Université catholique de Louvain

Louvain School of Management

Formulations and Algorithms for Integrated Maintenance and Production Planning Problems

Fahimeh SHAMSAEI

Thèse présentée en vue d'obtention du grade de Docteur en sciences économiques et de gestion

Membres du Jury:

Mathieu VAN VYVE	(UCL, Promoter)
Per J. AGRELL	(UCL, President)
Philippe CHEVALIER	(UCL)
Laurence WOLSEY	(UCL)
El-Houssaine AGHEZZAF	(Ghent University)

December 2015

To the brilliant star of my life, my husband, Mahmood
To my assets, my sons, Matin and Maani

Acknowledgements

Looking back on the first days of the thesis, I can remember the great enthusiasm of starting a new experience and the anxiety of not knowing exactly where I was heading. Both feeling still inhabit me, five years on. This thesis appears in its current form due to the assistance and guidance of several people. I would therefore like to offer my sincere thanks to all of them.

I would like to cordially thank Mathieu Van Vyve, my esteemed promoter, because of his guidance, understanding, patience, and most importantly, his critical comments during my graduate studies at UCL. He has been supportive since the days I arrived in Louvain-la-Neuve. Ever since, Mathieu has supported me not only by providing a research assistantship over almost five years, but also academically and emotionally through the rough road to finish this thesis. He helped me to come up with the thesis topic, and during the most difficult times when writing this thesis, he gave me the moral support and the freedom I needed to move on.

I want also to express my deep thanks to the people of the Center for Operations Research and Econometrics (CORE) and Louvain School of Management (LSM) at UCL, especially those members of my doctoral committee Prof. Philippe Chevalier and Prof. Per J. Agrell for the insightful discussions, offering valuable advice, and for their support during the whole period of the study. I would also like to thank the members of my doctoral Jury Prof. Laurence Wolsey Prof. El-Houssaine Aghezzaf for their excellent advises and suggestions.

Dozens of people have helped me immensely during my stay at UCL. First, I would like to specially thank Mrs. Catherine Germain the secretary of CORE who welcomed me to CORE in 2010. She has been a source of love and energy ever since. Her unconditional support has been essential all these years. A special acknowledgement goes to my friends in Louvain-la-Neuve who were always available and beside me whenever I needed, Mrs. Saloomeh Shariati, Mrs Katayoon Saeedi, Mrs. Hoda Chiniforooshan, Ms. Saghi Saghazadeh, Mrs. Hanieh Hajighasemi, Mrs. Pegah Khoshnevis, Mr. Adel Hatami, and other good friends of mine. They were true friends ever since we arrived in LLN in 2010. Their friendship is irreplaceable.

I also thank my parents for their faith in me and allowing me to be as ambitious as I wanted. It was under their watchful eye that I gained so much drive and an ability to tackle challenges head on. Also, I thank my sister and parents in law who provided me with unending encouragement and support.

Finally and most importantly, I would like to thank my husband Mahmood Rezaei who was certainly the most enduring support in this long journey. His support, encouragement, quiet patience and unwavering love were undeniably the bedrock upon which the past years of my life have been built.

Fahimeh Shamsaei
Louvain-la-Neuve
December 2015

Abstract

An important challenge in Lot Sizing faces to decision makers when there are capacity constraints. Decisions related to these problems have a direct impact at the operational level of the company, and consequently, they have an effect over the global performance of the company. One of the capacity constraints in Lot Sizing Problems arises when the machines produce with variable productivity. It means that their production capacity changes over time.
Moreover, when the production capacity decreases over time, it is vital to stop the production line and fix the machines to bring them back to full productivity. Thus, another problem is about the maintenance scheduling. Given that the machines do not work during the maintenance period, it is important to synchronize maintenance and Lot Sizing decisions. Taking into consideration the company's freight, transportation decisions should be incorporated into the model as well in order to exploit the freight efficiently. The latter aspect adds to the complexity of the problem significantly.
Throughout this thesis, we aim to propose an improved formulation to deal with Dynamic Productivity Lot Sizing Problem (DPLSP) by extending traditional approaches provided for the Capacitated Lot Sizing Problems (CLSPs). Many of these problems are hard and hence are unlikely to admit polynomial time solutions. Therefore, we approach DPLSP based on the already existing common sense or specialized heuristics for CLSPs. Because of the non-increasing capacity assumption we apply in our model, it is vital to integrated the aforementioned problem with the maintenance scheduling in order to bring the machines back to their full productivity. Specifically, in this research we plan to cope with a single (multi)-product(s), single (multi)-machine(s) CLSP, as a basic production/inventory problem with decreasing capacity. Then, we plan to synchronize maintenance and transport aspects to best satisfy the demand, while minimizing the total cost of the process.
We propose two types of algorithms to solve such an integrated problem. The

first algorithm (off-line algorithm) is suggested when the demand is deterministic. We exploit MIP-based Fix-and-Optimize heuristic which results is compared to the results obtained using the MIP solver. The second algorithm which called online algorithm is propounded when the demand is not known in advance. The effectiveness of the proposed online algorithms is measured by their competitive ratio, defined as the ratio between the maximum number of the demands can be satisfied and that of a hypothetical offline algorithm (Dynamic Programming) which knows the entire sequence of requests in advance and chooses its actions optimally.

Abbreviations

B&B	**B**ranch and **B**ound
B&C	**B**ranch and **C**ut
CLSP	**C**apacitated **L**ot **S**izing **P**roblem
CG	**C**olumn **G**eneration
CBM	**C**ondition- **B**ased **M**aintenance
CM	**C**orrective **M**aintenance
C&B	**C**ut and **B**ranch
DPLSP	**D**ynamic **P**roductivity **L**ot **S**izing **P**roblem
EMQ	**E**conomic **M**anufacturing **Q**uantity
F&O	**F**ix-and-**O**ptimize
F&R	**F**ix-and-**R**elax
GA	**G**enetic **A**lgorithm
IP	**I**nteger **P**rogramming
LD	**L**agrangian **D**ecomposition
LH	**L**agrangian **H**euristics
LR	**L**agrangian **R**elaxation
LP	**L**inear **P**rogramming

MRP	**M**aterial **R**equirements **P**lanning
MP	**M**athematical-based **P**rogramming
MIP	**M**ixed **I**nteger **P**rogramming
MILP	**M**ixed **I**nteger **L**inear **P**rogramming
MC	**M**ulti-**C**ommodity
PdM	**P**redictive **M**aintenance
PM	**P**reventive **M**aintenance
R&F	**R**elaxed-and-**F**ix
RCM	**R**eliability-**C**entered **M**aintenance
RH	**R**ounding **H**euristic
SPP	**S**et **P**artitioning **P**roblem
SPL	**S**imple **P**lant **L**ocation
SA	**S**imulated **A**nnealing
SR	**S**hortest **R**oute
TS	**T**abu **S**earch
TBM	**T**ime-**B**ased **M**aintenance
VNS	**V**ariable **N**eighborhood **S**earch

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Quality and maintenance are amongst the most important concerns of every production system. In the current competitive world, high quality is at least as important as low price. Renovation of the machinery in production systems along with the advancement of technology enables them to retain their competitive feature in terms of quality. Nevertheless, technological advancement does not guarantee that the machinery do not break down during their life. A machine's condition can turn into an out-of-control condition gradually or at once. This implies that the products would not have their potential perfect quality. Besides, such situation questions the validity of the utilized lot sizing model, in that the portion of defective products in each lot increases, and the nominal production capacity of the machine decreases. The probability of shifting to an out-of-control condition will be diminished by scheduling periodic inspection and performing maintenance.

The objective of maintenance scheduling is to achieve the best timing to perform the maintenance actions in order to minimize the total maintenance cost. On the other hand, the objective of production planning is to minimize the total production and inventory costs while satisfying the demand for all products. Since production planning and maintenance scheduling are interdependent they usually cannot be planned sequentially, though many companies do it. Although the resulting model is more complicated, the synchronization between production planning and maintenance scheduling induces significant reduction in total cost incurred by production interruptions, delays, and replanning.

During the last few decades, exhaustive research has been carried out in the realm of lot sizing problem and EMQ (Economic Manufacturing Quantity) as a classic model has been exploited to develop more complicated realistic models by relaxing its underlying assumptions. The developed models mainly stand on the fact that the lack of maintenance not only yields low quality products, but also decreases the productivity of the machinery. Therefore, maintenance is an inevitable measure to avoid defect products and capacity reduction.

Although Iravani and Duenyas [2002] have demonstrated that making maintenance and production decisions separately can be rather costly and that there are significant benefits for making these decisions in an integrated fashion, the author's survey of the literature did not discover a study that deals, in particular, with an efficient formulation for (non-cyclic) maintenance and the effect of production interruptions as a result of maintenance on the lot sizing problem. In this thesis we are going to fill this research gap by developing some

efficient models. In spite of the fact that the essential assumption in the considered models is the same (the demand is time varying), they are distinctive in receiving the data for the future requests. Their objective is to schedule maintenance and fulfill the requests when the demand is given ahead of time (first models) or while there is no knowledge about the future requests (last model). For the known demand, we consider an integrated capacitated lot sizing and maintenance scheduling problem for which the motivation is inspired from a real production problem in Cement Industry. For the second assumption, when the demand is not known in advance, we consider different random instances of integrated production planning and maintenance scheduling problems to solve which different randomize and deterministic online algorithms are proposed.
The key contributions and findings of this research are as follows:

- We have developed a dynamic capacitated lot sizing and maintenance scheduling model with decreasing capacity. The novelty of this model is that it includes the capacity reduction and maintenance scheduling constraints.
- We have adapted the existing Fix-and-Optimize approach of Sahling et al. [2009] for the MIML-CLSP and combined it with a maintenance scheduling and transportation problem. We have found, at least for the considered class of MILML-CLSP instances, this yields better results than the off-the-shelve solver.
- We have applied the competitive analysis of on-line algorithms to the maintenance scheduling problem in which the demand in uncertain.

The structure of this chapter is as follows. The next section is the key section which describes the problem and discusses the integrated maintenance and production planning problem with non-increasing capacity. The general goals of our research and the motivation are also clarified in this section. In the second section the different types of maintenance are considered. Finally, in the last section the outline of the thesis is presented.

1.1 Motivation and Problem Statement

A strong motivation for the topic of the thesis has been inspired from a real production planning problem in the Cement Industry. Calucem was founded in 2006, which is located in Pula, Croatia. It is one of the leading suppliers of calcium aluminate cement, supplying high quality and specialty cement to both the refractory and building chemistry industries. Calucem exports calcium aluminate cements from Croatia to more than 60 countries worldwide. The positive sales trends and capacity utilization of the facility in Croatia have helped reinforce its position as the second largest manufacturer and supplier of calcium aluminate cement.
In this section, a global view of the different parts of the production process at Calucem is described.

1.1.1 Raw Material

At Calucem, different types of raw materials are used in the cement process such as Limestone, Bauxite, Coal, Alumine, etc. Each of the different types of raw materials is available from one or several distinct sources. For each of these sources, the price as well as the possibilities and cost of shipment may vary. Moreover, all sources can only provide a limited amount of material per month. for each source, there are several modes of transportation depending on their availability. At Calucem, the storage of raw material consists in a number of bunkers and open air spots around the site. Each of these bunkers or spots has a limited capacity, and can hold only certain types of raw material. Furthermore, a limestone quarry, which is also the main provider of limestone, can be used for extensive storage of extra material. This spot has a virtually unlimited capacity at the scale of of the cement plant, but the drawback is an extra cost due to the transportation to and back from the mine.

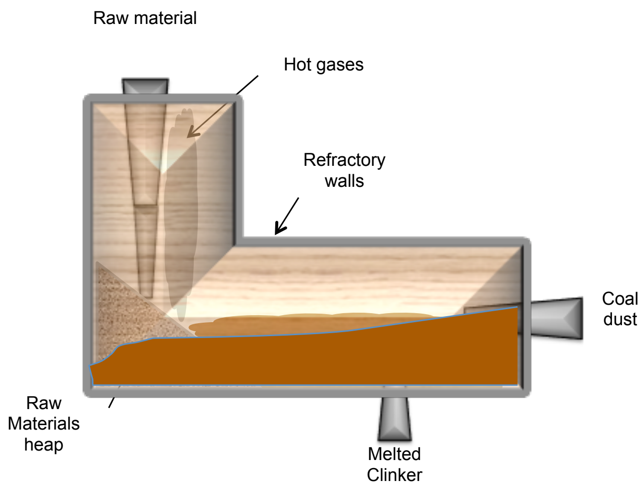
1.1.2 Kilns and Clinkers Procedure

A kiln [1] is an L-shaped furnace fed with raw material and coal, and yielding melted clinker. As illustrated in Figure 1.1, the kiln is mainly made of refractory bricks. Raw materials are brought through the upper opening, where they are preheated by combustion gases before stacking up. On the opposite end, coal dust is injected under pressure into the kiln and brings the temperature to $1400-1500°C$. This causes the raw materials to decompose, releasing mainly steam and CO_2, and to melt. This liquid is then collected as melted clinker, cooled down and stored.

The high temperature gradually weakens the kiln's wall. Moreover, the capacity of the kiln decreases gradually because of the deposit in its bottom. So, it

[1] An industrial oven

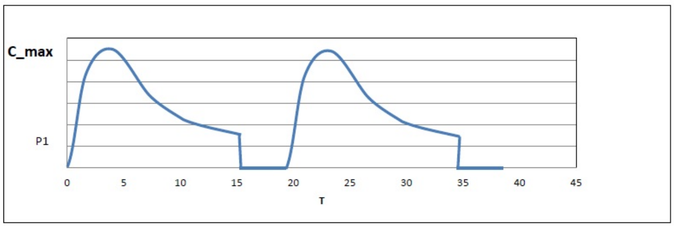
FIGURE 1.1: Capacity evaluation



becomes necessary to perform a global maintenance of the kiln either because the productivity is too low, or because of the rifts in the walls. During such maintenance, obviously the kiln cannot produce anything. The duration of a maintenance is 3 or 4 weeks, and only a limited number of kilns can be under the maintenance at the same time.

Moreover, the productivity of the kiln might be lower in the first periods following a maintenance (ramping up). As Figure 1.2 shows, a production cycle begins with a new system (in-control state), in which the kiln starts with a fraction of the nominal capacity to produce items of acceptable quantity and quality. After a few periods, it is able to reach its full capacity. Gradually, the productivity reduces over time, and at some point it becomes better to shut down the plant completely for a maintenance (out-of-control state), and restore the capacity to its nominal value. Then a new cycle begins. At Calucem, 8 kilns

FIGURE 1.2: Capacity evaluation



of approximately the same capacity are in use. The different types of cements are stored in 13 silos, 2 of large capacity and 11 smaller ones, and they are dedicated to specific types of clinkers.

1.1.3 Shipments Planning

Calucem produces 7 types of cement which are shipped to the client all over the world by sea transportation. The cost and the capacity of transportation varies over time because the company does not control the shipment operations. In particular, small ships with relatively high cost are available all-year round, while there are a few opportunities per year of cape-size boats, with much higher capacity and smaller per unit cost if fully loaded. Calucem's aim is to best satisfy the demand in the different final product types, while minimizing the total cost through the process. This includes potential compromises between shipping the products early or late with some penalty, and shipping it in an expensive way versus storing it longer on site to take advantage of a cheap vessel to come, among others. The Calucem's customers are mostly located in Europe and the United States of America. There is a local final product stock in Pula, which is managed by the company. Two other stocks are located in Sweden and USA.

A major problem with the current planning system of the company is that maintenance scheduling is done without taking into account the downstream transportation which results in substantial cost increases. So, the models we develop in this thesis aim at synchronizing maintenance and cheap transport to best satisfy the demand in the different final product types, while minimizing the total cost throughout the process. Therefore, the most important process in our research are the production process in the kilns and the transportation one.

1.2 Types of Maintenance

Maintenance is defined as a set of actions taken to prevent equipment from failing or to repair normal equipment degradation. The need for maintenance is predicated on actual or impending failure. Ideally, maintenance is performed to keep equipment and systems running efficiently, at least for the design life of the component(s). The design life of most equipment requires periodic maintenance. Belts need adjustment, wheel alignment needs to be maintained, and proper lubrication of rotating equipment is required. In some cases, certain components need replacement, (e.g., a wheel bearing on a motor vehicle) to ensure the main piece of equipment (in this case a car) lasts for its design life. Anytime we fail to perform maintenance activities intended by the equipment's designer, we shorten the operating life of the equipment. But what options do we have? Over the last 30 years, two approaches have been developed about how maintenance can be performed to ensure equipment meets its design life.

Rather than waiting for a component to fail (reactive maintenance), we can utilize proactive maintenance, which is performed without being caused by some form of failure.

1.2.1 Reactive Maintenance

Reactive maintenance is basically the “run it till it breaks” maintenance mode. No actions or efforts are taken to maintain the equipment, as the designer originally intended to ensure design life is reached. Advantages to reactive maintenance can be viewed as a double-edged sword. If we are dealing with new equipment, we can expect minimal incidents of failure. According to reactive maintenance, we should not expend manpower dollars or incur capital cost until something breaks. Since there is not any associated maintenance cost, this period could be regarded as a money-saving time. In reality, the cost during the time of saving maintenance is more compared to that under a different maintenance approach. This is due to the fact that while waiting for the equipment to break down, it wears out with a higher rate. Hence, the ramification could be more frequent replacements, and so more cost. The failure of one piece of a machine could expedite the failure of other pieces. This incurs a cost which could be avoidable if the maintenance program was more proactive.

1.2.2 Proactive Maintenance

Proactive maintenance primarily focuses on determining the root causes of machine failures, and dealing with the issues before problems occur. Proactive maintenance is often seen as a money-saving approach, because it allows the company to avoid machine failures and solves issues before they become problems. The goal of proactive maintenance is to view machine failures and similar problems as the anticipated problems to deal with before occurrence. one form of Proactive Maintenance is called Condition-Based Maintenance (CBM) based on which this research is defined. CBM typically involves active monitoring of the state of the equipment and predicting the systems' ability to perform without failure. When the condition is deemed too low, a maintenance is performed and the machine is restored to its nominal state [Garg and Deshmukh, 2006]. With CBM, organizations perform maintenance only when needed to prevent operational deficiencies or failures, to eliminate costly periodic maintenance, and to significantly reduce the likelihood of mechanical failures.

The objective of any form of maintenance is to buy the reliable utilization of a machine for some period of time. To achieve this objective, two other objectives should be pursued. First, we need to ensure that the reliability of the machine is as high as possible for the entire period of utilization. Second, little should be paid for each period of utilization. When considering the former objective, achieving the utilization using proactive maintenance as the mechanism of choice is likely to lead to higher reliability. However, the cost of reactive maintenance is much greater than a form of proactive maintenance providing an equivalent period of machine utilization Hu et al. [2012]. Therefore, taking into account proactive maintenance in maintenance scheduling usually provides more benefits than reactive maintenance (although the latter cannot completely be eliminated).

