CPLEX and Gurobi with a time limit of 30 minutes. The gap default value of 10^{-4} has been kept to consider a value as optimal. The coding language is Python and the modelling package is PulP in the first subsection and gurobipy after. All the other parameters of both Gurobi and CPLEX have been set to their default value.

Since my laptop is old (2010) and has not a good hardware (in terms of number of cores mainly) compared to current computers, all the results are here to give a trend of how the models behave more than to give precise values.

5.2 Analysis of the results

In this section, different results analysis are presented. First, because of our first modelling package choice, we compare two solvers. In the following subsection, one of the solver is chosen and a comparison between each of the 6 models is done. Finally, an evolution of results depending on the value of α is showed, considering one solver and one model.

5.2.1 Comparison between CPLEX and Gurobi

Because of the modelling language used in the beginning of this internship, we tested different instances with CPLEX and Gurobi. Indeed, PulP allows to choose the solver we want as an option since PulP is an interface between several solvers. So a brief comparison has been made with the results obtained in Table 1 and Table 2 (section C.1). Various values of α and $\beta_i, i \in P$ have been chosen in order to represent a large range of instance. Table C.1 summarizes these tables by computing the average of time (expressed in seconds) and gap (expressed in percentage, see section C to see how it computed) along with the standard deviation on computation time used by each solver to calculate the solution for each of the 6 models. For these computations, only 8 from the 10 instances of the subgroup have been taken into account, the easiest and hardest instances being removed each time in order to remove extreme cases that occur. In the average gap lines, OPT indicates that all instances have been solved to optimality. Note that 0.01% is written instead of OPT in some cells, it means that in average the gap equals 0.01% however there was at least one of the instances which has not been solved until optimality.

First, it is important to note the high value of the standard deviation for large instances and for both solvers which shows the high variations of the solving difficulty. From this first set of experiments, the following synthesis can be made:

	10-10	25-10	25-15	25-20
Average time difference	97%	119%	158%	114%
Standard deviation	132%	154%	163%	102%
Average gap difference	100%	100%	100%	100%

Table 5.1: Synthesis of the results obtained for each instance, all models considered

In this table, CPLEX solver is compared to Gurobi solver so if a value is lower than 100% it means that CPLEX solver was better than Gurobi solver otherwise Gurobi solver has been better. This table joins the 6 models so it is a general comparison which is made here. We can observe that apart from the smallest instances, Gurobi solver was better than CPLEX solver in computation time for an equivalent average gap. Its standard deviation is also always lower than the one obtained with CPLEX. One can note that the average gap difference is equal for both solvers although differences appear in the results. Indeed, the lack of performance of one solver on an instance is counterbalanced by its good performance on another instance compared to the other solver. Given this results, the next experiments are performed with Gurobi as solver. In order to have the best computation time, gurobipy is now used as modelling language since the benefits in time computation can be significant between PulP and gurobipy (Teymourifar [40]).

5.2.2 Comparison between each formulation

For this analysis, α equals 0.8 and all $\beta_i, i \in P$ are set to 0. As we can see on figure 5.1, gap values do not change much between the models. Indeed, even without identifying precisely the instances, we guess that points close ones from the others are from the same instance. This is confirmed by the Table C.2.

