Integrated Scheduling Problems in Healthcare and Logistics

Maryam Haghi

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

August 2020

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Maryam Haghi**

Entitled: **Integrated Scheduling Problems in Healthcare and Logistics**

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. Anjan Bhowmick	Chair
Dr. Nadia Lahrichi	External Examiner
Dr. Jia Yuan Yu	External to Program
Dr. Mingyuan Chen	Examiner
Dr. Daria Terekhov	Examiner
Dr. Ivan Contreras	Supervisor
Dr. Nadia Bhuiyan	Supervisor
Dr. Hossein Hashemi Doulabi	Supervisor

Approved by Dr. Ivan Contreras

Graduate Program Director

Dr. Mourad Debbabi

August 27, 2020 Dean of Gina Cody School of Engineering and Computer Science

Abstract

Integrated Scheduling Problems in Healthcare and Logistics

Maryam Haghi, Ph.D.

Concordia University, 2020

Scheduling is one of the important components of operation management in different services. The goal of scheduling is to allocate limited available resources over time for performing a set of activities such that one or more objectives are optimized. In this thesis, we study several interesting applications of scheduling in health care and logistics. We present several formulations and algorithms to efficiently solve the scheduling problems that arise in these areas.

We first study static and dynamic variants of a multi-appointment, multi-stage outpatient scheduling problem that arises in oncology clinics offering chemotherapy treatments. We present two integer programming formulations that integrate numerous scheduling decisions, features, and objectives of a major outpatient cancer treatment clinic in Canada. We also develop integrated and sequential scheduling strategies for the dynamic case in which arriving requests are processed at specific points of time. The results of computational experiments show that the proposed scheduling strategies can achieve significant improvements with respect to the several performance measures compared to the current scheduling procedure used at the clinic.

We next present a daily outpatient appointment scheduling problem that simultaneously determines the start times of consultation and chemotherapy treatment appointments for different types of patients in an oncology clinic under uncertain treatment times. We formulate this stochastic problem using two two-stage stochastic programming models. We also propose a sample average approximation algorithm to obtain high quality feasible solutions. We use an efficient specialized algorithm that quickly evaluates any given first-stage solution for a large number of scenarios. We perform several computational experiments to compare the performance of proposed two-stage stochastic programming models. In the next part of the experiments, we show that the quality of the first-stage solutions obtained by the sample average approximation is significantly higher than those of the expected value problem, and the value of stochastic solution is extremely high specially for higher degrees of uncertainty.

Finally, we address two variants of a cross-dock scheduling problem with handling times that simultaneously determines dock-door assignments and the scheduling of the trucks. In the general variant of the problem we assume that unit-load transfer times are door dependent, whereas in the specific case variant, unit-load transfer times are considered to be identical for all pairs of doors. We present constraint programming formulations for both variants of the problem, and we compare the performance of these models with mixed integer programming models from the literature. For the specific case, we propose several families of valid inequalities that are then used within a branch-and-cut framework to improve the performance of a time-index model. To solve the general problem efficiently, we also develop an approximate algorithm that first solves the specific case problem with the developed branch-and-cut algorithm to obtain a valid lower-bound, and then applies a matheuristic to obtain a valid upper-bound for the general problem and to compute the optimality gap. According to the computational experiments, we show that the proposed formulations and algorithms are able to solve the studied problems efficiently, and they outperform other models and heuristics that were previously developed for the problem in the literature.

To my family

Everything diminishes when it is used except knowledge- Imam Ali

Acknowledgements

Finishing my Ph.D. study and completing this thesis would not have been possible without the inspiration and kind support of many wonderful individuals. I would like to extend my sincerest thanks to all of them for being part of this journey and making this research possible.

First of all, I would like to express my sincere gratitude to my supervisors. I am truly grateful to Dr. Nadia Bhuiyan for giving me the opportunity of working on real world problems as part of my research, and for her continuous encouragement and support. I owe my deepest appreciation to Dr. Ivan Contreras for his immense knowledge, continuous support, patience, and encouragement. His invaluable suggestions, comments and guidance encouraged me to learn more day by day, and his deep insights helped me at various stages of my research. I also specially thank Dr. Hossein Hashemi Doulabi for his invaluable advice and comments, and for the long conversations we had to discuss different ideas.

Special thanks to Dr. Gerald Batist and the staff of Segal Cancer Centre and in particular Erin Cook, for their support and providing the required data. I also acknowledge MITACS for its financial support.

My heartfelt and sincere thanks to my parents for their unconditional love and support, and for all sacrifices they made to shape my life. You have been my source of strength and inspiration in all moments of my life, specially during the tough days of the Ph.D. program. Despite of being thousands of kilometers away, I've always felt you really close by my heart. I will forever be thankful to you for all invaluable lessons of life you've taught me, and I would never be able to pay back the love and what you have done for me so that I could achieve my goals.

Finally, from the bottom of my heart, I would like to say big thank you to my wonderful husband, Pedram, for always being there for me, from the saddest moments of life to the happiest ones. Words can not express how grateful I am to you for your endless love, support and understanding, and for always having belief in me. I am also deeply thankful to my beloved little girl, Bahar, who joined me in the middle of this journey and filled my life with true joy and happiness. Thanks for forgiving me for the moments that I should have spent with you, but I couldn't. I consider myself the luckiest person in the world to have such a lovely and caring family, always standing beside me with their endless love and unconditional support.

Contribution of Authors

This thesis has been prepared in *Manuscript-based* format. It contains three articles that are or will be submitted for publication in different journals. The first article is co-authored with Dr. Ivan Contreras and Dr. Nadia Bhuiyan. It also includes Dr. Gerald Batist, director of Segal Cancer Centre, who established research guidelines. The second article is co-authored with Dr. Ivan Contreras, Dr. Nadia Bhuiyan, and Dr. Hossein Hashemi Doulabi. Finally, the third article is co-authored with Dr. Ivan Contreras. The author of this thesis acted as the principal researcher and performed the mathematical formulations and algorithms development, programming of the solution methods, analysis of computational results, along with writing the first drafts of the articles.

The first article titled "Static and Dynamic Multi-Appointment, Multi-Stage Outpatient Chemotherapy Scheduling" is submitted in August 2020 to the journal *Omega*. The second article titled "Integrated Consultation and Chemotherapy Scheduling under Uncertain Treatment Times" is planned to be submitted in September 2020 to the journal *Expert Systems with Applications*. Finally, the third article titled "Formulations and Algorithms for Cross-dock Scheduling Problems with Handling Times" is submitted in August 2020 to the journal *Transportation Research Part B: Methodological*.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Scheduling integrates resource allocation and sequencing decisions and plays an important role in many service industries. Using sophisticated scheduling tools enables service providers to utilize their scarce resources efficiently while delivering high quality services to customers. Health care and logistics are two common examples of service industries in which scheduling has many applications. This thesis addresses two important scheduling problems in these areas: appointment scheduling in healthcare as a tool to manage patient arrivals such that resources are utilized properly and patient queues are avoided [52], and cross-dock scheduling in logistics as an important step to efficiently synchronize incoming and outgoing trucks such that intermediate storage inside the facility is kept low and deliveries are expedited [10].

In the first part of the thesis, we focus on one of the applications of outpatient appointment scheduling arising at the Segal Cancer Center, a major cancer center in Canada. Oncology centres are among the most highly demanded outpatient clinics due to the increasing cancer rates and growing demand for cancer treatments such as chemotherapy. Designing efficient scheduling systems for such clinics is required to ensure that all requests can be granted with a satisfactory service quality and with reasonable access and service times. We study a comprehensive chemotherapy scheduling problem that aims to properly coordinate multiple appointments of patients for several stages including blood test, oncologist consultation, drug preparation, and chemotherapy treatment. We develop integer programming formulations to model this problem. We also design an online scheduling algorithm to address dynamic arrival of appointment requests to the clinic.

In the second part of the thesis, we again study the application of appointment scheduling in oncology clinics, while considering a different setting. Stochastic service time is one of the main challenges in any appointment scheduling problem. Incorporating such uncertainties in the scheduling process is important for obtaining efficient and reliable schedules that satisfy decision-makers expectations in reality. There are several reasons for the uncertain duration of chemotherapy treatments. The main reasons include early termination of the treatment when patients cannot tolerate the injected drug, and longer treatment times due to the complications caused by patients' adverse reactions to drugs. We develop two-stage stochastic programming formulations to address an integrated daily consultation and chemotherapy scheduling problem where treatment duration times are stochastic. We also present a sampling-based algorithm to find high quality feasible solutions for this stochastic problem.

In the last part of this thesis, we address an important application of scheduling in modern logistics systems. Cross-docking is a logistic strategy that facilitates rapid movements of products from origins to destinations with minimal requirement of storage in between, which helps to improve the efficiency of the logistic operations by reducing costs and accelerating the process. Efficient scheduling of the incoming and outgoing trucks is a vital step to achieve these goals. Therefore, we study two variants of a cross-dock scheduling problem considering handling times. We propose constraint programming formulations for the studied problems that can provide high quality feasible solutions. We also develop exact and approximate algorithms to efficiently solve both problems.

The contributions of this thesis can be categorized as follows:

- Problem definition:
  - Addressing a comprehensive chemotherapy scheduling problem while considering the dynamic nature of appointment requests arrivals.
  - Addressing simultaneous scheduling of consultation and treatment appointments for different types of patients under uncertain treatment times.
  - Addressing a cross-dock scheduling problem with handling times including unloading, transfer and loading times of products.
- Problem modeling:
  - Development of two alternative integer programming formulations for a comprehensive chemotherapy scheduling problem.
  - Development of two alternative two-stage stochastic programming models for a chemotherapy scheduling problem with uncertain treatment times.
  - Development of constraint programming formulations for two variants of a cross-dock scheduling problem with handling times.
- Algorithmic development:
  - Development of integrated and sequential scheduling strategies for online scheduling of arriving requests to an oncology clinic.
  - Development of a sample average approximation algorithm for the stochastic chemotherapy scheduling problem that includes a specialized algorithm for quick evaluation of a given solution.
  - Development of an exact branch-and-cut algorithm that uses several families of valid inequalities, a matheuristic and an approximate algorithm for the cross-dock scheduling problem.

- Managerial insights:
  - Providing an analysis on the input parameters' effects on the performance of the designed scheduling algorithm for oncology clinics.
  - Providing an analysis on the effects of different levels of uncertainty on the value of the stochastic solution.
  - Providing an analysis on the impacts of incorporating accurate transfer times on the quality of the obtained cross-dock schedules.

The reminder of this thesis consists of four more chapters, three of which correspond to manuscripts that are or will be shortly submitted for publication in peer-reviewed scientific journals. Chapter 2 addresses a comprehensive multi-appointment, multi-stage chemotherapy scheduling problem that integrates numerous scheduling decisions, features, and objectives of a major outpatient cancer treatment clinic in Canada. It presents two mathematical formulations, two scheduling strategies and an online algorithm for this problem. Chapter 3 studies the integrated scheduling of consultation and treatment appointments for different types of chemotherapy patients, while taking into account stochastic duration of injection. It provides two two-stage stochastic programming formulations and a sample average approximation algorithm that includes a specialized algorithm for the evaluation of a given solution for this stochastic problem. Chapter 4 presents two variants of a cross-dock scheduling problem considering unloading, transfer and loading times of products that assigns incoming and outgoing trucks to doors and simultaneously determines the scheduling of trucks assigned to the same door. We develop constraint programming formulations, several families of valid inequalities, an exact branch-and-cut algorithm, two matheuristics, and an approximate algorithm for the studied problems. Finally, Chapter 5 provides conclusions and several avenues for future research.

Chapter 2

Static and Dynamic Multi-Appointment, Multi-Stage Outpatient Chemotherapy Scheduling

The content of this chapter is submitted as a manuscript for publication to the journal *Omega* in August 2020 [31].

Abstract

Outpatient oncology clinics offering chemotherapy treatment are among the most demanded multi-stage, multi-resource healthcare systems. In these clinics patients must pass through several interrelated stages during each appointment. In order to continuously coordinate all required patient appointments properly while utilizing scarce resources efficiently, we propose comprehensive scheduling problems to schedule multiple appointment requests of different types of patients for each of the stages. The proposed problem integrates numerous scheduling decisions, features, and objectives of a major outpatient cancer treatment clinic in Canada. We model and solve the static and dynamic cases of the considered scheduling problem. We present two

integer programming formulations for the static case. We develop integrated and sequential scheduling strategies for the dynamic case in which arriving requests are processed at specific points of time. Extensive computational results based on real data from the Segal Cancer Centre are reported.

2.1 Introduction

Current growing demand for healthcare services, along with the great emphasis on preventive actions and cost reduction, have made outpatient clinics an essential component of healthcare systems [14]. As a consequence, the development of efficient scheduling tools for such clinics has received increased attention in recent years. Oncology centres are among the most highly demanded outpatient clinics due to the increasing cancer rates and growing demand for cancer treatments such as chemotherapy. Based on the Canadian Cancer Society report [1], cancer is the leading cause of death in Canada and the average annual number of new cancer cases is expected to significantly increase in future years. Therefore, efficient scheduling systems for cancer-care provider clinics, including outpatient chemotherapy clinics, are required to ensure that all requests can be granted with a satisfactory service quality and with reasonable access and service times.

Outpatient chemotherapy clinics have some unique characteristics which make their scheduling decisions different and more complicated than scheduling problems in other typical outpatient clinics [36]. One of these characteristics is the existence of several stages including blood test, consultation with an oncologist (or doctor visit), drug preparation in pharmacy, and chemotherapy treatment (or infusion). Each patient must go through all or a subset of these stages following a predefined sequence, and consequently, different paths for patients exist in such clinics. Figure 2.1 repre-

[1] http://www.cancer.ca/en/cancer-information/cancer-101/canadian-cancer-statistics-publication/?region=qc

sents the flow of patients in an outpatient chemotherapy clinic. In order to visit an oncologist or start preparing a chemotherapy drug, the patient needs to have the results of a blood test which is done at most 48 hours in advance. The patient may come to the clinic with blood test results or may request for an appointment to perform the blood test at the clinic. If the patient needs to visit the oncologist before his/her treatment appointment, drug preparation can begin only once consultation with the physician is completed. All these stages in chemotherapy clinics are completely dependent on each other and a proper coordination between them is essential in order to provide patients with efficient care services [41]. However, there is a lack of studies that integrate multiple stages [41]. Moreover, the requirement of the simultaneous presence of a nurse and a chair for completing the treatment stage is another unique characteristic of chemotherapy scheduling problems that is rarely considered.

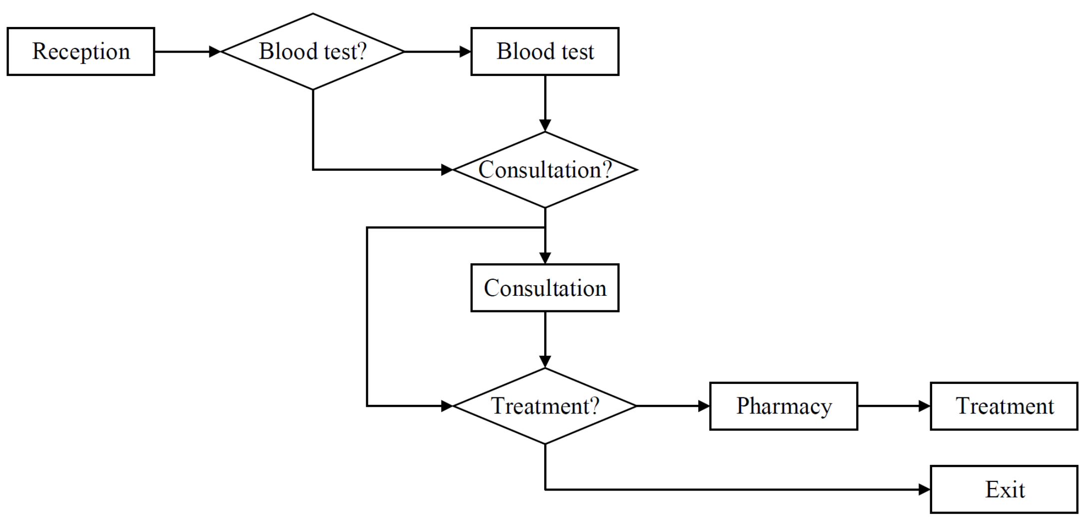


Figure 2.1: Flow of patients in an outpatient chemotherapy clinic

Chemotherapy services are typically administered in repetitive cycles. The frequency and structure of each cycle (i.e., types and doses of drugs in each treatment and rest periods between different treatments) are determined by regimens. Based on the patient's type of cancer, stage of the cancer growth, and current health situation, oncologists prescribe an appropriate regimen for each patient and recommend treatment dates. Deviations from these recommended dates may decrease the efficiency of the cure and put the patient's health in danger. It is thus better for the scheduler to immediately schedule all the required appointments in order to guarantee the availability of the later appointments [2].

There are two systems commonly used for planning and scheduling in outpatient chemotherapy clinics. The first one is the *next-day scheduling system*, where patients have a blood test and an oncologist consultation appointment one day before the treatment [20]. In such systems if the lifetime of a chemotherapy drug allows it, the drug can also be prepared by a pharmacist one day before the treatment appointment. Otherwise, it will be prepared on the treatment day, a few hours before infusion begins. Such drugs with a short lifetime are referred to as special drugs in this paper. The second scheduling system is the *same-day scheduling system*, where patients have all the required appointments on the same day.

There are also two approaches for assigning nurses to patients in chemotherapy clinics. In the first approach, referred to as *functional care delivery*, patients are assigned to arbitrary nurses on different days of their treatment, while in the second approach, known as *primary care delivery*, each patient must be assigned to one member of a specific group of nurses at every referral [44]. Each chemotherapy treatment appointment needs the simultaneous availability of two resources to start. A chair or a bed must be fully dedicated to each patient during the complete treatment stage. The complete assignment of a nurse to a patient is also necessary for setting up a patient. However, during the infusion a nurse can monitor up to four patients simultaneously.

In this paper we study a comprehensive chemotherapy scheduling problem, denoted as the *multi-appointment, multi-stage chemotherapy scheduling problem* (MM-CSP), considering unique characteristics and realistic assumptions arising at the Segal Cancer Center (SCC), a major cancer center in Canada. SCC is one of the most highly demanded oncology centers in Canada, as it currently schedules on a daily basis about 100 to 250 consultation appointments and 60 to 80 chemotherapy treatment appointments [62]. Managing such a high volume of patients requires an efficient scheduling strategy that enables the clinic not only to schedule each appointment in an optimized way, but also to provide a good coordination between different interrelated stages to increase the efficiency of the entire system and to improve the overall staff and patients' satisfaction.

At the SCC, different types of patients request multiple appointments for several stages, and each stage requires one or more types of resources with limited capacity to serve patients. Moreover, following the practice at the SCC we use a next-day system for scheduling decisions and a primary care delivery approach for nurse-to-patient assignments. However, our model can be easily modified to consider a functional care delivery approach as well. Other distinguishing features incorporated in our problem are: i) different patient types with different requirements and paths to follow, ii) heterogeneous infusion equipment, iii) nurses' daily workload balancing considerations, and iv) chemotherapy drugs' lifetime. To the best of our knowledge, this is the most comprehensive outpatient scheduling problem that has ever been studied in the literature.

The other main contributions of our work are the following. We propose a static version of the MMCSP in which we integrate all scheduling decisions. In the static MMCSP, all appointment requests pertaining to a fixed planning horizon are known at the beginning of the planning horizon. In order to assess the value of integrating all decisions, we also propose two sequential approaches for the static MMCSP in which either consultation appointments are scheduled before treatment appointments or the other way around. We refer to them as *consultation-treatment* (CT) and *treatment-consultation* (TC) approaches. We also consider the dynamic MMCSP in which appointment requests arrive in real time over a rolling horizon. We develop and computationally compare two integer programming (IP) formulations for the integrated static MMCSP. We then use the most promising of these IPs to solve the dynamic version in which a scheduling algorithm schedules the appointments dynamically as they arrive (in the form of waiting lists) at discrete points of time. In the dynamic case, it is assumed that the scheduling of patients is started by filling out a partially-filled schedule (i.e., some of the resources are occupied by already booked patients). The remainder of this paper is organized as follows. In Section 2.2, the most recent and relevant literature is reviewed. Section 2.3 formally describes the MMCSP. Section 2.4 presents two IP formulations for the integrated static MMCSP, whereas Section 2.5 presents the two sequential approaches for solving the static MMCSP. A description of the proposed online scheduling algorithm is given in Section 2.6. Extensive computational experiments and analyses on input parameters are reported in Section 2.7. Finally, Section 2.8 provides some conclusions.

2.2 Literature Review

There are many papers that have reviewed outpatient scheduling and appointment systems. In particular, Cayirli and Veral [14], Gupta and Denton [29] and Ahmadi-Javid et al. [1] provide three examples of these review papers. Furthermore, Lamé et al. [41] focus their review on the planning of outpatient chemotherapy clinics. Marynissen and Demeulemeester [49] review multi-stage scheduling problems and mention oncology clinics among the important applications of such problems. However, these studies highlight that most of the research has focused on scheduling patients for the treatment stage only rather than considering all required stages. For examples of single-stage problems, we refer to [2, 7, 13, 18, 26, 27, 32, 35, 37, 38, 44, 65]. There are few studies in which at least two stages have been taken into account. Sadki et al. [57], Sadki et al. [58], and Garaix et al. [22] focus on scheduling patients for same-day consultation and treatment appointments on a daily basis assuming that all patients require both appointments. In addition to consultation and treatment appointments, Bouras et al. [9] consider the drug preparation stage. Liang et al. [45] and Suss et al. [63] use discrete event simulation to model oncology clinics with a same-day scheduling system in which different types of patients request appointments. Contrary to our work, none of the above mentioned studies have modeled all the required stages of an oncology clinic using a mathematical program.

After an oncologist prescribes a regimen involving several chemotherapy treatments, each patient should start their treatment as soon as possible. It is thus required to determine a date and time for each appointment. Some studies only focus on determining a date for each appointment or start date of the treatment plan (see [26, 27]); while other studies consider a daily version of the problem and determine the start time of appointments of a given day (see [9, 13, 22, 32, 35, 37, 44, 45, 57, 58, 63]). However, both of these decisions have been jointly considered in some studies (see [2, 7, 18, 38, 65]), which is also the case in our work.

Turkcan et al. [65] develop two integer programming formulations for assigning a start date, start time and required resources for new patients requesting appointments for chemotherapy treatment. In the first formulation, new patients are assigned to a day of the planning horizon to start their treatment. The second formulation is then solved for each day of the planning horizon to determine start time of treatments along with the nurse and chair assignments. Condotta and Shakhlevich [18] focus on the scheduling of multiple appointments for the chemotherapy stage. In order to deal with the uncertainty of the number and type of future requests, the authors generate a template schedule for a set of artificial patients which contains more prebooked appointments than anticipated, providing flexibility for handling unexpected arrivals. Alvarado and Ntaimo [2] introduce a two-stage stochastic programming formulation for scheduling multiple chemotherapy appointments for a single patient where the acuity level of the patient, appointment duration, and the number of nurses on duty for each day are uncertain. Benzaid et al. [7] propose a two-step procedure for online scheduling of chemotherapy appointments for new patients. In the first step, the authors consider a planning horizon and assign a date and a time to each appointment request. In the second step, they solve a mathematical model for each day of the planning horizon to determine the number of required nurses on each day and also to assign patients to the available nurses. They have also proposed another daily-basis mathematical model to include patient cancellations and nurse absences in which they solve the problem for an updated set of patients and nurses without considering resource capacity constraints. However, all these studies focus only on the treatment stage. The need for booking multiple appointments for other stages of the oncology clinic, along with the treatment stage, has not been addressed in the literature.

The most relevant studies to our work are [63] and [38]. In particular, Suss et al. [63] propose a scheduling algorithm which uses lean principles to obtain a master appointment schedule for patient arrivals to SCC, so that different types of patients can have all the required appointments with the minimum waiting time. Using a discrete event simulation model, the authors test the proposed algorithm which calculates the best patient arrival rate based on the service rate at the pharmacy stage. Based on the results, it is observed that using even a simple patient appointment scheduling algorithm, patient waiting times can be reduced by at least 40%. They also highlight that a reduction of the non-value-added work by the staff in managing multiple appointment schedules and the efficient utilization of clinic's resources are other potential benefits of using scheduling algorithms. However, this paper considers a same-day scheduling system and only schedules single appointment of patients on a daily basis. It also focuses only on reducing waiting times and does not consider other key performance measures such as balancing the workload of nurses or assigning patients to preferred nurses and time slots. In a follow-up paper, Hooshangi-Tabrizi et al. [38] study an online multi-appointment scheduling problem for chemotherapy patients at SCC. The authors develop a mathematical programming formulation that assigns a date, starting time, chair, and a nurse to each arriving request with the goal of minimizing the number of unscheduled appointments, nurses' overtime and non-preferred nurse and time assignments. They also propose a daily rescheduling model which may change the assigned chair, nurse or starting time of appointments when either new information is revealed or an unexpected event like a nurse's absence or an appointment cancellation happens. Our work also differs from [38], as we consider a more comprehensive scheduling problem that incorporates requests arising from different patient categories and schedules appointments for all stages at the SCC.

Finally, we would like to highlight that our scheduling problem shares some features to other problems in flexible job shop scheduling (FJSS) [15]. However, our problem differs from FJSS problems in several aspects. In a FJSS problem, only one resource is required to process each job. However, in our problem a simultaneous presence of a chair and a nurse is necessary to start the process. Moreover, the full attention of a nurse is only required for starting the process and then, one nurse can handle several patients simultaneously. Online scheduling has been rarely considered for FJSS problems and thus, it is assumed that all machines are available at the scheduling starting point. However, we also solve the dynamic (or online) variant of the problem and we assume that all resources may not be available at the starting point. In FJSS problems, the waiting times of jobs between two stages are not important as long as a specific due date is respected. However, in our problem we are dealing with patients and therefore, minimizing the waiting time is a critical performance measure.

2.3 Problem Definition

Let $\mathcal{A}$ be the set of five services offered by the four main stages in an outpatient oncology clinic: 1) blood test for consultation, 2) blood test for treatment, 3) consultation, 4) drug preparation, and 5) chemotherapy treatment. Let $\mathcal{P}$ denote the set of all patients from different groups requesting at least one appointment for one or more services during the planning horizon. Let $\mathcal{D}$ be the set of days in the current planning horizon and $\mathcal{S}$ represent the set of time slots on each day. We define $\mathcal{FP}$ as the set of new consultation patients who request a visit to an oncologist for the first time, and $\mathcal{NP}$ as the set of new treatment patients who have previously visited an oncologist and want to receive their first treatment. New treatment patients may also require a consultation with the physician before their first treatment. We also define $\mathcal{RP}$ as the set of recurring patients whose treatment plan has already been determined and they may require multiple appointments. Let $\mathcal{CA}_p$ and $\mathcal{TA}_p$ denote the sets of consultation and treatment appointments requested by patient $p \in \mathcal{P}$, respectively. From the definition of patient groups, we note that for $p \in \mathcal{FP}$, $|\mathcal{CA}_p| = 1$ and $|\mathcal{TA}_p| = 0$, and for $p \in \mathcal{NP}$, $|\mathcal{CA}_p| \leq 1$ and $|\mathcal{TA}_p| = 1$.

For each requested consultation appointment $k' \in \mathcal{CA}_p$ of patient p, we define the parameter $BC_{pk'}$ equal to 1 if and only if a blood test appointment is also required. Similarly, for each requested treatment appointment $k \in \mathcal{TA}_p$ of patient p, the parameter BT_{pk} is equal to 1 when booking a blood test appointment is also needed. For new treatment patient $p \in \mathcal{NP}$, if in addition to the infusion, both blood test and consultation appointments are required, it is assumed that $BC_{p1} = 1$ and $BT_{p1} = 0$. However, if only a blood test appointment is needed, $BT_{p1} = 1$ and since in this case $|\mathcal{CA}_p| = 0$, BC_{p1} is not defined for the patient. Let PT^i_{pk} represent the processing time of service $i \in \mathcal{A}$ for the appointment k of patient p. Processing times of different services can differ from one patient to another and even among different appointments of a patient. For instance, one patient might need different drugs at different appointments and thus different infusion times are needed.

Let $\mathcal{O}$, $\mathcal{N}$ and $\mathcal{C}$ denote the sets of oncologists, nurses and chairs, respectively. It is assumed that the oncologist of each patient is known and the assignment of nurses to chairs is done in advance. We define $\mathcal{P}_o^O \subseteq \mathcal{P}$ as the set of all patients served by oncologist $o \in \mathcal{O}$, and $\mathcal{C}_{nd}^N \subseteq \mathcal{C}$ as the set of chairs assigned to nurse $n \in \mathcal{N}$ on day $d \in \mathcal{D}$. Furthermore, working days and hours of oncologists and nurses are also assumed to be known. Let $\mathcal{D}_o^O \subseteq \mathcal{D}$ and $\mathcal{D}_n^N \subseteq \mathcal{D}$ denote working days of oncologist o and nurse n, respectively. In this paper, chairs are considered to be heterogeneous. In some appointments, based on the prescribed drug, patients might be required to be assigned to a particular chair with a special port or based on the health status of the patient, some drugs are only allowed to be injected while the patient is lying down on a bed. Therefore, we define $\mathcal{C}_{pk}^P \subseteq \mathcal{C}$ as the set of chairs which are allowed for the treatment appointment k of patient p.

For the recurring patients, the exact dates of requested treatment and consultation appointments are defined by the prescribed regimen. However, for both groups of new patients, i.e., $p \in \mathcal{FP} \cup \mathcal{NP}$, the goal is to assign a date, as soon as possible, before a specific deadline and based on the patients' priorities. Furthermore, new treatment patients need to attend an information session before their first treatment appointment, which should also be taken into account. Let $\mathcal{D}_{pk}^i \subseteq \mathcal{D}$ and $\mathcal{S}_{pkd}^i \subseteq \mathcal{S}$ represent the sets of all possible days and time slots, respectively, which can be assigned to appointment k of patient p for service i, considering all the conditions. For example, $\mathcal{D}_{pk}^5$ includes only the exact infusion dates for the treatment appointment k of recurring patient p, while it includes the days after the information session and before the deadline for new treatment patient p. As another example, $\mathcal{D}_{pk}^3$ for a first consultation patient contains a subset of days before the deadline when the oncologist is available, either due to the working hours or regarding the partially-filled schedule in an online mode, and for each $d \in \mathcal{D}_{pk}^3$, associated time slots set, $\mathcal{S}_{pkd}^3$, excludes the time slots on which the oncologist is not available.

In the current work a next-day system is assumed. Therefore, special drug preparations should be scheduled on the same day as the infusion, whereas blood test, consultation and non-special drug preparations are scheduled one day before the treatment appointment. For each patient p and each $k \in \mathcal{TA}_p$, we define the parameter SD_{pk} equal to 1 when the required drug is considered to be special, otherwise it is assumed as a non-special drug. The preparation of a non-special drug for a treatment that is following a consultation appointment should start only once the oncologist visit is finished. Consider $k' \in \mathcal{CA}_p$ and $k \in \mathcal{TA}_p$ as the required consultation and treatment appointments for patient $p \in \mathcal{RP}$. We define the parameter $Se^{C}_{pk'k}$ equal to 1 when a doctor visit k' is followed by the infusion appointment k on the next day. Although the main scheduling system is considered to be a next-day system, for the after-weekend days (days that are located after holidays or weekends) a same-day system for blood test, non-special drug preparation and the infusion is considered. We define $\mathcal{D}^{\mathcal{H}}$ to be the set of after-weekend days. Figure 2.2 illustrates the precedence constraints between different stages for regular and after week-end days.

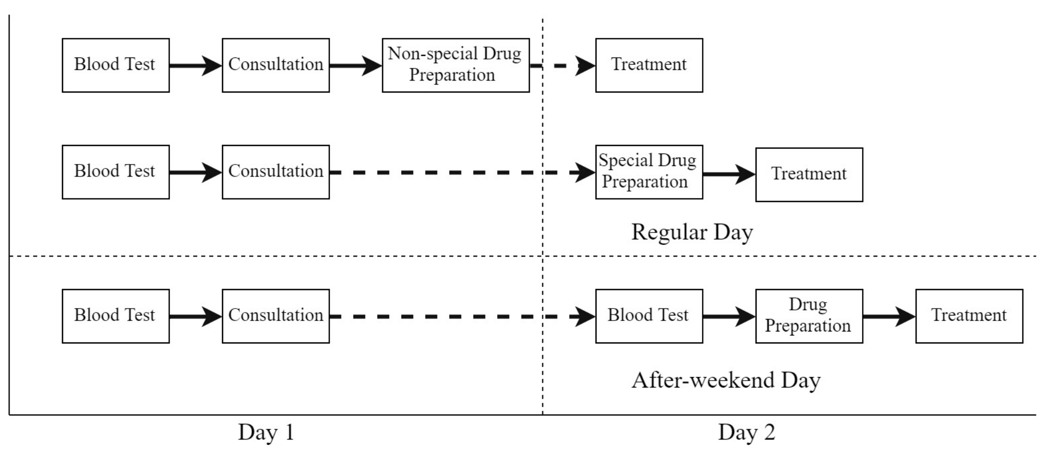


Figure 2.2: Precedence constraints between different stages for regular and after weekend days

In the MMCSP, the goal is to schedule as many appointments as possible requested by all patients $p \in \mathcal{FP} \cup \mathcal{NP} \cup \mathcal{RP}$ for different services $i \in \mathcal{A}$ offered by the SCC.

That is, for each requested appointment of each of the services a decision has to be made: either the appointment is scheduled or is left as unscheduled. If the current available resources are not enough to serve all the requested appointments, some of the appointments will be directed to a buffer (i.e., a virtual waiting list). In practice, after cancellations or changes in the appointments have been revealed, the scheduler will try to squeeze the appointments in the buffer into the schedule.

The MMCSP seeks to assign a date and a time and reserve the required resources for each appointment. Scheduling decisions are made by taking into account the following (conflicting) objective categories: i) minimizing oncologists' overtime and completion time, ii) minimizing nurses' overtime, iii) minimizing patients' waiting time, iv) balancing daily workload of nurses, and v) maximizing the assignment of patients to their primary nurses and preferred treatment times.

2.4 Formulations for the Static MMCSP

In this section we describe two integer programming (IP) formulations for the static MMCSP. They use the same set of variables but differ in the sets of constraints used to model the precedence constraints between services. In what follows, we provide the definition of the decision variables, the considered objective functions, and the constraints needed to model the MMCSP.

2.4.1 Decision variables

We recall that each nurse is responsible for a predefined set of chairs. Therefore, when assigning an appointment to a chair the assigned nurse is known. We use the following sets of binary decision variables.

- $y^1_{pk'ds}$ ($p \in \mathcal{P}, k' \in \mathcal{CA}_p, d \in \mathcal{D}^1_{pk}, s \in \mathcal{S}^1_{pkd}$): takes value 1 if and only if the blood test needed for consultation appointment k' of patient p is scheduled on day d

starting at time slot s;

- y^2_{pkds} $(p \in \mathcal{NP} \cup \mathcal{RP}, k \in \mathcal{TA}_p, d \in \mathcal{D}^2_{pk}, s \in \mathcal{S}^2_{pkd})$: takes value 1 if and only if the blood test needed for treatment appointment k of patient p is scheduled on day d starting at time slot s;

- $y^3_{pk'ds}$ $(p \in \mathcal{P}, k' \in \mathcal{CA}_p, d \in \mathcal{D}^3_{pk}, s \in \mathcal{S}^3_{pkd})$: takes value 1 if and only if consultation appointment k' of patient p is scheduled on day d starting at time slot s;

- y^4_{pkds} $(p \in \mathcal{NP} \cup \mathcal{RP}, k \in \mathcal{TA}_p, d \in \mathcal{D}^4_{pk}, s \in \mathcal{S}^4_{pkd})$: takes value 1 if and only if drug preparation for treatment appointment k of patient p is scheduled on day d starting at time slot s;

- x_{pkdsc} $(p \in \mathcal{NP} \cup \mathcal{RP}, k \in \mathcal{TA}_p, d \in \mathcal{D}^5_{pk}, s \in \mathcal{S}^5_{pkd}, c \in \mathcal{C}^P_{pk})$: takes value 1 if and only if treatment k of patient p is scheduled on day d starting at time slot s on chair c;

- $z^1_{pk'}$ $(p \in \mathcal{P}, k' \in \mathcal{CA}_p)$: takes value 1 if and only if consultation appointment k' of patient p is placed in the buffer;

- z^2_{pk} $(p \in \mathcal{NP} \cup \mathcal{RP}, k \in \mathcal{TA}_p)$: takes value 1 if and only if treatment appointment k of patient p is placed in the buffer.

2.4.2 Objectives

Consultation and treatment schedules have a direct impact on the quality of healthcare services offered to patients as well as the workload of physicians and nurses. It is thus critical to employ different performance measures (i.e., objective functions) that can integrate both clinic staff and patients perspectives when evaluating the efficiency of appointment schedules. In what follows, we overview eight different objectives that we include in our model: two are nurse-related, one is physician-related, and five are patient-related. For details on how these objectives are mathematically stated using the above decision variables, we refer the reader to Appendix A.

2.4.2.1 Nurse-related objectives

We use the following two objectives to improve nurses' satisfaction.

- $g_1(x)$: is a minmax objective that seeks to balance the nurses' daily workload as much as possible during the entire planning horizon by minimizing the sum of the maximum daily workload differences among all nurses pairs.

- $g_2(x)$: is a minsum objective that aims at minimizing the sum of daily nurses overtime or patients hand overs (i.e., a change of nurse in the middle of a patient's treatment).

2.4.2.2 Physician-related objectives

We consider the objective $g_3(y^3) = g_3^1(y^3) + g_3^2(y^3) + g_3^3(y^3)$, which incorporates oncologists preferences using the following three performance measures:

- $g_3^1(y^3)$: minimizes the sum of daily completion time of gynecology oncologists. This objective is relevant given that at the SCC gynaecologists usually leave the clinic as soon as all assigned patients are served. We note that in this paper, we have focused on the oncology, hematology and gynecology departments of the SCC.

- $g_3^2(y^3)$: minimizes the sum of daily overtime of oncologists except for the gynaecologists.

- $g_3^3(y^3)$: minimizes the number of patients assigned to the oncologists' break times. Although oncologists prefer to rest during breaks, sometimes it is needed to skip a break to be able to serve more patients.

2.4.2.3 Patient-related objectives

We use the following five objectives to improve patients' satisfaction.

- $g_4(x, y^1, y^2, y^3) = g_4^1(y^1, y^3) + g_4^2(x, y^2) + g_4^3(x, y^2)$: this objective seeks to reduce excessive direct waiting times of patients by considering the following three terms:
  - $g_4^1(y^1, y^3)$: minimizes the sum of patients' waiting time between blood test and consultation stages.
  - $g_4^2(x, y^2)$: minimizes the sum of recurring patients' waiting time between blood test and treatment stages whenever the blood test for the next treatment appointment has to be scheduled on the same day as the treatment appointment.
  - $g_4^3(x, y^2)$: minimizes patients' waiting time between blood test and treatment stages when both appointments are scheduled on a same after-weekend day.
- $g_5(x)$: minimizes the sum of waiting time (i.e., access time) for new treatment patients. This objective has the goal of assigning the soonest possible date to the patients with the highest priority.
- $g_6(x)$: minimizes the number of non-preferred assignments of patients to nurses.
- $g_7(x)$: minimizes the number of non-preferred time assignments for treatment appointments.
- $g_8(z^1, z^2)$: minimizes the number of unscheduled consultation and treatment appointments.

2.4.3 Modeling the feasible region

To model the set of all feasible schedules for the MMCSP, we need to simultaneously consider a large number of system's characteristics and assumptions that arise when scheduling a set of patients requesting multiple appointments over multiple stages, and requiring several resources from a chemotherapy clinic. In what follows, we introduce each class of constraints and provide an explanation for them.

2.4.3.1 Day-assignment constraints

For an arriving patient's request $k \in \mathcal{CA}_p \cup \mathcal{TA}_p$, $p \in \mathcal{P}$, and for each of its associated clinic service $i \in \mathcal{A}$, either a date and a start time is assigned to it or it is redirected to the buffer as an unscheduled request. The following are the assignment constraints for each of the five services:

$$\sum_{d \in \mathcal{D}^1_{pk'}} \sum_{s \in \mathcal{S}^1_{pk'd}} y^1_{pk'ds} + z^1_{pk'} = 1 \qquad p \in \mathcal{P}, k' \in \mathcal{CA}_p : BC_{pk'} = 1 \tag{2.1}$$

$$\sum_{d \in \mathcal{D}^2_{pk}} \sum_{s \in \mathcal{S}^2_{pkd}} y^2_{pkds} + z^2_{pk} = 1 \qquad p \in \mathcal{NP} \cup \mathcal{RP}, k \in \mathcal{TA}_p : BT_{pk} = 1 \tag{2.2}$$

$$\sum_{d \in \mathcal{D}^3_{pk'}} \sum_{s \in \mathcal{S}^3_{pk'd}} y^3_{pk'ds} + z^1_{pk'} = 1 \qquad p \in \mathcal{P}, k' \in \mathcal{CA}_p \tag{2.3}$$

$$\sum_{d \in \mathcal{D}^4_{pk}} \sum_{s \in \mathcal{S}^4_{pkd}} y^4_{pkds} + z^2_{pk} = 1 \qquad p \in \mathcal{NP} \cup \mathcal{RP}, k \in \mathcal{TA}_p \tag{2.4}$$

$$\sum_{d \in \mathcal{D}^5_{pk}} \sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^P_{pk}} x_{pkdsc} + z^2_{pk} = 1 \qquad p \in \mathcal{NP} \cup \mathcal{RP}, k \in \mathcal{TA}_p. \tag{2.5}$$

2.4.3.2 Buffer-related constraints

If a recurring patient p needs to have his/her k'^{th} visit with the doctor one day before his/her k^{th} treatment appointment, i.e., $Se^C_{pk'k} = 1$, and that consultation appointment cannot be scheduled, then the treatment appointment should also be redirected to the buffer. For such treatments, the patient should not receive the infusion before his/her status has been examined by the doctor. However, if the treatment appointment remains unscheduled, the consultation appointment can still be scheduled. These assumptions are reasonable because in practice, there are always some treatment cancellations, so it is most likely possible to schedule a treatment appointment later, after cancellations occur. This interrelationship between consultation and treatment appointments can be modeled as:

$$z^2_{pk} \geq z^1_{pk'} \qquad p \in \mathcal{RP}, k \in \mathcal{TA}_p, k' \in \mathcal{CA}_p : Se^C_{pk'k} = 1. \tag{2.6}$$

For the case of new treatment patients, we assume that either both consultation and treatment appointments are scheduled or both are not, which causes them to be assigned to the buffer. Given that one objective ($g^5(x)$) is to assign a start date for their first treatment as soon as possible, we cannot schedule in practice only one of these appointments. To ensure this is satisfied, we add the following constraints:

$$z^1_{pk} = z^2_{pk'} \qquad p \in \mathcal{NP}, k \in \mathcal{TA}_p, k' \in \mathcal{CA}_p. \tag{2.7}$$

2.4.3.3 Resource capacity constraints

There exist limited resources that are used in each of the four stages on an outpatient oncology clinic. In what follows, we discuss each one of them and provide their associated capacity constraints that need to be considered.