1.3 Outline of the Thesis

The main question which we try to answer during this research is ”what is the best time to do the maintenance?”. In order to do so, we consider two situations; the known and the unknown demand. The first part of the thesis (Chapter 2 & 3) is devoted to the integrated problem of production planning, maintenance and transportation scheduling when the demand is known in advance. The second part (Chapter 4) is dedicated to the integrated problem of production planning and maintenance scheduling when the future demand in unknown. Below, we provide an outline of each chapter.

Chapter 2 demonstrates that integrating maintenance scheduling and production planning can lead to substantial savings. However in these works the associated optimization problems are solved by enumerating all (exponentially many) maintenance schedules. This has led these researchers to either solve instances of very small sizes, or consider cyclic maintenance schedules only to limit the possibilities to a small (polynomial) number. We show here how to formulate these problems as (strong) mixed-integer linear programs, and then solve them using off-the-shelve MIP solvers. We demonstrate the efficiency of the proposed approach to solve problems of up to 10 products and 24 time periods, sizes that were simply unreachable before. We also illustrate the value of using non-cyclic maintenance schedules when the demand varies over time, with savings of up to 41% compared to cyclic schedules.

In Chapter 3, we characterize an improved formulation for DPLSP, as an extension of CLSP, and develop a Fix-and-Optimize heuristic to solve a multi-item, multi-machine, multilevel capacitated lot sizing problem (MIML-CLSP). The problem also includes condition-based maintenance and transportation schedules. The proposed model simultaneously determines the production plan, the timing of maintenance and the transportation decisions. The objective is to minimize the total production, maintenance, and transportation costs, while satisfying the demand of all products over the entire horizon without backlogging. In computational experiments the superiority of our Fix-and-Optimize heuristic over the MIP solver using various formulations and other classical heuristics is demonstrated.

In the second cluster, Chapter 4, we distinguish two classes of algorithms implemented to solve integrated lot sizing and maintenance problem with uncertain parameters, namely online algorithms and offline algorithms. Our main contribution is the application of competitive analysis of online algorithms to the mentioned problem. In computational experiments we compare the efficiency of two randomized online algorithms using different distributions for the arrival of the demands. The three studies presented in Chapter2-4 are currently in the process of getting published in scholarly journals.

After summarizing the main idea of the thesis in the beginning of Chapter 5, we conclude by pointing out the main research findings, and topics for possible future research linked to the integrated problems presented in Chapters 3 and 4.

Chapter 2

Solving integrated production and maintenance planning problems by MIP modeling

2.1 Introduction

Performing maintenance of productive equipment is a standard and natural practice in almost all industries. Without maintenance equipment will usually gradually consume more raw material and energy, operate at lower speed, lead to more defect products or to more down-time. The increase of consumption in raw material and energy, the increase of defect products and the corrective maintenance after failure will all increase the production cost per unit, while the lower speed of operation, the increase of defect products and the increase of down-time after failures will all decrease the nominal production capacity of the production system. Therefore, performing regular preventive maintenance to restore the equipment to an 'as-good-as-new' state is usual practice in most production systems. However, performing a maintenance is itself costly because it might necessitate the usage of specialized equipment or human skills and will also usually lead to a duration of reduced or zero capacity. Therefore plant managers do not want to perform maintenance too often either.

Lot sizing is the optimal timing and sizing of production lots in the face of dynamic, time-varying demand. The most important feature of lot sizing models are economies of scale related to production. More precisely, larger batches lead to lower production cost because of the presence of fixed costs, or to higher production capacity through the reduction of setup times. So the key trade-off in lot sizing models is between just-in-time, low-inventory but high-cost production plans, and production plans in large batches leading to low production cost and high capacity utilization but high inventory. Production planning generalizes lot sizing by also considering multiple products and multiple machines. An excellent reference book on this topic is Pochet and Wolsey [2006].

Maintenance scheduling and production planning are interdependent because the former essentially determines the production capacity that is used as input data of the latter. However, performing these two functions sequentially remains the usual approach in the industry, probably for simplicity and the belief that the inefficiency is small among other reasons. Nonetheless, it has been known since at least the work of Iravani and Duenyas [2002] that making maintenance and production decisions separately can be rather costly and that there are significant benefits for making these decisions in an integrated fashion. The reason is intuitive. If the maintenance schedule is determined without taking into account the full knowledge available about time-varying level of customer demand, it could be that a machine is in maintenance status during a period where full capacity is most needed.

This chapter of the thesis makes the following contributions. We give the first Mixed Integer Linear Programming (MILP) formulation for non-cyclic maintenance and the effect of production interruptions as a result of maintenance on the lot sizing problem. We also provide several alternative capacity models (i.e. different ways in which production decisions influence the production capacity) that substantially broaden the range of applicability of the proposed models. Next we show that these MILP formulations enable us to solve much larger problem instances than could previously be solved. Finally, we demonstrate the benefits of allowing non-cyclic maintenance schedules over cyclic ones.

The structure of this chapter is as follows. In the next section a brief literature review of the related works is given. Section 3 is the key section and presents the integrated maintenance and production planning problems and several variants and extensions, each time with the associated MILP formulation. An illustrative example is presented in section 4. Finally, we report on extensive computational experiments in section 5.

2.2 Literature Review

During last decade and even today, reactive maintenance or proactive maintenance have been existing widely in mechanical industry. The impact of machine breakdown and CM on the economic lot sizing decisions was first considered by Groenevelt et al. [1992a]. The subsequent literature has not been plentiful [Chakraborty et al., 2008], and has mostly been devoted to integrated production planning and maintenance scheduling where failure rate is deterministic or stochastic. The constant failure rate was considered by Groenevelt et al. [1992b] where it is assumed that a certain fraction of the items produced will be diverted to the safety stock in order to satisfy the demand during repair time of the machine. Since then several researchers [Kim et al., 1997] have extended the above seminal work to fit into various realistic situations. Various probability functions have been considered to reflect stochasticity of failure rates among which general distribution [Ben-Daya, 2002, Wang and Meng, 2009, Chakraborty et al., 2008], exponential distribution [Lee and Rosenblatt, 1989, Groenevelt et al., 1992a, Berg et al., 1994, Lin and Gong, 2006], Weibull distribution [Fituhi and Nourelfath, 2012, Lin et al., 2011], Gamma distribution [Aghezzaf et al., 2007] are the most frequent ones. Stochasticity of integrated production planning and maintenance scheduling, on the other hand, can be attributed to the required repair time to bring a machine back into an in-control state where distribution function of the repair time is exponential [Berg et al., 1994], or uniform [Chakraborty et al., 2008, Kim et al., 1997].

Maost researchers have considered cyclic (or periodic) maintenance in their PM models [Aghezzaf et al., 2007, Grigorieva et al., 2006]. In periodic PM we deal with optimizing how often each task in a set of predefined tasks should be executed. In comparison, there is a relatively limited literature on models presenting a general (not necessarily periodic) preventive maintenance policy [Fituhi and Nourelfath, 2012, Aghezzaf and Najid, 2008]. The objective of these models is to determine either the best time for doing preventive replacements by new items, i.e., perfect PM [Yao et al., 2004], or the optimal sequence for imperfect maintenance actions [Levitin and Lisnianski, 2000].

Since the integrated maintenance scheduling and production planning falls in the realm of combinatorial optimization, developing efficient solution methods is as important as developing efficient models. Hence, beside mathematical programming-based approaches [Ashayeri et al., 1996, Chakraborty et al., 2009, El-Ferik, 2008, Lee and Chen, 2000], Lagrangian relaxation and decomposition technique [Aghezzaf and Najid, 2008] are used to find bounds for these problems. Moreover, meta-heuristics [Sortrakul et al., 2005, Levitin and Lisnianski, 2000] have been extensively used to find good quality solutions.

In Ashayeri et al. [1996] a mixed integer linear programming model is developed to simultaneously plan preventive maintenance and production on multiple production lines, where each line has one bottleneck machine. The proposed model only indicates whether or not to produce a certain product in a certain period on a certain production line. The model schedules production jobs and preventive maintenance jobs, while minimizing costs associated with production, back orders, corrective maintenance and preventive maintenance. A branching solution procedure is suggested and its performance is discussed.

The model presented in Lee and Chen [2000] deals with the scheduling of production and preventive maintenance on a set of parallel machines where each machine must be maintained once during the planning horizon. The objective is to schedule jobs and maintenance activities so that the total weighted completion time is minimized. They studied two cases; in the first, the machines can be maintained simultaneously if necessary, and in the second only one machine can be maintained at any given time. They showed that, even when all jobs have the same weight, both cases of the problem are NP-hard. They also proposed a branch-and-bound algorithm based on column generation for solving both cases of the problem.

Sortrakul et al. [2005] have used Genetic Algorithm (GA) to solve the model of Cassady and Kutangolu [2005] in which they have proposed an integrated maintenance planning and production scheduling for a single machine, in order to find the optimal PM actions and job sequence minimizing the total weighted expected completion time.

Fituhi and Nourelfath [2012] have considered an integrated non-cyclic preventive maintenance and tactical production model for a single machine which determines simultaneously the optimal production plan and the instants of preventive maintenance actions. The problem is solved by enumerating all possible maintenance schedules in which the number of possibilities increases exponentially. They show substantial benefits for allowing non-cyclic maintenance schedules compared to cyclic ones.

2.3 Problem Statement

The exponentially decreasing capacity model that we consider is very similar to the one discussed in Aghezzaf et al. [2007], where they present a natural model of production with preventive maintenance. In the latter work, the probability of machine failure and loss of capacity increases over time, unless a preventive maintenance is performed. Since everything is dealt with in expectation, they eventually obtain a model with capacity decreasing exponentially over time, until maintenance is performed. Because they only consider cyclic maintenance schedules, there are only as many different maintenance schedules as time periods. An efficient solution procedure is then to enumerate all possible maintenance schedules. As we consider general (non-cyclic) maintenance, there are an exponential number of possible schedules. Therefore enumerating all maintenance schedules will not be an efficient approach in our case.

In our model, the deterioration of the equipment is modeled by a gradual reduction of the capacity. This is used to take various effects into account like increased defect products, decrease of production rate or increased needs of corrective maintenance after failure. Since we assume in the model that we know the capacity of the equipment at all times, the model we consider is closer to the concept of CBM. However, CM is also taken into account (although in expectation only) by the gradual reduction of capacity.

2.3.1 Mathematical Models

We consider a single machine on which a set of products P must be produced during a given planning horizon T consisting of N fixed periods with length τ, i.e., $T = N\tau$. For each product $i \in P$, a demand d_{it} must be satisfied at the end of period $t \in T$. The machine has a nominal capacity c_{max} which decreases exponentially at rate $0 \leq \alpha \leq 1$ in a production cycle. It is also assumed that if maintenance is performed in period $t \in T$ it starts from the very beginning of period t. Our integrated maintenance and production problem consists of a multi-product single-machine capacitated lot sizing problem and a maintenance problem. The decisions involve determination of lot sizes in each period and the periods in which maintenance is performed. Our objective is minimizing the total production, inventory, set up, and maintenance cost, while satisfying the demands for all products over the planning horizon. The required sets, parameters, and decision variables used in our model are the following.

Sets and Parameters

T: Set of periods in the planning horizon

P: Set of products

α: Capacity reduction coefficient

τ: Length of each period

f_{it}: Fixed production cost of product i in period t

p_{it}: Variable production cost per unit of product i in period t

h_{it}: Inventory holding cost per unit of product i by the end of period t

d_{it}: Demand for product i in period t

m_t: Maintenance cost in period t

c_{max}: Maximum (nominal) capacity of the machine

Decision variables

x_{it}: Lot size of product i in period t

I_{it}: Inventory level of product i at the end of period t

c_t: Available capacity in period t

y_{it}: $\begin{cases} 1 & \text{if product i is produced in period t} \\ 0 & \text{otherwise} \end{cases}$

z_t: $\begin{cases} 1 & \text{if maintenance is performed in period t} \\ 0 & \text{otherwise} \end{cases}$

2.3.1.1 The Integrated Non-cyclic Maintenance and Production Model(NCMP)

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

Subject to:

$$x_{it} + I_{i,t-1} = d_{it} + I_{it} \ ; \forall t \in T \ , \ i \in P \tag{2.2}$$

$$x_{it} \leq (\sum_{s \in T, s \geq t} d_{is}) y_{it} \ ; \forall t \in T \ , \ i \in P \tag{2.3}$$

$$\sum_{i \in P} x_{it} \leq c_t \ ; \forall t \in T \tag{2.4}$$

$$c_t = max(\alpha c_{t-1}, c_{max} z_t) \ ; \forall t \geq 2 \tag{2.5}$$

$$x_{it} \geq 0 \ , \ I_{it} \geq 0 \ , \ c_t \geq 0 \ , \ y_{it} \in \{0, 1\} \ , \ z_t \in \{0, 1\} \ ; \ \forall t \in T, i \in P \tag{2.6}$$

The objective function (2.1) consists of the total production, inventory, set up, and maintenance costs. The first set of constraints (2.2) is the flow conservation constraints defined for all products $i \in P$ and all periods $t \in T$. They guarantee that summation of the available inventory of a product and its produced quantity in each period is sufficient to satisfy demand of the pertinent period; the remainder inventory is obviously stocked for the subsequent periods. The second set of constraints (2.3) enforces $x_{it} = 0$ if $y_{it} = 0$ and does not impose a constraint on x_{it} if $y_{it} = 1$. In these constraints the quantity $\sum_{s \in T, s \geq t} d_{is}$ is an upper bound for x_{it}. Constraints (2.4) are the capacity restrictions defined for each period. They guarantee that the produced quantity in each period does not exceed the available capacity in the related period. Constraints (2.5) describe the evaluation of the capacity. If $z_t = 0$ and no maintenance is performed, c_t decreases exponentially at a rate of α. If $z_t = 1$ and maintenance is performed, c_t is restored to the nominal capacity c_{max}.

Note that the same problem without maintenance and given fixed capacities c_t is the well-known multi-item big-bucket lot sizing problem. It is strongly NP-Hard, and can be very difficult in practice [Miller et al., 2003].

The non-linear capacity function(2.5) can be linearized as follows.

$$c_t \leq c_{max} \quad ; \forall t \in T \tag{2.7}$$

$$c_t \leq \alpha c_{t-1} + c_{max} z_t \quad ; \forall t \geq 2 \tag{2.8}$$

2.3.1.2 More Complex Capacity Models

We describe here more complicated capacity models, and show that the proposed approach can accommodate many different situations. In particular, we consider situations where the capacity decrease depends on the previous production level, on the presence of a set up, or also follows an arbitrary given law between maintenances (i.e. other than exponential).

In some situations when a single machine is used to produce different products, the capacity of each period depends not only on the capacity of the previous period but also on the type of products and their lot sizes. For example, a kiln in Cement Industry can be used to produce various types of cement. Each product has its own composition, and has a specific affect on capacity reduction. It can be mathematically modeled as:

$$c_t = max(\alpha c_{t-1} - \sum_{i \in P} \beta_i x_{i,t-1}, c_{max} z_t) \;\; ; \; \forall t \geq 2 \tag{2.9}$$

Where capacity is additionally reduced by a term proportional to the production level $x_{i,t-1}$ and a given parameter $0 \leq \beta_i \leq 1$. This constraint is linearized by following the same technique we used for constraint (2.5):

$$c_t \leq c_{max} \;\; ; \; \forall t \in T \tag{2.10}$$

$$c_t \leq \alpha c_{t-1} - \sum_{i \in P} \beta_i x_{i,t-1} + c_{max} z_t \;\; ; \; \forall t \geq 2 \tag{2.11}$$

In lot sizing models, it is necessary to accurately model capacity utilization to obtain feasible production plans. This often requires one to model the capacity consumed when a machine starts a production batch, or when a machine switches from one product to another. In these cases, we obtain so-called set-up or start-up time models, changeover time models, or models with sequencing restrictions. In some applications, there is wear-and-tear especially when the equipment is started up. This is typical in the case of engines. To incorporate this into our model we introduce a new binary variable w_t equal to 1 if the equipment is started up in period t. Following Pochet and Wolsey [2006] (page

323), the start-up can be modeled as:

$$w_t \geq y_{it} - y_{i,t-1} \ ; \ \forall i, t \tag{2.12}$$

Indeed, a start-up has to be incurred if production of some item i occurs in period t, but not in period t-1. Then our capacity function can be altered as follows:

$$c_t = max(\alpha c_{t-1} - \gamma w_{t-1} c_{t-1}, c_{max} z_t) \ ; \ \forall t \geq 2 \tag{2.13}$$

where $0 \leq \gamma \leq 1$ is a non-negative coefficient represents loss of capacity. This can be similarly linearized as

$$c_t \leq \alpha c_{t-1} - \gamma c'_{t-1} + c_{max} z_t \ ; \ \forall t \geq 2 \tag{2.14}$$

where

$$c'_t = c_t w_t \ ; \ \forall t \tag{2.15}$$

Constraint (2.12) forces $w_t = 1$ if $y_{it} = 1$ and does not enforce any value on w_t if $y_{it} = 0$. The capacity in period t (c_t) is the maximum value of c_{max} and $(\alpha - \gamma)c_{t-1}$ if $w_t = 1$ and the maximum value of c_{max} and αc_{t-1} if $w_t = 0$.

$$c'_t \leq c_t \ ; \ \forall t \tag{2.16}$$

$$c'_t \leq c_{max} w_t \ ; \ \forall t \tag{2.17}$$

$$c'_t \geq 0 \ ; \ \forall t \tag{2.18}$$

$$c'_t \geq c_t - c_{max}(1 - w_t) \ ; \ \forall t \tag{2.19}$$

The set of constraints (2.16)-(2.19) linearizes constraint (2.15). They enforce $c'_t = c_t$ if $w_t = 1$ and $c'_t = 0$ if $w_t = 0$.

Constraints (2.11) and (2.14) can of course be combined to obtain the more general form:

$$c_t = max(\alpha c_{t-1} - \sum_{i \in P} \beta_i x_{i,t-1} - \gamma c'_t, c_{max} z_t) \ ; \ \forall t \geq 2 \tag{2.20}$$

In some applications, the required time to do maintenance is more than one period, i.e., for a couple of periods during maintenance there is no production. Since the required time for maintenance might occupy a few full periods plus a fraction of one period, the capacity of the first possible period after maintenance might be a fraction of the nominal capacity. For instance, in Cement Industry the maintenance actions take more than 2 weeks. Moreover, after the maintenance the kiln cannot produce with its full capacity.

We introduce the following new parameters to model these features:

p_k: the percentage of the nominal capacity available after k periods after the start of the last maintenance

Let us define the following new variable:

$$w_{tk}: \begin{cases} 1 & \text{if in period t, the last maintenance is performed in period k } (k \leq t) \\ 0 & \text{otherwise} \end{cases}$$

and the associated constraint

$$w_{tk} = w_{t-1,k}(1 - z_t) \ ; \ \forall t > k \tag{2.21}$$

Constraint (2.21) forces $w_{tk} = 0$ if $z_t = 1$, which means that the last maintenance in period t has been performed in the same period. The linear constraints (2.22)-(2.25) are equivalent to the set of the non-linear constraint (2.21) show this point better.

$$w_{tk} \leq w_{t-1,k} \ \ ; \ \forall t > k \tag{2.22}$$

$$w_{tt} = z_t \ \ ; \ \forall t > k \tag{2.23}$$

$$w_{tk} \leq 1 - z_t \ \ ; \ \forall t > k \tag{2.24}$$

$$w_{tk} \geq w_{t-1,k} - z_t \ \ ; \ \forall t > k \tag{2.25}$$

The general capacity function can then be written as follows:

$$c_t = (p_1 w_{tt} + p_2 w_{t,t-1} + p_3 w_{t,t-2} + ...)c_{max} \tag{2.26}$$

It implies that if we are in period t and maintenance is performed in this period then the capacity in period t is $p_1 c_{max}$, i.e., the machine has restored $p_1\%$ of its full capacity.