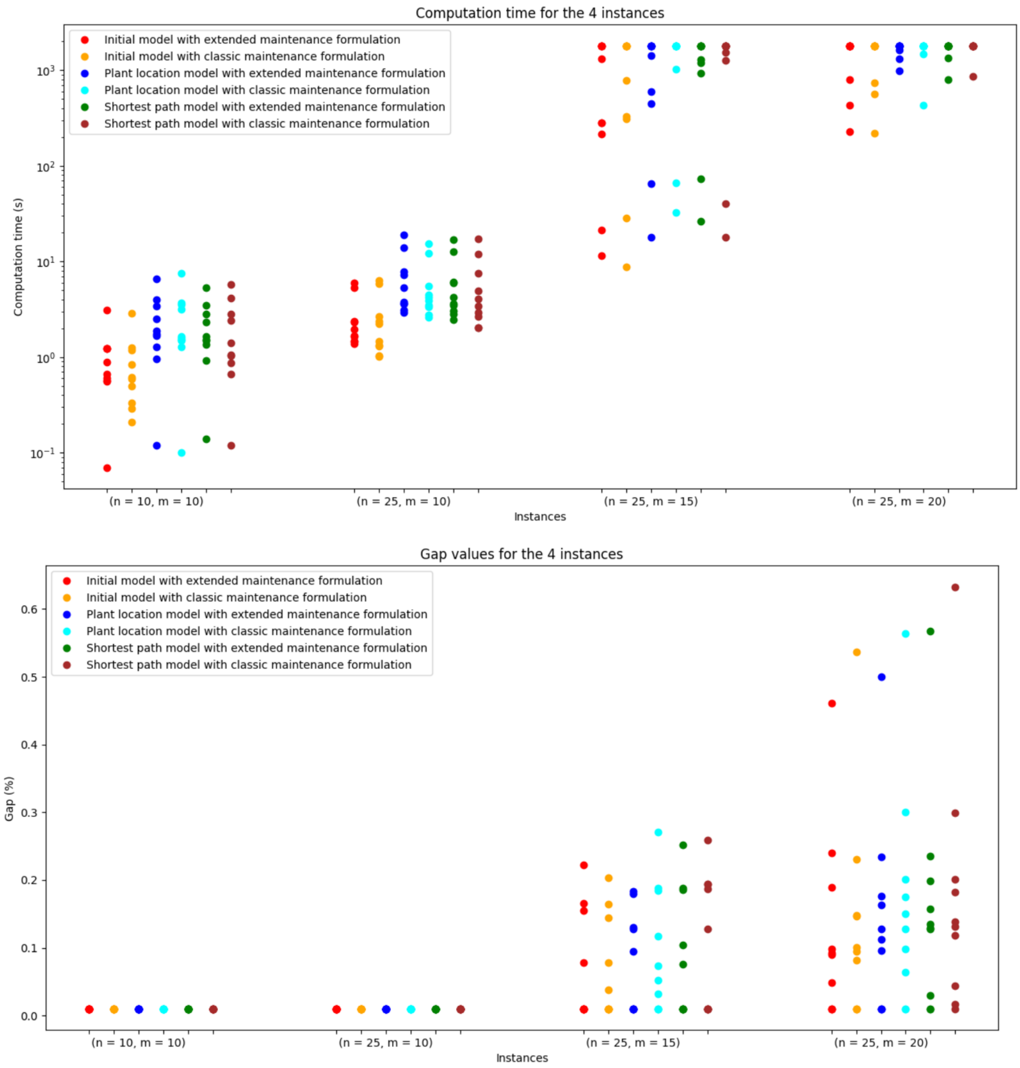

Figure 5.1: Results obtained for the different models

Nevertheless, it is possible to distinguish two models that seems to be faster than the others on figure 5.1, the two versions of the initial model. It is particularly visible with the instance ($m = 25, n = 10$). This trend is verified with the synthesis of the results obtained in annex C.2. Different conclusions can be made with table 5.2. First, we can confirm the observation made just above since the average computation time and the average gap of initial models are always better than the ones of other formulations with an equivalent maintenance formulation. Moreover, one can see that for the largest instances, the extended maintenance formulation gives nearly always the best results (7 times over 8) when for the smallest instances, it is the classic maintenance formulation (6 times over 8). This indicates that our maintenance formulation is efficient for large instances as it could have been expected. Indeed, the number of variables has increased with the extended maintenance formulation. This leads to a better relaxation but for small instances, it is not very interesting since the low bound will not be improved a lot (the research space is already small) and the addition of variables slows the solution process.

Instance			10-10	25-10	25-15	25-20
Extended maintenance formulation	Initial model	Average Time	1.01	2.55	932.71	1405.53
		Average Gap	OPT	OPT	0.07%	0.13%
	Plant location model	Average Time	2.41	7.01	1156.23	1653.90
		Average Gap	OPT	OPT	0.08%	0.14%
	Shortest path model	Average Time	2.10	6.16	1251.80	1653.99
		Average Gap	OPT	OPT	0.09%	0.16%
Classic maintenance formulation	Initial model	Average Time	0.87	2.57	1045.62	1412.23
		Average Gap	OPT	OPT	0.09%	0.16%
	Plant location model	Average Time	2.56	5.81	1372.52	1631.60
		Average Gap	OPT	OPT	0.10%	0.17%
	Shortest path model	Average Time	2.03	5.85	1366.53	1706.10
		Average Gap	OPT	OPT	0.09%	0.18%

Table 5.2: Summary of the differences in the results obtained in Table 3

5.2.3 Evolution with α

In this subsection, we want to determine the sensitivity of our model on the value of α. Four values of α are tested: 0.6, 0.7, 0.8 and 0.9. They are chosen in order to have a realistic model (a decrease more important than $\alpha = 0.6$ in the capacity would mean that producing items is really destructive for the machine). The solver used is Gurobi and we focus on the initial model with extended maintenance variables. The same instances as the one used in the previous subsection are used, only the value of α changes. The results are shown in Table C.2. The Table 5.3 shows the average computation time and average gap obtained.

Value of α		10-10	25-10	25-15	25-20
0.6	Average Time (s)	2.45	3.03	409.20	1606.71
	Average Gap (%)	OPT	OPT	0.02	0.12
0.7	Average Time (s)	1.07	24.88	143.29	1483.22
	Average Gap (%)	OPT	OPT	OPT	0.13
0.8	Average Time (s)	1.01	2.55	932.71	1405.53
	Average Gap (%)	OPT	OPT	0.07	0.13
0.9	Average Time (s)	0.75	4.09	5.58	355.80
	Average Gap (%)	OPT	OPT	OPT	0.01

Table 5.3: Synthesis of the results obtained for each instance with $\alpha \in \{0.6, 0.7, 0.8, 0.9\}$

One can see that apart from the instances run with $\alpha = 0.9$ which are solved rapidly, the computation time varies a lot from one value to another without any clear trend. A good example of this can be seen in the Table C.2. We can observe that an instance which is not solved to optimality can be solved in less than 3 seconds with a value of α increased by only 0.1 (line of instance 25_15_1) and the opposite is also possible (line of instance 25_15_9). Therefore, no conclusion can be made on the influence of α on our model apart from the fact that the value of α should be carefully chosen given the variation it can create.

Conclusions and future works

Conclusions

This report includes a detailed description of the literature used in the first place to understand the context of the project as well as previous works related to the subject. The following chapter details the analysis of three models given in the literature and adapted to our problem. Then, the next chapter focuses on the development of an extended maintenance formulation. This formulation is analyzed and a proof of its strength is given. Then valid inequalities are presented in the following chapter. Finally, the last chapter highlights an analysis of the performance of two solvers in the solving of our instances, which formulation gives the best results thanks to experimental tests as well as the influence of α on the difficulty in the solving of instances. During those six-month of internship spent working on the improvement of the production planning problem with integrated maintenance, we were able to develop a new maintenance formulation which improve this model and to justify this improvement by a mathematical proof. Unfortunately, results obtained on the different production variable reformulations are not conclusive and the analysis of the impact of the value of α does not show a clear trend on how the difficulty of the instances will evolve. Nevertheless, the results obtained on the maintenance formulation show its improvement on the solving although this improvement is low. So this reformulation is promising and further tests on larger instances should be done.

To conclude with my professional and international experience, this internship allowed me at the same time to rediscover the world of research after my second year internship and to develop new hard and soft skills. Indeed, it gave me an overview of what working in a laboratory and being part of a big research team look like. Moreover, this research project helped me to improve my autonomy, my perseverance and my critical mind. It is indeed essential to have a lot of self management all along the project development, from the beginning when determining the objectives, to the core work when exploring different approaches, new ideas and testing them passing by reading scientific articles. Finally, working on this project improved my writing skills in English by writing this report and the bibliographic review.

