Scheduling, inventory management and production planning: Formulations and solution methods

Pedram Hooshangitabrizi

A Thesis
in
the Department
of
Mechanical, Industrial, and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy (Industrial Engineering) at
Concordia University
Montreal, Quebec, Canada

May 2020

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Pedram Hooshangitabrizi**

Entitled: **Scheduling, inventory management and production planning: Formulations and solution methods**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Industrial Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Akshay Kumar Rathore	Chair
Dr. Raf Jans	External Examiner
Dr. Navneet Vidyarthi	External to Program
Dr. Masoumeh Kazemi Zanjani	Examiner
Dr. Daria Terekhov	Examiner
Dr. Ivan Contreras	Thesis Co-Supervisor
Dr. Nadia Bhuiyan	Thesis Co-Supervisor
Dr. Hossein Hashemi Doulabi	Thesis Co-Supervisor

Approved by Dr. Ivan Contreras

Graduate Program Director

Dr. Amir Asif

May 1, 2020 Dean of Gina Cody School of Engineering and Computer Science

Abstract

Scheduling, inventory management and production planning: Formulations and solution methods

Pedram Hooshangitabrizi, Ph.D.

Concordia University, 2020

This thesis presents formulations and solution methods for three types of problems in operations management that have received major attention in the last decade and arise in several applications. We focus on the use of mixed integer programming theory, robust optimization, and decomposition-based methods to solve each of these three problems.

We first study an online scheduling problem dealing with patients' multiple requests for chemotherapy treatments. We propose an adaptive and flexible scheduling procedure capable of handling both the dynamic uncertainty arising from appointment requests that appear on waiting lists in real time and capable of dealing with unexpected changes. The proposed scheduling procedure incorporates several circumstances prevalent at oncology clinics such as specific intervals between two consecutive appointments and specific time slots and chairs. Computational experiments show the proposed procedure achieves consistently better results for all considered objective functions compared to those of the scheduling system in use at the cancer centre of a major metropolitan hospital in Canada.

We next present an inventory management problem that integrates perishability, demand uncertainty, and order modification decisions. We formulate the problem as a two-stage robust integer optimization model and develop an exact column-and-row generation algorithm to solve it. Based on computational results, we show that considering order modification can significantly reduce the total cost. Moreover, comparing the results obtained by the proposed robust model to those obtained from the deterministic and stochastic variants, we note that their performances are similar

in the risk-neutral setting while solutions from the robust models are significantly superior in the risk-averse setting.

Finally, we study decomposition strategies for a class of production planning problems with multiple items, unlimited production capacity and, inventory bounds. Based on a new mixed integer programming formulation, we proposed a Lagrangian relaxation for the problem. We propose a deflected subgradient method and a stabilized column generation algorithm to solve the Lagrangian dual problem. Computational results confirm that the proposed formulation outperforms the previously proposed models and methods. Further analysis shows the impact of using decomposition techniques in providing tighter bounds.

To my family

Everything diminishes when it is used except knowledge- Imam Ali

Acknowledgements

I am truly grateful to the people in my life who contributed towards reaching this goal. I would like to thank Dr. Nadia Bhuiyan for providing the opportunity of investigating real world problems and for being always positive. My sincerest words of gratitude go to Dr. Ivan Contreras for guiding my research with his valuable ideas and for making me strong in many academic skills. I also thank Dr. Hoessin Hashemi Doulabi who has been my advisor and above all my friend. I would like to acknowledge the financial support provided by Mitacs. I am also thankful to Dr. Gerald Batist, Erin Cook, José, Samuel Suss, Marius Roland, and other staff members at the Segal Cancer Centre for their support and collaboration in defining the scheduling problem and providing the required clinical data.

I would like to thank my parents for giving me the gift of life and specially my mother who taught me the values of hard work and discipline. I am thankful to my dear father who despite not being with us is ever-present. I am greatly indebted to my wife, Maryam, for her endless love, support and encouragement through the tough days of the PhD program. Finally, I thank my beloved daughter, Bahar, whose smiles made this journey sweet for me.

Contribution of Authors

This dissertation is presented as a manuscript-based thesis. It contains three articles that have been accepted for publication or are under revision in different journals. The articles were submitted in the following chronological order. The first article titled "Improving patient-care services at an oncology clinic using a flexible and adaptive scheduling procedure" was published online in February 2020 in the journal *Expert Systems with Applications*. The second manuscript titled "Two-stage robust optimization for perishable inventory problems with order modification" was submitted in January 2020 to the journal *European Journal of Operational Research*. Finally, the third manuscript titled "Decomposition strategies for multi-item uncapacitated lot-sizing problems with inventory bounds" is planned to be submitted in March 2020.

The first manuscript is co-authored with Dr. Ivan Contreras and Dr. Nadia Bhuiyan, as supervisors, and Gerald Batist, as a medical oncologist, who established research guidelines and reviewed the papers before submission. The second manuscript is co-authored with Dr. Hossein Hashemi Doulabi, who joined our research team in July 2018 as the third supervisor, Dr. Ivan Contreras, and Dr. Nadia Bhuiyan. Finally, the third manuscript is co-authored with Dr. Ivan Contreras. The author of this thesis acted as the principal researcher with the corresponding duties such as collecting data, development of formulations and algorithms as well as programming of solution methods, analysis of computational results, and writing the first drafts of the articles.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Operations management practice and research have played a significant role in the development of large scale business ideas ranging from the Walmart supply chain management to the Toyota production system [67]. Addressing challenging and realistic problems in this area is thus a promising direction of research with the potential of leading to the development of theoretical and application breakthroughs. Operations management may include many concepts and applications such as production design and customer service, warehouse location analysis, management of service operation, inventory management, and scheduling operations strategies. This thesis addresses three main problems in operations management: scheduling in healthcare, inventory management for perishable products, and production planning with storage capacity.

In the first part of the thesis, we focus on a chemotherapy scheduling problem arising at the Segal Cancer Centre in Montreal, Canada. In outpatient clinics, timely access to health services is a very important factor for both patients and healthcare service providers. Efficiently designing a scheduling system can optimize clinical operations, increase patient's satisfaction, and reduce resource idle time and overall costs [101]. We address four practical aspects of appointment scheduling systems: dynamic uncertainty arising from appointment requests, requesting multiple appointments simultaneously, patient preferences, and handling last-minute changes. These aspects are modelled using integer programs and solved using a flexible and adaptive procedure. On the other hand, computational efficiency is addressed by solving large-scale instances using historical data provided by the cancer centre.

In the second part of the thesis, we study a perishable inventory management problem. There are many products such as food and blood products that deteriorate over time and become unfit for consumption after their shelf-life. In inventory management systems, it is thus crucial to make optimal ordering, production and allocation decisions while reducing waste. We focus on three main aspects of perishable inventory systems: shelf-life, demand uncertainty, and order modification. These are all formulated using a two-stage robust integer program with an uncertainty set under the assumption of periodic review system. An exact algorithm tailored to the proposed two-stage formulation is developed to solve the problem in the worst-case scenario. Extensive computational experiments are performed to draw managerial insights by analyzing the effect of considering different parameters and ordering policies.

Finally, in the last part of this thesis, we address a fundamental class of production planning problems that arises when multiple items with limited storage capacity are considered in lot-sizing problems. Limited storage capacity is considered because of several reasons: special physical structure of products, administrative policies, and warehouse infrastructure conditions [84]. It is important to find optimal production periods and amount of products stored in shared space to minimize the sum of setup, production and inventory costs. We provide an extended formulation for the multi-item uncapacitated lot-sizing problem with inventory bounds (MULSIB) based on a previously proposed formulation. Using the extended formulation, we are allowed to decompose the problem into smaller subproblems. Among the publications on MULSIBs, none is concerned with the development of exact decomposition-based algorithms for the MULSIB. We investigate different decomposition strategies for the MULSIB by decomposing it into smaller and relatively easy subproblems with the aim of finding tighter bounds.

The contributions of this thesis can be categorized as follows:

- Problem modelling.
  - Addressing critical and complex aspects of patient scheduling problems.
  - Addressing challenges in inventory replenishment planning using a two-stage mixed integer programming formation.
  - Introducing a way to enhance the decomposability of formulations.
- Algorithmic development.
  - A flexible and an adaptive procedure that combines various mixed integer programming models to solve offline and online scheduling problems
  - A column-and-row generation algorithm to solve a two-stage robust optimization model for perishable inventory management problems
  - Two decomposition-based algorithms to solve the Lagrangian dual problem associated with the MULSIB: a deflected subgradient method and a stabilized column generation algorithm.
- Managerial insights
  - Sensitivity analysis of the effect of different controllable parameters on the performance of scheduling systems
  - The effect of handling last-minute changes on the quality of schedules.
  - Analysis of the impact of incorporating order modifications on the performance of inventory systems.

– Evidence of the stable performance of solutions from robust counterparts in both risk-averse and risk-neutral settings.

This thesis consists of four more chapters, three of which correspond to manuscripts that have been published or submitted for revision in high impact journals in the area of management science and operations research. Chapter 2 first presents a literature review on related problems to outpatient scheduling problems. Afterwards, it describes a real-life problem associated with scheduling appointment requests for chemotherapy treatments. Using two different integer programming models, a flexible and adaptive procedure for handling appointment requests and unexpected events is proposed. A comparison of the schedules built by the proposed scheduling procedure to those actually used in reality is done to evaluate the proposed procedure and to obtain managerial insights.

Chapter 3 provides a formal definition of the perishable inventory management problem and proposes a two-stage robust formulation. It also presents a column-and-row generation algorithm to solve the formulated robust model. Computational experiments and sensitivity analyses are carried out to evaluate the performance of the algorithm and to analyze the effect of different aspects and parameters.

Chapter 4 studies a fundamental problem in production planning with slight modifications by considering multiple items and inventory bounds. A reformulation of the problem is proposed which allows us to apply decomposition-based techniques. Computational experiments show the proposed formulation provides better upper bounds than the state-of-the-art formulations and methods. Also, the proposed lower bounding method provides tighter bounds than other methods. Finally, Chapter 5 provides conclusions and future works presented in this thesis.

Chapter 2

Improving patient-care services at an oncology clinic using a flexible and adaptive scheduling procedure

P. Hooshangi-Tabrizi, I. Contreras, N. Bhuiyan, G. Batist. “Improving patient-care services at an oncology clinic using a flexible and adaptive scheduling procedure". published in *Expert Systems with Applications*, February 2020 [49].

Abstract

This paper studies an online scheduling problem dealing with patients' multiple requests for chemotherapy treatments at the cancer centre of a major metropolitan hospital in Canada. The proposed solution to the problem is an adaptive and flexible procedure that systematically combines two optimization models. The first model is intended to dynamically schedule incoming appointment requests, which arrive in the form of waiting lists, and the second model is used to reschedule already booked appointments with the goal of better allocating resources as new information becomes available. The performance and potential impact of the proposed procedure is as-

sessed using historical data provided by the cancer centre. Moreover, a sensitivity analysis is carried out to draw insights that may help hospital managers to deal more efficiently with both incoming requests and unexpected events.

2.1 Introduction

Health-care systems are continuously growing in complexity, size, and funding requirements. According to a World Bank report [106], Canada's total health expenditure was approximately 10.4% of GDP in 2014, which ranks Canada among the countries with the highest ratio of total health expenditure to GDP. In order to efficiently manage expenditures, health-care systems have started to use more sophisticated decision-making tools to better utilize valuable and shared resources. Moreover, since cancer is the leading cause of death in Canada and is responsible for 30% of all deaths [20], cancer centres face an increase in demand. Given the limited capacity and cost of inpatient care systems, which require a prolonged stay at the facility, as well as the technological advancements of outpatient clinics, currently there is a special emphasis on outpatient facilities [40]. Timely access to outpatient care is a very important factor for both patients and cancer clinics. Therefore, a well-designed scheduling system can optimize clinical operations, increase patient's satisfaction, and reduce resource idle time and overall costs [101]. Outpatient scheduling problems are common to a wide range of settings, including the scheduling of patients for general-practice physician appointments, chemotherapy, radiology, surgery, and hemodialysis.

This study focuses on a chemotherapy scheduling problem arising at the Segal Cancer Centre (SCC) in Montreal, Canada. The SCC provides chemotherapy-related services for those who suffer from cancer. Chemotherapy is a systemic treatment that uses various drugs to kill cancer cells. In outpatient chemotherapy clinics, there is often a sequence of stages to patient treatment. These include blood tests, consultation with an oncologist, drug preparation at the pharmacy, and infusions in the treatment area. At each stage, several resources (nurses, oncologists, phlebotomists, technicians, infusion chairs and beds, and examination rooms) must be shared among patients [99]. Currently at the SCC clerks manually schedule incoming appointment requests. Most requests arrive in the form of a waiting list that normally consists of a set of requests for a number of patients rather than a single appointment request for a single patient. Such requests are normally added to a waiting list after an oncologist consultation. In some cases, a patient may also call or come in person to book or reschedule appointments. Clerks currently book appointments from the waiting list one at a time, on a first come first served basis. The clerks continually deal with multiple patients, each requesting several appointments with different considerations and limitations. Thus, it is quite challenging and time-consuming to efficiently schedule all appointment requests of a list with the manual scheduling procedure currently in use. The difficulty increases whenever unexpected events or last-minute changes occur.

At the SCC, most appointment requests are for recurring patients. These patients are those who have already started receiving treatment and are no longer considered as new patients waiting to start their first treatment. The appointment requests are received in real time, and the total number of future requests is not known beforehand. Each appointment request is comprised of a target date, service type (type of drug and infusion duration), primary nurse (who follows that particular patient during treatment), preferred start time of appointments, and a clinical note that may state a specific start time or a special type of chair. The challenge faced by the SCC is to dynamically assign a day, start time, chair, and nurse to each appointment request that takes into account the existing partially filled schedules of the days in the planning horizon. The main objectives of the SCC are to minimize the number of appointments for which floating nurses (instead of primary nurses) are assigned to patients, to reduce the number of non-scheduled appointments and non-preferred appointment times, and to avoid overtime and nursing handovers. This study mainly deals with recurring patients, and so the focus is not on minimizing access time (the length of time between appointment request and first treatment) for new patients but rather on ensuring that once a treatment plan is prescribed, the target date of each treatment is respected to the extent possible. However, with the permission of the oncologist, an individual treatment plan can deviate slightly (by one or two days) to create a more efficient overall schedule.

Given the inherent complexity of appointment scheduling problems, most studies have considered simplified models with rather unrealistic assumptions, such as advance knowledge of all information regarding appointment requests, the processing of single appointments only, not assigning a nurse to a single patient during their treatment setup, and ignoring patient preferences. To the best of our knowledge, research on practical chemotherapy scheduling problems is still limited, and there have been few attempts to find optimal or near-optimal ways of scheduling and rescheduling patient appointments that take into account real aspects of the problem. The main contribution of this study is to propose and evaluate an adaptive and flexible scheduling procedure capable of handling both the dynamic uncertainty arising from appointment requests that appear on waiting lists in real time and also deals with unexpected events that may occur. The propose scheduling procedure incorporates several circumstances prevalent at the SCC. Specifically, it takes into account multiple, simultaneous appointment requests for a single patient that are accompanied by various oncologist-prescribed clinical requirements. These requirements may include specific intervals between two consecutive appointments (rest duration) and specific time slots and chairs. Patients receive notice of the dates of their appointments either at the time requested or after a number of appointment requests have been gathered into a waiting list in order to allocate available resources more efficiently. In both circumstances, patients are informed about their treatment start time a few days prior to the appointment date. The prescribed target date of treatments for a recurring patient can deviate by a controllable threshold parameter, determined by the patient's oncologist, to improve the quality of daily schedules. Each nurse can set up only one infusion at a time; however, a single nurse can monitor several patients simultaneously once the infusion setup phase is complete. Furthermore, a rescheduling model is integrated into the proposed scheduling procedure to accommodate last-minute changes and optimize resource allocation after appointment requests from waiting lists have been dealt with.

The remainder of the paper is organized as follows. Section 2.2 provides a literature review on related problems. Section 2.3 describes a real-life problem associated with scheduling appointment requests for chemotherapy treatments. The proposed flexible and adaptive procedure for handling appointment requests and unexpected events using two different integer programming (IP) models is detailed in Section 2.4. Section 2.5 presents the results of numerical experiments that evaluate the performance of the proposed procedure. The same section reports on sensitivity analyses that show the effects of assuming varying degrees of flexibility in the date and time of appointments on the performance of the system, and how the proposed adaptive procedure deals with unexpected events. Finally, Section 2.6 concludes the paper and proposes directions for future research.

2.2 Literature review

Chemotherapy scheduling problems are receiving increased attention in the literature, and several reviews of outpatient appointment and scheduling systems have been published, of which the most relevant are those of Cayirli and Veral [22], Gupta and Denton [40], Hulshof et al. [52], Lamé et al. [66], and Ahmadi-Javid et al. [2].

Sadki et al. [92] studied the scheduling of patients for chemotherapy treatments and oncologist consultations. The authors proposed a heuristic based on the Lagrangian relaxation method with the intention of minimizing the total weighted cost incurred by patient wait times and makespan (completion time of all treatments). In their study, oncologists and injection beds were considered as the major resources at the clinic but the constraints related to nurse availability were neglected. In the area of capacity planning for chemotherapy treatments, Saure et al. [95] and Gocgun and Puterman [36] studied the problem of multi-appointment scheduling, formulated it as a Markov decision process model, and proposed approximation schemes to solve instances of real size. The goal was to tailor available treatment capacity to incoming demand to specify the day of arriving requests. However, the model did not assign a time slot, nurse, or chair to the appointment request.

Turkcan et al. [102] developed IP models to formulate a chemotherapy scheduling problem that considered acuity levels of patients, aiming to achieve objectives such as minimizing treatment delays, reducing staff overtime, and maximizing staff utilization. The authors assumed each request was for a single appointment; however, in most oncology centres, requests for multiple appointments with rest periods between treatments are common. In this study, it is assumed that multiple appointments are requested.

In a research project comparable to the present study, Hahn-Goldberg et al. [42] developed an approach based on constraint programming to create a template for accommodating incoming patients in an online fashion. They proposed an algorithm to update the template if the infusion duration for a patient did not fit those of artificial appointments already scheduled in the template. Although the researchers considered the available capacity of resources such as nurses and chairs, in contrast to the present study they did not specifically determine which nurse would be on duty and monitor the patient and which chair would be assigned. Such information is essential when a partially filled appointment schedule is in use and it is being updated frequently to achieve optimal resource allocation. Moreover, they assumed that only one treatment would be scheduled for each incoming patient. In a similar study, Condotta and Shakhlevich [24] proposed creating multi-level templates to accommodate patient requests for chemotherapy appointments. Their goal was to minimize wait times and balance nurse workloads. Using integer linear programming models and data generated from artificial patients, they determined the day, time slot, and nurse of each patient; however, the template did not consider patients preferences for appointment start times.

Liang et al. [71] and Liu et al. [72] used simulation models to analyze the scheduling, staffing, and flow process stages inside an oncology clinic and to identify bottlenecks where improvements could be made. In particular, Liang et al. [71] proposed a mathematical model that would create a balanced schedule for both chemotherapy treatments and consultations for a single day but not for a planning horizon. Liang and Turkcan [70] addressed the daily scheduling of patients, taking into account nurse assignments in two different modes: functional and primary nursing care models. They assumed different acuity levels for patients and different skill levels for nurses; however, they did not consider chair capacity or the assignment of patients to specific chairs or rooms when creating schedules. They also proposed multi-objective optimization models with the objective of minimizing patient wait times, total nurse overtime, and excessive workloads to solve the problem.

Castaing et al. [21] proposed a two-stage stochastic programming problem for scheduling patients in an outpatient cancer centre on a daily basis (not over a planning horizon). They considered uncertainty in the duration of appointments and developed a heuristic to solve their formulated problem in a reasonable amount of time with a sufficient number of scenarios. Alvarado and Ntaimo [7] used mean-risk stochastic IP models to formulate the problem of scheduling chemotherapy appointments. They considered uncertainty in the acuity levels of patients, availability of nurses, and treatment duration. They also assumed that patients would not necessarily be assigned to their primary nurses (functional care delivery model) and that nurse-patient assignments would be restricted only by acuity levels and new patient setups. In the present study, the information for each treatment protocol (including assigned chair, time, drug type and its dosage) is known, and so uncertainty in infusion duration is not examined. Moreover, after discussions with the staff of the SCC about the importance of receiving treatment from the same nurse at each appointment, the primary care delivery system is taken into account in this study.

Ramos et al. [89] studied two sequential decision problems: a capacity planning problem for assigning a date to appointment requests, and a daily patient-scheduling problem for allocating a chair and time slot for each patient on each day. The authors used the procedure introduced by Saure et al. [95] to deal with the first problem over an infinite horizon and developed an IP model to solve the second problem. In their proposed methodology, once the number of patients for a specific day was known, different patterns of possible patient allocations to chairs were generated to produce a daily schedule. Although using this sequential approach might help to reduce the number of decision variables, implementation could be difficult when partially filled appointment schedules were considered in which some appointment requests had already been accommodated. Furthermore, although the authors took nurse availability into account, they did not distinguish between the part of infusion time when the assigned nurse sets up patient treatment from the part during which the nurse can monitor several patients. In the present study, two of the problems mentioned are considered simultaneously, i.e., capacity planning (determining treatment dates) and patient scheduling (resource allocation), and assigned nursing time is taken into account.

Heshmat et al. [48] addressed the problem of scheduling patient appointments and proposed an approach, inspired by cellular manufacturing, that involved creating clusters of patients. After clusters were formed, each nurse was first assigned to a cluster and then to a group of chairs with the aim of achieving the minimum makespan. By clustering patients and considering them as an input to a mathematical programming model, the authors attempted to reduce the dimensions of the problem. The quality of the final solution was highly dependent on the clusters. Moreover, because of the oversimplification of the problem, many important factors such as the optimal sequence of patients within a cluster, the care delivery system (primary or secondary nurse) and patient preferences were not taken into account.

Recently published studies of the chemotherapy scheduling problem are those of Garaix et al. [33] and Benzaid et al. [17]. The first study proposed a heuristic that uses a greedy randomized adaptive search procedure algorithm to solve the problem of scheduling patient appointments for chemotherapy treatment and consultations by considering drug preparation times. The authors assumed that the sequence of patients for the consultation and treatment stages were the same, which might not be practical. Also, nursing capacity was not taken into account. The authors of the second work examined chemotherapy appointment scheduling problem, a nurse planning problem, and a daily nurse-patient assignment problem in a three-stage procedure. The goal of the first stage was to determine a date and start time for each new patient with the aim of maximizing the number of patients starting their treatments. The second stage attempted to assign patients to nurses in a way that the required staffing level and length of waiting list were minimized. Finally, in the third stage, after simulating last-minute changes including cancellations and nurse absences, the daily assignment problem was executed for the final set of patients and nurses. It was assumed that, in order to deal with the simulated last-minute changes, only the assignment of patients to nurses could change; however, in actual practice (like at the SCC), the assignment of patients to chairs and even their start time might change within an acceptable tolerance (for instance, by up to one hour). Moreover, the authors relaxed the limitation on nurse workload, reassigning patients to nurses to find feasible solutions, which might be unrealistic in some worst-case scenarios for last-minute changes.

2.3 Problem description

Let $\mathcal{T}$ be a set of appointment requests presented in the form of a waiting list. For each patient $p \in \mathcal{P}$, $\mathcal{T}_p \subset \mathcal{T}$ is defined as the set of appointment requests for patient p. Chemotherapy treatments are prescribed in cycles with some rest periods in between. Let r_p denote the minimum amount of time patient $p \in \mathcal{P}$ needs to wait between one infusion and the next. The type and dosage of drugs are determined in the protocols prescribed in regimens. A regimen is a specific plan which determines the amount and frequency of receiving a set of cancer treatments in a cyclical manner. It is very important for patients that appointment dates prescribed in their regimen by the oncologist be respected; otherwise, the efficiency of the chemotherapy treatment may be adversely affected. Appointments that deviate slightly can be acceptable, however, if confirmed by the oncologist. An example of a regimen for lung cancer treatment is shown in Figure 2.1.

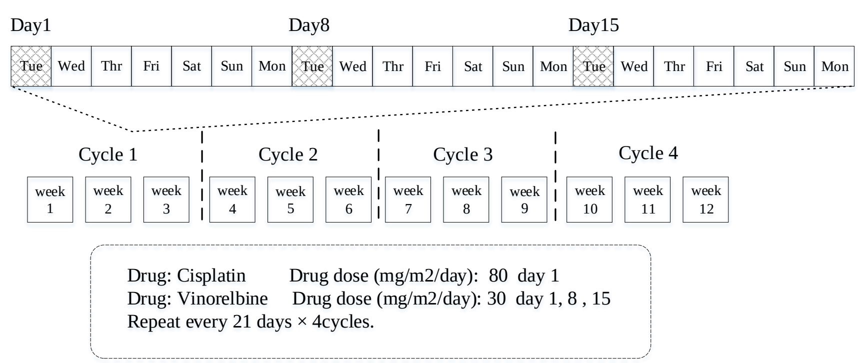


Figure 2.1: Example of a regimen for lung cancer

The model explicitly takes into account the use of two important types of limited resources for chemotherapy treatments: the set of chairs and rooms used to administer infusions, and the set of nurses who set up treatments and monitor the well-being of patients during infusions. Other resources such as oncologists, phlebotomists, and examination rooms are not taken into account in this study as they are not directly related to decisions about chemotherapy appointment scheduling.

Let $\mathcal{C}$ be the set of standard infusion chairs and rooms and the set of drop-in chairs that are normally used to serve walk-in patients, i.e., unscheduled or emergency patients. Drop-in chairs are also used for other types of infusion treatments such as hydration and transfusion. The treatment area is divided into two stations, each comprising a set of chairs and rooms. For any given day, the scheduling clerk has a schedule containing a set of patients already assigned to the available nurses and their associated chairs and rooms. In practice, the SCC uses a buffer, which acts as a virtual space when building the schedules. When the schedule is already fully booked, the scheduling clerk can place an appointment request in the buffer in the hope of a cancellation by another patient, which will allow the request to be moved into the actual schedule.

Let D denote the set of days in the planning horizon. A planning horizon is usually defined as a set of future days for each of which exists an associated partially filled schedule of already booked appointments. We define $\mathcal{S}$ as the set of time slots available each day. For patient $p \in \mathcal{P}$ and treatment $t \in \mathcal{T}_p$, we denote as $[l_t, u_t] \subseteq \mathcal{S}$ the interval of time slots during the day that patient p prefers to receive treatment t, where l_t and u_t are the lower and upper bounds on the start time, respectively. It is not current practice at the SCC to book appointments based on patient preference unless there is a clinical reason for doing so. All preferred appointment times must be confirmed by the head nurse.

We use nurse schedules to define part of the input to this outpatient scheduling problem. The head nurse, considering the number of nurses available during day $d \in \mathcal{D}$ and time slot $s \in \mathcal{S}$, determines the maximum number of patients m_{ds} that can be treated at time slot s of day d. For instance, because of the nurses' lunch break, the reception desk admits no patients between 12:00 p.m. and 1:00 p.m. Nurses take staggered lunch breaks, and nurses remaining on duty during that time monitor only patients whose infusion has already begun. To avoid overtime, the same principle is applied after 3:30 p.m. The clinic admits no additional patients for treatment, and the remaining nurses monitor only patients whose treatment has already started.

It is important to schedule treatments of patient $p \in \mathcal{P}$ so that they are assigned as often as possible to their primary nurses, denoted as $n_p \in \mathcal{N}$. Primary nurses can make patients feel more comfortable because they see a familiar face each day. The primary nurse can also help reduce error and identify problems more quickly, especially when the patient has a reaction to an ongoing infusion. Although the scheduling clerk takes into account, as much as possible, the assignment of patients to primary or even to secondary nurses (nurses who are paired with primary nurses, $c(n) \in \mathcal{N}$) when making appointments, numerous conflicts arise during the manual scheduling process, which in turn, can lead to overtime or many nursing handovers that changes in assigned nurse during treatment.

Many types of drugs (protocols) can be prescribed for the regimen of a patient. The treatment time for protocols ranges from 15 minutes to eight hours. It is assumed that the length of each treatment, denoted as $i_t, t \in \mathcal{T}$, is known and deterministic. The infusion process of each requested treatment can be divided into two stages. During the first stage, called the *direct nursing phase* ($i'_t, t \in \mathcal{T}$), the assigned nurse is completely occupied with the patient undergoing the treatment. During this stage, the nurse is not able to set up any other patient in any other chair or room. This dedicated nursing time can vary from 30 to 90 minutes depending on the patient's physical condition, the complexity of the infusion and the nurse's experience in setting up a treatment. Once a nurse finishes the setup for a patient, she can monitor up to four patients who have already started receiving their treatment. Thus, the total treatment time is the amount of time spent on both the nursing phase and injection process. This is one reason why, in the rotation plan, up to four chairs are assigned to each nurse each day; however, there may be exceptions on some days for specific clinical reasons.

After consultations, oncologist instructions are automatically transmitted electronically to the scheduling clerk. Thus, the online waiting list is continually being lengthened; however, at particular points of time (for example, every morning or twice a day), a number of appointment requests are selected for scheduling and a day, start time, nurse, and chair are assigned to each. The overall goal of this study is to develop a flexible decision-support tool that can be used to design appointment schedules in which more patients are assigned to their primary or secondary nurses, more patients receive appointments that correspond to their time preferences, fewer nurses work overtime or are switched during a patient's treatment (less nurse handovers), while all clinical limitations and written instructions from oncologists are respected. Moreover, in case of a last-minute change in the regimen of a patient or an unexpected resource unavailability, the tool must handle the situation by rescheduling a minimum number of already booked appointments in an adaptive manner. If an already booked appointment is subject to modification, the change in start time must conform to defined parameters.

2.4 Solution methodology

This subsection details a flexible and adaptive procedure that tackles the complexity associated with the online scheduling of patient appointments where the number of future incoming requests is not known. Using two IP models, the proposed procedure schedules appointment requests from waiting lists that are continually being updated and reschedules already booked appointments once either new information is received or a last-minute change occurs. In what follows, two mathematical formulations for the scheduling and rescheduling problems are presented, and the proposed scheduling procedure is described.

2.4.1 Scheduling appointment requests from a waiting list

This sub-section presents an IP to formulate the scheduling problem for all requests from an incoming waiting list over a pre-defined planning horizon. The sets and parameters used in the IP are presented in Table 2.1.

To formulate the problem, x_{tdcs} is defined as a binary variable that takes value 1 if and only if treatment t is scheduled to start in slot s of day d on chair c. Also, y_t is introduced as a binary variable taking value 1 if and only if treatment t on the waiting list is directed to the buffer (i.e., it is not scheduled). The goal of this proposed IP formulation for the scheduling of appointment requests is to minimize the following objective functions:

1. The first objective function evaluates the number of appointments on the waiting list for which patients are assigned to floating nurses (other than the patient's primary or secondary nurse). This objective, which is a function of variable x, can be calculated as follows:

   $$g_1(x) = \sum_{t \in \mathcal{T}} \sum_{d \in \mathcal{D}_t} \sum_{c \in \mathcal{C}_t} \sum_{s \in \mathcal{S}_t} c^1_{tdc} x_{tdcs},$$

   where, c^1_{tdc} parameter equals 1 if the nurse associated with chair c on day d is not the patient's primary nurse or co-nurse, 0 otherwise.