2.3.1.3 Cyclic Maintenance Schedules (CMP)

Several authors have described similar problems but with the additional restriction that the maintenance schedule must be cyclic. For all the capacity models presented, we show how to model this situation by adding the following constraints:

$$z_t + z_{t+k} \leq z_{t+2k} + 1 \ \ ; \ \forall t \in T \ , \ t + 2k \leq N\tau \tag{2.27}$$

$$z_t + z_{t+k} \leq z_{t-k} + 1 \ ; \forall t \in T \ , \ t - k \geq 1 \ , \ t + k \leq N\tau \qquad (2.28)$$

The set of constraints (2.27),(2.28) forces maintenance in period $t+2k$ and $t-k$ when maintenance occurs in period t and $t+k$. Therefore, only maintenance schedules of the form $\tau, \tau+\delta, \tau+2\delta, \tau+3\delta, \ldots$, where δ is the periodicity and $\tau \leq \delta$ is the period of the first maintenance, are feasible in this model. Clearly, a cyclic schedule is somewhat contradictory to the idea of a CBM, where the decision to perform a maintenance depends on the actual state of the machine. However, it is interesting to see that modeling cyclic schedules is rather simple, and we also use it in the next sections to compare our models and results with earlier works.

2.4 Numerical Example

We illustrate our modeling and algorithmic approach on a numerical example taken from Aghezzaf et al. [2007]. Let us consider a production process with the planning horizon of 10 production periods, each with an available maximal capacity of $c_{max} = 50$. The machine has to produce two kinds of products in lots so that the demands are satisfied. For each product, the demands (d), set up (f), production (p), holding (h), and maintenance costs (m) are presented in Table 2.1. The demands have been generated randomly in a specific range.

TABLE 2.1: Problem data for numerical example

Product	d_{it}	f_i	p_i	h_i	m
1	(0,30)	40	10	2	50
2	(0,30)	40	10	2	50

The value of using non-cyclic maintenance action when the demand varies from one period to another is illustrated. To get the optimal solution, this problem is solved with a linear mixed integer solver (Xpress). The following are the summary results for CMP and NCMP (Table 2.2).

TABLE 2.2: Optimal production and maintenance planning for CMP and NCMP

Period	CMP					NCMP				
	x_{1t}	x_{2t}	y_{1t}	y_{2t}	z_t	x_{1t}	x_{2t}	y_{1t}	y_{2t}	z_t
1	16	19	1	1	1	16	19	1	1	1
2	27	12.4	1	1	0	22	9	1	1	0
3	32	0	1	0	0	27	0	1	0	0
4	0	25.6	0	1	0	21	29	1	1	1
5	11	34	1	1	1	0	22	0	1	0
6	38	0	1	0	0	26.4	0	1	0	0
7	0	32	0	1	0	11.6	14	1	1	0
8	20	0	1	0	0	20	30	1	1	1
9	25	23	1	1	1	16	23	1	1	0
10	0	0	0	0	0	9	0	1	0	0
Total cost			4041.6					3990.8		

For 10 periods, we have only 10 possible cyclic maintenance schedules knowing that a maintenance is operated at the beginning of the planning horizon $(z_1 = 1)$. The total cost of an optimal integrated production and cyclic maintenance plan is 4041.6 and is obtained for $Z = (1, 0, 0, 0, 1, 0, 0, 0, 1, 0)$. In other words, the optimal integrated production and cyclic maintenance plan suggests that a maintenance is performed for the machine every 4 periods. On the other hand, if we remove the periodicity constraint in CMP, the total cost of an optimal integrated production and non-cyclic maintenance planning is 3990.8, and it is obtained for the combination of $Z = (1, 0, 0, 1, 0, 0, 0, 1, 0, 0)$. The optimal total cost obtained for NCMP is then lower than that for CMP. In fact, the removal of the maintenance periodicity constraint reduces the total cost by 1.26% in this case.

In order to validate the results of our modeling approach, we use the solution algorithm in Aghezzaf et al. [2007] where the integrated maintenance and production problem is solved by enumerating all possible maintenance schedules. Table 2.3 shows the results of implementing the latter algorithm for our example.

TABLE 2.3: Optimal production and maintenance planning for each maintenance senario

Period	Available capacities									
	$k = 1$	2	3	4	5	6	7	8	9	10
1	50	50	50	50	50	50	50	50	50	50
2	50	40	40	40	40	40	40	40	40	40
3	50	50	32	32	32	32	32	32	32	32
4	50	40	50	25.6	25.6	25.6	25.6	25.6	25.6	25.6
5	50	50	40	50	20.5	20.5	20.5	20.5	20.5	20.5
6	50	40	32	40	50	16.4	16.4	16.4	16.4	16.4
7	50	50	50	32	40	50	13.1	13.1	13.1	13.1
8	50	40	40	25.6	32	40	50	10.5	10.5	10.5
9	50	50	32	50	25.6	32	40	50	8.4	8.4
10	50	40	50	40	20.5	25.6	32	40	50	6.7
Maintenance cost	500	250	200	150	100	100	100	100	100	50
Production cost	3800	3808	3846	3891.6	-	-	-	-	-	-
Total cost	4300	4058	4046	4041.6	-	-	-	-	-	-

- infeasible solution

Unsurprisingly, the CMP by our modeling and Aghezzaf et al. [2007] both yield the same results indicated in Table 2.3. However, the optimal solution in Aghezzaf et al. [2007] is determined by varying the length of maintenance (k) as a parameter and solving 10 optimization problems whereas the cycle length is implicitly incorporated into our proposed model as a decision variable. This implies that our optimization problem is required to be solved only once.

2.5 Computational Experiments

In the previous sections, we have proposed novel MIP models for integrated lot-sizing and maintenance scheduling. In this section, our goal is to evaluate the performance of the proposed algorithmic approach (use a MIP solver to solve the proposed models). The computational experiments are divided into two parts. In the first part, sensitivity analysis is performed for NCMP1 in order to study the effect of changes of different parameters on the quality of the solution found (measured by the duality gap obtained at the branch-and-cut procedure) and the number of nodes. Moreover, the impact of adding Multi-Commodity (MC) formulation and Wagner-Whitin (WW) constraint for different sets of parameters is considered. In the second part, we compare the impact of using MC formulation for the other CM and NCM problems.

2.5.1 NCMP

In order to validate these conclusions, we have randomly generated many instances by varying some important parameters of the problem. Six parameters are considered for the analysis: number of products P, number of periods in the planning horizon T, capacity tightness w, the exponential decrease rate of capacity α, the set up cost f, and the maintenance cost m. The number of product P is selected from $\{5,10\}$, the number of periods T is selected from $\{12,24\}$. The demand for each product is assumed to be a random number in [0,40].

The factor w is introduced to control the capacity tightness. The nominal capacity is calculated according to $\sum_{i \in P, t \in T} d_{i,t}/T * w$, so that values of w slightly above 1 imply a tight production system. The cost structure is assumed to have the following properties: the set up cost for each product f_{it} is selected randomly from two different intervals [300,500] for low set up cost problem and [5000,8000] for high set up cost problems. The maintenance cost is selected randomly from [300,500] for low maintenance cost problems and [5000,8000] for high maintenance cost problems. The production cost is zero and holding costs are uniformly equal to 1.

For each combination of set up cost, maintenance cost and capacity tightness 5 test problems are generated. Each test problem is solved within a time limitation of 600 seconds to obtain the best feasible solution (Z_{BS}) and a lower bound (Z_{LB}) on the optimal of the problem. The duality gap is computed as follows:

$$Duality\ Gap = \frac{Z_{BS} - Z_{LB}}{Z_{LB}} * 100\% \tag{2.29}$$

Xpress-MP 7.8 using default settings is used to solve the problems on a single processor with 4Gb of RAM and clocked at 2.5GHz. Table 2.4 and Table 2.5 show the summary of the results obtained for model NCMP for different values of α and w.

The main message to be taken from Table 2.4 is that these medium-sized problems can all be solved to optimality in a few seconds to minutes. Moreover very high quality solutions are always available after a few seconds.

Other interesting aspects to notice are the following:

TABLE 2.4: Computational results of the medium size data set for NCMP

Case	Problem						Node	Time(s)	Gap(%)
	T	P	α	w	f_{it}	m_t			
1	12	5	0.8	1.6	[300,500]	[300,500]	10641	23.78	< 0.01
2	12	5	0.7	1.6	[300,500]	[300,500]	23414	99.52	< 0.01
3	12	5	0.8	1.2	[300,500]	[300,500]	50897	327.98	< 0.01
4	12	5	0.7	1.2	[300,500]	[300,500]	70796	557.95	< 0.01
5	12	5	0.8	1.6	[5000,8000]	[300,500]	1991	3.61	< 0.01
6	12	5	0.7	1.6	[5000,8000]	[300,500]	784	1.44	< 0.01
7	12	5	0.8	1.2	[5000,8000]	[300,500]	16419	59.16	< 0.01
8	12	5	0.7	1.2	[5000,8000]	[300,500]	13552	52.15	< 0.01
9	12	5	0.8	1.6	[300,500]	[5000,8000]	18244	51.88	< 0.01
10	12	5	0.7	1.6	[300,500]	[5000,8000]	8187	25.57	< 0.01
11	12	5	0.8	1.2	[300,500]	[5000,8000]	78880	620.86	< 0.01
12	12	5	0.7	1.2	[300,500]	[5000,8000]	34363	172.9	< 0.01

- when the ratio of set up cost to maintenance cost is very high the problem is easier to solve (e.g. case 1 vs. case 5 & 9),
- for different setup and maintenance costs, a faster rate of decrease of capacity (smaller α) is accompanied with a reduction in running time(e.g. case 9 vs. case 10),
- when setup cost is as large as maintenance cost, a faster rate of capacity reduction makes the problem more difficult to solve(e.g. case 1 vs. case 2),
- tightly capacitated systems are much more difficult to solve.

2.5.1.1 Wagner-Whitin Inequalities

To solve larger instances, it is necessary to improve the formulation of the problem. A well-known way to tighten the formulation of lot sizing problems is to add the so-called Wagner-Whitin (WW) inequalities to the original problem [Pochet and Wolsey, 2006]:

$$I_{i,t-1} \geq \sum_{u=t}^{l}(d_{i,u}(1 - \sum_{v=t}^{u}(y_{i,v}))) \ ; \forall t \in T \ , \ t \leq l \leq t + M \tag{2.30}$$

2.5.1.2 Multi-Commodity formulation

The other way to tighten the formulation of the original problem is to use the Multi-Commodity formulations. For lot sizing problems, this strong extended formulation has proved successful in a number of cases Pochet and Wolsey [2006]. We introduce the following variables:

xm_{itr}: Lot size of product i in period t to satisfy the demand of period r
IM_{itr}: Inventory level of product i at the end of period t after production to satisfy the demand of period r
The resulting Multi-Commodity formulation is as follow:

$$\begin{aligned} & xm_{itr} + IM_{i,t-1,r} = d_{ir}\delta_{tr} + IM_{itr}(1 - \delta_{tr}) \ ; \\ & \forall t \in T \ , \ r \in T \ , \ i \in P \ with \ t \leq r \end{aligned} \tag{2.31}$$

$$xm_{itr} \leq d_{ir} y_{it} \ ; \forall t \in T \ , \ r \in T \ , \ i \in P \ with \ t \leq r \tag{2.32}$$

$$x_{it} = \sum_{r \in T} xm_{itr} \ ; \forall t \in T \ , \ i \in P \ with \ t \leq r \tag{2.33}$$

$$xm_{itr} \geq 0 \ , \ IM_{itr} \geq 0 \ ; \quad \forall t \in T \ , \ r \in T \ , \ i \in P \tag{2.34}$$

where $\delta_{tr} = 1$ if $t = r$ and $\delta_{tr} = 0$ otherwise. The first set of constraints 2.31 are flow balance constraints. Constraint 2.32) is the tightened variable upper bound constraints, and the last one (2.33) links the original production variable to the corresponding MC variables.

We now experiment with instances with 10 items and a planning horizon of 24 periods. The other parameters are similar as before. For each combination, 5 test problems are solved by setting 600 seconds as the maximum running time. The results in Table 2.5 demonstrate that increasing the number of periods and products have significant effects on the gap. According to Table 2.5, tighter capacity ($w = 1.2$ compared to $w = 1.6$) yields larger average gaps when set up cost is at least as large as maintenance cost. On the other hand, tighter capacity yields smaller average gaps when set up cost is less than maintenance cost. Moreover, decreasing the value of α seems not to have much impact on the average gap. The results indicate that adding MC formulation to the original problem yields better solution and smaller gap for all the instances, however using WW constraint as an additional constraint results a better value for the lower bound in some instances.

2.5.2 Computational results for other CMPs with NCMPs

In this section, we experiment with the more complex capacity models of section 2.3.1.2. Our goal is to see whether the same performance can be expected as for the more basic capacity models. Again, we attempt to solve large instances with 24 time periods and 10 different products with the time limitation of 600 seconds. The results can be found in Table 2.6. $\bar{\beta}_i$ corresponds to the

TABLE 2.5: Computational results of the large size data set for NCMP

Case	Problem				Basic Model			+WW Inequalities			+MC formulations		
	α	w	f_{it}	m_t	BS	LB	Gap(%)	BS	LB	Gap(%)	BS	LB	Gap(%)
1	0.8	1.6	[300,500]	[300,500]	27350.8	26424.3	3.38	27582.7	26395.7	3.31	27004.1	26405.7	2.21
2	0.7	1.6	[300,500]	[300,500]	28311.4	27128	4.18	28155.4	27127.9	3.65	27920.5	27053.1	3.1
3	0.8	1.2	[300,500]	[300,500]	32154.7	29798.8	7.33	32065.7	29804.2	7.06	31421.4	29657.2	5.61
4	0.7	1.2	[300,500]	[300,500]	33292.9	31084.5	6.63	33067.3	31075.3	6.02	32518	30870.8	5.06
5	0.8	1.6	[5000,8000]	[300,500]	182055	167511.2	7.99	183666.4	171048	6.87	178124.4	169114.6	5.06
6	0.7	1.6	[5000,8000]	[300,500]	182326.6	168978.4	7.32	181893.4	169532.2	6.79	177675.4	167426.6	5.77
7	0.8	1.2	[5000,8000]	[300,500]	276281.6	235838.6	14.65	277949.4	235783.8	14.11	257920.4	229794.6	10.89
8	0.7	1.2	[5000,8000]	[300,500]	270633.2	230449	14.84	272408.6	230425.6	14.36	252786.4	225299.8	10.85
9	0.8	1.6	[300,500]	[5000,8000]	48644.26	46879.7	3.61	48543.2	46881	3.41	48122.9	46842	2.65
10	0.7	1.6	[300,500]	[5000,8000]	61838.5	57245.7	7.42	61786.3	56979.14	7.37	60694.8	57663.3	4.98
11	0.8	1.2	[300,500]	[5000,8000]	84781.6	80393.2	5.18	84378.78	80458.56	4.65	83605.44	80464.12	3.76
12	0.7	1.2	[300,500]	[5000,8000]	101293.58	96463.44	4.71	101418.94	96714.24	4.62	99261.9	96835.66	2.43

rate of capacity decrease per unit of production, γ is the relative decrease of capacity per set up, and $\bar{p_k}$ shows the average of the percentage of the capacity available after k periods after the last maintenance. Each line of the table corresponds to the average of 5 randomly generated instances. We also run each instance twice: once where general non-cyclic maintenance is allowed, and a second time where cyclic maintenance is imposed. Also, we experiment the impact of adding WW inequalities to improve the formulation.

The results indicate that

- More complicated capacity models do not make the problem more difficult (the gaps are similar),
- Multi-Commodity formulations substantially improve the results.
- Based on our experience with instances of smaller size, it is likely that the solutions obtained are close to optimum, even if the resulting gaps remain substantial (i.e. for small size instances, the optimum solution is usually found quickly, and the bulk of the computational time is used to improve the lower bound and prove optimality).
- when demand and costs are dynamic, using cyclic maintenance schedule can result in a total cost increase of up to 11% compared to non-cyclic maintenance schedules.

TABLE 2.6: Computational results for comparing CMPs with NCMPs

Problem							CMP			CMP+MC		
α	$\bar{\beta}_i$	γ	$\bar{p_k}$	w	f_{it}	m_t	BS	LB	Gap (%)	BS	LB	Gap (%)
0.8	0.31	-	-	1.2	[300,500]	[300,500]	39111.7	33457	14.46	36190.6	33814.6	6.57
0.8	-	0.35	-	1.2	[300,500]	[300,500]	35176.4	34688.3	1.39	35107.8	34751.5	1.01
0.9	-	-	0.62	1.6	[300,500]	[300,500]	36292.9	35703.5	1.62	36077.9	35652.1	1.18

Problem							NCMP			NCMP+MC		
α	$\bar{\beta}_i$	γ	$\bar{p_k}$	w	f_{it}	m_t	BS	LB	Gap (%)	BS	LB	Gap (%)
0.8	0.31	-	-	1.2	[300,500]	[300,500]	35050.8	31132.7	11.18	33706.3	30105.8	10.68
0.8	-	0.35	-	1.2	[300,500]	[300,500]	34439.1	32070.2	6.88	33121.2	31746	4.15
0.9	-	-	0.62	1.6	[300,500]	[300,500]	36434.1	35674.8	2.08	36232.1	35663.1	1.57

Chapter 3

Integrated production, maintenance and transportation planning by a fix-and-optimize heuristic

3.1 Introduction

Production planning consists in deciding how to transform raw materials or semi-finished products into final products in order to satisfy the demands while minimizing the total cost. Determining the quantity to produce for each item in each time period (Lot Sizing) is one of the most important problems in tactical production planning. Since the seminal papers by Wagner and Whitin [1958] and Manne [1958] in the late 1950s, a lot of research has been done on lot sizing problems. The single-item problem has been given special interest for its relative simplicity and for its importance as a subproblem of some more complex lot sizing problems.