Difficulties encountered and possibilities of future works

The main difficulty encountered during the internship was to write the proof of the strength of our extended maintenance formulation. Trying to be as clear as possible without omitting any details was not an easy task. In addition, it is an exercise I have not faced before so it took me time to understand what were the expectations. Another difficulty was to find which parts of the model should be improved first. After the reformulation of x_{it} and of the maintenance, I tried to find valid inequalities and also how a pre-processing in the data could be done. Unfortunately, apart from the (WW) inequality, I did not found other valid inequalities to apply to our model since the capacity is a variable when it is often a constant in other studies and the pre-processing did not improve our results. I did not have the time to explore deeper this idea which can be promising even if the solution will not then be exact anymore. It would also have been interesting to test larger instances in order to see if the evolution of models stays as it is or if one of the reformulations works better on larger instances. Finally, a test on different values of $\beta_i, i \in P$ would be interesting to estimate its influence on the production capacity.

Bibliography

[1] Garg and Deshmukh, "Maintenance management: literature review and directions," *Journal of Quality in Maintenance Engineering*, vol. 12, pp. 205–238, July 2006.

[2] Ramya, Rajendran, Ziegler, Mohapatra, and Ganesh, *Capacitated Lot Sizing Problems in Process Industries*. Cham: Springer International Publishing, 2019.

[3] Fitouhi and Nourelfath, "Integrating noncyclical preventive maintenance scheduling and production planning for a single machine," *International Journal of Production Economics*, vol. 136, pp. 344–351, Apr. 2012.

[4] Lee, "Machine scheduling with an availability constraint," *Journal of Global Optimization*, vol. 9, pp. 395–416, Dec. 1996.

[5] Aghezzaf, Jamali, and Ait-Kadi, "An integrated production and preventive maintenance planning model," *European Journal of Operational Research*, vol. 181, pp. 679–685, Sept. 2007.

[6] Aghezzaf and Sitompul and Najid, "Integrated production and preventive maintenance in production systems subject to random failures," in *Actes de Mosim'08*, pp. 1–8, 2008.

[7] Aghezzaf, Khatab, and Tam, "Optimizing production and imperfect preventive maintenance plannings integration in failure-prone manufacturing systems," *Reliability Engineering & System Safety*, vol. 145, pp. 190–198, Jan. 2016.

[8] Shamsaei and V. Vyve, "Solving integrated production and condition-based maintenance planning problems by mip modeling," *Flexible Services and Manufacturing Journal*, vol. 29, pp. 184–202, May 2016.

[9] Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT). `https://www.cirrelt.ca`.

[10] Wagner and Whitin, "Dynamic version of the economic lot size model," *Management Science*, vol. 5, pp. 89–96, Oct. 1958.

[11] Karimi, F. Ghomi, and Wilson, "The capacitated lot sizing problem: a review of models and algorithms," *Omega*, vol. 31, pp. 365–378, Oct. 2003.

[12] Florian, Lenstra, and R. Kan, "Deterministic Production Planning: Algorithms and Complexity," *Management Science*, vol. 26, pp. 669–679, July 1980.

[13] Dauzere-Peres, "Synthèse sur les problèmes de lot sizing et perspectives." Présentation à la 28ème Journée Francilienne de Recherche Opérationnelle.

[14] Aloulou, Dolgui, and Kovalyov, “A bibliography of non-deterministic lot-sizing models,” *International Journal of Production Research*, vol. 52, pp. 2293–2310, Apr. 2014.

[15] Akartunalı and Dauzère-Pérès, “Dynamic lot sizing with stochastic demand timing,” *European Journal of Operational Research*, vol. 302, pp. 221–229, Oct. 2022.

[16] Önal, Romeijn, Sapra, and V. den Heuvel, “The economic lot-sizing problem with perishable items and consumption order preference,” *European Journal of Operational Research*, vol. 244, pp. 881–891, Aug. 2015.

[17] Sarker, Mukherjee, and Balan, “An order-level lot size inventory model with inventory-level dependent demand and deterioration,” *International Journal of Production Economics*, vol. 48, pp. 227–236, Feb. 1997.

[18] Jans and Degraeve, “Modeling industrial lot sizing problems: a review,” *International Journal of Production Research*, vol. 46, pp. 1619–1643, Mar. 2008.

[19] Haase, *Lotsizing and Scheduling for Production Planning*, vol. 408 of *Lecture Notes in Economics and Mathematical Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994.

[20] Gicquel and Minoux, “Multi-product valid inequalities for the discrete lot-sizing and scheduling problem,” *Computers & Operations Research*, vol. 54, pp. 12–20, Feb. 2015.

[21] Brüggemann and Jahnke, “The discrete lot-sizing and scheduling problem: Complexity and modification for batch availability,” *European Journal of Operational Research*, vol. 124, pp. 511–528, Aug. 2000.

[22] Drexl and Kimms, “Lot sizing and scheduling — Survey and extensions,” *European Journal of Operational Research*, vol. 99, pp. 221–235, June 1997.

[23] Zipkin, “Computing Optimal Lot Sizes in the Economic Lot Scheduling Problem,” *Operations Research*, vol. 39, pp. 56–63, Feb. 1991.

[24] Fleischmann and Meyr, “The general lotsizing and scheduling problem,” *Operations-Research-Spektrum*, vol. 19, pp. 11–21, Mar. 1997.

[25] Goel, Grievink, and Weijnen, “Integrated optimal reliable design, production, and maintenance planning for multipurpose process plants,” *Computers & Chemical Engineering*, vol. 27, pp. 1543–1555, Nov. 2003.

[26] Chu, Proth, and Wolff, “Predictive maintenance: The one-unit replacement model,” *International Journal of Production Economics*, vol. 54, pp. 285–295, May 1998.

[27] Groenevelt, Pintelon, and Seidmann, “Production lot sizing with machine breakdowns,” *Management Science*, vol. 38, pp. 104–123, Jan. 1992.

[28] Grigoriev, Pierard, and Storm, “Modeling and solving the periodic maintenance problem,” *METEOR Research Memorandum*, 2004.

[29] Gao, Peng, Qu, and Wu, “Jointly optimizing lot sizing and maintenance policy for a production system with two failure modes,” *Reliability Engineering & System Safety*, vol. 202, p. 106996, Oct. 2020.

[30] Schreiber, Vernickel, Richter, and Reinhart, “Integrated production and maintenance planning in cyber-physical production systems,” *Procedia CIRP*, vol. 79, pp. 534–539, 2019.

[31] Zhu, "A joint optimization model of production scheduling and maintenance based on data driven for a parallel-series production line," *Journal of Mathematics*, vol. 2021, pp. 1–11, Sept. 2021.