Consultation services

The oncologists are the only resources associated with the consultation services and they are assumed to have unitary capacity. More precisely, the following constraints ensure that at most one patient is served by each oncologist in each time slot:

$$\sum_{p\in\mathcal{P}_o^O}\sum_{k'\in\mathcal{K}_p(d,s)}\sum_{s'=s(PT^3_{pk'})}^{s} y^3_{pk'ds'} \leq 1 \qquad o\in\mathcal{O}, d\in\mathcal{D}_o^O, s\in\mathcal{S}, \tag{2.8}$$

where $s(PT^3_{pk'}) = \max\left\{1, s - PT^3_{pk'} + 1\right\}$ and $\mathcal{K}_p(d,s) = \{k' \in \mathcal{CA}_p : d \in \mathcal{D}^3_{pk'}, s \in \mathcal{S}^3_{pk'd}\}$.

Blood test services

Phlebotomists are the required resources for performing blood tests. The following capacity constraints ensure that the number of patients assigned to a time slot s in day d is no larger than the number of available phlebotomists (R^B_{ds}):

$$\sum_{p\in\mathcal{P}}\sum_{k'\in\mathcal{K}^1_p(d,s)}\sum_{s'=s(B)}^{s} y^1_{pk'ds'} + \sum_{p\in\mathcal{NP}\cup\mathcal{RP}}\sum_{k\in\mathcal{K}^2_p(d,s)}\sum_{s'=s(B)}^{s} y^2_{pkds'} \leq R^B_{ds} \quad d\in\mathcal{D}, s\in\mathcal{S}, \tag{2.9}$$

where $s(B) = \max\{1, s - B + 1\}$, and B is the time needed for taking blood samples. $\mathcal{K}^1_p(d,s) = \{k' \in \mathcal{CA}_p : BC_{pk'} = 1, d \in \mathcal{D}^1_{pk'}, s \in \mathcal{S}^1_{pk'd}\}$ and $\mathcal{K}^2_p(d,s) = \{k \in \mathcal{TA}_P : BT_{pk} = 1, d \in \mathcal{D}^2_{pk}, s \in \mathcal{S}^2_{pkd}\}$.

Drug preparation services

Pharmacists are the required resources for drug preparation. The following capacity constraints ensure that the number of patients assigned to a time slot s in day d is no larger than the number of available pharmacists (R^D_{ds}):

$$\sum_{p \in \mathcal{NP} \cup \mathcal{RP}:} \sum_{k \in \mathcal{K}'_p(d,s)} \sum_{s'=s(PT^4_{pk})}^{s} y^4_{pkds'} \leq R^D_{ds} \qquad d \in \mathcal{D}, s \in \mathcal{S}, \tag{2.10}$$

where $s(PT^4_{pk}) = \max\left\{1, s - PT^4_{pk} + 1\right\}$, and $\mathcal{K}'_p(d,s) = \{k \in \mathcal{TA}_P : d \in \mathcal{D}^4_{pk}, s \in \mathcal{S}^4_{pkd}\}$.

Treatment services - Chairs

Similar to oncologists, chairs are also assumed to have unitary capacity. The following constraints ensure that at most one patient is seated in each chair at each time slot:

$$\sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{K}_p(c,d,s)} \sum_{s'=s(PT^5_{pk})}^{s} x_{pkds'c} \leq 1 \qquad c \in \mathcal{C}, d \in \mathcal{D}, s \in \mathcal{S}, \tag{2.11}$$

where $s(PT^5_{pk}) = \max\left\{1, s - PT^5_{pk} + 1\right\}$, and $\mathcal{K}_p(c,d,s) = \{k \in \mathcal{TA}_P : d \in \mathcal{D}^5_{pk}, s \in \mathcal{S}^5_{pkd}, c \in \mathcal{C}^P_{pk}\}$.

Treatment services - Nurses

Nurse availability requires a more granular modeling. We recall that during setup time, the nurse can only be with one patient at a time. However, during the infusion time a nurse can monitor up to four patients simultaneously. We denote as ST_{pk} the required time to set up a patient on the infusion chair. The following constraints ensure that at most one patient is assigned to a nurse at each time slot during setup time:

$$\sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{K}''_p(d,s)} \sum_{s'=s(ST_{pk})}^{s} \sum_{c \in \mathcal{C}^P_{pk} \cap \mathcal{C}^N_{nd}} x_{pkds'c} \leq 1 \qquad n \in \mathcal{N}, d \in \mathcal{D}^N_n, s \in \mathcal{S}, \tag{2.12}$$

where $s(ST_{pk}) = \max\{1, s - ST_{pk} + 1\}$, and $\mathcal{K}''_p(d,s) = \{k \in \mathcal{TA}_P : d \in \mathcal{D}^5_{pk}, s \in \mathcal{S}^5_{pkd}\}$.

At SCC, chemotherapy equipment is placed in two stations $j \in \{1, 2\}$ located within walking distance from each other. However, it is not possible for a nurse to monitor patients sitting in different stations and each nurse is only assigned to chairs belonging to the same station on each day. Therefore, it is required to have monitoring constraints for each station, separately. For each station, the following constraint should be respected:

$$\sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{K}''_p(d,s)} \sum_{c \in \mathcal{C}^P_{pk} \cap \mathcal{C}^j} \left(\sum_{s'=s(ST_{pk})}^{s} \frac{3}{4} x_{pkds'c} + \sum_{s'=s(PT^5_{pk})}^{s} \frac{1}{4} x_{pkds'c} \right) \leq R^N_{dsj}$$
$$d \in \mathcal{D}^N_n, s \in \mathcal{S}, \tag{2.13}$$

where R^N_{dsj} is the total number of available nurses at time slot s and day d in station j and $\mathcal{C}^j$ is the set of chairs located in station j.

At SCC, nurses start their shifts at different time slots which in turn has an impact on the availability of nurses (and their associated chairs) during specific time slots. Moreover, when solving the dynamic variant, we schedule arriving patients while considering a partially-filled schedule, i.e., some nurses (and chairs) are no longer available during specific time slots. Therefore, every time a patient is scheduled for treatment, the availability of the assigned nurse (and chair) needs to be considered for the entire treatment time. The following constraints are used to fix some of the x variables to zero whenever nurses and chairs are not available for the entire duration of the treatment time:

$$\sum_{c \in \mathcal{C}^P_{pk} \cap \mathcal{C}^N_{nd}} x_{pkdsc} \leq \frac{\sum_{s'=s}^{s(ST_{pk})} AN_{nds'}}{ST_{pk}}$$

$$p \in \mathcal{NP} \cup \mathcal{RP}, k \in \mathcal{TA}_p, d \in \mathcal{D}^5_{pk} \cap \mathcal{D}^N_n, s \in \mathcal{S}^5_{pkd}, n \in \mathcal{N} \quad (2.14)$$

$$x_{pkdsc} \leq \frac{\sum_{s'=s}^{s(PT^5_{pk})} AC_{cds'}}{PT^5_{pk}}$$

$$p \in \mathcal{NP} \cup \mathcal{RP}, k \in \mathcal{TA}_p, d \in \mathcal{D}^5_{pk}, s \in \mathcal{S}^5_{pkd}, c \in \mathcal{C}^P_{pk}, \quad (2.15)$$

where $s(PT^5_{pk}) = \min\left\{|\mathcal{S}|, s + PT^5_{pk} - 1\right\}$ and $s(ST_{pk}) = \min\{|\mathcal{S}|, s + ST_{pk} - 1\}$. AC_{cds} and AN_{nds} are binary parameters denoting whether a chair or nurse, respectively, is available in day d at time slot s.

2.4.3.4 Next-day precedence constraints

The MMCSP considers a next-day system for scheduling of appointments. It is thus important to make sure that required appointments for blood test, consultation and non-special drug preparation are scheduled one day before the treatment appointment. The following constraints ensure these precedence relationships:

$$\sum_{s \in \mathcal{S}^5_{p1d}} \sum_{c \in \mathcal{C}^P_{p1}} x_{p1dsc} = \sum_{s \in \mathcal{S}^j_{p1d}} y^j_{p1(d-1)s} \qquad j \in \{2, 3, 4\}, p \in \mathcal{P}^{5,j}, d \in \mathcal{D}^{5,j}_{p1}, \qquad (2.16)$$

where $\mathcal{P}^{5,2} = \{p \in \mathcal{NP} : BT_{p1} = 1\}$, denotes the set of new treatment patients requesting blood test appointments, $\mathcal{P}^{5,3} = \{p \in \mathcal{NP} : |\mathcal{CA}_p| = 1\}$ defines a set of new treatment patients who need a consultation appointment before their first infusion, and $\mathcal{P}^{5,4} = \{p \in \mathcal{NP} : SD_{p1} = 0\}$, represents the set of new treatment patients requiring non-special drugs. Moreover, $\mathcal{D}^{5,j}_{p1} = \{d \in \mathcal{D}^5_{p1} : d > 1, (d-1) \in \mathcal{D}^j_{p1}, d \notin \mathcal{D}^{\mathcal{H}}\}$ denotes the set of all possible days for the patient's infusion starting from the second day of the planning horizon which are not after-weekend days and their previous day is a valid day for receiving service j. Note that, because in constraints (2.16) different appointments dates are coordinated to form a next-day system, it is necessary to make sure that the selected infusion day is not an after-weekend day and also it is a valid day with respect to the other services that should be scheduled one day before.

2.4.3.5 Same-day precedence constraints

The two proposed IP formulations differ in the way same-day precedence constraints are modeled. To explain the underlying modeling principles of these two formulations we first focus on a simplified scheduling problem involving one patient with two consecutive activities $j \rightarrow i$, that must be scheduled on a predetermined day d. Let s_i denote the starting time of activity i and PT_i its processing time. Given the precedence $j \rightarrow i$, any feasible solution must satisfy

$$s_i - s_j \geq PT_j, \tag{2.17}$$

which implies that any solution that satisfies

$$s_i - s_j < PT_j, \tag{2.18}$$

is infeasible (see Figure 2.3).

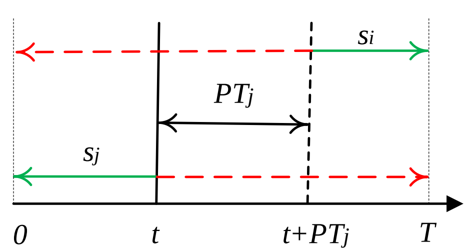


Figure 2.3: Incompatibilities between time periods when scheduling consecutive activities $j \rightarrow i$.

Let x^i_{sd} denote a binary variable equal to one if and only if activity i is scheduled to start at period $s \in \mathcal{T}$ at given day d. Condition (2.17) can be modeled as

$$\sum_s s x^i_{sd} \geq \sum_s (s + PT_j) x^j_{sd}, \tag{2.19}$$

whereas condition (2.18) can be modeled as

$$\sum_{s < t + PT_j} x^i_{sd} + \sum_{s \geq t} x^j_{sd} \leq 1 \qquad \forall t \in \mathcal{T}. \tag{2.20}$$

We note that both sets of constraints (2.19) and (2.20) provide necessary conditions for feasibility. Therefore, each of them independently provides a valid formulation for the simplified scheduling problem with one day. However, the associated linear programming (LP) relaxations may be different.

Consider now a slightly more general case in which there exists a set of days $\mathcal{D}$ to schedule one patient with two consecutive activities $j \to i$. However, we assume that activities must be scheduled during the same day. The same-day assumption and condition (2.17) can be modeled as

$$\sum_s x^i_{sd} = \sum_s x^j_{sd} \qquad \forall d \in \mathcal{D} \tag{2.21}$$

$$\sum_d \sum_s s x^i_{sd} \geq \sum_d \sum_s (s + PT_j) x^j_{sd}, \tag{2.22}$$

whereas the same-day assumption and condition (2.18) can be modeled as

$$\sum_{d' \neq d} \sum_s x^i_{sd'} + \sum_{s < t + PT_j} x^i_{sd} + \sum_{s \geq t} x^j_{sd} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D}. \tag{2.23}$$

We use either (2.21) and (2.22), or (2.23) as a basis to develop alternative valid inequalities to model several same-day precedence conditions of service pairs of new and recurring patients. In what follows, we discuss each of these precedence conditions and provide two alternative ways to model each of them.

Drug preparation-treatment precedence constraints

The preparation of special drugs must be scheduled on the same day as the treatment and be completed before the latter starts. For each $p \in \mathcal{NP}$ such that $SD_{p1} = 1$, this requirement is stated either as:

$$\sum_{s \in \mathcal{S}^5_{p1d}} \sum_{c \in \mathcal{C}^P_{p1}} x_{p1dsc} = \sum_{s \in \mathcal{S}^4_{p1d}} y^4_{p1ds} \qquad d \in \mathcal{D}^5_{p1} \cap \mathcal{D}^4_{p1} \tag{2.24}$$

$$\sum_{d \in \mathcal{D}^5_{p1}} \sum_{s \in \mathcal{S}^5_{p1d}} \sum_{c \in \mathcal{C}^P_{p1}} s x_{p1dsc} \geq \sum_{d \in \mathcal{D}^4_{p1}} \sum_{s \in \mathcal{S}^4_{p1d}} \left(s + PT^4_{p1}\right) y^4_{p1ds}, \tag{2.25}$$

or as

$$\sum_{\substack{d' \in \mathcal{D}^5_{p1} \\ d' \neq d}} \sum_{s' \in \mathcal{S}^5_{p1d'}} \sum_{c \in \mathcal{C}^P_{p1}} x_{p1d's'c} + \sum_{\substack{s' \in \mathcal{S}^5_{p1d} \\ s' < (s+PT^4_{p1})}} \sum_{c \in \mathcal{C}^P_{p1}} x_{p1ds'c} + \sum_{\substack{s' \in \mathcal{S}^4_{p1d} \\ s' \geq s}} y^4_{p1ds'} \leq 1$$

$$d \in \mathcal{D}^5_{p1} \cap \mathcal{D}^4_{p1}, s \in \mathcal{S}. \tag{2.26}$$

If treatment of a new patient is scheduled on an after-weekend day, preparation of the non-special drugs must also be scheduled on the treatment day. Thus, for each $p \in \mathcal{NP}$ such that $SD_{p1} = 0$, this condition is guaranteed by:

$$\sum_{s \in \mathcal{S}^5_{p1d}} \sum_{c \in \mathcal{C}^P_{p1}} x_{p1dsc} \leq \sum_{s \in \mathcal{S}^4_{p1d}} y^4_{p1ds} \qquad d \in \mathcal{D}^5_{p1} \cap \mathcal{D}^4_{p1} \cap \mathcal{D}^{\mathcal{H}}. \tag{2.27}$$

Please note that if non-special drug preparation is scheduled on an after-weekend day, it is not necessarily the case that treatment appointment is also assigned to the same day. Treatment appointment can also be scheduled on the following day. However, if treatment is assigned to an after-week-end day, drugs must also be prepared on the same day. In this case, it is important to make sure that drug is prepared before treatment starts. For $p \in \mathcal{NP}$ such that $SD_{p1} = 0$, this precedence constraint can be modeled either as:

$$\sum_{d \in (\mathcal{D}^5_{p1} \cap \mathcal{D}^{\mathcal{H}})} \sum_{s \in \mathcal{S}^5_{p1d}} \sum_{c \in \mathcal{C}^P_{p1}} (s - |\mathcal{S}|) x_{p1dsc} + |\mathcal{S}| \geq \sum_{d \in (\mathcal{D}^4_{p1} \cap \mathcal{D}^{\mathcal{H}})} \sum_{s \in \mathcal{S}^4_{p1d}} \left(s + PT^4_{p1}\right) y^4_{p1ds}, \quad (2.28)$$

or as

$$\sum_{\substack{s' \in \mathcal{S}^5_{p1d}: \\ s' < (s + PT^4_{p1})}} \sum_{c \in \mathcal{C}_p} x_{p1ds'c} + \sum_{\substack{s' \in \mathcal{S}^4_{p1d}: \\ s' \geq s}} y^4_{p1ds'} \leq 1 \quad d \in \mathcal{D}^5_{p1} \cap \mathcal{D}^4_{p1} \cap \mathcal{D}^{\mathcal{H}}, s \in \mathcal{S}. \quad (2.29)$$

Given that the exact treatment dates for recurring patients are known in advance, for the treatments whose target date is an after-weekend day the required drugs are considered as special drugs, i.e., they must be prepared the same day as the treatment day. Therefore, $\mathcal{D}^5_{pk} = \mathcal{D}^4_{pk}$ and $|\mathcal{D}^5_{pk}| = |\mathcal{D}^4_{pk}| = 1$, which implies there is no need to explicitly consider constraints similar to (2.24) and (2.27) for recurring patients. For each $p \in \mathcal{RP}$ and $k \in \mathcal{TA}_p$, such that $SD_{pk} = 1$, precedence constraints between drug preparation and treatment can be modeled either as:

$$\sum_{s \in \mathcal{S}^5_{pkd'}} \sum_{c \in \mathcal{C}^P_{pk}} s x_{pkd'sc} \geq \sum_{s \in \mathcal{S}^4_{pkd'}} \left(s + PT^4_{pk}\right) y^4_{pkd's}, \quad (2.30)$$

where d' denotes the singleton in $\mathcal{D}^5_{pk} = \mathcal{D}^4_{pk}$, or as

$$\sum_{\substack{s' \in \mathcal{S}^5_{pkd'}: \\ s' < (s + PT^4_{pk})}} \sum_{c \in \mathcal{C}^P_{pk}} x_{pkd's'c} + \sum_{\substack{s' \in \mathcal{S}^4_{pkd'}: \\ s' \geq s}} y^4_{pkd's'} \leq 1 \quad s \in \mathcal{S}. \quad (2.31)$$

Blood test-consultation precedence constraints

If a new patient needs both blood test and consultation appointments then both of them must be scheduled on a same day, and consultation should be started when the blood test is done and the results are ready. For each $p \in \mathcal{FP} \cup \mathcal{NP}$ such that $BC_{p1} = 1$, this can be modeled as follows:

$$\sum_{s \in \mathcal{S}^3_{p1d}} y^3_{p1ds} = \sum_{s \in \mathcal{S}^1_{p1d}} y^1_{p1ds} \qquad d \in \mathcal{D}^3_{p1} \cap \mathcal{D}^1_{p1}. \tag{2.32}$$

$$\sum_{d \in \mathcal{D}^3_{p1}} \sum_{s \in \mathcal{S}^3_{p1d}} s y^3_{p1ds} \geq \sum_{d \in \mathcal{D}^1_{p1}} \sum_{s \in \mathcal{S}^1_{p1d}} \left(s + PT^1_{p1}\right) y^1_{p1ds}, \tag{2.33}$$

or as

$$\sum_{\substack{d' \in \mathcal{D}^3_{p1}: \\ d' \neq d}} \sum_{s' \in \mathcal{S}^3_{p1d'}} y^3_{p1d's'} + \sum_{\substack{s' \in \mathcal{S}^3_{p1d}: \\ s' < (s + PT^1_{p1})}} y^3_{p1ds'} + \sum_{\substack{s' \in \mathcal{S}^1_{p1d}: \\ s' \geq s}} y^1_{p1ds'} \leq 1$$

$$d \in \mathcal{D}^3_{p1} \cap \mathcal{D}^1_{p1}, s \in \mathcal{S}. \tag{2.34}$$

Since appointment dates for new patients requesting several stages are determined by the model, it is necessary to define such relations between the assigned dates by constraints (2.32) to make sure that given appointments are coordinated properly. For recurring patients, it is assumed that the dates of appointments are known in advance and such relations are already considered. Therefore, there is no need to have similar constraints for recurring patients. However, we still need to define precedence constraints between consultation and blood test services during the same day for these patients. For each $p \in \mathcal{RP}$ and $k' \in \mathcal{CA}_p$, such that $BC_{pk} = 1$, these constraints can be stated either as:

$$\sum_{d \in \mathcal{D}^3_{pk}} \sum_{s \in \mathcal{S}^3_{pkd}} s y^3_{pkds} \geq \sum_{d \in \mathcal{D}^1_{pk}} \sum_{s \in \mathcal{S}^1_{pkd}} \left(s + PT^1_{pk}\right) y^1_{pkds}, \tag{2.35}$$

or as

$$\sum_{\substack{s'\in\mathcal{S}^3_{pkd}:\\ s'<(s+PT^1_{pk})}} y^3_{pkds'} + \sum_{\substack{s'\in\mathcal{S}^1_{pkd}:\\ s'\geq s}} y^1_{pkds'} \leq 1 \qquad d\in\mathcal{D}^3_{pk}\cap\mathcal{D}^1_{pk}, s\in\mathcal{S}. \tag{2.36}$$

Consultation-drug preparation precedence constraints

If patients need to see a physician one day before the infusion for some of their appointments, the preparation of non-special drugs must be done after the consultation is completed on the same day. For each $p\in\mathcal{NP}$ such that $SD_{p1}=0$ and $|\mathcal{CA}_p|=1$, these precedence constraints can be modeled either as:

$$\sum_{d\in\mathcal{D}^4_{p1}}\sum_{s\in\mathcal{S}^4_{p1d}} sy^4_{p1ds} \geq \sum_{d\in\mathcal{D}^3_{p1}}\sum_{s\in\mathcal{S}^3_{p1d}} \left(s+PT^3_{p1}\right) y^3_{p1ds}, \tag{2.37}$$

or as

$$\sum_{\substack{s'\in\mathcal{S}^4_{p1d}:\\ s'<(s+PT^3_{p1})}} y^4_{p1ds'} + \sum_{\substack{s'\in\mathcal{S}^3_{p1d}:\\ s'\geq s}} y^3_{p1ds'} \leq 1 \qquad d\in\mathcal{D}^4_{p1}\cap\mathcal{D}^3_{p1}, s\in\mathcal{S}. \tag{2.38}$$

Similarly, for each recurring patient $p\in\mathcal{RP}, k'\in\mathcal{CA}_P$ and $k\in\mathcal{TA}_p$ such that $SD_{pk}=0$ and $Se^C_{pk'k}=1$, the above precedence constraints can be modeled either as:

$$\sum_{d\in\mathcal{D}^4_{pk}}\sum_{s\in\mathcal{S}^4_{pkd}} sy^4_{pkds} + |\mathcal{S}|z^2_{pk} \geq \sum_{d\in\mathcal{D}^3_{pk'}}\sum_{s\in\mathcal{S}^3_{pk'd}} \left(s+PT^3_{pk'}\right) y^3_{pk'ds}, \tag{2.39}$$

or as

$$\sum_{\substack{s'\in\mathcal{S}^4_{pkd}:\\ s'<(s+PT^3_{pk})}} y^4_{pkds'} + \sum_{\substack{s'\in\mathcal{S}^3_{pkd}:\\ s'\geq s}} y^3_{pk'ds'} \leq 1 \qquad d\in\mathcal{D}^4_{pk}\cap\mathcal{D}^3_{pk}, s\in\mathcal{S}. \tag{2.40}$$

Note that constraints (2.39) should be activated if and only if both consultation and infusion appointments are scheduled.

Blood test-drug preparation precedence constraints

In a regular two-day scheduling system, blood test and special drugs preparation are scheduled on two consecutive days. However, if special drug preparation is assigned to an after-weekend day, the blood test should also be assigned to the same day and drug preparation must start after the blood test is complete and results are ready. Therefore, for each $p \in \mathcal{NP} \cup \mathcal{RP}$ and $k \in \mathcal{TA}_p$ such that $SD_{pk} = 1$ and $BT_{pk} = 1$, this condition can be modeled either as:

$$\sum_{d \in \mathcal{D}^4_{pk} \cap \mathcal{D}^{\mathcal{H}}} \sum_{s \in \mathcal{S}^4_{pkd}} (s - |\mathcal{S}|) y^4_{pkds} + |\mathcal{S}| \geq \sum_{d \in \mathcal{D}^2_{pk} \cap \mathcal{D}^{\mathcal{H}}} \sum_{s \in \mathcal{S}^2_{pkd}} \left(s + PT^2_{pk}\right) y^2_{pkds}, \tag{2.41}$$

or as

$$\sum_{\substack{s' \in \mathcal{S}^4_{pkd}: \\ s' < (s + PT^2_{pk})}} y^4_{pkds'} + \sum_{\substack{s' \in \mathcal{S}^2_{pkd}: \\ s' \geq s}} y^2_{pkds'} \leq 1 \qquad d \in \mathcal{D}^4_{pk} \cap \mathcal{D}^2_{pk} \cap \mathcal{D}^{\mathcal{H}}, s \in \mathcal{S}. \tag{2.42}$$

Similarly, if a patient's treatment appointment requires a blood test and the drug is non-special, then the pharmacist can start drug preparation after the result of the blood test is ready. For each $p \in \mathcal{NP} \cup \mathcal{RP}$ and $k \in \mathcal{TA}_p$, such that $SD_{pk} = 0$ and $BT_{pk} = 1$, this condition can be modeled either as:

$$\sum_{d \in \mathcal{D}^4_{pk}} \sum_{s \in \mathcal{S}^4_{pkd}} s y^4_{pkds} \geq \sum_{d \in \mathcal{D}^2_{pk}} \sum_{s \in \mathcal{S}^2_{pkd}} \left(s + PT^2_{pk}\right) y^2_{pkds}, \tag{2.43}$$

or as

$$\sum_{\substack{s' \in \mathcal{S}^4_{pkd}: \\ s' < (s + PT^2_{pk})}} y^4_{pkds'} + \sum_{\substack{s' \in \mathcal{S}^2_{pkd}: \\ s' \geq s}} y^2_{pkds'} \leq 1 \qquad d \in \mathcal{D}^4_{pk} \cap \mathcal{D}^2_{pk}, s \in \mathcal{S}. \tag{2.44}$$

Blood test-treatment precedence constraints

Regularly, blood test and treatment appointments of new treatment patients are scheduled on two consecutive days. However, if a treatment appointment is scheduled on an after-weekend day, the corresponding blood test appointment must also be scheduled on the same day. Therefore, for $p \in \mathcal{NP}$ such that $BT_{pk} = 1$, this condition can be modeled by:

$$\sum_{s \in \mathcal{S}^5_{p1d}} \sum_{c \in \mathcal{C}^P_{p1}} x_{p1dsc} \leq \sum_{s \in \mathcal{S}^2_{p1d}} y^2_{p1ds} \qquad d \in \mathcal{D}^5_{p1} \cap \mathcal{D}^2_{p1} \cap \mathcal{D}^H. \tag{2.45}$$

Sometimes recurring patients need to have infusions for several consecutive days. Therefore, they need to do blood test every two days. For example, suppose that a patient should receive chemotherapy from Tuesday to Friday and thus, he/she is required to do a blood test on Monday and Wednesday. Thus, the patient should have both a blood test and treatment on Wednesday and the treatment should be started after the test is done. Thus, for each $p \in \mathcal{RP}$ and $k \in \mathcal{TA}_p$ such that $BT_{p(k+1)} = 1$ and $d^5_{pk} = d^2_{p(k+1)}$, where $d^5_{pk} \in \mathcal{D}^5_{pk}$ and $d^2_{p(k+1)} \in \mathcal{D}^2_{p(k+1)}$, this situation is represented as follows:

$$\sum_{d \in \mathcal{D}^5_{pk}} \sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^P_{pk}} s x_{pkdsc} + |\mathcal{S}| z^2_{pk} \geq \sum_{d \in \mathcal{D}^2_{p(k+1)}} \sum_{s \in \mathcal{S}^2_{p(k+1)d}} \left(s + PT^2_{p(k+1)}\right) y^2_{p(k+1)ds}, \tag{2.46}$$

or as

$$\sum_{\substack{s' \in \mathcal{S}^5_{pkd}: \\ s' < (s + PT^2_{p(k+1)})}} \sum_{c \in \mathcal{C}^P_{pk}} x_{pkds'c} + \sum_{\substack{s' \in \mathcal{S}^2_{p(k+1)d}: \\ s' \geq s}} y^2_{p(k+1)ds'} \leq 1$$

$$d \in \mathcal{D}^5_{pk} \cap \mathcal{D}^2_{p(k+1)}, s \in \mathcal{S}. \tag{2.47}$$

2.4.4 Mathematical programming formulations

We have described in the previous sections the decision variables, objectives, and constraints needed to formulate the MMCSP. Depending on which condition is used to model same-day precedence constraints, the feasible region of the MMCSP can be represented in two different ways. Using condition (2.17), the MMCSP can be formulated as the following multi-objective integer linear program:

$$
\begin{aligned}
(M1)\quad \text{minimize}\quad & \left\{g_1(x), g_2(x), g_3\left(y^3\right), g_4\left(x, y^1, y^2, y^3\right), g_5(x), g_6(x), g_7(x), g_8\left(z^1, z^2\right)\right\} \\
\text{subject to}\quad & (2.1)-(2.16), (2.24), (2.25), (2.27), (2.28), (2.30), (2.32), (2.33), (2.35), \\
& (2.37), (2.39), (2.41), (2.43), (2.45), (2.46) \\
& y^i_{pkds} \in \{0,1\} \qquad i \in \{1, ..., 4\}, p \in \mathcal{P}, k \in \mathcal{CA}_p \cup \mathcal{TA}_p, d \in \mathcal{D}, s \in \mathcal{S} \\
& x_{pkdsc} \in \{0,1\} \qquad p \in \mathcal{P}, k \in k \in \mathcal{TA}_p, d \in \mathcal{D}, s \in \mathcal{S}, c \in \mathcal{C}.
\end{aligned}
$$

Using condition (2.18), the MMCSP can be formulated as the following multi-objective integer linear program:

$$
\begin{aligned}
(M2)\quad \text{minimize}\quad & \left\{g_1(x), g_2(x), g_3\left(y^3\right), g_4\left(x, y^1, y^2, y^3\right), g_5(x), g_6(x), g_7(x), g_8\left(z^1, z^2\right)\right\} \\
\text{subject to}\quad & (2.1)-(2.16), (2.26), (2.27), (2.29), (2.31), (2.34), (2.36), (2.38), (2.40), \\
& (2.42), (2.44), (2.45), (2.47) \\
& y^i_{pkds} \in \{0,1\} \qquad i \in \{1, ..., 4\}, p \in \mathcal{P}, k \in \mathcal{CA}_p \cup \mathcal{TA}_p, d \in \mathcal{D}, s \in \mathcal{S} \\
& x_{pkdsc} \in \{0,1\} \qquad p \in \mathcal{P}, k \in \mathcal{TA}_p, d \in \mathcal{D}, s \in \mathcal{S}, c \in \mathcal{C}.
\end{aligned}
$$

2.5 Sequential Approaches for the Static MMCSP

In the previous section we described two IP formulations that integrate the decisions for all five services offered by the clinic. In this section we provide two sequential approaches in which decisions are divided into two sets: consultation decisions and treatment decisions. These approaches are used to schedule all the required appointments in two sequential steps, by solving a mathematical model at each step. Decomposing the integrated problem into two smaller (and potentially easier) decision problems to solve may help to reduce the computational time at the expense of potentially decreasing the quality of the obtained solutions. Section 2.7 provides a comparison between the integrated and the two sequential approaches in a static and dynamic setting.

In the first approach, denoted as the consultation-treatment approach, we first solve a consultation model and then a treatment model. In the consultation model, we schedule all consultation and related blood test appointments. Based on the output of this model, and after updating the available resources, the treatment model schedules all chemotherapy treatments, drug preparation, and related blood test appointments.

In the consultation model, we determine three decisions ($y^1_{pkds}, y^3_{pkds}, z^1_{pk}$): date and start time of consultation blood test appointments, date and start time of consultation appointments, and the set of scheduled consultation appointments in the current planning horizon. The objectives are $g_3(y^3)$: minimizing oncologists' completion time and overtime; $g^1_4(y^1, y^3)$: patients' waiting times between blood test and consultation appointments, $g'_8(z^1)$: number of unscheduled consultation appointments, and $g'_5(y^3)$: access time for new treatment patients that also require a consultation appointment.

In the treatment model, we determine four decisions ($y^2_{pkds}, y^4_{pkds}, x_{pkdsc}, z^2_{pk}$): date and start time of treatment blood test appointments, date and start time of drug preparations, date, start time and chair of treatment appointments, and the set of scheduled treatment appointments in the current planning horizon. In this model, we consider eight objectives: $g_1(x)$: balancing workload of nurses on a daily basis, $g_2(x)$: minimizing nurses' overtime, $g^2_4(x, y^2) + g^3_4(x, y^2)$: minimizing patients' waiting times between blood test and treatment appointments, $g_5(x)$: minimizing access time of new treatment patients, $g_6(x)$: maximizing number of preferred nurse assignments, $g_7(x)$: maximizing preferred time assignments, and $g_8''(z^2)$: minimizing the number of unscheduled treatment appointments. With these objectives, the treatment model can be easily obtained from the integrated model. For additional details on the overall consultation-treatment approach, we refer to Algorithm 7 and Appendices B and C.

In the second approach, denoted as the treatment-consultation approach, we first solve a treatment model and then a consultation model. The treatment model first schedules all chemotherapy treatments, drug preparation, and related blood test appointments and then, the consultation model schedules all consultation and related blood test appointments. We note that the consultation and treatment models used in both approaches, are exactly the same with the exception that in the consultation model used in the treatment-consultation approach, we do not need to consider $g_5'(y^3)$ in the objective function given that $g_5(x)$ is considered in the first step (i.e., treatment model). For additional details on the overall treatment-consultation approach, we refer to Algorithm 8 and Appendix B.

2.6 An Online Scheduling Algorithm for the Dynamic MMCSP

In the dynamic version of the MMCSP, appointment requests arrive in real time over a rolling horizon in the form of waiting lists of newly requested appointments. Every time a waiting list is received, all appointments are scheduled by solving an instance of the static variant using either the integrated or sequential scheduling approaches. Deciding when to schedule appointments of a waiting list and the number of appointments to schedule at each waiting list, depends on the clinic scheduling policies and the scheduler preferences. Scheduling more appointments at a time (i.e., larger waiting lists) result in more efficient overall schedules, since having more information helps to construct better solutions. However, waiting to populate a waiting list with a large number of appointments causes delays in the confirmation of patient's appointments, which has a negative impact in the service provided.

Once a waiting list becomes available, the scheduling process starts by filling out a partially-filled schedule. Therefore, before starting to schedule new appointment requests, we need to determine resources' availability and capacities due to the already booked appointments. Moreover, in order to balance nurses workload more accurately, it is important to consider previously assigned loads of each nurse on the partial schedule. Also, to consider restrictions on the number of patients each nurse can monitor simultaneously, we take into account the number of already booked patients each nurse is planned to monitor. Algorithm 1 depicts the most important steps of our online algorithm.

Algorithm 1: Online Outpatient Scheduling Algorithm

Given a partial schedule:

- Define oncologists, nurses and chairs availability
- Calculate available capacity of phlebotomists and pharmacists
- Calculate current workload of nurses
- Calculate available capacity of nurses

Schedule requested appointments using either integrated or sequential approaches
Return new partial schedule
Add unscheduled appointments to the next waiting list

2.7 Computational Experiments

This section provides a summary of the results of an extensive computational study carried out to assess the empirical performance of the proposed static and dynamic outpatient scheduling models. We next present the details of the experimental design used to carry out our analyses.

2.7.1 Experimental design and evaluation

We use the weighted sum method to solve the multi-objective IPs proposed in Section 2.4. An Analytic Hierarchy Process (AHP) is applied to determine the coefficient of each objective function [56]. We note all objective function terms are first normalized, and then using the coefficients obtained by AHP, the weighted sum of all objectives is calculated. For additional details on the implementation aspects of the AHP method and obtained objective coefficients we refer to Tables 2 and 3 of Appendix A. All IPs were coded in C++ and solved with CPLEX 12.7.1 using Concert Technology on an Intel Xeon CPU E5-2687W v3 processor at 3.10 GHz and 750 GB of RAM under a Linux environment. We limit the maximum number of used threads by CPLEX to seven.

In Section 2.7.2, we first explain how the instances have been generated using real data from the SCC and the comparison criteria used in our experiments. In Section 2.7.3, we then compare the computational performance of formulations $M1$ and $M2$ on the static MMCSP. In Section 2.7.4, we analyze the impact of integrating all decisions into a single comprehensive model, as compared to using sequential approaches to solve the static and dynamic variants of MMCSP. In Section 2.7.5, we compare the integrated model with the actual plan used at the SCC at a given month. Finally, Section 2.7.6 provides further analysis and discussion.

2.7.2 Instance generation and comparison criteria

In order to build our problem instances, we used real data from SCC. For this purpose, we first recorded the actual initial partially-filled schedule of patients for consultation and treatment appointments at the beginning of the planning horizon, and then, after 22 business day (i.e., at the end of the planning horizon), we again recorded the actual complete schedule of patients in order to obtain the set of newly requested appointments. These records revealed an initial partially-filled schedule containing

1,587 pre-booked consultation and 559 pre-booked treatment appointments. During the same 22-day period, there were new requests for 733 consultation and 440 treatment appointments.

SCC has two chemotherapy stations. The first station has 13 regular chairs and the second station has 14 regular chairs and three rooms. We note that in addition to these 30 chairs and rooms, there are three other chairs and two other rooms in the first station which are reserved for drop-in patients. Therefore, we do not consider them in our study. There were 18 nurses working at SCC during the considered period. The direct nursing times required to set up a patient on chemotherapy equipment is considered to be 15 minutes and the number of patients each nurse can monitor simultaneously is four. For the purpose of our evaluation, we have focused our study on the oncology, hematology and gynecology departments which involve 25 physicians in total. The clinic operates from 8:00 to 18:00, and no patient can be set up for treatment during the lunch time, i.e., from 12:00 to 13:00. Furthermore, we have considered 15 minutes time slots which corresponds to 40 time slots per day. Pharmacy and blood lab finish at 16:00 and 15:00, respectively. Moreover, there are three pharmacists and four phlebotomists working on each day, which are not available during lunch time, i.e., from 12:00 to 13:00. The time needed for taking blood samples (15 min), preparing blood test results (15 min), and drug preparation (30 min) is considered to be equal for all patients.

We have generated three sets of instances based on real data. The first data set (T-set) contains ten small-size instances for the static case which can be solved optimally within an hour. These instances are generated using real data and in a way that reflects SCC characteristics on a small scale. The second data set (S-set) includes one instance that considers all the requested appointments of SCC to be scheduled over 22 days in the static version of the problem. For this instance, it is assumed that all the requests for the entire planning horizon are known in advance and can be scheduled in a single step. Finally, the third data set (D-set) consists of four instances that are used to solve a real-size problem dynamically. To build these instances we assume that requests arrive in the form of waiting lists and we should schedule all the requested appointments in several steps. Two, four, eight, and 12 waiting lists have been considered to build these problem instances. In the dynamic problem, there are different ways to decide when to schedule appointments. Two common approaches are based on the time or number of patients. In the former case, the scheduler defines several points of time (e.g., every three days), to schedule the appointment, while in the latter case, he/she waits until a specific number of requests arrive and then starts to schedule them. In this paper we have used the number of requests as a measure to build the waiting lists. Table 4 and Table 5 in Appendix D give the setting used for each of the generated instances.

We considered several individual performance measures derived from the components of the multi-objective function introduced in Section 2.4.2. In the following, these measures are defined and the calculation methods are described.

1. *Unbalanced workload* $\left(\frac{g_1(x)}{|\mathcal{D}|}\right)$: average daily maximum nurses' workload difference (in hours).
2. *Completion time* $\left(\frac{g_2^1(y^3)}{|\mathcal{D}|\overline{\mathcal{O}^{\mathcal{G}}}_{\mathcal{D}}}\right)$: average gynecologist daily completion time (in hours), where $\overline{\mathcal{O}^{\mathcal{G}}}_{\mathcal{D}}$ denotes the average number of gynecologists that work every day at the clinic.
3. *Oncologist overtime* $\left(\frac{g_2^2(y^3)}{|\mathcal{D}|\overline{\mathcal{O}}_{\mathcal{D}}}\right)$: average oncologist daily overtime (in hours), where $\overline{\mathcal{O}}_{\mathcal{D}}$ shows the average number of oncologists other than gynecologists that work every day at the clinic.
4. *Nurse overtime* $\left(\frac{g_3(x)}{|\mathcal{D}|\overline{\mathcal{N}}_{\mathcal{D}}}\right)$: average nurse daily overtime (in hours), where $\overline{\mathcal{N}}_{\mathcal{D}}$ represents the average number of nurses that work every day at the clinic.

5. *Waiting time* $\left(\frac{g_4(x,y^1,y^2,y^3)}{|\mathcal{D}|\overline{\mathcal{P}}_{\mathcal{D}}}\right)$: average waiting time of each scheduled patient per day (in hours), where $\overline{\mathcal{P}}_{\mathcal{D}}$ shows the average number of patients per day.

6. *Access time* $\left(\frac{\sum_{p,d,s,c} dx_{p1dsc}}{\mathcal{NP}_S}\right)$: average access time for each patient (in days), where $\mathcal{NP}_S$ is the number of scheduled new treatment patients.

7. *Non-preferred nurses*: percent of scheduled treatment appointments not assigned to a preferred nurse. It is calculated as the total number of such non-preferred assignments divided by the number of scheduled treatment appointments.

8. *Non-preferred times*: percent of scheduled treatment appointments not assigned to the preferred time slot. It is calculated as the total number non-preferred assignments divided by the number of scheduled treatment appointments.

9. *Buffer*: number of unscheduled appointments.

Given that the number of patients assigned to oncologists' break times are negligible, this performance measure has not been considered. In addition to the above individual measures, we also used an *overall performance measure indicator* (OPMI) that reflects the overall performance of the final schedule, either in static or dynamic problems. Similar to the objective function of the proposed models, OPMI is calculated as the weighted sum of normalized individual objective components. Therefore, for the static version, OPMI is equal to the objective function value of $M1$ and $M2$.

2.7.3 Comparison of formulations

We now computationally compare the two alternative IP formulations for the integrated scheduling model proposed in Section 2.4. Table 2.1 summarizes the obtained results using the T-set instances. For each formulation, we report the number of constraints, the number of explored nodes, LP gap, and computational time (in seconds).

It should be noted that a one hour time limit was considered for solving each instance and for those instances that couldn't be solved optimally within the given time limit, instead of the run time, the optimality gap is reported.

