STOCHASTIC PATIENT APPOINTMENT SCHEDULING FOR CHEMOTHERAPY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

NUR BANU DEMİR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

FEBRUARY 2019

Approval of the thesis:

STOCHASTIC PATIENT APPOINTMENT SCHEDULING FOR CHEMOTHERAPY

submitted by **NUR BANU DEMİR** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ____________

Prof. Dr. Yasemin Serin
Head of Department, **Industrial Engineering** ____________

Assist. Prof. Dr. Melih Çelik
Supervisor, **Industrial Engineering, METU** ____________

Assist. Prof. Dr. Serhat Gül ____________
Co-supervisor, **Industrial Engineering, TED University**

Examining Committee Members:

Assoc. Prof. Dr. Seçil Savaşaneril
Industrial Engineering, METU ____________

Assist. Prof. Dr. Melih Çelik
Industrial Engineering, METU ____________

Assist. Prof. Dr. Serhat Gül
Industrial Engineering, TED University ____________

Assist. Prof. Dr. Sakine Batun
Industrial Engineering, METU ____________

Assist. Prof. Dr. Vedat Bayram
Industrial Engineering, TED University ____________

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Nur Banu Demir

Signature :

ABSTRACT

STOCHASTIC PATIENT APPOINTMENT SCHEDULING FOR CHEMOTHERAPY

Demir, Nur Banu
M.S., Department of Industrial Engineering
Supervisor: Assist. Prof. Dr. Melih Çelik
Co-Supervisor : Assist. Prof. Dr. Serhat Gül

February 2019, 77 pages

Chemotherapy appointment scheduling is a challenging problem due to uncertainty in pre-medication and infusion durations. We formulate a two-stage stochastic mixed integer programming model for chemotherapy appointment scheduling problem under the limited availability and number of nurses, and infusion chairs. The objective is to minimize the expected weighted sum of nurse overtime and patient waiting time. We sampled the pre-medication and infusion durations based on real data of a major oncology hospital. The computation times for the problem are significantly long even for the case of single-scenario problems. In order to strengthen the formulation, valid bounds and symmetry breaking constraints are incorporated. A Progressive Hedging Algorithm is implemented in order to solve the improved formulation. We enhance the algorithm through a penalty update method, cycle detection and variable fixing mechanisms, and linearization of the model objective function. We conduct numerical experiments to compare the progressive hedging algorithm with several scheduling heuristics from the relevant literature. We generate managerial insights related to the impact of the number of nurses and chairs on appointment schedules. Finally,

we estimate the value of stochastic solution to assess the significance of considering uncertainty.

Keywords: chemotherapy, scheduling, appointment scheduling, stochastic programming, progressive hedging

ÖZ

BELİRSİZ SÜRELER ALTINDA KEMOTERAPİ RANDEVULARININ ÇİZELGELENMESİ

Demir, Nur Banu
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Melih Çelik
Ortak Tez Yöneticisi : Dr. Öğr. Üyesi. Serhat Gül

Şubat 2019 , 77 sayfa

Kemoterapi randevu planlama problemleri hastaların premedikasyon ve infüzyon sürelerinin belirsizliğinden kaynaklı olarak zorlayıcı bir problemdir. Bu problemi çözmek için iki fazlı stokastik karışık tam sayı programlama tekniğinden yararlanıldı. Problemde hemşirelerin ve kemoterapi koltuklarının sayıları ve uygunlukları göz önüne alındı. Amaç hastaların bekleme sürelerinin ve hemşirelerin fazla mesailerinin beklenen ağırlıklı toplamının en azlanması olarak belirlendi. Premedikasyon ve infüzyon süreleri Hacettepe Onkoloji Hastanesi'nin gerçek verilerine dayalı olarak oluşturuldu. Bir senaryolu problemlerde bile çözüm süresinin yüksek olması sebebiyle formülasyon sınırlar ve simetriyi kıran kısıtlarla güçlendirildi. Hastaların sırasını ve randevu saatlerini belirleyebilmek için İlerlemeli Tedbir algoritması uygulandı. Algoritma ceza katsayısının güncellenmesi, döngülerin tespit edilmesi, değişkenlerin sabitlenmesi ve modelin amaç fonksiyonunun doğrusallaştırılması yöntemleri ile geliştirildi. İlerlemeli tedbir algoritması literatürdeki bazı sezgisel çizelgeleme yöntemleriyle sayısal deneyler yapılarak karşılaştırıldı. Hemşire ve kemoterapi koltuk sayılarının randevu çizelgelerine olan yönetimsel etkileri incelendi. Son olarak, stokastik çözümün değeri değerlendirilerek belirsizliği ele almanın önemi tespit edildi.

Anahtar Kelimeler: kemoterapi, çizelgeleme, randevu çizelgeleme, stokastik programlama, ilerlemeli tedbir

To all cancer patients...

ACKNOWLEDGMENTS

I would like to thank my advisors Assist. Prof. Dr. Melih Çelik and Assist. Prof. Dr. Serhat Gül for their endless support during my MS study. I learned a lot from them, they taught me how to conduct a research. They always encouraged me to make my research better with their excellent guidance. Both of my advisors are mentors for me in my academic career.

I would like the thank the examining committee members of my thesis Assoc. Prof. Dr. Seçil Savaşaneril, Assist. Prof. Dr. Sakine Batun, and Assist. Prof. Dr. Vedat Bayram for their valuable feedbacks about my research study.

I wish to thank my parents Meryem Demir and Recai Demir for their firm belief in me during my education. I would like to thank Deniz Altınpulluk for his patience throughout my thesis study. He always motivated me to go ahead and made me smile whatever the circumstance is. I would like to express my sincere gratitude to my friend Dr. Emine Gündoğdu for helping me to learn C++. Her knowledge contributed me a lot in the coding implementations of my thesis study. I am grateful to my dear friends Duygu Pamukçu and Kadir Biçe for not leaving me alone in the research lab at nights.

I would like to thank the manager and nurses of the Hacettepe Oncology Unit. They gave me a chance to observe the processes in the unit and shared their knowledge and experiences with me.

I feel very lucky for being an MS student and a research assistant at Middle East Technical University, Industrial Engineering Department. The profesors in the department helped me to improve myself. I would like to thank my professors Prof. Dr. Haldun Süral and Assist. Prof. Dr. Melih Çelik who had a lot influences on me to join the academia.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiv

LIST OF FIGURES xvi

LIST OF ABBREVIATIONS xvii

CHAPTERS

1 INTRODUCTION 1

2 LITERATURE REVIEW 5

2.1 Deterministic Chemotherapy Planning and Scheduling Problems . . . 6

2.2 Stochastic Chemotherapy Planning and Scheduling Problems 9

3 THE CHEMOTHERAPY APPOINTMENT SCHEDULING PROBLEM . . 13

3.1 Chemotherapy Operations in the Hacettepe Outpatient Chemotherapy Unit 13

3.2 A Two-Stage Stochastic Mixed-Integer Programming Model for the Chemotherapy Appointment Scheduling Problem 15

3.2.1 Background Information on Stochastic Programming 16

3.2.2 Two-Stage Stochastic Mixed-Integer Programming Formulation 17

3.2.3 Improvements in the Formulation 22

4 SOLUTION METHODOLOGY 25

4.1 The Progressive Hedging Algorithm 26

4.2 Literature Review on the Progressive Hedging Algorithm 27

4.3 Modifications on the Progressive Hedging Algorithm 31

4.3.1 Handling the Quadratic Term in the Objective Function 31

4.3.2 Choice of Penalty Parameter Update Method 33

4.3.3 Cycling and Majority Value Detection 34

4.3.4 Varying CPLEX optimality gap value in the subproblems . . . 36

4.3.5 Lower Bound Structure for Nurse Overtime 37

5 COMPUTATIONAL EXPERIMENTS 39

5.1 Generating Problem Instances 39

5.2 Solving Scenario Subproblems 41

5.2.1 Selection of a technique to solve nonlinear scenario-subproblems 41

5.2.2 Effect of symmetry-breaking constraints and bound on nurse overtime 41

5.2.3 Tests for varying CPLEX optimality gap value in the subproblems 42

5.3 Sensitivity analysis on LPHA parameters 43

5.3.1 Effect of α in the performance of LPHA 43

5.3.2 Effect of initial value of ρ in the performance of LPHA 45

5.3.3 Effect of upper limit on penalty parameter ρ_1^u on the performance of LPHA 46

5.3.4 Effect of iteration limit for changing ρ_1^u to ρ_2^u on the performance of the LPHA 47

5.4 Assessment of the LPHA Performance 48

5.4.1 Comparison with CPLEX 48

5.4.2 Comparison with Scheduling Heuristics 48

5.5 Sensitivity Analysis on Model Parameters 51

5.5.1 Varying value of $\boldsymbol{\lambda}$ 51

5.5.2 Varying values of number of nurses and chairs 53

5.6 Estimating the Value of Stochastic Solution (VSS) 54

6 CONCLUSION 57

REFERENCES 59

APPENDICES

A APPENDIX 65

A.1 Deterministic Equivalent Model 65

A.2 Computational Results 66

LIST OF TABLES

TABLES

Table 3.1 Notation 19

Table 5.1 Pre-medication and infusion intervals and the probabilites of observing the defined classes 40