[32] Sloan and Shanthikumar, "Combined production and maintenance scheduling for a multiple-product, single-machine production system," *Production and Operations Management*, vol. 9, pp. 379–399, Jan. 2009.

[33] Absi, Dauzère-Pérès, and V. den Heuvel, "Integrated lot-sizing and maintenance planning," in *International Workshop on Lot-Sizing - IWLS'2019*, (Paris, France), pp. 17–20, Aug. 2019.

[34] Absi, Dauzère-Pérès, and V. den Heuvel, "Lot sizing and maintenance planning problems: complexity analysis and exact solution approaches," in *International Workshop on Lot-Sizing - IWLS'2022*, (Oslo, Norway), pp. 46–48, Aug. 2022.

[35] Krarup and Bilde, *Plant location, Set Covering and Economic Lot Size: An 0 (mn)-Algorithm for Structured Problems*, pp. 155–180. Basel: Birkhäuser Basel, 1977.

[36] Eppen and Martin, "Solving multi-item capacitated lot-sizing problems using variable redefinition," *Operations Research*, vol. 35, no. 6, pp. 832–848, 1987.

[37] Fiorotto, Jans, and D. Araujo, "An analysis of formulations for the capacitated lot sizing problem with setup crossover," *Computers & Industrial Engineering*, vol. 106, pp. 338–350, Apr. 2017.

[38] Pochet and Wolsey, *Production planning by mixed integer programming.* Springer Science & Business Media, Apr. 2006.

[39] Trigeiro, "A Simple Heuristic for Lot Sizing with Setup Times," *Decision Sciences*, vol. 20, pp. 294–303, June 1989.

[40] Teymourifar, "A comparison among optimization software to solve bi-objective sectorization problem," *Heliyon*, vol. 9, p. e18602, Aug. 2023.

A Code of the decomposition algorithm

```
1 Z = [1, 0.2, 1, 0.3]
2
3
4 def comb_plan(z):
5     Pi = [[] for _ in range(len(z))]
6     Pi[0] = [('1', 1)]
7     for i in range(1, len(z)):
8         if z[i] == 0.0:
9             for (plan, coef) in Pi[i - 1]:
10                Pi[i].append((plan + '0', coef))
11        else:
12            j = -1
13            temp = z[i]
14            last_tup = 0
15            while temp > Pi[i - 1][j][1]:
16                Pi[i].insert(0, (Pi[i - 1][j][0] + '1', Pi[i - 1][j][1]))
17                temp = round(temp - Pi[i - 1][j][1], 2)
18                j -= 1
19            if temp > 0:
20                if temp == Pi[i - 1][j][1]:
21                    Pi[i].insert(0, (Pi[i - 1][j][0] + '1', temp))
22                    j -= 1
23                else:
24                    Pi[i].insert(0, (Pi[i - 1][j][0] + '1', temp))
25                    last_tup = (Pi[i - 1][j][0] + '0', round(Pi[i - 1][j][1] - temp, 2))
26                    j -= 1
27            for (plan, coef) in Pi[i-1][:j+1]:
28                Pi[i].append((plan + '0', coef))
29            if last_tup != 0:
30                Pi[i].append(last_tup)
31    return Pi
32
33
34 print(comb_plan(Z))
```

In this way of coding the algorithm, we avoid to sort Π by keeping the right order of plans in each π_t and by discretizing some special cases. The first if loop line 8 is here in order to avoid a problem of indexes in the last for loop line 27. Then, if line 29 helps to keep the right order in Π when a plan has to be extended with a *1* for one part and a *0* for the other part. Indeed, this last part should be added to Π at the end because it will be the first part which will be extended with a *1* in the next step.

B Column-generation formulation

The following model is the model used with the column-generation. It has been taken from the work of my colleague Harcènage Dansou.

Sets and parameters
T : Set of periods in the horizon of the planning
P : Set of items
Ω : Set of indexes of all maintenance plans
f_{it} : Setup cost of item $i \in P$ at period $t \in T$
p_{it} : Production cost by unit of item $i \in P$ at period $t \in T$
h_{it} : Holding cost by unit of item $i \in P$ at the end of period $t \in T$
d_{it} : Demand of units of item $i \in P$ at period $t \in T$
m_t : Maintenance cost at period $t \in T$
M_j : Maintenance cost of the maintenance plan $j \in \Omega$ (Computed with m_t)
cap_t^j : Available capacity at period $t \in T$ depending on the maintenance plan $j \in \Omega$
C_{max} : Maximum capacity of the machine

Decision variables
x_{it} : Number of units of item $i \in P$ produced at period $t \in T$
I_{it} : Number of units of item $i \in P$ stocked at the end of the period $t \in T$
c_t : Available capacity at period $t \in T$

$$y_{i,t} = \begin{cases} 1 & \text{if item } i \in P \text{ is produced at period } t \in T \\ 0 & \text{otherwise} \end{cases}$$

$$w_j = \begin{cases} 1 & \text{if the maintenance plan } j \in \Omega \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

Model

$$(DW) \quad \min_{x,I,y,z,c} \quad \sum_{i\in P}\sum_{t\in T}(f_{it} * y_{it} + p_{it} * x_{it} + h_{it} * I_{it}) + \sum_{j\in\Omega} M_j * w_j \tag{B.1a}$$

s.t.

$$x_{it} + I_{it-1} = d_{it} + I_{it} \qquad \forall\, i \in P,\ t \in T,, \tag{B.1b}$$

$$x_{it} \leq \left(\sum_{s\geq t} d_{is}\right) * y_{it} \qquad \forall\, i \in P,\ t \in T, \tag{B.1c}$$

$$\sum_{i\in P} x_{it} \leq c_t \qquad \forall\, t \in T, \tag{B.1d}$$

$$\sum_{j\in\Omega} w_j = 1, \tag{B.1e}$$

$$c_t = \sum_{j\in\Omega} cap_t^j * w_j \qquad \forall\, t \in T, \tag{B.1f}$$

$$x_{it}, I_{it} \geq 0 \qquad \forall\, i \in P,\ t \in T, \tag{B.1g}$$

$$c_t \geq 0 \qquad \forall\, t \in T, \tag{B.1h}$$

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

$$w_j \in \{0,1\} \qquad \forall j \in \Omega \tag{B.1j}$$

C Results tables

Time represents the computation time and is expressed in **seconds** in this section. Gap is the objective value obtained at the end of the computation time divided by the best low bound found at the end of the computation time. It is considered that when this gap equals the default value of 10^{-4} or less, the value is optimal. In these tables, gap is represented with percentage so 10^{-4} is 0.01%. For each instance a time limit of 30 minutes was set. Objective values set at OPT means that the optimal solution was found at the end of the computation time.