Table 2.1: Comparison of two IP formulations for the static integrated scheduling model

	Constraints		***Nodes***		***LP Gap*** (%)		***Time*** (sec)	
	$M1$	$M2$	$M1$	$M2$	$M1$	$M2$	$M1$	$M2$
T1	10,351	11,986	13,703	6,215	4.00	3.81	34.07	27.30
T2	7,765	9,204	4,658	145	1.25	1.25	1.29	0.88
T3	5,961	8,612	3,148	5,488	17.01	16.67	5.33	3.52
T4	8,714	10,431	57,815	12,290	42.90	42.79	10.32	3.10
T5	6,508	9,905	81,917	17,967	6.16	6.05	time (0.48%)	385.66
T7	8,016	11,840	1,453,714	60,795	40.04	39.50	time (0.17%)	78.86
T8	11,717	13,513	3,349	3,283	68.24	68.24	3.21	2.75
T9	16,053	19,486	7,914,245	7,836	60.41	60.33	time (0.07%)	8.64
T10	17,098	19,867	4,666	3,330	46.11	46.05	3.39	4.23
			Average				1,092.03	53.10

The results of Table 2.1 seem to indicate that $M2$ outperforms $M1$ in terms of computational time and the number of explored nodes. In addition, the LP Gaps of $M2$ are always at least as good as the ones of $M1$. Therefore, we use $M2$ for the reminder of computational experiments.

2.7.4 A comparison between integrated and sequential approaches

We tested and compared the performance of the integrated and sequential approaches using the instances from S-set and D-set. For these experiments, we considered a time limit of 24 hours for the integrated model and 12 hours for each of the two models in the sequential approaches for the static problem. As for the dynamic case, these mentioned time limits are divided by the number of waiting lists to obtain the actual considered time limit for scheduling the appointments of each waiting list.

Tables 2.2 and 2.3 summarize the obtained results with both static and dynamic variants. Since having the complete information about all requests is the best choice for scheduling appointments, we have compared the results of the integrated and sequential approaches for the dynamic version, and the results of the sequential approaches for the static version, with the results obtained by the integrated model for the static instance S1. For this purpose, the values given in columns under the headings CT(%) and TC(%) are calculated as $100(X - S1_I)/S1_I$, where X denotes the solution obtained by either one of approaches for the D-set instances (or sequential approaches for the S-set instance). Therefore, any positive value in these columns indicate that the associated performance measure is worse as compared to the integrated approach, and any negative value shows an improvement.

Table 2.2 focuses on the static variant. The second column under the heading *value* provides the value of each of the individual measures obtained by the integrated model, while columns *CT* and *TC* denote the percent deviation in the associated performance measure obtained by CT and TC approaches, respectively, when compared to the integrated model. The last row reports the optimality gap obtained by CPLEX after the given time limit. In the case of CT and TC approaches, these gaps correspond to the average gap from both consultation and treatment models.

Table 2.2: Comparison of integrated and sequential approaches for static case.

Individual performance indicator	Value	*CT* (% dev)	*TC* (% dev)
Unbalanced workload(hour)	2.10	-3.24	-3.78
Completion time(hour)	9.40	0.00	0.35
Oncologist overtime(hour)	0.34	4.87	5.31
Nurse overtime(hour)	0.06	0.00	2.50
Waiting time(hour)	0.04	-26.41	-6.67
Access time(day)	4.15	-6.06	0.00
Non-preferred nurses(%)	77.09	-6.44	-8.99
Non-preferred times(%)	98.16	-0.30	-0.29
Buffer	26.00	15.38	0.00
OPMI	35.25	7.35	-0.65
Remaining gap (%)	4.66	1.54	7.68

From Table 2.2 we note that the sequential treatment-consultation approach provides a better solution in terms of the OPMI when compared to the integrated and consultation-treatment approaches. In theory, the integrated approach will always provide a solution at least as good as the sequential approaches. However, given that none of the models associated with both approaches can be optimaly solved in one day with CPLEX, the feasible solution obtained by TC is the best obtained solution among the three approaches.

Table 2.3 compares the OPMI % deviation, computational time, and the remaining % gap obtained by CPLEX after 24 hours for the dynamic instances from D-set using each of the approaches. The reported relative values for the OPMI are calculated with respect to the integrated model S1 instance.

Table 2.3: Comparison of integrated and sequential approaches for dynamic case.

	OPMI % dev			*Time* (sec)			*Rem. % gap*		
	I	*CT*	*TC*	*I*	*CT*	*TC*	*I*	*CT*	*TC*
D1	27.13	39.72	24.05	time	time	time	2.00	0.93	2.42
D2	42.17	53.76	53.27	time	57,448	65,744	2.38	0.73	1.18
D3	62.36	63.22	70.93	75,842	28,188	27,987	0.68	0.11	0.10
D4	73.13	84.68	88.47	59,941	20,091	14,866	0.21	0.07	0.03

According to Table 2.3, in terms of quality of the obtained solutions the integrated model is the best choice for the dynamic case. However, considering time needed to solve the problem, sequential approaches can be a better choice as they can provide good quality solution in shorter times. Selecting among the approaches ultimately depends on the clinic managers' perspective and current needs. If time is a key factor for them, or if they can wait to collect as many appointments as possible, a sequential approach would be preferable. If booking appointments should be done dynamically and with higher frequency while a given time limit is respected, the integrated model is the choice as it can provide solutions with the best quality in the dynamic case.

2.7.5 Comparison between the integrated model and the actual plan

We next compare the integrated model with the actual executed plan. In order to make a fair comparison between our proposed schedule and the actual schedule executed at SCC, in this section we schedule patients only for consultation and treatment stages and we ignore other stages because we do not have information about actual blood test appointments and drug preparations of SCC during the studied period. Moreover, since we do not know exactly when each request was made, we also neglect the objective component related to the access time, since it would be unfair to compare the results without having such information. Therefore, for this comparison we have used objective coefficients as presented in the second column of Table 3 in Appendix D. Similar to the previous section, we considered a time limit of 24 hours for the scheduling of all the requested appointments associated with one or more waiting lists.

A summary of the results is given in Table 2.4. Similar to Section 2.7.4, we compared the obtained results for dynamic instances and the actual plan with the results obtained from the static variant using the integrated model.

Table 2.4 shows that when using the integrated model most performance indicators can be significantly improved in both static and dynamic settings when compared to the executed schedule. We also calculated the OPMI associated with the executed plan and compared it with our obtained OPMI for the static problem. The results show that using the proposed model, the OPMI can also be improved significantly. Given that we considered a time limit for solving the problem instances, we were not able to solve all models to optimality. Within the given time limit the final optimality gap for the S1 problem is 29.54% and the average optimality gap over all waiting lists for D-set instances are 12.21%, 9.15%, 3.16%, and 1.16%, respectively. For each model, the optimality gap is calculated as (*best integer* - *best bound*)/*best integer*.

Table 2.4: Comparison of the integrated model with the executed actual plan

(a) Solution of the static problem (S1)

	S1
(1) *Unbalanced workload(hour)*	1.74
(2) *Completion time(hour)*	9.40
(3) *Oncologist overtime(hour)*	0.26
(4) *Nurse overtime(hour)*	0.03
(5) *Non-preferred nurses(%)*	74.18
(6) *Non-preferred times(%)*	98.22
(7) *Buffer*	1.00
(8) *OPMI*	8.91

(b) Comparison of the D-set instances and the real plan with the static problem

	D1(%)	D2(%)	D3(%)	D4(%)	Actual plan(%)
(1)	86.27	173.80	223.53	242.48	535.95
(2)	-0.23	3.80	4.03	8.17	7.36
(3)	4.65	9.88	8.14	6.40	1.74
(4)	41.67	166.67	37.50	308.33	575.005
(5)	3.54	8.88	11.79	13.11	12.37
(6)	0.00	0.00	0.14	0.00	0.28
(7)	-100	-100	0.00	-100.00	-100.00
(8)	60.81	132.58	176.05	189.77	422.57

As expected, solving the dynamic problem with fewer waiting lists, i.e., having more information before starting the scheduling process, results in better schedules, and as the number of waiting lists increases, the efficiency of the final schedule decreases.

2.7.6 Analysis and discussion

In this section we perform several analyses to assess the impact of different parameters and problem settings on the quality of the obtained solutions when using our proposed approaches.

2.7.6.1 Functional care approach v.s. primary care approach

Functional care delivery and primary care delivery models are two common methods for assigning nurses to patients for chemotherapy appointments. We recall that SCC uses a primary care delivery model for nurse assignments. However, in order to examine the impact of nurse-to-patient assignment policies on the quality of obtained schedules, we can adapt our models to schedule appointments considering a functional care approach and compare the obtained schedule with the one considering a primary care approach. For this comparison, we have considered a time limit of one hour divided by number of waiting lists. In order to fairly compare the results, we also considered an optimality gap limit for solving the models, so that we obtain a schedule with similar optimality gaps for both considered policies. The considered optimality gap for the static variant is 12%. For the D-set instances this gap is considered as 5%, 8.5%, 5.5%, and 3%. These values are selected based on the maximum gap obtained by solving the instances in one hour. A summary of results are reported in Table 2.5, where Pr and *Fn* are used to refer to primary and functional models, respectively.

Table 2.5: Impact of nurse-patient assignment policies on the quality of obtained schedules

	S1		D1		D2		D3		D4	
	Pr	*Fn*	*Pr*	*Fn*	*Pr*	*Fn*	*Pr*	*Fn*	*Pr*	*Fn*
Unbalanced workload	2.89	3.00	4.45	4.60	5.65	5.97	6.45	6.73	6.56	6.66
Completion time	9.51	9.42	9.46	9.44	9.98	10.06	9.97	10.01	11.07	11.05
Oncologist overtime	0.47	0.39	0.38	0.38	0.53	0.43	0.45	0.47	0.46	0.49
Nurse overtime	0.12	0.10	0.12	0.11	0.19	0.18	0.14	0.13	0.19	0.15
Waiting time	0.03	0.03	0.04	0.04	0.04	0.06	0.05	0.06	0.05	0.05
Access time	4.06	4.09	4.21	4.30	3.88	4.03	4.13	4.18	3.97	3.88
Non-preferred nurses	83.59	91.23	78.24	89.87	77.94	90.52	82.36	91.87	84.05	92.44
Non-preferred times	98.02	97.88	98.29	98.57	98.57	98.56	98.69	98.55	98.97	98.40
Buffer	26.00	26.00	30.00	32.00	35.00	37.00	47.00	44.00	56.00	45.00
OPMI	37.67	37.97	45.31	47.23	51.83	54.07	62.31	61.09	67.86	61.10
Time (sec)	3,480	1,570	1,906	926	655	682	396	398	543	584
Gap (%)	11.16	11.83	4.90	4.93	7.79	6.26	4.16	4.36	2.12	2.24

We note that the performance of these policies are similar for most performance indicators, expect for the non-preferred nurses in which the primary mode is always better than the functional mode. When we have fewer waiting lists, using a primary model results in slightly better schedules, but it needs more time to solve the problem.

As the number of lists increases, a functional model provides slightly better solutions in the same amount of time.

2.7.6.2 Impact of resource capacity

In order to assess the impact of the number of phlebotomists and pharmacists on the quality of the final schedule, we next present the results of experiments obtained from several scenarios considering an increase or decrease of capacity of these resources using the integrated model. We considered a time limit of one hour divided by the number of waiting lists. Similar to the experiments of the previous section, we also considered an optimality gap limit when solving the models. The considered optimality gap for the static problem is 20%. For the D-set instances this gap is considered as 10%, 11.5%, 5.5%, and 3.5%. These values are selected taking into account the maximum gap that was obtained by solving the instances in one hour.

Table 2.6: Solution of the problem instances with the current capacity

	S1	D1	D2	D3	D4
(1) *Unbalanced workload*(*hour*)	3.99	5.24	5.80	6.45	6.55
(2) *Completion time*(*hour*)	9.52	9.53	10.53	9.97	11.04
(3) *Oncologist overtime*(*hour*)	0.56	0.43	0.52	0.45	0.45
(4) *Nurse overtime*(*hour*)	0.15	0.10	0.22	0.14	0.15
(5) *Waiting time*(*hour*)	0.03	0.05	0.05	0.05	0.05
(6) *Access time*(*day*)	4.06	4.24	3.88	4.13	3.97
(7) *Non-preferred nurses*(%)	80.06	78.68	80.20	82.36	84.23
(8) *Non-preferred times*(%)	97.88	98.28	98.42	98.69	98.83
(9) *Buffer*	26.00	34.00	36.00	47.00	48.00
(10) *OPMI*	41.17	50.23	53.07	62.31	62.60
Time (sec)	2,187	865	628	394	545
Gap (%)	18.73	8.31	9.59	4.16	2.53

Table 2.6 reports the obtained results within the given limits for the current capacities, i.e., $R^B_{ds} = 4$ and $R^D_{ds} = 3$. Table 2.7 represents the relative OPMIs with respect to the OPMI considering the current capacities, time needed to solve the problem, and the average optimality gap. In this table, R'^B_{ds} and R'^D_{ds} represent

changes occurred in the number of available phlebotomists and pharmacists at each time slot, respectively. According to this table, it seems that with the current patient mix and considered parameters, four phlebotomists and five pharmacists are ideal for obtaining the best solution.

Table 2.7: Comparing the obtained solutions using different capacities with the current capacity

	R'^{B}_{ds}	R'^{D}_{ds}	OPMI (%)	Time (sec)	Gap(%)
	0	-1	83.28	2,106	16.14
	-1	0	56.06	2,371	10.64
S1	1	0	0.37	2,007	19.21
	0	1	-31.35	2,483	19.79
	0	2	-47.78	2,919	19.61
	0	-1	66.98	829	7.96
	-1	0	50.05	1,043	7.86
D1	1	0	-2.71	751	9.68
	0	1	-32.17	1,066	8.98
	0	2	-49.50	1,750	9.50
	0	-1	64.69	653	6.95
	-1	0	55.21	638	7.39
D2	1	0	-2.47	997	9.31
	0	1	-23.63	996	9.65
	0	2	-42.16	962	10.92
	0	-1	46.25	386	3.58
	-1	0	43.34	397	3.94
D3	1	0	-11.37	399	4.54
	0	1	-26.84	481	4.61
	0	2	-43.67	430	5.19
	0	-1	44.65	560	2.26
	-1	0	42.73	599	2.07
D4	1	0	-9.14	671	2.58
	0	1	-25.73	694	2.66
	0	2	-43.64	614	3.07

2.8 Conclusion

This paper studied a multi-appointment, multi-stage chemotherapy scheduling problem considering unique characteristics and realistic assumptions which arises in a major cancer center in Canada. In order to coordinate all the required patient appointments properly while utilizing valuable resources efficiently, an integrated model was proposed to schedule multiple appointment requests of different types of patients for each of the clinic stages including blood test, consultation, pharmacy, and treatment stages. We considered that patients may follow different paths in the clinic and may require special chemotherapy equipment for their treatment. Moreover, chemotherapy drugs' shelf-life has been taken into account for scheduling decisions. We modeled and solved the static and dynamic cases of the studied problem. In addition to an integrated scheduling model, two sequential approaches were also proposed to assess the value of integration. Extensive computational experiments were carried out using real data to evaluate the performance of the proposed methods. The results showed the potential for significant improvements in the actual schedule with respect to several performance measures.

In this paper, we calculated the workload of the nurses by simply considering the amount of time that each nurse spends to set up and monitor patients during the day. However, we know that different patients need different levels of attention due to their health status and acuity level. Thus, incorporating acuity levels of patients in future researches to compute the workload of nurses results in more accurate results. Flexibility in recurring patients' appointment dates and possibility of appointment cancellations can also be considered in future works.

Chapter 3

Integrated Consultation and Chemotherapy Scheduling under Uncertain Treatment Times

The content of this chapter will be submitted as a manuscript for publication to the journal *Expert Systems with Applications* in September 2020.

Abstract

This paper studies the integrated scheduling of consultation and treatment appointments for chemotherapy patients, while taking into account stochastic duration of injection. Patients may require one or both types of consultation and treatment appointments. The objective is to minimize the clinic's overtime and the waiting time of patients. To formulate the problem, we develop two two-stage stochastic programming models. We also propose a sample average approximation algorithm as the solution method. To improve the efficiency of our solution approach, we devise a specialized algorithm that quickly evaluates a given first-stage solution for a large set of scenarios, without solving the second-stage models. Several computa-

tional experiments are carried out to evaluate the performance of proposed models and algorithms.

3.1 Introduction

Oncology clinics providing chemotherapy treatments are among the most demanded outpatient clinics, due to the increasing cancer rates. Based on a Canadian Cancer Society report[1], the number of annual cancer cases is increasing. However, the mortality rate of cancer has been decreased in recent years because of the new advances in cancer treatments. To keep this decreasing trend, it is necessary to provide high quality services to the increasing number of cancer patients. A vital step in this regard, is to design scheduling tools that enable health providers to serve more patients with a satisfactory service quality, while efficiently utilizing the available resources.

Chemotherapy is one of the most effective cancer treatments that uses drugs to destroy cancer cells. Patients receive chemotherapy drugs through a variety of methods. These drugs are commonly administered intravenously, that may take from several minutes to hours to influence the patient, depending on the drug's type and dosage. Cancer patients usually require several chemotherapy sessions for a complete cure plan. The appointment dates and the drugs' type and dosage are prescribed in advance by an oncologist in the first consultation appointment. The treatment plan may require patients to visit their oncologist before some of their chemotherapy appointments, so that their health conditions are examined and the prescribed drug and dosage are revised if necessary.

There are two common procedures followed by oncology clinics to schedule consultations with oncologists and chemotherapy appointments. In *next-day scheduling system*, clinics schedule these two appointments on two different days , while in *same-*

[1]http://www.cancer.ca/en/cancer-information/cancer-101/canadian-cancer-statistics-publication/?region=qc

day scheduling system, they are scheduled on the same day [20]. The next-day system is more flexible as clinic staff has more time to resolve pharmacy-related issues and react appropriately to the last minute changes such as modifications in treatment. However, it requires patients to come twice to the clinic that increases transportation costs and the overall time patients spend to receive the service. Same-day scheduling system is more cost and time effective for the patients, however, it is less flexible for health staff to mitigate the negative impacts of the last-minute changes. Therefore, the next-day scheduling system is usually preferred by the staff, while same-day system is a better choice for the patients [42]. In this paper, we focus on the same-day scheduling policy.

Stochastic service time is one of the main challenges in any appointment scheduling problem. In absence of considering uncertain service times, the obtained appointment schedules are inefficient and do not satisfy the expectations of the decision maker in reality. There are several reasons for the uncertain duration of chemotherapy treatments. The main reasons are 1) early termination of chemotherapy infusion when patients cannot tolerate the injected drug and 2) longer infusion times due to the complications caused by patients' adverse reactions to the drugs. These deviations from the expected treatment time affect all other appointments and increase patients' waiting times and overtime of resources . Thus, incorporating uncertainty in the scheduling process is vital and results in more efficient and reliable schedules.

There are several papers that study chemotherapy planning and scheduling problems, however, there is still the lack of a study that considers stochastic treatment times along with other realistic assumptions in the literature. To the best of our knowledge, [13] and [22] are the only optimization-based papers that considered uncertain infusion durations for a set of patients. Castaing et al. [13] focus on scheduling treatment appointments of patients with respect to a given sequence. Garaix et al. [22] study the problem of determining a global sequence for both consultation and treatment appointments considering unlimited capacity for the nurses. Therefore, there is still a lack of studies in the literature that simultaneously determine the sequence of patients and the treatment start times considering limited capacity of resources. To address the existing gaps in the literature, this paper studies a daily consultation and chemotherapy scheduling problem (CCSP) in oncology clinics where treatment durations are stochastic. We consider three types of patients: 1) the first category of patients request to have only consultation visits with oncologists, 2) the second type of patients require both consultation and chemotherapy appointments, and 3) the last category of patients need to only receive the chemotherapy treatment. Since the variability of consultation times is not as much as it is for the treatment times, in this paper we only consider the variability in injection times and assume the consultation times to be deterministic. Furthermore, for the second type of patients, we take into account the required time for preparing the chemotherapy drug after the consultation and consider it to be deterministic. We assume that there are always enough resources available to prepare the chemotherapy drugs. Therefore, for the third type of patients, we are sure that the required drugs are prepared in advance and there is no need to consider drug preparation time. However, for the second type of patients, as the drug type or dosage may be revised by oncologists, the preparation can only start when the consultation is completed. Therefore, for these patients, there is always a gap between the two appointments to address preparation of the drug. We also take into account the fact that, each nurse can set up at most one patient on a chemotherapy chair, while he/she can monitor several patients simultaneously, during the rest of the infusion.

The main contributions of this paper are as follows:

- *Problem definition:* We consider simultaneous scheduling of consultation and treatment appointments for different types of patients, while incorporating uncertainty in the treatment times. We take into account several operational decisions such as assignment of patients to nurses and chairs, and also sequencing of patients on the same chair and/or nurse.

- *Formulation:* We develop two two-stage stochastic programming models to formulate the problem. One of these formulations uses the classic constraints of machine scheduling to determine the assignment of patients to resources and to sequence them. In the other formulation, assignment and sequences are determined using similar constraints that formulate the multi-traveling salesman problem (multi-TSP).

- *Solution method:* We use a sample average approximation (SAA) scheme to obtain high quality feasible solutions. To enhance SAA, we also develop an efficient specialized algorithm that quickly evaluates any given first-stage solution for a large number of scenarios.

The remainder of this paper is organized as follows. In Section 3.2, we provide a literature review. In Section 3.3, we describe the CCSP in more detail. Then, in Section 3.4, we propose two two-stage stochastic programming models for the stochastic CCSP. In Section 3.5, we present a sample average approximation algorithm and a specialized algorithm to solve the problem with a large number of scenarios. We report the computational results in Section 3.6. Finally, we provide the conclusion and some future research avenues in Section 3.7.

3.2 Literature Review

Planning and scheduling problems arising in oncology clinics are examples of multidisciplinary and multi-stage scheduling problems. Leeftink et al. [43] and Marynissen and Demeulemeester [49] have recently reviewed such scheduling problems in healthcare. For more specific review on the planning of outpatient chemotherapy clinics, please refer to [41].

In recent years, chemotherapy planning and scheduling problems have attracted the attention of many researchers and several studies have been conducted on these topics. However, according to the [49], most researchers restrictively focus only on the scheduling of patients for treatment appointments (e.g., [2, 7, 13, 18, 25, 27, 32, 35, 37, 38, 44, 65]). There are a few papers that have also considered other types of appointments such as consultation and blood test appointments. Liang et al. [45], propose a discreet event simulation for an oncology clinic where different types of patients request for consultation and/or chemotherapy appointments. In this simulation model, the authors have addressed unpunctual arrivals, stochastic service times, add-ons, and cancellations. Bouras et al. [9] develop a mixed integer programming model for a daily scheduling of consultation and treatment appointments in an oncology clinic, that minimizes patients waiting times between the appointments. In this work, authors consider only one patient type by assuming that all patients have both types of appointments. Suss et al. [63] propose a scheduling algorithm based on lean principles that determines the best patient arrival rate to an oncology clinic such that different types of patients can complete all the required appointments including blood test, consultation and treatment, with the minimum waiting time. They use a discrete event simulation model to evaluate the performance of the proposed algorithm. Haghi et al. [31] study a comprehensive multi-appointment, multi-stage scheduling problem in an oncology clinic. The authors present integrated and sequential approaches to schedule multiple-appointment requests of different types of patients for blood test, consultation, and chemotherapy appointments over a planning horizon. They have also developed an online scheduling tool to accommodate arriving requests dynamically.

The majority of recent literature addressing chemotherapy scheduling related topics, have either considered deterministic versions of the problem (see, [37, 44, 58]), or have studied online scheduling and dynamic nature of the appointment arrivals (see, [18, 27, 32, 38, 54]). However, the literature on the scheduling of chemotherapy appointments under uncertain infusion durations is scare. Castaing et al. [13] present a two-stage stochastic programming approach for the chemotherapy scheduling problem with uncertain duration of treatment appointments, where the objective is to minimize the expected waiting times of patients and the makespan. In this work, the first-stage decisions fix the appointment times and then in the second stage, the assignment of patients to chairs are determined and waiting time of patients and the makespan are calculated. The authors design a heuristic algorithm to solve instances with one nurse, three chairs and 12 patients. They also propose several lower-bounds for the problem to evaluate the quality of the solution obtained by the heuristic. This paper considers only the treatment appointments, and solves small instances. Furthermore, the authors assume that the sequence of patients is known in advance that significantly simplifies the problem. Alvarado and Ntaimo [2] propose three mean-risk stochastic programming formulations to schedule multiple chemotherapy appointments for a single new patient considering uncertainty in acuity level of the patient, appointment duration and number of nurses on duty on each day. This work focuses on the treatment appointment only, and considers the scheduling of only one patient. Göçgün [25] studies the problem of determining chemotherapy appointment dates dynamically considering the probability of appointment cancellations and formulates the problem as a Markov decision process. To solve the problem, the author develops a linear-programming-based approximate dynamic programming method that provides approximate solutions. The focus of this paper is only on the treatment stage and not the consultation stage. Furthermore, the proposed method only determines appointment dates and neglects the assignment of start times and required resources. The author also assumes that all treatment times are equal to exactly one time slot and ignored the variability of treatment duration. Garaix et al. [22] develop a heuristic approach for a same-day consultation and chemotherapy treatment scheduling problem considering uncertain treatment cancellations. Using this heuristic, authors determine a visit sequence of patients that is the same for both consultation and treatment appointments. Moreover, they assume that all oncologists are available all day and consultation times are equal for all patients. Also, the authors assume that the number of available nurses is large enough to serve all patients, and therefore do not consider them. Furthermore, they do not determine the assignment of patients to the chairs. They also assume that all patients require both consultation and treatment appointments, and thus, consider only one type of patients.

Our work differs from all the mentioned studies in several aspects. First, we consider three different patient types that require a consultation appointment, a treatment appointment, or both appointments. Second, we consider the limited number of nurses along with the chairs, and we also determine the exact assignments of patients to these resources. Third, we assume that the sequences that patients visit oncologists, chairs, and nurses are not given in advance. In our models, these sequences are determined independently, and thus could be different. This flexibility increases the chance of obtaining higher quality solutions.

3.3 Problem Definition and Notation

We consider $\mathcal{P}^C$ and $\mathcal{P}^T$ as the sets of patients requesting for the oncologist consultation and chemotherapy appointments, respectively. Furthermore, we let $\mathcal{P}^i$, $i \in \{1, 2, 3\}$ denote the set of patients of type i. We remind that the first-type patients only ask for the consultation appointment with one of the oncologists, patients of type two need both consultation and treatment appointments, and the last category of patients only need a treatment appointment. Therefore, we have $\mathcal{P}^C = \mathcal{P}^1 \cup \mathcal{P}^2$ and $\mathcal{P}^T = \mathcal{P}^2 \cup \mathcal{P}^3$. For $p \in \mathcal{P}^C$, we also define the deterministic parameter CT_p as the required time for consultation appointment. For $p \in \mathcal{P}^2$, we let DT_p represent the preparation time of chemotherapy drug, that is also a deterministic parameter.

In our problem, we assume that there are a limited number of oncologists, chairs, and nurses in the consultation and treatment stages. We use $\mathcal{O}$, $\mathcal{N}$ and $\mathcal{C}$ to denote the sets of oncologists, nurses and chairs, respectively. We assume that the oncologist of each patient is known in advance, and all chairs and nurses are identical. We define $\mathcal{P}_o^O \subseteq \mathcal{P}^C$ as the set of all patients of oncologist $o \in \mathcal{O}$. We also assume that each oncologist $o \in \mathcal{O}$ has a specific working time window with a start time of S_o^O and a finish time of F_o^O. For $p \in \mathcal{P}^C$, we let O_p denote the oncologist of the patient p.

We consider that the chemotherapy appointment for each patient consists of setup time and the infusion time. Each nurse can only set up one patient at a time, while he/she can monitor several patients during the infusion time. We consider the setup and infusion durations to be stochastic and define Ω as the given set of stochastic scenarios. Each scenario $\omega \in \Omega$ provides the realizations of the setup and infusion times of patients in that scenario. For $p \in \mathcal{P}^T$, $ST_{p\omega}$ and $TT_{p\omega}$ respectively denote the setup and infusion time of patient p in scenario ω.

In this problem, the goal is to schedule all the requested consultation and chemotherapy treatment appointments on a given day. In the first stage, the decision maker must decide about 1) allocation of the first appointment time of all patients, 2) allocation of patients to nurses and chairs in the treatment stage, and finally 3) sequencing of patients allocated to the same chair/nurse. The actual start time of treatment appointments are later observed in the second stage based on the realization of stochastic treatment times. The objective is to minimize the expected overtime of the clinic and waiting times of the patients.

3.4 Two-Stage Stochastic Programming Models

In this section, we propose two two-stage stochastic programming models to formulate the stochastic CCSP. In both of these models, in the first stage, we schedule all consultation appointments, assign chairs and nurses to patients requiring treatment appointments, and determine the sequence of patients assigned to the same chair/nurse. We also decide about treatment appointment times of type-three patients in the first stage. This is because we must inform all patients about their first appointment times (consultation or treatment) prior to the appointment day so that they can plan their arrival to the clinic. Since type-three patients have only chemotherapy appointments, their initial appointment times must be scheduled in the first stage. However, In the second stage, we determine the actual treatment appointment times of type-two and type-three patients, along with computing the clinic's over time and patients' waiting times in each scenario.

3.4.1 Model 1

To formulate the first two-stage stochastic programming model, we define the following first-stage variables: $v_{pp'}$ is a binary variable taking value 1 if and only if patients $p \in \mathcal{P}^C$ is visited before patient $p' \in \mathcal{P}^C$ by the same oncologist (not necessarily immediately). For each patient $p \in \mathcal{P}^C$, we also define sc_p as the scheduled start time of the consultation appointment. The binary variable x_{pcn} takes 1 if and only if we assign patient $p \in \mathcal{P}^T$ to chair $c \in \mathcal{C}$ and nurse $n \in \mathcal{N}$ during the chemotherapy treatment. Furthermore, we let binary variables $z_{pp'c}$ and $y_{pp'n}$ represent the sequence of patients on chairs and nurses, respectively. Variable $z_{pp'c}$ takes 1 if and only if we schedule patient $p \in \mathcal{P}^T$ before patient $p' \in \mathcal{P}^T$ (not necessarily immediately) on chair c. Similarly, $y_{pp'n}$ takes 1 if and only if nurse n sets up patient $p \in \mathcal{P}^T$ before patient $p' \in \mathcal{P}^T$ (not necessarily immediately). Finally, we define St^0_p as the initial scheduled start time for the treatment of patient $p \in \mathcal{P}^3$.

We also define the following second-stage variables: $st_{p\omega}$ denotes the actual chemotherapy appointment time for $p \in \mathcal{P}^T$ in scenario $\omega \in \Omega$. To consider clinic's overtime in the model, we define two overtime variables ot^1_ω and ot^2_ω. When overtime is required, ot^1_ω increases to a maximum threshold O^{max} that is a given parameter. Beyond this threshold, ot^2_ω takes a positive value. In order to avoid excessive overtime, ot^2_ω has a larger positive coefficient than ot^1_ω in the objective function. We assume that regular clinic hours start at S^C and end at F^C. Finally, we define $wt_{p\omega}$ as the waiting time of patient $p \in \mathcal{P}^T$ in scenario $\omega \in \Omega$.

According to the defined variables, we formulate the stochastic CCSP as the following two-stage stochastic program:s;

$$
\text{(M1)} \quad \text{minimize} \quad \mathop{\mathbb{E}}_{\omega\in\Omega}[Q(sc,x,z,y,\omega)] \tag{3.1}
$$

$$
\text{subject to} \quad v_{pp'} + v_{p'p} = 1 \qquad o \in \mathcal{O}, p, p' \in \mathcal{P}^O : p > p' \tag{3.2}
$$

$$
sc_{p'} \geq sc_p + CT_p - M(1 - v_{pp'}) \quad o \in \mathcal{O}, p, p' \in \mathcal{P}^O : p \neq p' \tag{3.3}
$$

$$
\sum_{c\in\mathcal{C}} \sum_{n\in\mathcal{N}} x_{pcn} = 1 \qquad p \in \mathcal{P}^T \tag{3.4}
$$

$$
z_{pp'c} + z_{p'pc} \leq \sum_{n\in\mathcal{N}} x_{pcn} \qquad p, p' \in \mathcal{P}^T : p > p', c \in \mathcal{C} \tag{3.5}
$$

$$
z_{pp'c} + z_{p'pc} \leq \sum_{n\in\mathcal{N}} x_{p'cn} \qquad p, p' \in \mathcal{P}^T : p > p', c \in \mathcal{C} \tag{3.6}
$$

$$
z_{pp'c} + z_{p'pc} \geq \sum_{n\in\mathcal{N}} x_{pcn} + \sum_{n\in\mathcal{N}} x_{p'cn} - 1 \qquad p, p' \in \mathcal{P}^T : p > p', c \in \mathcal{C} \tag{3.7}
$$

$$
y_{pp'n} + y_{p'pn} \leq \sum_{c\in\mathcal{C}} x_{pcn} \qquad p, p' \in \mathcal{P}^T : p > p', n \in \mathcal{N} \tag{3.8}
$$

$$
y_{pp'n} + y_{p'pn} \leq \sum_{c\in\mathcal{C}} x_{p'cn} \qquad p, p' \in \mathcal{P}^T : p > p', n \in \mathcal{N} \tag{3.9}
$$

$$
y_{pp'n} + y_{p'pn} \geq \sum_{c\in\mathcal{C}} x_{pcn} + \sum_{c\in\mathcal{C}} x_{p'cn} - 1
$$

$$p, p' \in \mathcal{P}^T : p > p', n \in \mathcal{N} \quad (3.10)$$

$$S^O_{O_p} \leq sc_p \leq F^O_{O_p} - CT_p \qquad p \in \mathcal{P}^C \quad (3.11)$$

$$st^0_p \geq S^C \qquad p \in \mathcal{P}^3 \quad (3.12)$$

$$v_{pp'} \in \{0, 1\} \qquad p, p' \in \mathcal{P}^C : p \neq p' \quad (3.13)$$

$$x_{pcn} \in \{0, 1\} \qquad p \in \mathcal{P}^T, c \in \mathcal{C}, n \in \mathcal{N} \quad (3.14)$$

$$z_{pp'c} \in \{0, 1\} \qquad p, p' \in \mathcal{P}^T : p \neq p', c \in \mathcal{C} \quad (3.15)$$

$$y_{pp'n} \in \{0, 1\} \qquad p, p' \in \mathcal{P}^T : p \neq p', n \in \mathcal{N}, \quad (3.16)$$

where

$$Q(sc, x, z, y, \omega) = \text{minimize } \alpha_1 ot^1_\omega + \alpha_2 ot^2_\omega + \beta_1 \sum_{p \in \mathcal{P}^2} wt_{p\omega} + \beta_2 \sum_{p \in \mathcal{P}^3} wt_{p\omega} \quad (3.17)$$

$$\text{subject to } st_{p\omega} \geq sc_p + CT_p + DT_p \qquad p \in \mathcal{P}^2 \quad (3.18)$$

$$st_{p'\omega} \geq st_{p\omega} + ST_{p\omega} + TT_{p\omega} - M(1 - \sum_{c \in \mathcal{C}} z_{pp'c})$$

$$p, p' \in \mathcal{P}^T : p \neq p' \quad (3.19)$$

$$st_{p'\omega} \geq st_{p\omega} + ST_{p\omega} - M(1 - \sum_{n \in \mathcal{N}} y_{pp'n})$$

$$p, p' \in \mathcal{P}^T : p \neq p' \quad (3.20)$$

$$st_{p\omega} \geq st^0_p \qquad p \in \mathcal{P}^3 \quad (3.21)$$

$$wt_{p\omega} \geq st_{p\omega} - st^0_p \qquad p \in \mathcal{P}^3 \quad (3.22)$$

$$wt_{p\omega} \geq st_{p\omega} - (sc_p + CT_p + DT_p) \qquad p \in \mathcal{P}^2 \quad (3.23)$$

$$ot^1_\omega + ot^2_\omega \geq st_{p\omega} + ST_{p\omega} + TT_{p\omega} - F^C \quad p \in \mathcal{P}^T \quad (3.24)$$

$$st_{p\omega} \geq S^C, wt_{p\omega} \geq 0 \qquad p \in \mathcal{P}^T \quad (3.25)$$

$$0 \leq ot^1_\omega \leq O^{max}, ot^2_\omega \geq 0. \quad (3.26)$$

Objective function (3.1) minimizes the expected second-stage objective. Constraints (3.2) determine the sequence of patients for each oncologist. Constraints (3.3) calculate the start time of consultation appointments based on the defined sequence. Constraints (3.4) assign a chair and a nurse to each patient with a chemotherapy appointment. Constraints (3.5)-(3.7) determine the sequence of patients assigned to the same chair. Similarly, constraints (3.8)-(3.10) fix the sequence of the patients that are allocated to the same nurse. Constraints (3.11) guarantee that consultation appointments of patents respect the working time window of the corresponding oncologist. Constraints (3.12) indicate that the start time of treatment appointments for type-three patients must be after the opening time of the clinic. Constraints (3.13)-(3.16) state the standard integrality conditions of the decision variables.

The second-stage objective function (3.17) minimizes the clinic's over time and the patients' waiting time. As discussed earlier, we consider two types of overtime (i.e., ot^1_ω and ot^2_ω) in the objective function. We use α_1 and α_2 to define the weight of these overtimes. We also consider β_1 and β_2 as the weights of waiting time for type-two and type-three patients, respectively.

Constraints (3.18) ensure that treatment appointment of a type-two patient can start after the finish time of the consultation appointment and the preparation of the chemotherapy drug. Constraints (3.19) state that if patient p' is scheduled after patient p on the same chair, then his/her start time of treatment must be greater than or equal to the time that patient p leaves the chair. Similarly, Constraints (3.20) guarantee that if patient p' is to be served after patient p by the same nurse, then his/her start time of treatment must be greater than or equal to the time that the nurse finishes setting up patient p. In constraints (3.19)-(3.20), M is a sufficiently big number. Constraints (3.21) ensure that the actual start time of treatment for any type-three patient must be larger than the scheduled time. This is to make sure that the patient is present at the clinic. Furthermore, constraints (3.22) compute the waiting times of these patients as the gap between the initial start time fixed in the first stage and the actual start time in the second stage. Similarly, constraints (3.23) calculate the waiting time of type-two patients as the difference between their consultation and treatment appointments. Constraints (3.24) compute the clinic's overtime with respect to the regular finish time of the clinic. Constraints (3.25)-(3.26) state the standard non-negativity conditions of the second-stage decision variables.

3.4.2 Model 2

In this section, we provide an alternative formulation for the stochastic CCSP, that partially models the problem as a multi-traveling salesman problem in the first stage. Multi-TSP is a generalization of the traveling salesman problem where the decision maker determines a set of tours for several salesmen that start from and return to the same city. In multi-TSP, each city is visited exactly once [6]. In the CCSP, we could consider chairs as salesmen, and the patients with treatment appointments as the cities that must be visited exactly once by one of the salesmen (chairs). We also define a dummy patient 0 as the origin city. Therefore, we could formulate the problem of allocating patients to the chairs and their sequencing as a multi-TSP. Patients that are assigned to the same tour must be served by a same chair. We let binary variable $z'_{pp'}$ take 1 if and only if there exists a tour from patient $p \in \mathcal{P}^T \cup \{0\}$ to the patient $p' \in \mathcal{P}^T \cup \{0\}$. We remind that $z'_{0p} = 1$ means that patient p is the first patient to be visited in one of the chair tours, and $z'_{p0} = 1$ represents the case that patient p is the last patient in a tour. According to these definitions, we formulate the assignment of patients to chairs and their sequencing by the following constraints:

$$\sum_{p \in \mathcal{P}^T} z'_{0p} \leq |\mathcal{C}| \tag{3.27}$$

$$\sum_{p \in \mathcal{P}^T} z'_{p0} \leq |\mathcal{C}| \tag{3.28}$$

$$\sum_{p' \in \mathcal{P}^T \cup \{0\}} z'_{pp'} = 1 \qquad p \in \mathcal{P}^T \quad (3.29)$$

$$\sum_{p' \in \mathcal{P}^T \cup \{0\}} z'_{p'p} = 1 \qquad p \in \mathcal{P}^T. \quad (3.30)$$

Constraints (3.27) -(3.28) guarantee that at most $|\mathcal{C}|$ tours are formed. If the left-hand sides of these constraints are less than the right-hand sides, it means that some chairs are not used. Constraints (3.29) and (3.30), respectively, indicate that patient $p \in \mathcal{P}^T$ has exactly one successor and one predecessor in the tour. It is worth mentioning that, since we will use these constraints within a scheduling model, sub-tours will not happen and therefore, sub-tour elimination constraints could be ignored.

Similarly, we can formulate the assignment of patients to nurses and their sequencing as multi-TSP problem by considering nurses as a second type of salesmen, and the patients with treatment appointments as the cities. We let binary variable $y'_{pp'}$ take 1 if and only if there exists one nurse-tour from patient $p \in \mathcal{P}^T \cup \{0\}$ to the patient $p' \in \mathcal{P}^T \cup \{0\}$. Thus, we can formulate allocation and sequencing of patients with respect to nurses as follows:

$$\sum_{p \in \mathcal{P}^T} y'_{0p} \leq |\mathcal{N}| \quad (3.31)$$

$$\sum_{p \in \mathcal{P}^T} y'_{p0} \leq |\mathcal{N}| \quad (3.32)$$

$$\sum_{p' \in \mathcal{P}^T \cup \{0\}} y'_{pp'} = 1 \qquad p \in \mathcal{P}^T \quad (3.33)$$

$$\sum_{p' \in \mathcal{P}^T \cup \{0\}} y'_{p'p} = 1 \qquad p \in \mathcal{P}^T. \quad (3.34)$$

According to the multi-TSP reformulation, we propose an alternative model for the CCSP as follows:

$$
\begin{aligned}
\text{(M2)} \quad & \text{minimize} \quad \mathop{\mathbb{E}}_{\omega \in \Omega}[Q(sc, z', y', \omega)] && (3.35)\\
& \text{subject to} \;\; (3.2), (3.3), (3.11) - (3.13), (3.27) - (3.34) \\
& \qquad z'_{pp'} \in \{0,1\}, y'_{pp'} \in \{0,1\} \qquad p, p' \in \mathcal{P}^T \cup \{0\} : p \neq p', && (3.36)
\end{aligned}
$$

where

$$
\begin{aligned}
Q(sc, z', y', \omega) = \text{minimize} \;\; & \alpha_1 ot^1_\omega + \alpha_2 ot^2_\omega + \beta_1 \sum_{p \in \mathcal{P}^2} wt_{p\omega} + \beta_2 \sum_{p \in \mathcal{P}^3} wt_{p\omega} && (3.37)\\
\text{subject to} \;\; & (3.18), (3.21) - (3.26) \\
& st_{p'\omega} \geq st_{p\omega} + ST_{p\omega} + TT_{p\omega} - M(1 - z'_{pp'}) \\
& \qquad p, p' \in \mathcal{P}^T : p \neq p' && (3.38)\\
& st_{p'\omega} \geq st_{p\omega} + ST_{p\omega} - M(1 - y'_{pp'}) \\
& \qquad p, p' \in \mathcal{P}^T : p \neq p. && (3.39)
\end{aligned}
$$

Constraints (3.38) and (3.39) are the equivalents of the constraints (3.19) and (3.20), respectively, that compute start time of treatment appointments with respect to the new variables $z'_{pp'}$ and $y'_{pp'}$.