2. The second objective function determines the number of appointments for which

Table 2.1: Sets and parameters used in the model

$\mathcal{P}$	Set of patients requesting appointments
$\mathcal{D}$	Set of days in the planning horizon (H)
$\mathcal{S}$	Set of slots for each day of the planning horizon
$\mathcal{N}$	Set of all nurses working in the clinic
$\mathcal{C}$	Set of all chairs available in the clinic
$\mathcal{T}_p$	Set of treatments prescribed in the regimen of patient p, $\mathcal{T}_p = \{t_p^1, ...\}$
$\mathcal{T}$	Set of all treatments requested for the patients on the waiting list, $\mathcal{T} = \bigcup_{p\in\mathcal{P}} \mathcal{T}_p$
d_t	Target day of treatment t, $d_t \in \mathcal{D}$
λ_t	Amount of flexibility (deviation) allowed for the date of treatment t
D_t	Subset of days on which treatment t can be received according to λ_t, $D_t \subset \mathcal{D}$, i.e. $D_t = \{\max(d_t - \lambda_t, 1), ..., \min(d_t + \lambda_t, \lvert\mathcal{S}\rvert)\}$
$\mathcal{S}_t$	A subset of slots in which treatment t can be received, $\mathcal{S}_t \subset \mathcal{S}$
$\mathcal{C}_t$	A subset of chairs in which treatment t can be received, $\mathcal{C}_t \subset \mathcal{C}$
i_t	Number of slots required to complete treatment t (duration)
i'_t	Number of slots required as nursing time for treatment t
l_t	A lower bound on the start time of treatment t, $l_t \geq 1$
u_t	An upper bound on the start time of treatment t, $u_t \leq \lvert\mathcal{S}\rvert - i_t + 1$
S_{cd}	Work hours (start and end times) of the nurse associated with chair c on day d, $S_{cd} = [s_{cd}, e_{cd}]$, $S_{cd} \subset \mathcal{S}$
$\mathcal{S}_t^P$	Preferred start time interval of treatment t, $\mathcal{S}_t^P = [l_t, u_t]$, $S_t^P \subset \mathcal{S}$
n_p	Primary nurse associated with patient p, $n_p \in \mathcal{N}$
$c(n)$	Co-nurse associated with each primary nurse, $c(n) \in \mathcal{N}$
$\mathcal{C}_p^1$	Set of chairs assigned to the primary nurse of patient p, $\mathcal{C}_p^1 \subset \mathcal{C}$
$\mathcal{C}_p^2$	Set of chairs assigned to the secondary nurse (the co-nurse of the primary nurse) of patient p, $\mathcal{C}_p^2 \subset \mathcal{C}$
$p(t)$	Patient associated with treatment t, $p(t) \in \mathcal{P}$
$n(c,d)$	Nurse assigned to chair c on day d, $n(c,d) \in \mathcal{N}$
r_p	Minimum number of rest days between consecutive treatments of patient p
a_{dcs}	A 0-1 parameter determining the current availability of slot s of chair c on day d
m_{ds}	Current maximum number of patients that can be admitted into slot s of day d
I_d	$= \{t \in \mathcal{T} : d - \lambda_t \leq d_t \leq d + \lambda_t\}$
$C_{tt'}^d$	$= \{c \in \mathcal{C}, d \in \mathcal{D} : n(c,d) = n_{p(t)} = n_{p(t')}\}$

patients do not receive their preferred time slot and is defined as follows:

$$g_2(x) = \sum_{t\in\mathcal{T}} \sum_{d\in\mathcal{D}_t} \sum_{c\in\mathcal{C}_t} \sum_{s\in\mathcal{S}_t} c_{ts}^2 x_{tdcs},$$

where,

$$c_{ts}^2 = \begin{cases} 1, & \text{if } t \in \mathcal{T}_p \text{ and } s \notin [l_t, u_t] \\ 0, & \text{otherwise.} \end{cases}$$

3. The third objective function calculates nurse overtime and can be stated as follows:

$$g_3(x) = \sum_{t\in\mathcal{T}} \sum_{d\in\mathcal{D}_t} \sum_{c\in\mathcal{C}_t} \sum_{s\in\mathcal{S}_t} c^3_{tds} \left(s + i_t - 1 - e_{cd}\right) x_{tdcs},$$

where,

$$c^3_{tds} = \begin{cases} 1, & \text{if } s > e_{cd} - i_t \\ 0, & \text{otherwise.} \end{cases}$$

Recall that $S_{cd} = [s_{cd}, e_{cd}]$ is the set of time slots representing the working hours of the nurse associated with chair c, where s_{cd} denotes the start time and e_{cd} represents the finish time of the nurse assigned to chair c on day d. According to the above equation, each treatment starting in any time slot within the interval $s \in [e_{cd} - i_t + 1, e_{cd} - 1]$, will finish after the end time of the assigned nurse, i.e. $s + i_t - 1 > e_{cd}$, which will cause an overtime of $(s + i_t - 1 - e_{cd})$ units of time to the clinic.

It is worth mentioning that, at the SCC, nursing handovers occur frequently (i.e., a nurse hands over the responsibility of care to another nurse on finishing her scheduled working hours). In other words, although the appointment schedule are arranged so that some patients start their treatment within a nurse's working hours and this nurse works overtime until the infusion is completed, in practice patients usually finish with another nurse within her predefined working hours. This is possible because appointments are booked so as to guarantee that nurses are available to monitor (not to set up, as a setup requires a nurse's complete attention) the patients undergoing treatment. The use of the proposed IP tool will minimize nurse overtime, which will indirectly minimize the number of nurse handovers.

4. Finally the fourth objective corresponds to the number of appointment requests

being directed to the buffer (and not scheduled) which can be defined as follows:

$$g_4(y) = \sum_{t \in \mathcal{T}} y_t.$$

Using the above-mentioned objectives and decision variables, the model for the off-line scheduling procedure that accommodates appointment on the waiting list can be formulated as follows:

$$\text{minimize} \quad \sum_{t \in \mathcal{T}} \sum_{d \in \mathcal{D}_t} \sum_{c \in \mathcal{C}_t} \sum_{s \in \mathcal{S}_t} (c^1_{tdc} + c^2_{ts} + c^3_{tds}(s + i_t - 1 - e_{cd}))x_{tdcs} + \sum_{t \in \mathcal{T}} y_t$$

subject to

$$\sum_{d \in \mathcal{D}_t} \sum_{c \in \mathcal{C}_t} \sum_{s \in \mathcal{S}_t} x_{tdcs} + y_t = 1 \qquad t \in \mathcal{T} \tag{2.1}$$

$$\sum_{d \in \mathcal{D}_t} \sum_{c \in \mathcal{C}_t} \sum_{s \in \mathcal{S}_t} d\left(x_{(t+1)dcs} - x_{tdcs}\right) \geq r_p \qquad p \in \mathcal{P}, (t, t+1) \in \mathcal{T}_p \tag{2.2}$$

$$\sum_{t \in \mathcal{T}} \sum_{s' = \max(s - i_t + 1, 1)}^{s} x_{tdcs'} \leq a_{dcs} \qquad d \in \mathcal{D}, c \in \mathcal{C}, s \in \mathcal{S} \tag{2.3}$$

$$\sum_{t' \in I_d \setminus \{t\}} \sum_{s'=s}^{\min(s+i_t-1, |S|)} (x_{t'dcs'} + x_{tdcs} - 1) \leq 0 \qquad t \in \mathcal{T}, d \in \mathcal{D}_t, c \in \mathcal{C}_t, s \in \mathcal{S}_t \tag{2.4}$$

$$\sum_{t' \in I_d \setminus \{t\}} \sum_{c' \in \mathcal{C}^d_{tt'} \setminus \{c\}} \sum_{s'=s}^{\min(s+i'_t-1, |S|)} (x_{t'dc's'} + x_{tdcs} - 1) \leq 0$$

$$t \in \mathcal{T}, d \in \mathcal{D}_t, c \in \mathcal{C}_t, s \in \mathcal{S}_t \tag{2.5}$$

$$\sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}_t} x_{tdcs} \leq m_{ds} \qquad d \in \mathcal{D}, s \in \mathcal{S} \tag{2.6}$$

$$\sum_{t \in \mathcal{T}} \sum_{d \in \mathcal{D}_t} \sum_{c \in \mathcal{C}_t} \sum_{s \in (\mathcal{S} \setminus \mathcal{S}_{cd} \cup \mathcal{S} \setminus \mathcal{S}_t)} x_{tdcs} = 0 \tag{2.7}$$

$$x_{tdcs}, y_t \in \{0, 1\} \qquad t \in \mathcal{T}, d \in \mathcal{D}_t, c \in \mathcal{C}_t, s \in \mathcal{S}_t. \tag{2.8}$$

Constraints (2.1) determine whether an appointment request should be scheduled or directed to the buffer. If it is decided that an appointment will be scheduled, a day, chair, and start time are assigned. These constraints consider the flexibility of λ_t days, as the maximum deviation from the target day, in D_t $t \in \mathcal{T}$. It should be noted that appointment requests that remain in the buffer can either be scheduled along with those on the next waiting list received, wait to be replaced with the cancelled appointments, or be accommodated by rescheduling appointment requests for the given day. The precedence relationships among the treatments of each patient are taken into account in constraints (2.2). The appointments for each patient must be at least r_p days apart to allow rest periods between successive appointments.

Constraints (2.3) guarantee that a treatment can only be started if the necessary number of consecutive slots (dependent on the duration of that treatment) are available for the assigned chair. If $a_{dcs} = 0$, it means that chair c on day d is not available at time slot s, and if treatment t starts at a time slot from $s - i_t + 1$ to s, the treatment certainly will not be ongoing at slot s. Furthermore, if $a_{dcs} = 1$, only one treatment can be ongoing at the available time slot. Constraints (2.4-2.5) prevent more than one treatment using the same chair in the same time slot on the same day. In particular, according to Constraints (2.4), once a time slot on a chair is assigned to a treatment, the succeeding slots, as many as required for the infusion duration, will be occupied and no other patient can be assigned to that specific slot to receive treatment. Similarly, according to Constraints (2.5), once a time slot is allotted to a patient, as many succeeding time slots as required to set up the treatment will be assigned to the nurse, who will be totally occupied in setting up that treatment and unable to set up any other patient in any chair. Constraints (2.6) limit the number of patients allocated to each time slot. For example, during break times, no patients will be set up to start treatment; however, previously admitted patients will continue to be monitored. Moreover, in order to avoid incurring overtime costs to the clinic, no patient is set up to start treatment in the later time slots of the day. Constraint (2.7) ensures that no nurse is assigned to set up a patient for treatment before she is scheduled to start work or after she finishes her workday. It also makes sure that all treatments start in time slots that are clinically justified ($s \in \mathcal{S}_t$). Finally, Constraints (2.8) enforce the zero-one restrictions on binary variables of the above linear IP formulation. Since minimizing total tardiness on a single machine is NP-hard [30], and minimizing overtime in the above IP is mathematically equivalent to minimizing the total tardiness, our outpatient scheduling problem is also NP-hard.

The weighted-sum method [75] is used to solve the above multi-objective optimization problem and the analytic hierarchy process (AHP) [91] is used to determine the coefficient of each objective function ($\sum_{i=1}^{4} \alpha_i = 1, \ 0 \leq \alpha_i \leq 1$). The AHP uses a pairwise comparison of the importance of objective function extracted from the opinions of the experts working in the SCC. According to the AHP, each objective function is first divided by the difference of the Nadir and Utopia points such that each one, regardless of its scale, is between 0 and 1. Table 2.2 shows the relative importance rates given on comparing each pair of objective functions, on a scale of 1 to 9, to determine the coefficients of objective functions for the multi-objective optimization models presented.

It is clear from Table 2.2, for instance, that the second objective function is eight times more important than the first, and the third objective function is three times

Table 2.2: Pairwise comparison of objectives functions

Objective functions	(1)	(2)	(3)	(4)
(1) Nurse-patient assignments	1	1/8	1/6	1/7
(2) Patient preference	8	1	1/2	1/2
(3) Total nurse overtime	6	2	1	1/3
(4) Being scheduled (Not sent to buffer)	7	2	3	1

less important than the fourth. To calculate the final weights, first the values in each column of the pairwise comparison are summed, and then each value is divided by the total of its column. Afterwards, the average of each row determines the relative importance of the associated objective function [91].

2.4.2 Rescheduling of already booked appointments

This section formulates the problem of the daily rescheduling of already booked appointments as a second IP. Since rescheduling occurs on a daily basis, each patient must have a single appointment on any particular day ($\hat{d} \in D$). In addition to the notation introduced in Table 2.1, L_n is defined as the maximum number of patients that can be assigned to a nurse ($n \in \mathcal{N}$) on a particular day. Since patients who already have an appointment are being rescheduled, s_t^0 is defined as the initial time slot for treatment t, and γ_t is defined as the maximum allowable number of time slots for which treatment $t \in \mathcal{T}_{\hat{d}}$ may be moved in case of change. Another controllable parameter, f, refers to the maximum percentage of appointments for which the start time can be changed. Moreover, I'_t is defined as the allowable amount of time by which the start time of treatment t may change as follows: $I'_t = \{\max(1, s_t^0 - \gamma_t), ..., \min(s_t^0 + \gamma_t, |\mathcal{S}|)\}$. To formulate the rescheduling problem, the same decision variables and objective functions as in the previous IP model are used taking a fixed day index ($\hat{d}$) into account. The daily rescheduling problem can be formulated as the following IP model:

$$\text{minimize} \quad \sum_{t \in \mathcal{T}_{\hat{d}}} \sum_{c \in \mathcal{C}_t} \sum_{s \in \mathcal{S}_t} c_{t\hat{d}cs} x_{t\hat{d}cs} + \sum_{t \in \mathcal{T}_{\hat{d}}} c'_t y_t$$

subject to (2.3), (2.6) - (2.8)

$$\sum_{c\in\mathcal{C}_t}\sum_{s\in(I'_t\cap S_t)} x_{t\hat{d}cs} + y_t = 1 \qquad t\in\mathcal{T}_{\hat{d}} \tag{2.9}$$

$$\sum_{t\in\mathcal{T}_{\hat{d}}}\sum_{c\in\mathcal{C}^1_{p(t)}}\sum_{s\in\mathcal{S}_t} x_{t\hat{d}cs} \le L_n \qquad n\in\mathcal{N} \tag{2.10}$$

$$\sum_{t\in\mathcal{T}_{\hat{d}}}\sum_{c\in\mathcal{C}_t}\sum_{s\in\mathcal{S}_t\setminus\{s^0_t\}} x_{t\hat{d}cs} \le \lfloor f\times|\mathcal{P}|\rfloor \tag{2.11}$$

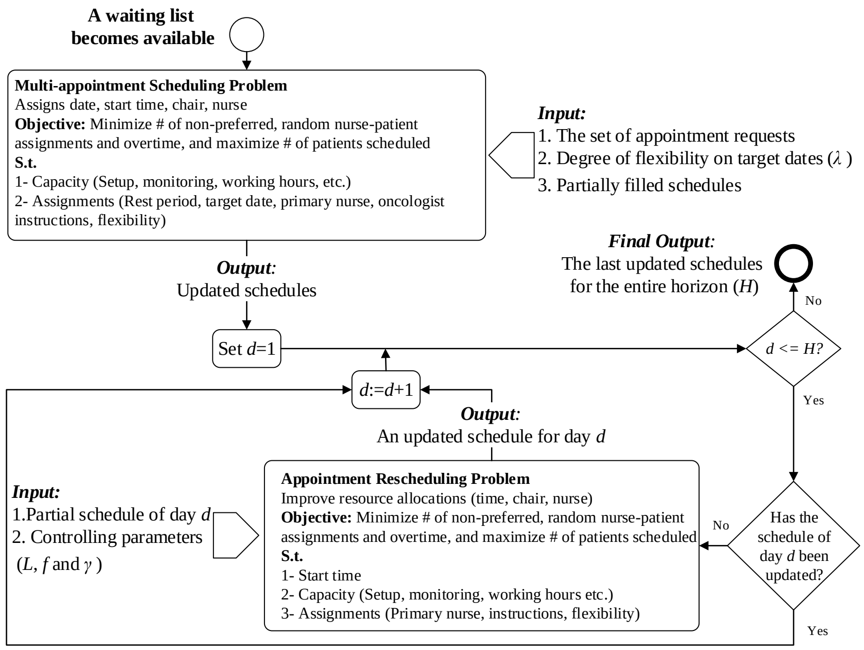
Constraints (2.9) are the equivalent of Constraints (2.1). Theses constraints also consider a flexibility of γ_t time slots on the initially booked appointment (s^1_t) for treatment $t\in\mathcal{T}_{\hat{d}}$. Constraints (2.10) guarantee that the number of patients assigned to each nurse, on all chairs assigned to that nurse, does not exceed the pre-defined threshold ($L_n, n\in\mathcal{N}$). According to constraint (2.11), the number of rescheduled appointments (in terms of time slot) is limited to only $f\%$ of the total number of patients. In this IP, constraints (2.3) are used by assuming that $a_{\hat{d}cs}=1 \quad \forall s\in\mathcal{S}, c\in\mathcal{C}$. It should be noted that, since all the patients of a particular day $\hat{d}$ are rescheduled, all slots of day $\hat{d}$ on all chairs are considered available. Moreover, Constraints (2.6) - (2.8) are used given that $\mathcal{T}=\mathcal{T}_{\hat{d}},\ D_t=\{\hat{d}\},\ \forall t\in\mathcal{T}$. According to Sadki et al. [92] and Condotta and Shakhlevich [24], determining the start time of chemotherapy appointments and assigning nurses to each treatment of a patient on a particular day is an NP-complete problem even under the most simple assumptions. Therefore, the above formulated rescheduling problem is also NP-hard.

2.4.3 The overall flexible and adaptive scheduling procedure

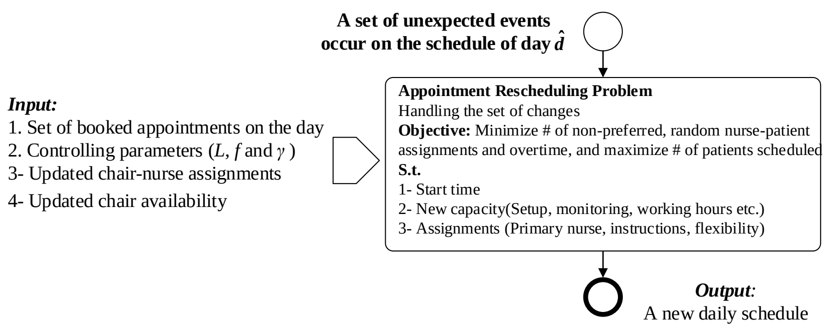
The proposed procedure receives a set of appointment requests as an input that needs to be processed on specific target days. Decisions are made sequentially as information about the requests arrives. When a set of appointment requests in the form of a waiting list is received, the procedure uses whatever information is available (duration of treatment, recommended dates, patient preferences, clinical requirements, etc.) along with the existing partially filled appointment schedule and information about the availability of nurses, chairs, and rooms.

In order for the proposed procedure to be adaptive (i.e., to provide appropriate responses to unexpected events and to facilitate optimal resource allocation), an additional module is considered, which uses the model presented for the rescheduling problem. Several types of unexpected events can occur. For example, a nurse may call in sick, which affects the nurse rotation plan. In such a situation, the head nurse reschedules the patients previously assigned to the absent nurse with the aim of reducing the potential negative impact of the absence. Another example of an unexpected event is a breakdown in the equipment of a chair. As a result, patients are reassigned to other chairs. A further example is a change in instructions: after a consultation, an oncologist decides to: change the drug type (and consequently the infusion duration), cancel (or postpone) one or more upcoming treatments, or request an urgent treatment. In any of these situations, the appointment schedule is affected, and some of the appointments have to be rescheduled with the aim of minimizing changes to previously booked appointments.

Two reasons drive the use of the rescheduling module. The first is to allow the proposed procedure to respond to unexpected events reactively. This reactive action (rescheduling) can make significant improvements even when no changes to the start time of previously booked appointments are allowed ($\gamma = 0$) because appointments can be moved horizontally in the schedule by changing chairs or assigned nurses. The second reason to apply a rescheduling phase to the created schedule is that more patients can be assigned to their primary or, minimally, to their secondary nurse in their preferred time slots. This can be achieved by running the rescheduling model for the appointment schedule of a specific day after patients on the waiting list have been accommodated, even if it is assumed that the number of changes to start times is zero. The diagram in Figure 2.2 shows how incoming requests and unexpected events are dealt with.



(a) The proposed procedure to handle a list of requests



(b) The proposed procedure to handle last-minute changes

Figure 2.2: A diagram explaining the steps of the proposed procedure

The procedure is triggered by either an incoming waiting list or an unexpected event. The scheduling phase of the procedure uses the set of appointment requests in the form of a waiting list (with all the required information), the current partially filled schedules for all days in the planning horizon (H) and a controlling parameter about the flexibility of the date of each request ($\lambda_t, t \in \mathcal{T}$) to accommodate all the appointment requests and determine their assigned time slots, chairs and nurses. The flexible parameter in the scheduling phase allows treatment appointment t to be booked λ_t days earlier or later than the prescribed date d_t. It should be noted that by assuming this flexibility, there still must be constraints to ensure that a sufficient rest period is considered between consecutive treatments (Constraints (2.2) of the first IP). After accommodating all appointment requests on the waiting list, for every day for which there has been a request, the daily rescheduling model is executed to improve resource allocation mentioned above as the second reason for the rescheduling model.

After the rescheduling model is executed, appointments can still be changed; however, the number of changes is limited by the controllable parameters (f, γ). This rescheduling is practical, given that patients are usually informed about their start time a few days prior to the day of the appointment. Therefore, changing the start time during the execution of the rescheduling model before confirming the appointment time does not have a negative impact on patients. If an oncologist recommends not changing a given patient's appointment for a clinical reason, such an instruction can be considered in S_t (see Table 2.1), which denotes the set of time slots in which treatment t can start.

Furthermore, the rescheduling phase of the proposed adaptive procedure can be executed for a given day ($\hat{d} \in \mathcal{D}$) if a last-minute change occurs or if data for that particular day is updated. The scheduling phase is executed using two flexible parameters, γ and f and the set of changes occurring on day $\hat{d}$ so as to deal with changes efficiently. In these circumstances, the rescheduling model is not executed to improve resource allocation; rather, the goal is to change the appointment schedule to handle last-minute changes occurring on a given day, while reducing as much as possible the negative impact of such changes on the already-existing schedule for that day.

2.5 Computational results

This section presents the results of computational experiments carried out to assess the performance of the proposed scheduling procedure in comparison with the current manual scheduling procedure used. The results of the sensitivity analysis performed to assess the impact of assuming varying flexibility parameters on the performance of the system are also presented. The overall procedure was coded in C++ and all mathematical models were solved with CPLEX 12.7.1 using Concert Technology and run on an Intel Xeon CPU E5-2687W v3 processor at 3.10 GHz and 750 GB of RAM under a Linux environment.

In order to compare the results of the proposed adaptive procedure to the scheduling system actually used at the SCC, we first took snapshots of the state of that system, once at the beginning of the planning horizon and once again at the end. The state of the system refers to the partially filled schedules that show the start time, the chair or room, and the nurse assigned to each patient already booked into the system for the planning horizon being studied. The remaining availability of nurses and chairs is also captured. By comparing these two recorded states, the number of patients added and all the information regarding their appointment requests can be easily extracted. The planning horizon (H) for our study was assumed to be 20 working days. There are two stations at the SCC, one with three drop-in chairs, 13 regular chairs, and two drop-in rooms and the other with 14 normal chairs and three rooms. The number of nurses working at the clinic is 20. The clinic opens at 7:00 a.m. and finishes all treatments by 6:00 p.m. In actual practice, since more than one scheduling clerk books patient appointments, it is hard to track the exact number of waiting lists, telephone calls, and in-person requests that the scheduling clerks received during the planning horizon. Therefore, the state of the system was captured at two different points of time, 20 working days apart, and the two states were compared to identify the exact number of appointment requests added to or removed from the earlier schedule.

For the purpose of evaluation, it was assumed that appointment requests were added to the system in the form of waiting lists according to one of the following options:

1. All requests are received in a single waiting list containing 220 patients (i.e., off-line).

2. Requests appear on three waiting lists, each containing 60 to 90 patients.

3. Requests appear on six waiting lists, each containing 30 to 45 patients.

4. Requests appear on 12 waiting lists, each containing 15 to 25 patients.

As mentioned, tracking the exact number of waiting lists and the information they contained, including the combination of patients within each list and the exact arrival time of requests, was not practical. Multiple clerks booked appointments on any given day, and appointment requests were continually being added to the system. Examination of the printed waiting lists revealed that each list contained a mix of appointment requests, most with target dates within the succeeding few days and a small number with more remote target dates. Therefore, to generate appropriate waiting lists (i.e., that resembled the actual waiting lists), the appointment requests were distributed among considered waiting lists within the planning horizon so that each list contained appointment requests with target dates that were both imminent and more distant.

Using the AHP method, the final weights of the four defined objective functions in both IP formulations presented in sections 2.4.1 and 2.4.2 are as follows: $\alpha_1 = 0.04$, $\alpha_2 = 0.23$, $\alpha_3 = 0.26$ and $\alpha_4 = 0.46$. Recall that α_1 is the weight associated with nurse assignments, α_2 with patient preferences, α_3 with total nurse overtimes (handovers) and α_4 with unscheduled appointment requests (in the buffer). Each time slot was assumed to be 30 minutes, and a shrinking horizon fashion was used. The maximum percentage of appointments for which the start time could be changed (f) was set at 20%. The number of appointments booked on drop-in chairs was set at zero and the maximum number of patients assigned to a nurse (L_n, $n \in \mathcal{N}$) was set at eight. During the planning horizon studied, appointment requests arrived for 220 patients requiring a total of 339 treatments with target dates within that horizon. Up to six appointments. might be requested for a patient. Furthermore, at the beginning of the planning horizon, 476 appointments had already been booked on all the days of the horizon. Therefore, 815 treatments (with a target dates within H) in total that were given during 20 working days. This number excludes services like hydration, transfusion, and treatments for walk-ins and emergency patients that are delivered on drop-in chairs.

Table 2.3 provides the input parameters used in both IP models, and Table 2.4 summarizes the design of the experiments that divided patients among varying numbers of waiting lists, i.e. $N_w = 1, 3, 6, 12$.

Table 2.3: Input parameters used in the models

Parameter	*Value*
Planning horizon (H)	20 working days
Number of requests for appointment	339
Number of booked appointments in the partial schedule	476
Number of chairs (including drop-ins)	35
Capacity of drop-in chairs and rooms	0
Number of nurses	20
Working hours of the clinic	[7:00 a.m., 6:00 p.m.]
Maximum allowable percentage of appointments to reschedule (f)	20 %
Nurse capacity (L_n)	8 patients
Monitoring capacity (m)	4 patients
Weight of nurse-patient assignments (α_1)	4%
Weight of patient preferences (α_2)	23%
Weight of nurse overtime (handovers) (α_3)	26%
Weight of unscheduled appointments in buffer (α_4)	46%

Table 2.4: Design of experiments to evaluate the proposed scheduling procedure

Experiment	*# of waiting lists (N_w)*	*# of patients in each list*	*# of requests in each list*
EX1	1	220	339
EX2	3	[60, 90]	[105, 118]
EX3	6	[30, 45]	[44, 72]
EX4	12	[15, 25]	[17, 53]

2.5.1 Evaluating the proposed scheduling procedure

In this sub-section, we provide the results of the evaluation of the proposed scheduling procedure by comparing the appointment schedule actually used to those generated by the experiments and the data set explained above. Table 2.5 presents the results of the scheduling system actually used at the SCC during the planning horizon studied.

Table 2.5: Results obtained from the scheduling system used at the SCC

Objective function	*value*
Number of assignments to primary nurses (N_{pri})	242
Number of assignments to secondary nurses (N_{sec})	40
Number of non-preferred appointments (N_{prf})	8
Total nurse overtime (O_{nrs})	39 hrs
Number of assignments to drop-in chairs ($N_{D/I}$)	7
Number of appointment requests directed to the buffer (N_{buf})	0

Table 2.6 shows the results obtained from the proposed flexible and adaptive procedure for the various experiments designed, i.e., EX1($N_w = 1$), EX2($N_w = 3$), EX3($N_w = 6$) and EX4($N_w = 12$); different levels of flexibility, i.e., flexibility of date (λ) and changing the start time (γ); and whether the rescheduling module is applied to a schedule while it is being generated or not. In this table, the positive (negative) values below the cells with % signs show the improvement (deterioration for negative values) obtained by the proposed procedure as compared to the scheduling system actually used at the SCC. Moreover, N_{dev} denotes the number of appointments that deviated from their target date, and N_{ch} the number of appointments that were changed after being booked. The best value obtained for each objective function is shown underlined and in bold.

Table 2.6: Results obtained using the flexible and adaptive scheduling procedure

Problem	*Data info*				*Results*								
	Experiment	*Rescheduling*	λ	γ	N_{pri}	N_{sec}	N_{prf}	O_{nrs} (min)	$N_{D/I}$	N_{buf}	N_{ch}	N_{dev}	*time(s)*
1	EX1	No	0	-	+33	+70	-388	+36	0	0	0	0	113
2	EX1	No	1	-	+45	+63	-363	+26	0	0	0	65	1078
3	EX1	Yes	0	0	+61	+195	+25	+72	0	0	0	0	358
4	EX1	Yes	1	0	+72	+183	+25	+72	0	0	0	65	1321
5	EX1	Yes	0	30	+69	+205	**+38**	**+79**	0	0	61	0	430
6	EX1	Yes	1	30	**+79**	+205	**+38**	+74	0	0	56	65	1385
7	EX2	No	0	-	+33	+78	-375	-18	0	4	0	0	23
8	EX2	No	1	-	+43	+75	-400	-18	0	2	0	84	143
9	EX2	Yes	0	0	+ 57	+198	+13	+72	0	2	0	0	286
10	EX2	Yes	1	0	+69	+183	+25	+67	0	1	0	87	397
11	EX2	Yes	0	30	+67	+205	**+38**	+72	0	1	90	0	370
12	EX2	Yes	1	30	+77	+198	**+38**	+72	0	0	89	85	480
13	EX3	No	0	-	+31	+78	-400	-26	0	4	0	0	14
14	EX3	No	1	-	+43	+60	-413	-38	0	1	0	78	83
15	EX3	Yes	0	0	+58	+195	-25	+74	0	2	0	0	322
16	EX3	Yes	1	0	+67	+175	+13	+72	0	2	0	76	390
17	EX3	Yes	0	30	+64	**+238**	**+38**	+74	0	1	90	0	420
18	EX3	Yes	1	30	+78	+195	**+38**	+72	0	0	96	76	483
19	EX4	No	0	-	+30	+80	-400	-23	0	7	0	0	10
20	EX4	No	1	-	+43	+63	-463	-26	0	0	0	82	57
21	EX4	Yes	0	0	+57	+203	-25	+72	0	2	0	0	378
22	EX4	Yes	1	0	+67	+178	0	+74	0	1	0	75	418
23	EX4	Yes	0	30	+65	+225	+25	+72	0	1	105	0	484
24	EX4	Yes	1	30	+76	+210	+25	+74	0	0	100	82	541

Table 2.6 demonstrates that, in most of the test problems, the proposed procedure provides results that are significantly superior to the system actually used at the SCC. As expected, the best results were obtained when assuming that all appointment requests were known in advance (single waiting list) and executing the rescheduling model after accommodating the appointment requests on each waiting list. This was especially true when the flexibility in the prescribed dates was set at one day ($\lambda = 1$), and changes of one half hour ($\gamma = 30$) to appointment start times were limited to 20% of all requests (test problem 6). The test problem with a single list of requests was solved optimally in less than 25 minutes, which is a reasonable amount of time to handle 378 appointments. Note that with the scheduling system currently in use at the SCC, scheduling such a volume of appointment requests manually would take several days.

Although it is evident that waiting longer for appointment requests to be gathered and then accommodating more patients achieves better results, even considering

12 waiting lists (which is the closest number of lists to reality) yields significant improvements compared to the scheduling system actually used. For instance, in the test problem 24 of Table 2.6, the number of patients assigned to primary and secondary nurses increased by 76% and 210% respectively. There is a significant decrease (74%) in the total nurse overtime during the planning horizon. There was also a certain improvement (25%) in the number of appointment times that corresponded to patient preferences. These results were obtained by changing 82 appointments ($N_{dev} = 82$) out of 339 requests by only one day and by moving 100 appointments ($N_{ch} = 100$) by only 30 minutes.

As can be seen in Table 2.6, use of the rescheduling model further improves the quality of daily schedules. The scheduling IP model dealt with partially filled schedules that already contained 476 already booked appointments that had not necessarily been booked as efficiently as they could have been. This set of booked appointments was used as input for the proposed procedure. Once the appointment requests on a waiting list were incorporated into partial schedules, the rescheduling model was executed for all existing appointments (both new and previously booked) to make sure that patients received the best possible appointment time. For test problems 3, 4, 9, 10, 15, 16, 21, and 22, the rescheduling mode further improved the schedules even when the permitted change of start time (flexibility of time slot) was set at zero (if $\gamma = 0$, only the chair of treatment t may change during the rescheduling process). It should be noted that even better results can be expected if, each time that a new waiting list is received, the scheduling model is executed on all appointment requests (new and already booked). However, from a practical perspective, such a scheduling problem is too complex to be solved. This is why a sequential approach with scheduling and rescheduling models has been proposed to accommodate appointment requests and to improve the quality of daily schedules.