C.1 CPLEX and Gurobi results

The results obtained with the two different solvers are presented here.

Instances	Extended maintenance formulation						Classic formulation					
	(ICLSP)		**(PLCLSP)**		**(SPCLSP)**		**(ICLSP)**		**(PLCLSP)**		**(SPCLSP)**	
	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap
10_10_1	0.86	OPT	1.71	OPT	1.02	OPT	0.73	OPT	1.40	OPT	1.14	OPT
10_10_2	1.06	OPT	2.83	OPT	2.02	OPT	1.09	OPT	1.91	OPT	2.66	OPT
10_10_3	1.02	OPT	2.41	OPT	1.78	OPT	1.36	OPT	1.68	OPT	1.64	OPT
10_10_4	1.86	OPT	3.51	OPT	3.37	OPT	1.63	OPT	2.56	OPT	3.38	OPT
10_10_5	0.84	OPT	1.53	OPT	1.31	OPT	0.99	OPT	2.15	OPT	1.95	OPT
10_10_6	0.27	OPT	0.77	OPT	0.55	OPT	0.38	OPT	0.65	OPT	0.54	OPT
10_10_7	0.48	OPT	1.17	OPT	1.20	OPT	0.40	OPT	1.10	OPT	1.33	OPT
10_10_8	0.97	OPT	1.61	OPT	1.72	OPT	0.57	OPT	1.98	OPT	1.42	OPT
10_10_9	0.30	OPT	0.98	OPT	0.34	OPT	0.21	OPT	0.66	OPT	0.52	OPT
10_10_10	1.13	OPT	4.10	OPT	3.63	OPT	1.71	OPT	2.11	OPT	1.86	OPT
25_10_1	2.44	OPT	3.63	OPT	4.75	OPT	1.74	OPT	3.34	OPT	4.36	OPT
25_10_2	2.90	OPT	6.93	OPT	4.61	OPT	4.27	OPT	6.08	OPT	7.75	OPT
25_10_3	3.92	OPT	5.86	OPT	8.70	OPT	3.74	OPT	6.17	OPT	6.33	OPT
25_10_4	4.91	OPT	7.78	OPT	7.08	OPT	4.18	OPT	8.40	OPT	6.89	OPT
25_10_5	13.46	OPT	13.22	OPT	28.51	OPT	9.66	OPT	18.04	OPT	20.43	OPT
25_10_6	5.82	OPT	8.75	OPT	16.06	OPT	6.76	OPT	8.67	OPT	9.06	OPT
25_10_7	3.67	OPT	4.58	OPT	6.42	OPT	2.24	OPT	4.48	OPT	8.78	OPT
25_10_8	41.05	OPT	120.49	OPT	307.09	OPT	61.28	OPT	108.29	OPT	108.55	OPT
25_10_9	3.86	OPT	6.12	OPT	5.91	OPT	3.58	OPT	6.38	OPT	6.40	OPT
25_10_10	1.19	OPT	2.00	OPT	4.36	OPT	1.85	OPT	2.76	OPT	2.77	OPT
25_15_1	28.98	OPT	68.35	OPT	82.94	OPT	24.97	OPT	42.12	OPT	58.28	OPT
25_15_2	2.03	OPT	3.95	OPT	6.25	OPT	2.00	OPT	2.77	OPT	2.36	OPT
25_15_3	1.78	OPT	2.87	OPT	3.64	OPT	1.67	OPT	2.82	OPT	3.03	OPT
25_15_4	51.86	OPT	410.32	OPT	315.77	OPT	80.52	OPT	292.70	OPT	293.14	OPT
25_15_5	12.49	OPT	33.99	OPT	49.68	OPT	11.51	OPT	26.21	OPT	39.17	OPT
25_15_6	6.30	OPT	26.31	OPT	37.96	OPT	10.85	OPT	22.68	OPT	18.43	OPT
25_15_7	5.78	OPT	17.73	OPT	15.95	OPT	4.85	OPT	11.61	OPT	17.05	OPT
25_15_8	19.17	OPT	44.26	OPT	59.35	OPT	13.35	OPT	30.27	OPT	36.94	OPT
25_15_9	8.54	OPT	36.98	OPT	47.46	OPT	12.59	OPT	35.19	OPT	43.94	OPT
25_15_10	51.03	OPT	111.65	OPT	119.95	OPT	40.22	OPT	81.29	OPT	142.81	OPT
25_20_1	87.51	OPT	146.97	OPT	294.67	OPT	95.41	OPT	107.26	OPT	121.58	OPT
25_20_2	1800	0.04%	1800	0.09%	1800	0.02%	1800	0.08%	1800	0.10%	1800	0.02%
25_20_3	93.01	OPT	250.30	OPT	436.37	OPT	131.77	OPT	217.85	OPT	392.99	OPT
25_20_4	351.04	OPT	615.31	OPT	672.60	OPT	354.68	OPT	790.83	OPT	818.00	OPT
25_20_5	63.85	OPT	141.15	OPT	324.90	OPT	76.92	OPT	200.19	OPT	246.43	OPT
25_20_6	1290.47	OPT	1800	0.08%	1800	0.02%	1362.04	OPT	1800	0.04%	1800	0.03%
25_20_7	6.16	OPT	8.87	OPT	16.54	OPT	6.40	OPT	6.33	OPT	15.12	OPT
25_20_8	66.60	OPT	237.00	OPT	247.57	OPT	68.58	OPT	120.96	OPT	181.75	OPT
25_20_9	1800	0.19%	1800	0.23%	1800	0.08%	1800	0.20%	1800	0.21%	1800	0.07%
25_20_10	63.05	OPT	173.33	OPT	190.74	OPT	54.63	OPT	87.97	OPT	148.81	OPT