Table 5.2 Effect of varying CPLEX optimality gap value on the performance of LPHA 42

Table 5.3 Effect of α change in the algorithm performance 44

Table 5.4 Effect of change in initial value of penalty parameter (ρ_1^u) in the performance of LPHA 45

Table 5.5 Effect of change in first upper limit of penalty parameter (ρ_1^u) on the performance of the LPHA 46

Table 5.6 Effect of change in iteration limit ($iterlimit$) in the algorithm performance 47

Table 5.7 Average gap values between the solutions of the LPHA and heuristics using different sequencing and job hedging levels 51

Table 5.8 The values of the average patient waiting time and nurse overtime in changing λ 52

Table 5.9 Sensitivity of the objective value and computational time to the number of chairs and nurses 54

Table 5.10 Average values of Value of stochastic solution (VSS) in changing λ values 55

Table A.1 The comparison of CPLEX and LPHA in terms of run times and solution quality 66

Table A.2 The objective function values and computational times in λ=0 67

Table A.3 The objective function values and computational times in λ=0.1 . . . 67

Table A.4 The objective function values and computational times in λ=0.2 . . . 68

Table A.5 The objective function values and computational times in λ=0.3 . . . 68

Table A.6 The objective function values and computational times in λ=0.4 . . . 69

Table A.7 The objective function values and computational times in λ=0.5 . . . 69

Table A.8 The objective function values and computational times in λ=0.6 . . . 70

Table A.9 The objective function values and computational times in λ=0.7 . . . 70

Table A.10The objective function values and computational times in λ=0.8 . . . 71

Table A.11The objective function values and computational times in λ=0.9 . . . 71

Table A.12The objective function values and computational times in λ=1 72

Table A.13Objective values and run times of job hedging heuristics for SPT rule 73

Table A.14Objective values and run times of job hedging heuristics for LPT rule 74

Table A.15Objective values and run times of job hedging heuristics for VAR rule 75

Table A.16Objective values and run times of job hedging heuristics for increasing CoV rule 76

Table A.17VSS values for 10 different instances in varying λ values 77

LIST OF FIGURES

FIGURES

Figure 3.1 Patient flow chart for treatment process 15

Figure 3.2 Chronology of Main Events in the Chemotherapy Unit 18

Figure 4.1 Lower Bound for Nurse Overtime within Progressive Hedging Algorithm 38

Figure 5.1 Average patient waiting time and nurse overtime values in changing λ 53

LIST OF ABBREVIATIONS

SMIP	Stochastic mixed integer programming
SMIP-R	Stochastic mixed integer programming reformulation
MDP	Markov decision processes
SOCP	Second-order cone programming
RP	Recourse Problem
EEV	Expected objective value of the expected value problem
VSS	Value of the stochastic solution
PHA	Progressive Hedging Algorithm
LPHA	Linearized Progressive Hedging Algorithm
SPT	Shortest processing time
LPT	Longest processing time
VAR	Increasing variance
CoV	Increasing coefficient of variation

CHAPTER 1

INTRODUCTION

Healthcare expenditures constitute a large share in the countries' economies. According to the statistics of the Organisation for Economic Co-operation and Development (OECD), the United States (US) has spent 17.1% of its gross domestic product on healthcare expenditures in 2017 [1]. On the other hand, budgets for healthcare expenditures are insufficient in developing countries, which results in a rate of death that could otherwise be prevented. In order to have a better healthcare system, resources should be correctly allocated to the clinics for satisying the needs of patients while reducing the expenditures. Since the resources dedicated to healthcare operations are limited, the need for efficient methods for planning araises every day. The importance of quantitative models and decision support tools has been realized to achieve efficient healthcare management. Hence, Operations Research plays an important role in improving healthcare operations.

Cancer is the second most prevalent cause of death globally according to statistics, after heart disase [2]. The estimated number of deaths from cancer was 9.6 million, and approximately 17 million new cancer cases are estimated to have appeared globally in 2018 [3]. According to the National Cancer Institute (NCI), it was also expected that 1,735,350 people were diagnosed as a cancer patient and 609,640 people will have lost their life due to cancer in the US in 2018. Furthermore, new cancer cases per year will increase to 23.6 million by 2030 [4]. In 2017, the estimated expenditures for the cancer treatment were $147.3 billion in the US. In Turkey, approximately 91,000 people die from cancer and 163,500 people are diagnosed as a cancer patient every year [5]. Since many people are affected by this disease, it is crucial to have planning systems which decrease costs and increase patient satisfaction at the same time.

Chemotherapy is a frequently used method in order to cure cancer patients. Chemotherapy drugs are injected to patients with the purpose of preventing the growth and spread of tumor cells in order to destroy cancerous tissues. Chemotherapy treatment is an exhausting process for patients, since side effects may be observed during or after the treatment. To reduce the intensity of the side effects, pre-medication drugs are injected to patients before initializing the treatment. After the pre-medication process, the patients receive chemotherapy drugs. These drugs should be given to patients based on predetermined frequency and doses.

Designing patient schedules is vitally important due to the limited capacity of the oncology clinics and time restrictions in the chemotherapy treatment. It is significant to note that oncology clinics should provide schedules that consider the trade-off between provider and patient satisfaction. Chemotherapy schedules are created in two phases in the literature and practice in general. Patients are assigned to days according to their treatment frequency in the first phase called *chemotherapy planning*. The latter phase is called *chemotherapy scheduling*. The patients are sequenced and their appointment times are set to create a schedule for a given day in the second phase. In this study, we concentrate on constructing daily schedules for chemotherapy patients.

There are two criteria that should be considered to evaluate the effectiveness of chemotherapy scheduling. These are nurse overtime and patient waiting time. Nurse overtime is undesirable for providers since it increases operating costs of the clinic. Working for more than the shift length may also decrease nurse satisfaction. The increased number of stressed out and dissatisfied nurses would lead to greater nurse turnover rates. Reducing patient waiting time is also important for clinics to improve patient satisfaction and service level. Since treatment durations are not actually known in advance of the treatment, predicted treatment durations are generally used for schedules. However, this may result in an undesirable amount of waiting time and overtime.

The most complicating factor for chemotherapy patient appointment scheduling is the uncertainty in treatment durations. The treatment durations are unknown due to reasons including a change in the prescription list according to patient's health status before initiating the treatment, complications with patients, and early treatment termination due to the patient not tolerating the treatment [6]. If the decision maker plans the daily schedule according to the longest possible treatment durations, patient waiting time can be reduced. However, this may lead to excessive nurse overtime. If the patients are scheduled based on the shortest possible treatment durations, this may result in high waiting times in the clinic. Therefore, the decision maker should handle the uncertainty in treatment durations wisely to avoid long waiting times and nurse overtime.

The limited availability of chairs and nurses in the system is another crucial factor in this problem. A patient should simultaneously seize a nurse and an infusion chair for the treatment. During the pre-medication stage, where the patient receives the pre-medication drugs if exist in the prescription, a nurse should always stay with the patient. During chemotherapy infusion, the drugs that are used in chemotherapy treatment are injected to patients. A single nurse can proctor multiple patients in this stage.

In this thesis, we study the problem of sequencing patients and setting appointment times (i.e., scheduled start time) for a chemotherapy unit by considering the availability of a limited number of nurses and chairs under uncertainty related to pre-medication and infusion durations. Decisions in our model include: (1) sequencing patients of a daily appointment list; (2) setting appointment times; (3) assignment of patients to nurses; and (4) assignment of patients to chairs. We modeled our chemotherapy appointment scheduling problem as a two-stage stochastic mixed integer programming (SMIP) formulation. We consider an objective function that minimizes the total expected cost of patient waiting and overtime across a large set of scenarios sampled through pre-medication and infusion time distributions. We propose a Progressive Hedging Algorithm (PHA) that utilizes the problem structure to find near-optimal chemotherapy schedules. In particular, we propose a penalty update method that considers convergence behavior of the primal and dual variables. The method includes a limit on penalty parameter whose value changes according to the iteration number. We test a cycle detection and variable fixing mechanism, and linearize the objective function to improve solution times of scenario subproblems. We also investigate the impact of varying CPLEX optimality gap for scenario subproblems. We compare the PHA with heuristics used in the relevant studies from the appointment scheduling literature. We solve problem instances based on data from a major oncology hospital. Our experiments provide insight into the issues related to the following questions.