In industrial contexts, lot sizing problem would be more complicated if there is a resource limitation (Capacitated Lot Sizing Problem (CLSP)). Indeed, CLSP deals with determining production quantities to satisfy demands where capacity of the production system is limited. In the literature, there are numerous articles dealing with CLSPs. Two general categories are available: single-level problems, where the demand for each item is independent of the demands for others, and multiple level problems, where the product structure consists of parent-component relationships, making the demand for intermediate and lower level items dependent on the demand for end items in higher levels. Within each category, problems can be constrained/unconstrained by resources. In sum, such classification results in four categories: single-level uncapacitated (SL-ULSP), single-level capacitated (SL-CLSP), multilevel uncapacitated (ML-ULSP), and multilevel capacitated lot sizing problems (ML-CLSP). In addition, each of these categories can be considered as single (multi)-item or single (multi)-machine lot sizing problems [Bahl et al., 1987]. A general overview of lot sizing problems can be found in Pochet and Wolsey [2006].

The production planning problem considered in this chapter is well positioned in their classification as a multi-item multi-machine multilevel capacitated lot sizing problem (MIML-CLSP).

The multi-item capacitated lot sizing problem (MI-CLSP) is a well known optimization problem that finds a wide variety of applications in real world problems. The problem consists in determining the quantity (lot sizes) and timing of production for several final products over a number of finite production periods to satisfy a given demand for each product in each period and to minimize the total cost. At each period, the production and holding cost functions are given, and the amount of production is subject to a capacity limit. This problem is NP-Hard in the strong sense [Chen and Thizy, 1990]. Hence, the problem is challenging from the computational point of view and yet interesting from the real world applicability perspective. When multiple parallel machines are available, the lot sizing problem does not only includes the timing and level of production, but also the allocation of production to the machines. The problem is complex as any product can be processed on any machine but with different process rates and costs.

This chapter proposes a MIP formulation and a Fix-and-Optimize heuristic for integrated MIML-CLSP, maintenance, and transportation problem. The model assumes that the machines used in the production process have limited capacity which decreases over time as a function of the previous capacity, the type of products, and their lot sizes produced in the previous period. We develop a mathematical programming-based heuristic that iteratively tries to improve a given feasible solution using a neighborhood search strategy.

The outline of this chapter is as follows. In the next section a brief literature review of the related works is given. Section 3 describes the problem and a MIP formulation of the MIML-CLSP is presented. In section 4 we present the Multi-Commodity reformulation of the problem which is used in the computational experiments. The heuristic algorithm to solve the problem is introduced in section 5. To analyze the performance of the heuristic, we compare the results obtained by our fix-and-optimize heuristic with the hybrid heuristic proposed by Melo and Wolsey Melo and Wolsey [2012] and a truncated MIP solver which are shown in section 6.

3.2 Literature Review

Plenty of works have been devoted to MI-CLSP, starting from the work of Trigeiro et al. [1989]. Exact solution methods are developed by Pochet and Wolsey [1991], and Belvaux and Wolsey [2001] through strengthening the LP formulations with valid inequalities and then using a mixed integer programming (MIP) solver. The second group of exact methods uses extended reformulations of the model, which basically adds new variables to the problem to strengthen it. Eppen and Martin [1987] have proposed the shortest path reformulation which is equivalent to the facility location reformulation but is smaller in size.
Because of the difficulty of MI-CLSPs, exact methods are not always able to solve instances of moderate size to optimality. Therefore, the use of heuristics and metaheuristics as solution methods for this problem is justified. Buschkuhl et al. [2010] have classified the heuristics used to solve MI-CLSPs into five major categories: mathematical programming-based heuristics such as Relax-and-Fix heuristic (R&F) [Stadtler, 2003], Fix-and-Optimize heuristic (F&O) [Pochet and Wolsey, 2006, Helber and Sahling, 2010, Sahling et al., 2009],

Lagrangian-based heuristics like Lagrangian relaxation [Trigeiro et al., 1989, Tempelmeier and Derstroff, 1996, Brahimi et al., 2010] and Lagrangian decomposition [Millar and Yang, 1994], decomposition and aggregation heuristics [Ozdamar and Bozyel, 2000], greedy heuristics [Nascimento et al., 2010], and metaheuristics such as Variable Neighborhood Search (VNS) [Hindi et al., 2003], Simulated Annealing (SA) [Ozdamar and Bozyel, 2000], Tabu Search (TS) [Hung et al., 2003], and Genetic Algorithm (GA) [Ozdamar and Bozyel, 2000].

Stadtler [2003] has proposed a MIP-based heuristic that solves a series of subproblems through internally rolling schedules with time windows. While lot sizing decisions are made sequentially within an internally rolling planning interval (or lot sizing window), and capacities are always considered over the entire planning horizon. For each submodel a model formulation based on the "Simple Plant Location" representation is developed and a MIP solver is used to solve the subproblems.

The seminal work of Trigeiro et al. [1989] illustrates how Lagrangian relaxation coupled with subgradient optimization can be used to solve small-medium size instances of MI-CLSPs. For each Lagrangian vector, they have applied a dynamic programming algorithm to independently solve to optimality a series of uncapacitated lot sizing problems, one for each item. If the Lagrangian solution is not feasible, a smoothing procedure is applied in order to restore feasibility whenever possible.

Tempelmeier and Derstroff [1996] have tackled MI-CLSP in decomposing the original problem in a set of uncapacitated problems that are solved to optimality. They have solved MI-CLSP as the core planning step of the broader Material Requirements Planning (MRP) system and considered the multi-level version of the problem, where one item is composed of a number of sub-assembly operations.

The approach presented by Hindi et al. [2003] is interesting for its use of a metaheuristic within a Lagrangian relaxation scheme which is solved via subgradient optimization. The algorithm uses a smoothing heuristic embedded into a Lagrangian relaxation scheme coupled with a variable neighborhood local search. The solution found by the smoothing scheme is further improved by taking it as a starting point for VNS algorithm. Hung et al. [2003] have solved CLSP with backlogging with an improved TS method. Neighboring solutions are generated by modifying the current setup pattern. The remaining LP is solved to optimality and post-optimization information is used to conduct the next move.

Pochet and Wolsey [2006] have described an improvement heuristic similar to R& F heuristic, which they call ”exchange”. It differs from R& F heuristic in that it does not relax the binary setup variables in the subproblems. At each step except a limited set of integer variables that are related to the subproblem at hand and are considered as decision variables, other integer variables are fixed at their best value found in the previous iterations. Each iteration deals with a new subproblem with a different subset of integer variables to be optimized. Note that when the last partition is solved, and in case that a better solution is found, this exchange procedure can be repeated (i.e., each subproblem may be solved more than once) until a local optimum is reached. Normally, this procedure needs to be given a feasible starting solution. The same approach is used in Helber and Sahling [2010] and Sahling et al. [2009] for multilevel CLSP, where the authors call it Fix-and-Optimize heuristic.

3.3 The Mathematical Model

We consider a production line consisting of parallel machines on which a set of products P must be produced during a given planning horizon T. For each product $i \in P$, a demand d_{it} must be satisfied at the end of period $t \in T$. Each time period represents a week. The machines have a nominal capacity c_{max} which decreases exponentially with coefficient β_i related to the quantity of product i. In order to construct the initial basic model, we assume that if a maintenance action is performed in period $t \in T$ it starts from the very beginning of period t. Our integrated problem consists of a multi-item multi-machine capacitated lot sizing problem, a maintenance problem and a transportation problem. The decisions include determination of lot sizes in each period, the periods in which maintenance is performed, the periods in which the transportation is done, and the type of the vessel used for transportation in that period. Furthermore, because of using the full capacity of the cheapest vessel the trade off between the inventory level in the production site and inventory level in the customer site (after transportation) is considered. Our objective is minimizing the total production, inventory, setup, maintenance, and transportation costs while satisfying the demands for all products over the planning horizon. Hereinafter, the required sets, parameters, and decision variables to construct our model are explicitly defined.

Sets and Parameters

T: Set of periods in the planning horizon

P: Set of products

K: Set of machines

V: Set of vessels to transport final products

β: Capacity reduction coefficient of quantity of products

t': the number of periods in which the capacity of the kiln in increasing p'_j: the percentage of the available capacity j periods after the last maintenance

f_{kt}: Fixed production cost of machine k in period t

p_{it}: Variable production cost per unit of product i in period t

h_{it}: Inventory holding cost per unit of product i by the end of period t

d_{it}: Demand for product i in period t

m_{kt}: Maintenance cost for machine k in period t

$c_{k,max}$: Maximum (nominal) capacity of machine k

cv_{vt}: Capacity of vessel v in period t

sv_{vt}: Shipping cost for vessel v in period t (per ton)

tv_{vt}: Transportation cost for vessel v in period t

hv_{it}: Inventory holding cost per unit of product i in period t after transportation

Decision variables

x_{ikt}: Lot size of product i on machine k in period t

xt_{ivt}: Lot size of product i is transported using vessel v in period t

I_{it}: Inventory level of product i at the end of period t after production

IT_{it}: Inventory level of product i at the end of period t after transportation

c_{kt}: Available capacity of machine k in period t

c'_{kt}: Available capacity of machine k in period t when the capacity decreases

y_{ikt}: $\begin{cases} 1 & \text{if product i is produced on machine k in period t} \\ 0 & \text{otherwise} \end{cases}$

z_{kt}: $\begin{cases} 1 & \text{if maintenance action for machine k is performed in period t} \\ 0 & \text{otherwise} \end{cases}$

w_{tkj}: $\begin{cases} 1 & \text{if in period t the last maintenance on machine k is performed in period j} \\ 0 & \text{otherwise} \end{cases}$

v_{vt}: $\begin{cases} 1 & \text{if vessel v is used in period t} \\ 0 & \text{otherwise} \end{cases}$

e_{kt}: $\begin{cases} 1 & \text{if the capacity of machine k is decreasing in period t} \\ 0 & \text{otherwise} \end{cases}$

The MIML-CLSP Model

$$\min \sum_{t\in T}\sum_{v\in V}\sum_{k\in K}\sum_{i\in P}(f_{kt}y_{ikt}+p_{it}x_{ikt}+h_{it}I_{it}+ht_{it}IT_{it}+m_{kt}z_{kt}+sv_{vt}xt_{ivt}+tv_{vt}v_{vt}) \tag{3.1}$$

Subject to:

$$\sum_{k\in K} x_{ikt} + I_{i,t-1} = \sum_{v\in V} xt_{ivt} + I_{it} \;\; ; \forall t \in T \;,\; i \in P \tag{3.2}$$

$$\sum_{v\in V} xt_{ivt} + IT_{i,t-1} = d_{it} + IT_{it} \;\; ; \forall t \in T \;,\; i \in P \tag{3.3}$$

$$xt_{ivt} \leq (\sum_{s\in T, s\geq t} d_{is})v_{vt}) \;\; ; \forall t \in T \;,\; i \in P \;,\; v \in V \tag{3.4}$$

$$\sum_{i \in P} x_{ikt} \leq c_{kt} \ \ ; \forall t \in T \ , \ k \in K \tag{3.5}$$

$$x_{ikt} \leq c_{max} y_{ikt} \ \ ; \forall t \in T \ , \ k \in K \ , \ i \in P \tag{3.6}$$

$$c_{kt} = \sum_{j=t-t'}^{t} (p'_{t-j+1} w_{t,k,j}) c_{max} + c'_{kt} \ \ ; \forall k \in K \ , \ t \geq 2 \tag{3.7}$$

$$\sum_{j=t-t'}^{t} w_{tkj} + e_{kt} = 1 \ \ ; \forall k \in K \ , \ t \geq 2 \tag{3.8}$$

$$\sum_{k \in K} z_{kt} \leq 2 \ \ ; \forall t \in T \tag{3.9}$$

$$\sum_{i \in P} xt_{ivt} \leq cv_{vt} v_{vt} \ \ ; \forall t \in T \ , \ v \in V \tag{3.10}$$

$$w_{tkj} = w_{t-1,kj}(1 - z_{kt}) \ \ ; \forall t > j \ , \ k \in K \tag{3.11}$$

$$\begin{aligned} & x_{ikt} \geq 0 \ , \ xt_{ivt} \geq 0 \ , \ I_{it} \geq 0 \ , \ IT_{it} \geq 0 \ , \ c_{kt} \geq 0 \ , \ c'_{kt} \geq 0 \ , \ y_{ikt} \in \{0,1\} \ , \\ & z_{kt} \in \{0,1\} \ , \ w_{tkj} \in \{0,1\} \ , \ v_{vt} \in \{0,1\} \ , \ e_{kt} \in \{0,1\} \ ; \\ & \forall t \in T \ , \ i \in P \ , \ k \in K \end{aligned} \tag{3.12}$$

The objective function (3.1) minimizes the total production, inventory, setup, maintenance, and transportation costs. The first set of constraints (3.2),(3.3) are the flow conservation constraints defined for all products $i \in P$ and all periods $t \in T$. They guarantee that summation of the available inventory of a product and its produced quantity on all machines (the transported quantity using all vessels) in each period is sufficient to satisfy the demand of the pertinent period; the remainder inventory is obviously stocked for the subsequent periods. $\sum_{v \in V} xt_{ivt}$ in constraint (3.3) shows the demand in constraint (3.2). The set of constraints (3.4) enforce $xt_{ivt} = 0$ if $v_{vt} = 0$ and do not impose any value on xt_{ivt} if $v_{vt} = 1$. In these constraints the quantity $\sum_{s \in T, s \geq t} d_{is}$ is an upper bound for xt_{ivt}. The set of constraints (3.5)-(3.6) are the capacity restrictions defined for each machine $k \in K$ in each period $t \in T$. They guarantee that the produced quantity on a machine in each period does not exceed the available capacity of that machine in the related period. They also ensure that a machine is set up to produce product $i \in P$ in period $t \in T$, if the product is produced in this period. Constraint (3.7) corresponds to the upper bound for each capacity in each period. Since the capacity function consists of two increasing and decreasing parts, constraint (3.8) determines the specific part of the function used in each period t. The number of machines which can be maintained simultaneously is limited through constraint (3.9). The capacity of vessel $v \in V$ in period $t \in T$ is restricted through constraint (3.10). Constraint (3.11) forces $w_{tkj} = 0$ if $z_{kt} = 1$, which means that the last maintenance in period t has been performed in the same period.

$$w_{tkj} \leq w_{t-1,kj}; \ \forall t > j+1 \ , \ t > 1 \ , \ k \in K \tag{3.13}$$

$$w_{tkt} = z_{kt}; \ \forall t > j+1 \ , \ t > 1 \ , \ k \in K \tag{3.14}$$

$$w_{tkj} \leq 1 - z_{kt}; \ \forall t > j+1 \ , \ t > 1 \ , \ k \in K \tag{3.15}$$

$$w_{tkj} \geq w_{t-1,kj} - z_{kt} \ ; \ \forall t > j+1 \ , \ t > 1 \ , \ k \in K \tag{3.16}$$

The set of constraints (3.13)-(3.16) shows the linear form of constraint (3.11). Since the capacity function in constraint (3.7) consists of two phases, increasing phase after performing maintenance and decreasing phase after reaching the full capacity, we define a binary variable e_{kt} which determines the capacity function used in each period. $e_{kt} = 0$ implies the capacity increasing to reach its full capacity and $e_{kt} = 1$ shows the capacity decreases over time until the kiln is turned off in order to perform the maintenance. This decision variable has a direct impact on the capacity function (3.7) and transforms it to a nonlinear constraint. However, capacity function (3.7) can be linearized as follows.

$$c'_{k,t} \leq (c_{k,t-1} - \sum_{i \in P} \beta_i x_{ik,t-1}) \;\; ; \forall t \in T \; , \; k \in K \tag{3.17}$$

$$c'_{k,t} \leq e_{kt} c_{max} \;\; ; \forall t \geq 2 \; , \; k \in K \tag{3.18}$$

3.4 Multi-Commodity formulation

We now present a Multi-Commodity (MC) formulation for the MIP problem we have considered in section 3. For lot sizing problems, this strong extended formulation has proved successful in a number of cases Pochet and Wolsey [2006]. We introduce the following variables:

xm_{iktr}: Lot size of product i on machine k in period t to satisfy the demand of period r

xtm_{ivtr}: Lot size of product i for transportation using vessel v in period t to satisfy the demand of period r

IM_{itr}: Inventory level of product i at the end of period t after production to satisfy the demand of period r

ITM_{itr}: Inventory level of product i at the end of period t after transportation to satisfy the demand of period r

The resulting Multi-Commodity formulation is as follow:

$$\begin{aligned}&\sum_{k \in K} xm_{iktr} + IM_{i,t-1,r} = \sum_{v \in V} xtm_{ivtr} + IM_{itr} \ ; \\ &\forall t \in T \ , \ r \in T \ , \ i \in P \ with \ t \leq r\end{aligned} \tag{3.19}$$

$$\begin{aligned}&\sum_{v \in V} xtm_{ivtr} + ITM_{i,t-1,r} = d_{ir}\delta_{tr} + ITM_{itr}(1 - \delta_{tr}) \ ; \\ &\forall t \in T \ , \ r \in T \ , \ i \in P \ with \ t \leq r\end{aligned} \tag{3.20}$$

$$xm_{iktr} \leq d_{ir} y_{ikt} \ \ ; \forall k \in K \ , \ t \in T \ , \ r \in T \ , \ i \in P \ with \ t \leq r \tag{3.21}$$

$$xtm_{ivtr} \leq d_{ir} v_{vt} \ \ ; \forall v \in V \ , \ t \in T \ , \ r \in T \ , \ i \in P \ with \ t \leq r \tag{3.22}$$

$$x_{ikt} = \sum_{r \in T} xm_{iktr} \ \ ; \forall k \in K \ , \ t \in T \ , \ i \in P \ with \ t \leq r \tag{3.23}$$

$$\begin{aligned}&xm_{iktr} \geq 0 \ , \ xtm_{ivtr} \geq 0 \ , \ IM_{itr} \geq 0 \ , \ ITM_{itr} \geq 0 \ ; \\ &\forall t \in T \ , \ r \in T \ , \ i \in P \ , \ k \in K\end{aligned} \tag{3.24}$$

where $\delta_{tr} = 1$ if $t = r$ and $\delta_{tr} = 0$ otherwise. The first two sets of constraints (3.19)-(3.20) are flow balance constraints on production machines and transportation vessels. The next two (3.21)-(3.22) are the tightened variable upper bound constraints, and the last one (3.23) links the original production variable to the corresponding MC variables.

3.5 The Fix-and-Optimize algorithm

Since the problem we consider in this chapter is strongly NP-Hard, we expect the solution time to optimality to increase exponentially with the size of the problem. However, we might be able to quickly solve instances of small sizes. The basic idea of Fix-and-Optimize heuristic [Sahling et al., 2009] is to solve a series of MIP problems derived from MIML-CLSP in an iterative fashion. In each subproblem most of the binary variables are tentatively fixed to 0 or 1. Only a subset of binary variables of the original problem is treated as decision variables and are optimized using a standard MIP solver. Since the number of free (i.e. not fixed) binary variables of the subproblem is much smaller than the original problem, the solution time for a subproblem is much smaller. The optimal solution of the subproblem yields a new temporary solution for the free binary variables of the current subproblem. Some of them are fixed in successive subproblems when a new subset of binary variables is optimized. In each subproblem the complete set of continues (i.e. not fixed) decision variables are considered. The major advantage of F&O heuristic is that in each iteration a feasible solution is obtained. The notation used for defining sets of F&O variables is given as follows.