3.5 Solution Methodology

In this section, we propose a sample average approximation (SAA) scheme to solve the CCSP with stochastic treatment times. SAA uses Monte Carlo simulation to iteratively generate a set of random samples and to approximate the expected value of the objective function. For more information on SAA method, we refer readers to [39, 46, 48, 61].

At every iteration of the SAA, we first solve an extensive form of the two-stage formulation with a finite number of scenarios, and then evaluate the expected objective value of the obtained solution by a significantly larger number of scenarios. This evaluation is usually simpler than the previous step as it is solved for a fixed first-stage solution. In this section, we also design an specialized algorithm to efficiently evaluate a given first-stage solution without solving the second-stage model. In the following, we explain the general framework of the proposed SAA and the specialized algorithm.

3.5.1 Sample average approximation

The main challenge in solving the stochastic CCSP is the large amount of scenarios that can happen in reality. Solving the extensive formulation of the proposed stochastic programming models for all the possible scenarios is computationally impossible. To tackle this issue, in a SAA scheme, we generate a random sample of $|N|$ scenarios, $N = \{\omega_1, ..., \omega_{|N|}\}$, and then approximate the expected second-stage objective function

$$\mathop{\mathbb{E}}_{\omega}\left[\alpha_1 ot^1_\omega + \alpha_2 ot^2_\omega + \beta_1 \sum_{p\in\mathcal{P}^2} wt_{p\omega} + \beta_2 \sum_{p\in\mathcal{P}^3} wt_{p\omega}\right],$$

by

$$\frac{1}{|N|}\sum_{n\in N}\left[\alpha_1 ot^1_\omega + \alpha_2 ot^2_\omega + \beta_1 \sum_{p\in\mathcal{P}^2} wt_{p\omega} + \beta_2 \sum_{p\in\mathcal{P}^3} wt_{p\omega}\right].$$

Therefore, we can approximate the M1 by the following SAA problem:

$$\text{minimize } \frac{1}{|N|}\sum_{n\in N}\left[\alpha_1 ot^1_\omega + \alpha_2 ot^2_\omega + \beta_1 \sum_{p\in\mathcal{P}^2} wt_{p\omega} + \beta_2 \sum_{p\in\mathcal{P}^3} wt_{p\omega}\right] \tag{3.40}$$

$$\text{subject to } (3.2) - (3.16), (3.18) - (3.26),$$

where, in constraints (3.18)-(3.26), scenario index $\omega \in \Omega$ is replaced by index of sample scenario $n \in N$. Similarly, we can also approximate the M2 by a SAA problem.

We let $\hat{x}_N$ and $\hat{f}_N$, denote the optimal first-stage solution and the optimal objective value of the SAA problem. According to the [39], $\hat{x}_N$ and $\hat{f}_N$ converge to their true counterparts with probability one, as the sample size $|N|$ increases. However, in the SAA algorithm, instead of solving one large SAA problem, we can generate $|M|$ independent samples of $|N|$ scenarios and solve the smaller SAA problems for each sample. Using the optimal objective values of these $|M|$ samples, i.e. $\hat{f}_N^1, ..., \hat{f}_N^{|M|}$, one can compute statistical lower-bound (LB), upper-bound (UB), and optimality gap of the original problem. Algorithm 2 shows the steps of the SAA algorithm that we have implemented.

3.5.2 Specialized algorithm

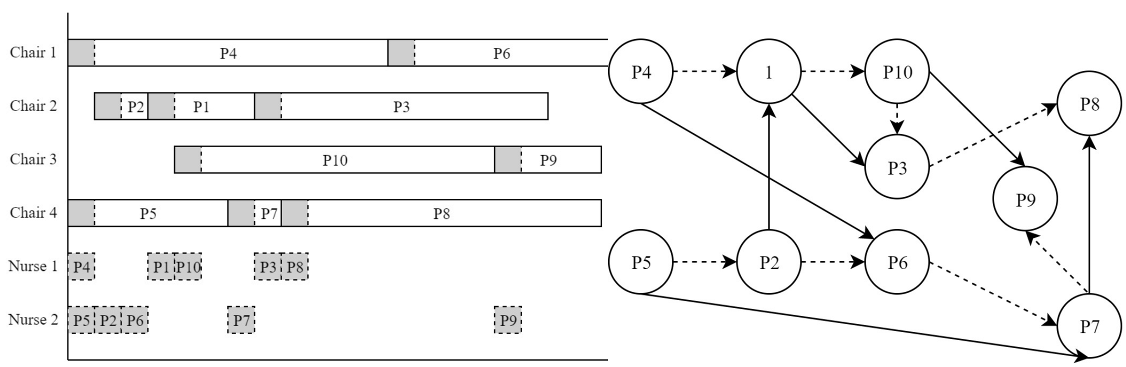
Here, we develop a specialized algorithm to quickly evaluate a given first-stage solution for a large number of scenarios. We show that, for a given first-stage solution, how we can optimally calculate the values of the second-stage variables without solving the second-stage model.

We remind that, in the proposed two-stage stochastic models, we determine the assignment of patients to chairs and nurses and their sequencing in the first stage. Then, in the second stage, we determine the actual start time of the treatment appointments according to the realization of the random parameters. The main idea of the specialized algorithm is that the treatment appointment of a patient can start only when the processing of previous patients assigned to the same chair and the same nurse are completed. To better describe the proposed specialized algorithm, we provide a small example with 10 patients, four chairs and two nurses. Let us suppose that the assignments and sequences are as illustrated in figure 3.1a. Figure 3.1b represents job-on-node presentation of these given assignments and sequences.

Algorithm 2: Sample Average Approximation

Choose algorithm parameters: $|M|, |N|, |T|$;
Set the best known upper-bound to infinity: $BU = \infty$;
for $m = 1$ *to* $|M|$ **do**
  Randomly generate $|N|$ samples: $N = \{\omega_1, ..., \omega_{|N|}\}$;
  Solve the SAA problem considering the generated $|N|$ samples;
  Obtain the optimal value, $\hat{f}_N^m(x)$, and the optimal first-stage solution, $\hat{x}_N^m$;
  Generate $|T|(|T| >> |N|)$ independent random samples:
  $T = \{\omega_1, ..., \omega_{|T|}\}$;
  for $j = 1$ *to* $|T|$ **do**
    Evaluate the first-stage decision, $\hat{x}_N^m$, in scenario ω_j;
    Obtain the optimal value, $\hat{f}^j(\hat{x}_N^m)$;
  end
  Calculate upper-bound estimator as $U_T^m = |T|^{-1} \sum_{j=1}^{|T|} \hat{f}^j(\hat{x}_N^m)$;
  Calculate UB variance estimator as $\sigma^2_{U_T^m} = \frac{1}{|T|(|T|-1)} \sum_{j=1}^{|T|} \left(\hat{f}^j(\hat{x}_N^m) - U_T^m\right)^2$;
  if $U_T^m < BU$ **then**
    $bi = m$;
    $BU = U_T^m$;
    $\sigma^2_{BU} = \sigma^2_{U_T^m}$;
  end
end
Calculate lower-bound estimator as $L_M^N = |M|^{-1} \sum_{m=1}^{|M|} \hat{f}_N^m(x)$;
Calculate LB variance estimator as $\sigma^2_{L_M^N} = \frac{1}{|M|(|M|-1)} \sum_{m=1}^{|M|} \left(\hat{f}_N^m(x) - L_M^N\right)^2$;
Calculate gap estimator as $\text{gap}_{N,M,T} = UB - L_M^N$;
Calculate gap variance estimator as $\sigma^2_{\text{gap}} = \sigma^2_{L_M^N} + \sigma^2_{UB}$;
Calculate confidence interval for the gap as $(\text{gap} - z_{\frac{\alpha}{2}}\sigma_{\text{gap}}, \text{gap} + z_{\frac{\alpha}{2}}\sigma_{\text{gap}})$;
Return $\hat{x}_N^{bi}$ as the best first-stage solution.

In this figure, for a fixed first-stage solution, solid arrows represent the precedence relations on chairs, and dashed arrows denote the precedence relations on nurses. According to this representation, treatment appointments of $P4$ and $P5$ can start at time zero, as they have no predecessors. The dashed arrow from $P5$ to $P2$ shows that $P2$ is served immediately after $P5$ by the same nurse. Therefore, the appointment of $P2$ can start only after the processing of $P5$. After setting up $P2$ by the assigned nurse, all the predecessors of $P6$ are completed, and thus the appointment of $P6$ can start. When the infusion of $P2$ is completed, the corresponding chair is released and $P1$ can immediately start the treatment, supposing that his/her nurse is already released by $P4$. We can further proceed on the presented graph and determine the start time of the nodes for which all the predecessors are completed, until all patients are scheduled. This is the procedure, that our proposed specialized algorithm follows to compute the optimal appointment start times in the second stage. Algorithm 3 summarizes the steps of the proposed algorithm, for a given scenario ω and a given first-stage solution z'_{pp}, y'_{pp}, sc_p, and st^0_p. According to the provided values of z'_{pp} and y'_{pp}, we can easily determine the predecessor and the successor of each patient with respect to the assigned chair and nurse.



(a) Illustration of the assignments and sequences

(b) Job-on-node presentation

Figure 3.1: An example of the assignment of patients to the chairs and nurses for the treatment stage

Algorithm 3: Specialized Algorithm

Initialize second-stage decision values:
$\forall p \in \mathcal{P}^2 : st_{p\omega} = sc_p + CT_p + DT_p,\ \ wt_{pk} = 0;$
$\forall p \in \mathcal{P}^3 : st_{p\omega} = st^0_p,\ \ wt_{pk} = 0;$
$ot^1_\omega = 0,\ \ ot^2_\omega = 0;$
Determine the predecessor and successor of the patients on chairs and nurses:
$\forall p \in \mathcal{P}^T : Chair(p),\ \ Chair\_Predecessor(p),\ \ Chair\_Successor(p);$
$\forall p \in \mathcal{P}^T : Nurse(p),\ \ Nurse\_Predecessor(p),\ \ Nurse\_Successor(p);$
Calculate actual start time for treatment appointments of the patients:
$Scheduled\_patients = 0,\ \ \mathcal{P}^{visited} = \emptyset;$
$\forall c \in \mathcal{C} : Chair\_time(c) = S^C,\ \ \ \forall n \in \mathcal{N} : Nurse\_time(n) = S^C;$
while $Scheduled\_patients < |\mathcal{P}^T|$ **do**
    for $p \in \mathcal{P}^T \setminus \mathcal{P}^{visited}$ **do**
        if $Chair\_Predecessor(p) = 0\ \ \&\ \ Nurse\_Predecessor(p) = 0$ **then**
            $st_{p\omega} = \max\{st_{p\omega}, Chair\_time(Chair(p)), Nurse\_time(Nurse(p))\};$
            $Scheduled\_patients + +,\ \ \mathcal{P}^{visited} = \mathcal{P}^{visited} \cup \{p\};$
            $Chair\_time(Chair(p)) = st_{p\omega} + ST_{p\omega} + TT_{p\omega};$
            $Nurse\_time(Nurse(p)) = st_{p\omega} + ST_{p\omega};$
            if $Chair\_Successor(p) \neq 0$ **then**
                $Chair\_Predecessor(Chair_Successor(p)) = 0;$
            end
            if $Nurse\_Successor(p) \neq 0$ **then**
                $Nurse\_Predecessor(Nurse_Successor(p)) = 0;$
            end
            $break;$
        end
    end
end
Calculate the performance measures:
for $p \in \mathcal{P}^2$ **do**
    $wt_{p\omega} = \max\{0, st_{p\omega} - (sc_p + CT_p + DT_p)\};$
end
for $p \in \mathcal{P}^3$ **do**
    $wt_{p\omega} = \max\{0, st_{p\omega} - st^0_p\};$
end
for $p \in \mathcal{P}^T$ **do**
    $ot^1_\omega = \max\{ot^1_\omega, st_{p\omega} + ST_{p\omega} + TT_{p\omega} - F^C\};$
end
if $ot^1_\omega \leq O^{max}$ **then**
    $ot^2_\omega = 0;$
else
    $ot^2_\omega = ot^1_\omega - O^{max},\ \ \ ot^1_\omega = O^{max};$
end

3.6 Computational Experiments

We provide the results of several computational experiments performed to assess the performance of the proposed formulations and SAA algorithm. We have implemented all models and algorithms in C++ using CPLEX 12.10.0 on an Intel Xeon CPU E5-2687W v3 processor at 3.10 GHz and 750 GB of RAM where the number of threads is set to seven.

We have generated our instances based on the date gathered from Segal Cancer Center (SCC), a major oncology clinic in Canada. SCC has 30 chemotherapy chairs to serve patients and each nurse is responsible for three or four chairs. Some days, the center is fully loaded and needs to use the full capacity of chairs, while some other days the demand is not that high and some chairs can remain unused. To better reflect the actual behaviour of the clinic, we have sampled from different days with different demand levels and generated 10 instances as shown in Table 3.1. In this table, Columns 2 to 7 respectively represent numbers of all patients, patients with consultation, patients with treatment, oncologists, chairs, and nurses in each instance. Columns 8 to 10 , denote the average, minimum and maximum expected treatment time (in hours) over the requested chemotherapy appointments in each instance. We suppose that 30 percent of patients with treatment appointments, need to visit the oncologists before their treatment. We also consider the drug preparation time and the expected setup time to be equal to 30 minutes and 15 minutes, respectively. Moreover, we consider the following weights in the objective function: $\alpha_1 = 8, \alpha_2 = 32, \beta_1 = 18$, and $\beta_2 = 42$.

We assume that, $\overline{ST_p}$ and $\overline{TT_p}$ are the expected setup and treatment times for $p \in \mathcal{P}^T$. To consider the deviation of uncertain parameters from these values, we introduce $\lambda_{p\omega}$ as a random parameter that follows a normal distribution with mean equal to zero and standard deviation of σ , and set $ST_{p\omega}+TT_{p\omega} = (1+\lambda_{p\omega})(\overline{ST_p}+\overline{TT_p})$. To have reasonable uncertain values for $ST_{p\omega}$ and $TT_{p\omega}$, we consider a truncated

Table 3.1: Real-sized instances characteristics

Instance	$\|P\|$	$\|P^C\|$	$\|P^T\|$	$\|O\|$	$\|C\|$	$\|N\|$	$Avg(\overline{TT})$	$Min(\overline{TT})$	$Max(\overline{TT})$
Exp1	117	98	29	5	18	5	2.96	0.5	6.75
Exp2	143	125	27	7	20	5	3.03	0.5	7
Exp3	120	96	35	6	24	6	3.64	1.25	7.25
Exp4	135	110	37	8	21	6	2.88	0.5	7.52
Exp5	150	116	50	7	28	8	2.75	0.5	6
Exp6	183	149	50	9	30	8	3.10	0.5	7.25
Exp7	98	63	51	4	30	8	2.55	0.5	6
Exp8	60	31	43	4	22	6	2.69	0.5	7.25
Exp9	78	51	40	4	19	5	2.57	0.5	7.25
Exp10	146	111	52	7	30	8	2.88	0.5	7

version of $\lambda_{p\omega}$ where it belongs to [-0.5, 0.5]. According to the considered distribution for the random deviations, we observe that for patient p, the mean for the random service time is equal to the $\overline{ST_p} + \overline{TT_p}$, and the standard deviation is equal to the $\sigma(\overline{ST_p} + \overline{TT_p})$, and thus the coefficient of variance (COV) is equal to σ.

3.6.1 Comparison of the formulations

In section 3.4, we proposed two formulations for the stochastic CCSP. In order to evaluate the performance of the proposed models, we solved deterministic and stochastic versions of the CCSP using M1 and M2, with a time limit of one hour. The results are reported in Table 3.2. According to this table, we observe that M2 outperforms M1 significantly, in terms of the CPU time and the quality of obtained solutions within the given time limit. Using M2 instead of M1, the CPU time can decrease on average by 98% in the deterministic problem and 87%in the stochastic problem. Furthermore, the number of variables decreases on average by about 91% in both versions of the problem, when we use M2 instead of M1. Number of the constraints can also be reduced on average by 93% in deterministic problem and by 69% in the stochastic problem. Therefore, for the rest of the experiments, we use M2 to apply SAA algorithm for stochastic problems.

Table 3.2: Comparison of M1 and M2

Instance		Time (sec)		Gap (%)		Nodes (#)		Variables (#)		Constraints (#)	
		M1	*M2*	*M1*	*M2*	*M1*	*M2*	*M1*	*M2*	*M1*	*M2*
$\lvert\Omega\rvert = 1$	Exp1	15	2	0.00	0.00	1,299	0	23,450	3,904	32,721	4,798
	Exp2	12	1	0.00	0.00	1,301	50	22,739	4,001	31,260	5,020
	Exp3	40	1	0.00	0.00	1,011	0	42,606	4,386	58,566	5,125
	Exp4	44	2	0.00	0.00	1,563	1,771	42,566	4,752	59,335	5,504
	Exp5	460	4	0.00	0.00	1,856	0	98,785	7,335	136,677	8,206
	Exp6	792	12	0.00	0.00	4,824	3,057	108,086	8,086	148,779	9,283
	Exp7	880	6	0.00	0.00	4,817	1,773	110,299	6,463	152,067	6,874
	Exp8	164	3	0.00	0.00	1,635	1,762	56,671	4,211	80,035	4,316
	Exp9	114	2	0.00	0.00	1,813	27	42,108	4,148	60,484	4,448
	Exp10	555	14	0.00	0.00	1,766	3,381	115,482	7,738	159,613	8,609
Average		308	5	0.00	0.00	2,189	1,182	66,279	5,502	91,954	6,218
$\lvert\Omega\rvert = 10$	Exp1	283	70	0.00	0.00	16,517	28,405	23,991	4,445	48,120	20,197
	Exp2	117	13	0.00	0.00	3,520	1,915	23,244	4,506	44,625	18,385
	Exp3	649	80	0.00	0.00	26,130	24,597	43,255	5,035	80,931	27,490
	Exp4	501	119	0.00	0.00	6,590	26,667	43,251	5,437	84,310	30,479
	Exp5	time	518	100.00	0.00	20,232	85,922	99,704	8,254	182,127	53,656
	Exp6	time	464	87.26	0.00	17,136	33,788	109,005	9,005	194,229	54,733
	Exp7	3,558	342	0.00	0.00	18,460	59,254	111,236	7,400	199,344	54,151
	Exp8	3,107	351	0.00	0.00	32,763	30,463	57,464	5,004	113,704	37,985
	Exp9	1,306	179	0.00	0.00	36,918	46,077	42,847	4,887	89,644	33,608
	Exp10	time	576	93.39	0.00	9,135	108,741	116,437	8,693	208,753	57,749
Average		2,033	271	28.07	0.00	18,740	44,583	67,043	6,267	124,579	38,843

3.6.2 Practical convergence of SAA algorithm

In this section, we analyze the convergence of the implemented SAA algorithm. We remind that, in the SAA method, we generate $|M|$ independent samples of $|N|$ scenarios. Selecting larger values of $|N|$ increases the probability of obtaining a more accurate estimate on the true objective value. However, solving the SAA problem with larger number of instances is computationally difficult. Therefore, it might be better to select a smaller value for $|N|$ and instead increase the value for $|M|$. The goal of this part of the experiments is to find proper values for $|M|$ and $|N|$ in a way that estimated optimality gap and estimated standard deviation of the gap are sufficiently small and the required CPU time is reasonable.

In order to select proper values of $|M|$ and $|N|$, we conduct several experiments with different combinations of $|M| \in \{5, 10, 20, 30, 40, 60, 80\}$ and $|N| \in \{5, 10, 20, 30, 50, 70\}$. We also consider the evaluation sample size (i.e. $|T|$) to be equal to 10000. To perform this analysis, we selected instance Exp4 and set the coefficient of variance to 0.5. According to the previous experiments, Exp4 is not too easy and it is not also too difficult and time consuming to solve. Therefore, it is a good candidate for the analysis. Figure 3.2 plots the estimated optimality gap for different values of $|M|$ and $|N|$.

In Figure 3.2, we observe that for small values of $|N|$ the estimated optimality gap is very large even when the value of $|M|$ increases. However, for larger values of $|N|$ (i.e., 50 and 70), even small values of $|M|$ can provide high quality solutions.

Figure 3.3 plots the estimated standard deviation for the optimality gap with different values of $|M|$ and $|N|$. As expected, the estimated standard deviation of the gap decreases as the sample sizes $|M|$ and $|N|$ increases. Furthermore, for larger values of $|N|$, we observe that even by solving small number of SAA problems, the estimated standard deviation is sufficiently small.

Figure 3.4 plots the required CPU time for the SAA algorithm with different

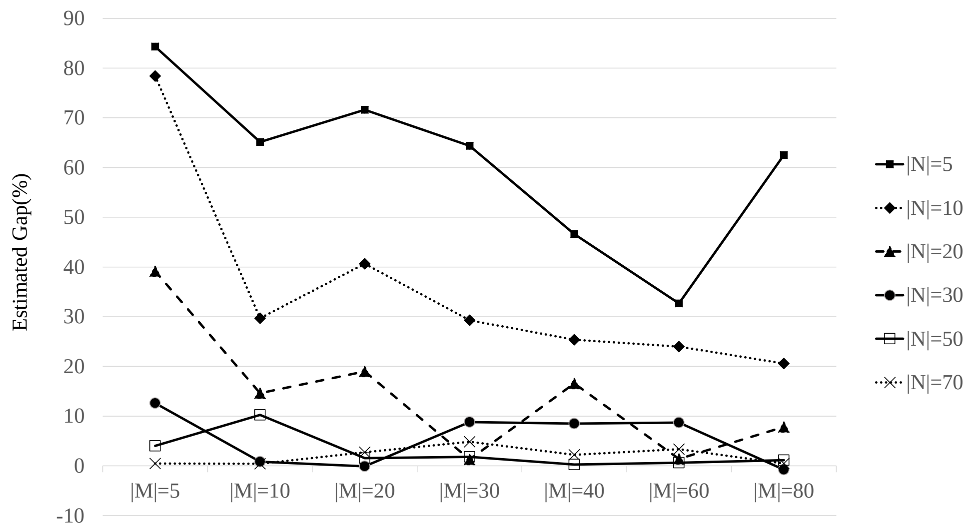


Figure 3.2: Estimated optimality gap for different values of $|M|$ and $|N|$

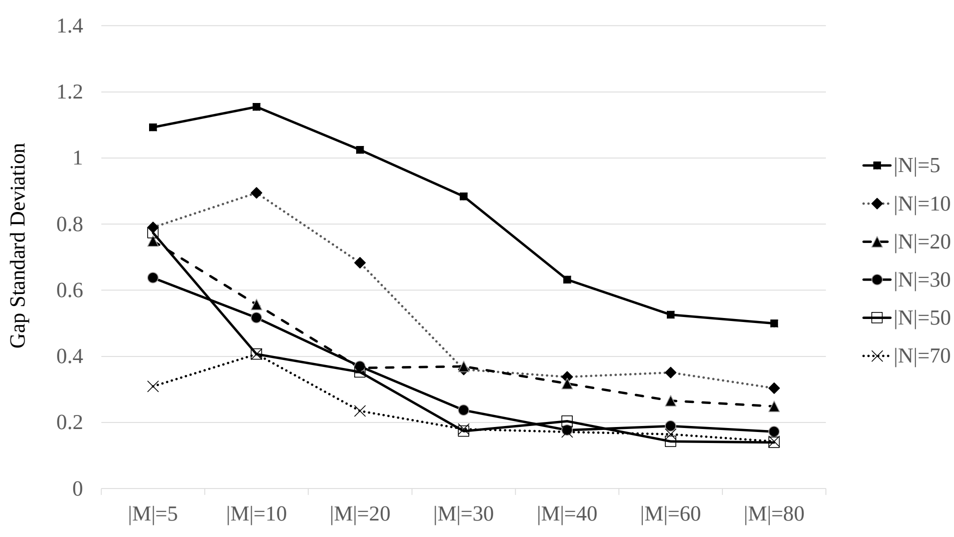


Figure 3.3: Standard deviation of the gap for different values of $|M|$ and $|N|$

values of $|M|$ and $|N|$. We observe that CPU time increases as the values of $|M|$ and $|N|$ increase. However, according to the considered random sample, we may observe some exceptions. The results of the performed experiments indicate that it is better to choose larger values for $|N|$ to estimate the true objective value of the stochastic CCSP more accurate. Furthermore, when we use larger values of $|N|$, we can obtain high quality solutions even by solving small number of SAA problems. Therefore, for the rest of the experiments we use sample sizes $|M| = 30$ and $|N| = 50$.

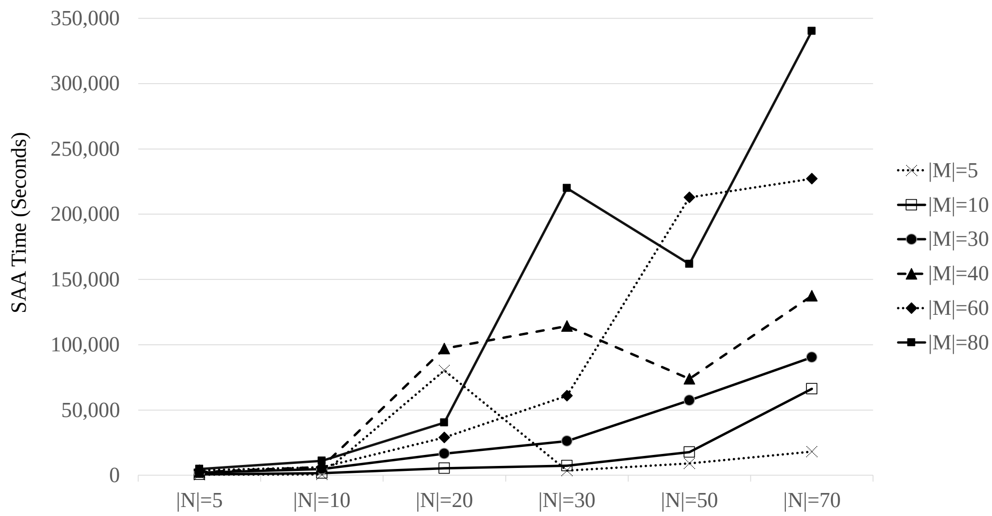


Figure 3.4: Total CPU time for different values of $|M|$ and $|N|$

3.6.3 Comparison of the SAA algorithm with the expected value problem

Now we compare the performance of the proposed SAA algorithm with the solutions obtained by the expected value problem (EVP). For this comparison we consider coefficient of variance to be 0.5. The results are reported in Table 3.3. In this table, first column represents the names of instances. In the second column, statistical lower-bounds obtained by the SAA are provided. Next two columns denote the estimated objective value obtained by the EVP and SAA. CPU time for the EVP and the SAA algorithm are presented in columns five and six. Seventh column represents the value of stochastic solution (VSS) that is computed by $UB_{EVP} - UB_{SAA}$. Last column denotes the percentage of decrease in the UB when we solve the stochastic problem by SAA instead of the EVP, that is calculated by $100(VSS)/UB_{EVP}$.

Table 3.3: Comparison of the SAA and EVP

Instance	LB	UB		Time		VSS	VSS(%)
	SAA	*EVP*	*SAA*	*EVP*	*SAA*		
Exp1	0.00	178.12	0.03	240	20,004	178.10	99.98
Exp2	0.00	152.47	0.05	73	4,761	152.42	99.97
Exp3	4.94	418.54	4.95	256	31,226	413.59	98.82
Exp4	5.53	153.00	5.63	354	57,403	147.37	96.32
Exp5	0.00	461.83	0.15	892	278,480	461.68	99.97
Exp6	6.01	333.58	6.09	1,067	152,517	327.49	98.17
Exp7	1.04	288.17	1.07	472	349,856	287.10	99.63
Exp8	9.03	229.27	9.39	371	531,788	219.88	95.90
Exp9	0.55	189.64	0.68	247	461,269	188.96	99.64
Exp10	0.12	437.39	0.27	764	535,162	437.12	99.94

According to this table, the objective value for the solution obtained by SAA algorithm is significantly lower than the objective value for the EVP solution, and the value of the stochastic solution is significant. The results of this table emphasize the importance of incorporating uncertainty in the model for obtaining high quality solutions.

3.6.4 Effects of different degrees of uncertainty on the quality of the solutions

In this section we examine the impact of different degrees of the uncertainty on the quality of solutions. We use the coefficient of variance as a control parameter to impose different degrees of uncertainty in the stochastic CCSP. We considered five degrees of uncertainty, and compared the solution obtained by the proposed SAA algorithm with the solution of the expected value problem for the first four instances. The results are reported in Table 3.4.

Table 3.4: Comparison of the SAA algorithm and the EVP with different degrees of uncertainty

COV	Instance	LB	UB		Time		VSS	VSS(%)
		SAA	*EVP*	*SAA*	*EVP*	*SAA*		
0.1	Exp1	0.00	53.97	0.13	239	16,854	53.84	99.75
	Exp2	0.00	42.67	0.14	72	3,903	42.53	99.66
	Exp3	0.19	132.79	0.49	261	27,851	132.30	99.63
	Exp4	1.80	40.07	1.93	355	39,625	38.15	95.19
0.3	Exp1	0.00	151.30	0.07	238	20,848	151.23	99.96
	Exp2	0.00	128.58	0.10	73	4,923	128.48	99.92
	Exp3	3.69	357.80	3.93	260	40,803	353.87	98.90
	Exp4	4.55	127.02	4.77	357	52,696	122.25	96.24
0.5	Exp1	0.00	178.12	0.03	240	20,004	178.10	99.98
	Exp2	0.00	152.47	0.05	73	4,761	152.42	99.97
	Exp3	4.94	418.54	4.95	256	31,226	413.59	98.82
	Exp4	5.53	153.00	5.63	354	57,403	147.37	96.32
0.7	Exp1	0.00	186.54	0.04	237	25,284	186.50	99.98
	Exp2	0.00	159.27	0.03	74	5,367	159.23	99.98
	Exp3	4.96	437.08	5.21	259	43,383	431.87	98.81
	Exp4	5.94	161.19	6.02	356	55,645	155.17	96.27
1	Exp1	0.00	191.52	0.03	234	101,664	191.49	99.98
	Exp2	0.00	163.16	0.05	77	5,711	163.11	99.97
	Exp3	5.34	445.35	5.49	256	116,253	439.86	98.77
	Exp4	6.08	166.61	6.06	353	62,303	160.55	96.36

According to Table 3.4, we observe that the statistical lower-bound and upper-bound obtained by the SAA algorithm usually increase when the coefficient of variance increases and higher degrees of uncertainty are imposed. This increase is most likely to happen for the instances that are more affected by the uncertainty and have positive values of the lower-bound. Furthermore, in all instances, we observe that the UB associated with the EVP and the value of stochastic solution increase as coefficient of variance increases. Moreover, the time needed to solve the problem increases as we introduce higher degrees of uncertainty in the stochastic CCSP, which shows that the complexity of the problem increases as we consider higher coefficients of variance.

3.7 Conclusion

In this paper we studied an integrated daily consultation and chemotherapy scheduling problem considering stochastic treatment times and different patient types. We proposed a two-stage stochastic programming model to formulate the problem. We also provided an alternative formulation which partially models the problem as Multi-TSP. The results of the computational experiments reveal that Multi-TSP based formulations outperforms the other formulation significantly. We also presented a SAA algorithm to solve the stochastic CCSP. A specialized algorithm was also designed to quickly evaluate a given first-stage solution for a large number of scenarios. We compared the quality of the solutions obtained by the SAA algorithm with the solutions of the expected value problem. Using the SAA algorithm, we were able to reduce the expected objective value by at least 95 percent in all the experiments. We also studied the impact of the uncertainty degree on the value of the stochastic solutions. We observed that by increasing the coefficient of variance and imposing higher degrees of uncertainty in to the problem, value of stochastic solution increases. Designing decomposition-based algorithms to solve the SAA problems more efficiently is an interesting direction for future researches. Extending the model to also incorporate treatment cancellations is also suggested.

Chapter 4

Formulations and Algorithms for Cross-dock Scheduling Problems with Handling Times

The content of this chapter is submitted as a manuscript for publication to the journal *Transportation Research Part B: Methodological* in August 2020 [30].

Abstract

We study a cross-dock scheduling problem with truckload and door-dependent handling times. It considers the simultaneous assignment and scheduling of incoming and outgoing trucks to inbound and outbound doors. Handling times include unloading, transfer and loading times of products. We study two variants of the problem: a general case with door-dependent unit-load transfer times, and a particular case with constant unit-load transfer times. For the latter case, we propose several families of valid inequalities that are effectively exploited within an exact branch-and-cut algorithm. We also present constraint programming formulations for both variants. Finally, we develop an approximate algorithm for the door-dependent case that uses

the solution of the constant handling time case to obtain a valid lower bound and an initial feasible solution. It then applies an iterated local search matheuristic to obtain improved solutions and to provide an optimality gap. Extensive computational experiments on benchmark instances confirm the efficiency of the proposed formulations and solution algorithms.

4.1 Introduction

Cross-docking is a logistic strategy for managing effectively and efficiently distribution activities in the supply chain. It facilitates the rapid movement of products between multiple origins and destinations with minimal requirements in terms of storage area and storage time in between. Cross-docking has been used by several companies in manufacturing, retailing, and shipping service providers, among others, to improve the efficiency of their logistic operations by reducing costs and accelerating the distribution process. In a cross-dock facility, incoming trucks bringing different types of products from several origins are unloaded at inbound doors. Arriving products are immediately sorted and consolidated based on their destinations, and then transferred to outbound doors to be loaded into outgoing trucks departing for their destinations. Optimizing the overall efficiency of cross-dock terminals has attracted significant attention from researchers and practitioners. In recent years, greater emphasize on reducing or eliminating inventory related costs using just-in-time systems has further increased the popularity of cross-docking. There exist several excellent reviews and surveys on cross-docking such as [10], [12], [40], and [64].

According to Buijs et al. [12], there are two general classes of cross-docking problems: local cross-dock management problems and cross-docking network management problems. Each of these two classes consists of three sub-classes: design, planning and scheduling, which are associated with strategic, tactical and operational decisions, respectively. The focus of the current study is on the integration of the operational decisions of two important interrelated classes of local cross-dock management problems: *cross-dock door assignment problems* (CDAPs) and *truck scheduling problems* (TSPs). In CDAPs, each incoming and outgoing truck must be assigned to an inbound and an outbound door, respectively, such that the total cost of transferring products within the cross-dock is minimized. In TSPs, the sequence of trucks, as well as the processing start time of each truck (i.e., unloading start time for incoming trucks and loading start time for outgoing trucks) should be determined in a way that the total time needed for completing all activities (makespan) or delays of outgoing trucks (tardiness) is minimized. Dock-door assignment and truck scheduling decisions are intertwined and both are needed to achieve the best performance of a cross-docking facility. The integration of these two classes of decisions gives rise to the so-called *cross-dock scheduling problems* (CDSPs). Although such integration seems to be the best way of managing a cross-dock, most studies in the literature have focused on CDAP or TSP independently, or have suggested sequential approaches to solve them [see, 55].

Sayed et al. [59] introduced a general class of CDSPs with truckload and door-dependent handling times. In this problem, denoted as the *cross-dock scheduling problem with handling times* (CDSPHT), incoming trucks carrying different products from different origins are assigned to inbound doors. Products received from each incoming truck are then unloaded, sorted and labeled according to their destination. The time required for each incoming truck to be processed depends on the truckload (i.e., amount and type of products) and the employees and equipment assigned to the assigned inbound door. When the processing of each incoming truck is completed, products are transferred to the outbound doors where their associated outgoing trucks are assigned to. Transfer times are considered to depend on the amount of products to be moved from each incoming truck to their associated outgoing truck. It may also depend on the distance between the assigned inbound and outbound doors. When all the required products of a given outgoing truck are transferred to the associated outbound door, loading of the outgoing truck starts and it continues until all products are fully loaded to the truck. Similar to incoming trucks, loading time of each outgoing truck also depends on the truckload and the employees and equipment assigned to the associated outbound door. The goal of the CDSPHT is to simultaneously assign incoming trucks to inbound doors and outgoing trucks to outbound doors and to determine the arrival time of trucks to doors such that the total time required to process all trucks (i.e., makespan) is minimized. We assume that the destination (i.e., outgoing truck) of each product arriving from each incoming truck is known in advance (i.e., pre-distribution cross-dock setting). Furthermore, each door of the cross-dock facility serves as either inbound or outbound door (i.e., exclusive door environment), which is also determined in advance. An important feature of the CDSPHT is that it incorporates handling times (i.e., loading, unloading and transfer times). We consider loading and unloading times to depend on the truckload and the assigned door, and transfer times to depend on the amount of product to be moved between doors.

In this paper we focus on the modeling and methodological challenges faced when solving CDSPHTs. We study two variants of the CDSPHT. In the first and most general variant, transfer times depend not only on the amount of product but also on the distance between doors and the used equipment at these doors. This dependency can be modeled by considering different unit-load transfer times for each inbound-outbound door pair. We refer to this general problem as the CDSPHT-G. In the second variant, we study a particular case in which unit-load transfer times are constant for all inbound-outbound door pairs. We refer to it as the CDSPHT-S.

The main contributions of this paper are the following. First, we introduce constraint programming (CP) formulations for both CDSPHT-G and CDSPHT-S and compare their performance with mixed integer programming (MIP) formulations previously introduced in the literature. Second, we propose several classes of valid inequalities for CDSPHT-S to strengthen the MIP formulations introduced in [59]. Third, we develop an exact branch-and-cut (BC) algorithm for the CDSPHT-S in which theses inequalities are separated at some nodes of the enumeration tree. Fourth, we present a matheuristic for the CDSPHT-G, that decomposes the problem into two independent parallel machine scheduling problems with release dates. We enhance the performance of this decomposition procedure by embedding it into an iterated local search (ILS) framework. Fifth, we develop an approximate algorithm that exploits the information generated by the solution of the CDSPHT-S with the BC algorithm and the ILS matheuristic to compute lower and upper bounds, and thus an optimality gap for the general CDSPHT-G.

The remainder of this paper is structured as follows. Section 4.2 reviews the most recent and related literature in cross-docking. In Section 4.3, the CDSPHT-G and CDSPHT-S are formally described and two MIP formulations are provided. Section 4.4 introduces CP formulations for both variants. In Section 4.5, several families of valid inequalities are presented for the particular case CDSPHT-S as well as implementation details of our BC algorithm. Sections 4.6 and 4.7 present the matheuristic and approximate algorithm, respectively. Section 4.8 reports the results of extensive computational experiments performed to evaluate the proposed formulations and solution algorithms. Finally, Section 4.9 concludes the paper and provides future research directions.

4.2 Literature Review

The increased popularity and use of cross-docking have encouraged the study of different modeling and solution approaches for complex decision problems arising at cross-dock facilities. Boysen and Fliedner [10] provide a classification of deterministic TSPs and review the related literature based on this classification. Van Belle et al. [67] describe different characteristics of cross-docking systems and classify related studies according to such characteristics. Buijs et al. [12] classify and review the cross-docking literature based on different decisions to be made within the cross-docking context. Ladier and Alpan [40] compare cross-docking studies with the industry practices and analyze the gaps between them. Theophilus et al. [64] provide an updated state-of-the-art review on truck scheduling problems at cross-docking terminals.

There exist different scheduling problems arising in a cross-dock scheduling context. Some of the existing work focuses on CDAPs [24, 28, 50, 51, 68], while some others investigate TSPs [3, 8, 11, 16, 19, 21]. There are also some papers that have studied both problems simultaneously [17, 33, 34, 55, 59, 60, 69]. Although the number of studies on CDSPs has increased in recent years, few papers have incorporated handling times into their models [33, 55, 59, 69]. In the following, the most recent and related papers to the current study are reviewed in more detail.

Wisittipanich and Hengmeechai [69] present an MIP formulation for a CDSP with a post-distribution setting. The authors develop a modified particle swarm optimization algorithm to solve the proposed problem, in which transfer times are assumed to only depend on the distance between pairs of inbound-outbound doors and not on the amount of products to be transferred. It is also assumed that loading and unloading times only depend on the amount of products to be processed and they do not depend on the assigned door. Heidari et al. [33] propose an MIP formulation for a CDSP with arrival times and pre-distribution setting that aims to minimize trucks waiting times and processing costs. In order to incorporate arrival uncertainty into the model, the authors present a bi-objective bi-level optimization approach. It is assumed that loading and unloading times of trucks depend on both the truck and assigned door. Moreover, for each incoming truck a sorting time for the unloaded shipment is taken into account, which only depends on the truck and not the door. However, the time needed for transferring products between incoming and outgoing trucks is not considered. Bodnar et al. [8] study a TSP with truck arrival and departure times, a pre-distribution setting and a mixed service mode door environment considering an uncapacitated temporary storage area. The authors propose an MIP formulation and an adaptive large neighborhood search algorithm with the objective of minimizing cost of temporary storage and tardiness cost of outgoing trucks. The proposed problem and algorithm have been extended by Rijal et al. [55] to include dock-door assignments along with truck scheduling decisions and to also minimize the cost of transferring products directly from incoming to the outgoing trucks. Although in the proposed problem travel distances between inbound and outbound doors have been considered in the objective function, it is assumed that the time required for transferring products is the same for all products regardless of the associated truck or door and is equal to one time period, and thus does not need to be considered in the model. Furthermore, the authors also assume that loading and unloading processes can also be done in one unit of time and thus processing times are also disregarded.

Ye et al. [70] propose an MIP formulation and a particle swarm optimization algorithm for a CDSP with the requirement of unloading and loading products in a given order and the objective of makespan minimization considering a pre-distribution setting. It is assumed that loading and unloading times only depend on the amount of product to be processed and not on the assigned door. Moreover, the time needed to transfer products from incoming to outgoing trucks is assumed to depend only on the associated assigned doors and to be independent from the amount and type of products to be transferred. Fonseca et al. [21] model a TSP as a two-machine flow shop scheduling problem with precedence constraints that aims to minimize the makespan. It is assumed that loading and unloading times only depend on the trucks and not on the doors. Transfer times between incoming and outgoing trucks are also neglected. The authors also generalize the model to the parallel-dock case in which it is assumed that all loading and unloading times are equal to one unit of time and transferring times are negligible. To solve the problem a hybrid Lagrangian metaheuristic is developed. Gaudioso et al. [23] study a CDSP with a post-distribution setting. The authors formulate the problem with an MIP formulation with the objective of makespan minimization and develop a Lagrangian heuristic algorithm to solve it.