1. What is the value of considering uncertainty in pre-medication and infusion durations when scheduling chemotherapy appointments?
2. What is the potential benefit of using the Progressive Hedging Algorithm over commonly used heuristics from the relevant appointment scheduling literature?
3. Which PHA routines and parameters must be carefully designed to enhance the algorithm performance and solution quality?
4. How does the nurse-to-chair ratio affect the trade-off between patient waiting time and nurse overtime?

The organization of the next chapters is as follows. In the next chapter, literature review and the main contributions of the thesis are presented. In Chapter 3, the problem description and model formulation are given. In Chapter 4, a brief literature review on the solution methodologies for the two-stage stochastic programs is provided. The progressive hedging algorithm and its implementation details are also presented in this chapter. The results of the computational experiments are demonstrated in Chapter 5. The concluding remarks and further research ideas are discussed at Chapter 6.

CHAPTER 2

LITERATURE REVIEW

In outpatient chemotherapy clinics, many different strategic, tactical and operational decisions can be addressed using operations research models. A detailed review of operations research studies on several different aspects of cancer care is provided in Saville et al. [7]. Chemotherapy planning and patient appointment scheduling have recently taken particular attention in this area. Deterministic and stochastic mathematical programming formulations, dynamic programming methods, Markov decision processes (MDP), heuristic approaches, simulation models have been used for studying chemotherapy planning and appointment scheduling problems. Our literature review specifically focuses on chemotherapy planning and scheduling problems. The reader is referred to Lame et al. [8] for an extensive review on the subject.

Since we study an outpatient chemotherapy appointment scheduling problem, articles from the general outpatient appointment scheduling literature may also help understand the nature of the problem. To keep our review compact, we refer the readers to Gupta and Denton [9], Cayirli and Veral [10], Javid et al. [11] for oupatient scheduling studies on primary care and specialty care. Furthermore, the differences between general outpatient scheduling and outpatient chemotherapy scheduling problems were discussed in Heshmat and Eltawil [12].

We do not restrict our review with the studies that focuses on only the treatment phase of chemotherapy (i.e. pre-medication and infusion). Some studies also consider the steps that need to be completed before the treatment starts. These steps are the following: (1) blood tests, (2) oncologist evaluation, (3) drug preparation in the pharmacy. The third step is always conducted on the day of treatment. However, the first and second steps may be conducted in an earlier day in some clinics. A patient's blood test result determines whether or not the patient is ready to go through the treatment. After the blood test results are received from the laboratory, the treatment of the patient is either approved or postponed after the oncologist evaluates the results. In case the treatment is approved, a drug preparation order is provided to the pharmacist. The treatment can start when the chemotherapy drug is prepared.

The remainder of the literature review is organized as follows: We investigate deterministic chemotherapy planning and scheduling problems and their solution approaches at Section 2.1. In Section 2.2, studies on stochastic chemotherapy planning and scheduling are discussed.

2.1 Deterministic Chemotherapy Planning and Scheduling Problems

In this category of articles, values of parameters for chemotherapy planning and scheduling problems are assumed to be deterministic. These parameters represent pre-medication and infusion durations, punctuality of patients, cancellations, results of the lab tests, and number of available nurses.

Turkcan et al. [13] handled chemotherapy planning and scheduling problems in a hierarchical manner. In the first formulation, the aim is to minimize treatment delays while assigning the new patients to treatment days. It was assumed that the plans of the patients that are already scheduled should be preserved. After determining the daily patient lists, the authors focused on the daily patient scheduling problem by considering resource availabilities and acuity levels of the patients. They split the whole time period into slots and assumed that the treatment length of the patient is predetermined in the planning phase. In a similar context, a multi-criteria optimization problem was studied by Condotta and Shakhlevich [14]. In this study, patients were assigned to days, which also determined the following appointment days of the patients, as the number of days between consecutive visits was known. Next, the appointment time setting, and patient-nurse allocations were made before the treatment day. The objectives are to minimize the patient waiting time and maximum nurse workload. The authors also revised the daily schedules on the days of treatment.

Heshmat and Eltawil [15] studied the chemotherapy planning problem with two aspects. The first one is finding optimal dosages of the chemotherapy drugs, whereas the second one is the assigning the new patients' first treatments to future days by considering the existing patients in the schedule. The aim was to minimize the amount of cancerous tissues, idle and overtime of nurses, and treatment delays of the patients. The authors also considered the pharmacists' working hours and proposed a sequential approach to determine the drug schedules and first-treatment days of the new patients.

The performances of constraint programming and MIP models were compared in [16]. It was shown that constraint programming outperforms its MIP counterpart by a range between 14% and 37% with respect to the makespan. An important constraint in their model limits the number of patients served by a nurse to four at any point in time. Oncologist evaluation and pharmacy preparation stages were studied in addition to the treatment stage in the model developed in [17]. Nurses were always assumed to be ready for the treatment, so the scarce resources were the oncologists and chemotherapy chairs. The authors compared the performances of the Lagrangian relaxation-based heuristics and local search heuristics, and concluded that Lagrangian relaxation-based heuristics perform better in terms of patient waiting times.

Liang et al. ([18]) discussed two different care delivery models for treatment of patients, namely functional and primary care delivery models. In the *functional care delivery model*, it is allowed to assign a different nurse to a patient at each treatment visit. On the other hand, a specific nurse is responsible for the patient's treatment in the *primary care delivery model*. The workload among nurses tends to show higher variability in the primary care delivery model, compared to the functional care delivery model. The authors constructed a mathematical programming model to minimize nurse overtime and total excess workload for the case of primary care delivery model. In their formulation, acuity levels of patients are taken into account and there is an upper bound on the assigned accuity level for each nurse in a time slot. Skill levels of the nurses are also included in the model. For the functional care delivery model, they aimed to minimize the total nurse overtime and patient waiting time within a multi-objective optimization model framework.

Santibanez et al. [19] summarized the difficulties of scheduling chemotherapy appointments and consequences of inefficient scheduling in terms of their impact on chemotherapy patients. While assigning appointment times to patients, the planner should consider the preferences of patients, capacity of the pharmacy and laboratory, schedules of the oncologists, and overtime for nurses. To determine the chemotherapy schedules, a timetable was prepared in [20], where it was assumed that laboratory tests and oncologist evaluation are completed on the previous day of the treatment. Pharmacist availability is an important concern in the scheduling process. As a conclusion, pharmacist and nurse efficiencies were improved, and the patient waiting times were reduced in the oncology clinic.

Heshmat et al. [21] proposed a two-stage solution approach for the chemotherapy appointment scheduling problem. In the first stage, patients were grouped using clustering algorithms. In the latter stage, the cluster of patients were assigned to nurses, chairs, and time slots by solving a mathematical programming model. Huggins et al. [22] considered a mathematical programming model to maximize chair utilization while considering the workloads of the pharmacists and nurses. They constructed two phases in the solution approach. In the first phase, a mathematical programming model was developed to assign patients to time slots. In the second phase, results of the mathematical programming model were validated with a simulation study. Two heuristics were developed to assign patients to the infusion chairs by Sevinc at al. [23]. They decomposed the problem into two phases. In the first phase, the laboratory tests are scheduled. An algorithm was constructed to determine the daily patient list for the laboratory. If a patient's health status is not well enough to take chemotherapy treatment, that patient's lab test is shifted to another day. The patients who are able to obtain an oncologist approval according to their test results are then assigned to infusion chairs in the second phase. To obtain a solution to this scheduling problem, the authors benefited from multiple knapsack problem heuristics.

The makespan and weighted flow time were minimized in [24] to schedule appointments of the chemotherapy patients. The authors expressed deterministic treatment durations in terms of time slots by assuming that a nurse can make only one setup at a given time, as in our study. Other aspects considered in the study are the patient priorities and lunch-coffee breaks of the nurses. The authors solved their appointment scheduling problem by formulating an integer programming model.

Our study differs from the studies summarized in this section since our study considers uncertainty in the durations of chemotherapy pre-medication and infusion.

2.2 Stochastic Chemotherapy Planning and Scheduling Problems

There are various parameters that might be considered uncertain in chemotherapy planning and scheduling problems. The articles discussed below consider the uncertainty associated with chemotherapy treatment and drug preparation durations, results of the lab tests, punctuality of patients, nurse availability, and acuity levels of the patients.

Gocgun and Puterman [25] used an MDP formulation to dynamically assign patients to future days while considering target treatment days of the patients. Since chemotherapy treatment is highly sensitive to the deviation from the target treatment day, the authors penalized the deviation from the target days. Moreover, they ensured that the time window constraints defined for the treatments were not violated. They obtained solutions for the MDP formulation using a linear programming based approximate dynamic programming approach. Alvarado and Ntaimo [26] formulated three different mean-risk stochastic integer programming models for the chemotherapy appointment planning and scheduling problem. Acuity levels, treatment durations, and nurse availability were represented by stochastic parameters in their problem description. Their model was formulated to provide a solution for a single patient. In other words, the authors assumed that the appointments of earlier patients were already booked. They made their planning and scheduling decision for the current patient by considering the remaining time slots in the future days. The objective was to minimize the deviation from the target treatment start day for the new patient, patient waiting time, and nurse overtime. In our study, we consider planning the daily schedules of multiple patients without using time slots.