Sets and Parameters

Φ_ϕ ; $\forall \phi \in \{y, z, v\}$: Set of the variables

Φ_ϕ^{opt} ; $\forall \phi \in \{y, z, v\}$: Set of index combinations for which binary variables are optimized in the current subproblem.

Φ_ϕ^{fix} ; $\forall \phi \in \{y, z, v\}$: Set of index combinations for which binary variables are fixed in the current subproblem.

$\Phi_y : \{(i, k, t) : i \in P, k \in K, t \in T\}$

$\Phi_z : \{(k, t) : k \in K, t \in T\}$

$\Phi_v : \{(v, t) : v \in V, t \in T\}$

$\bar{y}_{ikt}$: Values to which the binary variable y_{ikt} is fixed.

$\bar{z}_{kt}$: Values to which the binary variable z_{kt} is fixed.

$\bar{v}_{vt}$: Values to which the binary variable v_{vt} is fixed.

Model MIML-CLSP-SUB

By using the above notation, the subproblem for a given set of fixed binary variables can be stated as follows. Minimize objective function (3.1) subject to constraints (3.2)-(3.18) and the additional constraints (3.25)-(3.27).

$$y_{ikt} = \bar{y}_{ikt} \;\; ; \; \forall (i, k, t) \in \Phi_y^{fix} \tag{3.25}$$

$$z_{kt} = \bar{z}_{kt} \;\; ; \; \forall (k, t) \in \Phi_z^{fix} \tag{3.26}$$

$$v_{vt} = \bar{v}_{vt} \;\; ; \; \forall (v, t) \in \Phi_v^{fix} \tag{3.27}$$

These additional constraints suffice to limit the optimization of the binary variables to the set (3.28):

$$\Phi_\phi^{opt} = (\Phi_\phi) \setminus \Phi_\phi^{fix} \;\; ; \; \forall \phi \in y, z, v \tag{3.28}$$

3.5.1 Definitions of subproblems

F&O heuristic is based on decompositions of the original problem that results in multiple subproblems with a smaller number of binary variables. The numerical effort to solve each subproblem depends on the number and the type of binary variables considered simultaneously. An increasing number of free binary variables solved to optimality in each subproblem raises the required solution time. Therefore, the definition of subproblems is important. We define four different strategies to identify the binary variables considered in the current subproblem.

1. Time decomposition: Each subproblem corresponds to a time-window of 2 consecutive periods. In each subproblem all binary variables are treated as free and optimized while all the others are fixed. The subproblems are considered in ascending order of time periods 1,...,T.

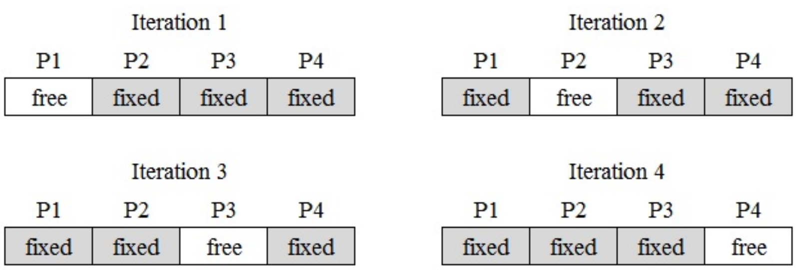
2. Product decomposition: There is one subproblem for each product. In each subproblem, only the setup variables $y_{i,k,t}$ related to a given product i are free and optimized, all other binary variables are fixed. In particular, all periods $t \in T$ and machines $k \in K$ are treated in each subproblem.

3. Resource decomposition: In each subproblem, only the setup variables $y_{i,k,t}$ of products that are manufactured on a given machine k within a time period of τ consecutive periods are free and optimized. All products $i \in P$ requiring machine k are considered in each subproblem.

4. Vessel decomposition: This is a subproblem where only but all the transportation decisions are free and optimized.

Each type of decomposition reflects a specific perspective on the overall optimization problem. For example, Figure 3.1 exemplifies the principle of F&O algorithm showing the iterations (subproblems) of the product decomposition for a small example with 4 products [Lang and Shen, 2011]. In each Iteration,

FIGURE 3.1: Fix-and-Optimize algorithm with product decomposition



only the set up variables of a single product are optimized, and the other variables are fixed to 0 or 1.

The decomposition types can be combined in different ways. Based on the preliminary results obtained from solving different instances, the following six variants are chosen for our proposed algorithm:

1. Variant 1 (T): Time decomposition only.
2. Variant 2 (TR): Time decomposition first, then resource decomposition.
3. Variant 3 (PRT): Product decomposition first,then resource decomposition, finally time decomposition.

4. Variant 4 (TPVR): Time decomposition first, then product decomposition, thereafter vessel decomposition, finally resource decomposition.

5. Variant 5 (PVTR): Product decomposition first,then vessel decomposition, thereafter time decomposition, finally resource decomposition.

6. Variant 6 (TR-PVTR): Variant 2 for some fixed amount of time (70 seconds) first, then Variant 5.

We observed that it can be worthwhile to treat each subproblem more than once, in particular until a local optimum is reached, which suggests an iterative layout of the heuristic.

3.5.2 Iterative Algorithm

The basic structure of our F&O algorithm is outlined in Algorithm 1. The algorithm starts by generating an initial feasible solution which yields an initial objective value Z_{new}. In order to obtain the initial feasible solution, we assume that each setup variable for all products on all machines in each period is equal to 1 ($y_{ikt} = 1$), and all vessels are available during all time periods ($v_{vt} = 1$). Since the maintenance action can be done at most for two machines at each period (3.9), the maintenance is scheduled on the machines every 7 periods in order to have enough capacity to satisfy the demands. After the initialization, all binary variables are fixed to their values in that solution. Therefore, the algorithm iterates through the ordered set of subproblems until it reaches the last decomposition order. Then it terminates and reports the solution that leads to the local optimum Z_{old}. In our algorithm, all problems and subproblems are not solved to optimality, but a relative MIP gap tolerance of 0.04 is used. The solution yields an objective function value Z that is at least as good as the currently best value Z_{old}. In each iteration, a new solution is accepted if it yields lower cost than the current solution.

Algorithm 1 Fix-and-Optimize Heuristic Algorithm

$(status, solution) \leftarrow Generate\ Initial\ Solution\ ()$
if $Status = No\ Feasible\ Solution\ Found$ **then**
  (status, $\oslash$); **return**
else
  repeat
    $Determine\ Objective\ Function\ Value\ Z$
    $Z_{new} \leftarrow Z$
    $\Phi_{\phi}^{fix} \leftarrow Save\ Decision\ Variables'\ Value\ \forall\phi \in \{y, z, v\}$
    for $s \in LS\ (List\ of\ Subproblems)$ **do**
      $Z_{old} \leftarrow Z_{new}$
      $Determine\ \Phi_{\phi}^{fix}\ \forall\phi \in \{y, z, v\}$
      $\Phi_{\phi}^{opt} = \Phi_{\phi} \setminus \Phi_{\phi}^{fix}\ \forall\phi \in \{y, z, v\}$
      $(status, solution) \leftarrow solve\ problem$
      if $status = Feasible\ Solution$ **then**
        $Z \leftarrow Objective\ Function\ Value$
        if $Z < Z_{old}$ **then**
          $Z_{new} \leftarrow Z$
          $Update\ Solution\ Values$
        end if
      end if
    end for
  until $Stopping\ Criterion\ is\ Met$ **return** (status,solution)
end if

3.6 Computational Experiments

The goal of these computational experiments is twofold:

- To compare the performance and the quality of the solution found with a MIP solver with other solution methods proposed in the literature as well as the proposed MIP heuristic run for the same amount of time.
- To determine the best combination and order of the various types of proposed decompositions (Time, Resource, Product, Vessels) within a fixed time budget.

More specifically, we compare the result of different variants of F&O heuristic of section 3.5.1 with the result of the following solution methods:

- run the MIP solver on formulation (3.1)-(3.18) for a limited amount of time and report the best solution found (Solver).
- run the MIP solver on the stronger multi-commodity formulation of Section 4 for a limited amount of time and report on the best solution (MC) and best lower bound (LB-MC) found.
- LP-based heuristic of Melo and Wolsey [2012].

We could not obtain sufficient data for solving the real-world MIML-CLSP application in Cement Industry due to difficulties of gathering the required data and reasons of confidentiality. Hence, we generated problem instances randomly, while trying to make their structure as realistic as possible, setting capacities rather tight. A total number of 100 different problem instances of small size, 10 different problem instances of medium size, and 100 different problem instances of large size with up to 20 periods, 7 products, 8 machines, and 6 vessels are grouped into three classes as described in 3.1. For each

TABLE 3.1: Dimensions of the test sets

	Instances	Time Period	Products	Kilns	Vessel types
small size	100	12	4	1	1
medium size	10	20	4	3	3
large size	100	20	7	8	6

product, the demand, set up, production, holding, transportation and maintenance costs are drawn randomly from a Uniform distribution with the parameters presented in Table 3.2. The factor w is introduced to control the capacity tightness. The nominal capacity is then calculated according to equation $\sum_{i \in P, t \in T} d(i,t)/NT/NM * w$, so that values of w slightly above 1 imply a tight production system. We set an absolute time limit for the MIP heuristics

TABLE 3.2: Problem's data

Parameter	Value
w	1.5
β_i	(0,0.3)
d_{it}	(0,40)
f_{it}	(300,500)
p_{it}	0
h_{it}	(50,100)
m_t	(1000,2000)
cv_{vt}	(15000,25000)
sv_{vt}	(10,20)
tv_{vt}	(10,20)
hv_{vt}	(50,100)

to 10 minutes. Xpress-MP 7.4 with default settings is used as our MIP solver. The experiments are run on a single processor with 4GB of RAM clocked at 2.5GHz.

To compare different methods across the different instances, the objective values obtained from all algorithmic approaches are normalized by dividing by the best solution known for each instance. The results we obtained are summarized in Figures 3.2 and 3.3, one for each instance class. Each curve corresponds to one heuristic approach. The curve corresponds to the average of the 10 instances for the medium size and 100 instances for the large size of the problem class. The curve gives, over time, the average of the (normalized) solution cost obtained by the corresponding heuristic. The average does not take into account the instances for which no solution has been found yet. This explains why the curve can sometimes increase.

A first qualitative observation that we made while running our experiments is that the heuristic of Melo and Wolsey [2012] regularly fails to provide feasible solutions to the problem. Indeed, this solution method relies on the LP relaxation solution values to fix some binary variables to zero. But the LP relaxation is typically weak and substantially overestimate the capacity available. The fixing then renders the problem infeasible due to lack of capacity. We tried various small modifications of these heuristics, but without being able to overcome this issue.

A second observation is that during the first 400 seconds, the multi-commodity formulation, while making the LP relaxation stronger, is penalizing. However, it outperforms the solver after 600 seconds for the large size data set. For the larger instances, the solver fails to find a feasible solution during the first minutes. And when it finds one, it is of extremely poor quality. The reason is that already solving the LP relaxations can take one minute, and no substantial enumeration can be done within this time period. Therefore the only feasible solutions reported by the MIP solver are those found by the built-in heuristics at the root node, and these are typically of bad quality (high cost). For the MC reformulation, the number of instances on which the average value of the normalized objective function is calculated are shown at three time points in (Figure 3.3).

The solver is capable to solve all the small size instances to optimality at most in 7 seconds, where the average computation time is less than 3 seconds. Therefore, there is no need to use heuristic approaches for these instances of small sizes. Compared to our heuristic, the solver obtains better solutions only for 40% of the medium size instances. Nevertheless, the solver is the loser against our heuristic in 95% of the large size instances. Our experimental results show that the larger the size of the problem is, the less efficient the employed solver is.
As it is shown, for both classes of instances most of the variants of F&O heuristic provide high quality results for the MIML-CLSP with respect to both solution time and solution quality. However, the superiority of F&O is more visible for the large size instances (Figure 3.3). Comparing F&O variant that only uses time decomposition (variant 1) to that which employs time and resource decompositions (variant 2), we found that the resulting solution quality was much worse. Therefore we decided to discard this variant from the rest

of our computational experiments for the second class of instances. Furthermore, during the first 70 seconds, variant 2 results in better solution but finally, variant 5 converges to 1 faster than the other variants. So, we have decided to combine these two variants which has been defined as variant 6. It means that during the first 70 seconds, variant 2 is executed and variant 5 is used for the rest of the time. For the first class of instances which is more easier to solve than the second one, variant 5 outperforms variant 6. However, the combination of both variant 2 and variant 5 leads to better solution for the second class of instances. For the second class, the solver obtains much better solution quality during the first 70 seconds but all the variants lead to better solution for the rest of the time.

3.6.1 Initial Feasible Solution

As it is described in section 3.5.2, the proposed Fix-and-Optimize heuristic algorithm starts with an initial feasible solution and iterates through the ordered set of subproblems until it reaches the last decomposition order. One approach for obtaining an initial feasible solution is to fix some binary variables to the predefined values which results were shown in Figures 3.2 and 3.3. Another method to define the initial solution is to run the MIP solver for the couple of seconds and then continue with the proposed F&O heuristic for the remaining time.

To evaluate the performance of the proposed method to obtain the initial feasible solution on the solution quality of F&O heuristic, we have randomly generated 10 different problem instances of large size with up to 20 periods, 7

products, 8 machines, and 6 vessels. we apply the most efficient variants of the F&O heuristic from our previous computational experiments to solve the instances. First, we run the solver for 100 seconds, then the F&O heuristic continues solving the instances for the remaining 500 seconds using the initial feasible solution obtained by the solver at the 100th second (The results obtained by assigning 100 (resp. 500) seconds to the solver (resp. F&O heuristic) are superior to those obtained when the times are considered 50 (resp. 550) seconds. Therefore, only the results of the former time division are reported.) The results have been compared to those of the F&O heuristic with the predefined initial feasible solution which are shown in Figure 3.4. In order to make the chart more visible, we have removed the curve related to the lower bound obtained via adding the multi-commodity formulation to the original problem. At the beginning, the lower bound is equal to 0.56 and after the time limitation of 600 seconds, it converges to 0.58. Moreover, we have added another figure (4.1) with the zoomed in curves to see better what happens between the different strategies.
The results indicate that:

- The XXX100 strategies outperform the XXX strategies, for XXX equal to TPVR, PVTR or TR-PVRT. Therefore, using the MIP solver to generate good initial solutions pays off for all F&O variants. In other words, the strategies built in the solver to generate from scratch feasible solutions are better than trying to improve a bad feasible solution by F&O.
- The XXX100 strategies eventually (after 600 seconds) outperform the solver, even though they are identical up to 100 seconds of time. This

means that it pays off to try hard improving already good solutions by F&O, rather than trying to generate other, better solutions from scratch. In the terminology of heuristics, intensification seems to be a better general strategy than diversification within this time span (100 to 600 seconds).

- The above two points are consistent with the fact that F&O is a local-search heuristic: it tries to modify slightly an existing solution to improve it.

- Regarding the order in which the decompositions in F&O should be run, the results we get are consistent with the previous results on the large instances. It seems that the time (T) and resources (R) decomposition are good at finding the ”low-hanging fruits”, and should be run first to generate the quicker gains first. The other decompositions should be run later to obtain further, but smaller, gains.