Finally, we recall that Sayed et al. [59] introduce the CDSPHT-G. The authors propose two time-index MIP formulations and two metaheuristic algorithms that can obtain good quality solutions for the problem. In the next section, we present the two formulations given that these form the basis for our BC algorithm for the CDSPHT-S. In Section 4.8, we compare these formulations with our CP formulations. We also compare the two metaheuristics presented in [59] with our proposed solution algorithms for the CDSPHT-G.

4.3 Problem Definition and MIP Formulations

Let $\mathcal{M}$, $\mathcal{N}$, $\mathcal{I}$, and $\mathcal{J}$ denote the set of incoming trucks, outgoing trucks, inbound doors, and outbound doors, respectively. Let $\mathcal{T}$ be a set of time slots in the planning horizon, and $\mathcal{K}$ be the set of products. For each product $k \in \mathcal{K}$, let $o(k) \in \mathcal{M}$ denote its origin truck, $d(k) \in \mathcal{N}$ its destination truck, and w_k the amount of product to be transferred from $o(k)$ to $d(k)$. We assume that the material handling equipment used to transfer products from the origin truck to the destination truck can carry one unit-load each time it transfers the product from an inbound to an outbound door. We define as d_{ij} the unit-load transfer time from inbound door i to outbound door j, which depend on the distance and equipment used between doors. Therefore, $w_k d_{ij}$ denotes the total time required to transfer all product $k \in \mathcal{K}$ from incoming truck

$o(k) \in \mathcal{M}$ using inbound door $i \in \mathcal{I}$ to outgoing truck $d(k) \in \mathcal{N}$ using outbound door $j \in J$. We define p^u_{mi} as the time required to unload incoming truck $m \in \mathcal{M}$ at inbound door $i \in \mathcal{I}$, and p^l_{nj} as the loading time of outgoing truck $n \in \mathcal{N}$ at outbound door $j \in \mathcal{J}$. Unloading and loading times depend on the skill of employee and the material handling equipment assigned to each door and thus, can differ from one door to another. Cross-dock terminals, relying on third party contractors to hire temporarily workforce when needed, frequently employ a variety of loaders with different skill levels (e.g., from beginners to well trained loaders). According to their level, loaders may use simple material handling equipment such as manual carts, or more advance equipment such as forklifts which in turn, affect the unloading, transfer and loading times [see, 51].

We assume that all trucks are available at the beginning of the planning horizon and preemption is not allowed. Each door can only process one truck at a time and the time horizon is enough to process all trucks. Moreover, it is assumed that sorting and transferring the products from an incoming truck to one or more outgoing trucks can only take place after all products are unloaded. Similarly, an outgoing truck cannot be assigned to an outbound door until all products are transferred at the door.

To formulate both CDSPHT-G and CDSPHT-S we define several sets of decision variables. Let x_{mit} be a binary variable that takes value one if and only if incoming truck $m \in \mathcal{M}$ is assigned to inbound door $i \in \mathcal{I}$, and is scheduled to start the unloading process at time $t \in \mathcal{T}$. Similarly, binary variable y_{njt} is equal to one if and only if outgoing truck $n \in \mathcal{N}$ is assigned to outbound door $j \in \mathcal{J}$, and is scheduled to start the loading process at time $t \in \mathcal{T}$. We also define z_{kij} as a binary variable that takes one if and only if product $k \in \mathcal{K}$ is transferred from inbound door $i \in \mathcal{I}$ to outbound door $j \in \mathcal{J}$. Finally, we define C_{max} as a continuous variable that captures the value of makespan. Using these three sets of variables, the CDSPHT-G can be formulated as [59]:

$$
\begin{aligned}
\text{(F1-G)} \quad & \text{minimize } C_{max} \\
& \text{subject to } \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} \left(t + p^l_{nj}\right) y_{njt} \leq C_{max} && n \in \mathcal{N} && (4.1) \\
& \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} x_{mit} = 1 && m \in \mathcal{M} && (4.2) \\
& \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} y_{njt} = 1 && n \in \mathcal{N} && (4.3) \\
& \sum_{m \in \mathcal{M}} \sum_{r=max\{0, t-p^u_{mi}+1\}}^{t} x_{mir} \leq 1 && t \in \mathcal{T}, i \in \mathcal{I} && (4.4) \\
& \sum_{n \in \mathcal{N}} \sum_{r=max\left\{0, t-p^l_{nj}+1\right\}}^{t} y_{njr} \leq 1 && t \in \mathcal{T}, j \in \mathcal{J} && (4.5) \\
& \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} t y_{d(k)jt} - \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} t x_{o(k)it} \geq \\
& \qquad \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \left(p^u_{o(k)i} + w_k d_{ij}\right) z_{kij} && k \in \mathcal{K} && (4.6) \\
& \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} z_{kij} = 1 && k \in \mathcal{K} && (4.7) \\
& \sum_{j \in \mathcal{J}} z_{kij} = \sum_{t \in \mathcal{T}} x_{o(k)it} && k \in \mathcal{K}, i \in \mathcal{I} && (4.8) \\
& \sum_{i \in \mathcal{I}} z_{kij} = \sum_{t \in \mathcal{T}} y_{d(k)jt} && k \in \mathcal{K}, j \in \mathcal{J} && (4.9) \\
& x_{mit} \in \{0, 1\} && m \in \mathcal{M}, i \in \mathcal{I}, t \in \mathcal{T} && (4.10) \\
& y_{njt} \in \{0, 1\} && n \in \mathcal{N}, j \in \mathcal{J}, t \in \mathcal{T} && (4.11) \\
& z_{kij} \geq 0 && k \in \mathcal{K}, i \in \mathcal{I}, j \in \mathcal{J}. && (4.12)
\end{aligned}
$$

The objective function minimizes the makespan. Constraints (4.1) capture the value of makespan. Constraints (4.2) and (4.3) assign each incoming and outgoing truck to a single door and time slot, respectively. Constraints (4.4) and (4.5) ensure that each inbound and outbound door serves at most one truck at a time. Constraints (4.6) guarantee that there is enough time for unloading and transferring all products required at each outgoing truck before the loading process starts. Constraints (4.7)–(4.9) assign a proper value to the variable z_{kij} to represent the selected route of product k. Constraints (4.10)–(4.12) define the integrality and non-negativity conditions of the decision variables.

Sayed et al. [59] provides an alternative formulation to CDSPHT-G that does not require the use of the routing variables z_{kij} to account for the transfer times in order to determine the time at which products arrive at outbound doors. Let X' denote the set of feasible solutions of CDSPHT-G. The following inequalities are valid for X':

$$y_{d(k)jt} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{kijt}} x_{o(k)is} \qquad k \in \mathcal{K}, j \in \mathcal{J}, t \in \mathcal{T}, \quad (4.13)$$

where $\mathcal{S}_{kijt} = \left\{ s \in \mathcal{T} | 0 \leq s \leq t - \left(p^u_{o(k)i} + w_k d_{ij} \right) \right\}$ denotes the set of time periods that allow the completion of the unloading and transfer operations needed before scheduling truck $d(k)$ at time t. These constraints state that for each product $k \in \mathcal{K}$, if outgoing truck $d(k)$ is scheduled to start loading at time t, the associated incoming truck $o(k)$ can be scheduled at time slots that provide enough time for unloading and transferring processes before time t. These inequalities model the same conditions as constraints (4.6), without the use of the z_{kij} variables. The CDSPHT-G can thus be stated as follows [59]:

$$\text{(F2-G)} \quad \text{minimize } C_{max}$$
$$\text{subject to } (4.1) - (4.5), (4.10), (4.11), (4.13).$$

Formulation F2-G has fewer variables than F1-G but a larger number of constraints. As shown in [59], none of these formulations dominate the other in terms of the quality of the linear programming relaxation bounds.

In the special case of CDSPHT-S, in which it is assumed that $d_{ij} = \overline{d}$ for all pairs $i \in \mathcal{I}, j \in \mathcal{J}$, we can eliminate from F1-G the routing variables z_{kij} given that the transfer time does no longer depend on the selected inbound and outbound doors. Therefore, the CDSPHT-S can be formulated as

$$\text{(F1-S)} \quad \text{minimize } C_{max}$$

$$\text{subject to } (4.1) - (4.5), (4.10), (4.11)$$

$$\sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} t y_{d(k)jt} \geq \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} \left(t + p^u_{o(k)i} + w_k \overline{d}\right) x_{o(k)it}$$

$$k \in \mathcal{K}. \quad (4.14)$$

Similar to (4.6), constraints (4.14) guarantee that there is enough time for unloading and transferring all the required products to each outgoing truck before the loading process can be scheduled to start.

We can also adapt F2-G to obtain an alternative formulation for CDSPHT-S as:

$$\text{(F2-S)} \quad \text{minimize } C_{max}$$

$$\text{subject to } (4.1) - (4.5), (4.10), (4.11)$$

$$y_{d(k)jt} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{kit}} x_{o(k)is} \qquad k \in \mathcal{K}, j \in \mathcal{J}, t \in \mathcal{T}, \quad (4.15)$$

where $\mathcal{S}_{kit} = \left\{s \in \mathcal{T} | 0 \leq s \leq t - \left(p^u_{o(k)i} + w_k \overline{d}\right)\right\}$. As we will see in Section 4.5, constraints (4.15) are rather weak as they are dominated by several classes of valid inequalities. Let X denote the set of feasible solutions of CDSPHT-S.

4.4 Constraint Programming Formulations

In this section, we present CP formulations for both CDSPHT-G and CDSPHT-S. We first provide a brief introduction to CP including some of the constraints commonly used to formulate scheduling problems. We then describe the proposed CP formulations.

CP is a technique for modeling and solving difficult combinatorial optimization problems. It is mainly used for finding feasible solutions or to prove infeasibility, but it can also be used to find optimal solutions with respect to a given objective function. Similar to mathematical programming formulations, CP formulations use set(s) of decision variables and constraints to formulate an optimization problem. However, in CP there exist specialized variables and constraints that facilitate model building, particularly for scheduling problems [see, 4, 5, for additional information].

Interval and *sequence* variables are two special types of decision variables in CP that can be used to formulate scheduling problems. An interval variable corresponds to a time interval during which a job is being processed. For each job $m \in M$, we define the interval variable x_m which is characterized by its start time, i.e., $Start(x_m)$, end time, i.e., $End(x_m)$, processing time, and presence status, i.e., $Presence(x_m)$. When solving a CP formulation, $Start(x_m)$, $End(x_m)$ and $Presence(x_m)$ are endogenous decisions. Moreover, interval variables can be defined as optional or present. If x_m is defined as optional, the formulation determines if the variable is present in the solution, i.e., $Presence(x_m) = 1$ or not, i.e., $Presence(x_m) = 0$. If x_m is defined as present, all feasible solutions must satisfy $Presence(x_m) = 1$. Defining optional interval variables is useful when a job has to be processed by exactly one of several available machines and the assignment of jobs to machines are endogenous decisions. For example, for each job $m \in M$ and machine $i \in I$, we define x'_{mi} as an optional interval variable that represents the processing interval of job m on machine i. Furthermore, x_m can be defined as a present interval variable denoting the actual processing interval of job m.

The specialized constraint $Alternative\,(x_m, \{x'_{mi}\}_{i \in I})$ can be used to state that job m must be assigned to exactly one machine. If job m is assigned to machine i, we have $Presence(x'_{mi}) = 1$ and for $i' \in \; I \setminus i$, we have $Presence(x'_{m'i}) = 0$, $Start(x_m) = Start(x'_{mi})$, and $End(x_m) = End(x'_{mi})$. Sequence variables are defined on a set of interval variables and provide a permutation for the associated interval variables that are present in the solution. These types of variables can be used to define the processing sequence of a set of jobs on a machine. We note that sequence variables define a sequence of interval variables without enforcing any constraint on the relative position of the intervals in temporal dimension. Suppose that for machine $i \in I$, y_i is a sequence variable that is defined on the optional interval variables $\{x'_{mi}\}_{m \in M}$. Specialized constraint $NoOverlap(y_i)$ can be used to make sure that the associated interval variables present in the solution are sequenced in a way that they do not overlap in time. That is, if both x'_{mi} and $x'_{m'i}$ are present (i.e., jobs m and m' are assigned to machine i), and in y_i, x'_{mi} is sequenced before $x'_{m'i}$, we will have $Start(x'_{m'i}) \geq End(x'_{mi})$. $EndBeforeStart(x'_{mi}, x'_{m'i'}, a)$ is another specialized constraint in CP formulations to represent $Start(x'_{m'i'}) \geq End(x'_{mi}) + a$ [see, 53, 66].

Using interval and sequence variables, in combination with the above mentioned specialized constraints, we can build CP formulations for our problems as follows. Let α^I_m and α^O_n be interval variables representing the time intervals when incoming truck $m \in \mathcal{M}$ and outgoing truck $n \in \mathcal{N}$ are being processed, respectively. We define β^I_{mi} and β^O_{nj} as optional interval variables to represent the time intervals for the processing of incoming truck $m \in \mathcal{M}$ if assigned to inbound door $i \in \mathcal{I}$, and outgoing truck $n \in \mathcal{N}$ if assigned to outbound door $j \in \mathcal{J}$, respectively. For incoming truck $m \in \mathcal{M}$ and inbound door $i \in \mathcal{I}$, the length of the interval variable β^I_{mi} is equal to p^u_{mi}. Similarly, the length of the interval variable β^O_{nj} for outgoing truck $n \in \mathcal{N}$ and outbound door $j \in \mathcal{J}$, is equal to p^l_{nj}. For each truck, exactly one of these interval variables will be present in any feasible solution. Let γ_i^I be a sequence variable that keeps the sequence of incoming trucks assigned to inbound door $i \in \mathcal{I}$. These sequence variables are defined on the set of interval variables β_{mi}^I that are present at the solution, i.e., γ_i^I = sequence of $\left\{\beta_{mi}^I | Presence\left(\beta_{mi}^I\right) = 1, m \in \mathcal{M}\right\}$. Similarly, γ_j^O denotes a sequence variable that keeps the sequence of outgoing trucks assigned to outbound door $j \in \mathcal{J}$. These sequence variables are defined on the set of interval variables β_{nj}^O that are present at the solution, i.e., γ_j^O = sequence of $\left\{\beta_{nj}^O | Presence\left(\beta_{nj}^O\right) = 1, n \in \mathcal{N}\right\}$. Using the above variables together with z_{kij} and C_{max} described in Section 4.3, the CDSPHT-G can be formulated as follows:

$$
\begin{aligned}
\text{(CP-G)}\quad & \text{minimize}\ C_{max} & & \\
& \text{subject to}\ C_{max} = \max_{n \in \mathcal{N}} \left\{End\left(\alpha_n^O\right)\right\} & & (4.16)\\
& Alternative\left(\alpha_m^I, \left\{\beta_{mi}^I\right\}_{i \in \mathcal{I}}\right) & m \in \mathcal{M} \quad & (4.17)\\
& Alternative\left(\alpha_n^O, \left\{\beta_{nj}^O\right\}_{j \in \mathcal{J}}\right) & n \in \mathcal{N} \quad & (4.18)\\
& NoOverlap\left(\gamma_i^I\right) & i \in \mathcal{I} \quad & (4.19)\\
& NoOverlap\left(\gamma_j^O\right) & j \in \mathcal{J} \quad & (4.20)\\
& If\left(Presence\left(\beta_{o(k)i}^I\right) = 1 \& Presence\left(\beta_{d(k)j}^O\right) = 1\right) & & \\
& \qquad Then\left(z_{kij} = 1\right) & k \in \mathcal{K}, i \in \mathcal{I}, j \in \mathcal{J} \quad & (4.21)\\
& End\left(\alpha_m^I\right) + w_k d_{ij} z_{kij} \leq Start\left(\alpha_n^O\right) & & \\
& & k \in \mathcal{K}, i \in \mathcal{I}, j \in \mathcal{J}. \quad & (4.22)
\end{aligned}
$$

The objective function minimizes the makespan. Constraints (4.16) force the makespan to be equal to the maximum end time among the interval variables α_n^O. Constraints (4.17) and (4.18) assign each incoming and outgoing truck to exactly one inbound and outbound door, respectively. Constraints (4.19) and (4.20) ensure that the processing of trucks assigned to the same door do not overlap in time. Constraints

(4.21) assign a proper value to the variable z_{kij} with respect to the route assigned for product k. Constraints (4.22) guarantee that there is enough time for completing the unloading process and transferring all the required amounts of products to each outgoing truck before the loading process is scheduled. In a similar way, the CDSPHT-S can be formulated as follows:

$$\text{(CP-S)} \quad \text{minimize} \; C_{max}$$

$$\text{subject to} \; (4.16) - (4.20)$$

$$EndBeforeStart\left(\alpha_m^I, \alpha_n^O, w_k \overline{d}\right) \qquad k \in \mathcal{K}. \tag{4.23}$$

We note that constraints (4.23) are a simplified version of (4.22) to guarantee that for each $k \in \mathcal{K}$, the unloading process of the origin truck should end before the loading process of the destination truck starts, given that there is enough time for transferring all the products between the end of the unloading process and the start of the loading processes.

4.5 An Exact Algorithm for the CDSPHT-S

In this section, we present an exact BC algorithm for the CDSPHT-S. It uses the linear programming (LP) relaxation of formulation F1-S as a lower bounding procedure at nodes of the enumeration tree. The formulation is strengthened by the incorporation of several classes of valid inequalities that exploit the structure of the CDSPHT-S. These inequalities improve the polyhedral description of the convex hull of X which in turn, has a substantial positive impact in the overall convergence of the BC algorithm.

4.5.1 Valid inequalities

We next present eight classes of valid inequalities for the CDSPHT-S. These inequalities are designed to better characterize the interaction between incoming and outgoing truck sequencing decisions. The first four classes focus on restricting the set of time periods for scheduling outgoing trucks by taking into account the times at which incoming trucks are scheduled. Analogously, the last four classes focus on restricting the set of time periods for scheduling incoming trucks by taking into account the times at which outgoing trucks are scheduled. At the end of this section, we mention how all these inequalities can be adapted to remain valid for the more general CDSPHT-G. For the ease of readability, the proof of validity for each of these inequalities is provided in Appendix E. We recall that in the CDSPHT-S, transfer times are assumed to be constant, i.e., $d_{ij} = \bar{d}$, for all door pairs $(i, j) \in \mathcal{I} \times \mathcal{J}$.

The first set of inequalities are obtained by noting that the left-hand-side of constraints (4.15) can be strengthened by considering other time periods compatible with the incoming truck scheduling decisions presented in the right-hand-side of (4.15), and also by exploiting the fact that the summation on the right-hand-side does not depend on j.

Proposition 1. *For $k \in \mathcal{K}$ and $t \in \mathcal{T}$, the inequality*

$$\sum_{j \in \mathcal{J}} \sum_{t' \leq t} y_{d(k)jt'} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{kit}} x_{o(k)is}, \tag{V1}$$

is valid for X.

Let $\mathcal{N}_m = \{n \in \mathcal{N} | \exists k \in \mathcal{K} : o(k) = m, d(k) = n\}$ be the set of outgoing trucks requiring products from incoming truck $m \in \mathcal{M}$. For $m \in \mathcal{M}$ and $n \in \mathcal{N}_m$, we define k_{mn} as the product to be transferred from m to n, that is, $o(k) = m$ and $d(k) = n$. For each $m \in \mathcal{M}$, we also define $w_{k_m}^{min} = \min_{n \in \mathcal{N}_m} \{w_{k_{mn}}\}$ as the minimum amount of product to be transferred from incoming truck m to any outgoing truck.

The second class of inequalities exploits the incompatibilities of assigning different outgoing trucks to an outbound door j over a specific period of time.

Proposition 2. *For* $m \in \mathcal{M}, j \in \mathcal{J}, t \in \mathcal{T}, \mathcal{S}_{mit} = \left\{s \in \mathcal{T} | 0 \leq s \leq t - \left(p^u_{mi} + w^{min}_{k_m}\overline{d}\right)\right\}$, *and* $\mathcal{R}_{mjt} = \left\{r \in \mathcal{T} | t - \min_{n \in \mathcal{N}_m} \left\{p^l_{nj}\right\} < r \leq t\right\}$, *the inequality*

$$\sum_{n \in \mathcal{N}_m} \sum_{r \in \mathcal{R}_{mjt}} y_{njr} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{mit}} x_{mis}, \tag{V2}$$

is valid for X.

We next provide two classes of inequalities in an *extended* solution space considering the following set of decision variables. For each $m \in \mathcal{M}$ and $i \in \mathcal{I}$, we define the variable u_{mi} equal to one if and only if incoming truck m is assigned to inbound door i.

Proposition 3. *For* $k \in \mathcal{K}, i \in \mathcal{I}$, *and* $t \in \mathcal{T}$, *the inequality*

$$\sum_{j \in \mathcal{J}} \sum_{t' \leq t} y_{d(k)jt'} \leq \sum_{s \in \mathcal{S}_{kit}} x_{o(k)is} + 1 - u_{o(k)i}, \tag{V3}$$

is valid for X.

Proposition 4. *For* $m \in \mathcal{M}, i \in \mathcal{I}, j \in \mathcal{J}$, *and* $t \in \mathcal{T}$, *the inequality*

$$\sum_{n \in \mathcal{N}_m} \sum_{r \in \mathcal{R}_{mjt}} y_{njr} \leq \sum_{s \in \mathcal{S}_{mit}} x_{mis} + 1 - u_{mi}, \tag{V4}$$

is valid for X.

The next four classes of valid inequalities can be obtained using similar arguments to the ones used in (V1)-(V4), when restricting the set of time periods for scheduling incoming trucks by taking into account the times at which outgoing trucks are scheduled. The fifth class of inequalities are obtained by considering a set of time periods compatible with the outgoing truck scheduling decisions, and by exploiting the fact that each truck can be assigned to exactly one door.

Proposition 5. *For $k \in \mathcal{K}$, $t \in \mathcal{T}$, and $\overline{\mathcal{S}}_{kt} = \Big\{ s \in \mathcal{T} | t + \min_{i \in \mathcal{I}} \{ p^u_{o(k)i} \} + w_k \overline{d} \leq s \leq \mathcal{T} \Big\}$, the inequality*

$$\sum_{i \in \mathcal{I}} \sum_{t' \geq t} x_{o(k)it'} \leq \sum_{j \in \mathcal{J}} \sum_{s \in \overline{\mathcal{S}}_{kt}} y_{d(k)js}, \tag{V5}$$

is valid for X.

We define $\mathcal{M}_n = \{m \in \mathcal{M} | \exists k \in \mathcal{K} : o(k) = m, d(k) = n\}$ as the set of incoming trucks that carry products requested by outgoing truck $n \in \mathcal{N}$. We also define $w^{min}_{k_n}$ as the minimum amount of product outgoing truck n must receive, i.e., $w^{min}_{k_n} = \min_{m \in \mathcal{M}_n} \{w_{k_{mn}}\}$. The sixth class of inequalities exploits the incompatibilities of assigning different incoming trucks to an inbound door i over a specific time period.

Proposition 6. *For $n \in \mathcal{N}, i \in \mathcal{I}, t \in \mathcal{T}$, $\overline{\mathcal{R}}_{nit} = \{ r \in \mathcal{T} | t \leq r < t + \min_{m \in \mathcal{M}_n} \{p^u_{mi}\}\}$, and $\overline{\mathcal{S}}_{nit} = \big\{ s \in \mathcal{T} | t + \min_{m \in \mathcal{M}_n} \{p^u_{mi}\} + w^{min}_{k_n} \overline{d} \leq s \leq \mathcal{T} \big\}$, the inequality*

$$\sum_{m \in \mathcal{M}_n} \sum_{r \in \overline{\mathcal{R}}_{nit}} x_{mit} \leq \sum_{j \in \mathcal{J}} \sum_{s \in \overline{\mathcal{S}}_{nit}} y_{njs}, \tag{V6}$$

is valid for X.

Finally, we provide the two last classes of inequalities in an *extended* solution space considering the following set of variables. For each $n \in \mathcal{N}$ and $j \in \mathcal{J}$, we define the variable v_{nj} equal to one if and only if outgoing truck n is assigned to outbound door j.

Proposition 7. *For $k \in \mathcal{K}, j \in \mathcal{J}$, and $t \in \mathcal{T}$, the inequality*

$$\sum_{i \in \mathcal{I}} \sum_{t' \geq t} x_{o(k)it'} \leq \sum_{s \in \overline{\mathcal{S}}_{kt}} y_{d(k)js} + 1 - v_{d(k)j}, \tag{V7}$$

is valid for X.

Proposition 8. *For $n \in \mathcal{N}, i \in \mathcal{I}, j \in \mathcal{J}$, and $t \in \mathcal{T}$, the inequality*

$$\sum_{m \in \mathcal{M}_n} \sum_{r \in \overline{\mathcal{R}}_{nit}} x_{mit} \leq \sum_{s \in \overline{\mathcal{S}}_{nit}} y_{njs} + 1 - v_{nj}, \tag{V8}$$

is valid for X.

We conclude this section by noting that all these inequalities can be adapted for the general CDSPHT-G. In particular, if we replace $\mathcal{S}_{kit}$ in inequalities (V1) and (V3) by $\mathcal{S}'_{kit} = \Big\{s \in \mathcal{T} | 0 \leq s \leq t - \Big(p^u_{o(k)i} + w_k \times \min_{j \in \mathcal{J}}\{d_{ij}\}\Big)\Big\}$, and $\mathcal{S}_{mit}$ in inequalities (V2) and (V4) by $\mathcal{S}'_{mijt} = \big\{s \in \mathcal{T} | 0 \leq s \leq t - \big(p^u_{mi} + w^{min}_{k_m} d_{ij}\big)\big\}$, the resulting inequalities are valid for the CDSPHT-G. Moreover, if we replace $\overline{\mathcal{S}}_{kt}$ in inequalities (V5) and (V7) by $\overline{\mathcal{S}}'_{kjt} = \Big\{s \in \mathcal{T} | t + \min_{i \in \mathcal{I}}\{p^u_{o(k)i}\} + w_k \times \min_{i \in \mathcal{I}}\{d_{ij}\} \leq s \leq \mathcal{T}\Big\}$, and $\overline{\mathcal{S}}_{nit}$ in inequalities (V6) and (V8) by $\overline{\mathcal{S}}'_{nijt} = \big\{s \in \mathcal{T} | t + \min_{m \in \mathcal{M}_n}\{p^u_{mi}\} + w^{min}_{k_n} d_{ij} \leq s \leq \mathcal{T}\big\}$, the resulting inequalities are valid for the CDSPHT-G.

4.5.2 A branch-and-cut algorithm

We now present an exact BC algorithm for solving the CDSPHT-S. The idea is to solve the LP relaxation of F1-S with a cutting-plane algorithm by initially incorporating constraints (4.1)-(4.5), (4.10), (4.11), (4.14), and

$$u_{im} = \sum_{t \in \mathcal{T}} x_{mit} \qquad m \in \mathcal{M}, i \in \mathcal{I} \tag{4.24}$$

$$v_{jn} = \sum_{t \in \mathcal{T}} y_{njt} \qquad n \in \mathcal{N}, j \in \mathcal{J} \tag{4.25}$$

$$u_{im} \geq 0 \qquad m \in \mathcal{M}, i \in \mathcal{I} \tag{4.26}$$

$$v_{jn} \geq 0 \qquad n \in \mathcal{N}, j \in \mathcal{J}, \tag{4.27}$$

and iteratively adding constraints (V1)-(V8) only when violated by the current LP solution by a minimum threshold value ϵ. Note there is no need to force the integrality conditions on the u and v variables given that constraints (4.24), (4.25), in combination with the integrality conditions of the x and y variables, will ensure u and v variables to take binary values. When no more violated inequalities are found, we resort to the branch-and-bound (BB) algorithm of CPLEX for solving the resulting formulation by enumeration. We also use a call-back function for generating additional violated inequalities (V1)-(V8) at some nodes of the enumeration tree. Preliminary experiments showed that adding this inequalities at the root node had a negative impact on the overall CPU time. Therefore, the separation problem of all inequalities is solved by inspection and is carried out at every other node for which the depth is multiple of 20 (excluding the root node). We limit the number of cutting-plane iterations at the nodes to two. In Section 4.8, we compare the relative impact of each of these classes of inequalities and evaluate promising combinations for separating only a subset of them. The outcome of these experiments is a fine-tuned version of our BC algorithm in which only a subset of these inequalities is used to further improve its convergence.

4.6 An Iterated Local Search Matheuristic for the CDSPHT-G

In this section we introduce a matheuristic for the CDSPHT-G that decomposes the problem into two independent parallel machine scheduling problems by temporarily fixing in an iterative manner the incoming and outgoing trucks' assignments and sequencing decisions. This procedure can be seen as a local search (LS) in which two very large neighborhoods, each associated with either incoming or outgoing trucks, are used to improve an initial feasible solution. We start by fixing the assignment and sequencing decisions of one side of the cross-dock, and then solve a parallel machine scheduling problem to obtain the trucks' assignment and sequencing decisions on the other side. This procedure alternates from one side to the other until the makespan cannot be further improved.

To describe the details of our algorithm, we assume that we start by fixing incoming trucks' assignments and sequencing decisions (in Section 4.8 we provide a performance comparison when starting on either side). Therefore, the problem reduces to a parallel machine scheduling problem with release dates that determines the outgoing trucks' assignments and sequences. Given that the schedule of incoming trucks is fixed, for each outgoing truck we can calculate a release date based on the time required to unload all the associated incoming trucks and to transfer all the products to such outgoing truck. We refer to this problem as the *outgoing cross-dock scheduling problem with release dates* (OCDSP-RD). For each $n \in \mathcal{N}$ and $j \in \mathcal{J}$, we define the parameter $RD_{nj} = \max_{m \in \mathcal{M}_n} \left\{ t^*_m + p^u_{md^*_m} + w_{k_{mn}} d_{d^*_m j} \right\}$, equal to the release time of outgoing truck n at door j, where t^*_m and d^*_m denote the start time of incoming truck m and its assigned door, respectively, in the current solution. The OCDSP-RD can be formulated as the following MIP:

$$
\begin{aligned}
\text{(MIP-OT)} \quad & \text{minimize} \ C_{max} \\
& \text{subject to} \ (4.1), (4.3), (4.5), (4.11) \\
& \qquad \sum_{t \in \mathcal{T}} t y_{njt} \geq RD_{nj} \qquad\qquad n \in \mathcal{N}, j \in \mathcal{J}. \qquad (4.28)
\end{aligned}
$$

Alternatively, the OCDSP-RD can also be formulated as the following CP:

$$
\begin{aligned}
\text{(CP-OT)} \quad & \text{minimize} \ C_{max} \\
& \text{subject to} \ (4.14), (4.17), (4.19) \\
& \qquad Start\left(\beta^O_{nj}\right) \geq RD_{nj} \qquad\qquad n \in \mathcal{N}, j \in \mathcal{J}. \qquad (4.29)
\end{aligned}
$$

The optimal solution value of OCDSP-RD provides a valid upper bound on the optimal solution value of CDSPHT-G. The proposed algorithm continues by fixing the obtained outgoing trucks' assignments and sequencing decisions from OCDSP-RD. By doing so, the problem reduces to a parallel machine scheduling problem with due dates to determine the best incoming trucks' assignments and sequencing decisions given the current outgoing trucks decisions. Given that the schedule of outgoing trucks is fixed, for each incoming truck we can calculate a due date in such a way that after unloading is completed, there is still enough time left to transfer all the products to the associated outgoing trucks before the scheduled start times of outgoing trucks. We refer to this problem as the *incoming cross-dock scheduling problem with due dates* (ICDSP-DD).

Given that the makespan is calculated from the completion time of outgoing trucks, if we derive a formulation for the ICDSP-DD directly from the F1-G, as is the case for the OCDSP-RD, the obtained formulation will lack an objective function. As suggested by Boysen et al. [11], makespan minimization can be expressed based on incoming trucks by maximizing the start time of the first scheduled incoming truck(s). In other words, the goal of the makespan minimization is to push back the schedule of the trucks as much as possible in order to complete all tasks at the earliest possible time. From the upper chart of Figure 4.1, we note that the objective function can be alternatively calculated based on the start time of incoming trucks. In this case, the goal is to finish all the tasks at time T and push forward the schedule of all the trucks as much as possible in a way that all the trucks are scheduled and unloaded in the smallest possible time interval.

The next result show that OCDSP-RD and ICDSP-DD are actually equivalent from an optimization perspective. The proof is given in Appendix F.

Proposition 9. *If $d_{ij} = d_{ji}$ for all door pairs $(i, j) \in \mathcal{I} \times \mathcal{J}$, any instance of the ICDSP-DD can be transformed into an instance of the OCDSP-RD.*

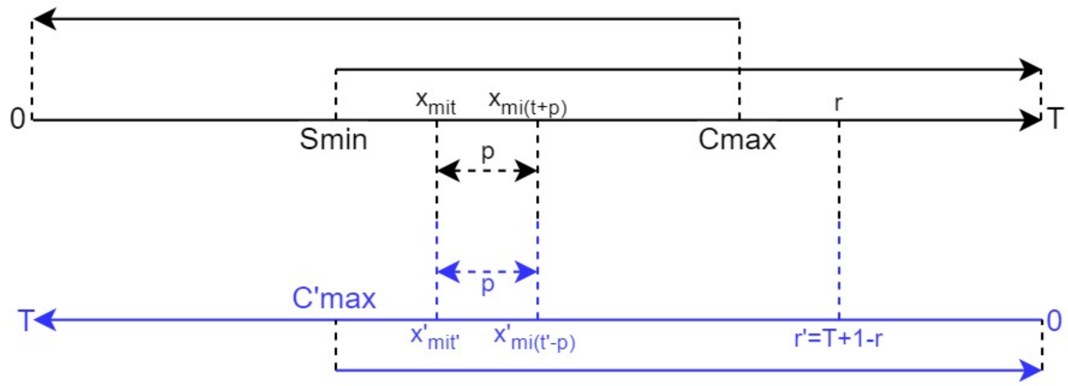


Figure 4.1: Illustration of reverse planning horizon

An immediate consequence of the above result is that formulations MIP-OT and CP-OT, and any other formulation and solution algorithm developed for the OCDSP-RD, can also be used to solve the ICDSP-DD. This is the approach we follow in our algorithms.

Algorithm 4 summarizes the steps of the proposed LS matheuristic, which requires an initial feasible solution $\mathcal{S}^0$ as an input. Moreover, there are two parameters that need to be set at the beginning of the algorithm. The first one (I_Side) relates to the cross-dock side to be fixed first. If $I\_Side = in$, the algorithm starts by fixing the inbound side and if $I\_Side = out$, the algorithm starts by fixing the outbound side. The second one ($Stoplimit$) corresponds to the number of allowed iterations without improvement before the algorithm terminates. At any iteration where the algorithm fails to improve the best-known solution, we check if the obtained solution is different from the ones previously evaluated. If the obtained solution has already been visited, the algorithm terminates.

The quality of the solutions obtained by Algorithm 1 can be further improved by incorporating a multi-start framework such as ILS [47]. ILS is an iterative procedure that starts from an initial solution which is then improved by using local search. Once a local optimal solution has been found, a new starting point is obtained by applying destroy and repair mechanisms to the solution obtained by local search. Finally, an

Algorithm 4: Local Search Matheuristic

Input: $I\_Side, Stop\_limit$ and initial solution $\mathcal{S}^0$
Initialize: $\mathcal{S}^{sol}_{in} = \emptyset, \mathcal{S}^{sol}_{out} = \emptyset, Current = 0, Best = \infty, counter = 0, itr = 0$
if $I\_Side = in$ **then**
  $\mathcal{S}^0 = \{(ass_{in}) \cup (seq_{in})\}$ $\mathcal{S}^{sol}_{in} \leftarrow \mathcal{S}^0,\ side \leftarrow in,\ \mathcal{TR} = \mathcal{M},\ \mathcal{DR} = \mathcal{I}$
else
  $\mathcal{S}^0 = \{(assign_{out}) \cup (seq_{out})\}$ $seq_{out} \leftarrow reverse(seq_{out})$
  $\mathcal{S}^{sol}_{out} \leftarrow \mathcal{S}^0,\ side \leftarrow out,\ \mathcal{TR} = \mathcal{N},\ \mathcal{DR} = \mathcal{J}$
end
while $Current \leq Best\ \&\ counter \leq Stop\_limit$ **do**
  $itr++$
  Calculate release dates of trucks: $\mathcal{RD} = \{RD_1, .., RD_{|\mathcal{TR}|}\}$
  Solve OCDSP-RD to obtain C_{max} and $\mathcal{S}^{itr} = \{(ass_{side}) \cup (seq_{side})\}$
  $Current \leftarrow C_{max},\ seq_{side} \leftarrow reverse(seq_{side})$
  $\mathcal{S}^{sol}_{side} = \mathcal{S}^{sol}_{side} \cup \mathcal{S}^{itr}$
  Update $side,\ \mathcal{TR},\ \mathcal{DR}$
  if $Current < Best$ **then**
    $Best \leftarrow Current,\ \mathcal{S}^{sol}_{side} \leftarrow \mathcal{S}^{itr},\ \mathcal{S}^{best} \leftarrow \mathcal{S}^{itr} \cup \mathcal{S}^{itr-1}$ $counter = 0$
  end
  if $Current = Best$ **then**
    if $\mathcal{S}^{itr} \in \mathcal{S}^{sol}_{side}$ **then**
      $break$
    else
      $counter++$
    end
  end
end
Output: best-known solution $\mathcal{S}^{best}$ and valid upper bound $Best$.

acceptance criterion determines from which solution the search will continue. Algorithm 5 depicts a summary of our ILS matheuristic. We start with the solution obtained from Algorithm 1 and perturb it to find a new initial solution and apply once more the LS matheuristic. This procedure is repeated for a fixed number of iterations $MaxIter$. The ILS framework of the proposed algorithm is similar to the ILS-VND algorithm proposed in [59]. There are several parameters in the algorithm. Parameter $p \in [0, 1]$ determines the degree of perturbation. We start with an initial value of p_0 and update it according to δ_p. We use λ to determine the acceptance criteria of a perturbed solution. It is initially set to λ_0 and is updated with respect to δ_λ. To obtain a perturbed solution, we first destroy the solution partially and then repair it using a MIP (or CP) formulation to obtain a complete solution. However, it may be the case that within the given CPU time limit, it is not possible to obtain an optimal or even feasible solution. In these cases, if a feasible (but not necessarily optimal) solution is obtained we apply a *variable neighborhood descend* (VND) procedure given in [59] to improve the solution. In this VND procedure, five different neighborhoods are explored with a given sequence. First, all solutions that are obtained by choosing any pair of incoming trucks and exchanging their doors and positions in the sequence, are explored. Then, all solutions that can be obtained by selecting one incoming truck at a time and move it to a different door and/or a different position in sequence, are explored. In the third and fourth steps, the same procedures are applied for the outgoing trucks and the obtained solutions are explored. Finally, in the fifth step, all solutions that can be obtained by simultaneously swapping a pair of incoming trucks and a pair of outgoing trucks assigned to the doors within a specific distance, are explored. If the MIP (or CP) formulation cannot find a feasible solution, we apply a repair procedure that follows similar steps as the sequential constructive algorithm in [59] to complete the solution. In order to repair a solution, each unscheduled incoming truck is assigned to an inbound door, as the last truck of the sequence, such that minimum estimated makespan is obtained. Moreover, each unscheduled outgoing truck is allocated to the outbound door with the earliest starting time to process that truck. After each outgoing truck is scheduled, a VND procedure considering the first four neighborhoods described earlier, is applied to improve the partial schedule. Once the solution is complete, another VND procedure considering all the five neighborhoods is applied to improve the final schedule.

Algorithm 5: Iterated Local Search Matheuristic

Input: $MaxIter, p_0, \lambda_0, \delta_p, \delta_\lambda$ and initial solution $\mathcal{S}^0$
Initialize: $p \leftarrow p_0, \lambda \leftarrow \lambda_0$
Apply Local Search Matheuristic on $\mathcal{S}^0$ to obtain $\mathcal{S}^*$ and C_{max}
$Best \leftarrow C_{max},\ Current \leftarrow C_{max},\ \mathcal{S}^{best} \leftarrow \mathcal{S}^*,\ \mathcal{S}'^* \leftarrow \mathcal{S}^*$
for $itr = 0$ *to* $MaxIter$ **do**
  Partially destroy $\mathcal{S}'^*$:
    Randomly select $V^{in} \subset \mathcal{M}$ such that $|V^{in}| = \lceil p \times |\mathcal{M}| \rceil$
    Randomly select $V^{out} = \{n \in \mathcal{N} | \exists m \in V^{in} : n \in \mathcal{N}_m\}$
    Mark all trucks in V^{in} and V^{out} as unscheduled in $\mathcal{S}'^*$
  Repair $\mathcal{S}'^*$ to obtain a complete solution:
    Solve CDSPHT-G with partially fixed decisions and get the $Status$
    if $Status$=*feasible* **then**
      Apply VND to improve the solution
  end
    if $Status$=*infeasible* **then**
      Use a simple heuristic to complete the solution
  end
  Apply Local Search Matheuristic using $\mathcal{S}'^*$ to obtain C_{max}
  if $C_{max} < Best$ **then**
    $Best \leftarrow C_{max},\ \mathcal{S}^{best} \leftarrow \mathcal{S}'^*$
  end
  if $\lambda \times C_{max} < Current$ **then**
    $Current \leftarrow C_{max},\ \mathcal{S}^* \leftarrow \mathcal{S}'^*$
  end
  $p \leftarrow p(1 - \delta_p),\ \lambda \leftarrow \lambda(1 + \delta_\lambda)$
end
Output: best-known solution $\mathcal{S}^{best}$ and valid upper bound $Best$.

4.7 An Approximate Algorithm for the CDSPHT-G

We next describe an approximate algorithm for the CDSPHT-G that provides both lower and upper bounds on the optimal solution value to provide an optiality gap. To obtain valid upper bounds, the algorithm uses the Matheuristics presented in the previous section. To compute valid lower bounds, it resorts to the solution of several (integer) relaxations of the CDSPHT-G. In particular, we consider four different integer relaxations obtained by relaxing the integrality conditions on either x or y variables on both formulations F1-G and F2-G. Another integer relaxation of CDSPHT-G is precisely CDSPHT-S. This is true given that the feasible region of

CDSPHT-G is a subset of the feasible region of CDSPHT-S (i.e., $X^{'} \subseteq X$) and the objective function value of CDSPHT-S is always equal to that of CDSPHT-G for every feasible solution in $X^{'}$. Yet another relaxation of the CDSPHT-G is a problem in which each side of the cross-dock is scheduled independently. We do so by solving two MIP-OT with $RD_{nj} = 0$ for each $n \in \mathcal{N}$ and $j \in \mathcal{J}$, one for each inbound/outbound side.