In a thesis study, Tanaka [27] assigned patients to the infusion chairs using online bin packing heuristics. It was assumed that each infusion chair has a seperate patient list for the treatments. In other words, a patient should wait until a specific chair is available for the treatment. Due to this reason, a single server model for each infusion chair was constructed. Different types of lab tests before the infusion were considered. A patient might have a scheduled lab test, and the durations of the lab tests may vary. The uncertainty in drug preparation and nursing durations before the infusion, such as taking vitals and assessment was taken into account. However, the study assumed that the treatment durations including pre-medication and chemotherapy infusion are deterministic. Different bin packing heuristics were implemented to solve the problem.

Mandelbaum et al. [28] considered uncertain treatment durations and unpunctual patients. They constructed a data-driven approach for the problem of patient appointment scheduling, assuming that the infusion chairs are the only servers of the system. Therefore, the impact of limited nurse availability on a schedule is not considered in this study.

A discrete event simulation model was developed to see the effect of different operational decisions on the patient waiting time, clinic overtime, and resource utilization in an outpatient oncology clinic in [29]. The authors included oncology visits, lab tests, drug preparation, and chemotherapy treatments in their model. The aim was to develop a coordinated appointment scheduling system by considering the flow of the patients. The stochastic elements in the simulation model included unpunctual arrivals of patients and all service durations. However, the uncertainty was not considered in the mathematical programming model used to create schedules.

The study that is most similar to the one in this thesis work was conducted by Castaing et al. [6], where the authors constructed a two-stage stochastic integer program to determine patient appointment times in an outpatient chemotherapy clinic in order to minimize the expected patient waiting time and total time required for all treatments (i.e., clinic closure time). The authors assumed that an initial schedule was already created. Their model allows to make small revisions on the existing schedule as they assumed the sequence of patients can not be changed. They set appointment times for a given sequence of patients and considered multiple chairs, but a single nurse in the clinic. The binary variables in the second stage makes the model difficult to solve. Therefore, a two-phase heuristic, which facilitates setting the values of those variables, was implemented to solve the problem. This study differs from Castaing et al. [6] in a number of ways. First, we do not make the restricting assumption that the patient sequence is fixed. Hence, our model does more than the refinement of the existing schedules. The model can be used to create an initial schedule. Moreover, we consider multiple nurses in the clinic. A patient can be assigned to any of the nurses for treatment. Therefore, our model is applicable to clinics operating based on functional care delivery model. On the other hand, the model of Castaing et al. [6] can be used by clinics functioning according to primary care delivery model. Furthermore, their model must be solved separately for each nurse, as it considers a single nurse.

CHAPTER 3

THE CHEMOTHERAPY APPOINTMENT SCHEDULING PROBLEM

In this thesis, we define the Chemotherapy Appointment Scheduling Problem based on our observations of the chemotherapy operations at the Hacettepe University Oncology Hospital in Ankara, Turkey, which is one of the prominent oncology centers in the country.

3.1 Chemotherapy Operations in the Hacettepe Outpatient Chemotherapy Unit

The Hacettepe Outpatient Chemotherapy Unit consists of 32 chemotherapy chairs, 28 of which are used for treatments. The remaining four chairs are reserved for supportive care. The number of patients receiving treatment varies between 60 and 85 on a given day. On average, 10 nurses, including a head nurse, provide service to patients each day. A daily shift starts at 8:00 and ends at 17:00 (unless overtime is needed), with a lunch break between 12:00-13:00. The daily schedule of patients is arranged by the head nurse, who assigns appointment times, chairs and nurses to the patients.

Our observations at the chemotherapy unit aimed to capture the operations conducted until and during the chemotherapy treatment. The blood tests and oncologist evaluations are carried out on a different day before the treatment. The results of these tests are examined by the oncologist to decide whether the patient has convenient health status to receive a chemotherapy treatment. When the oncologist approves the treatment protocol, she updates or confirms the dosages of the drugs. The drugs are then prepared in the pharmacy lab before the patient arrives at the hospital for her treatment. During each treatment visit, a patient receives two types of medications: (i) pre-medication drugs, which are injected prior to the chemotherapy infusion to help prevent side effects; and (ii) chemotherapy infusion drugs, which are used for cancer treatment.

There are certain time slots that the patients are assigned to according to their estimated treatment duration, which is predicted according to the types and dosages of the drugs injected. During a day, the unit makes use of four time slots: 8:00-10:30, 10:30-12:00, 13:00-15:30 and 15:30-17:00. If the estimated treatment duration of a patient is less than the length of a slot, the patient is given an appointment time so that the expected end of treatment is within the same slot. Otherwise, the patient can be assigned to more than one slot. If the estimated treatment duration of a patient is longer than those of other patients, the head nurse prefers to assign this patient to the beginning of the workday in order to prevent overtime. In the rare case that the estimated duration of the treatment exceeds 270 minutes, the appointment is generally split into two parts, one in the morning and one in the afternoon. Given the uncertainties in the treatment times, the two main objectives in designing the schedule are to ensure that (i) each patient can actually start their treatment at the pre-determined appointment time without waiting, and (ii) all patients are treated before 17:00, thereby avoiding overtime.

The treatment of a patient consists of a sequence of events, as illustrated in Figure 3.1. A patient that arrives to the unit is immediately registered and becomes available for treatment at the appointment time. If a nurse and/or chair is not available at the time, the patient has to wait for the treatment. A nurse calls the patient when an infusion chair and the nurse both become available. After the patient is seated in a chemotherapy chair, the nurse measures the fever and blood pressure, and establishes a vascular access to start the pre-medication process. Note that a small minority of the patients may not need pre-medication drugs. However, the vast majority starts the infusion process after the pre-medication process ends. When the infusion process is over, the patient is discharged, and the nurse calls the next patient to the chair that has become available.

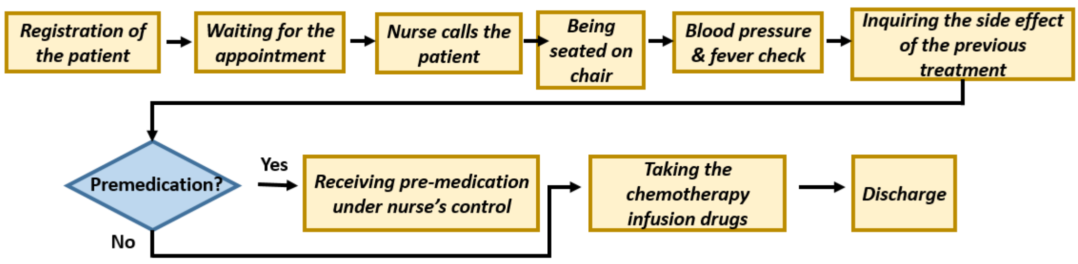

Figure 3.1: Patient flow chart for treatment process

In general, each nurse may be responsible for up to four patients at a given time. The Hacettepe Outpatient Chemotherapy Unit applies a modified form of the *functional care delivery*, where in line with this procedure each patient can be assigned to different nurses in subsequent visits. However, nurse-patient assignments in the clinic also consider the type of cancer and expected duration of treatment, which may limit the pool of nurses that can be assigned to the patient. The balance of both criteria is aimed to be satisfied to create equity between nurses.

3.2 A Two-Stage Stochastic Mixed-Integer Programming Model for the Chemotherapy Appointment Scheduling Problem

We formulate a two-stage stochastic mixed-integer programming (SMIP) model for the Chemotherapy Appointment Scheduling Problem, motivated by the operations at the Hacettepe Outpatient Chemotherapy Unit. In particular, we study the problem of sequencing patients and setting appointment times for a chemotherapy unit by considering the availability of nurses and chairs under uncertainty about pre-medication and infusion durations. We sequence patients of a daily appointment list, set appointment times, and assign patients to nurses and chairs in our model. We consider an objective function that minimizes a weighted combination of patient waiting time and overtime.

In Section 3.2.1, we provide background information on two-stage stochastic programming, followed by our assumptions and the details of our stochastic programming model in Section 3.2.2. We end this section by proposing a number of valid inequalities and bounds to strengthen the model in Section 3.2.3.

3.2.1 Background Information on Stochastic Programming

Stochastic Programming is a method for modeling optimization problems that include uncertainty. The uncertainty is included in the model with different data sets of stochastic parameters. The uncertain parameters are represented by defining *scenarios*, and each scenario is associated with a probability of occurence. The aim is to minimize (or maximize) the expected value of the objective function, which depends on the decisions and realizations of random variables at each stage.