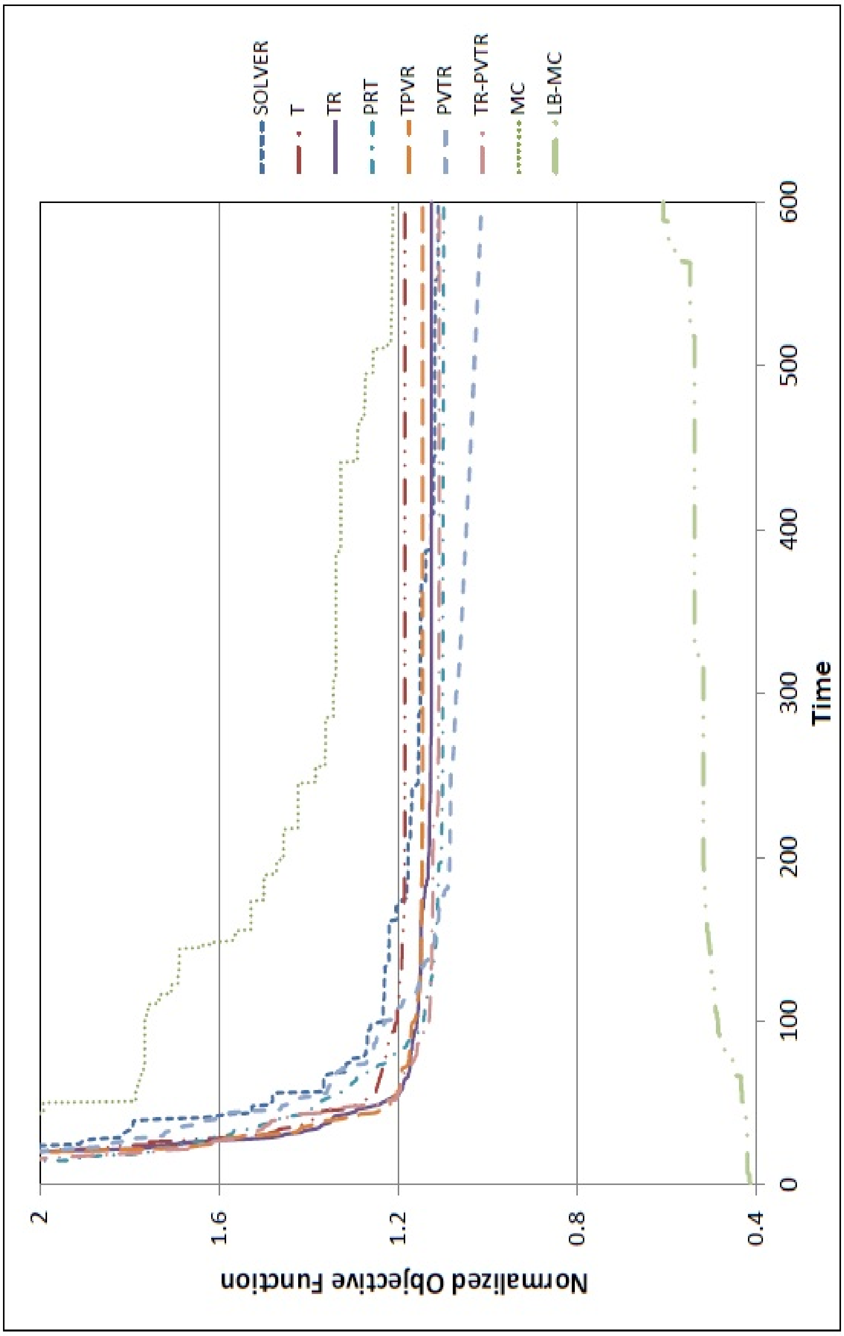


FIGURE 3.2: Results for medium size data set

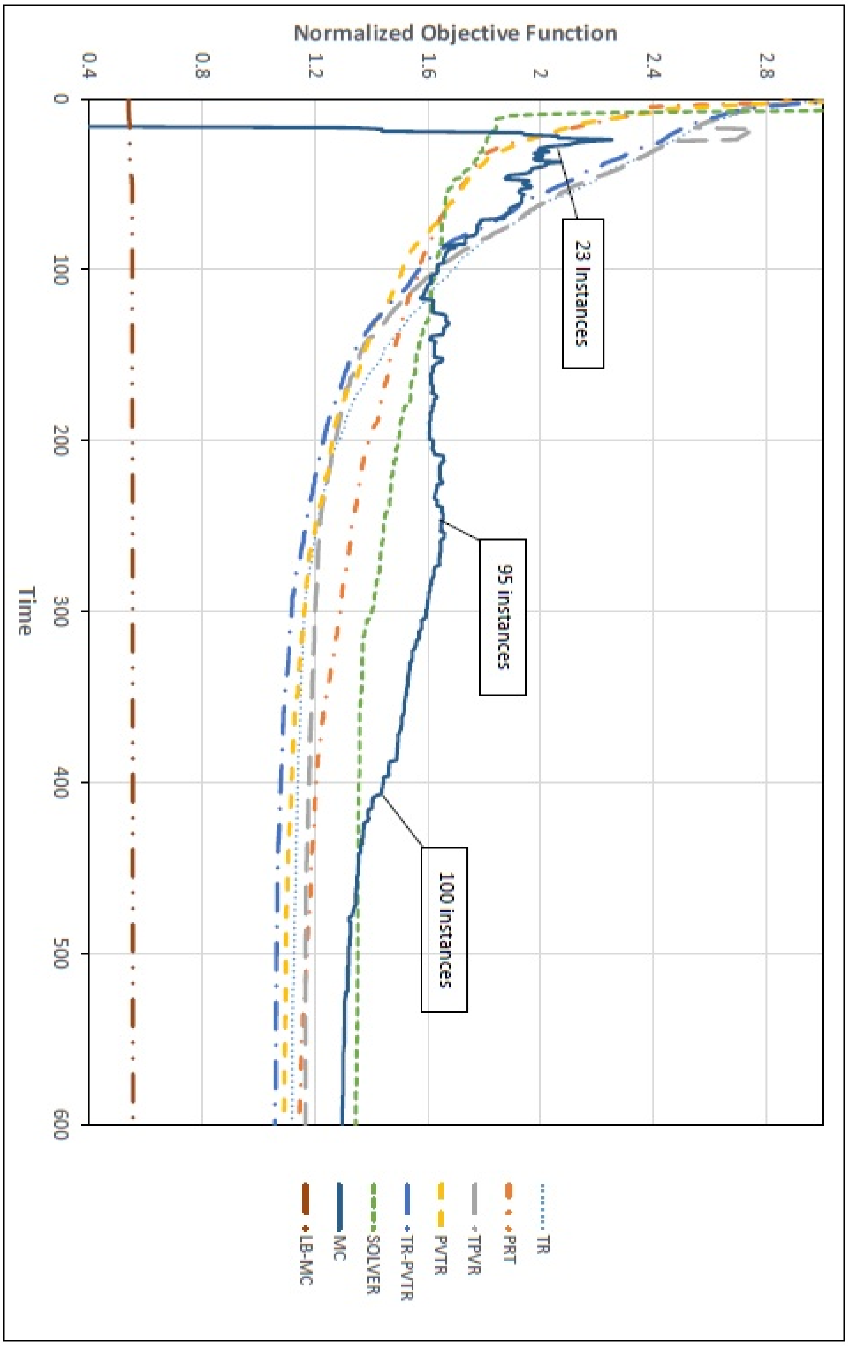


FIGURE 3.3: Results for large size data set

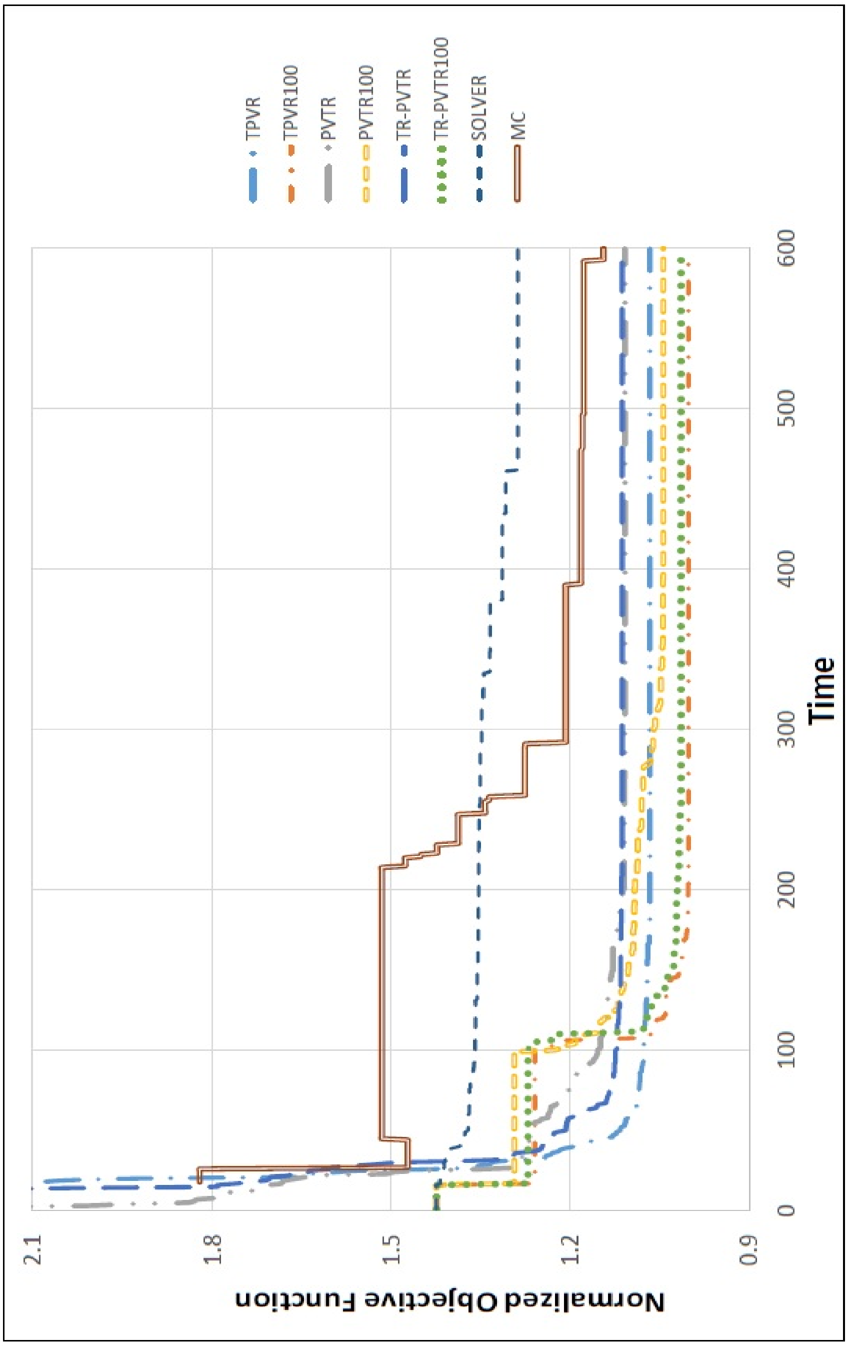


FIGURE 3.4: Results for large size data set to compare the feasible initial solution

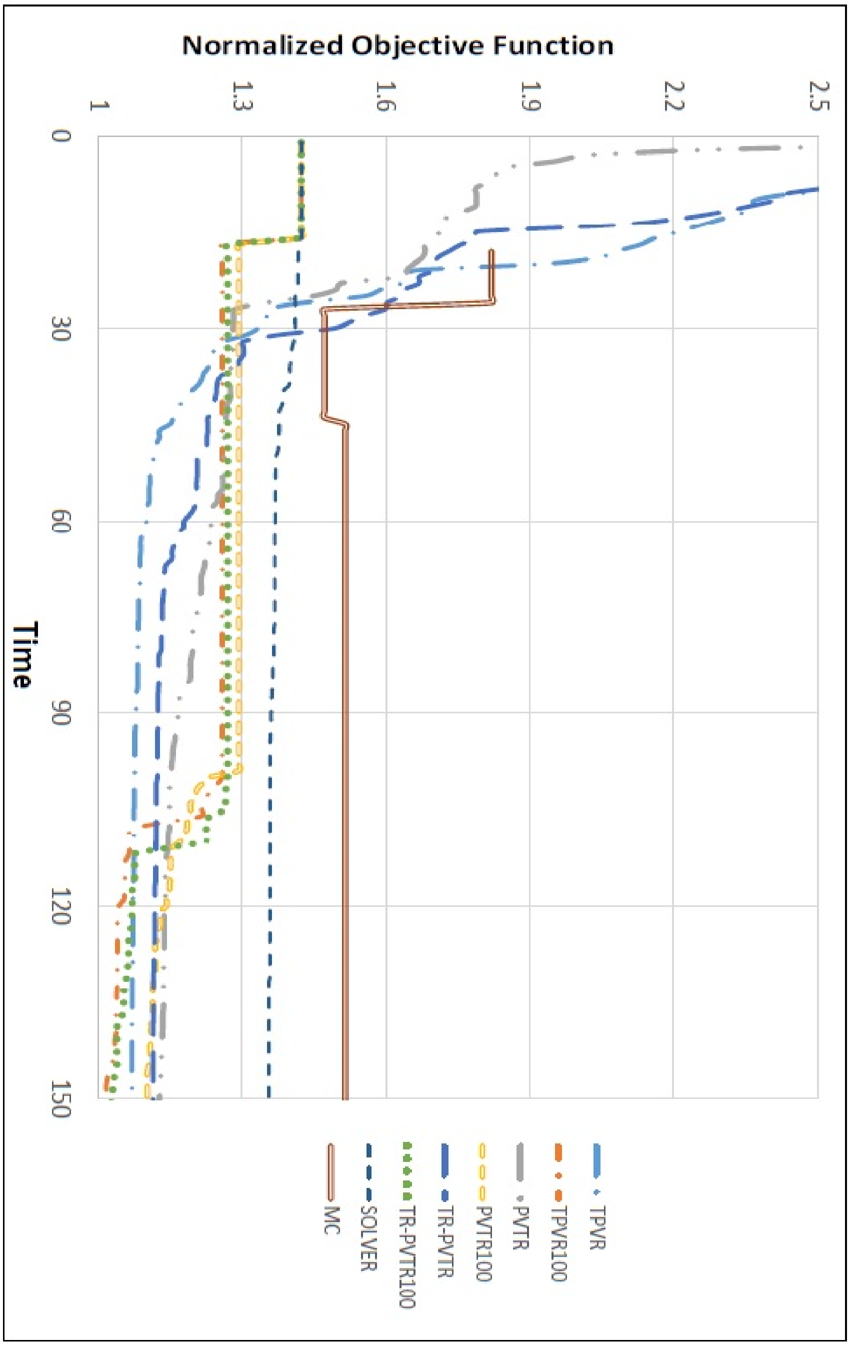


FIGURE 3.5: The zoomed part of Figure 3.4

Chapter 4

On the online maintenance scheduling problem

4.1 Introduction

Online algorithms have received considerable research interest during the past 15 years. An online algorithm receives a sequence of requests and performs an immediate action in response to each request. They arise in any situation where decisions must be made and resources allocated without knowledge of the future. The performance of online algorithms is usually evaluated using competitive analysis. Online algorithms are contrasted with offline algorithms, which receive the entire sequence of requests in advance. An offline algorithm takes an action in response to each request, but the choice of each action is based on the entire sequence of requests. In other words, an offline algorithm knows the entire sequence of requests beforehand and chooses its actions optimally.

It is obvious that the ignorance of the future is a great disadvantage in many systems. Thus, online algorithms will often perform much worse than their offline counterparts. In analyzing online algorithms, we usually, do not really care about the running time. The source of hardness here is lack of information, not complexity assumptions like $P \neq NP$. The effectiveness of an online algorithm is measured by using the concept of competitiveness, defined as the ratio between the objective value computed by the online algorithm and the objective value obtained by an optimal offline algorithm. An optimal online algorithm is then the one whose competitive ratio is closest to one. An online algorithm is c-competitive if for each input instance the objective value of the "schedule" produced by the algorithm is at most c times larger than the optimal objective value.

One important (and exclusive) distinction in online algorithms is, whether the algorithm is randomized or deterministic. A randomized online algorithm is an algorithm that uses random numbers to influence the choices it makes in the course of its computation. Thus, its behavior (typically quantified as running time or quality of output) varies from one execution to another even with a fixed input. However, deterministic online algorithms produce on a given input the same results following the same computation steps. Randomized algorithms are often conceptually very easy to implement. At the same time they are in run time often superior to their deterministic counterparts.

Online algorithms have been studied in many application areas including resource management in operating systems, data structuring, scheduling, networks, and computational finance. One famous example for an online problem is the ski rental problem. A skier must decide every day he goes skiing, whether to rent or to buy skis, unless or until he decides to buy them or stop skiing. The skier does not know how many days he can ski, because the weather is unpredictable. Call the number of days he will ski d. The cost to rent skis is 1 unit, while the cost of buying skis is b. The best possible deterministic online algorithm is to rent skis for $b - 1$ days and buy them if he/she wants to ski a day b. This algorithm is 2-competitive (for large b). However, there exists a randomized algorithm that achieves a competitive ratio of 1.58 (by randomly buying skis earlier).

4.2 Motivation and Problem Statement

The problem that we consider here is the scheduling of a condition-based maintenance for a machine or a server. In many industrial processes, a machine has to undergo a maintenance at the latest after a fixed number of running hours or production amount for example. This can be a contractual obligation towards the manufacturer for a warranty to stay valid, or a general rule to guarantee a high enough quality of the process. When the machine is in maintenance, the service it provides is interrupted. Therefore, it is better if the demand for the service provided by the machine is constant over time, to have as few maintenances as possible, and to only perform maintenance when absolutely required. However, when the demand is time-varying, it could be profitable to anticipate the timing of the maintenance to schedule it during a period of low demand.

In the problem we consider here, the demand is uncertain and not known in advance. Therefore it is not obvious if and why anticipating the maintenance could be profitable. Our goal is to define the simplest model possible that captures the problem we just described. Therefore we also assume that the productivity of the machine is very high and it can serve requests instantaneously. We also assume that there is no inventory, and during the period when the machine is undergoing maintenance, demand is simply lost.

Uncertainty is often a key aspect of operations research problems. It naturally arises in areas like finance, manufacturing, supply chain and transportation. For problems involving decisions over time, uncertainty often arises from the sequential nature of the data. Examples of this phenomenon include: the periodic control of inventory in retailers, when facing uncertain demands for their products; the changes in portfolio investment, when facing an uncertain market trend; and the allocation of monthly household income, when facing time-varying possibilities for investment or consumption. In all these problems, decisions are taken as some sequential data (demand, market value and investment/consumption opportunities, respectively) is being revealed.

To address uncertainty, operations research has used several models (e.g. stochastic, robust and recourse models Ben-Tal and Nemirovski [2002], Shapiro et al. [2009]) and methods (e.g. sensitivity analysis and statistical methods Bertsimas and Tsitsiklis [1997], Shapiro et al. [2009]). In this work we rather use the on-line paradigm and the associated competitive analysis (see below for formal definitions). This approach is especially suited when uncertainty is very high and the sequential nature of the process under consideration is key.

One particular approach to optimization under uncertainty shares several characteristics with competitive analysis. *Robust optimization* [Kouvelis and Yu, 1997, Ben-Tal and Nemirovski, 2002] addresses the general problem of optimizing a function over some domain under the assumption that the parameters defining the function and/or the domain fall in a certain set "around" the nominal value. The objective is to find a solution that is close to optimal for any choice of the parameters in this set.

Multi-stage robust optimization captures many features of on-line algorithms, sharing with competitive analysis the absence of any distributional assumption, the better tractability with respect to their stochastic counterpart and the production of very conservative solutions.

However, in robust optimization the aim is to minimize the objective function in absolute terms, while in competitive analysis the minimization is taken relative to the off-line optimum. This makes both approaches quantitatively incomparable.

4.2.1 Competitiveness and randomization

Formally, an on-line problem for an optimization problem is defined by a set of requests R, a set of actions A, and a set of cost functions $c_t : R^t \times A^t \to \mathbb{R}_0^+ \cup \{\infty\}$ for each period $t \in \mathbb{N}$. An algorithm $\mathcal{A}$ for this problem receives a sequence of requests $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ and replies with a sequence of actions $\mathcal{A}(\mathbf{r}) = (a_1, a_2, \ldots a_n)$, incurring in a cost $c_n(\mathbf{r}, \mathcal{A}(\mathbf{r}))$ that is to be minimized or maximized. However, if $\mathcal{A}$ is an *on-line* algorithm, the actions a_t must not depend on future requests or actions $\{r_{t'}, a_{t'}\}$ with $t' > t$.

We say that an on-line algorithm $\mathcal{A}$ for a minimization problem is c-competitive if it always produces solutions whose cost is not larger than c times the off-line optimum, this is:

$$\frac{c_n(\mathbf{r}, \mathcal{A}(\mathbf{r}))}{\min_{\mathbf{a}} c_n(\mathbf{r}, \mathbf{a})} \leq c, \qquad \text{for all } \mathbf{r}.$$

Competitiveness can be seen as a two-player game between the online algorithm (who chooses actions) and an adversary (who chooses requests). They play alternating actions and requests, the algorithm trying to minimize the competitive ratio for the sequence of requests in play, the adversary trying to maximize it. The adversary plays optimally at the instance achieving the minimum competitive ratio possible for the algorithm. For maximization problems, an analogous definition exists. A c-competitive algorithm for a maximization problem is one that always produces solutions whose cost is at least c times the off-line optimum. Note that $c \geq 1$ for maximization problems, while $c \leq 1$ for minimization problems.

4.2.2 Competitiveness and operations research

Online algorithms have been applied to combinatorial optimization problems [Ascheuer et al., 1999], with potential applications to the operations research field. For example, online versions of bin-packing [Seiden, 2002], facility location [Meyerson, 2001, Fotakis, 2008] and maximum matching [Karp et al., 1990, Feldman et al., 2009] and maximum weighted independent set in matroids Babaioff et al. [2007] have been studied in the competitive setting. Traditional scheduling is another area where competitive analysis is widely used [Albers, 1999, Imreh, 2009, Chekuri et al., 2001].

However, outside any of these two research areas, competitive analysis has seen little applications. Ball and Queyranne [2009] consider the reservation of seats in airplanes. Several authors have considered lot-sizing during the last 30 years [Bitran et al., 1984, Vachani, 1992, Van den Heuvel and Wagelmans, 2010]. To the best of our knowledge, this work is the first to apply competitive analysis to maintenance scheduling.

4.3 On-line maintenance scheduling

In maintenance scheduling, the uncertainty usually comes from the (unknown) time to next failure. In this case it is fairly realistic to assume that past behavior of similar machines provide good estimates for building a stochastic model. In the maintenance scheduling problem we introduce here, the uncertainty as in the problem discussed before, is about the arrival of customers or demand.