2.5.2 Impact of varying different controllable parameters

In this sub-section, the impact of different controllable and flexible parameters on the procedure is analyzed to provide insights for managers. In order to conduct such an evaluation process, experiment EX4 with $N_w = 12$ was used because a high frequency of waiting lists is closer to the reality of the SCC. However, it was clear that the longer the wait before gathering information about incoming requests, the better the results of the scheduling procedure.

Table 2.7 summarizes the results obtained for different values of f, λ and γ. Note that $f = 0$ implies that no appointment is allowed to change after being booked, and $f = 100$ denoting no restriction on the number of changed appointments.

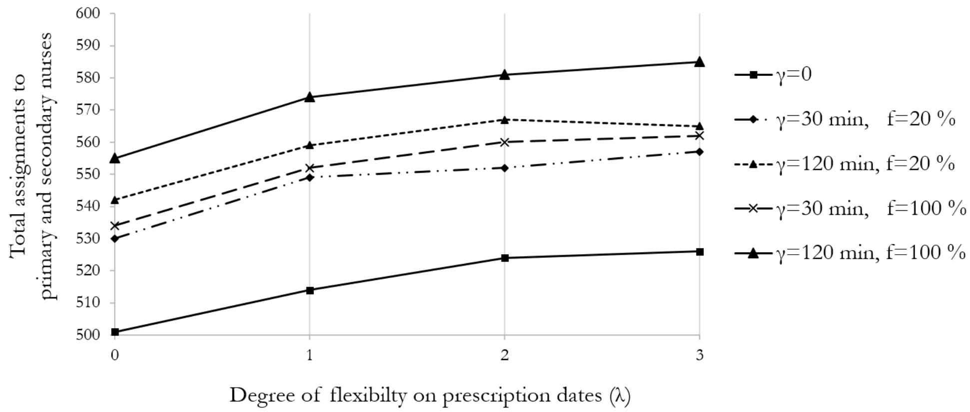
The trends revealed by this evaluation are presented in Figure 2.3. It can be easily seen that the more flexible the prescribed dates (λ), the better the results in terms of assigning patients to their primary and secondary nurses and giving them their preferred appointment times. However, this improvement becomes less intense as higher degrees of flexibility are assumed. Therefore, it can be concluded that assuming even a small amount of flexibility on the prescribed date (a single day) can result in significant improvements in most of the objective functions, and there is no need for the scheduling system to deviate from patients' prescribed dates more than that.

Furthermore, total overtime does not seem to increase or decrease as a higher degree of flexibility is taken into account. On the other hand, in almost all sub-figures of Figure 2.3, the effect of adjusting the percentage of changed start times of booked appointments is observable. As more changes of appointment start times (γ) are permitted during the rescheduling phase of the adaptive procedure, improved results are achieved but, similar to flexibility in dates, the rate of improvement lessens as higher values for γ are set. Moreover, allowing the adaptive procedure the capacity to change all appointments (if needed) rather than limiting the rate of allowable change

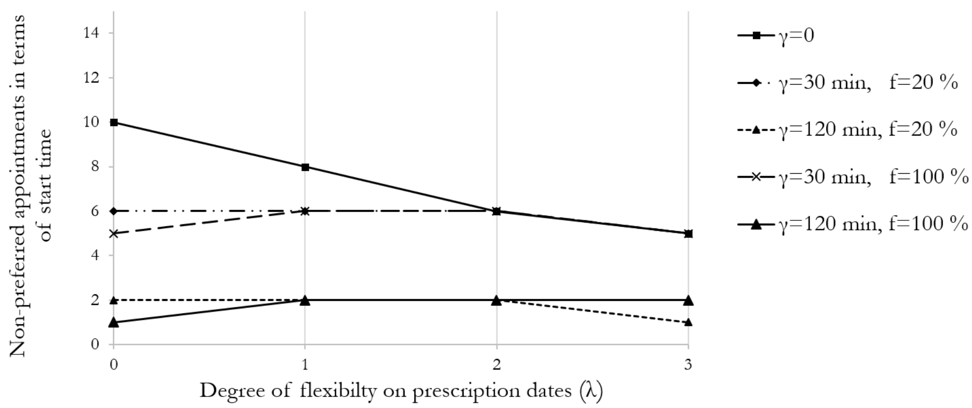
Table 2.7: Results of evaluating the effect of controllable parameters on the performance

Problem	*Data info*			*Results*								
	f (%)	λ	γ (*min*)	N_{pri}	N_{sec}	N_{prf}	O_{nrs} (min)	$N_{D/I}$	N_{buf}	N_{ch}	N_{dev}	*time*(s)
1	0	0	-	+57	+203	-25	+72	0	2	0	0	378
2	0	1	-	+67	+178	0	+74	0	1	0	75	418
3	0	1	-	+69	+190	-25	+74	0	2	0	81	468
4	0	3	-	+71	+183	+13	+72	0	1	0	104	540
Average				*+66*	*+188*	*-9*	*+73*	*0*	*2*	*0*	*65*	*451*
5	20	0	30	+65	**+225**	+25	+72	0	1	105	0	486
6	20	0	60	+72	+208	+63	+67	0	0	102	0	630
7	20	0	90	+74	+198	+75	+74	0	0	118	0	810
8	20	0	120	+74	+200	+75	+72	0	0	112	0	1020
9	20	1	30	+76	+210	+25	+74	0	0	100	82	546
10	20	1	60	+81	+180	+75	+72	0	0	105	74	684
11	20	1	90	+82	+183	+75	+77	0	0	105	81	852
12	20	1	120	+85	+178	+75	+77	0	0	108	78	1062
13	20	2	30	+77	+208	+25	+77	0	0	104	94	642
14	20	2	60	+83	+188	+75	+64	0	0	101	86	780
15	20	2	90	+88	+173	+75	+69	0	0	111	91	966
16	20	2	120	+88	+178	+75	+64	0	0	110	83	1146
17	20	3	30	+80	+203	+38	+67	0	0	103	99	690
18	20	3	60	+84	+190	+75	+74	0	0	108	105	822
19	20	3	90	+89	+183	+75	+74	0	0	104	100	1056
20	20	3	120	+88	+178	**+88**	+64	0	0	114	103	1200
Average				*+86*	*+184*	*+71*	*+69*	*0*	*0*	*108*	*97*	*980*
21	100	0	30	+69	+213	+38	+72	0	1	415	0	474
22	100	0	60	+76	+203	+63	+64	0	0	488	0	606
23	100	0	90	+79	+208	**+88**	+72	0	0	515	0	768
24	100	0	120	+80	+200	**+88**	+69	0	0	529	0	954
25	100	1	30	+76	+213	+25	+74	0	0	423	80	525
26	100	1	60	+85	+195	+75	+69	0	0	460	78	684
27	100	1	90	+87	+200	+75	+67	0	0	530	79	840
28	100	1	120	+87	+205	+75	**+79**	0	0	532	73	1026
29	100	2	30	+81	+203	+25	+74	0	0	415	88	606
30	100	2	60	+88	+188	+75	+77	0	0	478	94	786
31	100	2	90	+90	+200	+75	+72	0	0	510	92	966
32	100	2	120	+93	+188	+75	+69	0	0	502	88	1140
33	100	3	30	+83	+200	+38	+72	0	0	420	106	660
34	100	3	60	+89	+200	+75	+74	0	0	476	98	804
35	100	3	90	+91	+205	+75	+69	0	0	500	98	942
36	100	3	120	**+94**	+190	+75	+72	0	0	547	106	1176
Average				*+90*	*+197*	*+69*	*+71*	*0*	*0*	*493*	*98*	*948*

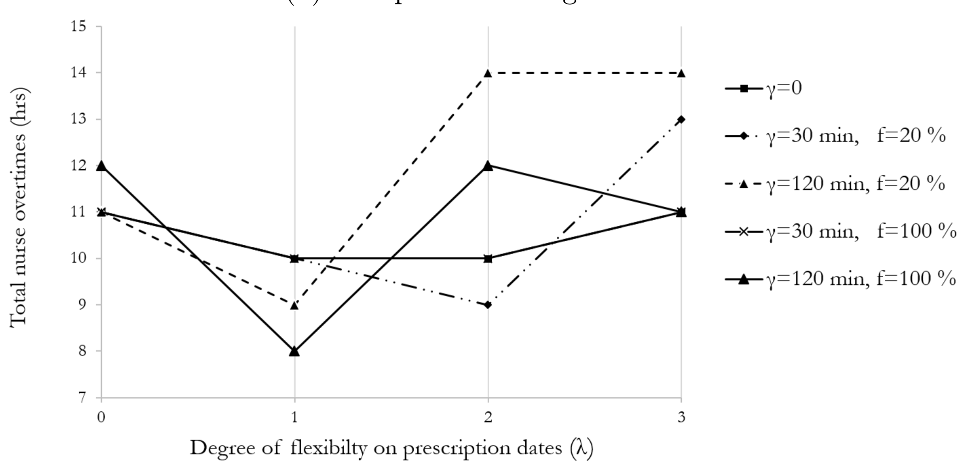
to 20% produces even better results. Thus, depending on the priorities they give to different objectives, decision makers must consider the trade-offs when setting the levels of controllable parameters and the degree of flexibility to obtain the desired results.



(a) Nurse assignments



(b) Non-preferred assignments



(c) Total nurse overtime

Figure 2.3: Sensitivity analysis of the effect of different controllable parameters on performance

2.5.3 Handling last-minute changes

In this sub-section, the focus is on analyzing the capacity of the adaptive procedure to handle unexpected changes. To perform this analysis, two sources of uncertainty were modelled: treatment modifications and last-minute nurse absences. Specifically, it is assumed that after consultations on the day before an appointment date, the oncologist modifies the patient's treatment plan by adding either one hour or two hours to the duration of the treatment with probability p_1 and p_2, respectively, or cancels the treatment with probability p_3. Thus, treatments remain unchanged with probability $1 - p_1 - p_2 - p_3$. For nurse absences, each nurse is absent on a given day with probability q_n. Scenarios for last-minute changes were generated using the procedure described in Algorithm 1. Schedules from the actual SCC system were used, and the rescheduling model was executed for varying scenarios generated for each day. The following values were used: $\gamma = 0$ and $\gamma = 60$ minutes, f=100%, and based on the data collected $p_1 = 5\%$, $p_2 = 5\%$, $p_3 = 10\%$, and $q_n = 5\%$, for each $n \in N_d$.

Table 2.8 presents the daily results of the SCC scheduling system. Recall that N_{pri} and N_{sec} denote the number of patients assigned to their primary and secondary nurses, respectively, and O_{nrs} shows the total nurse overtime. As can be seen in Table 2.8, on average, eight of 40 patients were assigned to their primary nurse, and two patients to their secondary nurse. The number of non-preferred appointments was zero, and the average total daily overtime was 78 minutes for the nine nurses on duty.

Using Algorithm 1, 50 scenarios per day were generated to evaluate the capacity of the rescheduling model to handle unexpected changes. The average daily number of each type of change in the generated scenarios is summarized in Table 2.9. On average, seven patients each day had the duration of their treatment modified, four appointments per day were cancelled, and 0.4 nurses (equivalent to eight nurses in 20 working days) were considered to be absent. It should be noted that, when a nurse is

Algorithm 1: Procedure for generating scenarios for last-minute changes

```
Result: A scenario for day d with P_d patients and N_d nurses on duty
Initialization, retrieve the complete schedule of day d ;
for p=1 to P_d do
    Generate a uniform random number u in [0,1];
    if u < p_1 then
        set i_p:=i_p+1 ;
    else
        if u < p_1 + p_2 then
            set i_p:=i_p+2 ;
        else
            if u < p_1 + p_2 + p_3 then
                Remove patient p from the schedule of day d ;
            else
                set i_p:=i_p ;
            end
        end
    end
end
for n=1 to N_d do
    Generate a uniform random number u in [0,1];
    if u < q_n then
        Remove nurse n from the schedule of day d;
        Reassign the chairs of the absent nurse to the available nurses;
        Update the availability of the reassigned chairs
    end
end
```

Table 2.8: Daily results of the SCC scheduling system

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	*Average*
Patient	49	47	36	24	45	46	44	45	22	42	35	47	48	27	41	48	45	41	26	43	40
Nurse	9	9	8	6	7	9	7	8	6	8	9	7	8	6	8	9	9	9	6	8	8
N_{pri}	5	11	8	4	4	10	9	16	4	10	3	7	13	5	5	11	11	20	2	9	***8***
N_{sec}	4	1	3	1	3	3	1	1	1	1	3	1	3	0	2	3	0	2	1	3	***2***
N_{prf}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	***0***
O_{nrs}(min)	30	0	0	0	150	30	60	60	150	120	120	120	90	90	180	120	60	30	0	150	***78***

absent, nurse-chair assignments are updated by randomly assigning the absent nurse's chairs to the other nurses on duty. Then, chair availability is updated based on the revised nurse-chair assignments.

Table 2.10 summarizes the results obtained by executing the proposed rescheduling model on the generated last-minute changes in two cases. The left-hand side of the

Table 2.9: Summary of changes generated by scenarios

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	*Average*
Modified	10	8	7	5	8	9	8	9	4	7	7	8	9	4	8	8	9	8	4	8	7
Canceled	5	4	3	2	4	4	4	4	2	3	3	4	4	2	4	4	4	4	2	4	4
Absent	0.4	0.3	0.5	0.2	0.4	0.4	0.3	0.4	0.2	0.3	0.5	0.4	0.3	0.3	0.4	0.6	0.4	0.3	0.2	0.3	0.4

table assumes $\gamma = 0$, which means that the start time of an appointment must not change and that only the assigned chair and nurse can change. On the right-hand side, it is assumed that the duration of an appointment can change by up to 60 minutes ($\gamma = 60$).

Table 2.10: Results of resolving the last-minute changes in the generated scenarios

Day	*Rescheduling with $\gamma = 0$ min*							*Rescheduling with $\gamma = 60$ min*						
	N_{pri}	N_{sec}	N_{prf}	O_{nrs} (min)	N_{buf}	N_{ch}	*time(s)*	N_{pri}	N_{sec}	N_{prf}	O_{nrs} (min)	N_{buf}	N_{ch}	*time(s)*
1	7	6	0	28	3	0	10	12	11	0	20	0	32	18
2	14	3	0	55	1	0	9	16	5	0	27	0	26	13
3	7	4	0	13	1	0	4	10	6	0	15	0	24	5
4	3	2	0	11	0	0	1	5	2	0	4	0	16	1
5	5	4	0	59	8	0	9	9	5	0	79	5	24	27
6	5	8	0	13	1	0	7	9	9	0	4	0	29	14
7	6	8	0	19	1	0	6	9	9	0	5	0	27	12
8	8	2	0	30	6	0	8	10	4	0	79	0	30	12
9	3	2	0	24	1	0	1	4	2	0	6	0	16	1
10	9	1	0	28	7	0	8	10	3	0	69	2	27	14
11	3	6	0	9	1	0	4	7	8	0	0	0	22	7
12	8	5	0	11	5	0	10	14	5	0	24	1	29	23
13	15	3	0	26	5	0	9	21	2	0	15	0	31	17
14	4	2	0	119	1	0	3	5	2	0	109	0	19	5
15	7	3	0	47	4	0	7	10	5	0	57	1	29	23
16	12	4	0	204	1	0	9	18	7	0	140	0	31	16
17	10	6	0	124	1	0	7	15	6	0	27	0	29	11
18	11	5	0	13	2	0	5	17	6	0	2	0	28	6
19	6	1	0	10	0	0	2	7	2	0	3	0	18	3
20	9	3	0	5	2	0	7	13	5	0	5	0	29	8
Average	***8***	***4***	***0***	***42***	3	0	6	***11***	***5***	***0***	***35***	0.5	26	12

According to the results presented in Table 2.10, the proposed adaptive procedure reacted efficiently to the generated changes. When $\gamma = 0$, after processing the modifications described in Table 2.9, the procedure can rebuild the schedule with a level of performance similar to that of the system actually used at the SCC in terms of nurse-patient assignments and patient preference ($N_{pri} = 8$, $N_{sec} = 4$ and N_{prf} $=0$) while decreasing average nurse overtime, i.e., $O_{nrs} = 42$ minutes. However, three appointments daily were cancelled to accommodate these changes. As well, slight modifications to already booked appointments (one hour at most) can help resolve unexpected changes more efficiently. In this case, the proposed procedure not only schedules appointments for all patients ($N_{buf} \approx 0$) but also achieves better nurse-patient assignments and decreases total nurse overtime. These improvements are achieved in a reasonable amount of time and with no additional resources.

2.6 Conclusion

This paper studied a chemotherapy scheduling problem that examined several critical and challenging issues arising in the Segal Cancer Centre (SCC) in Montreal, Canada. The first challenge came from the fact that appointment requests arrive dynamically (as an online waiting list that contains multiple requests and is continually updated), and the exact number of appointments is not known in advance. Depending on the regimen prescribed for a patient during a consultation, an oncologist may request a series of appointments with rest periods of specific lengths between treatments and may also request a specific set of chairs where the patient will receive treatment. Furthermore, a patient may have preferences for the start time of treatments, and a nurse is fully occupied with one patient during the setup phase of treatment. To tackle these complexities, a flexible and adaptive scheduling procedure was proposed that schedules incoming appointment requests, and reschedules these on a daily basis when either new information regarding the request is received or an unexpected last-minute change occurs. Both the scheduling problem and the rescheduling problem were formulated as integer programs. Various controllable and flexible parameters such as deviating from the dates prescribed by the oncologist by a pre-determined threshold for each patient, changing the start time of patients already booked, and specifying the maximum number of appointments that could be moved within the schedule were included in the proposed procedure. These parameters allowed a sufficient degree of flexibility in accommodating incoming requests along with last-minute changes.

Computational experiments were carried out to evaluate the performance of the procedure using real data gathered from the SCC. The proposed procedure achieved consistently better results for all objective functions compared to those of the scheduling system in use at the SCC. Moreover, several analyses were conducted on the results to evaluate the effect of different levels of flexibility and also to assess the performance of the proposed procedure in dealing with last-minute changes. Adopting the proposed procedure would allow the SCC to provide better patient care and utilize available resources more efficiently.

An interesting extension of this study could be the design of an accurate prediction tool that estimates the combination of patients and their characteristics in advance. These characteristics might include the treatment plan, drug types (which determine infusion duration), patient acuity level, required nursing skills (assigning the primary nurse whose skills best match the acuity level of each patient), patient preferences, and probability of no-show and cancellation. Similar to [42], forecasting the combination of future treatments of different durations (as a result of different drug types) can help to create an accurate template (prior to the arrival of an appointment request) that can be used when scheduling incoming requests. The pharmacy stage could also become a direction for future research: the types of drugs that must be prepared on the same day as treatment, together with assumptions about uncertainty in the duration of the drug preparation stage, could be studied.

Chapter 3

Two-stage robust optimization for perishable inventory problems with order modification

P. Hooshangi-Tabrizi, H. Hashemi-Doulabi, I. Contreras, N. Bhuiyan. “Two-stage RO for perishable inventory problems with order modification". submitted to *European Journal of Operational Research*, January 2020.

Abstract

In this paper, we study an inventory management problem that integrates perishability, demand uncertainty, and order modification decisions. We focus on perishable products with a fixed shelf-life where demands belong to an uncertainty set. We formulate the problem as a two-stage robust integer optimization model that minimizes the sum of ordering, purchasing, holding, shortage, wastage, and modification costs in the worst-case scenario. We develop an exact column-and-row generation algorithm to solve the problem. We perform extensive computational experiments to evaluate the efficiency of our algorithm and to carry out various sensitivity analyses. Further-

more, we compare solutions obtained from our two-stage robust model with those of the deterministic model and a commensurable stochastic model in both risk neutral and worst-case settings.

3.1 Introduction

Inventory control can be defined as a system to monitor levels of inventory continuously or periodically. This system decides on optimal replenishment cycles, order quantities and inventory allocations in order to satisfy demand. In the literature of inventory control, most of the work assume unlimited lifespans for products. However, in practice, there are many products that deteriorate over time and become unfit for consumption after their shelf-life. It is thus crucial to make optimal ordering, production and allocation decisions while reducing waste as much as possible. According to Gooch and Abdel [37], nearly 31 billion dollars worth of food is wasted each year in Canada. Retailers are responsible for 15% of such waste. Furthermore, the 2018 report of Canadian Blood Services showed that 13% of all platelet units processed for transfusion were outdated ([98]). In addition to food and blood, products like pharmaceuticals, cut flowers and some chemicals are other examples of perishables with a limited shelf-life ([32, 103]).

Perishable inventory management (PIM) problems concern the design of inventory plans by determining the proper times to place orders, the size of orders, and allocating products with different shelf-lives to demand. Normally in PIM problems, demand is satisfied either by 1) the recently received order that includes the most fresh items, 2) stocks carried from previous periods, or 3) products with any age that are procured urgently. A common objective in PIM problems is to minimize the sum of the fixed setup cost for placing orders, purchase cost, holding cost for storing products during a period, shortage cost in case of stock-outs, and wastage cost for products reaching their shelf-life. The shortage cost can be either considered as the lost sale, where demand is not covered (in retail stores for example), or the emergency procurement cost, where demand is finally met by outsourcing at a higher cost (in blood service centers for example). According to the nature of perishability, inventory models can be categorized into two main categories. In the first category, products are assumed to have a fixed shelf-life or constant rate of deterioration ([12, 13, 23, 62]). In the second category, the exact lifetime of perishables cannot be determined in advance and a randomly distributed shelf-life is usually considered ([60, 61, 63, 68]). Among the first category, there is a subclass in which although the shelf-life is fixed, we do not know the exact lifetime in advance while making orders ([88]). In these cases, we receive information on their exact lifetime only once the ordered items arrive. In this category, the shelf-life is fixed but different for each product.

Demand uncertainty is another important factor that arises in most real-life applications. It makes inventory decisions much more difficult, especially in the case of perishable products. Overestimation of demand causes significantly more wastage or outdated costs while its underestimation leads to a higher shortage cost. In the literature, various methodologies are used to model PIM problems with demand uncertainty. Some works propose simulation models to build easy rules for the ordering policy of stochastic PIM problems ([45, 46, 57, 107]). Moreover, markov decision process is another approach to formulate PIM problems ([31, 43, 44, 69]), where approximation and heuristic algorithms are commonly used to solve the proposed models. Stochastic programming is another prevalent approach to model PIM problems with stochastic demand ([27, 29, 39, 47, 88]). In the first-stage of stochastic models for PIM problems, the decision maker determines the ordering periods, order sizes, safety stock, and target levels. In the second stage, he decides on recourse actions including the allocation of on-hand inventory with different ages to demand as well as the amount of holding, shortage and wastage. To obtain statistically reliable solutions, some works have solved stochastic programming models using a sample average approximation framework (e.g., [27]). Decomposition-based integer programming algorithms are also common techniques for solving the resulting large-scale stochastic PIM problems ([93]). Another modeling approach is to use robust optimization techniques to minimize the total ordering, holding, shortage, and waste cost in the worst-case scenario ([51, 58, 94]).

The possibility of modifying an order that was initially determined at the beginning of the planning horizon is another important aspect in PIM problems. This aspect has been rarely addressed in the literature. There are several papers that have only studied the resource flexibility and alternative sources in multi-period inventory problems ([35, 55, 56, 85, 97, 109]). In these works, it is assumed that different sources with various fixed/variable costs, and limited capacities are available to cover demand. Other works consider a dual model of replenishment, one regular ordering at the beginning of the planning horizon for each period and one optional expedited ordering between consecutive periods in the case of shortage ([64, 110]). Usually, the later type of order is more expensive due to its short notice for suppliers. The possibility of modifying an order before receiving it can decrease the risk of shortage and expensive expedited orders. Furthermore, in the presence of order modifications, the risk of wastage can reduce as a result of the reduction in initial purchases at the beginning of the planning horizon.

Potential applications for PIM problems with the above three features, i.e., perishability with different fixed shelf-life, demand uncertainty, and order modifications, can arise in several settings. One concrete example is the inventory management of blood products such as platelet. The platelet units received by hospitals from blood banks usually have different shelf-lives. This is because of different ways to produce platelet, various blood collection intervals, and also a variety of laboratory conditions for processing blood. Because of demand uncertainty, it is always difficult to estimate the exact amount of future blood demand. On the other hand, fixing an ordering plan with a supplier such as a blood bank in advance can help the supplier to better plan the collection and processing of the future platelet demand. Therefore, by securing initial orders at the beginning and modifying the amount of these fixed orders before receiving them, both hospitals and suppliers can benefit from cost reductions through a well-established collaboration. Another potential application, where PIM problems can be used, is the inventory management of fresh produce such as fruits, vegetables, meats and dairy products in retail stores. Different harvesting intervals, various storage conditions and warehousing policies impact their shelf-life. Because of short and different shelf-lives of received fresh food in stores, strategies for reviewing and replenishing theses products need to be designed properly to achieve minimum wastage. The possibility of modifying the quantity of orders that are fixed previously between retail stores and fresh produce suppliers can reduce the risk of food wastage that happens because of demand overestimation.

Optimal inventory policies which are set on a particular distribution may perform poorly against another demand distribution even if the mean and variance are equally tuned ([18]). Moreover, in multi-period planning problems with demand uncertainty, it is a difficult task to identify representative stochastic scenarios when the problem is formulated as a two-stage model ([38]). Robust optimization is an appropriate method to deal with these difficulties. Obtaining a robust solution for an inventory management problem can even be more critical in the case of perishable products in real-life situations. In this case, even though a significant part of information is not easily available, robust optimization is tailored to the information at hand that leads to computationally tractable formulations. There are many works in the literature of inventory problems that consider demand belongs to an uncertainty set and apply a robust optimization approach ([9, 16, 18, 86, 87, 94, 96, 100, 104]). These works are interesting and practical in the sense that they provide stable solutions in the absence of probability distributions for uncertain parameters. However, they ignore several aspects such as the possibility of modifying future orders that are made in earlier periods, the ordering cost, and the shelf-life of products in case of perishability.

In this paper, we address these aspects in a PIM problem with 1) perishable products of different shelf-lives, 2) demand uncertainty, and 3) the possibility of modifying orders that are fixed at the beginning of the planning horizon. We formulate this inventory problem as a two-stage robust integer program and use a periodic review system in which the set of ordering periods and the size of the associated orders are fixed in the first stage. In the second stage, after the realization of uncertainty, we decide on the modification of orders fixed in the first-stage, and the amount of holding, shortage, and outdated product for each period. We refer the problem under study as the *perishable inventory management with modifications* (PIMM) problem. The main contributions of this work is to introduce a two-stage robust optimization model for a perishable inventory problem with several features such as demand uncertainty, perishable products with different fixed shelf-life, and the possibility of modifying future orders fixed in earlier periods. We also develop an exact column-and-row generation algorithm to solve the proposed robust optimization model. The proposed algorithm is based on exploiting the structure of two-stage robust optimization models and the Karush-Kuhn-Tucker (KKT) optimality conditions of linear programs. We perform extensive computational experiments to evaluate the efficiency, robustness, and limitations of the proposed column-and-row generation algorithm. We also analyze the impact of different parameters on the performance of the algorithm and on the quality of solutions obtained.

The rest of the paper is organized as follows. In Section 3.2 we provide a formal definition of the problem and propose a two-stage robust formulation for it. In Section 3.3 we present a column-and-row generation algorithm to solve the PIMM problem. Computational experiments and sensitivity analyses are carried out in Section 3.4.

Finally, we provide conclusions and some future research directions in Section 3.5.

3.2 Problem description and formulation

We consider a periodic review inventory system where products have a fixed shelf-life equal to M periods. This means that once the ordered products arrive, they expire after M periods. In this case, they are considered as outdated products and must be discarded. Let $\mathcal{T}$ and $\mathcal{P}$ represent the set of days in the planning horizon and the set of all ordering patterns with different review intervals denoted by R, respectively. For each $p \in \mathcal{P}$, a_{tp} accounts for a binary parameter indicating whether or not we place an order at period $t \in \mathcal{T}$ in the case that pattern p is chosen. We consider the following parameters:

- c^1: the fixed cost of each order,
- c^2: the variable regular unit purchase cost,
- c^3: the unit holding cost in each period,
- c^4: the unit shortage cost or equivalently the emergency procurement,
- c^5: the unit cost of an outdated product,
- c^6: the unit modification cost incurred when ordering extra products.

Furthermore, we denote the minimum and maximum purchase amounts by F^{min} and F^{max}, respectively. There is also a limit γ, defined as a percentage of F^{max}, on the maximum amount of extra purchases allowed to make when modifying the placed orders.

Regarding demand uncertainty, parameters $\bar{d}_t$ and $\hat{d}_t$ represent the mean and deviation of the demand in period t, respectively. At the beginning of the planning horizon, the decision maker fixes the ordering pattern and the quantities of regular purchases for each period $t \in \mathcal{T}$. We consider the possibility of adding extra units of the product to a previously fixed order by paying the unit modification cost c^6. We must fulfill the demand of each period by a combination of regular purchases, the available stock, the extra purchases made during the planning horizon, and an emergency procurement. We also take into account that the ordered products have various shelf-lives. In particular, let λ_m denote the percentage of arriving products with a shelf-life of m days, where $1 \leq m \leq M$. In this paper, we consider the following assumptions:

- Mixing products with different ages is permitted to cover demand,
- The shelf-life of products received in the same period are deterministic but they may be different,
- In the case order modification, there is a predetermined maximum limit for extra purchases in each period,
- In the case of emergency procurement, there is always enough inventory in the supplier's side to avoid shortages via extra purchases.

We consider that demand uncertainty can be represented as:

$$
\begin{aligned}
&d^t(\xi) = \bar{d}_t + \hat{d}_t \xi_t && t \in \mathcal{T}, \\
&\Xi(\Gamma) := \left\{ \xi \in \mathbb{R}^{|\mathcal{T}|} \middle| -1 \leq \xi_t \leq 1, \sum_{t \in \mathcal{T}} |\xi_t| \leq \Gamma \right\},
\end{aligned}
$$

where $\Xi(\Gamma)$ is the budgeted uncertainty set with parameter Γ denoting the maximum number of periods for which the demand can take the maximum possible value.

Regarding the first-stage decision variables, we use w_p as a binary variable equal to 1 if and only if the order pattern $p \in \mathcal{P}$ is selected. Moreover, y_t is another binary variable determining whether or not we make an order at period $t \in \mathcal{T}$. We also use x_t as a continuous variable to denote the initial purchased quantity at period t. In the second stage, we define $I_t^m(\xi)$ as a continuous variable representing the inventory level of products with a shelf-life of m days at the beginning of period t in scenario ξ. We also define continuous variables $s_t(\xi)$ and $o_t(\xi)$ as the amounts of emergency procurement (shortage) and outdated products, respectively, at the end of period t in scenario ξ. Furthermore, we define $u_t^m(\xi)$ as a continuous variable equal to the amount of the demand in period t that is fulfilled using items with m days of shelf-life in scenario ξ. Finally, $x'_t(\xi)$ is a continuous variable denoting the quantity of extra purchase made in the second-stage in period t under realization ξ.

Figure 3.1 shows the network of inventory balance in the PIMM problem. Any arc between node 0 and any other node t (representing period t) implies that an order is received at the beginning of such period. In the cases that there is no arc between node 0 and a period node, it means that no order is received in that period. For each node t, flow conservation conditions need to be satisfied. At each node, inputs are the inventory from previous period (I_t), regular purchase (x_t), extra purchase (x'_t), and emergency procurement (s_t). Moreover, the output of each node consists of the outdated amount (o_t), realized demand (d_t), and the amount of inventory kept for next period (I_{t+1}).