The most commonly used form of a stochastic program is a *two-stage stochastic programming* model, where the uncertainty on the random variables is fully revealed at an instant. The decisions that are made without full information of the random events are called *first-stage decisions*. After uncertainty is revealed, the *second-stage decisions* are taken. The second-stage decisions depend on the first-stage decisions and realizations of scenarios. The general representation of the two-stage stochastic linear program is given as follows:

$$\min \text{ z} = c^T x + E_\xi[\min q(\omega)^T y(\omega)] \tag{3.1}$$

$$\text{s.t. } Ax = b \tag{3.2}$$

$$T(\omega)x + Wy(\omega) = h(\omega) \tag{3.3}$$

$$x \geq 0, y \geq 0 \tag{3.4}$$

In this model, the first-stage decision vector is represented by x, and the first stage costs and parameters are given by vectors c, b, and matrix A. The second-stage decision vector is denoted by y and the uncertainty is shown with the random events $\omega \in \Omega$. $q(\omega)$, $h(\omega)$ and $T(\omega)$ are the parameters of the second-stage problem. The "here-and-now" decision x is made before the uncertainty about future realizations of ξ is resolved. After the actual realization of ξ becomes known, the second stage problem is optimized [30].

The objective function (3.1) includes a mathematical expectation with respect to ξ. Since it may be challenging to compute the expected value, the following deterministic equivalent formulation can be constructed:

$$\text{min z} = c^T x + \mathcal{Q}(x) \tag{3.5}$$

$$\text{s.t. } Ax = b \tag{3.6}$$

$$x \geq 0 \tag{3.7}$$

where $\mathcal{Q}(x)$ is defined as the recourse function:

$$\mathcal{Q}(x) = E_{\xi}\mathrm{Q}(x, \xi(\omega)) \tag{3.8}$$

$$\mathrm{Q}(x, \xi(\omega)) = \min_y \{q(\omega)^T y \mid Wy = h(\omega) - T(\omega)x, y \geq 0\} \tag{3.9}$$

3.2.2 Two-Stage Stochastic Mixed-Integer Programming Formulation

In this section, we discuss our two-stage SMIP model for the chemotherapy appointment scheduling problem. The advantage of using SMIP model can be solving the stochastic program using the MIP formulation. We first list the assumptions made in the model:

Assumptions:

- We ignore some of the events shown in Figure 3.1, and focus on the three main events in between the patient appointment time and treatment completion time in the model. The events we consider include patient waiting, pre-medication, and infusion (see Figure 3.2).
- Patients can be treated by any available nurse, as is the case for *functional care delivery*. Nurses are also assumed to be identical.
- A nurse can perform only one patient's pre-medication process at any time.
- A nurse can proctor the infusion process of multiple patients while conducting the pre-medication process of a patient.
- Patients become ready exactly on the appointment time for treatment.
- Each patient's health status is well enough to complete the treatment. Therefore, we omit the exceptional cases that the a nurse and/or chair may become idle due to early termination of treatments.

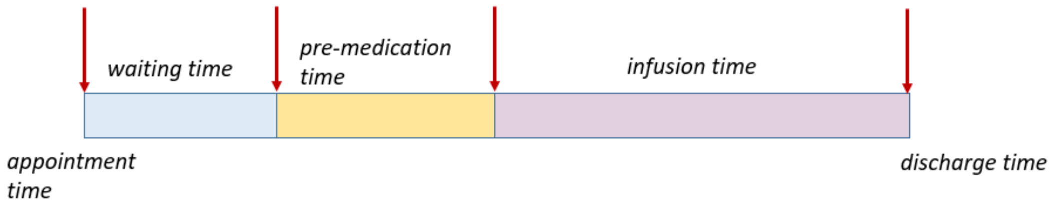

Figure 3.2: Chronology of Main Events in the Chemotherapy Unit

At the first stage of the model, patients are sequenced and their appointment times are set. Then, pre-medication and infusion durations are realized. At the second-stage, patients are assigned to nurses and chairs. Note that nurse and chair assignments are trivial, because each patient is assigned to the first available chair and nurse in all scenarios. Furthermore, the sequence of patients is not allowed to be changed in the second stage. The remaining decision variables at the second stage help determine patient waiting time, discharge time, and nurse overtime.

Nurse overtime and patient waiting time are the criteria considered in the model. Overtime is calculated separateley for each nurse that works for more than the planned shift length. Overtime is also a surrogate measure for idle time due to the positive correlation between these two values. Chairs become idle due to two reasons. First, treatment of a patient may finish earlier than the appointment time of the subsequent patient. Second, longer than expected durations of pre-medications may prevent nurses from initiating the treatment of a waiting patient. Waiting occurs when the treatment starts later than the appointment time of the patient, which may happen when pre-medications or infusions for the previous patients may last longer than expected. The unavailability of nurses or chairs may also cause waiting.

We assume a finite set of scenarios representing uncertainty in pre-medication and infusion durations. Given these scenarios, we formulate the following two-stage SMIP model for our problem based on sets, parameters, first and second stage decision variables summarized in Table 3.1.

Table 3.1: Notation

	Sets
I	Set of patients
C	Set of chairs
Ω	Set of scenarios
N	Set of nurses
	Parameters
s_i^ω	preparation time of patient $i \in I$ in scenario $\omega \in \Omega$
t_i^ω	infusion length of patient $i \in I$ in scenario $\omega \in \Omega$
p^ω	probability of scenario $\omega \in \Omega$
H	Shift duration of nurses
L	Overtime limit of nurses
λ	tradeoff parameter in the objective function in the interval $[0, 1]$
	First Stage Decision Variables
b_{ij}	$= \begin{cases} 1, & \text{if patient } i \in I \text{ precedes patient } j \in I \text{ in daily appointment list} \\ 0, & \text{otherwise} \end{cases}$
a_i	appointment time of patient $i \in I$
	Second Stage Decision Variables
x_{in}^ω	$= \begin{cases} 1, & \text{if patient } i \in I \text{ is assigned to nurse } n \in N \text{ in scenario } \omega \in \Omega \\ 0, & \text{otherwise} \end{cases}$
y_{ic}^ω	$= \begin{cases} 1, & \text{if patient } i \in I \text{ is assigned to chair } c \in C \text{ in scenario } \omega \in \Omega \\ 0, & \text{otherwise} \end{cases}$
w_i^ω	waiting time of patient $i \in I$ in scenario $\omega \in \Omega$
d_i^ω	discharge time of patient $i \in I$ in scenario $\omega \in \Omega$
O_n^ω	overtime of nurse $n \in N$ in scenario $\omega \in \Omega$

$$min \quad \mathcal{Q}(\textbf{a,b}) \tag{3.10}$$

$$b_{ij} + b_{ji} = 1 \qquad \forall i, j \in I, j > i \tag{3.11}$$

$$a_j \geq a_i - M(1 - b_{ij}) \qquad \forall i, j \in I, j \neq i \tag{3.12}$$

$$b_{ij} \in \{0, 1\} \qquad \forall i, j \in I \tag{3.13}$$

$$a_i : integer \qquad \forall i \in I \tag{3.14}$$

where

$$\mathcal{Q}(\textbf{a,b}) = E_\xi[\mathrm{Q}(\textbf{a,b}, \xi(\omega))]$$

is the expected recourse function, and

$$\mathrm{Q}(\textbf{a,b}, \xi(\omega)) = \min\left\{\lambda \sum_{i\in I} p^\omega w_i^\omega + (1 - \lambda) \sum_{n\in N} p^\omega O_n^\omega\right\}$$

$$\sum_{n\in N} x_{in}^\omega = 1 \qquad \forall i \in I \tag{3.15}$$

$$\sum_{c\in C} y_{ic}^\omega = 1 \qquad \forall i \in I \tag{3.16}$$

$$a_i + w_i^\omega + s_i^\omega + t_i^\omega = d_i^\omega \qquad \forall i \in I \tag{3.17}$$

$$a_j + w_j^\omega \geq a_i + w_i^\omega + s_i^\omega - M(3 - b_{ij} - x_{in}^\omega - x_{jn}^\omega) \qquad \forall i, j \in I, j \neq i, \forall n \in N \tag{3.18}$$

$$a_j + w_j^\omega \geq d_i^w - M(3 - b_{ij} - y_{ic}^\omega - y_{jc}^\omega) \qquad \forall i, j \in I, j \neq i, \forall c \in C \tag{3.19}$$

$$a_j + w_j^\omega \geq a_i + w_i^\omega - M(1 - b_{ij}) \qquad \forall i, j \in I, j \neq i \tag{3.20}$$

$$O_n^\omega \geq d_i^\omega - H - M(1 - x_{in}^\omega) \qquad \forall i \in I, \forall n \in N \tag{3.21}$$

$$O_n^\omega \leq L \qquad \forall n \in N \tag{3.22}$$

$$d_i^\omega, w_i^\omega \geq 0 \qquad \forall i \in I \tag{3.23}$$

$$O_n^\omega \geq 0 \qquad \forall n \in N \tag{3.24}$$

$$x_{in}^\omega \in \{0, 1\} \qquad \forall i \in I, \forall n \in N \tag{3.25}$$

$$y_{ic}^\omega \in \{0, 1\} \qquad \forall i \in I, \forall c \in C \tag{3.26}$$

The objective function in (3.10) only includes the expected second-stage cost, since there is no contribution from the first stage to the objective function. The expected second-stage cost aims to minimize the expected weighted sum of total patient waiting time and nurse overtime. First-stage constraints are given by (3.11)-(3.14). The patient precedence relations are determined with constraints (3.11). If patient $i \in I$ is scheduled before the patient $j \in I$, the corresponding b_{ij} value should be equal to 1. Constraint (3.12) establishes the relationship between the binary precedence variables and appointment times, in that if patient $i \in I$ precedes patient $j \in I$ in the list, then the appointment time of patient $i \in I$ should be no later than that of patient $j \in I$. Binary and integrality restrictions on the first-stage variables are represented in constraints (3.13) and (3.14), respectively.