4.3.1 The on-line maintenance scheduling problem

A machine faces a sequence of service requests at times $\{t_i\}_{i=1..n}$, sorted in increasing order. At any time, the machine can be either *operational* or under *maintenance* status. When a service request arrives, the machine serves it instantaneously if it is operational, otherwise the request is discarded. The machine can serve at most L requests in a row before needing a maintenance. The maintenance resets this counter, but it leaves the machine in maintenance status for the T units of time after it starts (we could assume $T = 1$ without loss of generality, but we keep T for notational clarity). The objective of the problem is to define a maintenance scheduling plan that maximizes the number of service requests processed by the machine. We consider an on-line model where a maintenance decision at time t must be taken without any knowledge about future service requests $\{t_i : t_i > t\}$.

Note first that the off-line problem is easily solved in polynomial time using dynamic programming as follows. Let $H(i)$ be the optimal solution value up to time t_i (including request i). The optimal solution value is then $H(n)$. Because the maintenance restores the machine to its initial state (regardless of its previous condition), the following recursion holds: $H(i) = \max_{0\leq j<i}(H(j) + \min(L, N(t_j + T, t_i)))$, where $N(t, t')$ denotes the number of requests in the interval $(t, t']$ and $H(0) = t_0 = 0$.

In the on-line setting, let us consider NAT the most natural deterministic algorithm that places a maintenance directly after L requests have been served.

Theorem 4.1. NAT *is $L/(2L-1) > 1/2$-competitive, and this bound is best possible for deterministic on-line algorithms.*

Proof. We first prove the lower bound. Let $\{t_i\}_{i=1..n}$ be an arbitrary instance, and let $\{s_i\}_{i=0..m}$ be the maintenance finishing times placed by NAT , with the convention that $s_0 = 0$ and $s_{m+1} = \infty$.

We prove that the competitive ratio holds for each interval $(s_{i-1}, s_i]$, thereby proving the result. In the last window $(s_m, \infty]$, NAT serves all requests and the claim holds.

So let us fix $i \leq m$. In the interval $(s_{i-1}, s_i]$, NAT serves L requests including the last one at $s_i - T$, then misses some (say M) requests in $(s_i - T, s_i]$. If $M < L$, the claim holds. Otherwise, the number of requests in the interval

$[s_i - T, s_i]$ is $M + 1$ including one at $s_i - T$. But any algorithm can only satisfy L of these requests, since any maintenance initiated within that interval will finish after s_i. Therefore, in each interval $(s_{i-1}, s_i]$, the optimal solution serves at most $L - 1 + L$ requests and the result follows.

We can obtain an upper bound on the optimal competitive ratio for this problem by considering the behavior of an arbitrary deterministic algorithm against the following family of instances. Let us fix $T = 1$ and, for $k \in \{0, \ldots, L-1\}$, let J_k be the instance composed by $L - 1$ service requests, one at each of the times in $\{i/L : i = 1, \ldots, L\}$, followed by L service requests placed simultaneously at time $1 + (k - 0.5)/L$. In instance J_k, it is impossible to serve the $k + 1^{th}$ request and the L simultaneous ones because their arrival differ by $1 - 1/2L < 1$. Therefore the optimal solution of instance J_k serves $L + k$ requests.

We can make the following observations about these instances $\{J_k\}$ and an on-line algorithm that is optimal for them.

- All instances are indistinguishable until time 1.
- The algorithm will perform exactly one maintenance, and w.l.o.g. at times $\{i/L : i = 1, \ldots, L\}$.

The algorithm that places a maintenance at i/L for $i = 1, \ldots, L$ achieves a competitive ratio of $i/(L+i-1)$ on J_{i-1}. Therefore a best possible deterministic algorithm for this family of instances is to place a maintenance at time 1 and is $L/(2L-1)$-competitive. □

However, a better algorithm can be obtained using randomization.

4.3.1.0.1 Algorithm 1 : Let x_i be i.i.d. random variables with the following distribution. It takes value $j \in \{K, \ldots, L-1\}$ with probability $\frac{1}{L}$ and it takes value L with probability $\frac{K}{L}$. We iteratively place a maintenance each time we have served x_i requests (incrementing i after each maintenance).

Theorem 4.2. *Algorithm 1 is c-competitive for* $c = \frac{2KL+L(L-1)-K(K-1)}{2L(L+K-1)}$. *The best value for* K *is* $1-L+\sqrt{2L(L-1)}$ *rounded up or down, with* c *and* $1-\frac{K}{L}$ *converging to* $2-\sqrt{2} \approx 0.585$ *when* $L \to \infty$.

For the analysis of this algorithm, the following lemma is crucial:

Lemma 4.3. *There is an instance* $\{t'_i\}_{i=1..n}$ *achieving the worst competitive ratio for Algorithm 1 , such that some fixed optimal solution* OPT *serves all requests.*

Proof. Let $\{t_i\}_{i=1..n}$ be an arbitrary worst instance for Algorithm 1 , and OPT be a sequence of optimal maintenance times. Starting from $I \equiv \{t_i\}_{i=1..n}$, let us build a modified instance $I' \equiv \{t'_i\}_{i=1..n'}$, by deleting all the requests placed during maintenance times established by OPT. Clearly, OPT serves in I' as many requests as in I. On the other hand, for a fixed choice of the random numbers sampled by Algorithm 1 , the maintenances on I' are always shifted forward with respect to their corresponding maintenances in I, implying that the algorithm serves in I at least as many requests as in I'. It follows that I' is also a worst instance. $\square$

We remark that Lemma 4.3 only works for the worst case instances of Algorithm 1 (or similar algorithms differing only by the distributions from where x_i's are sampled).

Proof. **of Theorem 4.2** Consider an arbitrary worst-case instance for Algorithm 1 that satisfies the conditions described in Lemma 4.3. We can partially describe this instance by a sequence $\{a_i\}_{i=1..m}$, that captures the total number of service requests between consecutive maintenances of an optimum OPT. Note that OPT serves exactly $\sum_{i=1}^{m} a_i$ requests and also that, for every i the service requests associated to a_i and a_{i+1} are separated by at least T units of time. Note also that a_i is always bounded by L, otherwise the sequence cannot induce a worst-case instance. Let c be an arbitrary lower bound on the optimal competitive ratio of Algorithm 1 .

In the sequence $\{a_i\}_{i=1..m}$, suppose k and l are the largest indices so that $a_1+a_2+\ldots+a_{k-1} \leq K$ and $a_1+a_2+\ldots+a_{l-1} < L$. To make the exposition simpler, in the remaining of the proof we assume that $a_1+a_2+\ldots+a_{k-1} = K$ (this assumption can be easily removed). Let $A = \sum_{i \leq l} a_i$ and $Q = \sum_{i>l} a_i$. Algorithm 1 serves at least:

$$\frac{K}{L}(L + cQ) + \sum_{i=K}^{L-1} \frac{i}{L} + c\left(\frac{L-K}{L}Q + \sum_{k \leq i < j \leq l} \frac{a_i}{L} a_j\right) \tag{4.1}$$

requests. The first term corresponds to the event $x_1 = L$. The second term corresponds to the requests served in the first maintenance cycle when $x_1 < L$. The term $c\left(\frac{L-K}{L}Q + \sum_{k \leq i < j \leq l} \frac{a_i}{L} a_j\right)$ corresponds to the requests served in the remaining maintenance cycles when $x_1 < L$. More specifically, in the event that the first maintenance finishes before the block a_j, the algorithm will capture at least ca_j of that block in expectation. Then it suffices to observe that this event has probability $\sum_{k\leq i<j}\frac{a_i}{L}$. The bound critically uses the fact that requests corresponding to different blocks of requests a_i are separated by at least T units of time.

The worst-case instances will minimize $\sum_{k\leq i<j\leq l} a_i a_j$ subject to the constraint $\sum_{i\leq l} a_i = A$. Our claim is that the minimum of this expression is always achieved when the full set of requests in $\{a_i\}_{i=k..l}$ (with a total of $A-(k-1)$ requests) is concentrated in the blocks a_i with highest index possible. More precisely, the minimum is attained when a_l is maximized, a_{l-1} is maximized subject to the choice for a_l and so on. Since a_l is upper bounded by L, it follows that $a_l = L$ and $a_{l-1} = (A-(K-1)-L)^+ = (A-L-K+1)^+ \geq (A-L-K+1)$.

Plugging the previous bound in (4.1), we conclude that Algorithm 1 serves at least:

$$K + \frac{cKQ}{L} + \frac{L(L-1)-K(K-1)}{2L} + cQ - \frac{cKQ}{L} + c\frac{A-L-K+1}{L}L =$$

$$c(Q+A) + \frac{2KL + L(L-1) - K(K-1)}{2L} - c(L+K-1)$$

requests as a function of A. Choosing $c = \frac{2KL+L(L-1)-K(K-1)}{2L(L+K-1)}$ guarantees that Algorithm 1 serves at least $c(Q+A)$ requests. The maximum of c as a function of K, is achieved at the positive root of $-K^2+2(1-L)K+L^2-1$ which is equal to $1-L+\sqrt{2L(L-1)}$. $\square$

4.3.1.0.2 Upper bounds We can obtain an upper bound on the optimal competitive ratio for this problem by considering the same family J_k as above. Again, these instances are indistinguishable until time 1 and an optimal randomized algorithm will place exactly one maintenance at time i/L for some $i = 1, ..., L$. Therefore such an algorithm can be characterized by probabilities $\{p_i\}_{i=1..L}$ that the algorithm places its maintenance at time i/L. From the algorithm performance on J_k, it follows that

$$\max f : f \leq \frac{\sum_{i=1}^{L} ip_i + L\sum_{i=1}^{k} p_i}{L+k} \text{ for all } 0 \leq k \leq L-1, \sum_{i=1}^{L} p_i = 1, p_i \geq 0$$

is an upper bound in the competitive ratio achievable by any on-line algorithm for this problem. The dual of this linear program is

$$\min z : z \geq \sum_{k=0}^{L-1} iy_k + \sum_{k=i}^{L-1} Ly_k \text{ for all } 1 \leq i \leq L, \sum_{k=0}^{L-1} (L+k)y_k \geq 1, y_i \geq 0.$$

Noting that $\sum_{k=0}^{L-1} i + \sum_{k=i}^{L-1} L = L^2$ and $\sum_{k=0}^{L-1}(L+k) = (3L^2 - L)/2$, it is easy to check that the solution

$$\left(y_k = y \equiv \left(\frac{3L^2 - L}{2}\right)^{-1} \text{ for all } 1 \leq k \leq L, z = L^2 y \right)$$

is feasible for the dual problem, implying that $z = z(L)$ is an upper bound in the competitive ratio. Therefore we obtain:

Theorem 4.4. *No randomized algorithm for the on-line maintenance scheduling problem can be f-competitive, for $f > \frac{2L}{3L-1}$.*

Note that the optimal primal solution of the above LP is $p_L = \frac{L+1}{3L-1}$ and $p_i = \frac{2}{3L-1}$ for $i < L$. This suggests the following algorithm that we will test in the following section.

4.3.1.0.3 Algorithm 2: Let x_i be i.i.d. random variables with the following distribution. It takes value $j \in \{1, \ldots, L-1\}$ with probability $\frac{2}{3L-1}$ and it takes value L with probability $\frac{L+1}{3L-1}$. We iteratively place a maintenance each time we have served x_i requests (incrementing i after each maintenance).

4.3.1.0.4 Discussion about the case $L = 2$. We now make a few observations for the problem with $L = 2$, trying to improve the lower bound of 0.75 and maybe match the upper bound of 0.8. We initially restrict our attention to instances of the type of Lemma 4.3 (batches of 1 or 2 requests separated by T units of time).

The upper bounds obtained is based on the instances 1 and $1-2$. One possible algorithm would iteratively proceed as follows: with probability p serve two requests then maintain, and with probability $(1-p)$ serve one request, maintain, then serve two requests, then maintain. For $p = 0.6$, this achieves the desired competitive ratio of 0.8 for these two instances. However, it only achieves $(0.6 * (2 + 2.4) + 0.4 * 3)/5 = 0.768$ for the instance $2-1-2$. Optimizing for p against these 3 instances, one obtains $p = 2/3$ and a competitive ratio of $7/9 \approx 0.777$. It can easily be checked by hand that this competitive ratio is achieved for all instances with maximum three batches of size 1 or 2. Since Lemma 4.3 still applies for this algorithm, and one iteration of the algorithm can only cover maximum three batches, an inductive argument easily proves the result that this algorithm yields a competitive ratio of $7/9$ for $L = 2$. Obviously, one could try to generalize this idea and hope to improve the result further.

Another idea is embodied in the following algorithm: with probability 0.4, perform action A1, and with probability 0.6 perform action A2, then iterate (each random decision being independent). Action A1 is the following sequence: serve one request, then place a maintenance. If by the time the next request arrives, you observe that you could not have delayed the maintenance to serve one more request and maintain before serving the current one, then serve two requests and place a (second) maintenance (and iterate). In the other case, do nothing (and iterate). Action A2 is to serve two requests then place a maintenance (and iterate).

We could not find any instance for which this last algorithm achieves a competitive ratio worse than 0.8. However, Lemma 4.3 does not apply anymore because the number of requests served between two maintenances depends not only on the randomization, but also on the instance itself. That makes the proof of a competitive ratio harder.

Table 1 summarizes the bounds obtained with randomized algorithms.

4.4 Computational Experiments

The results of the previous sections shed some light on the behavior of two on-line algorithms compared to the optimal solution in the worst case. Of course, in practice, we might not only be interested in the worst-case but also in the other (better) cases. More precisely, we might want to know whether,

TABLE 4.1: Upper and lower bounds obtained on the competitiveness for the on-line maintenance scheduling problem.

L	LB	UB	L	LB	UB
1	1	1,000	11	0,606	0,688
2	0,777	0,800	12	0,604	0,686
3	0,667	0,750	13	0,603	0,684
4	0,65	0,727	14	0,602	0,683
5	0,633	0,714	15	0,6	0,682
6	0,625	0,706	16	0,599	0,681
7	0,619	0,700	17	0,598	0,680
8	0,614	0,696	18	0,598	0,679
9	0,611	0,692	19	0,597	0,679
10	0,608	0,690	∞	0,585	0,667

for certain classes of instances, we should expect the worst case or something substantially better.

In this analysis, we will look at the following instance classes. We also justify each particular choice. We normalize $T = 1$.

- POI(λ): in this class, the requests arrive following a Poisson process with parameter λ. This is the standard assumption when the request arrivals are independent of each other. We expect these instances to be fairly easy for on-line algorithms in general.

- T-UNI(α): in this class, batches of requests arrive, spaced by $T = 1$ units of time. Each batch is composed of $U[\alpha, L]$ requests spaced very close to each other in time (ϵ units of time). The class is specified by the value of α. This type of instances constitute the worst case instances in the analysis of Theorem 4.2. Therefore we expect that this type of instances will be favorable to Algorithm 1 .

- POI(λ)-UNI(α): in this class, batches of requests arrive following a Poisson process. Each batch is composed of $U[\alpha, L]$ requests spaced very close to each other in time (ϵ units of time). The class is specified by the values of α and λ. This is an intermediate class between the two first. It is also a realistic model for request arrivals.

The online algorithms we want to compare are the (deterministic) Algorithm NAT (ALG3), and the two randomized Algorithm 1 (ALG1) and 2 (ALG2). The results are given in Table 4.2. For each class we generated 100 instances of 1000 requests. We then ran each algorithm once, and computed the optimal solution as well. We report the average competitive ratio for each class and each algorithm. From the results we can conclude the following:

- ALG2 performs worse than the other ones for all problem classes, including UNI. This might suggest that the upper bound result is loose, and that the probability distribution coming from the upper bound proof does not generalize to other instances than the specific ones used in the proof. So we are not going to comment on this algorithm further.

- ALG1 and AGL3 perform much better than their worst case guarantee (0.58 and 0.5 respectively) for most problem classes.

- The instances POI(λ) are fairly easy for all 3 algorithms (they all achieve 0.94 or higher). The deterministic algorithm 3 performs marginally better than the other two.

TABLE 4.2: The average performance measures for different distribution-parameter settings

Distribution	Performance Measures				
	I100			I20	
	ALG1	ALG2	ALG3	ALG1	ALG3
POI 10	0.942	0.938	**0.958**	0.922	0.945
POI 4	0.959	0.949	**0.965**	0.947	0.956
POI 1	0.989	0.985	**0.991**	0.984	0.985
POI 10-UNI[33,99]	0.806	0.649	**0.914**	0.56	0.646
POI 4-UNI[33,99]	0.779	0.653	**0.876**	0.605	0.737
POI 1-UNI[33,99]	0.740	0.645	**0.790**	0.626	0.671
POI 10-UNI[66,99]	0.782	0.618	**0.839**	0.562	0.579
POI 4-UNI[66,99]	0.806	0.644	**0.861**	0.639	0.681
POI 1-UNI[66,99]	0.749	0.639	**0.753**	0.626	0.638
POI 10-UNI[90,99]	0.772	0.688	**0.785**	0.512	0.553
POI 4-UNI[90,99]	0.811	0.619	**0.846**	0.602	0.620
POI 1-UNI[90,99]	**0.728**	0.607	0.630	0.584	0.461
POI 10-UNI[95,99]	**0.803**	0.678	0.888	0.594	0.583
POI 4-UNI[95,99]	0.802	0.662	**0.829**	0.641	0.633
POI 1-UNI[95,99]	**0.712**	0.609	0.689	0.586	0.542
UNI[33,99]	**0.654**	0.572	0.621	0.597	0.593
UNI[66,99]	**0.651**	0.582	0.611	0.604	0.600
UNI[90,99]	**0.614**	0.540	0.549	0.560	0.546
UNI[95,99]	**0.613**	0.539	0.528	0.552	0.524

- The instances T-UNI(α) are harder than POI(λ)-UNI(α), which are then harder than POI(λ) for all three algorithms.
- Algorithm 1 performs better than the other two algorithms for instances T-UNI(α). This is not too surprising since the parameter of Algorithm 1 is optimized for that kind of instances.
- Regarding instances POI-UNI, the results are mixed. Whenever the time between the arrival of batches is high ($\lambda = 1$), and the size of the batches

is high, ALG1 tends to outperform ALG3 in expectation. In the other cases, we get the opposite behavior.

Of course, these results are not too surprising. It only makes sense to anticipate a maintenance when there are periods with very low demand long enough to place a maintenance separating periods with lots of requests. Besides, these results give information about the performance in expectation, and it might be interesting to understand better the shape of the distributions.
Therefore, we have also computed the average of the 20th percentile (worst instances) for ALG1 and ALG3 for each class. We can see that indeed, as is suggested by the better competitive ratio, ALG1 looks better compared to ALG3 when looking at the worst cases. That is, ALG1 is more robust. A secondary observation is that when ALG1 performs better in expectation, it always performs substantially better in the worst instances (20th percentile). On the other hand, there are instance classes where ALG3 performs substantially better in expectation, while performing not much better in the worst instances. This suggests that for these problem classes, the deterministic algorithm ALG3 is especially performing well on the easy or favorable instances.
To analyze this in more details, we plot in Figures 4.1-4.3 the histograms for 3 instance classes (500 instances for each class), that are representative of the behavior we could observe more generally.
Figure 4.1 gives the empirical distribution for class POI1-UNI90, for which ALG1 performs well. It can be seen that the shape of the two distributions are similar, with ALG1 shifted to the right. The only qualitative difference is for the worst case instance: ALG3 has a substantial number of instances with

a competitive ratio between 0.5 and 0.6 while ALG1 has a much thinner tail. This is to be expected since ALG1 is 0.58 competitive, while ALG3 is 0.5 competitive.