Table 1: Run of instances with CPLEX

Instances	Extended maintenance formulation						Classic formulation					
	(ICLSP)		**(PLCLSP)**		**(SPCLSP)**		**(ICLSP)**		**(PLCLSP)**		**(SPCLSP)**	
	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap
10_10_1	0.80	OPT	0.92	OPT	1.66	OPT	0.29	OPT	0.87	OPT	1.41	OPT
10_10_2	1.01	OPT	2.51	OPT	2.60	OPT	1.41	OPT	2.91	OPT	3.12	OPT
10_10_3	0.68	OPT	1.89	OPT	1.55	OPT	0.64	OPT	1.30	OPT	1.76	OPT
10_10_4	1.11	OPT	2.42	OPT	2.84	OPT	1.15	OPT	2.95	OPT	2.90	OPT
10_10_5	1.05	OPT	1.74	OPT	2.82	OPT	0.65	OPT	1.69	OPT	1.12	OPT
10_10_6	0.94	OPT	1.19	OPT	1.19	OPT	0.30	OPT	0.90	OPT	1.13	OPT
10_10_7	0.53	OPT	2.60	OPT	2.28	OPT	0.58	OPT	2.74	OPT	2.53	OPT
10_10_8	0.54	OPT	1.55	OPT	1.90	OPT	1.01	OPT	2.05	OPT	2.42	OPT
10_10_9	0.54	OPT	1.39	OPT	1.63	OPT	0.49	OPT	1.10	OPT	0.71	OPT
10_10_10	0.77	OPT	2.69	OPT	2.28	OPT	0.77	OPT	1.70	OPT	1.65	OPT
25_10_1	1.78	OPT	2.33	OPT	4.35	OPT	1.26	OPT	4.09	OPT	5.99	OPT
25_10_2	2.83	OPT	5.15	OPT	4.41	OPT	1.61	OPT	5.47	OPT	5.70	OPT
25_10_3	2.86	OPT	12.50	OPT	9.16	OPT	2.39	OPT	8.68	OPT	9.23	OPT
25_10_4	2.85	OPT	6.46	OPT	7.73	OPT	2.82	OPT	6.39	OPT	6.28	OPT
25_10_5	9.17	OPT	14.12	OPT	8.72	OPT	8.19	OPT	13.09	OPT	12.39	OPT
25_10_6	6.57	OPT	14.56	OPT	12.24	OPT	5.41	OPT	8.16	OPT	12.39	OPT
25_10_7	2.63	OPT	6.37	OPT	5.73	OPT	1.73	OPT	6.74	OPT	4.15	OPT
25_10_8	78.03	OPT	168.99	OPT	133.58	OPT	86.00	OPT	154.38	OPT	175.76	OPT
25_10_9	2.17	OPT	4.04	OPT	5.61	OPT	2.50	OPT	5.36	OPT	9.37	OPT
25_10_10	1.32	OPT	2.42	OPT	3.77	OPT	1.36	OPT	3.28	OPT	2.89	OPT
25_15_1	15.17	OPT	59.93	OPT	47.71	OPT	14.36	OPT	51.65	OPT	43.92	OPT
25_15_2	2.25	OPT	5.51	OPT	5.12	OPT	1.40	OPT	4.55	OPT	4.19	OPT
25_15_3	3.24	OPT	6.13	OPT	6.27	OPT	1.54	OPT	4.90	OPT	4.15	OPT
25_15_4	40.46	OPT	247.92	OPT	102.37	OPT	64.60	OPT	221.39	OPT	146.35	OPT
25_15_5	6.78	OPT	22.08	OPT	25.05	OPT	8.14	OPT	23.43	OPT	17.99	OPT
25_15_6	4.13	OPT	16.98	OPT	13.09	OPT	4.84	OPT	9.86	OPT	15.19	OPT
25_15_7	4.79	OPT	13.80	OPT	11.75	OPT	7.18	OPT	10.13	OPT	10.32	OPT
25_15_8	7.75	OPT	28.24	OPT	29.68	OPT	6.66	OPT	17.99	OPT	18.67	OPT
25_15_9	16.06	OPT	32.35	OPT	15.75	OPT	10.35	OPT	22.53	OPT	26.63	OPT
25_15_10	23.25	OPT	65.94	OPT	62.04	OPT	33.76	OPT	45.80	OPT	65.81	OPT
25_20_1	21.03	OPT	76.20	OPT	109.07	OPT	31.42	OPT	97.60	OPT	87.71	OPT
25_20_2	956.19	OPT	1800	0.05%	1800	0.06%	1064.27	OPT	1800	0.07%	1800	0.08%
25_20_3	60.16	OPT	228.52	OPT	202.43	OPT	82.45	OPT	134.92	OPT	127.07	OPT
25_20_4	146.24	OPT	470.93	OPT	250.22	OPT	200.70	OPT	573.17	OPT	549.74	OPT
25_20_5	51.01	OPT	142.34	OPT	170.32	OPT	40.95	OPT	113.13	OPT	190.90	OPT
25_20_6	1573.03	OPT	1800	0.07%	1800	0.06%	1718.79	OPT	1800	0.05%	1800	0.08%
25_20_7	6.43	OPT	20.64	OPT	14.15	OPT	6.42	OPT	15.06	OPT	14.94	OPT
25_20_8	66.18	OPT	358.08	OPT	151.97	OPT	48.11	OPT	926.66	OPT	110.95	OPT
25_20_9	1800	0.19%	1800	0.23%	1800	0.23%	1800	0.20%	1800	0.21%	1800	0.23%
25_20_10	40.94	OPT	153.15	OPT	171.15	OPT	35.43	OPT	164.72	OPT	177.59	OPT