In the second stage, nurse and chair assignments are made based on the realized treatment durations. For each scenario representing a set of chemotherapy durations, a subproblem is solved to obtain second stage decisions. The constraints for the second stage scenario subproblems are expressed by (3.15)-(3.26). Constraints (3.15) and (3.16) enforce every patient to be assigned to exactly one nurse and one chair, respectively. Constraints (3.17) calculate the patient discharge time, which is equal to the sum of the patient appointment time, waiting time, and durations of pre-medication and infusion. Constraints (3.18) ensure that if patient $i \in I$ and $j \in I$ are assigned to the same nurse, and patient $i \in I$ is scheduled before $j \in I$, then the treatment start time of patient $j \in I$ should be after the end of pre-medication of patient $i \in I$. Similar relation also holds for chair-precedence relationship, which is demonstrated in constraints (3.19). If two patients are assigned to the same chair, the latter one's treatment may begin only after the discharge time of the former one. By constraints (3.20), a patient's treatment start time should be earlier than those of the patients that are scheduled later. Lastly, overtime of a nurse should be either zero or the difference between the discharge time of the last discharged patient assigned to this nurse and the shift length, as determined by constraints (3.21). In oncology units, it is desired to have limited overtime to enhance nurses' working conditions and preserve equity among the nurses. A predetermined bound is included to ensure this in the model, which also tightens the feasible region of the formulation. The associated constraint is provided in equation (3.22). The remaining constraints are the sign and binary restrictions on the second-stage decision variables.

3.2.3 Improvements in the Formulation

Based on the preliminary runs, it is observed that solving even one-scenario problem instances necessitates significantly long solution times. Consequently, symmetry-breaking constraints and valid bounds are added in order to strenghten the formulation.

The first set of constraints relates to breaking symmetry with respect to the overtime of nurses. Since nurses are assumed to be identical, the schedules of the nurses are interchangable. By convention, we enforce the overtime of nurse with index 1 to be no less than those of the other nurses. In this way, overtime amount may be assigned in non-increasing order of nurse indices. The mathematical representation of this constraint is as in equation (3.27):

$$O_n^\omega \geq O_{n+1}^\omega \qquad \forall n \leq |N| - 1, \forall \omega \in \Omega \tag{3.27}$$

Note that balancing nurse workload is not among the main objectives in our study. However, the upper limit introduced by constraint (3.22) defines an acceptable bound and hence prevents excessive imbalance. Therefore, adding constraint (3.27) into the model results in an implementable solution. Furthermore, the preferred nurses for overtime work can be changed each day, since the model proposes a schedule for a single day.

Another issue with symmetry occurs while assigning patients to the nurses. To overcome this issue, the first patient scheduled is assigned to nurse 1. If patient $i \in I$ is the first patient in the schedule, then the summation of the b_{ij} values over $j \in I$ should be $|I| - 1$, where $|I|$ refers to the size of the patient set. The mathematical representation of this constraint is given in equation (3.28). For the first patient in the schedule, the left-hand-side of the constraint is equal to 1, forcing x_{i1}^ω to be equal to 1 for patient $i \in I$ for nurse 1 in every $\omega \in \Omega$. A similar set of constraints that assign the first patient scheduled to chair 1 is not observed to have substantial effect in the solution times.

$$\sum_{j \neq i} b_{ij} - (|I| - 2) \leq x_{i1}^{\omega} \qquad \forall i \in I, \omega \in \Omega \tag{3.28}$$

Lastly, a lower bound on nurse overtime is defined using the structure of the problem. If one can equally distribute the summation of treatment durations to chairs, the resulting average total treatment time per chair provides the smallest possible value for the maximum of patient discharge times. Therefore, the calculated average value is a lower bound for the sum of nurse shift time and overtime of nurse 1, since the overtime of nurse 1 is at least as high as all others according to equation (3.27). The mathematical representation of this constraint is shown in equation (3.29).

$$O_1^{\omega} + H \geq \frac{\sum_{i \in I}(s_i^{\omega} + t_i^{\omega})}{|C|} \qquad \forall \omega \in \Omega \tag{3.29}$$

As proposed in [31], the appointment and waiting times of the first patient in the schedule can be assigned to zero to reduce the number of decision variables. According to the results of preliminary experiments, adding such time assignments does not have a significant effect to reduce the computation times, and thus these are not included in the formulation.

Equations (3.27)-(3.29) are added to the main model and used in the computational experiments.

CHAPTER 4

SOLUTION METHODOLOGY

SMIP models are in general computationally demanding mainly due to the large amount of variables and constraints depending on the number of scenarios [32]. Decomposition based solution methods are generally utilized to solve such models in the literature. The well-known stage decomposition algorithm proposed by Slyke and Wets [33], *L-shaped method*, cannot be applied on stochastic programs in case there are binary variables in the second stage as in our SMIP formulation. To handle those cases, Laporte and Louveaux [34] extended the method and proposed *Integer L-shaped algorithm*. However, one should assume that only binary variables exist in the first stage to implement the algorithm. Some other solution methods concern the difficulty associated with having binary variables in the second stage through approaches based on value function and set convexification [35], [36]. These disjunctive decomposition-based branch-and-cut algorithms are known to be very effective, but they can also be applied on models with only binary variables in the first stage. In our formulation, the first stage includes both binary and general integer variables.

Scenario decomposition based approaches are also frequently used to solve SMIP models. The Progressive Hedging Algorithm (PHA) has been increasingly preferred among them (Gul et al. [37], Hvattum et al. [38], Gonçalves et al. [39]). The PHA, proposed by Rockafellar and Wets [40], is based on augmented Lagrangian relaxation technique. The algorithm decomposes the problem into single-scenario subproblems, and aims to obtain an *admissible* solution that is feasible for all scenarios by aggregating the scenario solutions at each iteration. It requires reformulation of the model to attain a structure appropriate for scenario decomposition. It then relaxes the constraints that enforce the first-stage decisions to be the same for every scenario in the reformulation and penalizes the violation of these constraints by introducing a Lagrangian term and quadratic penalty term in the objective function. The algorithm is shown to converge to the optimal solution for convex models. Since our model includes general integer and binary decision variables, it is not convex. Therefore, PHA is used as a heuristic approach for the Chemotherapy Appointment Scheduling Problem. Note that there are several evidences in the literature showing that the PHA is a good heuristic for SMIP models (Gul et al. [37], Crainic et al. [41], Watson and Woodruff [42])

In the next section, general steps of the PHA are given, followed by a literature review on different applications of the PHA. We make use of a number of approaches from the literature to modify the PHA, which is discussed in the last part of this chapter.