FIGURE 4.1: The empirical distribution for class POI1-UNI90

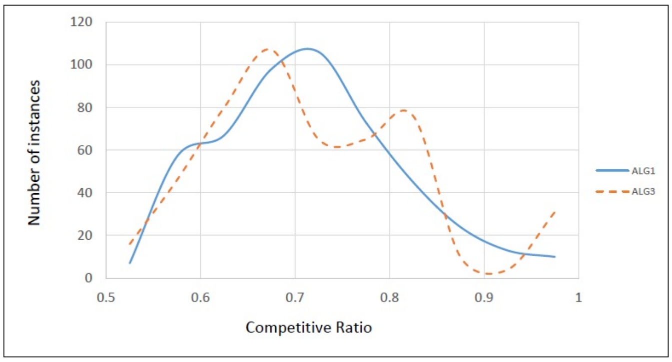


Figure 4.2 shows the empirical distribution for class POI1-UNI66. ALG1 and ALG3 have very similar distributions vanishing at 0.5 and 1.

FIGURE 4.2: The empirical distribution for class POI1-UNI66

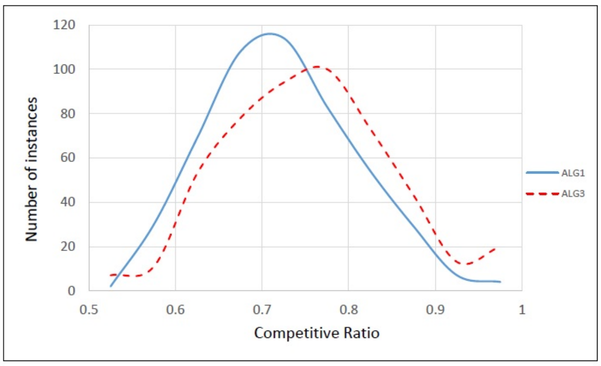
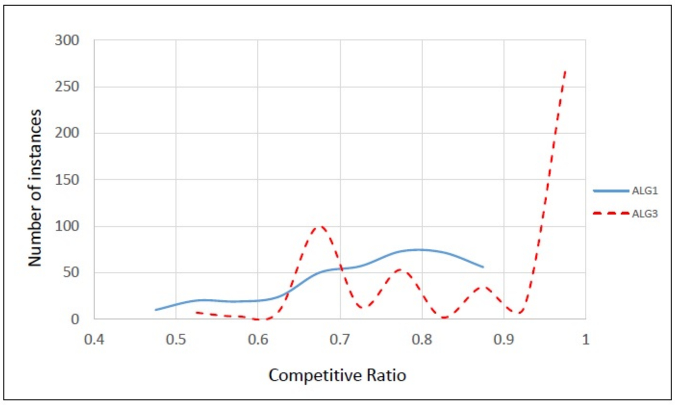


Figure 4.3 shows the empirical distribution for class POI4-UNI95. Here we see that ALG3 gets nearly the optimal solution in many cases, contrary to ALG1. In these instances, the time between the arrival of batches is smaller than T on average, so that it is not often that anticipating a maintenance can pay off. In other words, placing a maintenance will always mean losing many requests, so that serving L requests before placing a maintenance is often the (close to) optimal decision.

FIGURE 4.3: The empirical distribution for class POI4-UNI95



To conclude, we can say that the randomized ALG1 is more robust and performs in expectation better or nearly as well as the more natural algorithm ALG3, except for instances with the following characteristics: irregular arrivals with very high arrival rates periods separated with idle or very calm times, but where the duration of these calm periods are themselves irregular and frequently shorter than the duration of a maintenance. In these cases, anticipating the maintenance can be costly compared to the more natural deterministic algorithm.

Chapter 5

Conclusions

In this thesis we studied two types of problems to answer the question of ”what is the best time to do the maintenance?”. For the first problem addressed in Chapter 2 and 3, we considered an integrated production planning and maintenance scheduling in which the capacity of the machines decreases over time and there is enough information about the future demand. For the second problem addressed in part 2 (Chapter 4) of the thesis, we provided algorithms to obtain the best schedules for doing the maintenance actions when the demand is not known in advance. In this chapter we discuss theoretical and practical implications for each part of the thesis and future research directions.

5.1 Theoretical and Practical Implications

This thesis contributes to the scientific discussion by providing novel models and solution methods for the integrated production planning and maintenance scheduling problems. Our proposed approaches lead to better utilization of the production capacity and better allocation of maintenance scheduling, which consequently, vouch the lower total cost and higher service level by increasing the number of the satisfied requests.

Findings of Part 1

We have proposed several models for multi-item, single-machine capacitated lot sizing problem joint maintenance scheduling problem that has applications in many different areas. The motivation comes from a production system in Cement Industry, but the model can be used to represent many other aspects like the increase of defect products and the increase of downtime from corrective maintenance. We have described several models, mostly differing by the way capacity is impacted by production decisions. In the most general model, it is assumed that the capacity of the machine in each period is a function of the capacity of the previous period, lot size of the previous period and the presence of a setup in the previous period. These models are naturally formulated as the non-linear mixed integer optimization programs, then, some linearization techniques are proposed to to solve them them as the linear mixed-integer programs.

Our results show the value of using non-cyclic maintenance actions versus the

cyclic ones when the demand varies from one period to another. This advantage is considerable from both aspects of cost and solution time. While the existing solution methods in the literature enumerate all the possible schedules to solve the aforementioned problems (which is time consuming for large size problems), our proposed methodology solves the problem by only one implementation. In order to tighten the formulation and improve the quality of the solution (duality gap), the impact of adding some valid inequalities such as Wagner-Within inequalities and Multi-Commodity formulations for the large size instances of the problem is considerable.
Moreover, we have presented a Fix-and-Optimize heuristic to solve the multi-item, multi-machine multilevel CLSP joint CBM and transportation problem in which the capacity reduction follows the general formulation. The performance of the proposed algorithm is considered through different classes of instances close with different parameters. The results show that using a MIP solver suffices to obtain the optimal solution in 3 seconds, however, its efficiency for the large size instances with the parameters close to those of the real problem in Cement Industry is questioned. The solution approach based on Fix-and-Optimize consistently yield high-quality solutions in moderate time, and is superior compared to truncated MIP solver for realistic amount of time.

Findings of Part 2

Our contributions are summarized as follows. We presented a new, very simple on-line problem that captures an essential aspect of maintenance scheduling under uncertainty: how to avoid placing a maintenance when the machine is

most needed. Despite its simplicity, we have not been able to fully characterize optimum on-line algorithms. We provide a best possible 0.5-competitive deterministic algorithm, and give lower and upper bounds of 0.585 and 0.667 (in the limit) on the competitive ratio of randomized algorithms. The gap we obtain leaves room for improvement. In our view, improving either the lower or the upper bound will need to specifically consider algorithms that take into account what has happened before the last maintenance.
We also empirically tested the performance of the on-line algorithms for several specific request distributions. These demonstrated that the randomized algorithm tailored to behave well in the worst cases, also performs surprisingly well in absolute and in relative terms for most problems classes.

5.2 Future Prospects

Many future research perspectives exist for integrated capacitated lot sizing problems and maintenance scheduling when the capacity decreases over time. This section discusses some of the potential research categorized into three streams. First, we discuss challenges related to integrated CLSP and maintenance scheduling with stochastic parameters. Then, we discuss about the difficulty of adding another assumption about the capacity reduction in real problems. Finally, we discuss potential directions related to the improvements of the solution methods for such a problem.

- *Stochastic parameters:* decision makers may face uncertainty in the information when they organize the activities at a company. In Lot Sizing problems, uncertainty pertains to demand, lead time, capacity, etc. Hence, when a company faces uncertainty in its production/inventory decisions it might lead to an increase in the total costs, reduction in the service level and quality of the products, and necessity of resorting to safety stocks. Approaches dealing with lot sizing problems over a discrete planning horizon with stochastic demand are rare. In this realm, some approaches have been developed for the uncapacitated version of the problem. We believe there is a strong industrial need to develop approaches for solving integrated lot sizing and maintenance scheduling problems where the production capacity decreases, and the company faces stochastic demands over a discrete time horizon.

- *More complex capacity function:* the general capacity function considered in Chapter 3 assumes that the capacity of the machine in each period is a function of the capacity and the lot sizes of the previous period. In lot sizing models, it is necessary to accurately model capacity utilization to obtain feasible production plans. This often requires one to model the capacity consumed when a machine starts a production batch, or when a machine switches from one product to another. In the application of the aforementioned Cement Industry in Chapter 3, the presence of a setup in the previous period in not considered. Indeed, a start-up has to be incurred if production of some item i occurs in period t, but not in period t-1. Future work will address the case of adding the binary start-up variable to the general capacity function, which makes the MIML-CLSP formulation much more relevant for the industrial application and also more complex.

- *Solution methods:* despite the success of our heuristic method compared to exact methods, a solution method still needs to be devised for the generic integrated lot sizing and maintenance problem. Moreover, working on the better initial feasible solution to implement the proposed Fix-and-Optimize heuristic can result in better implementation of the proposed F&O heuristic.

Bibliography

E.H. Aghezzaf and N. Najid. Integrated production planning and preventive maintenance in deteriorating production systems. *European Journal of Operational Research*, 178(17):3382–3392, 2008.

E.H. Aghezzaf, M.A. Jamali, and D. Ait-Kadi. An integrated production and preventive maintenance planning model. *European Journal of Operational Research*, 181(2):679–685, 2007.

S. Albers. Better bounds for online scheduling. *SIAM Journal on Computing*, 29(2):459–473, 1999.

N. Ascheuer, M. Grötschel, S.O. Krumke, and J. Rambau. Combinatorial online optimization. In *Operations Research Proceedings 1998*, pages 21–37. Springer, 1999.

J. Ashayeri, A. Teelen, and W. Selen. A production and maintenance planning model for the process industry. *International Journal of Production Research*, 34(12):3311–3326, 1996.

M. Babaioff, N. Immorlica, and R. Kleinberg. Matroids, secretary problems, and online mechanisms. In *Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms*, pages 434–443. Society for Industrial and Applied Mathematics, 2007.

H.C. Bahl, L.P. Ritzman, and J.N.D. Gupta. Determining lot sizes and resource requirements: a review. *Operations Research*, 35(3):329–345, 1987.

M.O. Ball and M. Queyranne. Toward robust revenue management: Competitive analysis of online booking. *Operations Research*, 57(4):950–963, 2009.

G. Belvaux and L.A. Wolsey. Modelling practical lot sizing problems as mixed-integer programs. *Management Science*, 47(7):993–1007, 2001.

M. Ben-Daya. The economic production lot-sizing problem with imperfect production processes and imperfect maintenance. *International Journal of Production Economics*, 76(3):257–264, 2002.

A. Ben-Tal and A. Nemirovski. Robust optimization-methodology and applications. *Mathematical Programming*, 92(3):453–480, 2002.

M. Berg, M.J.M. Posner, and H. Zhao. Production inventory system with unreliable machines. *Operations Research*, 42(1):111–118, 1994.

D. Bertsimas and J.N. Tsitsiklis. *Introduction to linear optimization*. Athena Scientific Belmont, MA, 1997.

G.R. Bitran, T.L. Magnanti, and H. H. Yanasse. Approximation methods for the uncapacitated dynamic lot size problem. *Management Science*, 30(9): 1121–1140, 1984.

N. Brahimi, S. Dauzere-Peresb, and L.A. Wolsey. Polyhedral and lagrangian approaches for lot sizing with production time windows and setup times. *Computers and Operations Research*, 37(1):182–188, 2010.

L. Buschkuhl, F. Sahling, S. Helber, and H. Tempelmeier. Dynamic capacitated lot sizing problems: a classication and review of solution approaches. *OR Spectrum*, 32(2):231–261, 2010.

C.R. Cassady and E. Kutangolu. Integrating preventive maintenance planning and production scheduling for a single machine. *IEEE Transaction on Reliability*, 54(2):304–309, 2005.

T. Chakraborty, B.C. Giri, and K.S. Chaudhuri. Production lot sizing with process deterioration and machine breakdown. *European Journal of Operational Research*, 185(2):606–618, 2008.

T. Chakraborty, B.C. Giri, and K.S. Chaudhuri. Production lot sizing with process deterioration and machine breakdown under inspection schedule. *Omega*, 37(2):257–271, 2009.

C. Chekuri, R. Motwani, B. Natarajan, and C. Stein. Approximation techniques for average completion time scheduling. *SIAM Journal on Computing*, 31(1):146–166, 2001.

W.H. Chen and J.M. Thizy. Analysis of relaxations for the multi-item capacitated lot-sizing problem. *Annals of Operations Research*, 26(1):29–72, 1990.

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

G.D. Eppen and R.K. Martin. Solving multi-item capacitated lot-sizing problems using variable re-definition. *Operation Research*, 35(6):832–848, 1987.

J. Feldman, A. Mehta, V. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *50th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126. IEEE, 2009.

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

D. Fotakis. On the competitive ratio for online facility location. *Algorithmica*, 50(1):1–57, 2008.

A. Garg and S.G. Deshmukh. Maintenance management: literature review and directions. *Journal of Quality in Maintenance Engineering*, 12(3):205–238, 2006.

A. Grigorieva, J. Van de Klundertb, and F.C.R. Spieksma. Modeling and solving the periodic maintenance problem. *European Journal of Operational Research*, 172(3):783–797, 2006.

H.A. Groenevelt, L. Pintelon, and A. Seidmann. Production lot sizing with machines breakdowns. *Management Science*, 38(1):104–123, 1992a.

H.A. Groenevelt, L. Pintelon, and A. Seidmann. Production batching with machine breakdown and safety stocks. *Operations Research*, 40(5):959–971, 1992b.

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

K.S. Hindi, K. Fleszar, and C. Charalambous. An effective heuristic for the clsp with set-up times. *Journal of Operation research Society*, 54(5):490–498, 2003.

J. Hu, L. Zhang, and W. Liang. Opportunistic predictive maintenance for complex multi-component systems based on dbn-hazop model. *Process Safety and Environmental Protection*, 90(5):376–388, 2012.

Y.F. Hung, C.P. Chen, C.C. Shih, and M.H. Hung. Using tabu search with ranking candidate list to solve production planning problems with setups. *Computers and Industrial Engineering*, 45(4):615–634, 2003.

Cs. Imreh. Online scheduling with general machine cost functions. *Discrete Applied Mathematics*, 157(9):2070–2077, 2009.

S.M.R. Iravani and I. Duenyas. Integrated maintenance and production control of a deteriorating production system. *IIE Transactions*, 34(5):423–435, 2002.

R.M. Karp, U.V. Vazirani, and V.V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22th annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.

C.H. Kim, Y. Hong, and S.Y. Kim. An extended optimal lot sizing model with an unreliable machine. *Production Planning and Control*, 8(6):577–585, 1997.

P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*, volume 14. Springer, 1997.

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

C.Y. Lee and Z.L. Chen. Scheduling jobs and maintenance activities on parallel machines. *Naval Research Logistics*, 47(2):145–165, 2000.

H.L. Lee and M.J. Rosenblatt. A production and maintenance planning model with restoration cost dependent on detection delay. *IIE Transactions*, 21(4): 368–375, 1989.

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

G.C. Lin and D.C. Gong. On a production-inventory system of deteriorating items subject to random machine breakdowns with a fixed repair time. *Mathematical and Computer Modelling*, 43(7-8):920–932, 2006.

Y.H. Lin, J.M. Chen, and Y.C. Chen. The impact of inspection errors, imperfect maintenance and minimal repairs on an imperfect production system. *Mathematical and Computer Modeling*, 53(9-10):1680–1691, 2011.

A.S. Manne. Programming of economic lot sizes. *Management Science*, 4(2): 115–135, 1958.

R.A. Melo and L.A. Wolsey. Mip formulations and heuristics for two-level production-transportation problems. *Computers and Operations Research*, 39(11):2776–2786, 2012.

A. Meyerson. Online facility location. In *Proceeding of 42th IEEE Symposium on Foundations of Computer Science*, pages 426–431. IEEE, 2001.

H.H. Millar and M. Yang. Lagrangian heuristics for the capacitated multi-item lot sizing problem with backordering. *International Journal of Production Economics*, 34(1):1–15, 1994.

A.J. Miller, G.L. Nemhauser, and M.W.P. Savelsbergh. On the polyhedral structure of a multi-item production planning model with set up times. *Mathematical Programming*, 94(2-3):375–405, 2003.

M.C.V. Nascimento, M.G.C. Resende, and F.M.B. Toledo. Grasp heuristic with path-relinking for the multi-plant capacitated lot sizing problem. *European Journal of Operational Research*, 200(3):747–754, 2010.

L. Ozdamar and M.A. Bozyel. The capacitated lot sizing problem with overtime decisions and setup times. *IIE Transactions*, 32(11):1043–1057, 2000.

Y. Pochet and L.A. Wolsey. Solving multi-item lot sizing problems using strong cutting planes. *Management Science*, 37(1):53–67, 1991.

Y. Pochet and L.A. Wolsey. *Production Planning by Mixed Integer Programming*. Springer Series in Operations Research and Financial Engineering, New York, US, 2006.

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

S.S. Seiden. On the online bin packing problem. *Journal of the ACM (JACM)*, 49(5):640–671, 2002.

A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming: modeling and theory*, volume 9. SIAM, 2009.

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

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

H. Tempelmeier and M. Derstroff. A lagrangean-based heuristic for dynamic multi-level multi-item constrained lotsizing with setup times. *Management Science*, 42(5):738–757, 1996.

W.W. Trigeiro, L.J. Thomas, and J.O. McClain. Capacitated lot sizing with setup times. *Management Science*, 35(3):353–366, 1989.

R. Vachani. Performance of heuristics for the uncapacitated lot-size problem. *Naval Research Logistics*, 39(6):801–813, 1992.

W. Van den Heuvel and Albert P.M. Wagelmans. Worst-case analysis for a general class of online lot-sizing heuristics. *Operations Research*, 58(1): 59–67, 2010.

H.M. Wagner and T.M. Whitin. Dynamic version of the economic lot-size model. *Management Science*, 5(1):89–96, 1958.

C.H. Wang and F.C. Meng. Optimal lot size and offline inspection policy. *Computers and Mathematics with applications*, 58(10):1921–1929, 2009.

X. Yao, E.G. Fernandez, M.C. Fu, and S.I. Marcus. Optimal preventive maintenance scheduling in semiconductor manufacturing. *IEEE Transaction on Semiconductor Manufacturing*, 17(3):345–356, 2004.