We use a combination of these relaxations in our algorithm to provide a lower bound. In particular, we solve CDSPHT-S using our BC algorithm and solve one of the other relaxations described above. Section 4.8 provides a comparison of all these relaxation procedures to determine which one is the most promising to use in our approximate algorithm. We then use the solution of CDSPHT-S as initial solution in one of our matheuristics and we execute it two times, one by fixing first the inbound side and another by fixing first the outbound side. Algorithm 6 depicts the proposed approximate algorithm.

Algorithm 6: Approximate Algorithm

Solve F1-S using the BC algorithm and obtain LB and solution $\mathcal{S}$ (if it exists)
Solve another relaxation problem and obtain LB' and solution $\mathcal{S}'$
if $LB' > LB$ **then**
  $LB \leftarrow LB'$
end
if $\mathcal{S} = \emptyset$ **then**
  $\mathcal{S} \leftarrow \mathcal{S}'$
end
Apply matheuristic with $\mathcal{S}$ and $I\_Side = in$ to obtain UB and $\mathcal{S}^{best}$
Apply matheuristic with $\mathcal{S}$ and $I\_Side = out$ to obtain the $UB'$and $\mathcal{S}'$
if $UB' < UB$ **then**
  $UB \leftarrow UB', \quad \mathcal{S}^{best} \leftarrow \mathcal{S}'$
end
Calculate optimality gap: $\Delta = 100 \times \frac{UB-LB}{UB}$
Output: UB and Δ.

4.8 Computational Experiments

This section summarizes the results of extensive computational experiments carried out to evaluate the performance of the proposed formulations and solution algorithms. All MIP/CP formulations and algorithms were coded in C/C++ and solved with CPLEX 12.10.0 using Callable Library and Concert Technology on an Intel Xeon CPU E5-2687W v3 processor at 3.10 GHz and 750 GB of RAM under a Linux environment. The maximum number of used threads was set to seven.

We have used a set of 44 benchmark instances given by Sayed et al. [59] to perform the experiments. These instances were generated following the procedures described in [28] and [50] for the CDAP. To generate the flow matrix, for each product k the amount of product w_k that outgoing truck $d(k)$ must receive from incoming truck $o(k)$ is randomly generated as an integer quantity that follows a uniform distribution $U[1, 5]$. Generated instances consist of four subsets of instances that differ in terms of the density of the flow matrix: $25\%, 35\%, 50\%$, and 75%. The density of the flow matrix is defined as the ratio between the number of products that should be transferred from incoming to outgoing trucks and $|\mathcal{M}| \times |\mathcal{N}|$. The number of considered incoming/outgoing trucks is 8, 9, 10, 11, 12, 15, 20, and 50. The number of considered inbound/outbound doors is 4, 5, 6, 7, 10, and 30. Each instance is characterized by three values and is referred to as AxBxC, where A, B, and C correspond to the number of the incoming and outgoing trucks, the number of inbound and outbound doors, and the density of the flow matrix, respectively.

The experiments are divided in four parts. In the first part, we compare the performance of the MIP and CP formulations given in Sections 4.3 and 4.4 for the CDSPHT-G and CDSPHT-S when solved by CPLEX using its default settings. In the second part, we analyze the strength and usefulness of the valid inequalities described in Section 4.5 when used within the BC algorithm. We also compare the best configuration of our BC algorithm against F1-S when solved by CPLEX. In the third part, we evaluate the performance of the approximate algorithm for the CDSPHT-G presented in Section 4.7 when using different integer relaxations and matheuristics for computing lower and upper bounds, respectively. In the last part, we study the impact in the quality of the obtained solutions when solving the CDSPHT-G with door-dependent transfer times, or its relaxation CDSPHT-S with constant transfer times.

4.8.1 A comparison of formulations for the CDSPHT-G and CDSPHT-S

For CDSPHT-G, we compare the performance of the proposed CP-G formulation with both F1-G and F2-G formulations proposed in [59]. To make the comparison as fair as possible, we implemented and run F1-G and F2-G with the same version of CPLEX and default settings on the same computer. Table 4.1 summarizes the results of this comparison. For these experiments, we have considered a time limit of 24 hours. The first four rows provide for each of the formulations: the number of instances solved to optimality, the number of instances for which even a feasible solution was not found within the time limit, the number of instances in which the best-known solution was obtained, and the number of instances that provided the best CPU time for finding the optimal solution, respectively. Furthermore, we report for each formulation: the arithmetic and geometric means of the % deviations from the best upper bounds obtained by at least one of the formulations, the arithmetic and geometric means of obtained optimality gaps by CPLEX, and the arithmetic and geometric means of CPU times. To calculate the geometric mean for a set of values that includes zero, we have added one to each value and then, we have subtracted one from the obtained mean.

According to Table 4.1, CP-G outperforms both MIP formulations with respect to most measures. We note that the average optimality gap is calculate based on

Table 4.1: A comparison of MIP and CP formulations for the CDSPHT-G.

	F1-G	F2-G	CP-G
Optimal instances (#)	19/44	19/44	24/44
Infeasible instances (#)	10/44	6/44	4/44
Best UB (#)	30/44	30/44	40/44
Best time (#)	6/44	4/44	16/44
Arit. mean of dev (%)	0.14	0.14	0.00
Geo. mean of dev (%)	0.09	0.09	0.00
Arit. mean of gap (%)	2.65	4.96	6.93
Geo. mean of gap (%)	1.15	2.02	2.04
Arit. mean of time (sec)	50,109	49,433	37,008
Geo. mean of time (sec)	2,183	1,922	642

the instances for which at least one feasible solution is found, and since CP-G could find feasible solutions for more instances than the MIPs, the average optimality gap reported for CP-G is slightly higher than those reported for the MIPs. It is worth mentioning that out of all 44 instances, 27 instances could be solved optimally using at least one of the formulations. Out of those, 17 instances were solved by all formulations, seven were solved by CP-G, two were solved by both F1-G and F2-G, one was solved by F1-G, and one was solved by F2-G. For 14 of the instances, only a feasible solution could be found and for three instances no feasible solution was found within 24 hours. For detailed results obtained by each formulation, we refer to Table 6 in the Appendix G.

We now compare the MIP and CP formulations for the CDSPHT-S provided in Sections 4.3 and 4.4. For these experiments, we have used a time limit of one hour. Table 4.2 summarizes the obtained results. In this table, in addition to the information given in the previous table we also report the number of instances for which the best lower bound was obtained.

According to Table 4.2, both formulations have a similar performance in terms of the number of instances solved to optimality. However, F1-S could not find even a feasible solution for eight of the considered instances, while CP-S was able to find at least one feasible solution for all instances. In general, we can observe that F1-S

Table 4.2: A comparison of MIP and CP formulations for the CDSPHT-S.

	F1-S	CP-S
Optimal instances (#)	23/44	23/44
Infeasible instances (#)	8/44	0/44
Best UB (#)	30/44	44/44
Best LB (#)	39/44	24/44
Arit. mean of gap (%)	2.35	7.47
Geo. mean of gap (%)	0.88	2.59
Arit. mean of time (sec)	1,859	1,672
Geo. mean of time (sec)	112	92

works better in terms of the quality of lower bounds, while CP-S works well at finding upper bounds. Detailed results obtained by F1-S and CP-S are given in Table 7 in the Appendix G.

4.8.2 Impact of valid inequalities in BC algorithm for the CDSPHT-S

We next present extensive computational experiments preformed to asses the proposed BC algorithm using different subsets of inequalities described in Section 4.5. We have proposed eight families of inequalities for the CDSPHT-S. To evaluate the impact of these families of inequalities, we have classified them according to the type of variables on the left-hand-side, the type of incompatible decisions that are aggregated on the left-hand-side, and the use of extra variables on the right-hand-side. According to this classification, 27 combinations of inequalities are obtained and depicted in Table 4.3. In this table, the first column represents the number of the combinations, the second determines the type of variables on the left-hand-side of the inequalities. X, Y, respectively, denote that the inequalities considered in the combination have inbound-related variables and outbound-related variables on the left-hand-side, while XY denotes that a mix of inequalities that have inbound- and outbound-related variables on the left-hand-side. The column under the heading *Decision*, shows whether the aggregated decisions on the left-hand-side of the inequality are related to the doors (D), trucks (T), or both (DT). The fourth column denotes the variant of the inequalities that has been used. Variant a represents inequalities that have a summation over doors on the right-hand-side, while b corresponds to the inequalities that have an extra variable on the right-hand-side and are disaggregated with respect to the doors. ab denotes that both variants a and b are considered. Finally, the last column specifies the inequalities that are considered in each combination.

Table 4.3: Characteristics of different combinations of tested inequalities in BC algorithm.

Combination	Variable	Decision	Variant	Inequalities
1	X	D	a	5
2			b	7
3			ab	5,7
4		T	a	6
5			b	8
6			ab	6,8
7		DT	a	5,6
8			b	7,8
9			ab	5,6,7,8
10	Y	D	a	1
11			b	3
12			ab	1,3
13		T	a	2
14			b	4
15			ab	2,4
16		DT	a	1,2
17			b	3,4
18			ab	1,2,3,4
19	XY	D	a	1,5
20			b	3,7
21			ab	1,3,5,7
22		T	a	2,6
23			b	4,8
24			ab	2,4,6,8
25		DT	a	1,2,5,6
26			b	3,4,7,8
27			ab	1,2,3,4,5,6,7,8

To assess the impact of the inequalities, we have tested each of the combinations given in Table 4.3 on 11 instances with our BC algorithm for solving the CDSPHT-S given a time limit of two hours. For these experiments, we have selected those instances which are not too easy to solve but also not too difficult to solve so that the optimal solution can be found in 24 hours. To perform each of these experiments, we have initially solved F1-S using the BB algorithm of CPLEX, and then, at the nodes for which the depth is a multiple of 20, we solve several separation problems to see if any of the considered inequalities are violated. All the violated inequalities are added to the formulation. For the combinations that we consider the *ab* variant of the inequalities, instead of adding all the violated inequalities, we only add the most violated ones. Table 4.4, summarizes the results obtained for each combination over 11 instances. We have also included the results obtained by F1-S without considering any of the inequalities in the first row of the table. For each row of the table, we report the number of instances solved optimally, the number of instances for which the best CPU time among all the combinations is obtained, the number of instances for which a feasible solution could not be found within the time limit, average CPU time, average number of explored nodes, average number of added user cuts, and the average percentage of time spent to solve the separation problems, respectively. According to Table 4.4, combination (3) outperforms all other combinations and combination (23) shows the worst performance. Therefore, from now on we use combination (3) of the inequalities in our BC algorithm.

In order to better evaluate the impact of using our proposed BC algorithm, we tested the best version of our BC algorithm on all the instances given a time limit of two hours. For comparison purposes, we have also solved the instances by simply solving the F1-S using CPLEX considering similar setting as the BC. Table 4.5 summarizes the obtained results. Using the proposed BC algorithm the number of instances that can be solved optimally increases. Furthermore, we are able to find feasible solutions for three instances for which we cannot find even a single feasible solution using CPLEX. Furthermore, the proposed BC algorithms reduces the average CPU time and the number of explored nodes significantly. Detailed results are

Table 4.4: A comparison of BC algorithm with different combinations of inequalities for the CDSPHT-S.

Valid inequalities				Opt.(#)	Best(#)	Inf.(#)	Time	Node(#)	Cut(#)	Sep.(%)
Original Problem				7	0	0	3,996	1,765,207	NA	NA
1			a	11	3	0	1,036	99,836	1,070	2
2		D	b	4	0	0	5,533	605,258	5,986	9
3			ab	11	4	0	865	94,118	1,094	4
4			a	5	0	0	4,466	843,671	1,500	4
5	X	T	b	3	0	0	5,469	716,736	7,512	14
6			ab	4	1	0	5,187	997,903	1,664	7
7			a	10	1	0	2,014	102,515	1,652	2
8		DT	b	5	0	0	5,521	285,688	9,209	12
9			ab	9	0	0	2,390	90,132	1,842	3
10			a	10	1	0	1,853	195,983	1,073	3
11		D	b	4	0	0	5,275	479,729	6,217	9
12			ab	9	0	0	1,938	166,250	1,180	4
13			a	7	0	0	3,837	398,766	1,456	2
14	Y	T	b	4	0	0	5,853	461,325	7,800	8
15			ab	5	0	0	4,828	408,346	1,710	3
16			a	8	0	0	2,696	43,277	1,878	1
17		DT	b	4	0	0	5,267	165,195	10,219	7
18			ab	8	0	0	2,821	48,327	2,138	2
19			a	9	0	0	2,388	140,277	2,136	4
20		D	b	4	0	1	5,693	169,640	9,782	5
21			ab	8	1	0	2,776	189,204	2,137	8
22			a	5	0	0	4,877	181,091	2,737	2
23	XY	T	b	3	0	1	6,381	264,067	11,525	8
24			ab	4	0	0	5,407	207,169	2,971	3
25			a	6	0	0	3,744	34,830	3,520	1
26		DT	b	4	0	1	5,432	92,195	13,055	5
27			ab	7	0	1	3,406	32,625	3,574	2

given in Table 8 in Appendix G.

4.8.3 Analysis of the approximate algorithm for the CDSPHT-G

We next present several experiments carried out to evaluate the performance of our proposed approximate algorithm with respect to the different settings and also to assess the quality of the obtained bounds as compared to the ones provided by MIP and CP formulations. As discussed in Section 4.7, different relaxation problems can be

Table 4.5: Performance of BC algorithm using combination (3).

	CPLEX	BC
Optimal instances (#)	27/44	31/44
Infeasible instances (#)	8/44	5/44
Best UB (#)	33/44	36/44
Best time (#)	15/44	31/44
Arit. mean of gap (%)	2.57	2.76
Geo. mean of gap (%)	0.68	0.63
Arit. mean of explored nodes	1,079,549	84,307
Geo. mean of explored nodes	10,423	1,328
Arit. mean of time (sec)	3,204	2,358
Geo. mean of time (sec)	138	74

used within our approximate algorithm. To evaluate the impact of these relaxations on the performance of the algorithm, we considered eight different settings for the approximate algorithm and compared them with respect to the obtained optimality gaps. For this comparison, we considered one hour time limit for solving the relaxation problems. After a relaxed problem is solved, we applied the local search matheuristic (Algorithm 1) on the obtained solution, using CP-OT to calculate the upper bound. The summary of the obtained results are reported in Table 4.6. The second and third columns of this table show the results obtained by relaxing outbound variables and inbound variables, respectively, in F1-G. The headings *in* and *out* indicate that the matheuristic starts by either fixing the incoming trucks decisions or the outgoing trucks decisions, respectively. We note that when we relax the outbound side in the relaxed problem, we will obtain the decisions related to the inbound side, and thus, the matheuristic starts by fixing these decisions. The fourth and fifth columns give the results of relaxing outbound and inbound variables in F2-G, respectively. As mentioned, we can also start the algorithm by solving the CDSPHT-S. We recall that, according to Table 4.2, F1-S outperforms CP-S in terms of the quality of the lower bounds. Since in the approximate algorithm the quality of the LB is very important, we have used F1-S to solve the specific case problem. The results obtained using this model are reported in the sixth and seventh columns. Finally, we have also used

MIP-OT, considering all release dates to be zero, to solve two independent parallel machine scheduling problems as the last relaxation problems. These results are shown in the last two columns. For each of the considered relaxation problems the number of optimal instances, the number of instances for which no feasible solution is found within the time limit, and arithmetic and geometric means of the optimality gaps are reported. For the detail results we refer to the Table 9 in Appendix G.

Table 4.6: A comparison of various integer relaxations for the CDSPHT-G.

	F1-G		F2-G		F1-S		MIP-OT	
	in	*out*	*in*	*out*	*in*	*out*	*in*	*out*
Optimal instances (#)	6/44	7/44	12/44	18/44	27/44	21/44	5/44	5/44
Infeasible instances (#)	4/44	4/44	13/44	11/44	6/44	6/44	0/44	0/44
Arti. mean of gap (%)	13.39	9.53	8.96	3.64	3.15	3.56	10.07	10.56
Geo. mean of gap (%)	8.81	6.03	3.57	1.45	0.86	1.24	6.92	7.02

From Table 4.6, we note that solving CDSPHT-S as the relaxed problem in the approximate algorithm outperforms the other relaxations in terms of the final optimality gap and the number of optimal instances. According to the obtained results, we also note that for the large-size instances involving 50 incoming/outgoing trucks, the two independent scheduling problems solved by MIP-OT are the only relaxations that can provide a feasible solution with a given optimality gap.

To have a better comparison of the relaxed problems, we have computed the %gap for the best upper bounds obtained by any of the formulations within 24 hours, or any setting of the approximate algorithm presented in Table 4.6 with respect to the lower bounds obtained by each of the relaxation problems, as well as the LP bounds obtained with F1-G and F2-G. Table 4.7, summarizes the obtained results. For each of the relaxation problems the number of instances for which the optimality of the best bound is proved, the number of instances for which the best lower bound is obtained, and the arithmetic and geometric means of the optimality gap are reported. According to this table, F1-S significantly outperforms other problems in terms of the quality of the provided lower bound and also the quality of the obtained % gap, by proving optimality for 30 instances. For five of such instances the approximate algorithm is capable of proving optimality for the first time. These instances are highlighted in bold in the Table 10 in Appendix G, where the detail results are provided.

Table 4.7: A comparison of best-known solutions and different relaxations in approximate algorithm.

	F1-G			F2-G			F1-S	MIP-OT
	LP	*Relax X*	*Relax Y*	*LP*	*Relax X*	*Relax Y*		
Proven optimality (#)	8/44	10/44	8/44	8/44	18/44	13/44	30/44	10/44
Best LB (#)	8/44	11/44	8/44	8/44	19/44	13/44	35/44	17/44
Arit. mean of gap (%)	16.29	7.51	11.27	16.77	7.00	13.25	4.17	8.26
Geo. mean of gap (%)	9.74	4.00	6.77	9.92	2.50	5.62	0.94	4.92

From the results that we have obtained so far, in our approximate algorithm it is better to always use the BC algorithm for the CDSPHT-S as the main lower bounding procedure. We also solve two parallel machine scheduling problems, to potentially improve the lower bounds and to make sure that we always have a starting solution, specially for larger instances. We recall that at every iteration of the matheuristic, we need to solve a OCDSP-RD. In section 4.6, we have provided two formulations to solve it: MIP-OT and CP-OT. Each of them can be used within the approximate algorithm, and so two variants of the algorithm can be considered. In order to evaluate the overall performance of our approximate algorithm, we have compared the obtained results with the formulations for 40 instances with 8 to 20 trucks. For these experiments, we have considered a time limit of one hour for solving each of the relaxations and a time limit of 300 seconds for each of the formulations solved within the matheuristic. In the worst case, the approximate algorithm takes up to three hours to terminate. Therefore, to have a fair comparison, we have rerun the formulations with a time limit of three hours. We only applied the local search matheuristic (Algorithm 1) in the approximate algorithm in these experiments.

Table 4.8 summarizes the comparison between two variants of the proposed approximate algorithm and the three formulations for the CDSPHT-G. For each of these solution algorithms we report: the number of optimal instances, the number of instances for which no solution is found, the number of instances for which the best bound is obtained, arithmetic and geometric means of the % deviation from the best upper bound obtained using any of these five approaches , arithmetic and geometric means of the final % gap, and arithmetic and geometric means of CPU time. For the detail results we refer to Table 11 in Appendix G

Table 4.8: A comparison between approximate algorithms, MIP and CP formulations for the CDSPHT-G.

	Approximate algorithm		Formulations		
	CP-OT	MIP-OT	F1-G	F2-G	CP-G
Optimal instances (#)	27/40	25/40	16/40	17/40	22/40
Infeasible instances (#)	0/40	0/40	13/40	6/40	2/40
Best UB (#)	36/40	34/40	21/40	27/40	31/40
Arit. mean of dev (%)	0.23	0.30	0.36	0.30	0.20
Geo. mean of dev (%)	0.12	0.17	0.23	0.20	0.14
Arit. mean of gap (%)	3.60	3.67	3.72	6.06	8.89
Geo. mean of gap (%)	1.06	1.16	1.47	2.26	2.47
Arit. mean of time (sec)	1,567	1,321	6,649	6,322	5,304
Geo. mean of time (sec)	102	107	506	466	315

According to Table 4.8, both variants of the approximate algorithm significantly outperform the formulations in terms of the CPU time and number of obtained optimal (or feasible) solutions. Among the two variants, the variant that uses CP-OT has a slightly better performance. It is worth mentioning that using these approximate algorithms, we were able to prove the optimality for four instances (12x6x35, 12x6x50, 15x7x35, and 20x10x50) in less than two hours, whereas all formulations fail to do so in 24 hours. We also improved the best-known solution for one of the instances (15x7x50) with respect to the upper bound obtained by all formulations in 24 hours. Finally, we also found a feasible solution for one of the instances (15x6x75) for which none of the formulations could find a solution in 24 hours.

In order to examine the impact of using the more sophisticated ILS matheuristic, we have solved 40 instances with two variants of the approximate algorithm using the ILS matheuristic with 10 iterations. Since we apply the matheuristic twice, once by fixing inbound and once by fixing outbound decisions, in total the matheuristic is applied 20 times within each variant of the algorithm. The results are summarized in Table 4.9. We note that for calculating the number of instances with the best bound and the % deviations we have considered the best-known upper bound that we have found for each instance using any of the formulations or solution algorithms. Detail of the results can be found in Table 12 in Appendix G.

As can be seen from Table 4.9, using the ILS matheuristic significantly increases the CPU time, while having only a marginal impact on the quality of the obtained upper bounds. We also note that for the ILS matheuristic, MIP-OT outperforms CP-OT. A positive outcome of the use of the ILS matheuristic within the approximate algorithm is that the optimality of one additional instance (15x6x25) was proved for the first time and the upper bounds for two instances (15x6x50 and 15x6x75) were improved.

Table 4.9: Impact of using ILS matheuristic on the performance of approximate algorithm.

	CP-OT		MIP-OT	
	LS	ILS	LS	ILS
Optimal instances (#)	27/40	28/40	25/40	29/40
Infeasible instances (#)	0/40	0/40	0/40	0/40
Best UB (#)	33/40	37/40	32/40	39/40
Arit. mean of dev (%)	0.32	0.16	0.39	0.09
Geo. mean of dev (%)	0.19	0.08	0.23	0.04
Arit. mean of gap (%)	3.60	3.46	3.67	3.40
Geo. mean of gap (%)	1.06	1.00	1.16	0.94
Arit. mean of time (sec)	1,567	5,280	1,321	2,813
Geo. mean of time (sec)	102	303	107	387

Considering an easy to solve relaxation problem within the proposed approximate algorithm enables us to obtain feasible solutions with a given optimality gap in just a few hours, even for the large-size instances for which none of the formulations could find a single feasible solution within 24 hours. The results obtained for the large-size instances using the LS and ILS matheuristic within the approximate algorithm are provided in Table 4.10. These results are obtained using only CP-OT. In the case of the MIP-OT, we noted that given the CPU limit of 300 seconds set to solve MIP-OT, no solutions were found and the algorithm was interrupted.

Table 4.10: Detailed results of approximate algorithm for large-size CDSPHT-G instances.

Instance	LS			ILS		
	Time	*UB*	*Gap(%)*	*Time*	*UB*	*Gap(%)*
50x30x25	1,145	150	0.00	11,177	150	0.00
50x30x35	7,663	197	8.63	74,686	195	7.69
50x30x50	11,514	290	9.66	108,479	289	9.34
50x30x75	17,694	440	10.68	162,747	440	10.68

For a better comparison between the proposed approximate algorithm and the formulations, the summary of the results obtained by the formulations within 24 hours and different variants of the approximate algorithm for all instances are reported in Table 4.11. According to this table, approximate algorithms with LS matheuristic have a very good performance as compared to formulations, given that they can prove the optimality of more instances and find high quality solutions for other instances in just a fraction of time needed by the formulations. Moreover, approximate algorithms with ILS matheuristic outperform the formulations significantly in terms of time, quality of solutions, and the number of instances solved optimally.

Finally, we compare the solutions obtained by our approximate algorithms with the solutions obtained by the two state-of-the-art metaheuristics given in [59]. Table 4.12 represents a summary of the results. In these experiments, to compute the % deviations, we have considered the best bound obtained so far, by one of the formulations, the approximate algorithms, or the heuristics from [59]. The best solution for instance 20x10x75 was obtained by one of the heuristics from [59]. For all the other instances, the best solutions were found by at least one of the algorithms reported in

Table 4.11: A comparison of approximate algorithms and MIP and CP formulations for the CDSPHT-G using a time limit of 24 hours.

	Formulations			Approximate algorithms			
				CP-OT		MIP-OT	
	F1-G	F2-G	CP-G1	*LS*	*ILS*	*LS*	*ILS*
Optimal instances (#)	19/44	19/44	24/44	28/44	29/44	25/44	29/44
Infeasible instances (#)	10/44	6/44	4/44	0/44	0/44	4/44	4/44
Best UB (#)	30/44	29/44	38/44	35/44	41/44	32/44	39/44
Arit. mean of Dev (%)	0.14	0.27	0.12	0.32	0.14	0.39	0.09
Geo. mean of Dev (%)	0.09	0.20	0.06	0.20	0.08	0.23	0.04
Arit. mean of Gap (%)	2.65	4.96	6.93	3.93	3.77	3.67	3.40
Geo. mean of Gap (%)	1.15	2.02	2.04	1.27	1.20	1.16	0.94
Arit. mean of time (sec)	50,109	49,433	37,008	2,289	12,915	2,183	3,539
Geo. mean of time (sec)	2,183	1,922	642	149	491	163	523

the current paper. The first two rows of this table, report, respectively, the number of instances for which the best bound and the optimal bound are obtained. Next rows present the arthimetic and geometric means of the % deviations for the best bound obtained by each of the four variants of the proposed approximate algorithm, and for the average, maximum, and minimum bounds obtained by each of the heuristics. According to this table, our proposed approximate algorithms outperform the metaheuristics of [59] in terms of the quality of the obtained solutions. For the detail results we refer to Table 13 in Appendix G.

Table 4.12: A comparison of approximate algorithms and state-of-the-art metaheuristics.

	Approximate algorithms				Sayed et al. [59]					
	CP-OT		MIP-OT		ILS-VND			GRASP-VND		
	LS	*ILS*	*LS*	*ILS*	*Avg.*	*max*	*min*	*Avg.*	*max*	*min*
Best Bound (#)	34/44	40/44	31/44	38/44	12/44	12/44	31/44	8/44	8/44	26/44
Optimal Bound (#)	28/44	29/44	25/44	30/44	12/44	12/44	27/44	8/44	8/44	24/44
Arit. Mean of dev (%)	0.33	0.16	0.40	0.11	1.33	2.28	0.42	1.88	2.68	0.60
Geo. Mean of dev (%)	0.21	0.09	0.25	0.05	1.01	1.68	0.28	1.46	2.10	0.42

4.8.4 Impact of considering door-dependent transfer times

In this paper we have studied two variants of the CDSPHT. In the first variant, we assumed that unit-load transfer time depends on the origin and destination doors, while in the second variant, we relaxed the assumption of being door-dependent. We are now interested to see how this relaxed assumption can impact the estimated makespan and obtained solutions. In other words, if we solve the problem by assuming that unit-load transfer time does not depend on the doors, and then we use the obtained assignment and sequences to schedule trucks in an environment with door-dependant unit-load transfer time, what is the percentage of increment in the makespan, with respect to the case were we directly solve the CDSPHT-G. To do so, we compared the upper bounds obtained for the CDSPHT-G directly, with the ones that can be calculated based on the solution of the CDSPHT-S. We have also included the results for the case were we first solve a CDSP without considering handling times (i.e. transfer times are equal to zero), and then we calculate the makespan for the CDSPHT-G based on the obtained solution. For this comparison, we have chosen those instances that could be solved optimally for all variants of the problem using one of the proposed formulations or algorithms. Table 4.13 presents this comparison. In the second and third columns of this table we report the CPU time and the optimal value for the CDSPHT-G obtained by directly solving the problem. The CPU time reported for each of the instances in the second column, is the best time that was obtained among the provided formulations for the CDSPHT-G. For those instances that could not be optimally solved with the formulations, the best time among the variants of the approximate algorithm is reported. The next two columns denote the CPU time and the bound obtained for the general problem through solving the specific case problem using F1-S. In the fifth and sixth columns under the heading *T-Dec (%)* and *B-Inc (%)*, we report the percentage of decrease in CPU time and increase in the makespan, when CDSPHT-G is indirectly solved through CDSPHT-S.

The next four columns report the CPU time, makespan, % of decrease in time, and % of increase in the makespan when we solve F1-S considering transfer times to be zero, and then, based on the obtained solution, we calculate the makespan for the CDSPHT-G.

Table 4.13: Impact of solving CDSPHT-G directly and indirectly through CDSPHT-S and CDSP.

Instance	CDSPHT-G		CDSPHT-S				CDSP			
	Time	*UB*	*Time*	*UB*	*T-Dec(%)*	*B-Inc(%)*	*Time*	*UB*	*T-Dec(%)*	*B-Inc(%)*
8x4x25	0	27	0	31	0.00	14.81	0	33	0.00	22.22
8x4x35	1	43	1	48	0.00	11.63	0	52	100.00	20.93
8x4x50	12	59	3	64	75.00	8.47	1	61	91.67	3.39
8x4x75	165	75	48	76	70.91	1.33	49	83	70.30	10.67
9x4x25	0	33	0	36	0.00	9.09	0	35	0.00	6.06
9x4x35	0	48	0	52	0.00	8.33	0	53	0.00	10.42
9x4x50	45	54	4	59	91.11	9.26	6	62	86.67	14.81
10x4x25	0	34	0	35	0.00	2.94	0	38	0.00	11.76
10x4x35	13	51	3	56	76.92	9.80	1	55	92.31	7.84
10x4x50	347	76	217	81	37.46	6.58	906	76	-161.10	0.00
10x5x25	0	35	0	38	0.00	8.57	0	37	0.00	5.71
10x5x35	1	44	0	48	100.00	9.09	0	54	100.00	22.73
10x5x50	2	69	0	71	100.00	2.90	0	74	100.00	7.25
10x5x75	2,523	95	347	99	86.25	4.21	2,082	100	17.48	5.26
11x5x25	4	36	1	39	75.00	8.33	0	44	100.00	22.22
11x5x35	68	53	2	54	97.06	1.89	1	58	98.53	9.43
11x5x50	1,849	74	220	78	88.10	5.41	183	78	90.10	5.41
12x5x25	263	46	11	51	95.82	10.87	3	54	98.86	17.39
12x5x35	6,315	63	341	68	94.60	7.94	1,470	67	76.72	6.35
12x6x25	0	41	0	45	0.00	9.76	0	44	0.00	7.32
12x6x35	143	52	136	56	4.90	7.69	68	59	52.45	13.46
12x6x50	1,438	76	1,423	79	1.04	3.95	1,181	82	17.87	7.89
15x6x25	410	55	342	59	16.59	7.27	202	62	50.73	12.73
15x6x35	6,542	84	199	87	96.96	3.57	12	92	99.82	9.52
15x7x25	4	46	6	48	-50.00	4.35	1	49	75.00	6.52
15x7x35	3,657	71	3,579	73	2.13	2.82	789	75	78.42	5.63
20x10x25	27,277	63	55	70	99.80	11.11	19	71	99.93	12.70
20x10x35	2,776	90	27	91	99.03	1.11	14	94	99.50	4.44
20x10x50	1,589	117	954	118	39.96	0.85	1,981	121	-24.67	3.42
Arit. mean	1,912		273		48.23	6.69	309		52.09	10.12
Geo. mean	63		17			5.41	14			8.40

According to Table 4.13, solving the CDSPHT-G indirectly through solving CDSPHT-S reduces on average the CPU time by 48%, whereas it increases on average the makespan by 7%. However, looking into the instances, we observe that the makespan increases up to 15% when the dependency of the transfer times are not considered. In the case of solving the problem without considering the handling times, the CPU time decreases on average by 52%, at the expense of increasing the makespan about 10%, while for some instances this increase can be up to 23%. These results reveal that transfer times have a significant impact on the makespan and considering them while solving the problem is important for obtaining more efficient schedules. We expect that this impact will be more significant for cross-docks in which distances between doors differ significantly.

4.9 Conclusion

In this paper we addressed two variants of a cross-dock scheduling problem with handling times: a general case with door-dependent unit-load transfer time and a particular case with constant unit-load transfer time. For each of these problems, a constraint programming (CP) formulation was developed. By comparing our CP formulation for the general problem with two mathematical programming formulations that have been proposed in the literature, we showed that CP outperforms those MIP formulations in terms of both the quality of the solution and the CPU time when solved with CPLEX. We also developed several families of valid inequalities for the specific case problem and described how can these be adapted for the general one. The proposed inequalities were used within a branch-and-cut framework to improve the performance of an MIP formulation. We also designed approximate algorithms that combine integer relaxations with a matheuristic to compute optimality gaps for the general case. Finally, we analyzed the impact of incorporating handling times in the model on the quality of the obtained makespan. We compared three ways to calculate makespan for the general problem: solving general problem directly, solving general problem through the specific case, and solving the problem through the specific case where all transfer times are zero. The results suggest that considering true values of transferring times can significantly improve the makespan and thus, it is important for obtaining efficient schedules.

Future research directions include the evaluation of the impact of the valid inequalities when used within a BC algorithm for the general case problem. Incorporating the storage capacity within the cross-dock facility and integrating employee time tabling in to the current models are other promising future research directions.

Chapter 5

Conclusions

This thesis studied two important applications of scheduling in service industries: outpatient appointment scheduling in oncology clinics under uncertain arrival times of appointment requests and uncertain service times, and cross-dock scheduling while considering loading, transfer and unloading times of products.

HiIn Chapter 2, we studied a multi-appointment, multi-stage chemotherapy scheduling problem considering unique characteristics and realistic assumptions. We presented two alternative integer programming formulations, that were compared computationally. We used the most promising model to develop integrated and sequential scheduling strategies and an online scheduling algorithm to accommodate arriving requests dynamically. We performed several computational experiments to evaluate the performance of the proposed algorithms using historical data gathered from a major cancer center in Canada. The results revealed that the proposed algorithms can potentially make significant improvements in the clinic schedule with respect to several performance measures.

In Chapter 3 we addressed an integrated daily consultation and chemotherapy scheduling problem considering stochastic treatment times and different patient types. We developed two two-stage stochastic programming models for this problem. According to the conducted computational experiments, one of the proposed formulations that partially models the problem as Multi-TSP, outperforms the other formulation significantly. We also presented a SAA algorithm to solve the stochastic problem and to compute statistical lower-bound, upper-bound, and optimality gap. A specialized algorithm was also designed to quickly evaluate a given first-stage solution for a large number of scenarios. We assessed the quality of the solutions obtained by the SAA algorithm with the solutions of the expected value problem. The results of the experiments showed that, comparing to the expected value problem, the SAA algorithm is able to reduce the expected objective value by at least 95 percent.

Finally, in Chapter 4 we presented two variants of a cross-dock scheduling problem with handling times: a general case with door-dependent unit-load transfer times and a specific case with constant unit-load transfer times. We developed constraint programming formulations for each variant of the problem. We carried out several computational experiments to evaluate the performance of the proposed CP for the general problem in comparison with two mixed integer programming models from the literature. The results verified that the CP outperforms the MIP formulations in terms of both the quality of the solution and the CPU time. We also developed several families of valid inequalities for the specific case problem that were used within a branch-and-cut framework to improve the performance of a time-index formulation. Conducted computational experiments revealed that using the best combination of valid inequalities within a BC algorithm can significantly improve the performance of the formulation. We also designed and computationally evaluated several approximate algorithms that combine a relaxation problem with a matheuristic to compute bounds for the general problem and also to provide optimality gaps. The results showed that these algorithms outperform the formulations and other developed heuristics in the literature in terms of the quality of solutions.

There are several possibilities to extend the ideas presented in this thesis. In the following several directions for future researches are provided.

- Conducting simulation-based studies to determine the best time for scheduling appointments in an online appointment scheduling system.
- Considering acuity level of patients to compute nurses' daily workload for balancing purposes.
- Considering treatment cancellations in the stochastic chemotherapy scheduling problem.
- Designing decomposition-based algorithms to efficiently solve SAA problems.
- Developing a branch-and-cut algorithm for the general variant of the studied cross-dock scheduling problem.
- Considering a limited storage capacity within the cross-dock facility.

Bibliography

[1] 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.

[2] 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.

[3] A. Amini, R. Tavakkoli-Moghaddam, and A. Omidvar. Cross-docking truck scheduling with the arrival times for inbound trucks and the learning effect for unloading/loading processes. *Production & Manufacturing Research*, 2(1):784–804, 2014.

[4] K. Apt. *Principles of constraint programming.* Cambridge university press, 2003.

[5] P. Baptiste, C. Le Pape, and W. Nuijten. *Constraint-based scheduling: applying constraint programming to scheduling problems*, volume 39. Springer Science & Business Media, 2012.

[6] T. Bektas. The multiple traveling salesman problem: an overview of formulations and solution procedures. *omega*, 34(3):209–219, 2006.

[7] M. Benzaid, N. Lahrichi, and L.-M. Rousseau. Chemotherapy appointment

scheduling and daily outpatient–nurse assignment. *Health care management science*, pages 1–17, 2019.

[8] P. Bodnar, R. de Koster, and K. Azadeh. Scheduling trucks in a cross-dock with mixed service mode dock doors. *Transportation Science*, 51(1):112–131, 2017.

[9] A. Bouras, M. Masmoudi, N. E. H. Saadani, and Z. Bahroun. A three-stage appointment scheduling for an outpatient chemotherapy unit using integer programming. In *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 0916–0921. IEEE, 2017.

[10] N. Boysen and M. Fliedner. Cross dock scheduling: Classification, literature review and research agenda. *Omega*, 38(6):413–422, 2010.

[11] N. Boysen, M. Fliedner, and A. Scholl. Scheduling inbound and outbound trucks at cross docking terminals. *OR spectrum*, 32(1):135–161, 2010.

[12] P. Buijs, I. F. Vis, and H. J. Carlo. Synchronization in cross-docking networks: A research classification and framework. *European Journal of Operational Research*, 239(3):593–608, 2014.

[13] 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.

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

[15] I. A. Chaudhry and A. A. Khan. A research survey: review of flexible job shop scheduling techniques. *International Transactions in Operational Research*, 23 (3):551–591, 2016.

[16] F. Chen and C.-Y. Lee. Minimizing the makespan in a two-machine cross-docking flow shop problem. *European Journal of Operational Research*, 193(1):59–72, 2009.

[17] F. Chen and K. Song. Minimizing makespan in two-stage hybrid cross docking scheduling problem. *Computers & Operations Research*, 36(6):2066–2073, 2009.

[18] 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.

[19] P. M. Cota, B. M. Gimenez, D. P. Araújo, T. H. Nogueira, M. C. de Souza, and M. G. Ravetti. Time-indexed formulation and polynomial time heuristic for a multi-dock truck scheduling problem in a cross-docking centre. *Computers & Industrial Engineering*, 95:135–143, 2016.

[20] R. Dobish. Next-day chemotherapy scheduling: a multidisciplinary approach to solving workload issues in a tertiary oncology center. *Journal of Oncology Pharmacy Practice*, 9(1):37–42, 2003.

[21] G. B. Fonseca, T. H. Nogueira, and M. G. Ravetti. A hybrid lagrangian metaheuristic for the cross-docking flow shop scheduling problem. *European Journal of Operational Research*, 275(1):139–154, 2019.

[22] T. Garaix, S. Rostami, and X. Xie. Daily outpatient chemotherapy appointment scheduling with random deferrals. *Flexible Services and Manufacturing Journal*, pages 1–25, 2019.

[23] M. Gaudioso, M. F. Monaco, and M. Sammarra. A lagrangian heuristics for the truck scheduling problem in multi-door, multi-product cross-docking with constant processing time. *Omega*, page 102255, 2020.

[24] S. Gelareh, F. Glover, O. Guemri, S. Hanafi, P. Nduwayo, and R. Todosijević. A comparative study of formulations for a cross-dock door assignment problem. *Omega*, 91:102015, 2020.

[25] Y. Göçgün. Dynamic scheduling with cancellations: an application to chemotherapy appointment booking. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 8(2):161–169, 2018.

[26] Y. Göçgün. Dynamic scheduling with cancellations: an application to chemotherapy appointment booking. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 8(2):161–169, 2018.

[27] 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.

[28] M. Guignard, P. M. Hahn, A. A. Pessoa, and D. C. da Silva. Algorithms for the cross-dock door assignment problem. In *Proceedings of the fourth international workshop on model-based metaheuristics*, pages 145–162, 2012.

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

[30] M. Haghi and I. Contreras. Formulations and algorithms for cross-dock scheduling problems with handling times. *submitted to Transportation Research Part B: Methodological*, 2020.

[31] M. Haghi, I. Contreras, N. Bhuiyan, and G. Batist. Multi-appointment, multi-stage outpatient scheduling at oncology clinics. *submitted to Omega*, 2020.

[32] S. Hahn-Goldberg, M. W. Carter, J. C. Beck, M. Trudeau, P. Sousa, and K. Beat-

tie. Dynamic optimization of chemotherapy outpatient scheduling with uncertainty. *Health care management science*, 17(4):379–392, 2014.

[33] F. Heidari, S. H. Zegordi, and R. Tavakkoli-Moghaddam. Modeling truck scheduling problem at a cross-dock facility through a bi-objective bi-level optimization approach. *Journal of Intelligent Manufacturing*, 29(5):1155–1170, 2018.

[34] D. Hermel, H. Hasheminia, N. Adler, and M. J. Fry. A solution framework for the multi-mode resource-constrained cross-dock scheduling problem. *Omega*, 59: 157–170, 2016.

[35] A. F. Hesaraki, N. P. Dellaert, and T. de Kok. Generating outpatient chemotherapy appointment templates with balanced flowtime and makespan. *European Journal of Operational Research*, 275(1):304–318, 2019.

[36] M. Heshmat and A. Eltawil. Comparison between outpatient appointment scheduling and chemotherapy outpatient appointment scheduling. *The Egyptian International Journal of Engineering Sciences & Technology*, 19(2):326–332, 2016.

[37] 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.

[38] 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*, 150:113267, 2020.

[39] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[40] A.-L. Ladier and G. Alpan. Cross-docking operations: Current research versus industry practice. *Omega*, 62:145–162, 2016.

[41] 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.

[42] P. K. Lau, M. J. Watson, and A. Hasani. Patients prefer chemotherapy on the same day as their medical oncology outpatient appointment. *Journal of oncology practice*, 10(6):e380–e384, 2014.