4.1 The Progressive Hedging Algorithm

We first reformulate our SMIP model to facilitate scenario decomposition. In the reformulation for PHA, which we call as SMIP-R (see Appendix A.1 for the model), *non-anticipativity constraints* are explicitly formulated. To formulate these constraints, a copy of a first-stage variable must be created for each scenario. The constraints then enforce the first-stage decision variables to be equal to each other across all scenarios to prevent anticipation of the future. Since the appointment time values (a_i) and the precedence relationship variables (b_{ij}) are dependent on each other, there is no need to include non-anticipativity constraints for the latter. In SMIP-R, non-anticipativity constraints for the appointment time values are given by:

$$a_i^\omega = a_i \quad \forall i \in I, \forall \omega \in \Omega, \tag{4.1}$$

where the a_i values represent the consensus variable for appointment time of patient $i \in I$. Prior to the PHA implementation, these non-anticipativity constraints are relaxed, and their violation are penalized with two terms in the objective function. In the first term, the amount of violation for each patient is multiplied by Lagrangian multipliers denoted by μ_i^ω $\forall i \in I, \forall \omega \in \Omega$. The second term is a quadratic component that takes the square of these violations, and multiplies them by $\frac{\rho}{2}$, where ρ denotes the penalty parameter. The revised objective function is given below.

$$\lambda \sum_{i \in I} \sum_{\omega \in \Omega} p^{\omega} w_i^{\omega} + (1-\lambda) \sum_{n \in N} \sum_{\omega \in \Omega} p^{\omega} O_n^{\omega} + \sum_{i \in I} \sum_{\omega \in \Omega} \mu_i^{\omega}(a_i^{\omega} - a_i) +$$
$$\frac{\rho}{2} \sum_{i \in I} \sum_{\omega \in \Omega} ||a_i^{\omega} - a_i||^2 \tag{4.2}$$

To have a separable formulation, a_i in (4.2) is replaced by $\hat{a}_i$, a consensus parameter that is equal to the weighted average of appointment times over all scenarios, where the weights are equal to the probabilities of scenarios.

$$\hat{a}_i = \sum_{\omega \in \Omega} p^{\omega} a_i^{\omega} \quad \forall i \in I \tag{4.3}$$

The resulting structure of the objective function and constraint set allows the SMIP-R to be decomposed into scenario subproblems, which are solved independently at each iteration of the algorithm.

The general steps of the PHA, shown also in Algorithm 1, are as follows: In step 1, algorithm parameters are initialized. Penalty parameter of the quadratic term ρ is set as a positive constant value of ρ^0, and the Lagrangian multipliers $\mu_i^{\omega(v)}$ $\forall i \in I, \forall \omega \in \Omega$ are initialized as 0 at iteration v=1. In step 2, each scenario subproblem is solved. Note that quadratic penalty and Lagrangian terms are ignored in the first iteration. In step 3, the appointment time values obtained for every patient and scenario are used to calculate the consensus parameters $\hat{a}_i$ for every patient $i \in I$. In step 4, the penalty parameter ρ is updated by multiplying it with a pre-determined positive constant α. By using the value of the penalty parameter, Lagrange multipliers are updated considering the distance between scenario solutions and the consensus parameter value. At the end of each iteration, the termination criterion is checked. If all first-stage solutions are identical, the algorithm stops. Otherwise, the algorithm returns to step 2 and solves scenario subproblems with the updated values of the multipliers and consensus parameter.

4.2 Literature Review on the Progressive Hedging Algorithm

The PHA is used in various application areas where the problem involves stochastic components. In this section, a literature review for different applications and modifications of the PHA is provided.

Algorithm 1 Progressive Hedging Algorithm

Step 1: Initialization:
Let v=1
$\rho^{(v)} = \rho^0$
$\mu_i^{\omega(v)} = 0 \quad \forall i \in I, \forall \omega \in \Omega$
Step 2: Solving scenario-subproblems:
if $v = 1$;
For each $\omega \in \Omega$; ignore Lagrangian and quadratic penalty terms and solve
scenario subproblems to obtain:
$a_i^{\omega(v)} \quad \forall i \in I, \forall \omega \in \Omega$
end if
else
For each $\omega \in \Omega$; scenario subproblems to obtain:
$a_i^{\omega(v)} \quad \forall i \in I, \forall \omega \in \Omega$
end else
Step 3: Calculate the consensus parameters:
$\hat{a}_i = \sum_{\omega \in \Omega} p^\omega a_i^{\omega(v)} \quad \forall i \in I$
Step 4: Update penalty parameters:
if $v > 1$;
$\rho^{(v+1)} = \alpha \rho^{(v)}$
end if
Step 5: Update Lagrange Multipliers:
$\mu_i^{\omega(v+1)} = \mu_i^{\omega(v)} + \rho^{(v)}(a_i^{\omega(v)} - \hat{a}_i) \quad \forall i \in I, \forall \omega \in \Omega$
Step 6: Termination:
if all first stage solutions are the same, then stop.
$a_i^{\omega(v)} = \hat{a}_i \quad \forall i \in I, \forall \omega \in \Omega$
end if
else
Set v= v+1;
end else
return to Step 2

An inventory routing problem was studied with uncertain customer demands by Hvattum et al. [38]. Scenario trees were used to represent the stochastic nature, and the PHA is extended with some innovations such as dynamic multiple penalty parameters and heuristic intermediate solutions. The penalty parameter is increased when the distance between the scenario solutions and the consensus parameter increases. When the consensus parameters at consecutive iterations moves farther away from each other, the penalty parameter is reduced. They also fixed the integer variables that are identical for all scenario subproblems at an intermediate iteration.

Gul et al. [37] formulated a multi-stage stochastic mixed integer program for a surgery planning problem. The authors used bundle concept to represent common decisions in different scenarios. The bundle constraints, as non-anticipativity constraints, are dualized and placed in the objective function of the deterministic equivalent model used in PHA. Since the quadratic term in the objective function included only binary decision variables, the subproblems solved in the algorithm became mixed integer programs. The penalty term was updated according to the method suggested by Hvattum et al. [38]. The Lagrangian penalty parameters were also updated by checking the majority of the scenario decisions.

Gade et al. [43] studied the stochastic unit commitment problem and used the PHA to solve their problem. The authors proposed a method to find lower bounds from the PHA in any iteration by taking the expectation of the related scenario solutions. The method can be applicable to both two-stage and multi-stage stochastic mixed-integer programs. By using dual prices, a mathematical model for every scenario was solved by using an objective function that consists of similar terms with the PHA except the quadratic term. The authors concluded that the bound obtained from an intermediate iteration of the PHA can be quite tight. In the same study, they bundled scenarios, and solved the resulting bundle subproblems to increase the speed of convergence. They observed that scenario bundling not only provides smaller number of iterations but also improves lower bound quality. The same authors worked on the stochastic unit commitment problem in [44] and solved scenario subproblems approximately in the early iterations of the PHA, since obtaining exact solutions from the scenario subproblems is not a necessity for the algorithm. After the second iteration, they linearly reduced the optimality gap value that is used to terminate the CPLEX iterations. Since variable fixing provides fast solutions in the later iterations, reducing gap does not increase the solution times significantly.

A number of modifications to improve convergence behavior and computation times of the PHA are suggested by Watson and Woodruff [42], who classified the changes made in the algorithm in four categories: computing effective ρ values, accelerating convergence, termination criteria and detecting cyclic behaviour. In the first category, ρ was updated proportionally to the cost coefficient associated with the relevant decision variable. The aim was to construct a heuristic approach to determine the values of ρ for improving the convergence. For the second category, they considered variable fixing and slamming. The values of the decision variables that did not change within some predefined amount of iterations were fixed to reduce solution times. By giving up slightly on the solution quality, the solution times could be significantly improved. In slamming, decision variables are fixed before the convergence where the scenario solutions are close enough to each other. The decision variable is recommended to be fixed to the maximum value observed in the scenario solutions to preserve feasibility. Termination criterion is determined by examining the total value of deviation from the consensus variable. A limit for total deviation can be set to allow small differences within the scenarios to reduce the solution times. For the fourth category, the behaviour of the Lagrangian penalty term was observed to detect cycles. When a cyclic behaviour was observed for a particular first-stage variable, the values of the variable in all scenario subproblems were fixed to the maximum of them.

The PHA was applied to stochastic networks by Mulvey and Vladimirou [45], where a new technique to update the penalty parameter dynamically is suggested. The technique can prohibit stalling and ill-conditioning solution structures, which occurs when the algorithm gets stuck at suboptimal solutions. The penalty parameter update technique is given by:

$$\rho^{(v+1)} = (\alpha \rho^{(v)})^{\eta} \tag{4.4}$$

where η is a positive parameter used to control the value of the penalty parameter. In their method, the choices of α and η have significant importance for convergence. On the other hand, the authors also suggest that the penalty parameter can be reduced when the scenario solutions are close to convergence to the consensus variable. To decrease the solution times of the scenario subproblems, obtaining inexact solutions may also be beneficial.

Other studies that propose modifications to the PHA include Gonçalves et al. [39], where serial and parallel implemantations of PHA were studied on a operations planning problem of a hydrothermal system, Atakan et al. [46], where the PHA was used to solve the nodal relaxations in the nodes of branch-and-bound method for stochastic mixed integer programs, and Helgason and Wallace [47], who used approximate individual scenario solutions in the PHA in an application in the fishery management.

4.3 Modifications on the Progressive Hedging Algorithm

When the PHA is applied to the Chemotherapy Appointment Scheduling Problem, even single-scenario subproblems require substantial computational times, which deem realistically-sized instances of the problem impossible to solve in reasonable time. A number of issues contribute to this issue. The first one is the quadratic term used in the objective function of the subproblems, whereas the second one is the choice of penalty parameter update method. We investigate other improvement ideas on the PHA by detecting and favoring solutions found in the majority of the scenario subproblems, detecting cycles, and varying the optimality gap parameter of CPLEX to obtain approximate scenario subproblem solutions to decrease solution times. We also propose a lower bounding scheme for nurse overtime by benefiting from the structure of the PHA. In the following sections, these modifications on the PHA are presented.