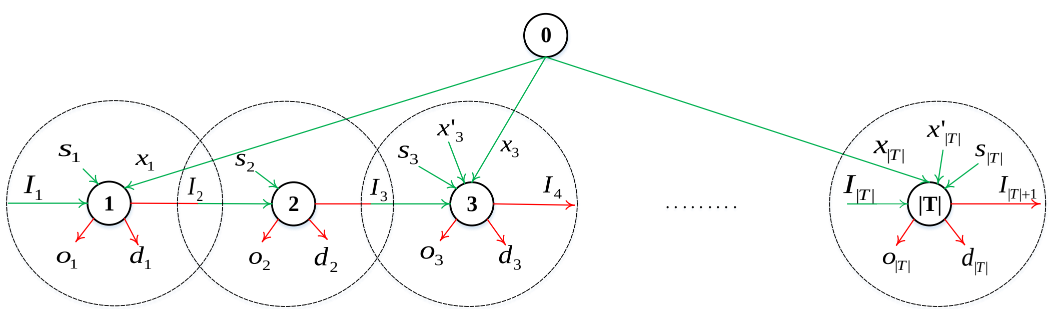


Figure 3.1: Network of the perishable inventory management problem.

The PIMM problem can be stated as the following two-stage robust mixed integer program (TSR-MIP):

$$\text{minimize}\Big(\sum_{t\in\mathcal{T}}(c_t^1 y_t + c_t^2 x_t) + \sup_{\xi\in\Xi(\Gamma)} Q(w,y,x,\xi)\Big) \tag{3.1}$$

subject to

$$\sum_{p\in\mathcal{P}} w_p = 1 \tag{3.2}$$

$$\sum_{p\in\mathcal{P}} a_{tp} w_p = y_t \qquad t\in\mathcal{T} \tag{3.3}$$

$$F^{min} y_t \leq x_t \leq F^{max} y_t \qquad t\in\mathcal{T} \tag{3.4}$$

$$w\in\{0,1\}^{|\mathcal{P}|} y\in\{0,1\}^{|\mathcal{T}|} \tag{3.5}$$

$$x_t \geq 0 \qquad t\in\mathcal{T}, \tag{3.6}$$

where,

$$Q(w,y,x,\xi) = \text{minimize} \ \sum_{t\in\mathcal{T}}\Big(\sum_{m=1}^{M} c_t^3 I_{t+1}^m(\xi) + c_t^4 s_t(\xi) + c_t^5 o_t(\xi) + c_t^6 x_t'(\xi)\Big) \tag{3.7}$$

subject to

$$x_t'(\xi) \leq \gamma F^{max} y_t \qquad t\geq 2 \tag{3.8}$$

$$I_1^m(\xi) = i_m + \lambda_m(x_1 + x_1'(\xi)) \qquad 1\leq m\leq M \tag{3.9}$$

$$I_t^m(\xi) = I_{t-1}^{m+1}(\xi) - u_{t-1}^{m+1}(\xi) + \lambda_m(x_t + x_t'(\xi)) \qquad t\geq 2, m\leq M-1 \tag{3.10}$$

$$I_{|\mathcal{T}|+1}^m(\xi) = I_{|\mathcal{T}|}^{m+1}(\xi) - u_{|\mathcal{T}|}^{(m+1)}(\xi) \qquad m\leq M-1 \tag{3.11}$$

$$I_t^M(\xi) = \lambda_M(x_t + x_t'(\xi)) \qquad t\geq 2 \tag{3.12}$$

$$u_t^m(\xi) \leq I_t^m(\xi) \qquad t\in\mathcal{T}, m\geq 2 \tag{3.13}$$

$$x_1'(\xi) = 0 \tag{3.14}$$

$$o_t(\xi) = I_t^1(\xi) - u_t^1(\xi) \qquad t\in\mathcal{T} \tag{3.15}$$

$$s_t(\xi) + \sum_{m=1}^{M} u_t^m(\xi) = \bar{d}_t + \hat{d}_t \xi_t \qquad t \in \mathcal{T} \quad (3.16)$$

$$x_t'(\xi) \geq 0,\, s_t(\xi) \geq 0,\, o_t(\xi) \geq 0 \qquad t \in \mathcal{T} \quad (3.17)$$

$$I_t^m(\xi) \geq 0 \qquad 1 \leq t \leq |\mathcal{T}| + 1, 1 \leq m \leq M \quad (3.18)$$

$$u_t^m(\xi) \geq 0 \qquad t \in \mathcal{T}, 1 \leq m \leq M. \quad (3.19)$$

The first term in the objective function (3.1) minimizes the sum of fixed and variable regular ordering costs over the planning horizon. The second term represents the second-stage cost in the worst-case scenario. Constraints (3.2)-(3.3) indicate that only one ordering pattern must be selected and the ordering periods are determined with respect to such pattern. Constraints (3.4) ensure that order quantities take values between the lower and upper bounds for the regular purchases. If there is no order placed for a specific period, the purchase quantity for that period is equal to zero. Constraints (3.5) are the integrality conditions on w and y variables. Constraints (3.6) imply that x variables are continuous and non-negative. From now on, we denote as $\mathcal{X}$ the set of solutions associated with constraints (3.2)-(3.6).

The second-stage objective function (3.7) represents the minimization of holding, shortage, wastage and modification costs for uncertainty scenario ξ. Constraints (3.8) set the maximum limit on the extra purchase that can be made for each period. Constraints (3.9)-(3.12) are the inventory balance constraints for products with different ages. Note that i_m represents the initial inventory of shelf life m at the beginning of the planning horizon. For each age of the inventory at each period, the stock kept from the previous period is added to the regular and extra purchases of that specific age. Then the inventory level for that specific age in the next period is calculated by subtracting the amount of covered demand (u) from the sum of available stock and the extra and regular purchases. Constraints (3.13) guarantee that the amount of covered demand is less than the available stock level. Constraints (3.14) ensure that no modification is required in the first period of the planning horizon. Constraints (3.15) imply that there should be no order placed at the end of the planning horizon (at the beginning of period $|\mathcal{T}|+1$). Constraints (3.16) and (3.17) calculate the amounts of outdated and shortage (emergency procurement) for each time period. Constraints (3.18) and (3.19) defines the non-negativity conditions for all second-stage variables. Finally, we denote as $\mathcal{Y}$ the set of solutions associated with constraints (3.8)-(3.19).

3.3 An Exact solution algorithm for the PIMM problem

In this section, using the structure of the TSR-MIP and a property of the uncertainty set, we first present an equivalent representation of the PIMM problem that is in the format of a min-max problem. We then present an exact column-and-row generation algorithm that relies on a mixed-integer linear programming reformulation of the min-max problem.

3.3.1 Equivalent representation of the TSR-MIP

The TSR-MIP presented in the previous section has both the *relatively complete recourse* and *fixed recourse* properties. The first property implies that for every first-stage solution in $\mathcal{X}$, there is always a feasible second-stage solution in $\mathcal{Y}$, i.e., $\forall(w, y, x) \in \mathcal{X}$, $\forall\xi \in \Xi(\Gamma)$, $\exists(x', I, s, o, u) \in \mathcal{Y}$. The second property implies that the coefficient of the recourse variables, i.e., $(x', I, s, o, u) \in \mathcal{Y}$, is not uncertain. The TSR-MIP holds this property as the uncertainty appears only in demand that is not multiplied by any of the recourse variables. Furthermore, as the budgeted uncertainty $\Xi(\Gamma)$ is a bounded polyhedron, described by a finite number of linear constraints, it has a finite number of extreme points. We denote as $\mathcal{K} = \{\bar{\xi}^1, \bar{\xi}^2, ..., \bar{\xi}^{|\mathcal{K}|}\}$ the set of all extreme points of $\Xi(\Gamma)$. We next present the budgeted uncertainty set as the convex hull of its extreme points, i.e., $\Xi(\Gamma) := \text{Conv}(\mathcal{K})$.

Lemma 3.3.1. *(Horst [50], Rockafellar [90]) Assume that the uncertainty set Ξ is given as the convex hull of a finite set, $\mathcal{K} = \{\bar{\xi}^1, \bar{\xi}^2, ..., \bar{\xi}^{|\mathcal{K}|}\}$, i.e., $\Xi(\Gamma) := Conv(\mathcal{K})$. The optimal value of $\underset{\xi\in\Xi}{maximize}\, Q(\xi)$ is equal to $\underset{k\in\{1,..,|\mathcal{K}|\}}{max} Q(\bar{\xi}^k)$ if $Q(\xi)$ is a convex function over Ξ.*

Theorem 3.3.1. *The TSR-MIP is equivalent to the following problem:*

$(\textit{Min-Max-MIP})$:

$$\underset{(w,y,x),\left\{(x'^k,I^k,s^k,o^k,u^k)\right\}_{k=1}^{|\mathcal{K}|}}{minimize} \quad \underset{k\in\mathcal{K}}{maximize} \sum_{t\in\mathcal{T}} \Big(c_t^1 y_t + c_t^2 x_t + \sum_{m=1}^{M} c_t^3 I_{t+1}^{mk} + c_t^4 s_t^k + c_t^5 o_t^k + c_t^6 x_t'^k\Big)$$

$$\textit{subject to}$$

$$(w, y, x) \in \mathcal{X}$$

$$\left\{(x'^k, I^k, s^k, o^k, u^k)\right\}_{k=1}^{|\mathcal{K}|} \in \mathcal{Y} \qquad .$$

Proof. For every fixed $(\hat{w}, \hat{y}, \hat{x}) \in \mathcal{X}$ and $\xi \in \Xi$, the second stage problem $(Q(\hat{w}, \hat{y}, \hat{x}, \xi))$ can be presented as follows:

$$\text{minimize} \sum_{t\in\mathcal{T}} \Big(\sum_{m=1}^{M} c_t^3 I_{t+1}^m(\xi) + c_t^4 s_t(\xi) + c_t^5 o_t(\xi) + c_t^6 x_t'(\xi)\Big)$$

subject to

$$a_i^{3T} I + a_i^{4T} s + a_i^{5T} o + a_i^{6T} x' + a_i^{7T} u = b_i(\xi) - a_i^{0T}\hat{w} - a_i^{1T}\hat{y} - a_i^{2T}\hat{x} \qquad \forall i = 1, ..., L$$

$$a_j'^{3T} I + a_j'^{4T} s + a_j'^{5T} o + a_j'^{6T} x' + a_j'^{7T} u \geq b_j'(\xi) - a_j'^{0T}\hat{w} - a_j'^{1T}\hat{y} - a_j'^{2T}\hat{x} \;\; \forall j = 1, ..., L'$$

$$I, s, o, x', u \geq 0.$$

In the above model, L equality and L' inequality constraints along with non-negativity conditions of variables represent $\mathcal{Y}$ feasible space, and $b(\xi)$ is an affine function of ξ uncertainty parameter. Using the fixed recourse property of the TSR-MIP, we know that the above problem and its dual are always feasible. Therefore, we can formulate the dual problem of the second-stage problem, i.e., $Q(\hat{w},\hat{y},\hat{x},\xi)$, as follows:

$$\text{maximize} \sum_{i=1}^{L} \theta_i \Big(b_i(\xi) - a_i^{0T}\hat{w} - a_i^{1T}\hat{y} - a_i^{2T}\hat{x} \Big) + \sum_{j=1}^{L'} \theta'_j \Big(b'_j(\xi) - a_j^{'0T}\hat{w} - a_j^{'1T}\hat{y} - a_j^{'2T}\hat{x} \Big)$$

subject to

$$\begin{aligned}
&\sum_{i=1}^{L} \theta_i a_{ik}^3 + \sum_{j=1}^{L'} \theta'_j a_{jk}^{'3} \le c^3 && \forall k \in K(I)\\
&\sum_{i=1}^{L} \theta_i a_{ik}^4 + \sum_{j=1}^{L'} \theta'_j a_{jk}^{'4} \le c^4 && \forall k \in K(s)\\
&\sum_{i=1}^{L} \theta_i a_{ik}^5 + \sum_{j=1}^{L'} \theta'_j a_{jk}^{'5} \le c^5 && \forall k \in K(o)\\
&\sum_{i=1}^{L} \theta_i a_{ik}^6 + \sum_{j=1}^{L'} \theta'_j a_{jk}^{'6} \le c^6 && \forall k \in K(x')\\
&\sum_{i=1}^{L} \theta_i a_{ik}^7 + \sum_{j=1}^{L'} \theta'_j a_{jk}^{'7} \le 0 && \forall k \in K(u)\\
&\theta'_j \ge 0 && j = 1, ..., L'.
\end{aligned}$$

In the above dual formulation, $\theta \in \mathbb{R}^L$, $\theta' \in \mathbb{R}^{L'}$ are dual variables associated to the equality and inequality constraints, respectively, and $K(x)$ denotes the number of x variables. According to this dual formulation, since $Q(\hat{w},\hat{y},\hat{x},\xi)$ is the maximum of a set of affine functions, it is a convex function in terms of ξ. Therefore, we can use the above lemma to obtain:

$$\underset{\xi\in\Xi}{\text{maximize}}\, Q(\hat{w},\hat{y},\hat{x},\xi) = \max_{\bar{\xi}\in\mathcal{K}} Q(\hat{w},\hat{y},\hat{x},\bar{\xi}).$$

The above equation implies that, it is enough to maximize the problem with respect to $|\mathcal{K}|$ extreme points to make sure that the obtained solution is robust with respect to any uncertainty parameter taken from Ξ. Therefore in the The TSR-MIP, if we replace $\sup_{\xi \in \Xi(\Gamma)} Q(w, y, x, \xi)$ with $\max_{\bar{\xi} \in \mathcal{K}} Q(w, y, x, \bar{\xi})$, in the equivalent formulation, we will have as many second-stage variables as the number of extreme points, i.e., $\{(x'^k, I^k, s^k, o^k, u^k)\}_{k=1}^{|\mathcal{K}|}$. □

3.3.2 Column-and-row generation algorithm

We use a column-and-row generation (C&RG) algorithm based on the method developed by Zeng and Zhao [108]. They study a two-stage robust optimization problem that satisfies the above-mentioned fixed and relatively complete recourse assumptions. The idea of the C&RG algorithm is to solve the equivalent Min-Max-MIP problem with respect to a subset of the extreme points of the uncertainty set ($\mathcal{K}' \subset \mathcal{K}$) as the total number of extreme points ($|\mathcal{K}|$) can be exponential. The C&RG algorithm starts with a subset of extreme points ($\mathcal{K}'$) of the uncertainty set $\Xi(\Gamma)$ in a master problem (MP). After solving the MP, new extreme points are added iteratively to the initial subset by solving an associated subproblem (SP). This process of alternating between the MP and SP continues until either the algorithm converges to an optimum solution or a pre-defined stopping criterion is met. The steps of our C&RG algorithm are given in Algorithm 2. We recall that $\mathcal{X}$ represents the feasible space of the first-stage variables.

Algorithm 2 is guaranteed to converge in a finite number of iterations because our bounded uncertainty set $\Xi(\Gamma)$ has a finite number of vertices. In practice, the algorithm converges in fewer iterations than the number of all vertices in set Ξ. In the next section, we show how to formulate the corresponding MP and SP in the proposed C&RG algorithm to solve the Min-Max-MIP.

Algorithm 2: Column-and-row generation algorithm

1: Set $\Xi = \varnothing$, $\mathcal{K}' = \varnothing$, $i = 0$, $UB = +\infty$, and $LB = -\infty$.
2: Solve MP to derive an initial first-stage solution $(\hat{w}^0, \hat{y}^0, \hat{x}^0) \in \mathcal{X}$.
3: Update $LB = \sum_{t\in\mathcal{T}} \left(c_t^1 \hat{y}_t^0 + c_t^2 \hat{x}_t^0\right)$.
4: Solve SP to identify the initial worst-case scenario:
$\xi^{*0} = \underset{\xi\in\Xi}{\text{argmax}}\, Q(\hat{w}^0, \hat{y}^0, \hat{x}^0, \xi)$.
5: Update $UB = \min\left\{UB,\ \sum_{t\in\mathcal{T}} \left(c_t^1 \hat{y}_t^0 + c_t^2 \hat{x}_t^0\right) + Q(\hat{w}^0, \hat{y}^0, \hat{x}^0, \xi^{*0})\right\}$.
6: Construct $\Xi := \Xi \cup \{\xi^{*0}\}$, $\mathcal{K}' := \mathcal{K}' \cup \{k(\xi^{*0})\}$ and set $i = 1$. If $(\frac{UB-LB}{LB}) \leq \epsilon$, stop the algorithm. Otherwise, go to Step 7.
7: Solve MP with Ξ and $\mathcal{K}'$ to obtain $(\hat{w}^i, \hat{y}^i, \hat{x}^i)$
8: Update LB as:

$$LB = \sum_{t\in\mathcal{T}} \left(c_t^1 \hat{y}_t^i + c_t^2 \hat{x}_t^i\right) + \max_{k\in\mathcal{K}'} \left\{ \sum_{t\in\mathcal{T}} \left(\sum_{m=1}^{M} (c_t^3 I_{t+1}^{mk}) + c_t^4 s_t^k + c_t^5 O_t^k + c_t^6 x_t'^k \right) \right\}.$$

9: Solve SP to identify the worst-case scenario:

$$\xi^{*i} = \underset{\xi\in\Xi}{\text{argmax}}\, Q(\hat{w}^i, \hat{y}^i, \hat{x}^i, \xi).$$

10: Calculate the worst-case value of the current solution and update UB:

$$UB = \min\{UB,\ \sum_{t\in\mathcal{T}} \left(c_t^1 \hat{y}_t^i + c_t^2 \hat{x}_t^i\right) + Q(\hat{w}^i, \hat{y}^i, \hat{x}^i, \xi^{*i})\}.$$

11: If $(\frac{UB-LB}{LB}) \leq \epsilon$ the algorithm has converged. Otherwise, $\Xi := \Xi \cup \{\xi^{*i})\}$, $\mathcal{K}' := \mathcal{K}' \cup \{k(\xi^{*i})\}$, $i = i + 1$ and return to Step 7.

3.3.2.1 Master problem of the C&RG algorithm

We formulate the Master Problem (MP) of our C&RG algorithm as the following MIP:

$$\text{minimize} \sum_{t\in\mathcal{T}} \left(c_t^1 y_t + c_t^2 x_t\right) + q \tag{3.20}$$

subject to

$$(w, y, x) \in \mathcal{X}$$

$$q \geq \sum_{t\in\mathcal{T}} \Big(\sum_{m=1}^{M} (c_t^3 I_{t+1}^{mk}) + c_t^4 s_t^k + c_t^5 o_t^k + c_t^6 x_t'^k \Big) \qquad k \in \mathcal{K}' \tag{3.21}$$

$$x_t^{'k} \leq \gamma F^{max} y_t \qquad t \geq 2, k \in \mathcal{K}' \tag{3.22}$$

$$I_1^{mk} = i_m + \lambda_m(x_1 + x_1^{'k}) \qquad 1 \leq m \leq M, k \in \mathcal{K}' \tag{3.23}$$

$$I_t^{mk} = I_{t-1}^{(m+1)k} - u_{t-1}^{(m+1)k} + \lambda_m(x_t + x_t^{'k}) \qquad t \geq 2, m \leq M-1, k \in \mathcal{K}' \tag{3.24}$$

$$I_{|\mathcal{T}|+1}^{mk} = I_{|\mathcal{T}|}^{(m+1)k} - u_{|\mathcal{T}|}^{(m+1)k} \qquad m \leq M-1, k \in \mathcal{K}' \tag{3.25}$$

$$I_t^{Mk} = \lambda_M(x_t + x_t^{'k}) \qquad t \geq 2, k \in \mathcal{K}' \tag{3.26}$$

$$I_t^{mk} \geq u_t^{mk} \qquad t \in \mathcal{T}, m \geq 2, k \in \mathcal{K}' \tag{3.27}$$

$$x_t^{'k} = 0 \qquad k \in \mathcal{K}' \tag{3.28}$$

$$o_t^k = I_t^{1k} - u_t^{1k} \qquad t \in \mathcal{T}, k \in \mathcal{K}' \tag{3.29}$$

$$s_t^k + \sum_{m=1}^{M} u_t^{mk} = \bar{d}_t + \hat{d}_t \xi_t^k \qquad t \in \mathcal{T}, k \in \mathcal{K}' \tag{3.30}$$

$$x_t^{'k}, s_t^k, o_t^k \geq 0 \qquad t \in \mathcal{T}, k \in \mathcal{K}' \tag{3.31}$$

$$I_t^{mk} \geq 0 \qquad 1 \leq t \leq |\mathcal{T}|+1, 1 \leq m \leq M, k \in \mathcal{K}' \tag{3.32}$$

$$u_t^{mk} \geq 0 \qquad t \in \mathcal{T}, 1 \leq m \leq M, k \in \mathcal{K}'. \tag{3.33}$$

Constraints (3.21) represent the worst-case cost in the second stage to be minimized. These constraints imply the optimality cuts are iteratively generated and added to the MP after finding a new extreme point by solving the SP. Constraints (3.22)-(3.33) are equivalent to constraints (3.8)-(3.19), respectively.

3.3.2.2 Subproblem of the C&RG algorithm

To generate an optimality cut, we need to solve the following problem for a given first-stage solution $(\hat{w}, \hat{y}, \hat{x})$ obtained from the MP:

$$\underset{\xi \in \Xi}{\text{maximize}} \ \underset{(x', I, o, s, u) \in \mathcal{Y}}{\text{minimize}} \ Q(\hat{w}, \hat{y}, \hat{x}, \xi).$$

In order to reformulate this problem as a MIP, we write the KKT conditions for the inner minimization problem that results in a single maximization problem with the following structure:

$$\underset{\xi \in \Xi}{\text{maximize}} \;\; Q(\hat{w}, \hat{y}, \hat{x})$$

subject to

$$(x', I, o, s, u) \in \mathcal{Y}$$

$$\theta \in D(\mathcal{Y})$$

$$(b_j(\xi) - a_j^{0T}\hat{w} + a_j^{1T}\hat{y} - a_j^{2T}\hat{x} + a_j^{3T}I - a_j^{4T}s - a_j^{5T}o - a_j^{6T}x' - a_j^{7T}u)\theta_j = 0$$

$$j = 1, ..., (L + L'),$$

where $(x', I, o, s, u) \in \mathcal{Y}$ denotes the primal feasibility conditions, $\theta \in D(\mathcal{Y})$ represents the dual feasibility conditions, and $L + L'$ constraints imply the KKT complementary slackness conditions for both of the equality and inequality constraints of the second-stage problem in the TSR-MIP. The extended version of this model is as follows:

$$\underset{\xi, x', I, o, s, u, \theta}{\text{maximize}} \;\; \sum_{t \in \mathcal{T}} \left(c_t^1 \hat{y}_t + c_t^2 \hat{x}_t + \sum_{m=1}^{M} c_t^3 I_{t+1}^m + c_t^4 s_t + c_t^5 o_t + c_t^6 x'_t \right) \tag{3.34}$$

subject to

$$x'_t \leq \gamma F^{max} \hat{y}_t \qquad t \geq 2 \tag{3.35}$$

$$I_1^m = i_m + \lambda_m(\hat{x}_1 + x'_1) \qquad 1 \leq m \leq M \tag{3.36}$$

$$I_t^m = I_{t-1}^{m+1} - u_{t-1}^{m+1} + \lambda_m(\hat{x}_t + x'_t) \qquad t \geq 2, m \leq M - 1 \tag{3.37}$$

$$I_{|\mathcal{T}|+1}^m = I_{|\mathcal{T}|}^{m+1} - u_{|\mathcal{T}|}^{m+1} \qquad m \leq M - 1 \tag{3.38}$$

$$I_t^M = \lambda_M(\hat{x}_t + x'_t) \qquad t \geq 2 \tag{3.39}$$

$$I_t^m \geq U_t^m \qquad t \in \mathcal{T}, m \geq 2 \tag{3.40}$$

$$x'_1 = 0 \tag{3.41}$$

$$o_t = I_t^1 - u_t^1 \qquad t \in \mathcal{T} \tag{3.42}$$

$$s_t + \sum_{m=1}^{M} u_t^m = \bar{d}_t + \hat{d}_t(\xi_t) \qquad t \in \mathcal{T} \tag{3.43}$$

$$x'_t \geq 0,\ s_t \geq 0,\ o_t \geq 0 \qquad t \in \mathcal{T} \tag{3.44}$$

$$I_t^m \geq 0 \qquad 1 \leq t \leq |\mathcal{T}| + 1, 1 \leq m \leq M \tag{3.45}$$

$$u_t^m \geq 0 \qquad t \in \mathcal{T}, 1 \leq m \leq M \tag{3.46}$$

$$\theta_t^1 - \sum_{m=1}^{M-1} \lambda_m \theta_{tm}^3 - \lambda_M \theta_t^5 + \theta_t^{10} = c_t^6 \qquad t \geq 2 \tag{3.47}$$

$$-\sum_{m=1}^{M} \lambda_m \theta_m^2 + \theta^7 + \theta_1^{10} = 0 \tag{3.48}$$

$$\theta_t^9 + \theta_t^{11} = c_t^4 \qquad t \in \mathcal{T} \tag{3.49}$$

$$\theta_t^8 + \theta_t^{12} = c_t^5 \qquad t \in \mathcal{T} \tag{3.50}$$

$$\theta_{(t+1)(m-1)}^3 - \theta_{tm}^6 + \theta_t^9 + \theta_{tm}^{14} = 0 \qquad 1 \leq t \leq |\mathcal{T}| - 1, m \geq 2 \tag{3.51}$$

$$\theta_{(m-1)}^4 - \theta_{|\mathcal{T}|m}^6 + \theta_{|\mathcal{T}|}^9 + \theta_{|\mathcal{T}|m}^{14} = 0 \qquad m \geq 2 \tag{3.52}$$

$$\theta_t^8 + \theta_t^9 + \theta_{t1}^{14} = 0 \qquad t \in \mathcal{T} \tag{3.53}$$

$$\theta_m^2 - \theta_{2(m-1)}^3 + \theta_{1m}^6 + \theta_{1m}^{13} = 0 \qquad m \geq 2 \tag{3.54}$$

$$\theta_1^2 - \theta_1^8 + \theta_{11}^{13} = 0 \tag{3.55}$$

$$\theta_{|\mathcal{T}|m}^3 - \theta_{(m-1)}^4 + \theta_{|\mathcal{T}|m}^6 + \theta_{|\mathcal{T}|m}^{13} = c_{|\mathcal{T}|}^3 \qquad 2 \leq m \leq M - 1 \tag{3.56}$$

$$\theta_{|\mathcal{T}|1}^3 - \theta_{|\mathcal{T}|}^8 + \theta_{|\mathcal{T}|1}^{13} = c_{|\mathcal{T}|}^3 \tag{3.57}$$

$$-\theta_{|\mathcal{T}|(M-1)}^3 + \theta_{|\mathcal{T}|-1}^5 + \theta_{(|\mathcal{T}|-1)M}^6 + \theta_{(|\mathcal{T}|-1)M}^{13} = c_{|\mathcal{T}|-1}^3 \tag{3.58}$$

$$-\theta_{(M-1)}^4 + \theta_{|\mathcal{T}|}^5 + \theta_{|\mathcal{T}|M}^6 + \theta_{|\mathcal{T}|M}^{13} = c_{|\mathcal{T}|}^3 \tag{3.59}$$

$$\theta_m^4 + \theta_{(|\mathcal{T}|+1)m}^{13} = c_{|\mathcal{T}|+1}^3 \qquad m \leq M - 1 \tag{3.60}$$

$$\theta_{t1}^3 - \theta_t^8 + \theta_{t1}^{13} = c_t^3 \qquad 2 \leq t \leq |\mathcal{T}| - 1 \tag{3.61}$$

$$-\theta_{(t+1)(M-1)}^3 + \theta_t^5 + \theta_{tM}^6 + \theta_{tM}^{13} = c_t^3 \qquad 2 \leq t \leq |\mathcal{T}| - 2 \tag{3.62}$$

$$\theta^3_{tm} - \theta^3_{(t+1)(m-1)} + \theta^6_{tm} + \theta^{13}_{tm} = c^3_t \qquad 2 \le t \le |\mathcal{T}| - 1, 2 \le m \le M - 1 \tag{3.63}$$

$$\theta^1_t \le 0 \qquad t \ge 2 \tag{3.64}$$

$$\theta^6_{tm} \ge 0 \qquad t \in \mathcal{T}, m \ge 2 \tag{3.65}$$

$$\theta^{10}_t, \theta^{11}_t, \theta^{12}_t \ge 0 \qquad t \in \mathcal{T} \tag{3.66}$$

$$\theta^{13}_{tm} \ge 0 \qquad 1 \le t \le |\mathcal{T}| + 1, 1 \le m \le M \tag{3.67}$$

$$\theta^{14}_{tm} \ge 0 \qquad t \in \mathcal{T}, 1 \le m \le M \tag{3.68}$$

$$(\gamma F^{max} \hat{y}_t - x'_t) \times \theta^1_t = 0 \qquad t \ge 2 \tag{3.69}$$

$$(I^m_t - u^m_t) \times \theta^6_{tm} = 0 \qquad t \in \mathcal{T}, m \ge 2 \tag{3.70}$$

$$\theta^{10}_t \times x'_t = 0 \qquad t \in \mathcal{T} \tag{3.71}$$

$$\theta^{11}_t \times s_t = 0 \qquad t \in \mathcal{T} \tag{3.72}$$

$$\theta^{12}_t \times o_t = 0 \qquad t \in \mathcal{T} \tag{3.73}$$

$$\theta^{13}_{tm} \times I^m_t = 0 \qquad 1 \le t \le |\mathcal{T}| + 1, 1 \le m \le M \tag{3.74}$$

$$\theta^{14}_{tm} \times u^m_t = 0 \qquad t \in \mathcal{T}, 1 \le m \le M \tag{3.75}$$

$$\xi_t = \xi^+_t - \xi^-_t \qquad t \in \mathcal{T} \tag{3.76}$$

$$0 \le \xi^+_t, \xi^-_t \le 1 \qquad t \in \mathcal{T} \tag{3.77}$$

$$\sum_{t \in \mathcal{T}} (\xi^+_t + \xi^-_t) \le \Gamma. \tag{3.78}$$

In this model, constraints (3.35)-(3.46) represent the primal feasible space $\mathcal{Y}$. Constraints (3.47)-(3.68) indicate the associated dual feasible space $D(\mathcal{Y})$. Constraints (3.69)-(3.75) are the corresponding KKT complementary slackness conditions. Variables $\theta^1, ..., \theta^{14}$ are variables of $D(\mathcal{Y})$ that are associated with the constraints of $\mathcal{Y}$ feasible space. Finally, constraints (3.76)-(3.78) shows the linear representation of the budgeted uncertainty set. In other words, these constraints ensure that demand uncertainty relates to a budgeted uncertainty set where the budget of uncertainty is equal to Γ. The nonlinear constraints (3.69)-(3.75) of the above formulation can be reformulated by introducing binary variables and sufficiently large coefficients. In particular, we can linearize constraints (3.69) using the following constraints:

$$\theta_t^1 \leq M z_t \qquad t \in \mathcal{T} \quad (3.69\text{-}1)$$

$$(\gamma F^{max} \hat{y}_t - x_t') \leq M(1 - z_t) \qquad t \in \mathcal{T} \quad (3.69\text{-}2)$$

$$z \in \{0, 1\}^{|\mathcal{T}|}. \quad (3.69\text{-}3)$$

We follow a similar approach to linearize constraints (3.70)-(3.75).

3.4 Computational results and discussions

In this section, we carry out extensive numerical experiments to evaluate our robust algorithm and to provide various sensitivity analyses. In Section 3.4.1, we present computational results to assess the efficiency of the proposed C&RG algorithm. Section 3.4.2 studies the effect of order modification on the performance of the algorithm for different combinations of input parameters. In Section 3.4.3, we analyze the effect of considering the possibility of receiving items with different shelf-lives on optimal solutions. In Section 3.4.4, we study the effect of the budget and level of uncertainty on the obtained objective values. Finally in Section 3.4.5, we compare the proposed robust formulation with a commensurable two-stage stochastic model in both worst-case and risk-neutral settings.