Table 2: Run of instances with GUROBI

Instance				10-10	25-10	25-15	25-20
Extended maintenance formulation	Initial model	Average Time	CPLEX	0.92	5.12	16.76	476.94
			Gurobi	0.78	3.86	10.15	364.35
		Standard deviation	CPLEX	0.46	3.53	21.68	680.08
			Gurobi	0.25	2.60	13.84	580.74
		Average Gap	CPLEX	OPT	OPT	OPT	0.01%
			Gurobi	OPT	OPT	OPT	OPT
	Plant location model	Average Time	CPLEX	1.97	7.11	42.77	645.51
			Gurobi	1.88	8.19	30.68	628.65
		Standard deviation	CPLEX	0.87	2.97	33.81	728.56
			Gurobi	0.60	4.80	21.57	733.88
		Average Gap	CPLEX	OPT	OPT	OPT	0.03%
			Gurobi	OPT	OPT	OPT	0.02%
	Shortest path model	Average Time	CPLEX	1.60	10.26	52.12	720.86
			Gurobi	2.16	7.24	26.42	581.90
		Standard deviation	CPLEX	0.89	8.24	36.82	681.99
			Gurobi	0.54	2.74	19.45	752.90
		Average Gap	CPLEX	OPT	OPT	OPT	0.01%
			Gurobi	OPT	OPT	OPT	0.02%
Classic maintenance formulation	Initial model	Average Time	CPLEX	0.87	4.52	15.00	493.00
			Gurobi	0.78	3.24	10.85	402.77
		Standard deviation	CPLEX	0.48	2.56	12.27	688.33
			Gurobi	0.38	2.38	9.99	637.23
		Average Gap	CPLEX	OPT	OPT	OPT	0.02%
			Gurobi	OPT	OPT	OPT	OPT
	Plant location model	Average Time	CPLEX	1.68	7.70	31.52	640.63
			Gurobi	1.95	7.25	23.29	701.53
		Standard deviation	CPLEX	0.61	4.54	23.70	750.58
			Gurobi	0.84	2.80	17.04	736.93
		Average Gap	CPLEX	OPT	OPT	OPT	0.03%
			Gurobi	OPT	OPT	OPT	0.02%
	Shortest path model	Average Time	CPLEX	1.76	8.75	44.96	688.70
			Gurobi	2.00	8.19	25.34	605.50
		Standard deviation	CPLEX	0.90	4.95	43.25	721.09
			Gurobi	0.88	3.13	20.22	751.51
		Average Gap	CPLEX	OPT	OPT	OPT	0.01%
			Gurobi	OPT	OPT	OPT	0.03%

Table C.1: Summary of the differences in the results obtained in Tables 1 and 2

C.2 Results on the different models

The results obtained with the six different models (Initial model **(ICLSP)**, Plant location model **(PLCLSP)** and Shortest path model **(SPCLSP)**, each of them combined with the extended maintenance formulation first and the classic maintenance formulation after) are presented here.

Instances	Extended maintenance formulation						Classic formulation					
	(ICLSP)		**(PLCLSP)**		**(SPCLSP)**		**(ICLSP)**		**(PLCLSP)**		**(SPCLSP)**	
	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap
10_10_1	0.66	OPT	0.96	OPT	0.92	OPT	0.33	OPT	1.58	OPT	0.87	OPT
10_10_2	3.09	OPT	6.53	OPT	5.37	OPT	2.88	OPT	7.59	OPT	5.74	OPT
10_10_3	0.56	OPT	1.89	OPT	1.63	OPT	0.62	OPT	1.52	OPT	1.06	OPT
10_10_4	0.60	OPT	1.73	OPT	1.51	OPT	0.50	OPT	1.63	OPT	1.04	OPT
10_10_5	0.07	OPT	0.12	OPT	0.14	OPT	0.29	OPT	0.10	OPT	0.12	OPT
10_10_6	1.22	OPT	3.99	OPT	3.48	OPT	1.25	OPT	3.68	OPT	4.17	OPT
10_10_7	1.23	OPT	3.39	OPT	2.32	OPT	1.19	OPT	3.53	OPT	2.42	OPT
10_10_8	0.88	OPT	1.69	OPT	1.48	OPT	0.59	OPT	1.49	OPT	1.41	OPT
10_10_9	0.57	OPT	1.27	OPT	1.36	OPT	0.21	OPT	1.27	OPT	0.66	OPT
10_10_10	1.23	OPT	2.52	OPT	2.83	OPT	0.83	OPT	3.17	OPT	2.84	OPT
25_10_1	5.93	OPT	18.88	OPT	12.78	OPT	5.90	OPT	15.45	OPT	11.86	OPT
25_10_2	1.63	OPT	3.74	OPT	3.50	OPT	1.33	OPT	3.92	OPT	3.41	OPT
25_10_3	1.45	OPT	3.61	OPT	3.63	OPT	1.04	OPT	2.77	OPT	2.03	OPT
25_10_4	5.33	OPT	13.94	OPT	17.03	OPT	6.37	OPT	12.28	OPT	17.10	OPT
25_10_5	1.68	OPT	3.09	OPT	2.81	OPT	1.01	OPT	2.59	OPT	2.04	OPT
25_10_6	1.94	OPT	7.21	OPT	6.11	OPT	2.67	OPT	3.50	OPT	7.50	OPT
25_10_7	2.39	OPT	5.32	OPT	4.26	OPT	2.25	OPT	4.45	OPT	4.03	OPT
25_10_8	1.47	OPT	3.61	OPT	2.44	OPT	1.45	OPT	3.34	OPT	2.95	OPT
25_10_9	2.34	OPT	7.78	OPT	5.97	OPT	2.36	OPT	5.51	OPT	4.93	OPT
25_10_10	1.38	OPT	2.95	OPT	3.05	OPT	1.31	OPT	4.26	OPT	2.67	OPT
25_15_1	1800	0.223%	1800	0.183%	1800	0.252%	1800	0.204%	1800	0.271%	1800	0.259%
25_15_2	281.81	OPT	450.93	OPT	930.62	OPT	309.78	OPT	1026.07	OPT	1271.43	OPT
25_15_3	1800	0.155%	1800	0.179%	1800	0.188%	1800	0.145%	1800	0.185%	1800	0.187%
25_15_4	1315.13	OPT	1800	0.095%	1800	0.076%	1800	0.038%	1800	0.073%	1800	0.128%
25_15_5	21.20	OPT	65.54	OPT	73.44	OPT	28.64	OPT	66.85	OPT	40.52	OPT
25_15_6	11.53	OPT	17.98	OPT	26.35	OPT	8.86	OPT	32.3	OPT	17.97	OPT
25_15_7	1800	0.079%	1800	0.130%	1800	0.104%	1800	0.078%	1800	0.118%	1800	0.119%
25_15_8	280.41	OPT	599.21	OPT	1300.57	OPT	779.06	OPT	1800	0.033	1800	0.010
25_15_9	1800	0.166%	1800	0.128%	1800	0.186%	1800	0.164%	1800	0.188%	1800	0.195%
25_15_10	217.05	OPT	1428.62	OPT	1187.00	OPT	329.88	OPT	1800	0.053%	1535.42	OPT
25_20_1	1800	0.090%	1800	0.113%	1800	0.127%	1800	0.095%	1800	0.127%	1800	0.131%
25_20_2	1800	0.049%	1800	0.096%	1800	0.130%	1800	0.082%	1800	0.098%	1800	0.118%
25_20_3	1800	0.190%	1800	0.177%	1800	0.199%	1800	0.148%	1800	0.202%	1800	0.201%
25_20_4	1800	0.461%	1800	0.500%	1800	0.567%	1800	0.536%	1800	0.563%	1800	0.633%
25_20_5	1800	0.240%	1800	0.234%	1800	0.235%	1800	0.231%	1800	0.301%	1800	0.300%
25_20_6	1800	0.099%	1800	0.128%	1800	0.135%	1800	0.101%	1800	0.151%	1800	0.138%
25_20_7	226.91	OPT	1311.04	OPT	793.04	OPT	218.87	OPT	426.37	OPT	861.03	OPT
25_20_8	799.46	OPT	1641.01	OPT	1346.84	OPT	738.3	OPT	1489.63	OPT	1800	0.0163%
25_20_9	428.94	OPT	986.94	OPT	1800	0.0296%	565.16	OPT	1800	0.0641%	1800	0.0441%
25_20_10	1800	0.0927%	1800	0.1637%	1800	0.1572%	1800	0.1465%	1800	0.1757%	1800	0.1826%