[43] A. G. Leeftink, I. A. Bikker, I. Vliegen, and R. Boucherie. Multi-disciplinary planning in health care: a review. *Health Systems*, pages 1–24, 2018.

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

[45] 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.

[46] J. Linderoth, A. Shapiro, and S. Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142(1): 215–241, 2006.

[47] H. R. Lourenço, O. C. Martin, and T. Stützle. Iterated local search: Framework and applications. In *Handbook of Metaheuristics*, pages 129–168. Springer, 2019.

[48] W.-K. Mak, D. P. Morton, and R. K. Wood. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations research letters*, 24(1-2):47–56, 1999.

[49] J. Marynissen and E. Demeulemeester. Literature review on multi-appointment scheduling problems in hospitals. *European Journal of Operational Research*, 272 (2):407–419, 2019.

[50] W. Nassief, I. Contreras, and R. As' ad. A mixed-integer programming formulation and lagrangean relaxation for the cross-dock door assignment problem. *International Journal of Production Research*, 54(2):494–508, 2016.

[51] W. Nassief, I. Contreras, and B. Jaumard. A comparison of formulations and relaxations for cross-dock door assignment problems. *Computers & Operations Research*, 94:76–88, 2018.

[52] M. Pinedo, C. Zacharias, and N. Zhu. Scheduling in the service industries: An overview. *Journal of systems science and systems engineering*, 24(1):1–48, 2015.

[53] T. Qin, Y. Du, J. H. Chen, and M. Sha. Combining mixed integer programming and constraint programming to solve the integrated scheduling problem of container handling operations of a single vessel. *European Journal of Operational Research*, 2020.

[54] 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.

[55] A. Rijal, M. Bijvank, and R. de Koster. Integrated scheduling and assignment of trucks at unit-load cross-dock terminals with mixed service mode dock doors. *European Journal of Operational Research*, 278(3):752–771, 2019.

[56] R. W. Saaty. The analytic hierarchy process—what it is and how it is used. *Mathematical modelling*, 9(3-5):161–176, 1987.

[57] 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.

[58] A. Sadki, X. Xie, and F. Chauvin. Planning oncologists of ambulatory care units. *Decision Support Systems*, 55(2):640–649, 2013.

[59] S. I. Sayed, I. Contreras, J. A. Diaz, and D. E. Luna. Integrated cross-dock door assignment and truck scheduling with handling times. *TOP*, pages 1–23, 2020.

[60] M. Shakeri, M. Y. H. Low, S. J. Turner, and E. W. Lee. A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock. *Computers & Operations Research*, 39(11):2564–2577, 2012.

[61] A. Shapiro and T. Homem-de Mello. A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming*, 81(3):301–325, 1998.

[62] S. Suss, N. Bhuiyan, K. Demirli, and G. Batist. Toward implementing patient flow in a cancer treatment center to reduce patient waiting time and improve efficiency. *Journal of oncology practice*, 13(6):e530–e537, 2017.

[63] 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.

[64] O. Theophilus, M. A. Dulebenets, J. Pasha, O. F. Abioye, and M. Kavoosi. Truck scheduling at cross-docking terminals: A follow-up state-of-the-art review. *Sustainability*, 11(19):5245, 2019.

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

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

[66] O. Unsal and C. Oguz. An exact algorithm for integrated planning of operations in dry bulk terminals. *Transportation Research Part E: Logistics and Transportation Review*, 126:103–121, 2019.

[67] J. Van Belle, P. Valckenaers, and D. Cattrysse. Cross-docking: State of the art. *Omega*, 40(6):827–846, 2012.

[68] H. Wang and B. Alidaee. The multi-floor cross-dock door assignment problem: Rising challenges for the new trend in logistics industry. *Transportation Research Part E: Logistics and Transportation Review*, 132:30–47, 2019.

[69] W. Wisittipanich and P. Hengmeechai. Truck scheduling in multi-door cross docking terminal by modified particle swarm optimization. *Computers & Industrial Engineering*, 113:793–802, 2017.

[70] Y. Ye, J. Li, K. Li, and H. Fu. Cross-docking truck scheduling with product unloading/loading constraints based on an improved particle swarm optimisation algorithm. *International Journal of Production Research*, 56(16):5365–5385, 2018.

Appendices

A Detailed Description of the Multi-Objective Function

In this section we describe in detail the objectives defined in Section 2.4.2. Since most of these objectives are non-linear in nature, their linearization is also provided using auxiliary variables defined in Table 1.

Table 1: Auxiliary variables used to linearize the objective function terms

Variables	***Definition***
$bl_d \geq 0$	maximum workload difference between all pairs of nurses on day $d \in \mathcal{D}$
$co_{od} \geq 0$	completion time of oncologist $o \in \mathcal{O}^{\mathcal{G}}$ on day $d \in \mathcal{D}_o^O$
$ot_{od}^O \geq 0$	overtime of oncologist $o \in \mathcal{O} \setminus \mathcal{O}^{\mathcal{G}}$ on day $d \in \mathcal{D}_o^O$
$ot_{nd}^N \geq 0$	overtime of nurse $n \in \mathcal{N}$ on day $d \in \mathcal{D}_n^N$
$wt_{pk}^1 \geq 0$	first type waiting time of patient $p \in \mathcal{P}$ on appointment $k \in \mathcal{CA}_p$
$wt_{pk}^2 \geq 0$	second type waiting time of patient $p \in \mathcal{P}$ on appointment $k \in \mathcal{TA}_p$
$wt_{pk}^3 \geq 0$	third type waiting time of patient $p \in \mathcal{P}$ on appointment $k \in \mathcal{TA}_p$

The first objective balances the nurses daily workload and can be formulated as:

$$
\begin{aligned}
g_1(x) \quad = \quad & \sum_{d \in \mathcal{D}} \max_{\substack{n,n' \in \mathcal{N}^{g_1} \\ n > n'}} \Bigg\{ \Bigg| L_{nd} + \sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{K}_p} \sum_{s \in \mathcal{S}_{pkd}^5} \sum_{c \in \mathcal{C}_{pk}^P \cap \mathcal{C}_{nd}^N} PT_{pk}^5 x_{pkdsc} \\
& - L_{n'd} - \sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{K}_p} \sum_{s \in \mathcal{S}_{pkd}^5} \sum_{c \in \mathcal{C}_{pk}^P \cap \mathcal{C}_{n'd}^N} PT_{pk}^5 x_{pkdsc} \Bigg| \Bigg\},
\end{aligned}
$$

where for day $d \in \mathcal{D}$, $\mathcal{N}^{g_1} = \{n \in \mathcal{N} : d \in \mathcal{D}_n^N\}$, denotes the set of nurses working on that day and $\mathcal{K}_p = \{k \in \mathcal{TA}_p : d \in \mathcal{D}_{pk}^5\}$, represents the set of treatment appointments of patient p that are allowed to be scheduled on day d. Moreover, L_{nd} and $L_{n'd}$ denote the initial load of nurses n and n', respectively, on day d, due to the partial schedule in the dynamic variant. For the static variant, initial load of each nurse is set to zero.

In order to linearize $g_1(x)$, we consider $g_1(x) = \sum_d bl_d$, and for $d \in \mathcal{D}$ and $n, n' \in$

$\mathcal{N}^{g_1}$ such that $n > n'$, we define the following constraints:

$$\begin{aligned} bl_d \geq L_{nd} &+ \sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{K}_p} \sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^{g_1}_{ndpk}} PT^5_{pk} x_{pkdsc} \\ &- L_{n'd} - \sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{K}_p} \sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^{g_1}_{n'dpk}} PT^5_{pk} x_{pkdsc} \end{aligned} \tag{1}$$

$$\begin{aligned} bl_d \geq L_{n'd} &+ \sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{K}_p} \sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^{g_1}_{n'dpk}} PT^5_{pk} x_{pkdsc} \\ &- L_{nd} - \sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{K}_p} \sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^{g_1}_{ndpk}} PT^5_{pk} x_{pkdsc}. \end{aligned} \tag{2}$$

The second objective minimizes nurses overtime (patients hand overs) and can be formulated as:

$$g_2(x) = \sum_{n \in \mathcal{N}} \sum_{d \in \mathcal{D}^N_n} \max_{p \in \mathcal{NP} \cup \mathcal{RP}} \left\{ \left(\sum_{k \in \mathcal{K}_p} \sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^P_{pk} \cap \mathcal{C}^N_{nd}} \left(s + PT^5_{pk}\right) x_{pkdsc} - FN_{nd} \right)^+ \right\},$$

where FN_{nd} denotes end of the regular working hours of nurse n on day d.

In order to linearize $g_2(x)$, we consider $g_2(x) = \sum_n \sum_d ot^N_{nd}$, and for $n \in \mathcal{N}$ and $d \in \mathcal{D}^N_n$, we define the following constraint:

$$ot^N_{nd} \geq \sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{K}_p} \sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^P_{pk} \cap \mathcal{C}^N_{nd}} \left(s + PT^5_{pk}\right) x_{pkdsc} - FN_{nd}. \tag{3}$$

The third objective consists of three terms that can be modeled as:

$$g^1_3(y^3) = \sum_{o \in \mathcal{O}^{\mathcal{G}}} \sum_{d \in \mathcal{D}^O_o} \max_{p \in \mathcal{P}^O_o} \left\{ \sum_{k' \in \mathcal{K}'_p} \sum_{s \in \mathcal{S}^3_{pk'd}} \left(s + PT^3_{pk'}\right) y^3_{pk'ds} \right\}$$

$$g^2_3(y^3) = \sum_{o \in \mathcal{O} \setminus \mathcal{O}^{\mathcal{G}}} \sum_{d \in \mathcal{D}^O_o} \max_{p \in \mathcal{P}^O_o} \left\{ \left(\sum_{k' \in \mathcal{K}'_p} \sum_{s \in \mathcal{S}^3_{pk'd}} \left(s + PT^3_{pk'}\right) y^3_{pk'ds} - FO_{od} \right)^+ \right\}$$

$$g_3^3(y^3) = \sum_{o\in\mathcal{O}^{\mathcal{G}}} \sum_{p\in\mathcal{P}_o^O} \sum_{k'\in\mathcal{CA}_p} \sum_{d\in\mathcal{D}_o^O\cap\mathcal{D}_{pk'}^3} \sum_{s\in\mathcal{S}_{pk'd}^3\cap\mathcal{S}_{od}^B} y_{pk'ds}^3,$$

where for day d and patient p, $\mathcal{K}'_p = \{k' \in \mathcal{CA}_p : d \in \mathcal{D}_{pk'}^3\}$, represents the set of all patient's consultation appointments that can be scheduled on day d. Furthermore, FO_{od} denotes the end of regular working hours of oncologist o on day d and $\mathcal{S}_{od}^B$ represents the set of time slots corresponding to the break period of oncologist o on day d. Moreover, $\mathcal{O}^{\mathcal{G}}$ denote the set of gynecology oncologists. We note that $(X)^+ = max\{X, 0\}$.

In order to linearize $g_3^1(y^3)$, we consider $g_3^1(y^3) = \sum_o \sum_d co_{od}$, and for $o \in \mathcal{O}^{\mathcal{G}}$ and $d \in \mathcal{D}_o^O$, we define the following constraint:

$$co_{od} \geq \sum_{p\in\mathcal{P}_o^O} \sum_{k'\in\mathcal{K}'_p} \sum_{s\in\mathcal{S}_{pk'd}^3} \left(s + PT_{pk'}^3\right) y_{pk'ds}^3. \tag{4}$$

In order to linearize $g_3^2(y^3)$, we consider $g_3^2(y^3) = \sum_o \sum_d ot_{od}^O$, and for $o \in \mathcal{O} \setminus \mathcal{O}^{\mathcal{G}}$ and $d \in \mathcal{D}_o^O$, we define the following constraint:

$$ot_{od}^o \geq \sum_{p\in\mathcal{P}_o^O} \sum_{k'\in\mathcal{K}'_p} \sum_{s\in\mathcal{S}_{pk'd}^3} \left(s + PT_{pk'}^3\right) y_{pk'ds}^3 - FO_{od}. \tag{5}$$

The fourth objective consists of three terms. The first term can be stated as:

$$g_4^1(y^1, y^3) = \sum_{p\in\mathcal{P}} \sum_{k'\in\mathcal{K}_p^1} \sum_{d\in\mathcal{D}_{pk'}^3\cap\mathcal{D}_{pk'}^1} \left(\sum_{s\in\mathcal{S}_{pk'd}^3} s y_{pk'ds}^3 - \sum_{s\in\mathcal{S}_{pk'}^1} \left(s + PT_{pk'}^1\right) y_{pk'ds}^1 \right)^+,$$

where $\mathcal{K}_p^1 = \{k' \in \mathcal{CA}_p : BC_{pk'} = 1\}$, represents consultation appointments of patient p that require blood test. In order to linearize this objective, we consider $g_4^1(y^1, y^3) =$

$\sum_p \sum_{k'} wt^1_{pk'}$, and for $p \in \mathcal{P}$ and $k' \in \mathcal{K}^1_p$, we define the following constraint:

$$wt^1_{pk'} \geq \sum_{d \in \mathcal{D}^3_{pk'} \cap \mathcal{D}^1_{pk'}} \sum_{s \in \mathcal{S}^3_{pk'd}} s y^3_{pk'ds} - \sum_{d \in \mathcal{D}^3_{pk'} \cap \mathcal{D}^1_{pk'}} \sum_{s \in \mathcal{S}^1_{pk'}} \left(s + PT^1_{pk'}\right) y^1_{pk'ds}. \tag{6}$$

The second term of the fourth objective can be stated as:

$$\begin{aligned} g^2_4(x, y^2) \;\; = \;\; & \sum_{p \in \mathcal{RP}} \sum_{k \in \mathcal{K}^2_p} \sum_{d \in \mathcal{D}^5_{pk} \cap \mathcal{D}^2_{p(k+1)}} \left(\sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^P_{pk}} s x_{pkdsc} - \right. \\ & \left. \sum_{s \in \mathcal{S}^2_{p(k+1)d}} \left(s + PT^2_{p(k+1)}\right) y^2_{p(k+1)ds} - |\mathcal{S}| z^2_{p(k+1)} \right)^+ , \end{aligned}$$

where $\mathcal{K}^2_p = \{k \in \mathcal{TA}_p : BT_{p(k+1)} = 1, d^5_{pk} = d^2_{p(k+1)}\}$. In order to linearize this objective term, we consider $g^2_4(x, y^2) = \sum_p \sum_k wt^2_{pk}$, and for $p \in \mathcal{RP}$ and $k \in \mathcal{K}^2_p$, we define the following constraint:

$$\begin{aligned} wt^2_{pk} \geq & \sum_{d \in \mathcal{D}^5_{pk} \cap \mathcal{D}^2_{p(k+1)}} \sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^P_{pk}} s x_{pkdsc} - \\ & \sum_{d \in \mathcal{D}^5_{pk} \cap \mathcal{D}^2_{p(k+1)}} \sum_{s \in \mathcal{S}^2_{p(k+1)d}} \sum_{c \in \mathcal{C}^P_{p(k+1)}} \left(s + PT^2_{p(k+1)}\right) y^2_{p(k+1)ds} - |\mathcal{S}| z^2_{p(k+1)}. \end{aligned} \tag{7}$$

The last term of the fourth objective can be modeled as:

$$g^3_4(x, y^2) = \sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{K}''_p} \sum_{d \in \mathcal{D}_{pk}} \left(\sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^P_{pk}} s x_{pkdsc} - \sum_{s \in \mathcal{S}^2_{pkd}} \left(s + PT^2_{pk}\right) y^2_{pkds} \right)^+ ,$$

where $\mathcal{K}''_p = \{k \in \mathcal{TA}_p : BT_{pk} = 1\}$, represents treatment appointments that require blood test as well and $\mathcal{D}_{pk} = \mathcal{D}^5_{pk} \cap \mathcal{D}^2_{pk} \cap \mathcal{D}^{\mathcal{H}}$.

In order to linearize this objective term, we consider $g^3_4(x, y^2) = \sum_p \sum_k wt^2_{pk}$, and for $p \in \mathcal{NP} \cup \mathcal{RP}$ and $k \in \mathcal{K}''_p$, we define the following constraint:

$$wt^3_{pk} \geq \sum_{d \in \mathcal{D}^{g_4^3}_{pk}} \sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^P_{pk}} s x_{pkdsc} - \sum_{d \in \mathcal{D}^{g_4^3}_{pk}} \sum_{s \in \mathcal{S}^2_{pkd}} \left(s + PT^2_{pk}\right) y^2_{pkds}. \tag{8}$$

The fifth objective minimizes the access time and can be stated as:

$$g_5(x) = \sum_{p \in \mathcal{NP}^{Pr}} \sum_{d \in \mathcal{D}^5_{p1}} \sum_{s \in \mathcal{S}^5_{p1d}} \sum_{c \in \mathcal{C}^P_{pk}} CP_p d x_{p1dsc},$$

where $\mathcal{NP}^{Pr}$, is set of new treatment patients with the highest priority and CP_p is the cost defining the priority grade of the patient.

For the consultation model of the proposed consultation-treatment approach described in Section 2.5, this objective can be alternatively expressed in terms of the y^3 variables as:

$$g'_5(y^3) = \sum_{p \in \mathcal{NP}'} \sum_{d \in \mathcal{D}^3_{p1}} \sum_{s \in \mathcal{S}^3_{p1d}} CP_p (d+1) y^3_{p1ds},$$

where $\mathcal{NP}' = \{p \in \mathcal{NP}^{Pr} : |\mathcal{CA}_p| = 1\}$.

The sixth objective maximizes the number of patients assigned to their primary or secondary nurses. For each primary nurse, there is a secondary nurse who is preferred to be assigned to the patient when the primary nurse is not available. This objective can be formulated as:

$$g_6(x) = \sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{TA}_p} \sum_{n \in \mathcal{N}} \sum_{d \in \mathcal{D}^5_{pk} \cap \mathcal{D}^N_n} \sum_{s \in \mathcal{S}^5_{pkd}} \sum_{c \in \mathcal{C}^N_{nd}} CN_{pn} x_{pkdsc},$$

where CN_{pn} is the cost of assigning patient p to the nurse n. For the primary nurse of each patient, the cost is zero, for the secondary nurse it takes a small value and for all other arbitrary nurse assignments, the cost will be higher. We note that by considering zero costs for any nurse-to-patient assignments, the model can easily consider a functional care delivery model rather than the primary model.

The seventh objective minimizes the number of non-preferred time assignments. Since chemotherapy treatments are usually given in cycles and the patients need to come to the clinic frequently for the infusion, sometimes it is difficult to have treatment appointments at certain times of the day. For example, patients may study in the morning and can only come to the clinic in the afternoons, or they may work in the evenings and only morning appointments work for them. Considering these types of preferences can improve patients' satisfaction and convenience. This objective can be formulated as:

$$g_7(x) = \sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{TA}_p} \sum_{d \in \mathcal{D}^5_{pk}} \sum_{s \in \mathcal{S}^5_{pkd} \setminus \mathcal{S}^P_{pk}} \sum_{c \in \mathcal{C}^P_{pk}} CT_{pk} x_{pkdsc},$$

where $\mathcal{S}^P_{pk}$ is the set of all preferred time slots for the treatment appointment k of patient p and CT_{pk} is the cost of non-preferred time assignments. The cost is assumed to depend on both the patient and appointment, because for the appointments that patient has more serious restrictions on the time, by imposing a higher cost, it is more likely possible to respect such restrictions.

The last objective function minimizes the the number of unscheduled patients and can be stated as:

$$g_8(z^1, z^2) = \sum_{p \in \mathcal{FP} \cup \mathcal{RP}} \sum_{k' \in \mathcal{CA}_p} CB^C_p z^1_{pk'} + \sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{TA}_p} CB^T_p z^2_{pk},$$

where CB^C_p and CB^T_p are the costs associated with nonscheduled consultation and treatment appointments of patient p, respectively. We note that by defining different costs for different patients, we can determine patients priorities for being scheduled.

For the consultation model of the proposed sequential approaches described in

Section 2.5, the eighth objective can be rewritten as:

$$g'_8(z^1) = \sum_{p \in \mathcal{P}} \sum_{k' \in \mathcal{CA}_p} CB_p^C z^1_{pk'}.$$

Similarly, for the treatment model of the sequential approaches, this objective can be stated as:

$$g''_8(z^2) = \sum_{p \in \mathcal{NP} \cup \mathcal{RP}} \sum_{k \in \mathcal{TA}_p} CB_p^T z^2_{pk}.$$

B Algorithms

Algorithm 7: Consultation-Treatment Approach

1: Initialization:

- Generate $\mathcal{P}^C$: set of patients with consultation appointments, and $\mathcal{P}^T$: set of patients with treatment appointments.
- Generate set of all consultation appointments: $\mathcal{AP}^C = \bigcup_{p \in \mathcal{P}^C} \mathcal{CA}_p$.
- Generate set of all treatment appointments: $\mathcal{AP}^T = \bigcup_{p \in \mathcal{P}^T} \mathcal{TA}_p$.

2: Solve *Consultation-Model* with $\mathcal{P}^C$ and $\mathcal{AP}^C$.

3: Update the number of available *Phlebotomist*.

4: Update set of appointments to be scheduled at the next step:

- For each unscheduled appointment of $\mathcal{AP}^C$, if $\mathcal{AP}^T$ includes an associated treatment appointment which would follow the consultation appointment, remove such appointment from $\mathcal{AP}^T$.

5: Update domain of next step variables:

- $p \in \mathcal{NP} \cap \mathcal{P}^C$: update set of possible days for drug preparation, $\mathcal{D}^4_{p1}$, to include the scheduled consultation day or its following day, depending on the drug lifetime.
- $p \in \mathcal{NP} \cap \mathcal{P}^C$: update set of possible days for treatment, $\mathcal{D}^5_{p1}$, to include only the following day of the scheduled consultation day.
- For $p \in \mathcal{P}^C \cap \mathcal{P}^T$: if the k^{th} consultation appointment should be followed by k'^{th} treatment appointment on the next day, update set of possible time slots for drug preparation on the scheduled consultation day, $\mathcal{S}^4_{pk'd}$, to include only time slots after completing the consultation.

6: Solve *Treatment-Model* with $\mathcal{P}^T$ and $\mathcal{AP}^T$.

7: Update set of scheduled appointments at the previous step:

- For $p \in \mathcal{NP} \cap \mathcal{P}^T$: if the required treatment appointment could not be scheduled in the second step, remove the associated scheduled consultation appointment from the schedule.

8: Calculate performance measures and return the schedule of all appointments.

Algorithm 8: Treatment-Consultation Approach

1: Initialization:

- Generate $\mathcal{P}^C$: set of patients with consultation appointments, and $\mathcal{P}^T$: set of patients with treatment appointments.
- Generate set of all consultation appointments: $\mathcal{AP}^C = \bigcup_{p \in \mathcal{P}^C} \mathcal{CA}_p$.
- Generate set of all treatment appointments: $\mathcal{AP}^T = \bigcup_{p \in \mathcal{P}^T} \mathcal{TA}_p$.

2: Solve *Treatment-Model* with $\mathcal{P}^T$ and $\mathcal{AP}^T$.

3: Update the number of available *Phlebotomist*.

4: Update set of appointments to be scheduled at the next step:

- For $p \in \mathcal{NP} \cap \mathcal{P}^T$: if the required treatment appointment could not be scheduled, remove the associated consultation appointment from $\mathcal{AP}^C$.

5: Update domain of next step variables:

- For $p \in \mathcal{NP} \cap \mathcal{P}^T$: update set of possible days for blood test, $\mathcal{D}^1_{p1}$, to include only the day before the scheduled treatment day.
- For $p \in \mathcal{NP} \cap \mathcal{P}^C$: update set of possible days for consultation, $\mathcal{D}^3_{p1}$, to include only the previous day of the scheduled treatment day.
- For $p \in \mathcal{P}^T \cap \mathcal{P}^C$: if the k^{th} treatment appointment should follows the k'^{th} consultation appointment on the next day, update set of possible time slots for blood test and consultation on the scheduled drug preparation day, $\mathcal{S}^1_{pk'd}$ and $\mathcal{S}^3_{pk'd}$, to include only time slots before starting drug preparation.

6: Solve *Consultation-Model* with $\mathcal{P}^C$ and $\mathcal{AP}^C$.

7: Update set of scheduled appointments at the previous step:

- For each unscheduled appointment of $\mathcal{AP}^C$: if $\mathcal{AP}^T$ includes an associated scheduled treatment appointment which would follow the consultation appointment, remove that appointment from the schedule.

8: Calculate performance measures and return the schedule of all appointments.

C Sequential Models

We next provide mathematical programming formulations for the consultation and treatment models, that are derived from M2.

Consultation model:

$$\text{minimize}\left\{g_3\left(y^3\right), g_4^1\left(y^1, y^3\right), g_5'\left(y^3\right), g_8'\left(z^1\right)\right\}$$

$$\text{subject to} \quad (2.1), (2.3), (2.8), (2.34), (2.36)$$

$$\sum_{p\in\mathcal{P}} \sum_{k'\in\mathcal{K}_p^1(d,s)} \sum_{s'=s(B)}^{s} y^1_{pk'ds'} \le R^B_{ds} \qquad d\in\mathcal{D}, s\in\mathcal{S} \tag{9}$$

$$y^i_{pkds} \in \{0,1\} \qquad i\in\{1,3\}, p\in\mathcal{P}, k\in\mathcal{CA}_p, d\in\mathcal{D}, s\in\mathcal{S}. \tag{10}$$

Treatment model:

$$\text{minimize}\left\{g_1\left(x\right), g_2\left(x\right), g_4^2\left(x, y^2\right), g_4^3\left(x, y^2\right), g_5\left(x\right), g_6\left(x\right), g_7\left(x\right), g_8''\left(z^2\right)\right\}$$

$$\text{subject to} \quad (2.2), (2.4), (2.5), (2.10)-(2.15), (2.26), (2.27), (2.29), (2.31), (2.42), (2.44), (2.45), (2.47)$$

$$\sum_{s\in\mathcal{S}^5_{p1d}} \sum_{c\in\mathcal{C}^P_{p1}} x_{p1dsc} = \sum_{s\in\mathcal{S}^j_{p1d}} y^j_{p1(d-1)s} \qquad j\in\{2,4\}, p\in\mathcal{P}^{5,j}, d\in\mathcal{D}^{5,j}_{p1} \tag{11}$$

$$\sum_{p\in\mathcal{NP}\cup\mathcal{RP}} \sum_{k\in\mathcal{K}_p^2(d,s)} \sum_{s'=s(B)}^{s} y^2_{pkds'} \le R^B_{ds} \qquad d\in\mathcal{D}, s\in\mathcal{S} \tag{12}$$

$$y^i_{pkds} \in \{0,1\} \qquad i\in\{2,4\}, p\in\mathcal{NP}\cup\mathcal{RP}, k\in\mathcal{TA}, d\in\mathcal{D}, s\in\mathcal{S} \tag{13}$$

$$x_{pkdsc} \in \{0,1\} \qquad p\in\mathcal{NP}\cup\mathcal{RP}, k\in\mathcal{TA}, d\in\mathcal{D}, s\in\mathcal{S}, c\in\mathcal{C}. \tag{14}$$

D Computational Experiments Supplements for Chapter 2

Objective coefficients:

We have used the AHP method to determine the objective coefficients with respect to the pair-wise comparison among all the objective components. Usually these pair-wise comparisons are done by assigning a number from 1/9 to 9 that reflects the importance of one objective when compared the other one. However, in this paper, we have used a different scale to emphasize that the scheduling of appointments is the most important goal and to make sure that the corresponding objective weight is high enough to reflect this importance. We thus considered that the eighth objective is 40 and we used rates 1/5 to 5 to compare the other objectives. These comparisons are made according to the information we obtained during our interviews with the head oncologist and head nurse of the clinic. Table 2 gives the corresponding pairwise comparison matrix. According to this matrix, we can calculate the coefficients indicating the relative importance of objective terms. For this purpose, the sum of the values of each column of the matrix must be calculated first, and then each value should be divided by its column total value to obtain the relative importance matrix. The average of each row in the relative importance matrix represents the corresponding weight of each objective function.

Table 3 summarizes the obtained objective coefficients according to the described procedure. The second and third columns of this table denote the considered weights for the integrated model. The second column weights are used when we consider all stages and all the objectives. The weights in the third column are used when we want to compare our results to the actual plan where we only consider two stages and ignore waiting time and access time minimization. The fourth and fifth columns represent the weights for the consultation model in CT and TC approaches, respec-

Table 2: Pair-wise comparison matrix

	g_1	g_2^1	g_2^2	g_2^3	g_3	g_4	g_5	g_6	g_7	g_8
g_1	1	2	2	4	2	0.5	0.33	2	4	0.03
g_2^1	0.5	1	1	4	1	0.33	0.25	0.5	4	0.03
g_2^2	0.5	1	1	4	1	0.33	0.25	0.5	4	0.03
g_2^3	0.25	0.25	0.25	1	0.25	0.25	0.25	0.25	0.5	0.03
g_3	0.5	1	1	4	1	0.33	0.25	0.5	4	0.03
g_4	2	3	3	4	3	1	0.5	2	4	0.03
g_5	3	4	4	4	4	2	1	2	4	0.03
g_6	0.5	2	2	4	2	0.5	0.5	1	4	0.03
g_7	0.25	0.25	0.25	2	0.25	0.25	0.25	0.25	1	0.03
g_8	40	40	40	40	40	40	40	40	40	1

tively. Finally, the obtained coefficients for the treatment model are reported in the sixth column. Please note that in the cases where we consider a subset of objective functions instead of all of them, a similar pair-wise matrix as shown in Table 2 is used for calculating the coefficients. The only difference is that we first remove the column and rows related to the objective that are not considered, and we then perform the calculations on the obtained smaller matrix.

Table 3: Objective function coefficients

	Integrated		Consultation		Treatment
	Model	*Real*	*CT*	*TC*	
$g_1(x)$	3	4	-	-	3
$g_2^1(y^3)$	2	3	2	3	-
$g_2^2(y^3)$	2	3	2	3	-
$g_2^3(y^3)$	1	1	1	1	-
$g_3(x)$	2	3	-	-	2
$g_4(x, y^1, y^2, y^3)$	4	-	4	5	4
$g_5(x)$	6	-	6	-	5
$g_6(x)$	3	4	-	-	2
$g_7(x)$	1	1	-	-	1
$g_8(z^1, z^2)$	76	81	85	88	83

Problem instance setting:

Table 4 summarizes the setting of problem instances in the T-set. The number of consultation and treatment appointments that were previously booked in the initial schedule are shown in columns 11 and 12. Table 5 summarizes the setting of problem instances in the S-set and D-set.

Table 4: Setting of instances in the T-set

	$\lvert\mathcal{O}\rvert$	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{FP}\rvert$	$\lvert\mathcal{NP}\rvert$	$\lvert\mathcal{RP}\rvert$	$\lvert\mathcal{K}^C\rvert$	$\lvert\mathcal{K}^T\rvert$	$\lvert\mathcal{K}_I^C\rvert$	$\lvert\mathcal{K}_I^T\rvert$
T1	3	13	5	1	3	2	35	22	18	4	6
T2	4	14	4	1	4	2	29	21	14	26	9
T3	3	9	6	2	3	1	61	53	13	44	5
T4	5	11	7	2	1	1	35	23	17	52	23
T5	5	4	4	3	6	6	59	48	28	10	4
T6	4	5	6	3	9	4	67	57	25	27	8
T7	5	5	6	4	2	3	80	65	23	60	12
T8	3	12	11	4	2	3	31	21	16	79	55
T9	3	9	11	5	2	2	70	50	31	112	37
T10	2	12	12	5	3	2	49	36	21	75	84

Table 5: Setting of instances in the S-set and D-set

		WLs	$\lvert\mathcal{FP}\rvert$	$\lvert\mathcal{NP}\rvert$	$\lvert\mathcal{RP}\rvert$	$\lvert\mathcal{K}\rvert$
S-set	S1	-	96	33	698	1173
D-set	D1	2	48	17	349	587
	D2	4	24	8	174	293
	D3	8	12	4	87	147
	D4	12	8	3	58	98

E Proofs of Valid Inequalities

E.1 Proof of Proposition 1

Proof. Given $r \geq 0$, we rewrite inequalities (4.15) for $t' = t - r$ and thus, we have $\mathcal{S}_{kit'} = \left\{ s \in \mathcal{T} | 0 \leq s \leq t - r - \left(p^u_{o(k)i} + w_k \overline{d} \right) \right\} \subseteq \mathcal{S}_{kit}$. Therefore, for any $t' \leq t$, we have:

$$y_{d(k)jt'} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{kit'}} x_{o(k)is} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{kit}} x_{o(k)is}, \tag{15}$$

where the second inequalities follows from $\mathcal{S}_{kit'} \subseteq \mathcal{S}_{kit}$. Moreover, we know that for any outgoing truck and a given outbound door j, at most one time slot can be assigned. Therefore, we have:

$$\sum_{t' \leq t} y_{d(k)jt'} \leq \sum_{t \in \mathcal{T}} y_{d(k)jt} \leq 1, \tag{16}$$

where the first inequality follows from $\{t' \in \mathcal{T} | t' \leq t\} \subseteq \mathcal{T}$. From (15) and (16), and considering the fact that the maximum value for the left-hand-side of inequalities (15) is one, for $k \in \mathcal{K}$, $j \in \mathcal{J}$, and $t \in \mathcal{T}$, we obtain:

$$\sum_{t' \leq t} y_{d(k)jt'} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{kit}} x_{o(k)is}. \tag{17}$$

Moreover, we know that each truck will be assigned to at most one door and one time slot. Therefore, for each $k \in \mathcal{K}$ and $t \in \mathcal{T}$, we have:

$$\sum_{j \in \mathcal{J}} \sum_{t' \leq t} y_{d(k)jt'} \leq \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}} y_{d(k)jt} = 1, \tag{18}$$

where the first inequality follows from $\{t' \in \mathcal{T} | t' \leq t\} \subseteq \mathcal{T}$. Combining (17) and (18), we obtain:

$$\sum_{j \in \mathcal{J}} \sum_{t' \leq t} y_{d(k)jt'} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{kit}} x_{o(k)is},$$

and the result follows. $\square$

E.2 Proof of Proposition 2

Proof. Using the definition of $\mathcal{N}_m$, for $m \in \mathcal{M}, n \in \mathcal{N}_m, j \in \mathcal{J}$, and $t \in \mathcal{T}$, we can rewrite inequalities (4.15) as:

$$y_{njt} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{mnit}} x_{mis}, \tag{19}$$

where, $\mathcal{S}_{mnit} = \left\{ s \in \mathcal{T} | 0 \leq s \leq t - \left(p^u_{mi} + w_{k_{mn}} \overline{d} \right) \right\}$. For $m \in \mathcal{M}, n \in \mathcal{N}_m, j \in \mathcal{J}$, and $t \in \mathcal{T}$, using inequalities (19) and given that for each $n \in \mathcal{N}_m$, $\mathcal{S}_{mnit} \subseteq \mathcal{S}_{mit}$, we have:

$$y_{njt} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{mnit}} x_{mis} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{mit}} x_{mis}. \tag{20}$$

Moreover, given that at any given door j and any given time slot t, at most one truck can be scheduled, we have:

$$\sum_{n \in \mathcal{N}_m} y_{njt} \leq \sum_{n \in \mathcal{N}} y_{njt} \leq 1. \tag{21}$$

where the first inequality follows from $\mathcal{N}_m \subseteq \mathcal{N}$. Given that the right-hand-side of (20) does not depend on n and combining (21) and (20), for $m \in \mathcal{M}, j \in \mathcal{J}$, and $t \in \mathcal{T}$, the following inequalities are obtained:

$$\sum_{n \in \mathcal{N}_m} y_{njt} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{mit}} x_{mis}. \tag{22}$$

According to (21), at any given door and time slot, it is not possible to schedule more than one truck in $n \in \mathcal{N}_m$. Moreover, it is also not possible to schedule more than a truck in a set of periods before t that would create an overlap of two or more trucks. This means that, if any of the trucks is scheduled to start the process at time $t \in \mathcal{T}$, the door will be occupied for a duration equal to the processing time of such truck, and thus no other truck can be scheduled during that time period. For $j \in \mathcal{J}$ and $t \in \mathcal{T}$, this situation can be stated as follows:

$$\sum_{n \in \mathcal{N}_m} \sum_{r \in \mathcal{R}_{mjt}} y_{njr} \leq \sum_{n \in \mathcal{N}} \sum_{r \in \mathcal{R}_{jt}} y_{njr} \leq 1, \tag{23}$$

where, $\mathcal{R}_{jt} = \left\{ r \in \mathcal{T} | t - \min_{n \in \mathcal{N}} \left\{ p^l_{nj} \right\} < r \leq t \right\}$ and the first inequality follows from $\mathcal{N}_m \subseteq \mathcal{N}$. Combining (22) and (23), we obtain

$$\sum_{n \in \mathcal{N}_m} \sum_{r \in \mathcal{R}_{mjt}} y_{njr} \leq \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}_{mit}} x_{mis},$$

and the result follows. □

E.3 Proof of Proposition 3

Proof. To prove validity for inequalities (V3), we need to consider the two possible values that variables u can take. If incoming truck $o(k)$ is assigned to the door i, i.e. $u_{o(k)i} = 1$, inequality (V3) reduces to

$$\sum_{j \in \mathcal{J}} \sum_{t' \leq t} y_{d(k)jt'} \leq \sum_{s \in \mathcal{S}_{kit}} x_{o(k)is},$$

which is valid and strengthens (V2) by exploiting the fact that

$$\sum_{i' \in \mathcal{I} \setminus \{i\}} \sum_{s \in \mathcal{S}_{kit}} x_{o(k)' is} = 0,$$

when $u_{o(k)i} = 1$, in any feasible solution in X. If incoming truck $o(k)$ is not assigned to the door i, i.e. $u_{o(k)i} = 0$, inequality (V3) reduces to

$$\sum_{j \in \mathcal{J}} \sum_{t' \leq t} y_{d(k)jt'} \leq \sum_{s \in \mathcal{S}_{kit}} x_{o(k)is} + 1,$$

which is redundant and thus, valid for any feasible solution in X. Therefore, the result follows. $\square$

Propositions 4 to 8 can be proved using similar arguments to the ones used above. Therefore, we omit these proofs.

F Proof of Proposition 9

Proof. If we define S_{min} as the latest possible time at which the first incoming truck is scheduled (at any door), we can reformulate the CDSPHT-G as follows:

$$
\begin{aligned}
\text{(F3-G)} \quad & \text{maximize } S_{min} \\
& \text{subject to } (4.2) - (4.12) \\
& \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} t x_{mit} \geq S_{min} \qquad m \in \mathcal{M}.
\end{aligned}
\tag{24}
$$

We note that both F1-G and F3-G provide the same set of optimal assignments and sequencing decisions for incoming and outgoing trucks. However, the optimal scheduled start times will be different. In particular, any time t' in F3-G, will correspond to time $t' - S_{min}$ in F1-G. Let C^*_{max} and S^*_{min} be the optimal solution values for the F1-G and F3-G, respectively, where $C^*_{max} = |\mathcal{T}| - S^*_{min}$. To formulate the ICDSP-DD, we reduce F3-G as follows:

$$
\begin{aligned}
\text{(MIP-IT1)} \quad & \text{maximize } S_{min} \\
& \text{subject to } (4.2), (4.4), (4.10), (24) \\
& \sum_{t \in \mathcal{T}} (t + p^u_{mi}) \, x_{mit} \leq DD_{mi} \qquad m \in \mathcal{M}, i \in \mathcal{I},
\end{aligned}
\tag{25}
$$

where DD_{mi} denotes the due date of incoming truck m at door i and is calculated based on a given schedule of outgoing trucks. Suppose that t_n^* and d_n^* represent the start time of outgoing truck $n \in \mathcal{N}$ and the door assigned to it, respectively, in the given outgoing trucks schedule. For each $m \in \mathcal{M}$ and $i \in \mathcal{I}$, we have $DD_{mi} = \min_{n \in \mathcal{N}_m} \left\{ t_n^* - w_{k_{mn}} d_{id_n^*} \right\}$.

Consider a new coordinate system obtained by inverting the order of time periods from zero to $|\mathcal{T}|$ as shown in the lower chart of Figure 4.1. The relation between any time slot r in the original coordinate system and any time slot t' in the inverted coordinate system is $t = |\mathcal{T}| - t'$. Therefore, $S_{min} = |\mathcal{T}| - C'_{max}$. Moreover, suppose that $x'_{mit'}$ is defined in the inverted system and corresponds to x_{mit} in the original coordinate system, i.e., $x'_{mit'} \equiv x_{mit} \equiv x_{mi(|\mathcal{T}|-t')}$. Now, considering $t = |\mathcal{T}| - t'$, we can rewrite the objective function of MIP-IT1 as:

$$\begin{aligned} \text{maximize } \; S_{min} &\equiv \text{maximize } \; \{|\mathcal{T}| - C'_{max}\} \\ &\equiv |\mathcal{T}| + \text{maximize } \; - C'_{max} \\ &\equiv |\mathcal{T}| - \text{minimize } \; C'_{max}. \end{aligned} \tag{26}$$

Similarly, constraints (24) can be rewritten as:

$$\sum_{i \in \mathcal{I}} \sum_{|\mathcal{T}|-t'=0}^{|\mathcal{T}|-t'=|\mathcal{T}|} (|\mathcal{T}| - t')\, x_{mi(|\mathcal{T}|-t')} \geq |\mathcal{T}| - C'_{max},$$

which are equivalent to

$$\sum_{i \in \mathcal{I}} \sum_{t'=|\mathcal{T}|}^{t'=0} t' x_{mi(|\mathcal{T}|-t')} \leq C'_{max}.$$

The above constraint is still stated based on the original coordinate system using the variables defined for that system. If we rewrite it according to the inverted coordinate system, we have:

$$\sum_{i \in \mathcal{I}} \sum_{t' \in \mathcal{T}} t' x'_{mit'} \leq C'_{max} \qquad m \in \mathcal{M}. \tag{27}$$

Replacing t with $|\mathcal{T}| - t'$, and rewriting constraints (4.2) in the inverted system, they will be the same, but with the new $x'_{mit'}$ variables.