All computational experiments were performed in C++ using Concert Technology of IBM CPLEX 12.9.0 on an Intel Xeon CPU E5-2687W v3 processor at 3.10 GHz and 750 GB of RAM in Linux. We dealt with constraints (3.69)-(3.75) by exploiting the capability of CPLEX in directly handling implications using IloIfThen logical constraints instead of using Big M. We create test problems based on the data set of Gunpinar and Centeno [39] and Zhou et al. [110] for a blood (platelet) inventory management problem. The average daily demands ($\bar{d_t}$) are 24, 16, 32, 16, 24, 0, and 8 from Monday to Sunday, respectively and the deviations are assumed to be 25% of the mean values ($\hat{d_t} = \sigma\bar{d_t}$ and $\sigma = 0.25$). As stated in Zhou et al. [110], excluding testing, transportation, and arrangement times, platelets have a maximum of three days of shelf-life. Initial inventories are generated between zero and 10 units. Furthermore, we set $\gamma = 20\%$, $\delta = 25\%$, and $\lambda = (0, 0, 1)$. The optimality gap is set to 0.005% for all 564 experiments. Table 3.1 summarizes the parameters used for the experiments.

Table 3.1 Input parameters used in the algorithm

Parameter	*Value*
Number of Days ($\lvert\mathcal{T}\rvert$)	7,14,21,28
Shelf-life time (M)	up to 3 days
Number of patterns (P)	20
Fixed ordering cost (c^1)	225 $
Regular purchase cost (c^2) per unit	538 $
Holding cost per unit and per time(c^3)	1.25 $
Shortage cost per unit (c^4)	1500 $
Outdating cost per unit (c^5)	150 $
Modification cost per unit (c^6)	$(1+\delta)\times$ 538 $
Percent of extra purchase cost(δ)	25%
Maximum purchase (F^{max})	70 units
Maximum percent for modifications (γ)	20%
Budget of demand uncertainty (Γ)	$0.25 \times \lvert\mathcal{T}\rvert$

3.4.1 Computational performance of the column-and-row generation algorithm

We solve four sets of test problems each including 10 instances with the planning horizon of one, two, three and four weeks. Table 3.2 shows the results of the proposed algorithm for various test instances. In this table, for each instance with different settings of modifications (γ, δ), we present different cost values along with the number of iterations and time to find the optimal solution. Recall that γ is a limit on the extra purchases that is stated as a percentage of the maximum purchase capacity. For example, $\gamma = 0.5$ means that the extra purchase in each period can be at most 50% of

F_{max}. Moreover, δ is the additional cost incurred for each unit of extra purchases. We solve the instances within a time limit of 18000 seconds (five hours) and report the best obtained upper bound. We also set a time limit of 3600 seconds (one hour) for solving every SP associated to the C&RG algorithm. We observe that, our proposed algorithm optimally solves all weekly and bi-weekly instances very quickly. It also solves the third set of instances in less than two hours on average. However, some instances of the four-week planning could not be solved optimally within the limit but the average gap for those instances is less than 1% that is reasonable and shows that our algorithm is reliable for large instances. According to "*Extra purchase* (\$)" column of Table 3.2, the role of modification becomes more significant as the planning horizon increases. This is because of the fact that we fix the initial purchases for the all periods at the beginning. Therefore, the longer the planning horizon is, the more we may need to modify the orders, particularly for later periods.

3.4.2 Impact of modification on optimal solutions

In this set of experiments, we carry out an analysis to further evaluate the value of order modification. We create various test instances for different planning horizons ($|\mathcal{T}| = 7, 14, 21,$ and 28) and different combinations for the fixed ordering cost and purchase capacity (c^1, F_{max}). We conduct these experiments in two modes: with and without modifications. For the mode with modifications, we set $\gamma = 50\%$ and $\delta = 25\%$. As one can see in Table 3.2, a large portion of the total cost is related to the fixed purchase cost that appears in regular ($c_2 x$), extra ($c_6 x = c_2 x + \delta c_2 x$) and emergency purchases ($c_4 S$). Therefore, in order to have a better viewpoint about the actual effect of modifications in performance, we subtract the fixed purchase cost from all three mentioned procurement costs and then compare the computational results with and without modifications. This new setting helps us understand what percent of the variable cost can be reduced by modification. We have regular purchase cost

Table 3.2 Results obtained by the C&RG algorithm for various (γ, σ)

Data Info.				*Column-and-row generation algorithm*								
$\|T\|$	*Number*	γ	σ	*Total cost* (\$)	*Regular purchase* (\$)	*Inventory* (\$)	*Extra purchase*(\$)	*Shortage*(\$)	*Wastage*(\$)	*Iteration.*	*Time* (s)	*Gap* (%)
7	1	0	-	62031	59854	157	0	1120	0	4	0.2	0.0
	2	0.1	0.25	61515	59446	147	0	1020	0	5	0.3	0.0
	3	0.2	0.25	61515	59446	156	1012	0	0	5	0.4	0.0
	4	0.5	0.25	61515	59446	156	1012	0	0	5	0.5	0.0
	5	0.5	0.2	61480	59419	156	1004	0	0	5	0.6	0.0
	6	0.5	0.15	61414	58557	228	1952	0	0	5	0.7	0.0
	7	0.5	0.1	61333	58519	228	1909	0	0	3	0.7	0.0
	8	0.5	0.05	61249	58454	228	1891	0	0	3	0.5	0.0
	9	0.5	0.01	61177	58398	228	1876	0	0	3	0.6	0.0
	10	1	0.01	61177	58398	228	1876	0	0	3	0.6	0.0
	Average			61441	58994	191	1253	214	0	4	0.5	0.0
14	1	0	-	142090	133965	408	0	5513	402	8	12	0.0
	2	0.1	0.25	135932	126760	366	4707	2298	0	8	23	0.0
	3	0.2	0.25	135926	126751	379	6995	0	0	6	14	0.0
	4	0.5	0.25	135925	126757	375	6991	0	0	8	21	0.0
	5	0.5	0.2	135647	126750	382	6714	0	0	8	21	0.0
	6	0.5	0.15	135368	126742	382	6443	0	0	8	25	0.0
	7	0.5	0.1	135088	126734	416	5905	0	231	8	27	0.0
	8	0.5	0.05	134721	122226	505	10400	0	239	8	8	0.0
	9	0.5	0.01	134316	122213	517	10236	0	0	8	11	0.0
	10	1	0.01	134316	122213	517	10236	0	0	5	13	0.0
	Average			135933	126111	425	6863	781	87	8	18	0.0
21	1	0	-	224363	198634	598	0	22431	0	11	1522	0.0
	2	0.1	0.25	210944	191612	594	16037	0	0	10	12684	0.0
	3	0.2	0.25	210355	193922	586	13146	0	0	8	3914	0.0
	4	0.5	0.25	210363	193931	639	13092	0	0	11	10510	0.0
	5	0.5	0.2	209837	193903	594	12639	0	0	13	9566	0.0
	6	0.5	0.15	209320	193872	638	12108	0	0	11	7713	0.0
	7	0.5	0.1	208795	193840	638	11616	0	0	15	10072	0.0
	8	0.5	0.05	208100	187625	795	17655	0	0	10	98	0.0
	9	0.5	0.01	207428	187609	795	16998	0	0	10	75	0.0
	10	1	0.01	207428	187621	781	16526	0	473	9	69	0.0
	Average			210693	192257	666	12982	2243	47	11	5622	0.0
28	1	0	0	307363	283633	905	0	18318	3811	9	limit	1.0
	2	0.1	0.25	289066	261025	864	19422	4154	0	6	limit	1.5
	3	0.2	0.25	289159	265221	871	18986	0	481	6	limit	1.9
	4	0.5	0.25	288928	260794	871	23663	0	0	7	limit	1.4
	5	0.5	0.2	290000	263416	855	21654	0	474	6	limit	2.3
	6	0.5	0.15	285998	261871	879	19344	0	302	7	limit	1.0
	7	0.5	0.1	282616	259631	802	18582	0	0	21	17950	0.0
	8	0.5	0.05	281478	253057	1072	24648	0	0	15	1247	0.0
	9	0.5	0.01	280549	253037	1059	23277	0	475	11	482	0.0
	10	1	0.01	280539	253029	1052	23047	0	709	16	1889	0.0
	Average			287570	261471	923	19262	2247	625	10	12962	0.9

$c'^2 = c^2 - c^2 = 0$, shortage cost $c'^4 = c^4 - c^2$ and modification cost $c'^6 = c^6 - c^2$. The reason for revising shortage and modification cost is that a part of these expenses is related to the regular purchase cost, regardless of the realized uncertainty, and must be paid to the suppliers.

Table 3.3 shows the results of this analysis. In this table, $\Delta = \frac{Obj_1^* - Obj_2^*}{Obj_1^*} \times 100$ is defined as the relative percentage deviation between the objective functions of two cases, where Obj_1^* and Obj_2^* are the optimal values in the case of without and with modifications, respectively. In other words, Δ represents the percentage of improvement as a result of considering modifications. Similar to previous sections, all instance are solved within a time limit of 18000 seconds and 3600 seconds for solving each SP associated to the C&RG algorithm. According to this table, on average, order modification improves the worst-case total cost by 16%, 31%, 33%, and 34% for instances with $|\mathcal{T}| = 7$, 14, 21, and 28 days, respectively. Furthermore, the results with modifications outperform those of the case with no modification in terms of the obtained gap, the number of iterations and required time to find optimal or near-optimal solutions. Therefore, finding the initial purchase amount of each period and then modifying it seemed easier for the algorithm rather than finding the exact amount of purchases from the beginning.

Figure 3.2 depicts the improvement obtained by order modification for different combinations of cost and capacity parameters. In order to be fair, we do not compute the improvement for instances with a gap above 5%. We can see in Figures 3.2 (a) and 3.2 (b) that allowing order modification makes more improvements as the planning horizon increases. From Figure 3.2 (a), we note that the fixed ordering cost can play a major role in determining the effect of modification in the performance. As the fixed ordering cost increase, the amount of improvements made by modifications decreases because at higher values of c_1, the ordering cost will take a significant portion of the total cost. Furthermore according to Figure 3.2 (b), the purchase capacity is slightly affecting the amount of improvement only at lower values of capacity in the bi-weekly planning.

3.4.3 Impact of shelf-life on optimal solutions

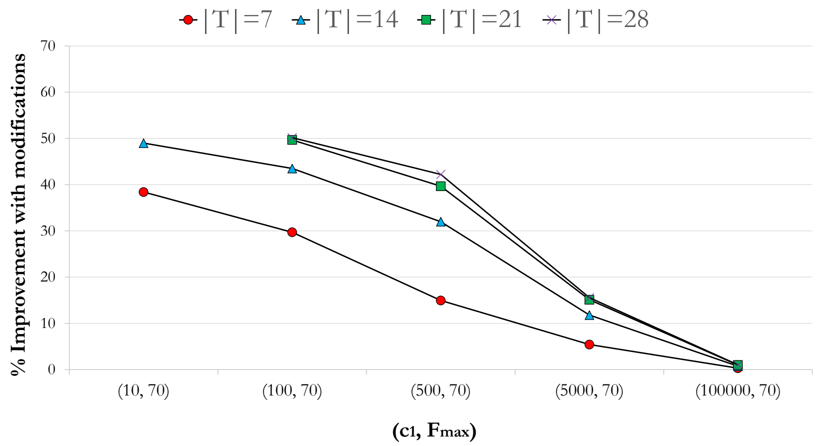
In this section, we assess the effect of perishable item's shelf-life on the quality of the obtained solution. We assume the possibility of receiving ordered items with different shelf-lives. For instance, $(\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 1)$ denotes that all the received items have the shelf-life of three days, while $(\lambda_1 = 0, \lambda_2 = 0.5, \lambda_3 = 0.5)$ implies that 50% of items have the shelf-life of three days and the other 50% will be outdated in two days. Table 3.4 shows the results obtained with different shelf-lives. According to

Table 3.3 Results obtained by the C&RG algorithm for various (c^1, F_{max}) with and without modifications

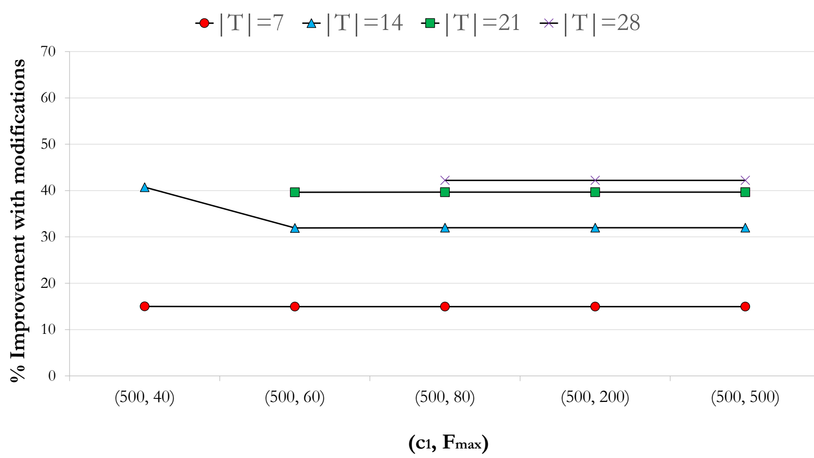
Data Info.				*Without modification* ($\gamma = 0$)					*With modification* ($\gamma = 0.5, \sigma = 0.25$)					Δ(%)
$\lvert T\rvert$	*Number*	c^1	F_{max}	# or orders	*Total cost* ($)	*Iteration*	*Time* (s)	*Gap* (%)	# or orders	*Total cost* ($)	*Iteration*	*Time* (s)	*Gap* (%)	
7	1	500	40	4	3233	3.0	0.5	0.0	4	2748	2	0.3	0.0	15
	2	500	60	4	3223	5.0	0.5	0.0	4	2741	3	0.3	0.0	15
	3	500	80	4	3223	5.0	0.6	0.0	4	2741	3	0.3	0.0	15
	4	500	200	4	3223	5.0	0.8	0.0	4	2741	3	0.3	0.0	15
	5	500	500	4	3223	5.0	0.9	0.0	4	2741	3	0.5	0.0	15
	6	10	70	4	1263	3.0	0.2	0.0	6	778	3	1.2	0.0	38
	7	100	70	4	1623	4.0	0.2	0.0	4	1141	3	0.2	0.0	30
	8	500	70	4	3223	5.0	0.3	0.0	4	2741	3	0.2	0.0	15
	9	5000	70	3	17288	4.0	0.4	0.0	3	16350	2	0.2	0.0	5
	10	100000	70	3	302288	4.0	0.4	0.0	3	301350	2	0.3	0.0	0.3
	Average			4	34181	4	1	0.0	4	33607	3	0.4	0.0	16
14	1	500	40	12	11742	10	10317	0.0	8	6960	5	13	0.0	41
	2	500	60	8	9802	8	21	0.0	6	6672	7	3	0.0	32
	3	500	80	8	9802	8	21	0.0	6	6667	7	8	0.0	32
	4	500	200	8	9802	8	22	0.0	6	6667	8	10	0.0	32
	5	500	500	8	9802	8	24	0.0	6	6667	8	10	0.0	32
	6	10	70	12	5862	8	11075	0.0	12	2991	3	10826	0.0	49
	7	100	70	8	6602	8	5	0.0	8	3731	3	7	0.0	43
	8	500	70	8	9802	8	6	0.0	6	6667	8	1	0.0	32
	9	5000	70	6	38170	12	4	0.0	6	33667	7	2	0.0	12
	10	100000	70	6	608170	10	8	0.0	6	603676	7	3	0.0	0.7
	Average			8	71956	9	2150	0.0	7	68437	6	1088	0.0	31
21	1	500	40	18	23145	6	limit	17.8	12	11166	9	2577	0.0	-
	2	500	60	12	17374	10	344	0	9	10488	8	37	0.0	40
	3	500	80	12	17374	10	246	0	9	10484	14	72	0.0	40
	4	500	200	12	17374	10	254	0	9	10484	12	98	0.0	40
	5	500	500	12	17374	10	262	0	9	10484	12	103	0.0	40
	6	10	70	18	13750	5	limit	137.4	18	5209	5	17850	0.0	-
	7	100	70	12	12574	10	174	0	12	6329	6	1507	0.0	50
	8	500	70	12	17374	10	225	0	9	10484	12	16	0.0	40
	9	5000	70	9	60062	20	13	0	9	50984	13	79	0.0	15
	10	100000	70	9	915082	21	57	0	9	905990	9	39	0.0	1
	Average			13	111148	11	3758	15.5	11	103210	10	2253	0.0	33
28	1	500	40	24	29680	6	limit	32.5	16	15393	7	limit	0.1	-
	2	500	60	16	25321	13	limit	9.6	12	14305	14	209	0	-
	3	500	80	16	24740	15	limit	4.7	12	14297	23	6358	0	42
	4	500	200	16	24740	14	limit	4.7	12	14297	19	6805	0	42
	5	500	500	16	24740	14	limit	4.7	12	14297	19	4812	0	42
	6	10	70	24	25909	2	limit	482.4	24	7488	5	limit	0.9	-
	7	100	70	16	17925	16	limit	4	16	8936	7	limit	0.2	50
	8	500	70	16	24740	14	limit	4.7	12	14297	23	6310	0	42
	9	5000	70	12	80880	32	114	0	12	68297	20	3849	0	16
	10	100000	70	12	1220893	26	363	0	12	1208308	12	2783	0	1
	Average			17	149957	15	14448	54.7	14	137992	15	8513	0.1	34

the table, although the proposed algorithm solves all instances optimally, for some combinations of shelf-life, the problem is more difficult as it takes more time to find the optimal solution (e.g. instance number # 10). Moreover, when the received items are likely to be outdated earlier, the amount of extra purchases increases to reduce the risk of shortage.

Figure 3.3 shows the effect of different shelf-lives on the total cost. From this figure, we can notice that as the probability of receiving orders with lowest shelf-life (λ_1) increases, the total cost increases. This happens because of the fact that the risk of wastage is more and we need more extra and emergency purchases when the received items are likely to be outdated soon. However, the total cost might not be



(a) Impact of fixed cost



(b) Impact of purchase capacity

Figure 3.2: Effect of considering modifications on performance.

affected by different shelf-lives (as in instance # 1 to 8) when the portion of aged items in received orders is not remarkable. In these cases, there is a range for shelf-life for which the obtained performance is almost the same. This is because the ordering pattern and purchase amounts do not change with slight changes in the combination of shelf-life probabilities. When there is a big jump in the total cost, it implies that the first-stage solutions, including the periods to make orders and the amount of orders, have totally changed and the current combination of shelf-life has had a significant impact on the first-stage decisions.

Table 3.4 Results obtained from different combinations for shelf-life

Data Info.				*Results obtained with $\mathcal{T} = 14, \gamma = 0.2, \delta = 0.25, F_{max} = 70, c^1 = 225$*								
Number	λ_1	λ_2	λ_3	*Total cost* (\$)	*Regular purchase* (\$)	*Inventory* (\$)	*Extra purchase*(\$)	*Shortage*(\$)	*Wastage*(\$)	*Iteration.*	*Time* (s)	*Gap* (%)
1	0	0	1	135926	126751	379	6995	0	0	6	15	0.00
2	0	0.1	0.9	135925	126757	375	6991	0	0	8	709	0.00
3	0	0.2	0.8	135928	126757	375	6997	0	0	7	904	0.00
4	0	0.3	0.7	135925	126757	379	6987	0	0	8	6192	0.00
5	0	0.4	0.6	135925	126757	379	6987	0	0	8	6809	0.00
6	0	0.5	0.5	135925	126757	379	6987	0	0	8	4570	0.00
7	0	0.6	0.4	135925	126757	379	6987	0	0	9	15149	0.00
8	0	0.7	0.3	135955	126648	397	7109	0	0	10	9090	0.00
9	0	0.8	0.2	136834	124130	343	5162	5398	0	6	1790	0.00
10	0	0.9	0.1	137483	123853	303	10626	0	0	9	17606	0.00
11	0	1	0	137481	122274	366	12345	0	245	7	97	0.00
12	0.1	0.9	0	137480	122277	365	12588	0	0	7	19	0.00
13	0.2	0.8	0	137480	122277	364	12343	0	245	6	19	0.00
14	0.3	0.7	0	137480	122277	365	12588	0	0	6	19	0.00
15	0.4	0.6	0	137863	122277	297	12588	0	0	5	37	0.00
16	0.5	0.5	0	137863	122277	289	12349	0	246	6	37	0.00
17	0.6	0.4	0	137863	122277	289	12349	0	246	5	29	0.00
18	0.7	0.3	0	137863	122277	297	12343	0	245	6	23	0.00
19	0.8	0.2	0	137863	122277	289	12349	0	246	6	26	0.00
20	0.9	0.1	0	137863	122277	289	12349	0	246	6	25	0.00
21	1	0	0	137864	122272	297	12349	0	245	4	23	0.00
Average	0.26	0.48	0.26	136986	124141	343	9922	257	94	7	3104	0.00

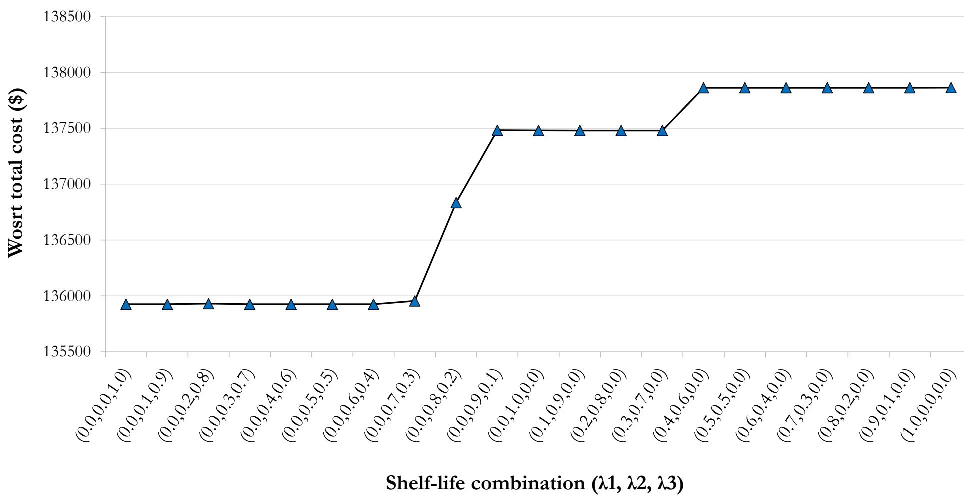


Figure 3.3: Effect of different shelf-lives.

3.4.4 Impact of budget of uncertainty on optimal solutions

In our approach, decision-makers can select the level of conservatism of the robust model by choosing a proper value of the budget of uncertainty parameter (Γ). Moreover, considering different levels of uncertainty (deviation from the mean values) could result in choosing different budget of uncertainty for decision-makers. We study the effect of budgets (α parameter in $\Gamma = \alpha|\mathcal{T}|$) and levels of uncertainty (σ parameters) on the quality of the obtained solutions and on the performance of the proposed

C&RG algorithm. We use instances with the planning horizon of 21 days and set $\gamma = 0.2$, $\delta = 0.25$, $c^1 = 225$ and $F_{max} = 70$. Table 3.5 and Figure 3.4 present the results of this analysis. In the table, the nominal cost is the objective function of the deterministic model. We observe that, in the majority of instances, as the level of uncertainty (σ) increases from 0.25 to 0.75, the solution time increases. At each level of uncertainty, by increasing the budget of uncertainty, the algorithm leads to greater optimal solution values. However, the marginal increase of the optimal cost decreases as the budget of uncertainty grows. At all three levels of uncertainty (σ=0.25, 0.5, and 0.75), the additional protection obtained by using a higher budget becomes null after $\alpha = 0.85$. This can be explained by the fact that we plan for $|\mathcal{T}| = 21$ days and according to the used data set, $\bar{d}_t = 0$ for t=6, 13, and 20. Therefore, there is no uncertainty in the demand of theses three days ($\hat{d}_t = \sigma \bar{d}_t = 0$) and as a result, the budget of uncertainty considered for them has no impact on the total cost.

Table 3.5 Effect of budget of uncertainty on the performance of the algorithm

Data info		σ=0.25				σ=0.50				σ=0.75			
Number	α	*Total cost*	*Iteration*	*Time*(s)	*% Gap*	*Total cost*	*Iteration*	*Time*(s)	*% Gap*	*Total cost*	*Iteration*	*Time*(s)	*% Gap*
1	0.00	187817	1	0	0.0	187817	1	0	0.0	187817	1	0	0.0
2	0.05	192930	5	111	0.0	198132	4	975	0.0	203534	4	2770	0.0
3	0.10	197855	7	351	0.0	208423	7	4643	0.0	220042	6	8501	0.0
4	0.15	202713	9	1935	0.0	218230	8	10353	0.0	234883	8	17416	0.0
5	0.20	206541	8	3906	0.0	227001	9	17958	0.0	252421	7	limit	2.8
6	0.25	210355	8	3802	0.0	235235	9	limit	0.2	259269	5	limit	1.5
7	0.30	213961	13	17409	0.0	244842	10	limit	1.1	272303	6	limit	0.8
8	0.35	217286	15	17594	0.0	251483	10	limit	1.1	282188	10	limit	0.1
9	0.40	220593	15	17913	0.0	258705	10	limit	1.1	295668	7	limit	1.8
10	0.45	223855	12	17190	0.0	262776	9	16388	0.0	300644	9	limit	0.2
11	0.50	227406	13	limit	0.4	268166	12	17796	0.0	310285	9	limit	0.5
12	0.55	228499	15	17830	0.0	272243	13	17062	0.0	316131	8	limit	0.4
13	0.60	231085	17	limit	0.3	275171	11	17546	0.0	324051	8	limit	1.5
14	0.65	233496	17	limit	0.3	278778	14	10994	0.0	328845	8	limit	1.1
15	0.70	235223	17	11116	0.0	282435	15	5357	0.0	335160	10	limit	1.3
16	0.75	237568	14	2520	0.0	287043	15	4077	0.0	336628	10	3054	0.0
17	0.80	238752	19	5997	0.0	289407	12	2959	0.0	340060	6	1727	0.0
18	0.85	239930	17	3451	0.0	291759	12	3303	0.0	343593	10	2301	0.0
19	0.90	240814	5	269	0.0	294020	15	1551	0.0	346983	4	976	0.0
20	0.95	240814	5	89	0.0	294020	12	1440	0.0	346983	6	805	0.0
21	1.00	240814	5	91	0.0	294020	11	803	0.0	346983	7	1583	0.0
Average	0.50	222300	11	8459	0.05	258081	10	9828	0.17	294499	7	11320	0.57

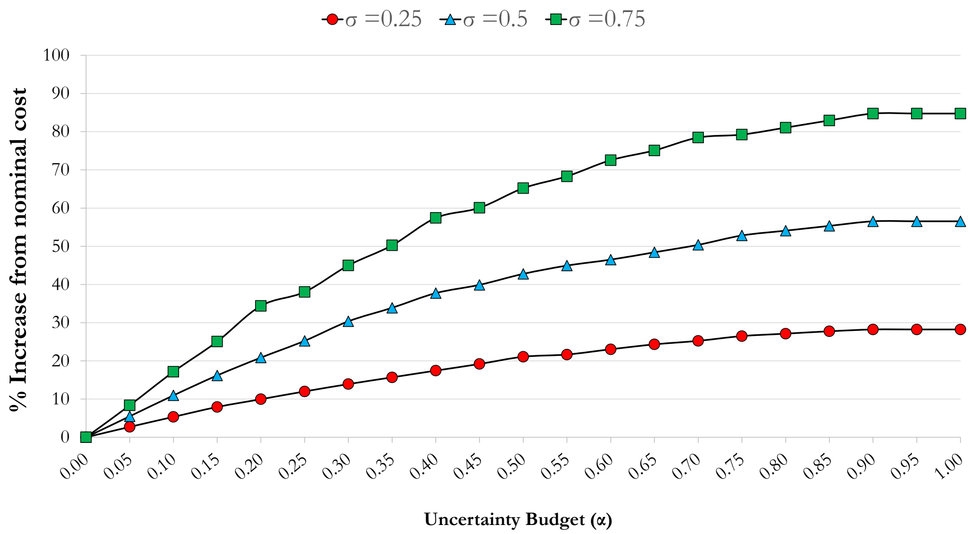


Figure 3.4: Effect of uncertainty budget.

3.4.5 Value of robustness: Robust vs. Stochastic vs. Deterministic

In the final analysis, we compare the solutions obtained from the deterministic model (with mean demand values), a two-stage stochastic variant of the PIMM problem, and our proposed robust approach on instances with the planning horizon of 14 days. In order to make this evaluation as fairly as possible, we define $d_t(\omega)$ as independent random variables that follow a uniform distribution over the interval $d_t(\omega) \sim U[\bar{d}_t - \sigma \times \bar{d}_t, \bar{d}_t + \sigma \times \bar{d}_t]$, where σ defines the uncertainty level as the percentage of deviation from mean values. The two-stage stochastic programming model for the PIMM problem with stochastic demands can be stated as follows:

$$\text{minimize} \;\; \sum_{t \in \mathcal{T}} \left(c_t^1 y_t + c_t^2 x_t\right) + \mathbb{E}_\omega \Big[\sum_{m=1}^{M} \Big(c_t^3 I_{t+1}^m(\omega) \Big) + c_t^4 s_t(\omega) + c_t^5 o_t(\omega) + c_t^6 x_t'(\omega) \Big]$$

subject to

$$(w, y, x) \in \mathcal{X}$$

$$(x', I, s, o, u) \in \mathcal{Y}(\omega), \;\; \omega \in \Omega.$$

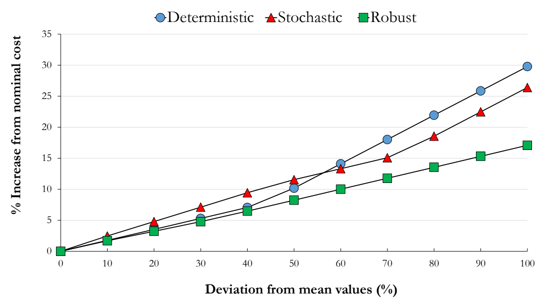
In the above model, Ω denotes the set of all generated demand scenarios for the second-stage problem and $\mathcal{Y}(\omega)$ denotes the feasible space of the second-stage model for each scenario in Ω. Recall that $\mathcal{X}$ represent the feasible space of the first-stage model. To obtain optimal first-stage solutions for the above two-stage stochastic model, we use a sample average approximation (SAA) algorithm that is similar to the four-step SAA algorithm presented in [26]. We set the SAA algorithm with 20 different samples of size 500 generated from the mentioned uniform distribution to estimate a lower bound, and we use a sample size of 15000 scenarios to estimate an upper bound.

In the first step of this analysis, we compare the solutions obtained from the three mentioned models in 30 settings with different values of the level and budget of uncertainty (σ, α). In each setting, we use the SP of the proposed C&RG algorithm to evaluate the worst-case performance of the solutions from the deterministic, stochastic and robust models. Table 3.6 and Figure 3.5 present the results of the worst-case performance of the solutions from different models in different settings. Recall that nominal costs are associated with the evaluation of the objective function using the mean value of uncertain demand (the objective function of the deterministic model). As we can see, our proposed robust approach significantly outperforms the deterministic and stochastic models. For Γ=1, 3, and 7, our proposed robust approach finds solutions that are 30%, 39%, and 42% less expensive that those of the stochastic model on average. It should be noted that compared to the stochastic model, the deterministic model provides better worst-case performance at smaller values of the uncertainty level (deviation σ). However, the stochastic model outperforms the deterministic model when the uncertainty level is higher. The robust model always outperforms two other models. We can also observe that as the uncertainty level increases, the superiority of our robust model becomes more sensible.