Table 3: Run of instances with $\alpha = 0.8$

C.3 Results on α

The results obtained with the four different values of α (0.6, 0.7, 0.8 and 0.9) are presented here.

Instances	$\alpha = 0.6$		$\alpha = 0.7$		$\alpha = 0.8$		$\alpha = 0.9$	
	Time	Gap (%)	Time	Gap (%)	Time	Gap (%)	Time	Gap (%)
10_10_1	1.80	OPT	0.75	OPT	0.66	OPT	0.49	OPT
10_10_2	1.08	OPT	0.70	OPT	3.09	OPT	0.39	OPT
10_10_3	4.93	OPT	2.61	OPT	0.56	OPT	0.89	OPT
10_10_4	0.55	OPT	1.12	OPT	0.60	OPT	0.38	OPT
10_10_5	4.49	OPT	1.69	OPT	0.07	OPT	0.83	OPT
10_10_6	1.34	OPT	0.65	OPT	1.22	OPT	1.69	OPT
10_10_7	1.10	OPT	0.92	OPT	1.23	OPT	1.47	OPT
10_10_8	2.09	OPT	0.93	OPT	0.88	OPT	0.74	OPT
10_10_9	6.25	OPT	0.69	OPT	0.57	OPT	0.23	OPT
10_10_10	0.91	OPT	0.63	OPT	1.23	OPT	0.40	OPT
25_10_1	9.47	OPT	4.76	OPT	5.93	OPT	27.55	OPT
25_10_2	2.06	OPT	9.64	OPT	1.63	OPT	0.48	OPT
25_10_3	2.12	OPT	5.61	OPT	1.45	OPT	1.85	OPT
25_10_4	2.11	OPT	1.88	OPT	5.33	OPT	2.82	OPT
25_10_5	1.19	OPT	11.34	OPT	1.68	OPT	0.23	OPT
25_10_6	1.75	OPT	5.13	OPT	1.94	OPT	1.09	OPT
25_10_7	2.11	OPT	1.36	OPT	2.39	OPT	4.21	OPT
25_10_8	1.67	OPT	191.78	OPT	1.47	OPT	0.71	OPT
25_10_9	4.13	OPT	1.22	OPT	2.34	OPT	1.66	OPT
25_10_10	3.66	OPT	16.06	OPT	1.38	OPT	0.26	OPT
25_15_1	148.11	OPT	71.82	OPT	1800	0.223	2.47	OPT
25_15_2	18.86	OPT	154.40	OPT	281.81	OPT	2.51	OPT
25_15_3	335.91	OPT	231.17	OPT	1800	0.1551	3.27	OPT
25_15_4	18.25	OPT	474.04	OPT	1315.13	OPT	2.80	OPT
25_15_5	1520.02	OPT	55.32	OPT	21.2	OPT	7.48	OPT
25_15_6	36.04	OPT	15.57	OPT	11.53	OPT	3.50	OPT
25_15_7	145.13	OPT	15.53	OPT	1800	0.079	3.35	OPT
25_15_8	33.16	OPT	49.14	OPT	280.41	OPT	2.22	OPT
25_15_9	1800	0.062	224.70	OPT	1800	0.166	25.55	OPT
25_15_10	36.52	OPT	141.18	OPT	217.05	OPT	2.68	OPT
25_20_1	1800	0.310	1800	0.182	1800	0.090	142.31	OPT
25_20_2	1800	0.044	1800	0.065	1800	0.049	21.93	OPT
25_20_3	1800	0.397	1800	0.233	1800	0.190	413.31	OPT
25_20_4	1800	0.093	180.49	OPT	1800	0.4611	131.28	OPT
25_20_5	1800	0.113	1578.27	OPT	1800	0.240	805.68	OPT
25_20_6	1017.69	OPT	1800	0.216	1800	0.099	91.48	OPT
25_20_7	1800	0.081	1800	0.032	226.91	OPT	10.89	OPT
25_20_8	1800	0.072	473.39	OPT	799.46	OPT	16.00	OPT
25_20_9	649.37	OPT	1800	0.198	428.94	OPT	125.08	OPT
25_20_10	1800	0.0547	1800	0.338	1800	0.093	1800	0.052

Table C.2: Run of instances with $\alpha \in \{0.6, 0.7, 0.8, 0.9\}$