In constraints (4.4), we have an interval for r which is defined as $max\left\{0, t - p^u_{mi} + 1\right\} \leq r \leq t$. If we replace r and t with the corresponding values, we obtain:

$$\begin{aligned}
max\left\{0, |\mathcal{T}| - t' - p^u_{mi} + 1\right\} \leq |\mathcal{T}| - r' \leq |\mathcal{T}| - t' \qquad &\Rightarrow \\
max\left\{0, |\mathcal{T}| - t' - p^u_{mi} + 1\right\} - |\mathcal{T}| \leq -r' \leq -t' \qquad &\Rightarrow \\
t' \leq r' \leq |\mathcal{T}| - max\left\{0, |\mathcal{T}| - t' - p^u_{mi} + 1\right\} \qquad &\Rightarrow \\
t' \leq r' \leq |\mathcal{T}| + min\left\{0, -|\mathcal{T}| + t' + p^u_{mi} - 1\right\} \qquad &\Rightarrow \\
t' \leq r' \leq min\left\{|\mathcal{T}|, t' + p^u_{mi} - 1\right\}. &
\end{aligned}$$

Therefore, we can rewrite constraints (4.4) as:

$$\sum_{m \in \mathcal{M}} \sum_{r'=t'}^{min\left\{|\mathcal{T}|, t' + p^u_{mi} - 1\right\}} x'_{mir'} \leq 1 \qquad t' \in \mathcal{T}, i \in \mathcal{I}. \tag{28}$$

Using the definition of DD_{mi}, constraints (25) can be rewritten as:

$$\sum_{t \in \mathcal{T}} \left(t + p^u_{mi}\right) x_{mit} \leq t^*_n - w_{k_{mn}} d_{id^*_n} \qquad m \in \mathcal{M}, i \in \mathcal{I}, n \in \mathcal{N}_m. \tag{29}$$

Moreover, by substituting t by $|\mathcal{T}| - t'$ and t^*_n by $|\mathcal{T}| - t'^*_n$, we can rewrite constraints (29) as:

$$\sum_{t' \in \mathcal{T}} \left(t' - p^u_{mi}\right) x'_{mit'} \geq t'^*_n + w_{k_{mn}} d_{id^*_n} \qquad m \in \mathcal{M}, i \in \mathcal{I}, n \in \mathcal{N}_m. \tag{30}$$

According to Figure 4.1, variables $x'_{mit'}$ that are used in constraints (27), (28), and (30), are pointing to the completion time of the processes and not to the start times. More precisely, $x'_{mit'}$ will take value one if and only if truck m is assigned to door i and will complete the process at time t'. However, we can rewrite these constraints using the $x''_{mit'}$ variables instead, which are defined based on start times:

$$\sum_{i \in \mathcal{I}} \sum_{t' \in \mathcal{T}} (t' + p^u_{mi})\, x''_{mit'} \leq C'_{max} \qquad m \in \mathcal{M} \quad (31)$$

$$\sum_{m \in \mathcal{M}} \sum_{r'=max\left\{1, t'-p^u_{mi}+1\right\}}^{t'} x''_{mir'} \leq 1 \qquad t' \in \mathcal{T}, i \in \mathcal{I} \quad (32)$$

$$\sum_{t' \in \mathcal{T}} t' x''_{mit'} \geq t''^*_n + p^l_{nd^*_n} + w_{k_{mn}} d_{id^*_n} \qquad m \in \mathcal{M}, i \in \mathcal{I}, n \in \mathcal{N}_m. \quad (33)$$

We note that in constraints (30), t'^*_n is obtained based on directly mapping of the variables in the inverted coordinate system, and thus it points to the completion time of truck n. When we change the variables to point to the start times in constraints (33), this also applies for the outbound side, and so we need to update t'^*_n to $t''^*_n + p^l_{nd^*_n}$, in which t''^* is obtained based on start time variables. If we define $RD'_{mi} = \max_{n \in \mathcal{N}_m} \left\{ t''^*_n + p^l_{nd^*_n} + w_{k_{mn}} d_{id^*_n} \right\}$, we can reformulate the ICDSP-DD as:

$$\text{(MIP-IT2)} \quad \text{minimize} \quad C'_{max}$$

$$\text{subject to} \quad (4.2), (4.10), (30), (31)$$

$$\sum_{t' \in \mathcal{T}} t' x''_{mit'} \geq RD'_{mi} \qquad m \in \mathcal{M}, i \in \mathcal{I}. \quad (34)$$

We recall that given the optimal solution value C^*_{max} of F1-G, we have $C^*_{max} = |\mathcal{T}| - S^*_{min}$. Moreover, we also showed that $S^*_{min} = |\mathcal{T}| - C'^*_{max}$. Therefore, we conclude that $C^*_{max} = C'^*_{max}$. This means that both MIP-IT2 and F1-G with fixed outbound side decisions, result in exactly the same optimal solution value. Comparing MIP-

IT1 and MIP-IT2, we observe that both models are exactly the same except for a few details. In MIP-T2, as we have modified the coordinate system, we are obtaining a sequence of trucks that is the inverse of the original sequence. Therefore, both MIP-T1 and MIP-T2 result in the same assignment decisions and the obtained sequencing decisions are the same but in a reverse order. Therefore, when we solve ICDSP-DD with MIP-IT2, we should reverse the obtained sequence to be valid for the original problem. Using MIP-IT2, we note that we have formulated the ICDSP-DD as the OCDSP-RD and the result follows. □

G Detailed Tables of Computational Experiments for Chapter 3

Table 6 gives the details of the comparison between MIP and CP formulations developed for the CDSPHT-G. The first column provides each instance setting. The second column provides the best upper bounds obtained using any of these three formulations. Underlined numbers correspond to the instances that none of the formulations can solve them to optimality. For each of these formulations, three columns are provided. First column reports CPU times. Using *time* in this column corresponds to an instance that cannot be solved by the formulation within the time limit. In the second column, final optimality gaps obtained by CPLEX after the time limit are reported. These gaps are calculated as $100 \times (UB - LB)/UB$, where UB and LB are the bounds provided by CPLEX. The third column represents the % deviation of the obtained UB by each formulation with respect to the best reported bound and is calculated by $100 \times (UB - Best)/Best$. Whenever a formulation cannot find a feasible solution for the problem, we have used NA in the corresponding entries of the table. At the bottom of the table arithmetic and geometric means for each column are reported.

Table 6: Comparison of formulations for CDSPHT-G

Instance	Best	F1-G			F2-G			CP-G		
		Time	*Gap(%)*	*Dev(%)*	*Time*	*Gap(%)*	*Dev(%)*	*Time*	*Gap(%)*	*Dev(%)*
8x4x25	27	0.39	0.00	0.00	0.24	0.00	0.00	0.17	0.00	0.00
8x4x35	43	0.95	0.00	0.00	1.32	0.00	0.00	2.80	0.00	0.00
8x4x50	59	192.24	0.00	0.00	41.06	0.00	0.00	12.47	0.00	0.00
8x4x75	75	7,311.81	0.00	0.00	1,537.30	0.00	0.00	164.59	0.00	0.00
9x4x25	33	0.03	0.00	0.00	0.12	0.00	0.00	0.04	0.00	0.00
9x4x35	48	0.98	0.00	0.00	0.98	0.00	0.00	0.10	0.00	0.00
9x4x50	54	167.55	0.00	0.00	210.91	0.00	0.00	45.20	0.00	0.00
9x4x75	95	time	5.26	0.00	time	12.63	0.00	747.16	0.00	0.00
10x4x25	34	0.03	0.00	0.00	0.15	0.00	0.00	0.07	0.00	0.00
10x4x35	51	37.22	0.00	0.00	13.45	0.00	0.00	16.03	0.00	0.00
10x4x50	76	time	3.95	0.00	time	6.58	0.00	347.12	0.00	0.00
10x4x75	117	time	16.95	0.85	time	16.95	0.85	31,912.00	0.00	0.00
10x5x25	35	0.24	0.00	0.00	0.19	0.00	0.00	0.06	0.00	0.00
10x5x35	44	2.52	0.00	0.00	2.43	0.00	0.00	0.66	0.00	0.00
10x5x50	69	2.44	0.00	0.00	2.74	0.00	0.00	18.62	0.00	0.00
10x5x75	95	time	10.53	0.00	time	10.53	0.00	2,522.75	0.00	0.00
11x5x25	36	26.50	0.00	0.00	4.13	0.00	0.00	26.31	0.00	0.00
11x5x35	53	67.81	0.00	0.00	1,341.62	0.00	0.00	230.03	0.00	0.00
11x5x50	74	time	2.70	0.00	time	2.70	0.00	1,849.49	0.00	0.00
11x5x75	105	time	NA	NA	time	14.15	0.95	time	25.71	0.00
12x5x25	46	6,813.79	0.00	0.00	262.64	0.00	0.00	343.42	0.00	0.00
12x5x35	63	time	3.17	0.00	time	4.76	0.00	6,314.89	0.00	0.00
12x5x50	87	time	8.05	0.00	time	11.36	1.15	time	11.49	0.00
12x5x75	130	time	NA	NA	time	19.08	0.77	time	26.92	0.00
12x6x25	41	1.98	0.00	0.00	0.42	0.00	0.00	0.14	0.00	0.00
12x6x35	52	time	1.92	0.00	time	9.62	0.00	time	17.31	0.00
12x6x50	76	time	6.58	0.00	time	9.21	0.00	time	21.05	0.00
12x6x75	115	time	NA	NA	time	12.07	0.87	time	21.74	0.00
15x6x25	55	time	3.57	1.82	time	5.45	0.00	time	7.27	0.00
15x6x35	84	time	1.19	0.00	6,542.35	0.00	0.00	time	3.57	0.00
15x6x50	114	time	NA	NA	time	18.26	0.88	time	21.93	0.00
15x6x75	NA	time	NA	NA	time	NA	NA	NA	NA	NA
15x7x25	46	41.46	0.00	0.00	46.69	0.00	0.00	3.67	0.00	0.00
15x7x35	71	time	2.82	0.00	time	2.82	0.00	time	15.49	0.00
15x7x50	93	time	3.23	0.00	time	6.38	1.08	time	20.43	0.00
15x7x75	157	time	NA	NA	time	21.66	0.00	NA	NA	NA
20x10x25	63	27,277.01	0.00	0.00	time	1.59	0.00	time	4.76	0.00
20x10x35	90	2,775.66	0.00	0.00	4,958.17	0.00	0.00	time	12.22	0.00
20x10x50	117	time	2.52	1.71	time	2.52	1.71	time	17.09	0.00
20x10x75	198	time	17.59	0.51	time	NA	NA	time	35.35	0.00
50x30x25	150	time	NA	NA	NA	NA	NA	53,302.80	0.00	0.00
50x30x35	203	time	NA	NA	NA	NA	NA	time	14.78	0.00
50x30x50	NA	time	NA	NA	NA	NA	NA	NA	NA	NA
50x30x75	NA	time	NA	NA	NA	NA	NA	NA	NA	NA
Arithmetic mean		50,108.95	2.65	0.14	49,432.77	4.96	0.22	37,008.10	6.93	0.00
Geometric mean		2,183.26	1.15	0.09	1,921.69	2.02	0.16	641.64	2.04	0.00

Table 7: Comparison of formulations for CDSPHT-S

Instance	F1-S				CP-S			
	Time	*LB*	*UB*	*Gap (%)*	*Time*	*LB*	*UB*	*Gap (%)*
8x4x25	0.12	27	27	0.00	0.11	27	27	0.00
8x4x35	0.68	43	43	0.00	1.13	43	43	0.00
8x4x50	6.36	59	59	0.00	8.55	59	59	0.00
8x4x75	179.27	75	75	0.00	32.18	75	75	0.00
9x4x25	0.02	33	33	0.00	0.03	33	33	0.00
9x4x35	0.16	48	48	0.00	0.14	48	48	0.00
9x4x50	7.73	54	54	0.00	11.20	54	54	0.00
9x4x75	time	84	95	11.58	234.57	95	95	0.00
10x4x25	0.03	34	34	0.00	0.04	34	34	0.00
10x4x35	2.62	51	51	0.00	7.52	51	51	0.00
10x4x50	time	75	76	1.32	54.80	76	76	0.00
10x4x75	time	101	118	14.41	time	94	117	19.66
10x5x25	0.07	35	35	0.00	0.07	35	35	0.00
10x5x35	0.41	44	44	0.00	0.15	44	44	0.00
10x5x50	0.34	69	69	0.00	1.85	69	69	0.00
10x5x75	time	83	95	12.63	127.05	95	95	0.00
11x5x25	1.59	36	36	0.00	13.55	36	36	0.00
11x5x35	7.66	53	53	0.00	61.08	53	53	0.00
11x5x50	2,504.61	74	74	0.00	153.07	74	74	0.00
11x5x75	time	93	NA	NA	time	97	105	7.62
12x5x25	65.28	46	46	0.00	95.00	46	46	0.00
12x5x35	time	62	63	1.59	548.04	63	63	0.00
12x5x50	time	82	88	6.82	time	76	87	12.64
12x5x75	time	111	NA	NA	time	109	130	16.15
12x6x25	0.21	41	41	0.00	0.20	41	41	0.00
12x6x35	426.72	52	52	0.00	time	44	52	15.38
12x6x50	time	75	78	3.85	time	68	76	10.53
12x6x75	time	106	116	8.62	time	102	115	11.30
15x6x25	2,567.22	55	55	0.00	time	48	55	12.73
15x6x35	time	83	84	1.19	time	81	84	3.57
15x6x50	time	98	116	15.52	time	86	113	23.89
15x6x75	time	127	NA	NA	time	123	175	29.71
15x7x25	3.58	46	46	0.00	0.21	46	46	0.00
15x7x35	time	70	71	1.41	time	63	71	11.27
15x7x50	time	90	93	3.23	time	89	93	4.30
15x7x75	time	125	NA	NA	time	118	155	23.87
20x10x25	224.92	63	63	0.00	time	60	63	4.76
20x10x35	24.82	90	90	0.00	time	79	90	12.22
20x10x50	time	116	119	2.52	time	97	117	17.09
20x10x75	time	161	NA	NA	time	128	197	35.03
50x30x25	191.61	150	150	0.00	2.49	150	150	0.00
50x30x35	time	176	NA	NA	time	173	194	10.82
50x30x50	time	239	NA	NA	time	229	286	19.93
50x30x75	time	330	NA	NA	time	324	439	26.20
Arithmetic mean	1,859.72			2.35	1,672.52			7.47
Geometric mean	112.05			0.88	92.58			2.59

Table 8: Performance of the proposed BC algorithm

Instance	CPLEX				BC			
	Time	*UB*	*Gap(%)*	*Node(#)*	*Time*	*UB*	*Gap(%)*	*Node(#)*
8x4x25	0.09	27	0.00	0	0.12	27	0.00	0
8x4x35	0.57	43	0.00	57	0.61	43	0.00	57
8x4x50	9.96	59	0.00	32,306	3.77	59	0.00	4,522
8x4x75	128.46	75	0.00	374,634	46.28	75	0.00	51,068
9x4x25	0.02	33	0.00	0	0.02	33	0.00	0
9x4x35	0.12	48	0.00	0	0.13	48	0.00	0
9x4x50	5.22	54	0.00	4,610	4.31	54	0.00	715
9x4x75	6,675.43	95	0.00	5,918,523	2,186.45	95	0.00	566,570
10x4x25	0.03	34	0.00	0	0.03	34	0.00	0
10x4x35	2.13	51	0.00	101	3.02	51	0.00	323
10x4x50	2,392.31	76	0.00	1,650,375	237.23	76	0.00	40,622
10x4x75	time	118	18.53	4,358,995	time	117	23.08	883,100
10x5x25	0.07	35	0.00	0	0.07	35	0.00	0
10x5x35	0.36	44	0.00	0	0.36	44	0.00	0
10x5x50	0.36	69	0.00	0	0.33	69	0.00	0
10x5x75	2,932.62	95	0.00	1,985,471	349.25	95	0.00	52,669
11x5x25	1.31	36	0.00	19	0.67	36	0.00	0
11x5x35	7.51	53	0.00	4,502	1.85	53	0.00	421
11x5x50	1,278.39	74	0.00	265,851	210.56	74	0.00	15,677
11x5x75	time	NA	NA	4,139,856	time	105	3.81	219,721
12x5x25	37.23	46	0.00	18,837	9.07	46	0.00	2,311
12x5x35	time	63	1.59	1,627,120	333.33	63	0.00	32,342
12x5x50	time	88	9.13	1,394,551	time	87	3.45	281,110
12x5x75	time	131	15.27	1,659,203	time	132	16.86	92,132
12x6x25	0.19	41	0.00	0	0.21	41	0.00	0
12x6x35	time	52	1.92	6,612,240	159.04	52	0.00	29,156
12x6x50	1,782.42	76	0.00	512,291	1,361.61	76	0.00	83,653
12x6x75	time	NA	NA	4,444,739	time	115	9.57	578,143
15x6x25	800.84	55	0.00	50,427	342.31	55	0.00	12,057
15x6x35	1,959.01	84	0.00	117,065	216.01	84	0.00	6,987
15x6x50	time	115	20.57	1,050,201	time	116	22.41	42,308
15x6x75	time	NA	NA	2,550,005	time	NA	NA	10,026
15x7x25	4.46	46	0.00	311	4.41	46	0.00	282
15x7x35	time	71	1.41	2,186,801	3,526.12	71	0.00	242,146
15x7x50	time	NA	NA	889,857	time	93	2.15	187,503
15x7x75	time	155	20.65	1,344,708	time	157	26.37	12,987
20x10x25	243.87	63	0.00	20,909	51.19	63	0.00	1,089
20x10x35	26.96	90	0.00	2,562	20.81	90	0.00	154
20x10x50	time	118	3.47	455,670	887.44	117	0.00	5,004
20x10x75	time	NA	NA	588,724	time	NA	NA	1,714
50x30x25	215.38	150	0.00	0	204.62	150	0.00	0
50x30x35	time	NA	NA	NA	time	NA	NA	NA
50x30x50	time	NA	NA	NA	time	NA	NA	NA
50x30x75	time	NA	NA	NA	time	NA	NA	NA
Arithmetic mean	3,204		2.57	1,079,549	2,358		2.76	84,307
Geometric mean	138		0.68	10,423	74		0.63	1,328

Table 9: Comparison of different relaxations in the approximate algorithm for the CDSPHT-G

Instance	F1-G		F2-G		F1-S		MIP-OT	
	in	*out*	*in*	*out*	*in*	*out*	*in*	*out*
8x4x25	3.57	3.57	0.00	0.00	3.57	3.57	3.57	3.57
8x4x35	2.33	6.67	0.00	0.00	0.00	2.27	6.98	14.89
8x4x50	11.86	9.84	1.69	0.00	0.00	0.00	8.33	6.78
8x4x75	18.42	11.84	0.00	0.00	0.00	0.00	13.16	12.00
9x4x25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9x4x35	0.00	2.04	0.00	0.00	0.00	0.00	2.04	2.04
9x4x50	15.79	7.02	0.00	0.00	0.00	0.00	15.79	17.24
9x4x75	18.75	17.35	NA	1.05	0.00	0.00	17.71	19.39
10x4x25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10x4x35	13.73	3.85	1.96	0.00	0.00	3.77	7.55	9.26
10x4x50	12.66	13.92	23.68	6.58	0.00	0.00	17.72	16.67
10x4x75	21.85	15.97	29.91	NA	24.58	24.58	21.01	21.67
10x5x25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10x5x35	17.34	0.00	0.00	0.00	0.00	0.00	4.35	4.35
10x5x50	0.00	0.00	0.00	0.00	0.00	0.00	1.43	0.00
10x5x75	17.89	14.74	22.11	7.37	0.00	0.00	16.84	16.84
11x5x25	8.11	5.41	2.70	0.00	0.00	2.70	12.82	15.00
11x5x35	5.66	3.77	1.89	0.00	0.00	0.00	5.66	7.41
11x5x50	20.51	8.97	18.92	1.35	0.00	0.00	10.26	10.26
11x5x75	21.82	14.15	NA	NA	5.71	5.71	14.29	18.18
12x5x25	10.64	8.33	4.35	0.00	0.00	0.00	10.42	10.42
12x5x35	10.94	7.69	20.31	3.17	0.00	0.00	14.71	13.43
12x5x50	18.89	14.44	23.86	10.23	5.75	5.75	12.50	14.44
12x5x75	21.64	19.70	NA	NA	17.42	17.42	19.23	21.64
12x6x25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12x6x35	18.52	11.32	17.31	7.69	0.00	0.00	13.21	14.81
12x6x50	12.66	12.66	24.36	10.53	0.00	0.00	11.84	15.19
12x6x75	12.07	17.09	NA	15.65	10.43	10.43	17.95	17.95
15x6x25	8.93	8.77	19.64	8.93	1.79	1.79	10.53	12.07
15x6x35	10.59	5.75	9.52	2.38	0.00	1.18	3.53	4.65
15x6x50	31.09	18.97	NA	23.48	21.05	22.41	18.26	19.66
15x6x75	39.23	30.05	NA	NA	NA	NA	23.63	21.91
15x7x25	4.17	2.13	0.00	0.00	2.13	2.13	2.13	0.00
15x7x35	12.50	6.76	15.49	2.82	0.00	0.00	2.82	5.48
15x7x50	9.18	14.74	22.92	NA	2.15	2.15	19.39	15.96
15x7x75	32.91	22.78	NA	NA	25.16	25.16	18.99	17.42
20x10x25	11.76	7.69	6.25	6.25	0.00	3.08	13.04	11.76
20x10x35	13.04	0.00	10.99	0.00	0.00	1.10	1.11	1.11
20x10x50	16.53	4.13	NA	12.50	0.00	0.00	3.36	4.17
20x10x75	30.15	28.94	NA	NA	NA	NA	17.68	17.68
50x30x25	NA	NA	NA	NA	NA	NA	0.00	0.00
50x30x35	NA	NA	NA	NA	NA	NA	8.63	9.09
50x30x50	NA	NA	NA	NA	NA	NA	9.66	9.66
50x30x75	NA	NA	NA	NA	NA	NA	10.88	10.68
Arithmetic mean	13.39	9.53	8.96	3.64	3.15	3.56	10.07	10.56
Geometric mean	8.81	6.03	3.57	1.45	0.86	1.24	6.92	7.02
Optimal (#)	6	7	12	18	27	21	5	5
Infeasible (#)	4	4	13	11	6	6	0	0

Table 10: Comparison of % gap of best-known solution with respect to different relaxations

Instance	Best	F1-G			F2-G			F1-S	MIP-OT
		LP	*Relax X*	*Relax Y*	*LP*	*Relax X*	*Relax Y*		
8x4x25	27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8x4x35	43	11.63	2.33	2.33	11.63	0.00	0.00	0.00	6.98
8x4x50	59	20.34	6.78	11.86	20.34	0.00	1.69	0.00	6.78
8x4x75	75	24.00	10.67	17.33	24.00	0.00	0.00	0.00	12.00
9x4x25	33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9x4x35	48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9x4x50	54	16.67	1.85	11.11	16.67	0.00	0.00	0.00	11.11
9x4x75	95	31.58	14.74	17.89	32.63	1.05	26.32	0.00	16.84
10x4x25	34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10x4x35	51	15.69	1.96	13.73	15.69	0.00	1.96	0.00	3.92
10x4x50	76	21.05	10.53	9.21	25.00	6.58	23.68	0.00	14.47
10x4x75	117	33.33	14.53	20.51	33.33	12.82	29.91	23.93	19.66
10x5x25	35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10x5x35	44	9.09	0.00	4.55	9.09	0.00	0.00	0.00	0.00
10x5x50	69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10x5x75	95	28.42	14.74	17.89	29.47	7.37	22.11	0.00	16.84
11x5x25	36	13.89	2.78	5.56	13.89	0.00	0.00	0.00	5.56
11x5x35	53	5.66	3.77	5.66	5.66	0.00	1.89	0.00	5.66
11x5x50	74	18.92	4.05	16.22	18.92	1.35	18.92	0.00	5.41
11x5x75	105	32.38	13.33	18.10	32.38	23.81	32.38	5.71	14.29
12x5x25	46	15.22	4.35	8.70	15.22	0.00	4.35	0.00	6.52
12x5x35	63	19.05	4.76	9.52	19.05	3.17	19.05	0.00	7.94
12x5x50	87	22.99	11.49	16.09	22.99	9.20	22.99	5.75	11.49
12x5x75	130	29.23	18.46	19.23	33.85	25.38	33.85	16.15	19.23
12x6x25	41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12x6x35	52	17.31	9.62	15.38	17.31	7.69	17.31	**0.00**	11.54
12x6x50	76	21.05	9.21	9.21	22.37	10.53	22.37	**0.00**	11.84
12x6x75	115	24.35	15.65	11.30	27.83	15.65	27.83	10.43	16.52
15x6x25	55	18.18	5.45	7.27	18.18	7.27	18.18	**0.00**	7.27
15x6x35	84	9.52	2.38	9.52	9.52	2.38	9.52	0.00	2.38
15x6x50	114	28.95	17.54	28.07	28.95	22.81	28.95	21.05	17.54
15x6x75	178	38.20	28.09	38.20	38.20	38.20	38.20	32.02	21.91
15x7x25	46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15x7x35	71	15.49	2.82	11.27	15.49	2.82	15.49	**0.00**	2.82
15x7x50	93	17.20	12.90	4.30	20.43	3.23	20.43	2.15	15.05
15x7x75	155	32.26	21.29	31.61	32.26	30.97	32.26	25.16	17.42
20x10x25	63	4.76	4.76	4.76	4.76	4.76	4.76	0.00	4.76
20x10x35	90	11.11	0.00	11.11	10.00	0.00	10.00	0.00	1.11
20x10x50	117	13.68	0.85	13.68	12.82	10.26	12.82	**0.00**	1.71
20x10x75	198	30.30	28.79	29.80	32.83	32.83	32.83	24.24	17.68
50x30x25	150	NA	NA	NA	NA	NA	NA	NA	0.00
50x30x35	197	NA	NA	NA	NA	NA	NA	NA	8.63
50x30x50	290	NA	NA	NA	NA	NA	NA	NA	9.66
50x30x75	440	NA	NA	NA	NA	NA	NA	NA	10.68
Arithmetic mean		16.29	7.51	11.27	16.77	7.00	13.25	4.17	8.26
Geometric mean		9.74	4.00	6.77	9.92	2.50	5.62	0.94	4.92
Optimal (#)		8	10	8	8	18	13	30	10

Table 11: Performance comparison between approximate algorithms and formulations for CDSPHT-G

Instance	Best	Approximate algorithm						Formulations								
		CP-OT			MIP-OT			F1-G			F2-G			CP-G		
		Time	*Gap*	*Dev*	*Time*	*Gap*	*Dev*	*Time*	*Gap*	*Dev*	*Time*	*Gap*	*Dev*	*Time*	*Gap*	*Dev*
8x4x25	27	1	3.57	3.70	1	3.57	3.70	0	0.00	0.00	0	0.00	0.00	0	0.00	0.00
8x4x35	43	2	0.00	0.00	2	2.27	2.33	1	0.00	0.00	1	0.00	0.00	2	0.00	0.00
8x4x50	59	5	0.00	0.00	7	0.00	0.00	111	0.00	0.00	99	0.00	0.00	19	0.00	0.00
8x4x75	75	53	0.00	0.00	62	0.00	0.00	time	5.33	0.00	time	1.33	0.00	103	0.00	0.00
9x4x25	33	1	0.00	0.00	1	0.00	0.00	0	0.00	0.00	0	0.00	0.00	0	0.00	0.00
9x4x35	48	1	0.00	0.00	1	0.00	0.00	1	0.00	0.00	1	0.00	0.00	0	0.00	0.00
9x4x50	54	5	0.00	0.00	8	0.00	0.00	121	0.00	0.00	277	0.00	0.00	56	0.00	0.00
9x4x75	95	2,233	0.00	0.00	2,395	0.00	0.00	time	NA	NA	time	21.05	0.00	5,336	0.00	0.00
10x4x25	34	1	0.00	0.00	1	0.00	0.00	0	0.00	0.00	0	0.00	0.00	0	0.00	0.00
10x4x35	51	5	0.00	0.00	5	0.00	0.00	27	0.00	0.00	23	0.00	0.00	19	0.00	0.00
10x4x50	76	220	0.00	0.00	251	0.00	0.00	time	9.21	0.00	time	NA	NA	321	0.00	0.00
10x4x75	118	3,618	20.34	0.00	4,112	20.34	0.00	time	NA	NA	time	25.42	0.00	time	42.37	0.00
10x5x25	35	1	0.00	0.00	1	0.00	0.00	0	0.00	0.00	0	0.00	0.00	0	0.00	0.00
10x5x35	44	2	0.00	0.00	2	0.00	0.00	3	0.00	0.00	3	0.00	0.00	1	0.00	0.00
10x5x50	69	4	0.00	0.00	4	0.00	0.00	3	0.00	0.00	3	0.00	0.00	17	0.00	0.00
10x5x75	95	362	0.00	0.00	388	0.00	0.00	time	20.00	0.00	time	17.89	0.00	2,750	0.00	0.00
11x5x25	36	2	0.00	0.00	3	0.00	0.00	22	0.00	0.00	5	0.00	0.00	26	0.00	0.00
11x5x35	53	4	0.00	0.00	5	0.00	0.00	502	0.00	0.00	446	0.00	0.00	216	0.00	0.00
11x5x50	74	228	0.00	0.00	263	0.00	0.00	time	8.00	1.35	time	4.05	0.00	1,901	0.00	0.00
11x5x75	105	3,622	5.71	0.00	3,885	5.71	0.00	time	NA	NA	time	22.86	0.00	time	43.40	0.95
12x5x25	46	13	0.00	0.00	15	0.00	0.00	5,922	0.00	0.00	181	0.00	0.00	434	0.00	0.00
12x5x35	63	346	0.00	0.00	372	0.00	0.00	time	11.11	0.00	time	4.76	0.00	6,550	0.00	0.00
12x5x50	87	3,614	5.75	0.00	3,858	5.75	0.00	time	NA	NA	time	17.05	1.15	time	11.49	0.00
12x5x75	130	3,677	17.42	1.54	4,186	16.15	0.00	time	NA	NA	time	NA	NA	time	27.48	0.77
12x6x25	41	2	0.00	0.00	2	0.00	0.00	2	0.00	0.00	0	0.00	0.00	1	0.00	0.00
12x6x35	52	143	0.00	0.00	147	0.00	0.00	time	11.54	0.00	time	11.54	0.00	time	17.31	0.00
12x6x50	76	1,439	0.00	0.00	1,483	0.00	0.00	time	14.10	2.63	time	11.69	1.32	time	21.05	0.00
12x6x75	115	3,663	10.43	0.00	3,926	10.43	0.00	time	NA	NA	time	11.21	0.87	time	21.74	0.00
15x6x25	55	349	1.79	1.82	384	1.79	1.82	time	7.14	1.82	time	10.71	1.82	time	10.91	0.00
15x6x35	84	210	0.00	0.00	299	1.18	1.19	time	5.88	1.19	time	5.88	1.19	time	4.76	0.00
15x6x50	114	3,885	17.54	0.00	4,198	18.26	0.88	time	NA	NA	time	24.79	2.63	time	27.83	0.88
15x6x75	178	7,675	21.91	0.00	4,282	21.91	0.00	time	NA	NA	time	NA	NA	time	NA	NA
15x7x25	46	12	2.13	2.17	15	2.13	2.17	9	0.00	0.00	66	0.00	0.00	3	0.00	0.00
15x7x35	71	3,657	0.00	0.00	3,753	0.00	0.00	time	NA	NA	time	8.45	0.00	time	15.49	0.00
15x7x50	93	3,779	2.15	0.00	3,834	2.15	0.00	time	NA	NA	time	4.26	1.08	time	20.43	0.00
15x7x75	155	4,246	17.42	0.00	4,243	17.42	0.00	time	NA	NA	time	NA	NA	time	NA	NA
20x10x25	63	448	0.00	0.00	183	0.00	0.00	time	4.69	1.59	time	3.17	0.00	time	6.25	1.59
20x10x35	90	2,798	0.00	0.00	344	0.00	0.00	time	3.30	1.11	3,318	0.00	0.00	time	13.19	1.11
20x10x50	117	2,318	0.00	0.00	1,589	0.00	0.00	time	NA	NA	time	NA	NA	time	18.49	1.71
20x10x75	198	10,052	17.68	0.00	4,319	17.68	0.00	time	NA	NA	time	NA	NA	time	35.68	0.51
Arithmetic mean		1,567	3.60	0.23	1,321	3.67	0.30	6,649	3.72	0.36	6,322	6.06	0.30	5,304	8.89	0.20
Geometric mean		102	1.06	0.12	107	1.16	0.17	506	1.47	0.23	466	2.26	0.20	315	2.47	0.14
Optimal (#)			27			25			16			17			22	
Infeasible (#)			0			0			13			6			2	
Best UB (#)			36			34			21			27			31	

Table 12: Impact of using ILS matheuristic on the performance of the approximate algorithm

Instance	Best	CP-OT						MIP-OT					
		LS			ILS			LS			ILS		
		Time	*Gap*	*Dev*	*Time*	*Gap*	*Dev*	*Time*	*Gap*	*Dev*	*Time*	*Gap*	*Dev*
8x4x25	27	1	3.57	3.70	5	3.57	3.70	1	3.57	3.70	8	3.57	3.70
8x4x35	43	2	0.00	0.00	9	0.00	0.00	2	2.27	2.33	13	0.00	0.00
8x4x50	59	5	0.00	0.00	20	0.00	0.00	7	0.00	0.00	36	0.00	0.00
8x4x75	75	53	0.00	0.00	97	0.00	0.00	62	0.00	0.00	184	0.00	0.00
9x4x25	33	1	0.00	0.00	5	0.00	0.00	1	0.00	0.00	8	0.00	0.00
9x4x35	48	1	0.00	0.00	6	0.00	0.00	1	0.00	0.00	12	0.00	0.00
9x4x50	54	5	0.00	0.00	17	0.00	0.00	8	0.00	0.00	42	0.00	0.00
9x4x75	95	2,233	0.00	0.00	2,278	0.00	0.00	2,395	0.00	0.00	3,890	0.00	0.00
10x4x25	34	1	0.00	0.00	7	0.00	0.00	1	0.00	0.00	11	0.00	0.00
10x4x35	51	5	0.00	0.00	19	0.00	0.00	5	0.00	0.00	23	0.00	0.00
10x4x50	76	220	0.00	0.00	245	0.00	0.00	251	0.00	0.00	554	0.00	0.00
10x4x75	117	3,618	20.34	0.85	3,763	20.34	0.85	4,112	20.34	0.85	8,703	19.66	0.00
10x5x25	35	1	0.00	0.00	7	0.00	0.00	1	0.00	0.00	12	0.00	0.00
10x5x35	44	2	0.00	0.00	12	0.00	0.00	2	0.00	0.00	20	0.00	0.00
10x5x50	69	4	0.00	0.00	38	0.00	0.00	4	0.00	0.00	40	0.00	0.00
10x5x75	95	362	0.00	0.00	492	0.00	0.00	388	0.00	0.00	753	0.00	0.00
11x5x25	36	2	0.00	0.00	15	0.00	0.00	3	0.00	0.00	21	0.00	0.00
11x5x35	53	4	0.00	0.00	22	0.00	0.00	5	0.00	0.00	33	0.00	0.00
11x5x50	74	228	0.00	0.00	305	0.00	0.00	263	0.00	0.00	653	0.00	0.00
11x5x75	105	3,622	5.71	0.00	3,813	5.71	0.00	3,885	5.71	0.00	6,446	5.71	0.00
12x5x25	46	13	0.00	0.00	30	0.00	0.00	15	0.00	0.00	48	0.00	0.00
12x5x35	63	346	0.00	0.00	389	0.00	0.00	372	0.00	0.00	643	0.00	0.00
12x5x50	87	3,614	5.75	0.00	3,731	5.75	0.00	3,858	5.75	0.00	6,177	5.75	0.00
12x5x75	130	3,677	17.42	1.54	4,360	16.15	0.00	4,186	16.15	0.00	9,443	16.15	0.00
12x6x25	41	2	0.00	0.00	14	0.00	0.00	2	0.00	0.00	22	0.00	0.00
12x6x35	52	143	0.00	0.00	208	0.00	0.00	147	0.00	0.00	244	0.00	0.00
12x6x50	76	1,439	0.00	0.00	1,579	0.00	0.00	1,483	0.00	0.00	2,016	0.00	0.00
12x6x75	115	3,663	10.43	0.00	4,215	10.43	0.00	3,926	10.43	0.00	6,842	10.43	0.00
<u>15x6x25</u>	55	349	1.79	1.82	410	1.79	1.82	384	1.79	1.82	756	0.00	0.00
15x6x35	84	210	0.00	0.00	299	0.00	0.00	299	1.18	1.19	1,193	0.00	0.00
15x6x50	113	3,885	17.54	0.88	6,397	16.81	0.00	4,198	18.26	1.77	9,527	16.81	0.00
15x6x75	175	7,675	21.91	1.71	44,288	20.57	0.00	4,282	21.91	1.71	10,348	20.57	0.00
15x7x25	46	12	2.13	2.17	70	0.00	0.00	15	2.13	2.17	95	0.00	0.00
15x7x35	71	3,657	0.00	0.00	4,347	0.00	0.00	3,753	0.00	0.00	5,312	0.00	0.00
15x7x50	93	3,779	2.15	0.00	5,381	2.15	0.00	3,834	2.15	0.00	5,921	2.15	0.00
15x7x75	155	4,246	17.42	0.00	10,026	17.42	0.00	4,243	17.42	0.00	10,000	17.42	0.00
20x10x25	63	448	0.00	0.00	3,975	0.00	0.00	183	0.00	0.00	1,323	0.00	0.00
20x10x35	90	2,798	0.00	0.00	27,715	0.00	0.00	344	0.00	0.00	3,176	0.00	0.00
20x10x50	117	2,318	0.00	0.00	14,555	0.00	0.00	1,589	0.00	0.00	7,261	0.00	0.00
20x10x75	198	10,052	17.68	0.00	68,029	17.68	0.00	4,319	17.68	0.00	10,703	17.68	0.00
Arithmetic mean		1,567	3.60	0.32	5,280	3.46	0.16	1,321	3.67	0.39	2,813	3.40	0.09
Geometric mean		102	1.06	0.19	303	1.00	0.08	107	1.16	0.23	387	0.94	0.04
Optimal (#)		27			28			25			29		
Infeasible (#)		0			0			0			0		
Best UB (#)		33			37			32			39		

Table 13: Comparison of % deviation from the best bound obtained by our approximate algorithms and the metaheuristics from [59]

Instance	Best	Approximate algorithms				Heuristics of [59]					
		CP-OT		MIP-OT		ILS-VND			GRASP-VND		
		LS	*ILS*	*LS*	*ILS*	*avg*	*max*	*min*	*avg*	*max*	*min*
8x4x25	27	3.70	3.70	3.70	3.70	2.44	3.70	0.00	3.04	3.70	0.00
8x4x35	43	0.00	0.00	2.33	0.00	0.23	2.33	0.00	0.42	2.33	0.00
8x4x50	59	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8x4x75	75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	1.33	0.00
9x4x25	33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9x4x35	48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9x4x50	54	0.00	0.00	0.00	0.00	0.85	3.70	0.00	4.48	5.56	0.00
9x4x75	95	0.00	0.00	0.00	0.00	0.02	1.05	0.00	1.60	3.16	0.00
10x4x25	34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10x4x35	51	0.00	0.00	0.00	0.00	0.35	1.96	0.00	3.06	3.92	0.00
10x4x50	76	0.00	0.00	0.00	0.00	0.95	2.63	0.00	2.37	3.95	0.00
10x4x75	117	0.85	0.85	0.85	0.00	0.94	1.71	0.00	1.23	1.71	0.00
10x5x25	35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10x5x35	44	0.00	0.00	0.00	0.00	2.05	4.55	0.00	4.23	4.55	0.00
10x5x50	69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.43	1.45	0.00
10x5x75	95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11x5x25	36	0.00	0.00	0.00	0.00	2.11	2.78	0.00	2.50	2.78	0.00
11x5x35	53	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11x5x50	74	0.00	0.00	0.00	0.00	2.00	4.05	0.00	3.03	5.41	0.00
11x5x75	105	0.00	0.00	0.00	0.00	1.28	1.90	0.00	1.77	2.86	0.95
12x5x25	46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	2.17	0.00
12x5x35	63	0.00	0.00	0.00	0.00	1.59	3.17	0.00	3.21	4.76	1.59
12x5x50	87	0.00	0.00	0.00	0.00	1.79	3.45	1.15	2.41	3.45	1.15
12x5x75	130	1.54	0.00	0.00	0.00	1.55	3.08	0.00	2.51	3.85	0.00
12x6x25	41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.95	2.44	0.00
12x6x35	52	0.00	0.00	0.00	0.00	1.50	3.85	0.00	1.65	1.92	0.00
12x6x50	76	0.00	0.00	0.00	0.00	2.29	3.95	0.00	2.66	3.95	1.32
12x6x75	115	0.00	0.00	0.00	0.00	0.82	0.87	0.00	0.87	0.87	0.87
15x6x25	55	1.82	1.82	1.82	0.00	3.93	5.45	1.82	4.76	5.45	1.82
15x6x35	84	0.00	0.00	1.19	0.00	2.31	3.57	1.19	2.55	3.57	1.19
15x6x50	113	0.88	0.00	1.77	0.00	3.26	4.42	1.77	3.31	4.42	1.77
15x6x75	175	1.71	0.00	1.71	0.00	1.47	2.86	0.57	1.45	2.29	0.00
15x7x25	46	2.17	0.00	2.17	0.00	2.35	4.35	0.00	2.30	4.35	2.17
15x7x35	71	0.00	0.00	0.00	0.00	2.34	2.82	1.41	2.45	2.82	0.00
15x7x50	93	0.00	0.00	0.00	0.00	2.37	3.23	1.08	3.10	4.30	2.15
15x7x75	155	0.00	0.00	0.00	0.00	1.29	1.94	0.65	1.34	1.94	1.29
20x10x25	63	0.00	0.00	0.00	0.00	5.20	7.94	3.17	5.84	7.94	3.17
20x10x35	90	0.00	0.00	0.00	0.00	2.29	3.33	0.00	2.44	3.33	1.11
20x10x50	117	0.00	0.00	0.00	0.00	3.93	5.13	2.56	3.66	4.27	2.56
20x10x75	**197**	0.51	0.51	0.51	0.51	0.68	1.02	0.00	0.84	1.02	0.51
50x30x25	150	0.00	0.00	NA	NA	0.00	0.00	0.00	0.00	0.00	0.00
50x30x35	195	1.03	0.00	NA	NA	2.77	3.08	2.05	2.97	3.59	2.05
50x30x50	289	0.35	0.00	NA	NA	1.37	1.73	1.04	1.42	1.73	0.69
50x30x75	440	0.00	0.00	NA	NA	0.43	0.68	0.23	0.52	0.68	0.23
Arithmetic mean		0.33	0.16	0.40	0.11	1.33	2.28	0.42	1.88	2.68	0.60
Geometric mean		0.21	0.09	0.25	0.05	1.01	1.68	0.28	1.46	2.10	0.42
Best Bound (#)		34	40	31	38	12	12	31	8	8	26
Optimal Bound (#)		28	29	25	30	12	12	27	8	8	24