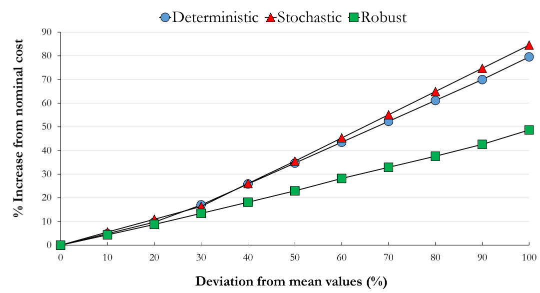
In the second part of our comparison, using a sample of 15000 randomly generated

Table 3.6 % Increase from nominal cost in worst-case settings

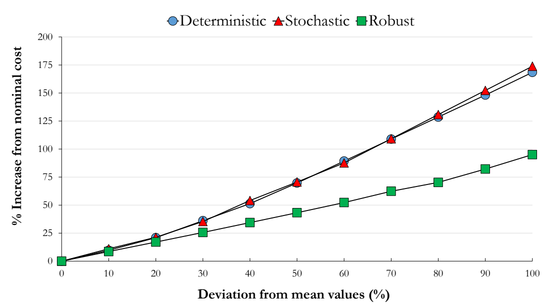
Data info		$\Gamma = 1$ $(\alpha = 0.1)$			$\Gamma = 3$ $(\alpha = 0.25)$			$\Gamma = 7$ $(\alpha = 0.50)$		
Number	σ	*Deterministic*	*Stochastic*	*Robust*	*Deterministic*	*Stochastic*	*Robust*	*Deterministic*	*Stochastic*	*Robust*
1	10	1.8	2.5	1.7	4.9	5.5	4.4	9.7	11.1	8.6
2	20	3.5	4.8	3.2	9.7	10.9	8.8	20.9	21.3	17.0
3	30	5.3	7.1	4.8	17.0	16.4	13.5	36.1	35.4	25.6
4	40	7.1	9.4	6.5	25.8	26.1	18.1	51.3	54.1	34.4
5	50	10.2	11.5	8.2	34.7	35.5	22.9	69.7	70.6	43.2
6	60	14.1	13.3	10.0	43.5	45.3	28.2	89.3	87.6	52.3
7	70	18.0	15.1	11.8	52.3	55.1	32.9	108.9	109.2	62.4
8	80	21.9	18.6	13.5	61.1	64.9	37.6	128.5	130.7	70.3
9	90	25.9	22.5	15.3	70.0	74.7	42.6	148.1	152.3	82.2
10	100	29.8	26.4	17.1	79.5	84.5	48.7	168.4	173.9	95.0
Average		15.1	14.3	10.0	43.7	45.9	28.1	91.3	92.8	53.6



(a) $\Gamma = 1$



(b) $\Gamma = 3$



(c) $\Gamma = 7$

Figure 3.5: Solution performance in worst-case setting.

scenarios, we assess the average-case performance of the first-stage solutions obtained by our robust optimization ($\Gamma = 1$), the deterministic and stochastic programming models. Table 3.7 provides the expected cost, standard deviation and the relative percentage deviation from nominal cost for the solutions of those models. Moreover, Figure 3.6 illustrates the results of this comparison. In this figure we observe that, the best performance is obtained for the solutions of the stochastic model and the worst performance is for those of the deterministic model. The proposed robust model significantly outperforms the deterministic model in risk-neutral settings while it is only slightly worse than the stochastic model. However, on average, the standard deviation of the robust model is almost 70% less than that of the stochastic model which makes the results of our robust model more reliable.

Table 3.7 Solution performance in a risk-neutral setting

Data info		*Deterministic*			*Stochastic*			*Robust*		
Number	σ	*Exp. Cost*	*S.D.*	*% Increase*	*Exp. Cost*	*S.D.*	*% Increase*	*Exp. Cost*	*S.D.*	*% Increase*
1	0.1	125316	28.2	2.4	123480	38.4	0.9	124592	2.9	1.8
2	0.2	128167	51.5	4.8	124453	71.3	1.7	125749	13.9	2.8
3	0.3	131488	78.6	7.5	125667	117.8	2.7	127370	26.1	4.1
4	0.4	134542	103.5	10.0	126927	148.6	3.8	129049	38.3	5.5
5	0.5	138970	139.7	13.6	129128	175.6	5.6	131414	54.9	7.4
6	0.6	142018	164.6	16.1	130113	227.1	6.4	133074	67.4	8.8
7	0.7	146115	198.0	19.4	131497	269.5	7.5	135277	82.5	10.6
8	0.8	151103	238.7	23.5	133400	330.8	9.0	137839	99.6	12.7
9	0.9	155580	275.3	27.2	135484	353.4	10.8	140160	117.7	14.6
10	1	162579	332.4	32.9	138758	454.6	13.4	144081	144.8	17.8
Average		141587.8	161.1	15.7	129890.7	218.7	6.2	132860.5	64.8	8.6

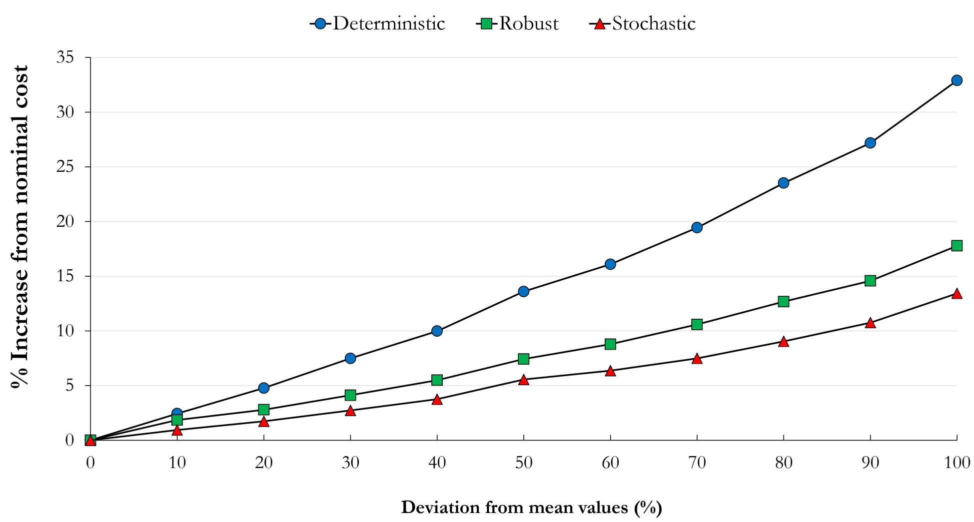


Figure 3.6: Solution performance in a risk-neutral setting.

3.5 Conclusions

We studied a perishable inventory management problem considering demand uncertainty with the possibility of modifying previously placed orders. We formulated the problem as a two-stage robust optimization model with a budget of uncertainty to control the level of conservatism. To solve the formulated problem, we developed an exact robust algorithm based on the column-and-row generation method. Our extensive computational analysis demonstrated that the proposed robust approach was capable of solving different test instances optimally in a reasonable amount of time. We showed that considering order modification could significantly reduce the total cost. Furthermore, although considering various shelf-lives might add some degree of complexity to the problem, it would make the model more realistic and flexible enough to help in reducing the total cost. We carried out various sensitivity analyses on different parameters to provide important insights for decision-makers in choosing the right value of each parameter. We also showed that our robust algorithm had an acceptable performance in risk-neutral settings, while it provided the best performance in worst-case settings as compared to the deterministic and stochastic variants of the problem.

For future work, it would be interesting to consider multiple types of items that can be substituted with each other if needed. Another research direction is to extend this work to the case of multiple demand classes where one class only accepts fresh items while the other may accept items of any lifetime. It is also interesting to consider problems with the possibility of using the expired items in the primary or a secondary market after re-processing items at the end of their shelf-life.

Chapter 4

Decomposition strategies for multi-item uncapacitated lot-sizing problems with inventory bounds

P. Hooshangi-Tabrizi, I. Contreras. "Decomposition strategies for multi-item uncapacitated lot-sizing problems with inventory bounds". to be submitted to *TBD*, March 2020.

Abstract

We study the multi-item uncapacitated lot-sizing problem with inventory bounds. We present a new formulation based on two sets of variables to determine the production and inventory intervals for each item. Based on the new formulation, we propose a Lagrangian relaxation to obtain tight lower bounds. Exploiting the structure of the problem, we decompose the associated Lagrangian function into smaller problems that can be solved efficiently. In order to solve the Lagrangian dual problem, we propose a deflected subgradient method and a stabilized column generation algorithm. In both algorithms, we solve a series of multi-choice knapsack problems, to

evaluate the Lagrangian function or to solve the pricing subproblem, with existing ad hoc algorithms. Computational experiments are performed to evaluate the proposed formulation and algorithms on a set of benchmark instances.

4.1 Introduction

Single-item uncapacitated lot-sizing problem (ULS) is a fundamental production planning problem that deals with the planning of a single item production to meet demand over a discretized planning horizon. The goal is to find when and how much to produce the item with the aim of minimizing the sum of setup (order placement), production and inventory costs. Although the ULS is too restrictive in terms of applicability, it is still important to be addressed as it can be occurring after relaxing complex production planning models [84].

Due to a variety of reasons, such as warehouse infrastructure conditions, voluminous products and administrative policies, the quantity of items to be stored at the end of each period may be subject to an *inventory bound* (IB) [5]. In this case, although items may be produced unrestrictedly, they cannot be stored in unlimited quantities from one period to the next [73]. IBs are even more relevant when dealing with multi-item production environments, where multiple types of products are produced and stored in a shared storage space.

The *single-item lot-sizing problem with inventory bounds* (ULS-IB) was initially studied by Love [73]. Pochet and Wolsey [83] and Barany et al. [14] perform polyhedral studies to identify the complete characterization of the convex hull of integer solutions. Atamtürk and Küçükyavuz [10] identifies several classes of facet-defining inequalities for the ULS-IB with linear and fixed inventory costs. Polynomial time dynamic programming algorithms are developed in [11, 105] to solve the ULS-IB. In [28], an extended formulation describing the convex hull of the solution space is presented.

The ULS-IB with backlogging and lost sales is addressed in [53, 54]. Furthermore, Hwang and van den Heuvel [53] and Phouratsamay et al. [81] propose polynomial and pseudo-polynomial algorithms for the ULS-IB considering different cost structures. More general production planning problems that take IBs into account are studied in [78], [84] [6], [77], [3], and [41].

Akbalik et al. [4, 5] study the *multi-item lot-sizing problem with inventory bounds* (MULSIB) and prove its NP-hardness, even when the Wagner-Within cost structure is considered. According to the Wagner-Within structure, producing and storing one unit of item in a period cost more than producing it in the subsequent period. Melo and Ribeiro [76] present three formulations for the MULSIB using the facility location, shortest path, and (l,S)-inequality approaches. The authors also propose a relax-and-fix matheuristic to find integer solutions. To the best of our knowledge, Acevedo-Ojeda [1] is the only work presenting a decomposition-based approximate algorithm for the MULSIB. It is a column generation algorithm of a Dantzig-Wolfe reformulation of an extended mixed-integer programming (MIP) formulation. Although the results obtained in Acevedo-Ojeda [1] are promising in terms of strengthening the linear programming (LP) relaxation bounds of the based formulation, the proposed algorithm can only solve small-size instances.

The main contribution of this paper is to present and computationally compare decomposition strategies for solving the MULSIB. In particular, we introduce an extended MIP formulation of the integer programming (IP) formulation given in [1]. The extended formulation, in contrast to the IP of [1], allow us to decompose the MULSIB by periods when using well-known decomposition algorithms. We present a Lagrangian relaxation that can provide tighter lower bounds than the LP relaxation of the extended MIP. The proposed Lagrangian function can be decomposed into a series of so-called *multi-choice knapsack problems* (MCKPs), one MCKP for each each period. We use an state-of-the-art exact algorithm to solve the MCKPs. In order to solve the associated Lagrangian dual problem (LDP), we develop two solution algorithms: i) a subgradient optimization algorithm that approximately solves the LDP, and ii) a column generation algorithm that solves an equivalent Dantzig-Wolfe reformulation of the extended MIP in which the convexified constraints correspond to the relaxed constraints of the proposed LR. In both algorithms, we use stabilization techniques to accelerate their convergence. Finally, we report the results of extensive computational experiments to evaluate the relative performance of the proposed formulation and decomposition algorithms on a set of benchmark instances.

The reminder of this paper is organized as follows. In Section 4.2, we formally define the MULSIB and present three existing mathematical programming formulations and our new extended MIP formulation. Section 4.4 presents the details of the proposed solution algorithms for the MULSIB. Section 4.5 provides computational results and Section 4.6 concludes this paper.

4.2 Problem description and formulations

Let M be a set of items and T a finite set of time periods. We define d_t^i as the demand of period $t \in T$ that must be satisfied for item $i \in M$. We assume that all of the demand of each period is immediately satisfied at the beginning of the period and backlogging is not allowed. Any units of items that are produced at the beginning of a period, but not used immediately, must be stored as an inventory in a shared storage space. For each period $t \in [1, ..., |T|-1]$, the quantity of all stored items from one period to the next is restricted by the inventory bound u_t. We assume that any stored item consumes only one unit of storage capacity. Let p_t^i and q_t^i be the variable and fixed production costs for item i in period t, respectively. We also define h_t^i as the unit holding cost of item i in period t. In other words, h_t^i is incurred for each unit of item i in stock between period t and $t+1$. We also assume that all demands are positive and there are no initial and final inventory stocks. The MULSIB consists of determining the optimal lot size of each item $i \in M$ at each period $t \in T$ with the objective of minimizing the total setup, production and inventory costs, in such a way that the IBs are respected at each period.

The MULSIB was initially formulated in [5] as follows. For each $i \in M$ and $t \in T$, we define continuous variables x_t^i equal to the amount of item i produced in period t. Moreover, we define binary variables y_t^i that take value 1 if and only if there is production of item i in period t. Finally, we define continuous variables s_t^i equal to the amount of item i in stock at the end of period t. The MI-ULS-IB can be formulated as the following MIP [5]:

$$
\begin{aligned}
(\mathbf{CF}) \quad \text{minimize} \quad & \sum_{i \in M} \sum_{t \in T} (p_t^i x_t^i + h_t^i s_t^i + q_t^i y_t^i) && && (4.1)\\
\text{subject to} \quad & s_{t-1}^i + x_t^i = d_t^i + s_t^i && i \in M, t \in T && (4.2)\\
& x_t^i \leq M y_t^i && i \in M, t \in T && (4.3)\\
& \sum_{i \in M} s_t^i \leq u_t && t \leq |T| - 1 && (4.4)\\
& x_t^i, s_t^i \geq 0 && i \in M, t \in T && (4.5)\\
& y_t^i \in \{0, 1\} && i \in M, t \in T. && (4.6)
\end{aligned}
$$

The objective function (4.1) minimizes the sum of variable production, holding, and fixed production costs. Constraints (4.2) are the inventory balance equation. Constraints (4.3) enforce the fixed setup variable to take value 1 ($y_t^i = 1$) whenever there is production of item i in time period t, i.e. $x_t^i > 0$. Constraints (4.4) limit the total inventory of items at the end of period t by the storage capacity u_t. Finally, constraint (4.5) and (4.6) denote the non-negativity and integrality conditions.

4.2.1 Facility location formulation

The MULSIB can be reformulated using the facility location approach in lot-sizing problems [65, 76]. In the *facility location formulation* (FLF), for each $i \in M$, $t \in T$, and $l \geq t$, x^i_{tl} is defined as a continuous variable determining the fraction of demand d^i_l produced in time period t to meed the demand of period t. The MULSIB can be formulated as the following MIP [76]:

$$
\begin{aligned}
(\mathbf{FLF})\ \ \text{minimize} & \sum_{i \in M} \sum_{t \in T} \Big(\sum_{l=t}^{|T|} c^i_{tl} x^i_{tl} + q^i_t y^i_t\Big) && && (4.7)\\
\text{subject to}\ & \sum_{k=1}^{t} x^i_{kt} = 1 && i \in M, t \in T && (4.8)\\
& x^i_{kt} \leq y^i_k && i \in M, t \in T, k \leq t && (4.9)\\
& \sum_{i \in M} \sum_{k=1}^{t} \sum_{l=t+1}^{|T|} d^i_l x^i_{kl} \leq u_t && t \leq |T| - 1 && (4.10)\\
& x^i_{tl} \geq 0 && i \in M, t \in T, l \geq t && (4.11)\\
& y^i_t \in \{0, 1\} && i \in M, t \in T, && (4.12)
\end{aligned}
$$

where $c^i_{tl} = d^i_l(p^i_t + \sum_{k=t}^{l-1} h^i_k)$. The objective function (4.7) minimizes the sum of all related production and holding costs. Constraints (4.8) guarantee that all of the demand of item i in period t is satisfied by productions form period $k = 1$ to period $k = t$. Constraints (4.9) are the setup enforcing equations and constraints (4.10) are bounding the level of inventory in each period. Finally, constraints (4.11) and (4.12) are the non-negativity and integrality conditions, respectively.

4.2.2 Cumulative-demand reformulation

The MULSIB can be further reformulated based on accumulating demands in production intervals. In the *cumulative-demand formulation* (CDF), for each $i \in M$,

$t \in T$, $l \geq t$, w^i_{tl} is defined as a binary variable taking value 1 if and only if item i in period t is produced to cover all demand from t to l. Figure 4.1 depicts a solution of the CDF with $|M| = 4$ and $|T| = 10$. For instance, $w^1_{13} = 1$ represents the production of the cumulative demand from periods t_1 to t_3 for item i_1 in period 1. This implies that an inventory equal to d_{t_2} is stored for one period, and an inventory equal to d_{t_3} is stored for two periods. Another example, $w^3_{44} = 1$ shows that the demand of item i_3 is produced in period t_4 and no stock of this item is stored at the end of period t_4.

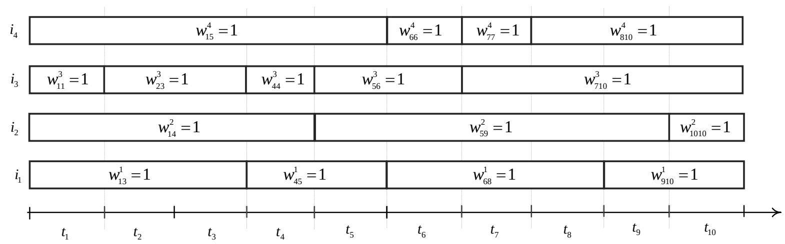


Figure 4.1: A solution of (CDF) for an instance with $|M|$ =4, $|T|$ =10

The MULSIB can be formulated as the following IP [1]:

$$\textbf{(CDF)} \text{ minimize} \sum_{i \in M} \sum_{t \in T} \sum_{l=t}^{|T|} c'^i_{tl} w^i_{tl} \tag{4.13}$$

$$\text{subject to } \sum_{k=1}^{t} \sum_{l=t}^{|T|} w^i_{kl} = 1 \qquad i \in M, t \in T \tag{4.14}$$

$$\sum_{i \in M} \sum_{k=1}^{t} \sum_{l=t+1}^{|T|} D^i_{tl} w^i_{kl} \leq u_t \qquad t \leq |T| - 1 \tag{4.15}$$

$$w^i_{tl} \in \{0, 1\} \qquad i \in M, t \in T, l \geq t, \tag{4.16}$$

where for each $i \in M$, $t \leq |T| - 1$, $l \geq t$, $D^i_{tl} = \sum_{k=t+1}^{l} d^i_k$, and for each $i \in M$, $t \in T$,

$l \geq t$, c'^{i}_{tl} is defined as:

$$c'^{i}_{tl} = \begin{cases} q^i_t + p^i_t d^i_t, & \text{if } l = t, \\ q^i_t + p^i_t(d^i_t + D^i_{tl}) + \sum_{k=t}^{l-1} h^i_k D^i_{kl}, & \text{if } l \geq t+1. \end{cases}$$

The objective function (4.13) minimizes the sum of setup, production, and holding costs. Constraints (4.14) ensure that the demand of each period is covered by only one production interval form period k to period l when $k \in [1, ..., t]$, and $l \in [t, ..., |T|]$. Constraints (4.15) limit the total inventory amount at the end of period t by the inventory bound u_t and constraints (4.16) are the integrality conditions.

4.3 An Extended formulation for the MULSIB

In this section we present a new formulation specially designed to exploit the underlying structure of the MULSIB. In particular, in Section 4.4 we will show that the well-known MCKP appears as a subproblem for the MULSIB when decomposing the problem per period. We note that all formulations introduced in Section 4.2 suffer from the lack of decomposability by periods. This is because both the x and w variables appear multiple times in the IB constraints. This issue can be addressed by adding a new set of variables to determine the inventory levels at each period without having to keep the information of when the items have been produced. In the proposed extended MIP formulation, in addition to the production interval variables w^i_{tl}, for each $i \in M$, $t \in T$, and $l \geq t$ we define additional binary variables z^i_{tl} equal to 1 if and only if there is an inventory of item i to cover all demand from period t until period l. We highlight that, contrary to w^i_{tl} variables, the interval variables z^i_{tl} do not keep the information on when the inventory has been produced. Figure 4.2 shows the relation between w^i_{tl} and associated z^i_{tl} variables for an instance with $w^i_{14} = 1$. Since the cumulative demand from period t_1 to period t_4 of item i is produced at the beginning of period t_1, the variables z^i_{14}, z^i_{24}, z^i_{34}, and z^i_{44} encode the associated inventory levels of item i to cover the demand of the successive periods up to t_4.

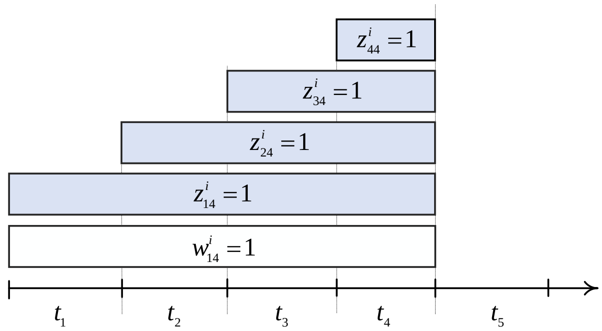


Figure 4.2: The relation between production (w) and inventory (z) variables in (EF)

Using these new set of variables, the MULSIB can be formulated as follows:

$$(\mathbf{EF}) \text{ minimize } \alpha \sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} c^i_{tl} w^i_{tl} + (1-\alpha)\sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} c''^i_{tl} z^i_{tl} \tag{4.17}$$

subject to (4.14)

$$\sum_{l=t}^{|T|} z^i_{tl} = 1 \qquad i \in M, t \in T \tag{4.18}$$

$$\sum_{i\in M}\sum_{l=t+1}^{|T|} D^i_{tl} z^i_{tl} \leq u_t \qquad t \leq |T|-1 \tag{4.19}$$

$$\sum_{k=1}^{t} w^i_{kl} = z^i_{tl} \qquad i \in M, t \in T, l \geq t \tag{4.20}$$

$$z^i_{(t-1)l} + w^i_{tl} = z^i_{tl} \qquad i \in M, t \geq 2, l \geq t \tag{4.21}$$

$$z^i_{tl} \in \{0,1\} \qquad i \in M, t \in T, l \geq t, \tag{4.22}$$

$$0 \leq w^i_{tl} \leq 1 \qquad i \in M, t \in T, l \geq t, \tag{4.23}$$

where we define c''^i_{tl} for $i \in M$, $t \in T$, and $l \geq t$ as:

$$c''^{i}_{tl} = \begin{cases} q^i_t + p^i d^i_t + q^i_{t+1}, & \text{if } t = 1 \text{ and } l = t, \\ q^i_t + p^i d^i_t + h^i_t D^i_{tl}, & \text{if } t = 1 \text{ and } l > t \\ p^i d^i_t + q^i_{l+1}, & \text{if } t > 1 \text{ and } l = t \text{ and } l \neq |T|, \\ p^i d^i_t, & \text{if } t > 1 \text{ and } l = t \text{ and } l = |T|, \\ p^i d^i_t + h^i_t D^i_{tl}, & \text{if } t > 1 \text{ and } l > t, \\ 0, & \text{otherwise.} \end{cases}$$

The objective function (4.17) minimizes the sum of setup, production, and holding costs. The parameter $\alpha \in [0, 1]$ determines the proportion of the cost incurred by the production interval variables w. This parameter is used to include a portion of the cost information in both w and z variables. This is particularly useful later while decomposing EF into independent subproblems.

Constraints (4.18) choose only one inventory interval for each item in each period. Constraints (4.19) are the inventory bound constraint and constraints (4.20)–(4.21) are the linking constraints between sets of production (w) and inventory (z) interval variables. In particular, constraints (4.20) denote that the inventory available in period t, equal to the cumulative demand from periods t to l, must be produced in one of the previous periods, i.e., $[1, ..., t]$. Moreover, constraints (4.21) imply that the inventory interval of item i from period t to period l ($z^i_{tl} = 1$) is either produced in period t, i.e. $w^i_{tl} = 1$, or it was part of the inventory interval at the beginning of period $t-1$, i.e, $z^i_{(t-1)l} = 1$. Finally, constraints (4.22) are the integrality conditions on inventory interval variables and constraints (4.23) are the non-negativity conditions on production interval variables.

An interesting property of EF is that, although z^i_{tl} variables provide redundant information in presence of production interval variables w^i_{tl}, they allow the MULSIB to be decomposed period-by-period. This property will be exploited later in developing decomposition algorithms to obtain strong lower bounds on solutions of the MULSIB.

4.4 Lagrangian relaxation

Lagrangian relaxation (LR) is a well-known method for solving combinatorial optimization problems. The idea of LR is to relax the complicating constraints while maintaining "relatively easy" constraints. This method penalizes the violation of the relaxed constraints in the objective function. In other words, LR exploits the subproblem tractability to obtain lower bounds on the value of the original problem [25]. In the case of model (EF), if we relax constraints (4.14), (4.20) and (4.21) in a Lagrangian fashion, penalizing their violations with multiplier vectors π^1, π^2, and π^3 of appropriate dimension, we obtain the Lagrangian function for the MULSIB as follows:

$$\begin{aligned} L(\pi) = \text{minimize } \; & \alpha \sum_{i \in M} \sum_{t \in T} \sum_{l=t}^{|T|} c_{tl}^{i} w_{tl}^{i} + (1-\alpha) \sum_{i \in M} \sum_{t \in T} \sum_{l=t}^{|T|} c_{tl}^{''i} z_{tl}^{i} + \\ & \sum_{i \in M} \sum_{t \in T} \pi_t^{1i} (1 - \sum_{k=1}^{t} \sum_{l=t}^{|T|} w_{kl}^{i}) + \\ & \sum_{i \in M} \sum_{t \in T} \sum_{l=t}^{|T|} \pi_{tl}^{2i} (z_{tl}^{i} - \sum_{k=1}^{t} w_{kl}^{i}) + \\ & \sum_{i \in M} \sum_{t=2}^{|T|} \sum_{l=t}^{|T|} \pi_{tl}^{3i} (z_{tl}^{i} - z_{(t-1)l}^{i} - w_{tl}^{i}) \end{aligned} \tag{4.24}$$

subject to (4.18), (4.19), (4.22), (4.23).

Note that $L(\pi)$ is separable into two subproblems: (1) a problem in the space of the z variables, and (2) a problem in the space of w variables. The first subproblem can be expressed as:

$$L_z(\pi) = \text{minimize } (1-\alpha)\sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} c''^{i}_{tl} z^i_{tl} + \tag{4.25}$$
$$\sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} \pi^{2i}_{tl} z^i_{tl} + \sum_{i\in M}\sum_{t=2}^{|T|}\sum_{l=t}^{|T|} \pi^{3i}_{tl}(z^i_{tl} - z^i_{(t-1)l})$$

subject to (4.18), (4.19), (4.22).

The second subproblem can be expressed as:

$$L_w(\pi) = \sum_{i\in M}\sum_{t\in T} \pi^{1i}_t + \text{minimize } \alpha\sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} c^i_{tl} w^i_{tl} + \tag{4.26}$$
$$\sum_{i\in M}\sum_{t\in T}\sum_{k=1}^{t}\sum_{l=t}^{|T|} -\pi^{1i}_t w^i_{kl} +$$
$$\sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} -\pi^{2i}_{tl}(\sum_{k=1}^{t} w^i_{kl}) +$$
$$\sum_{i\in M}\sum_{t=2}^{|T|}\sum_{l=t}^{|T|} -\pi^{3i}_{tl}(w^i_{tl})$$

subject to (4.23).

4.4.1 Solution to Lagrangian subproblems

The subproblem $L_z(\pi)$ can be further decomposed into $|T|$ independent problems ($L^t_z(\pi)$), one for each period $t \in T$, of the form:

$$L^t_z(\pi)_{|t=1} = \text{minimize} \sum_{i\in M}\Big(\sum_{l=t}^{|T|}\big((1-\alpha)c''^{i}_{tl} + \pi^{2i}_{tl}\big)z^i_{tl} - \sum_{l=t+1}^{|T|}\pi^{3i}_{(t+1)l} z^i_{tl}\Big) \tag{4.27}$$
$$L^t_z(\pi)_{|1<t<|T|} = \text{minimize} \sum_{i\in M}\Big(\sum_{l=t}^{|T|}\big((1-\alpha)c''^{i}_{tl} + \pi^{2i}_{tl}\big)z^i_{tl} - \sum_{l=t+1}^{|T|}\pi^{3i}_{(t+1)l} z^i_{tl} + \sum_{l=t}^{|T|}\pi^{3i}_{tl} z^i_{tl}\Big)$$
$$L^t_z(\pi)_{|t=|T|} = \text{minimize} \sum_{i\in M}\Big(\sum_{l=t}^{|T|}\big((1-\alpha)c''^{i}_{tl} + \pi^{2i}_{tl}\big)z^i_{tl} + \sum_{l=t}^{|T|}\pi^{3i}_{tl} z^i_{tl}\Big)$$

$$\text{subject to} \quad (4.18), (4.19), (4.22) \text{ fixed at } t.$$

Note that each of the above subproblems is a MCKP and can be efficiently solved to optimality with the exact algorithm developed by Pisinger [82]. In the definition of the MCKP, all coefficients are assumed to be positive and integer. In our implementation, we deal with all negative coefficients by adding a sufficiently large constant to all items in the MCKP. Furthermore, non-integer coefficients are handled by rounding values after multiplying by another sufficiently large factor.

Furthermore, the subproblem $L_w(\pi)$ can be formulated as follows:

$$L_w(\pi) = \sum_{i \in M} \sum_{t \in T} \pi_t^{1i} + \min_{e \in E_w} \sum_{t \in T} \sum_{l=t}^{|T|} f_{tl}^i w_{tl}^{ei}, \tag{4.28}$$

where f_{tl}^i is the coefficient of variable w_{tl}^i after rearranging the elements in (4.25) and E_w is the set of all extreme point associated with constraints (4.23). Note that this subproblem can be easily solved as follows:

$$w_{tl}^{*i} = \begin{cases} 1, & \text{if } f_{tl}^i \leq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, we obtain the following result.

Proposition 1. $L(\pi) = \sum_{t \in T} L_z^t(\pi) + L_w(\pi)$.

Maximizing $L(\pi)$ leads to the best dual bound that can be obtained form the Lagrangian relaxation as follows:

$$(\textbf{LDP}) \;\; z_D = \underset{\pi \in \mathbb{R}^m}{\text{maximize}} \;\; L(\pi).$$

Now in the following subsection, we present two algorithms to solve the associated LDP with the MULSIB.

4.4.2 Solving the Lagrangian dual via subgradient algorithm

We now apply subgradient optimization to solve the LDP associated with the MUS-LIB. Classical subgradient algorithms are known to suffer from slow convergence when solving the LDP. In order to avoid this convergence issue, we propose a *deflected subgradient algorithm* (DSA) based on the methods developed by [19, 25]. The idea behind this method is to consider the combination of the movement direction in the previous iteration (d^{k-1}) and the subgradient of the current iteration (s^k) to obtain the current direction, $d^k = s^k + \theta^k d^{k-1}$, whereas the classical subgradient method uses only the subgradient vector to find the direction, $d^k = s^k$. The effectiveness of DSAs relies on the choice of the deflection parameter θ^k. We use the following rule proposed by [19] to find the deflection parameter:

$$\theta^k = \begin{cases} -\frac{s^k d^{k-1}}{||d^{k-1}||}, & \text{if } s^k d^{k-1} < 0 \\ 0, & \text{otherwise.} \end{cases}$$

For a given vector (π), let z^* and $w*$ denote the optimal solution to $L_z(\pi)$ and $L_w(\pi)$, respectively. Then subgradient vectors of $L(\pi)$ are given by :

$$s_t^{1i}(\pi^k) = 1 - \sum_{k=1}^{t} \sum_{l=t}^{n} w_{tl}^{*ki} \qquad i \in M, t \in T \tag{4.29}$$

$$s_{tl}^{1i}(\pi^k) = z_{tl}^{*ki} - \sum_{k=1}^{t} w_{tl}^{*ki} \qquad i \in M, t \in T, l \geq t \tag{4.30}$$

$$s_{tl}^{2i}(\pi^k) = z_{tl}^{*ki} - w_{tl}^{*ki} - z_{(t-1)l}^{*ki} \qquad i \in M, t \geq 2, l \geq t. \tag{4.31}$$

The proposed DSA is depicted in Algorithm 3.

Algorithm 3: Deflected subgradient algorithm

Input: Parameters β, ϵ, k^{max} and UB (upper bound on the optimal solution value)

0: Initialize $Z_D = -\infty$, $d^0 = 0$; $k = 1$; $\pi^k = 0$;

1: For $t \in T$, solve $PP^t(\pi^k)$ and $L_w^t(\pi^k)$ to find optimal value $v^t(\pi^k)$, z^{*k}, and w^{*k}

2: Evaluate the Lagrangian function $L(\pi^k) = \sum_{t\in T} v^t(\pi^k)$

3: If an improved lower bound is obtained, i.e., $L(\pi^k) > Z_D$, update the bound $Z_D = L(\pi^k)$

4: Evaluate subgradient vectors as (4.29)-(4.31)

5: Convert the subgradient vectors into a single vector $s = vec(s^1, s^2, s^3)$;

6: If $s^k d^{k-1} < 0$, $\theta^k = -\beta \frac{s^k d^{k-1}}{||d^{k-1}||}$. Otherwise, $\theta^k = 0$.

7: Obtain the direction $d^k = s^k + \theta^k d^{k-1}$.

8: Calculate the step length $t^k = \frac{UB - Z_D}{||d^k||}$.

9: Update dual multipliers $\pi^{k+1} = \pi^k + t^k d^k$.

10: If $(\frac{UB - Z_D}{Z_D}) \leq \epsilon$, terminate the algorithm. Otherwise, set $k = k + 1$ and return to Step 1.

4.4.3 Solving the Lagrangian dual via column generation

The LDP can be reformulated as a max-min problem as follows:

$$
\begin{aligned}
z_D = \; & \max_{\pi\in\mathbb{R}^m} \; \min_{w,z} \; \Big\{ \alpha \sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} c_{tl}^{i} w_{tl}^{i} + (1-\alpha)\sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} c_{tl}^{''i} z_{tl}^{i} + \\
& \sum_{i\in M}\sum_{t\in T} \pi_t^{1i}(1 - \sum_{k=1}^{t}\sum_{l=t}^{|T|} w_{kl}^{i}) + \sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} \pi_{tl}^{2i}(z_{tl}^{i} - \sum_{k=1}^{t} w_{kl}^{i}) + \\
& \sum_{i\in M}\sum_{t=2}^{|T|}\sum_{l=t}^{|T|} \pi_{tl}^{3i}(z_{tl}^{i} - z_{(t-1)l}^{i} - w_{tl}^{i}) : (4.18), (4.19), (4.22), (4.23) \Big\} \qquad (4.32) \\
= \; & \max_{\pi\in\mathbb{R}^m} \; \min_{e\in E_w, e'\in E_z} \; \Big\{ \alpha \sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} c_{tl}^{i} w_{tl}^{ei} + (1-\alpha)\sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} c_{tl}^{''i} z_{tl}^{e'i} + \\
& \sum_{i\in M}\sum_{t\in T} \pi_t^{1i}(1 - \sum_{k=1}^{t}\sum_{l=t}^{|T|} w_{kl}^{ei}) + \sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} \pi_{tl}^{2i}(z_{tl}^{e'i} - \sum_{k=1}^{t} w_{kl}^{ei}) + \\
& \sum_{i\in M}\sum_{t=2}^{|T|}\sum_{l=t}^{|T|} \pi_{tl}^{3i}(z_{tl}^{e'i} - z_{(t-1)l}^{e'i} - w_{tl}^{ei}) \Big\}
\end{aligned}
$$

$$
\begin{aligned}
= & \max_{\pi \in \mathbb{R}^m} \eta && (4.33)\\
& \text{subject to } \eta \leq \alpha \sum_{i \in M} \sum_{t \in T} \sum_{l=t}^{|T|} c_{tl}^{i} w_{tl}^{ei} + (1-\alpha) \sum_{i \in M} \sum_{t \in T} \sum_{l=t}^{|T|} c_{tl}^{''i} z_{tl}^{ei} + \\
& \qquad \sum_{i \in M} \sum_{t \in T} \pi_t^{1i} (1 - \sum_{k=1}^{t} \sum_{l=t}^{|T|} w_{kl}^{ei}) + \\
& \qquad \sum_{i \in M} \sum_{t \in T} \sum_{l=t}^{|T|} \pi_{tl}^{2i} (z_{tl}^{ei} - \sum_{k=1}^{t} w_{kl}^{ei}) + \\
& \qquad \sum_{i \in M} \sum_{t=2}^{|T|} \sum_{l=t}^{|T|} \pi_{tl}^{3i} (z_{tl}^{ei} - z_{(t-1)l}^{ei} - w_{tl}^{ei}) \qquad e \in E_w, e' \in E_z, && (4.34)
\end{aligned}
$$

where the variable η is introduced to represent a lower bound on $L(\pi)$. Moreover, E_w is the set of extreme point associated with constraints (4.23) and E_z is the set of all extreme points obtained by convexifying constraints (4.18), (4.19), (4.22). Observe that we have thus represented the LDP as a linear program with an exponential number of constraints. It is well-known that the above reformulated LDP is equivalent to the dual of a *Dantzig-Wolfe reformulation* (DW) in which the set of convexified constraints correspond to the remaining constraints in the $L(\pi)$ [34]. The idea behind the DW is to convexify a set of "relatively easy" constraints and keep other complicating constraints to better describe the integer convex hull implicitly and to provide a tighter bound than the LP relaxation of the original problem. By convexifying constraints, the original variables are replaced by a convex combination of the extreme points of a subspace (E).

The linear relaxation of the equivalent DW of (4.33)–(4.34) can be solved by column generation based methods. A column generation algorithm divides the original linear space of the DW into two or more inter-related problems: a restricted master problem (RMP), which contains a small subset of variables, and one or several pricing problems (PPs). In other words, solving the RMP is equivalent to solving the DW directly with a set of variables equal to zero. The PPs are repeatedly solved to identify additional variables to be added to the RMP if needed. The RMP associated with the DW of (EF) can be formulated as follows:

$$(\textbf{RMP}) \text{ minimize} \sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|} \alpha c_{tl}^{i} w_{tl}^{i} + \sum_{i\in M}\sum_{t\in T}\sum_{l=t}^{|T|}\sum_{e\in E^t}(1-\alpha) c_{tl}^{''i} a_{etl}^{i}\lambda_e^t \tag{4.35}$$

subject to (4.14)

$$\sum_{e\in E^t}\lambda_e^t = 1 \qquad t\in T \tag{4.36}$$

$$\sum_{k=1}^{t} w_{kl}^{i} = \sum_{e\in E^t} a_{etl}^{i}\lambda_e^t \qquad i\in M, t\in T, l\geq t \tag{4.37}$$

$$\sum_{e\in E^{(t-1)}} a_{e(t-1)l}^{i}\lambda_e^{(t-1)} + w_{tl}^{i} = \sum_{e\in E^t} a_{etl}^{i}\lambda_e^t \qquad i\in M, t\geq 2, l\geq t \tag{4.38}$$

$$\lambda_c^t \geq 0 \qquad t\in T, e\in E_t^z. \tag{4.39}$$

In the above RMP, constraints (4.36) are convexity constraints and (4.37)-(4.38) are the reformulation of constraints (4.20)-(4.21), respectively. These reformulated constraints are based on a subset of extreme points of the space created through convexifying constraints (4.18), (4.19), and (4.22). In other words, $E_z^t, t\in T$ is the subset of feasible inventory intervals for the subproblem of the MULSIB associated with period t. For $t\in T$, the PP is formulated exactly as the MCKP presented in Section 4.4.1, i.e. $(L_z^t(\pi))$ or (4.18), (4.19), (4.22), (4.27). Figure 4.3 illustrates an example of how feasible inventory intervals can be structured using the subproblem $(L_z^t(\pi))$.

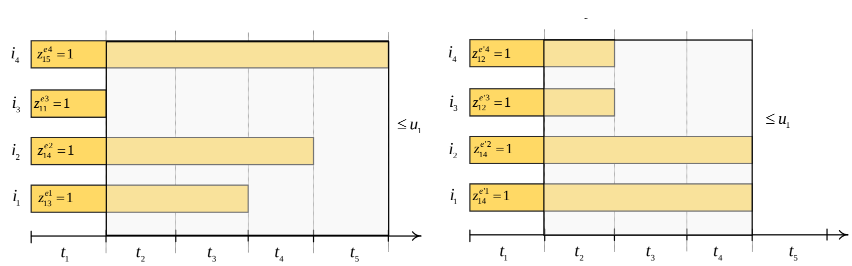


Figure 4.3: Feasible inventory intervals for $t = 1$

We define $\hat{\pi}_t^{1i}$, $\hat{v}_t$, $\hat{\pi}_{tl}^{2i}$, and $\hat{\pi}_{tl}^{3i}$ as the optimal dual solution associated with constraints (4.14), (4.36), (4.37), and (4.38), respectively. After solving each PP, the column λ_q^t with associated non-zeros, i.e., inventory intervals obtained by problem (L_z^t), are added to (RMP) if its reduced cost is negative, or $L_z^t - \hat{v}_t > 0$. Using the block diagonal structure of PPs, we can obtain a valid lower bound after solving (RMP) and associated PPs, $(L_z^t)\ t \in T$, as follows [8, 74]:

$$v(RMP) \geq \sum_{i \in M} \sum_{t \in T} \hat{\pi}_t^{1i} + \sum_{t=1}^{|T|} L_z^t(\pi).$$

At each iteration of the column generation algorithm, the objective value of (RMP), $v(RMP)$, provides an upper bound and $LB(\pi) = \sum_{i \in M} \sum_{t \in T} \hat{\pi}_t^{1i} + \sum_{t=1}^{|T|} L_z^t(\pi)$ provides a valid lower bound.

4.4.3.1 Stabilizing the column generation algorithm using In-Out separation strategy

Stabilization techniques are devised to reduce the convergence issues of column generation algorithms. These issues include dual oscillations (referred to as "bang-bang"), the tailing-off effect (removing only a marginal volume of the dual space), and the degeneracy caused by alternative optimal solutions of (RMP) [8, 80]. We propose a *stabilized column generation algorithm* (SCG) to accelerate the convergence rate. We use an In-Out separation strategy based on the stabilization technique proposed by [15] and [79]. The idea of this method is to keep the dual variables associated with (RMP), π_{RMP}, in the proximity of an incumbent dual solution, denoted by $\bar{\pi}$, that is used as a stability center rather than solving the PP with π_{RMP} solutions, i.e., $\alpha' = 1$ [59]. An implementation of the proposed SCG is shown in Algorithm 4.

Algorithm 4: Stabilized In-Out column generation algorithm

Input: Parameters $\alpha^0 = 0.1$, ϵ, and UB (the value of best knows solutions)

0: Initialize the core stabilization point, $\bar{\pi} = 0$.

Using the core point, for $t \in T$, solve $(L_z^t(\bar{\pi}))$ to calculate the valid lower bound $LB(\bar{\pi}) = \sum_{i \in M} \sum_{t \in T} \bar{\pi}_t^{1i} + \sum_{t=1}^{|T|} L_z^t(\bar{\pi}))$.

1: Initialize (RMP) with artificial variables to ensure the feasibility

2: Solve (RMP) to obtain the value Z_{RMP} and optimal dual solution π_{RMP}.

3: If $Z_{RMP} > UB$, $\alpha' = \alpha^0 \frac{UB - LB(\bar{\pi})}{Z_{RMP} - LB(\bar{\pi})}$. Otherwise, $\alpha' = \alpha^0$.

4: Find a separation point (in the dual space) as follows:

$$\pi_{ST} = \alpha' \pi_{RMP} + (1 - \alpha')\bar{\pi}.$$

5: For $t \in T$, solve $(L_z^t(\pi_{ST}))$ to obtain the column with minimum reduced cost λ.

6: If an improved valid lower bound is obtained, i.e., $LB(\pi_{ST}) > LB(\bar{\pi})$, update $\bar{\pi}$ and L as:

$$\bar{\pi} = \pi_{ST}, \quad LB(\bar{\pi}) = LB(\pi_{ST}).$$

7: If $(\frac{Z_{RMP} - LB(\bar{\pi})}{LB(\bar{\pi})}) \leq \epsilon$, terminate the algorithm. Otherwise, go to the next step.

8: If column λ has a negative reduced cost with respect to π_{RMP}, add it to (RMP) and return to Step 2. Otherwise, return to Step 3.

4.5 Computational experiments

In this section, we present extensive computational experiments to compare the results obtained from the proposed formulation and solution algorithms. All experiments were implemented in C using Callable Library of IBM CPLEX 12.10.0 on an Intel Xeon CPU E5-2687W v3 processor at 3.10 GHz and 750GB of RAM in Linux. After some tuning, we set $\beta = 1.62$ for the DSA. The parameter β is halved every 700 consecutive iterations without improvement in the lower bound and it is reset to the initial value as soon as it gets below 0.001. During the tuning phase, we also noticed that the DSA started with a good convergence rate but the convergence became slow later on. In order to prevent slow convergence, after 1000 iterations, we switch the deflected SA into a non-deflected SA. In both stabilized SCG and DSA algorithms, we set $\epsilon = 0.001$. Moreover, in (EF) we set $\alpha = 0.5$.

In order to evaluate the performance of the proposed formulation (EF), we used the same 60 instances used by [76]. These instances only consider fixed production costs and take different capacities into account. In the first 30 instances, the number of items are set to $|M| = \{15, 30, 45\}$, the number of time periods is $|T| = 50$, and the inventory bounds are set to $u_t = \{500, 1000, 1500\}$. In the other 30 instances, tighter inventory bounds, $u_t = \{375, 500, 1125\}$, are taken into account. The demand parameters d_t^i are integer values between zero and 25, and fixed setup costs q_t^i are integer values between 25 and 150.

Furthermore, in order to assess the performance of the DSA as compared to the SCG, we used an adaptation of the same instances with shorter planning horizons ($|T| = 12$) with tighter capacities $u_t = \{250, 350, 500\}$. Finally, using the best-known solutions obtained from all methods (newly and previously developed methods), we compared the quality of bounds obtained by different methods on the initially used 60 large-sized instances.

4.5.1 Formulations

In this section, we carry on numerical experiments to evaluate the performance of the proposed formulation (EF) as compared to the previously developed formulations. i.e., (CDF), (FLF), and the best method in the literature for the MULSIB proposed by Melo and Ribeiro [76]. Tables 4.1 and 4.2 summarise the result of these comparisons. The first two columns of each table denote the used instances identifying the number of items and number of time periods. For each formulation, we present the obtained $gap = \frac{UB-LB}{LB} \times 100$, upper bound ($UB$), deviation $gap = \frac{UB-Best}{Best} \times 100$, and running time. We also present the best achieved upper bound ($Best$) in the last column. We set a time limit of 3600 seconds (one hour) for solving each formulation. In order to have a fair comparison, we implemented the (FLF) and (CDF) formulations proposed by [76] and [1], respectively, and performed the experiments with our solver.

Table 4.1 Comparing the results obtained by different formulations and the best-known solutions.

Data Info.		*Cumulative-demand formulation*				*Extended formulation*				*Facility location formulation*				[76]'s algorithm				*Best*
Number	*Instance*	*Gap* (%)	*UB* (%)	*Dev* (%)	*Time* (s)	*Gap* (%)	*UB* (%)	*Dev* (%)	*Time* (s)	*Gap* (%)	*UB* (%)	*Dev* (%)	*Time* (s)	*Gap* (%)	*UB* (%)	*Dev* (%)	*Time* (s)	
1	15_50_500	3.04	**9628**	0.0	3600	3.28	9635	0.1	3600	4.43	9722	1.0	3600	3.2	**9628**	0.0	600	9,628
2	15_50_500	3.34	10992	0.4	3600	3.15	**10948**	0.0	3600	3.9	11018	0.6	3600	3.6	11,017	0.6	600	10,948
3	15_50_500	3.27	9811	0.4	3600	2.99	**9771**	0.0	3600	4.29	9883	1.1	3600	3.3	9814	0.4	600	9,771
4	15_50_500	2.97	10159	0.1	3600	3.04	**10153**	0.0	3600	3.97	10230	0.8	3600	3.3	10,186	0.3	600	10,153
5	15_50_500	4.36	9181	1.0	3600	3.46	**9088**	0.0	3600	5.01	9206	1.3	3600	3.7	9119	0.3	600	9,088
6	15_50_500	4.04	11551	0.8	3600	3.76	11522	0.6	3600	4.51	11578	1.1	3600	3.1	**11,457**	0.0	600	11,457
7	15_50_500	3.21	9370	0.2	3600	3.32	**9350**	0.0	3600	3.52	9367	0.2	3600	3.4	9384	0.4	600	9,350
8	15_50_500	3.47	8771	0.7	3600	3.61	8762	0.6	3600	4.34	8812	1.2	3600	2.8	8707	**0.0**	600	8,707
9	15_50_500	2.75	9527	0.0	3600	3.11	**9523**	0.0	3600	3.69	9574	0.5	3600	3.5	9597	0.8	600	9,523
10	15_50_500	3.26	9982	0.3	3600	3.3	9964	0.1	3600	4.15	10014	0.6	3600	3.3	9,954	**0.0**	600	9,954
11	30_50_1000	1.53	18149	0.1	3600	1.45	**18126**	0.0	3600	2.13	18239	0.6	3600	1.6	18,145	0.1	600	18,126
12	30_50_1000	1.45	16655	0.1	3600	1.37	**16633**	0.0	3600	1.72	16677	0.3	3600	1.5	16,649	0.1	600	16,633
13	30_50_1000	1.66	19943	0.1	3600	1.63	**19924**	0.0	3600	1.97	19981	0.3	3600	1.8	19,949	0.1	600	19,924
14	30_50_1000	1.52	**18390**	0.0	3600	1.58	**18390**	0.0	3600	2.09	18470	0.4	3600	1.8	18,434	0.2	600	18,390
15	30_50_1000	1.73	19318	0.2	3600	1.58	**19273**	0.0	3600	2.36	19418	0.8	3600	1.8	19,323	0.3	600	19,273
16	30_50_1000	1.5	18294	0.1	3600	1.48	**18280**	0.0	3600	1.99	18364	0.5	3600	1.7	18,311	0.2	600	18,280
17	30_50_1000	1.46	19943	0.1	3600	1.68	19972	0.2	3600	1.99	20025	0.5	3600	1.5	**19,933**	0.0	600	19,933
18	30_50_1000	1.67	16734	0.2	3600	1.59	**16705**	0.0	3600	2.29	16811	0.6	3600	2	16,765	0.4	600	16,705
19	30_50_1000	1.54	17951	0.1	3600	1.58	**17938**	0.0	3600	2.06	18021	0.5	3600	1.8	17,977	0.2	600	17,938
20	30_50_1000	1.55	18525	0.2	3600	1.46	**18493**	0.0	3600	2.12	18605	0.6	3600	2	18,590	0.5	600	18,493
21	45_50_1500	1.04	**28468**	0.0	3600	1.1	**28468**	0.0	3600	1.29	28518	0.2	3600	1.3	28,522	0.2	600	28,468
22	45_50_1500	1.34	28723	0.3	3600	1.04	**28631**	0.0	3600	1.59	28777	0.5	3600	1.2	28,666	0.1	600	28,631
23	45_50_1500	0.96	28254	0.1	3600	0.89	**28229**	0.0	3600	1.17	28300	0.3	3600	1.4	28,386	0.6	600	28,229
24	45_50_1500	0.88	**25797**	0.0	3600	1	25823	0.1	3600	1.56	25954	0.6	3600	1.5	25,960	0.6	600	25,797
25	45_50_1500	1.2	**30612**	0.0	3600	1.24	30616	0.0	3600	1.82	30786	0.6	3600	1.4	30,651	0.1	600	30,612
26	45_50_1500	1.11	26521	0.2	3600	0.99	**26477**	0.0	3600	1.29	26554	0.3	3600	1.3	26,548	0.3	600	26,477
27	45_50_1500	1.28	26773	0.3	3600	1	**26691**	0.0	3600	1.24	26745	0.2	3600	1.4	26,799	0.4	600	26,691
28	45_50_1500	1.23	30212	0.1	3600	1.11	**30170**	0.0	3600	1.49	30279	0.4	3600	1.4	30,266	0.3	600	30,170
29	45_50_1500	1.01	**28998**	0.0	3600	1.01	**28988**	0.0	3600	1.5	29126	0.5	3600	1.5	29,140	0.5	600	28,988
30	45_50_1500	1	28057	0.0	3600	0.98	**28043**	0.0	3600	1.32	28129	0.3	3600	1.4	28,164	0.4	600	28,043
Average		2.01		0.2		**1.96**		**0.06**		2.56		0.57		2.15		0.28		
# of best solutions			5				**23**				0				3			

According to the results of Table 4.1, (EF) outperforms other formulations by obtaining the best solution for 23 out of 30 instances. The formulation (CDF) has found five best solutions, while three best solutions are reported by the best method developed by [76]. Also, the best average gap and best average deviation have been achieved by (EF). Moreover, the results of Table 4.2 show that (EF) outperforms the other methods even when tighter bounds are considered for the same set of instances. The number of best solutions achieved by (CDF), (EF), (FLF), and [76]' method are zero, 21, zero, and nine, respectively. In terms of average gap and average deviation, (EF) achieves the best ones among all other methods.

Table 4.2 Results of different formulations and the best-known solutions on the instances with tighter capacities.

Data Info.		*Cumulative-demand formulation*				*Extended formulation*				*Facility location formulation*				[76]'s algorithm				*Best*
Number	*Instance*	*Gap* (%)	*UB* (%)	*Dev* (%)	*Time* (s)	*Gap* (%)	*UB* (%)	*Dev* (%)	*Time* (s)	*Gap* (%)	*UB* (%)	*Dev* (%)	*Time* (s)	*Gap* (%)	*UB* (%)	*Dev* (%)	*Time* (s)	
31	15_50_375	3.64	13395	0.6	3600	3.3	13321	0.0	3600	3.44	13328	0.1	3600	3.2	**13,319**	0.0	600	13,319
32	15_50_375	3.18	10523	0.8	3600	2.68	10451	0.1	3600	2.63	**10441**	0.0	3600	2.8	10,469	0.3	600	10,441
33	15_50_375	2.93	10183	0.6	3600	2.73	**10124**	0.0	3600	2.87	10140	0.2	3600	2.6	10,127	0.0	600	10,124
34	15_50_375	3.76	12272	1.0	3600	2.94	12160	0.1	3600	3.33	12191	0.3	3600	2.9	**12,152**	0.0	600	12,152
35	15_50_375	2.98	10629	0.1	3600	3	**10615**	0.0	3600	3.64	10665	0.5	3600	2.9	10,616	0.0	600	10,615
36	15_50_375	3.44	9372	0.7	3600	3.1	**9311**	0.0	3600	3.85	9377	0.7	3600	3.2	9336	0.3	600	9,311
37	15_50_375	3.07	12007	0.7	3600	2.53	**11923**	0.0	3600	2.84	11948	0.2	3600	2.5	11,929	0.1	600	11,923
38	15_50_375	3.18	14503	0.6	3600	2.77	**14418**	0.0	3600	2.94	14424	0.0	3600	2.8	14,422	0.0	600	14,418
39	15_50_375	3.25	13858	0.3	3600	3.11	**13816**	0.0	3600	4.58	14012	1.4	3600	2.9	**13,814**	0.0	600	13,814
40	15_50_375	3.69	13202	1.0	3600	3.29	13131	0.4	3600	3.21	13110	0.3	3600	2.8	**13,075**	0.0	600	13,075
41	30_50_500	1.55	24404	0.3	3600	1.27	24327	0.0	3600	1.7	24420	0.4	3600	1.3	**24,325**	0.0	600	24,325
42	30_50_500	1.62	21998	0.3	3600	1.34	**21927**	0.0	3600	1.85	22032	0.5	3600	1.5	21,967	0.2	600	21,927
43	30_50_500	1.76	26160	0.5	3600	1.28	**26033**	0.0	3600	1.62	26099	0.3	3600	1.3	26,035	0.0	600	26,033
44	30_50_500	1.82	23684	0.2	3600	1.68	23635	0.0	3600	1.94	23692	0.3	3600	1.7	**23,628**	0.0	600	23,628
45	30_50_500	2.02	21480	0.8	3600	1.33	**21320**	0.0	3600	2.04	21466	0.7	3600	1.5	21,355	0.2	600	21,320
46	30_50_500	1.5	25351	0.2	3600	1.32	25299	0.0	3600	1.9	25433	0.6	3600	1.3	**25,292**	0.0	600	25,292
47	30_50_500	1.92	27868	0.5	3600	1.56	27762	0.1	3600	2.43	27982	0.9	3600	1.5	**27,738**	0.0	600	27,738
48	30_50_500	1.85	23465	0.6	3600	1.33	**23327**	0.0	3600	1.74	23410	0.4	3600	1.7	23,402	0.3	600	23,327
49	30_50_500	1.7	23204	0.4	3600	1.44	23135	0.1	3600	1.76	23193	0.4	3600	1.3	**23,109**	0.0	600	23,109
50	30_50_500	1.5	21032	0.4	3600	1.15	**20951**	0.0	3600	1.78	21072	0.6	3600	1.5	21,011	0.3	600	20,951
51	45_50_1125	1.68	35196	0.7	3600	1.03	**34959**	0.0	3600	1.42	35087	0.4	3600	1.2	35,026	0.2	600	34,959
52	45_50_1125	1.14	38410	0.2	3600	0.92	**38317**	0.0	3600	1.03	38346	0.1	3600	1.1	38,364	0.1	600	38,317
53	45_50_1125	0.96	32759	0.2	3600	0.83	**32704**	0.0	3600	1.37	32876	0.5	3600	1.4	32,855	0.5	600	32,704
54	45_50_1125	1.21	36092	0.4	3600	0.83	**35946**	0.0	3600	1.28	36102	0.4	3600	1.2	36,084	0.4	600	35,946
55	45_50_1125	1.28	32064	0.6	3600	0.75	**31883**	0.0	3600	1.39	32083	0.6	3600	1.3	32,068	0.6	600	31,883
56	45_50_1125	1.17	34254	0.3	3600	0.92	**34159**	0.0	3600	1.21	34249	0.3	3600	1.3	34,271	0.3	600	34,159
57	45_50_1125	1.12	34195	0.3	3600	0.83	**34090**	0.0	3600	0.99	34135	0.1	3600	0.9	34,110	0.1	600	34,090
58	45_50_1125	1.18	39176	0.2	3600	1.04	**39105**	0.0	3600	1.21	39170	0.2	3600	1.1	39,131	0.1	600	39,105
59	45_50_1125	1.1	32526	0.1	3600	0.99	**32483**	0.0	3600	1.56	32654	0.5	3600	1.2	32,540	0.2	600	32,483
60	45_50_1125	1.32	36164	0.3	3600	1.05	**36051**	0.0	3600	1.31	36142	0.3	3600	1.1	36,060	0.0	600	36,051
Average		2.08		0.46		**1.74**		**0.03**		2.16		0.40		1.83		0.13		
# of best solutions			0				**21**				0				9			

4.5.2 Column generation vs subgradient optimization

This section compares the quality of lower bounds obtained by Cplex using (EF), the proposed SCG, and DSA. We used, in Table 4.3, 30 adapted instances to find the best bound of Cplex (in one hour), the SCG (within the time limit), the DSA (within time limits of 20, 60, and 80 seconds), and the LP relaxation model. For each method, using the best achieved upper bound by Cplex, we present the gap, the percentage of the LP gap closed (*G-closed*), lower bound, and running time.

From Table 4.3, both the SCG and DSA improved the gap as compared to the bound obtained from LP relaxation in all instances; however, the running times of LP relaxation are shorter. Therefore, although the SCG and DSA take longer, they provide tighter bounds. Furthermore, in larger instances (particularly 21-27, 29, and

Table 4.3 Comparing the quality of lower bounds obtained by different methods on smaller instances

Instance	*Cplex (EF)*					*Stabilized column generation*				*Deflected subgradient*				*LP relaxation*		
	Gap (%)	*G-closed (%)*	*UB*	*LB*	*Time (s)*	*Gap (%)*	*G-closed (%)*	*LB*	*Time (s)*	*Gap (%)*	*G-closed (%)*	*LB*	*Time (s)*	*Gap (%)*	*LB (%)*	*Time (s)*
15_12_250_01	**0.00**	**100**	4167	**4167.0**	**34**	1.38	39	4110.20	16	1.21	46	4117.09	20	2.26	4075.00	0.08
15_12_250_02	**0.00**	**100**	4678	**4678.0**	**2426**	2.30	26	4572.70	20	2.15	31	4579.35	20	3.10	4537.26	0.07
15_12_250_03	**0.00**	**100**	4213	**4213.0**	**99**	1.72	34	4141.90	23	1.63	37	4145.38	20	2.61	4105.87	0.07
15_12_250_04	**0.00**	**100**	4514	**4514.0**	**157**	1.31	33	4455.70	15	1.15	41	4462.60	20	1.97	4426.96	0.07
15_12_250_05	**0.00**	**100**	3951	**3951.0**	**2**	0.85	46	3917.70	19	0.64	60	3925.89	20	1.58	3889.56	0.07
15_12_250_06	**0.00**	**100**	5062	**5062.0**	**77**	1.40	26	4992.20	20	1.24	34	4999.91	20	1.88	4968.44	0.07
15_12_250_07	**0.00**	**100**	3948	**3948.0**	**21**	1.54	32	3888.20	24	1.38	39	3894.30	20	2.25	3860.94	0.08
15_12_250_08	**0.00**	**100**	3866	**3866.0**	**122**	1.61	25	3804.60	19	1.43	33	3811.36	20	2.15	3784.54	0.07
15_12_250_09	**0.00**	**100**	4114	**4114.0**	**15**	1.43	26	4055.80	16	1.27	35	4062.42	20	1.94	4035.76	0.07
15_12_250_10	**0.00**	**100**	4407	**4407.0**	**327**	1.94	28	4323.00	21	1.92	29	4324.09	20	2.69	4291.67	0.07
30_12_350_01	**0.00**	**100**	9502	**9502.0**	**251**	0.25	63	9477.90	236	0.32	54	9471.99	60	0.69	9437.29	0.14
30_12_350_02	**0.00**	**100**	8844	**8844.0**	**1783**	0.19	77	8827.50	167	0.41	49	8807.95	60	0.80	8773.42	0.15
30_12_350_03	**0.25**	**72**	10894	**10867.3**	**limit**	0.41	54	10849.60	134	0.54	40	10835.84	60	0.89	10797.63	0.15
30_12_350_04	**0.08**	**91**	9944	**9935.8**	**limit**	0.17	81	9926.90	117	0.44	50	9900.21	60	0.89	9856.21	0.15
30_12_350_05	**0.00**	**100**	9947	**9947.0**	**352**	0.36	57	9911.80	218	0.38	54	9909.63	60	0.82	9865.93	0.15
30_12_350_06	**0.15**	**82**	9826	**9811.0**	**limit**	0.25	70	9801.50	188	0.47	44	9780.35	60	0.84	9744.59	0.15
30_12_350_07	**0.00**	**100**	10708	**10708**	**1148**	0.31	68	10674.40	160	0.36	63	10669.44	60	0.98	10604.29	0.16
30_12_350_08	**0.00**	**100**	8921	**8921.0**	**34**	0.14	76	8908.50	172	0.17	70	8905.79	60	0.57	8870.08	0.15
30_12_350_09	**0.31**	**62**	9474	**9445.1**	**limit**	0.32	60	9443.40	175	0.52	36	9425.19	60	0.81	9397.55	0.16
30_12_350_10	**0.00**	**100**	9872	**9872.0**	**2786**	0.26	67	9846.70	138	0.41	48	9832.02	60	0.78	9795.21	0.15
45_12_500_01	**0.00**	**100**	15491	**15491**	**2040**	0.04	86	15484.10	1001	0.14	58	15469.35	80	0.33	15440.11	0.23
45_12_500_02	0.21	58	16076	16041.5	limit	**0.11**	**79**	**16058.70**	**711**	0.29	42	16028.73	80	0.50	15995.33	0.24
45_12_500_03	0.14	69	15784	15761.5	limit	**0.09**	**80**	**15769.80**	**1199**	0.17	62	15756.84	80	0.46	15712.49	0.24
45_12_500_04	0.15	70	14299	14277.0	limit	**0.01**	**99**	**14298.00**	**1025**	0.23	55	14266.88	80	0.50	14228.41	0.25
45_12_500_05	0.16	65	16990	16962.1	limit	**0.05**	**89**	**16981.40**	**767**	0.29	36	16940.41	80	0.45	16913.18	0.25
45_12_500_06	0.21	57	14353	14323.3	limit	**0.12**	**75**	**14335.40**	**1013**	0.25	50	14317.68	80	0.49	14282.95	0.23
45_12_500_07	0.28	51	14602	14561.3	limit	**0.14**	**75**	**14580.90**	**1154**	0.32	43	14554.99	80	0.57	14519.08	0.22
45_12_500_08	**0.00**	**100**	16564	**16564.0**	**735**	0.09	72	16548.40	960	0.17	49	16535.91	80	0.33	16508.97	0.25
45_12_500_09	0.18	63	16088	16059.8	limit	**0.01**	**97**	**16086.00**	**645**	0.29	39	16040.77	80	0.48	16010.71	0.23
45_12_500_10	0.25	50	15076	15038.8	limit	**0.02**	**95**	**15072.40**	**800**	0.26	47	15036.19	80	0.50	15000.83	0.25
Average	0.08	86				0.63	61			0.68	46			1.17		
# of best lower bounds			22				8				0				0	

30), the SCG achieves better lower bounds than Cplex (even after running for one hour) within much shorter running times. For the instances 21-30, on average, the SCG closes 85 % of the integrality gap, while the mean closed gap obtained by Cplex for the instances with 45 items is 68 %. In all instance, although the SCG finds better bounds than the DSA, both algorithms obtain mean gaps very close together (0.63 and 0.68) but the running times for the DSA are significantly shorter.

4.5.3 Comparing lower bounds on larger instances

This section assesses the quality of lower bounds obtained by LP relaxation, Cplex, the DSA, and the best bound obtained by [76]. It should be noted, although the SCG provided promising results for the adapted instances in Section 4.5.2, for the original instances with $|T| = 50$, it was not able to improve LP-bounds within reasonable amounts of time; however, the DSA was successful in providing tighter bounds in significantly shorter running times. Therefore in this section, for original instances used in Section 4.5.1, we evaluate only the bounds obtained by the DSA. Tables 4.4 and 4.5 summarize the results of this assessment. In each table, we present the integrality gap and running time for LP relaxation. We also preset the percentage of the integrality gap closed by Cplex using (EF), the DSA and [76]' algorithm. We should note that the best known solutions obtained in tables 4.2 and 4.3 are used to find initial upper bounds in the DSA.

Table 4.4 Comparing the quality of lower bounds obtained by different methods.

Data Info.		*LP relaxation*		*Cplex (EF)*		*Deflected subgradient*		[76]'s algorithm	
Number	*Instance*	*gap* (%)	*Time* (s)	*G-closed* (%)	*Time* (s)	*G-closed* (%)	*Time* (s)	*G-closed* (%)	*Time* (s)
1	15_50_500	3.99	7	**20**	**3600**	18	600	20	600
2	15_50_500	3.68	10	14	3600	16	600	**20**	**600**
3	15_50_500	3.70	10	19	3600	20	600	**23**	**600**
4	15_50_500	3.60	9	15	3600	16	600	**18**	**600**
5	15_50_500	4.34	9	20	3600	20	600	**23**	**600**
6	15_50_500	3.87	8	18	3600	16	600	**20**	**600**
7	15_50_500	3.82	10	13	3600	11	600	**21**	**600**
8	15_50_500	3.58	10	17	3600	20	600	**22**	**600**
9	15_50_500	3.67	9	15	3600	9	600	**26**	**600**
10	15_50_500	4.02	9	**21**	**3600**	20	600	18	600
11	30_50_1000	1.58	30	8	3600	**22**	**750**	5	600
12	30_50_1000	1.56	28	12	3600	**26**	**750**	10	600
13	30_50_1000	1.77	30	8	3600	**15**	**750**	5	600
14	30_50_1000	1.79	28	12	3600	**23**	**750**	13	600
15	30_50_1000	1.69	27	7	3600	**18**	**750**	9	600
16	30_50_1000	1.64	27	10	3600	**21**	**750**	7	600
17	30_50_1000	1.59	33	7	3600	**19**	**750**	6	600
18	30_50_1000	1.75	27	9	3600	**21**	**750**	6	600
19	30_50_1000	1.68	30	6	3600	**19**	**750**	6	600
20	30_50_1000	1.60	25	9	3600	**20**	**750**	9	600
21	45_50_1500	1.16	51	5	3600	**19**	**900**	4	600
22	45_50_1500	1.11	48	7	3600	**20**	**900**	3	600
23	45_50_1500	0.96	41	7	3600	**24**	**900**	12	600
24	45_50_1500	0.98	48	8	3600	**30**	**900**	12	600
25	45_50_1500	1.30	50	6	3600	**10**	**900**	2	600
26	45_50_1500	1.04	53	5	3600	**21**	**900**	1	600
27	45_50_1500	1.06	45	6	3600	**20**	**900**	7	600
28	45_50_1500	1.17	47	5	3600	**17**	**900**	8	600
29	45_50_1500	1.06	47	4	3600	**20**	**900**	8	600
30	45_50_1500	1.05	47	7	3600	**22**	**900**	8	600
Average				11		**19**		12	
# of best lower bounds				2		**20**		8	

Table 4.5 The quality of lower bounds obtained by different methods on the instances with tighter capacities.

Data Info.		*LP relaxation*		*Cplex (EF)*		*Deflected subgradient*		[76]'s algorithm	
Number	*Instance*	*gap* (%)	*Time* (s)	*G-closed* (%)	*Time* (s)	*G-closed* (%)	*Time* (s)	*G-closed* (%)	*Time* (s)
31	15_50_375	3.95	10	17	3600	18	600	**19**	600
32	15_50_375	3.18	9	19	3600	**23**	**600**	21	600
33	15_50_375	3.39	9	19	3600	22	600	**24**	**600**
34	15_50_375	3.49	10	18	3600	**18**	**600**	17	600
35	15_50_375	3.53	8	15	3600	**19**	**600**	18	600
36	15_50_375	3.72	9	17	3600	18	600	**21**	**600**
37	15_50_375	3.04	11	17	3600	19	600	**19**	**600**
38	15_50_375	3.40	7	**19**	**3600**	17	600	19	600
39	15_50_375	3.60	9	14	3600	18	600	**20**	**600**
40	15_50_375	3.33	10	14	3600	**16**	**600**	16	600
41	30_50_500	1.40	30	10	3600	**19**	**750**	7	600
42	30_50_500	1.48	28	9	3600	**20**	**750**	11	600
43	30_50_500	1.46	29	13	3600	**20**	**750**	12	600
44	30_50_500	1.80	29	8	3600	**17**	**750**	6	600
45	30_50_500	1.45	29	9	3600	**18**	**750**	8	600
46	30_50_500	1.43	29	10	3600	**19**	**750**	9	600
47	30_50_500	1.62	29	9	3600	**16**	**750**	7	600
48	30_50_500	1.50	29	11	3600	**19**	**750**	8	600
49	30_50_500	1.49	29	11	3600	**15**	**750**	13	600
50	30_50_500	1.32	29	13	3600	**25**	**750**	8	600
51	45_50_1125	1.09	44	6	3600	**18**	**900**	8	600
52	45_50_1125	1.01	43	9	3600	**21**	**900**	3	600
53	45_50_1125	0.88	46	5	3600	**21**	**900**	0	600
54	45_50_1125	0.88	41	6	3600	**19**	**900**	8	600
55	45_50_1125	0.81	37	8	3600	**25**	**900**	12	600
56	45_50_1125	1.00	42	8	3600	**19**	**900**	3	600
57	45_50_1125	0.89	46	7	3600	**21**	**900**	5	600
58	45_50_1125	1.09	42	4	3600	**13**	**900**	5	600
59	45_50_1125	1.08	38	8	3600	**20**	**900**	5	600
60	45_50_1125	1.11	42	5	3600	**14**	**900**	3	600
Average				11		**19**		11	
# of best lower bounds				1		**24**		5	

As can be observed from Table 4.4, the DSA has found better lower bounds in 20 out of 30 instances, while Cplex (EF) has obtained best bounds for only two instances and the [76]'s algorithm has reported better lower bounds in eight instances. It should be noted that the results of the DSA have been achieved in significantly shorter amounts of time than Cplex. Furthermore, on average, the DSA has closed the integrality gap almost twice as much as other two methods. Moreover, Table 4.5 shows that, the DSA was able to obtain the best bound in 24 instances, while Cplex, even after one hour, could find the best lower bound for only one instance. Also, the best algorithm of [76] obtained better bounds in only five instances. The average closed gap for the DSA is the highest among all methods. Therefore in general, the results obtained by the proposed DSA outperform the results of other methods.

4.6 Conclusions

We studied the multi-item uncapacitated lot-sizing problem with inventory bounds and proposed a new formulation based on production and inventory interval variables. The new formulation allowed us to decompose the problem into smaller subproblems that can be solved efficiently. We proposed a Lagrangian relaxation approach to handle complicating constraints of the proposed formulation. In order to solve the associated Lagrangian dual problem, we proposed two methods based on subgradient optimization and column generation. In both methods, we used stabilization techniques to accelerate the convergence of the algorithms and to reduce the dual oscillation effects.

Computational results confirm that the proposed formulation outperforms the previously proposed models and methods. In almost 75 % of the used instances, our formulation achieved better integer solutions. Moreover, we have shown that in a set of adapted instances with a shorter planning horizon, the stabilized column generation algorithm outperforms Cplex and the deflected subgradient algorithm on the instances with larger number of items and both of the proposed algorithms achieved better lower bounds than those of the linear programming relaxation approach. Although the stabilized column generation algorithm provided promising results for adapted instances, even with stabilization techniques, it was too slow to converge for original instances. Furthermore, we showed that our proposed deflected subgradient algorithm obtained the best lower bound in 70 % of the original instances and, on average, it closed the integrality gap almost twice as much as the other developed methods.

As an interesting direction of research for future work, one can develop a branch and cut algorithm to solve the problem that exploits non-dominated cuts describing the convex hull. It is also interesting to investigate decomposition-based algorithms for the problem with both production and inventory capacities.

Chapter 5

Conclusions

This thesis studied three important problems in operations management:

- Online multi-appointment patient scheduling.
- Inventory management of perishable products.
- Multi-item uncapacitated lot-sizing with inventory bounds.

In Chapter 2, we addressed several critical and challenging issues arising in scheduling patients for chemotherapy treatments. These include dynamically arriving appointment requests, primary-care delivery, patient preferences, fully occupation of nurses in setup phase, and unexpected last-minute changes. To tackle these complexities, a flexible and adaptive scheduling procedure was proposed that schedules incoming appointment requests, and reschedules these on a daily basis when either new information regarding the request is received or an unexpected last-minute change occurs. Two mixed integer programming models are used in the proposed procedure. Based on the obtained computational results, we showed adopting the proposed procedure would allow oncology clinics to provide better patient care and utilize available resource more efficiently.

In Chapter 3, we addressed three aspects of perishable inventory systems: perishability, demand uncertainty, and order modification of previously placed orders. We formulated the problem as a two-stage robust optimization model with a budget of uncertainty to control the level of conservatism. We solved the problem via an exact robust algorithm based on the column-and-row generation method. Computational analysis demonstrated the capability of the proposed robust approach in solving different test instances. We showed that considering order modification could significantly reduce the total cost. We also carried out different sensitivity analyses to provide managerial insights. Our robust algorithm had an acceptable performance in risk-neutral settings and it provided the best performance in risk-averse settings as compared to the deterministic and stochastic variants of the problem.

Finally, Chapter 4 presented a new formulation and two decomposition-based algorithms to solve the multi-item uncapacitated lot-sizing problem. The new formulation was based on both production and inventory interval variables and it allowed us to decompose the problem into smaller and relatively tractable subproblems. We proposed a Lagrangian relaxation approach to handle complicating constraints and we solved it by subgradient and column generation algorithms. We used stabilization techniques to accelerate the convergence of the algorithms. Using benchmark instances, the proposed formulation showed to be significantly better than previously developed formulations. We also showed the proposed decomposition-based algorithm is superior in closing optimality gaps and finding tighter bounds.

We believe that there are several topics that are yet to be addressed in the context of each presented problem. The integration of the procedure presented in Chapter 2 with an accurate prediction tool, that estimates the combination of patients and their characteristics in advance, could be an interesting research avenue for this topic. Furthermore, from the modelling perspective, the problem can be extended by considering uncertain drug preparation times. For the problem studied in Chapter 3, it would be interesting to consider multiple types of items and the substitution assumption for items. Considering multiple demand classes and the possibility of reusing the expired items in the primary or a secondary market can open new lines of research. Finally, investigating other decomposition-based algorithms for the extended versions of the problem presented in Chapter 4 can be a promising research direction.

Bibliography

[1] A. Acevedo-Ojeda. *Multi-level production planning with raw-material perishability and inventory bounds*. PhD thesis, Concordia University, 2019.

[2] A. Ahmadi-Javid, Z. Jalali, and K. J. Klassen. Outpatient appointment systems in healthcare: A review of optimization studies. *European Journal of Operational Research*, 258(1):3–34, 2017.

[3] K. Akartunalı and A. J. Miller. A computational analysis of lower bounds for big bucket production planning problems. *Computational Optimization and Applications*, 53(3):729–753, 2012.

[4] A. Akbalik, B. Penz, and C. Rapine. Capacitated lot sizing problems with inventory bounds. *Annals of Operations Research*, 229(1):1–18, 2015.

[5] A. Akbalik, B. Penz, and C. Rapine. Multi-item uncapacitated lot sizing problem with inventory bounds. *Optimization Letters*, 9(1):143–154, 2015.

[6] A. Akbalık, S. Kebe, B. Penz, and N. Sbihi. Exact methods and a heuristic for the optimization of an integrated replenishment-storage planning problem. *International Transactions in Operational Research*, 15(2):195–214, 2008.

[7] M. Alvarado and L. Ntaimo. Chemotherapy appointment scheduling under uncertainty using mean-risk stochastic integer programming. *Health care management science*, 21(1):87–104, 2018.

[8] H. M. B. Amor, J. Desrosiers, and A. Frangioni. On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157(6): 1167–1184, 2009.

[9] T. Aouam and N. Brahimi. Integrated production planning and order acceptance under uncertainty: A robust optimization approach. *European Journal of Operational Research*, 228(3):504–515, 2013.

[10] A. Atamtürk and S. Küçükyavuz. Lot sizing with inventory bounds and fixed costs: Polyhedral study and computation. *Operations Research*, 53(4):711–730, 2005.

[11] A. Atamtürk and S. Küçükyavuz. An o(n^2) algorithm for lot sizing with inventory bounds and fixed costs. *Operations Research Letters*, 36(3):297–299, 2008.

[12] Z. T. Balkhi and L. Benkherouf. On an inventory model for deteriorating items with stock dependent and time-varying demand rates. *Computers & Operations Research*, 31(2):223–240, 2004.

[13] E. Balugani, F. Lolli, R. Gamberini, B. Rimini, and M. Babai. A periodic inventory system of intermittent demand items with fixed lifetimes. *International Journal of Production Research*, pages 1–13, 2019.

[14] I. Barany, T. Van Roy, and L. A. Wolsey. Uncapacitated lot-sizing: The convex hull of solutions. In *Mathematical programming at Oberwolfach II*, pages 32–43. Springer, 1984.

[15] W. Ben-Ameur and J. Neto. Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks: An International Journal*, 49(1):3–17, 2007.

[16] A. Ben-Tal, B. Golany, and S. Shtern. Robust multi-echelon multi-period inventory control. *European Journal of Operational Research*, 199(3):922–935, 2009.

[17] M. Benzaid, N. Lahrichi, and L.-M. Rousseau. Chemotherapy appointment scheduling and daily outpatient–nurse assignment. *Health care management science*, pages 1–17, 2019. doi: `doi.org/10.1007/s10729-018-9462-6`.

[18] D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Operations research*, 54(1):150–168, 2006.

[19] P. M. Camerini, L. Fratta, and F. Maffioli. On improving relaxation methods by modified gradient techniques. In *Nondifferentiable optimization*, pages 26–34. Springer, 1975.

[20] Canadian-Cancer-Society. Canadian cancer statistics: A 2018 special report, 2018.

[21] J. Castaing, A. Cohn, B. T. Denton, and A. Weizer. A stochastic programming approach to reduce patient wait times and overtime in an outpatient infusion center. *IIE Transactions on Healthcare Systems Engineering*, 6(3):111–125, 2016.

[22] T. Cayirli and E. Veral. Outpatient scheduling in health care: a review of literature. *Production and operations management*, 12(4):519–549, 2003.

[23] X. Chao, X. Gong, C. Shi, and H. Zhang. Approximation algorithms for perishable inventory systems. *Operations Research*, 63(3):585–601, 2015.

[24] A. Condotta and N. Shakhlevich. Scheduling patient appointments via multilevel template: A case study in chemotherapy. *Operations Research for Health Care*, 3(3):129–144, 2014.

[25] I. Contreras, J.-F. Cordeau, and G. Laporte. The dynamic uncapacitated hub location problem. *Transportation Science*, 45(1):18–32, 2011.

[26] I. Contreras, J.-F. Cordeau, and G. Laporte. Stochastic uncapacitated hub location. *European Journal of Operational Research*, 212(3):518–528, 2011.

[27] P. Cunha, F. M. Raupp, and F. Oliveira. A two-stage stochastic programming model for periodic replenishment control system under demand uncertainty. *Computers & Industrial Engineering*, 107:313–326, 2017.

[28] M. Di Summa and L. A. Wolsey. Lot-sizing with stock upper bounds and fixed charges. *SIAM Journal on Discrete Mathematics*, 24(3):853–875, 2010.

[29] M. Dillon, F. Oliveira, and B. Abbasi. A two-stage stochastic programming model for inventory management in the blood supply chain. *International Journal of Production Economics*, 187:27–41, 2017.

[30] J. Du and J. Y.-T. Leung. Minimizing total tardiness on one machine is np-hard. *Mathematics of operations research*, 15(3):483–495, 1990.

[31] G. O. Ferreira, E. F. Arruda, and L. G. Marujo. Inventory management of perishable items in long-term humanitarian operations using markov decision processes. *International journal of disaster risk reduction*, 31:460–469, 2018.

[32] K. Fu, X. Gong, and G. Liang. Managing perishable inventory systems with product returns and remanufacturing. *Production and Operations Management*, 28(6):1366–1386, 2019.

[33] T. Garaix, S. Rostami, and X. Xie. Daily outpatient chemotherapy appointment scheduling with random deferrals. *Flexible Services and Manufacturing Journal*, pages 1–25, 2019. doi: `doi.org/10.1007/s10696-018-9326-x`.

[34] A. M. Geoffrion. *Lagrangian relaxation for integer programming*. Springer, 1974.

[35] B. C. Giri. Managing inventory with two suppliers under yield uncertainty and risk aversion. *International Journal of Production Economics*, 133(1):80–85, 2011.

[36] Y. Gocgun and M. L. Puterman. Dynamic scheduling with due dates and time windows: an application to chemotherapy patient appointment booking. *Health care management science*, 17(1):60–76, 2014.

[37] M. Gooch and F. Abdel. The cost of canada's annual food waste, 2014. https://vcm-international.com.

[38] N. Gülpınar and B. Rustem. Robust optimal decisions with imprecise forecasts. *Computational statistics & data analysis*, 51(7):3595–3611, 2007.

[39] S. Gunpinar and G. Centeno. Stochastic integer programming models for reducing wastages and shortages of blood products at hospitals. *Computers & Operations Research*, 54:129–141, 2015.

[40] D. Gupta and B. Denton. Appointment scheduling in health care: Challenges and opportunities. *IIE transactions*, 40(9):800–819, 2008.

[41] J. Gutiérrez, M. Colebrook, B. Abdul-Jalbar, and J. Sicilia. Effective replenishment policies for the multi-item dynamic lot-sizing problem with storage capacities. *Computers & operations research*, 40(12):2844–2851, 2013.

[42] S. Hahn-Goldberg, M. W. Carter, J. C. Beck, M. Trudeau, P. Sousa, and K. Beattie. Dynamic optimization of chemotherapy outpatient scheduling with uncertainty. *Health care management science*, 17(4):379–392, 2014.

[43] R. Haijema. Optimal issuing of perishables with a short fixed shelf life. In *International Conference on Computational Logistics*, pages 160–169. Springer, 2011.

[44] R. Haijema. Optimal ordering, issuance and disposal policies for inventory management of perishable products. *International Journal of Production Economics*, 157:158–169, 2014.

[45] R. Haijema, J. van der Wal, and N. M. van Dijk. Blood platelet production: Optimization by dynamic programming and simulation. *Computers & Operations Research*, 34(3):760–779, 2007.

[46] R. Haijema, N. van Dijk, J. van der Wal, and C. S. Sibinga. Blood platelet production with breaks: optimization by sdp and simulation. *International Journal of Production Economics*, 121(2):464–473, 2009.

[47] E. M. T. Hendrix, R. Haijema, R. Rossi, and K. G. Pauls-Worm. On solving a stochastic programming model for perishable inventory control. In *International Conference on Computational Science and Its Applications*, pages 45–56. Springer, 2012.

[48] M. Heshmat, K. Nakata, and A. Eltawil. Solving the patient appointment scheduling problem in outpatient chemotherapy clinics using clustering and mathematical programming. *Computers & Industrial Engineering*, 124:347–358, 2018.

[49] P. Hooshangi-Tabrizi, I. Contreras, N. Bhuiyan, and G. Batist. Improving patient-care services at an oncology clinic using a flexible and adaptive scheduling procedure. *Expert Systems with Applications*, page 113267, 2020.

[50] R. Horst. On the global minimization of concave functions. *Operations Research Spectrum*, 6(4):195–205, 1984.

[51] S.-M. Hosseini-Motlagh, M. R. G. Samani, and S. Homaei. Blood supply chain management: robust optimization, disruption risk, and blood group compati-

bility (a real-life case). *Journal of Ambient Intelligence and Humanized Computing*, pages 1–20, 2019.

[52] P. J. Hulshof, N. Kortbeek, R. J. Boucherie, E. W. Hans, and P. J. Bakker. Taxonomic classification of planning decisions in health care: a structured review of the state of the art in or/ms. *Health systems*, 1(2):129–175, 2012.

[53] H.-C. Hwang and W. van den Heuvel. Improved algorithms for a lot-sizing problem with inventory bounds and backlogging. *Naval Research Logistics (NRL)*, 59(3-4):244–253, 2012.

[54] H.-C. Hwang, W. van den Heuvel, and A. P. Wagelmans. The economic lot-sizing problem with lost sales and bounded inventory. *IIE Transactions*, 45(8): 912–924, 2013.

[55] E. Iakovou, D. Vlachos, and A. Xanthopoulos. A stochastic inventory management model for a dual sourcing supply chain with disruptions. *International Journal of Systems Science*, 41(3):315–324, 2010.

[56] K. Inderfurth, P. Kelle, and R. Kleber. Inventory control in dual sourcing commodity procurement with price correlation. *Central European Journal of Operations Research*, 26(1):93–119, 2018.

[57] K. Kanchanasuntorn and A. Techanitisawad. An approximate periodic model for fixed-life perishable products in a two-echelon inventory–distribution system. *International Journal of Production Economics*, 100(1):101–115, 2006.

[58] S. M. Kazemi, M. Rabbani, R. Tavakkoli-Moghaddam, and F. A. Shahreza. Blood inventory-routing problem under uncertainty. *Journal of Intelligent & Fuzzy Systems*, 32(1):467–481, 2017.

[59] J. E. Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[60] I. Konstantaras and K. Skouri. A note on a production-inventory model under stock-dependent demand, weibull distribution deterioration, and shortage. *International Transactions in Operational Research*, 18(4):527–531, 2011.

[61] C. Kouki and O. Jouini. On the effect of lifetime variability on the performance of inventory systems. *International Journal of Production Economics*, 167:23–34, 2015.

[62] C. Kouki, Z. Jemai, and S. Minner. A lost sales (r, q) inventory control model for perishables with fixed lifetime and lead time. *International Journal of Production Economics*, 168:143–157, 2015.

[63] C. Kouki, M. Z. Babai, Z. Jemai, and S. Minner. A coordinated multi-item inventory system for perishables with random lifetime. *International Journal of Production Economics*, 181:226–237, 2016.

[64] C. Kouki, M. Z. Babai, and S. Minner. On the benefit of dual-sourcing in managing perishable inventory. *International Journal of Production Economics*, 204:1–17, 2018.

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

[66] G. Lamé, O. Jouini, and J. Stal-Le Cardinal. Outpatient chemotherapy planning: A literature review with insights from a case study. *IIE Transactions on Healthcare Systems Engineering*, 6(3):127–139, 2016.

[67] H. L. Lee and C. S. Tang. Socially and environmentally responsible value chain innovations: New operations management research opportunities. *Management Science*, 64(3):983–996, 2018.

[68] Z. Lian, L. Liu, and M. F. Neuts. A discrete-time model for common lifetime inventory systems. *Mathematics of Operations Research*, 30(3):718–732, 2005.

[69] Z. Lian, X. Liu, and N. Zhao. A perishable inventory model with markovian renewal demands. *International Journal of Production Economics*, 121(1):176–182, 2009.

[70] B. Liang and A. Turkcan. Acuity-based nurse assignment and patient scheduling in oncology clinics. *Health care management science*, 19(3):207–226, 2016.

[71] B. Liang, A. Turkcan, M. E. Ceyhan, and K. Stuart. Improvement of chemotherapy patient flow and scheduling in an outpatient oncology clinic. *International Journal of Production Research*, 53(24):7177–7190, 2015.

[72] E. Liu, X. Ma, A. Sauré, L. Weber, M. L. Puterman, and S. Tyldesley. Improving access to chemotherapy through enhanced capacity planning and patient scheduling. *IISE Transactions on Healthcare Systems Engineering*, pages 1–13, 2019. doi: 10.1080/24725579.2018.1442376.

[73] S. F. Love. Bounded production and inventory models with piecewise concave costs. *Management Science*, 20(3):313–318, 1973.

[74] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations research*, 53(6):1007–1023, 2005.

[75] R. T. Marler and J. S. Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, 41 (6):853–862, 2010.

[76] R. A. Melo and C. C. Ribeiro. Formulations and heuristics for the multi-item uncapacitated lot-sizing problem with inventory bounds. *International Journal of Production Research*, 55(2):576–592, 2017.

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

[78] Y. Park. An integrated approach for production and distribution planning in supply chain management. *International Journal of Production Research*, 43 (6):1205–1224, 2005.

[79] A. Pessoa, E. Uchoa, M. P. De Aragão, and R. Rodrigues. Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. *Mathematical Programming Computation*, 2(3-4):259–290, 2010.

[80] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck. Automation and combination of linear-programming based stabilization techniques in column generation. *INFORMS Journal on Computing*, 30(2):339–360, 2018.

[81] S.-L. Phouratsamay, S. Kedad-Sidhoum, and F. Pascual. Two-level lot-sizing with inventory bounds. *Discrete Optimization*, 30:1–19, 2018.

[82] D. Pisinger. A minimal algorithm for the multiple-choice knapsack problem. *European Journal of Operational Research*, 83(2):394–410, 1995.

[83] Y. Pochet and L. A. Wolsey. Polyhedra for lot-sizing with wagner—whitin costs. *Mathematical Programming*, 67(1-3):297–323, 1994.

[84] Y. Pochet and L. A. Wolsey. *Production planning by mixed integer programming*. Springer Science & Business Media, 2006.

[85] K. Puranam, D. C. Novak, M. T. Lucas, and M. Fung. Managing blood inventory with multiple independent sources of supply. *European Journal of Operational Research*, 259(2):500–511, 2017.

[86] R. Qiu and J. Shang. Robust optimisation for risk-averse multi-period inventory decision with partial demand distribution information. *International Journal of Production Research*, 52(24):7472–7495, 2014.

[87] R. Qiu, M. Sun, and Y. F. Lim. Optimizing (s, S) policies for multi-period inventory models with demand distribution uncertainty: Robust dynamic programing approaches. *European Journal of Operational Research*, 261(3):880–892, 2017.

[88] S. Rajendran and A. R. Ravindran. Platelet ordering policies at hospitals using stochastic integer programming model and heuristic approaches to reduce wastage. *Computers & Industrial Engineering*, 110:151–164, 2017.

[89] C. Ramos, A. Cataldo, and J.-C. Ferrer. Appointment and patient scheduling in chemotherapy: a case study in chilean hospitals. *Annals of Operations Research*, pages 1–29, 2018. doi: `doi.org/10.1007/s10479-018-3085-7`.

[90] R. T. Rockafellar. *Convex analysis*, volume 28. Princeton university press, 1970.

[91] T. L. Saaty. How to make a decision: the analytic hierarchy process. *European journal of operational research*, 48(1):9–26, 1990.

[92] A. Sadki, X. Xie, and F. Chauvin. Appointment scheduling of oncology outpatients. In *Automation Science and Engineering (CASE), 2011 IEEE Conference on*, pages 513–518. IEEE, 2011. doi: `doi: 10.1109/CASE.2011.6042417`.

[93] F. S. P. Santos and F. Oliveira. An enhanced l-shaped method for optimizing

periodic-review inventory control problems modeled via two-stage stochastic programming. *European Journal of Operational Research*, 275(2):677–693, 2019.

[94] M. C. Santos, A. Agra, and M. Poss. Robust inventory theory with perishable products. *Annals of Operations Research*, pages 1–22, 2019.

[95] A. Saure, J. Patrick, S. Tyldesley, and M. L. Puterman. Dynamic multi-appointment patient scheduling for radiation therapy. *European Journal of Operational Research*, 223(2):573–584, 2012.

[96] C.-T. See and M. Sim. Robust approximation to multiperiod inventory management. *Operations research*, 58(3):583–594, 2010.

[97] A. Sheopuri, G. Janakiraman, and S. Seshadri. New policies for the stochastic inventory control problem with two supply sources. *Operations research*, 58(3): 734–745, 2010.

[98] G. Sher. Year end review of canadian blood services, 2018. https://blood.ca.

[99] S. Suss, N. Bhuiyan, K. Demirli, and G. Batist. Achieving level patient flow in an outpatient oncology clinic. *IISE Transactions on Healthcare Systems Engineering*, 8(1):47–58, 2018.

[100] A. Thorsen and T. Yao. Robust inventory control under demand and lead time uncertainty. *Annals of Operations Research*, 257(1-2):207–236, 2017.

[101] P.-F. J. Tsai and G.-Y. Teng. A stochastic appointment scheduling system on multiple resources with dynamic call-in sequence and patient no-shows for an outpatient clinic. *European Journal of Operational Research*, 239(2):427–436, 2014.

[102] A. Turkcan, B. Zeng, and M. Lawley. Chemotherapy operations planning and

scheduling. *IIE Transactions on Healthcare Systems Engineering*, 2(1):31–49, 2012.

[103] A. R. Vila-Parrish, J. S. Ivy, and R. E. King. A simulation-based approach for inventory modeling of perishable pharmaceuticals. In *2008 Winter Simulation Conference*, pages 1532–1538. IEEE, 2008.

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

[105] L. A. Wolsey. Lot-sizing with production and delivery time windows. *Mathematical Programming*, 107(3):471–489, 2006.

[106] World-Bank. Health expenditure, 2014.

[107] G. Xu, J. Feng, F. Chen, H. Wang, and Z. Wang. Simulation-based optimization of control policy on multi-echelon inventory system for fresh agricultural products. *International Journal of Agricultural and Biological Engineering*, 12 (2):184–194, 2019.

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

[109] W. Zhang, Z. Hua, and S. Benjaafar. Optimal inventory control with dual-sourcing, heterogeneous ordering costs and order size constraints. *Production and Operations Management*, 21(3):564–575, 2012.

[110] D. Zhou, L. C. Leung, and W. P. Pierskalla. Inventory management of platelets in hospitals: optimal inventory policy for perishable products with regular and

optional expedited replenishments. *Manufacturing & Service Operations Management*, 13(4):420–438, 2011.