4.3.1 Handling the Quadratic Term in the Objective Function

The scenario subproblems solved at each iteration of the PHA includes a quadratic term in the objective function, which makes them difficult to solve. This term looks as follows:

$$\frac{\rho}{2}\sum_{i\in I}\sum_{w\in\Omega}||a_i^{\omega(v)}-\hat{a}_i||^2=\frac{\rho}{2}\sum_{i\in I}\sum_{\omega\in\Omega}(a_i^{\omega(v)})^2-\rho\sum_{i\in I}\sum_{\omega\in\Omega}(a_i^{\omega(v)}\hat{a}_i)+\frac{\rho}{2}\sum_{i\in I}\hat{a}_i{}^2 \tag{4.5}$$

Since $\hat{a}_i$ is a parameter, the only quadratic part is the first term on the right-hand side of (4.5). Our aim is to use a method to handle this term in a fast manner without sacrificing significantly from the solution quality. For this end, we attempt three different approaches:

1. Nonlinear optimization using commercial solvers

2. Second-order cone programming (SOCP)

3. Linearization of the quadratic term

For the first approach, Nonlinear Optimization package of CPLEX 12.8 is used. The quadratic term is included in the objective function without making any change in the algorithm. In the second approach, additional variables are defined to convert the quadratic term into a appropriate format for second-order cone programming. For the third approach, two alternatives are available in the PHA literature. The quadratic term is linearized using a piecewise linear function in Watson et al. [48]. Alternatively, Helseth [49] dynamically approximated the quadratic term by benefiting from tangent line approximation. In particular, linear cuts that approximate the quadratic term are added to scenario subproblems to have a linear objective function.

Based on our preliminary experiments, we decided to use the third approach since it outperforms the other two in terms of the computational times significantly. In fact, due to the use of linearized penalty components in the objective function, we call our algorithm as *linearized progressive hedging algorithm* (LPHA). The linearization method that we use, which is based on the approach in [49], is explained next.

Taylor's approximation can be useful to estimate $(a_i^{\omega(v)})^2$. A general form of Taylor's approximation is given by:

$$f(m) \approx f(n) + f'(n)(m-n) \tag{4.6}$$

where *n* is an operating point that helps to evaluate $f(m)$. In our problem, two different values can be used as the operating point to estimate the value of $a_i^{\omega(v)}$. These include the consensus parameter value and the appointment time obtained for the same scenario in iteration $(v-1)$, respectively. Based on our preliminary experiment results, we chose the second option while formulating the linear cuts. A new variable $g_i^{\omega(v)}$ is defined to represent $(a_i^{\omega(v)})^2$. The definition of parameters and decision variables for the proposed linear cuts are given as follows:

$g_i^{\omega(v)}$: variable used to approximate the value of $(a_i^{\omega})^2$ for patient i in scenario ω

$$(a_i^{\omega(v)})^2 \geq (a_i^{\omega(v-1)})^2 + 2(a_i^{\omega(v-1)})(a_i^{\omega(v)} - a_i^{\omega(v-1)}) \tag{4.7}$$

$$(g_i^{\omega(v)}) \geq (a_i^{\omega(v-1)})^2 + 2(a_i^{\omega(v-1)})(a_i^{\omega(v)} - a_i^{\omega(v-1)}) \tag{4.8}$$

The cuts in (4.8) are added to scenario subproblems at each iteration until it is noticed that the value of g_i^{ω} does not change from one iteration to another.

4.3.2 Choice of Penalty Parameter Update Method

In the PHA, the penalty parameter ρ is updated by multiplying it with constant α in every consecutive iteration. When the multiplier α is set to 1, the penalty parameter becomes stable in the algorithm. Previous studies have shown that low values of ρ tend to improve the quality of the solutions at the expense of longer computational times. On the other hand, high values of the penalty parameter can lead to oscillations in the solutions and the algorithm generally converges faster to a low-quality solution. Therefore, it is crucial to find a balance between solution time and quality using an appropriate level of the penalty parameter over iterations. Designing a well-performing penalty update method is challenging due to this trade-off. Besides there is no straightforward way to select the initial value of the penalty parameter; it is mostly dependent on the objective function coefficients and scales of the decision variable values in the problem.

One way to resolve the trade-off between solution quality and computational time is to apply a dynamic penalty parameter scheme by updating the value of the parameter in subsequent iterations. In this study, we use a method inspired by the approach discussed in Hvattum and Lokketangen [38]. In this method, two parameters are made use of by exploiting the relationship between the primal and dual. The first parameter, denoted by Δ_p measures the square of the difference between consensus parameters for the appointment time of each patient at consecutive iterations. If Δ_p is increasing, this means that consensus parameter is changing at a greater rate. This also implies that the current value of consensus parameter is not promising, and therefore it is risky to enforce convergence immediately. The second parameter denoted by Δ_d, compares the convergence behaviour of the first-stage variable to the consensus parameter at consecutive iterations. If Δ_d is increasing, this indicates that the first-stage variables in different scenario subproblems are farther away from the consensus parameter.

In our implementation of the penalty parameter update method, when Δ_d increases in consecutive iterations, penalty parameter ρ is multiplied by a constant α that is greater than 1. By this way, the algorithm becomes inclined to converge faster. If the difference between Δ_d in consecutive iterations is not positive, the behaviour of the Δ_p is observed. If Δ_p is rising, the penalty parameter is reduced by a factor of

$\frac{1}{\alpha}$. If both consecutive Δ_p and Δ_d differences are negative, the penalty parameter is preserved constant.

Based on preliminary results, we observe that penalty parameter can quickly climb to very large values as a result of multiplying with α. In terms of convergence, higher penalty parameter values lead the algorithm to prematurely converge to suboptimal solutions. To avoid this issue, the value of penalty parameter is bounded above by a pre-determined parameter ρ^u, so that the algorithm can converge to a higher quality solution.

It is observed in our preliminary experiments that keeping the penalty parameter small is beneficial in terms of solution quality. However, if the predefined bound for the penalty parameter is kept smaller than needed during the iterations, this may result in excessive computational times. To avoid this, we preserve a small bound at the earlier iterations to obtain high quality solutions, and then we increase the limit in the later iterations to achieve fast convergence. The details of the penalty update method is provided below in Algorithm 2. In the algorithm, ρ_1^u and ρ_2^u refer to two different penalty parameter limits, where $\rho_1^u < \rho_2^u$. Furthermore, $iterlimit$ corresponds to the iteration value at which the upper bound is changed from ρ_1^u to ρ_2^u.

4.3.3 Cycling and Majority Value Detection

Cycles are frequently observed within a typical implementation of the PHA. This behaviour may prevent convergence. We benefit from the idea of cycling prevention suggested by Watson and Woodruff [42]. The appointment time values may appear identical over several consecutive PHA iterations. The direct consequence of this is that the consensus parameter values remain constant over the iterations. On the other hand, Lagrangian multipliers change in every iteration since we update their values using Step 5 of Algorithm 1. While updating the Lagrangian multipliers, the differences between appointment time values and consensus parameter are penalized by multiplying it with a penalty parameter. Due to this reason, Lagrangian multipliers vary over iterations even if the appointment time and consensus parameter values stay the same. Hence, any cycling behaviour of the algorithm can be detected by examining the behaviour of the Lagrangian multipliers.

Algorithm 2 Penalty Update Method with Limit

$\Delta_p{}^{(v)} = \sum_{i \in P} (\hat{a}_i{}^{(v)} - \hat{a}_i{}^{(v-1)})^2$

$\Delta_d{}^{(v)} = \sum_{i \in P} \sum_{\omega \in \Omega} (a_i^{\omega(v)} - \hat{a}_i{}^{(v)})^2$

if $v \leq iterlimit$

  $\rho^u = \rho_1^u$

else

  $\rho^u = \rho_2^u$

if $\Delta_d{}^{(v)} - \Delta_d{}^{(v-1)} > 0 \quad \& \quad \rho^{(v)} < \rho^u;$

  $\rho^{(v+1)} = \alpha \rho^{(v)}$

elseif $\Delta_d{}^{(v)} - \Delta_d{}^{(v-1)} > 0$

  $\rho^{(v+1)} = \rho^u$

elseif $\Delta_p{}^{(v)} - \Delta_p{}^{(v-1)} > 0$

  $\rho^{(v+1)} = \frac{1}{\alpha} \rho^{(v)}$

elseif $\rho^{(v)} \leq \rho^u;$

  $\rho^{(v+1)} = \rho^{(v)}$

else $\rho^{(v+1)} = \rho^u$

Watson and Woodruff [42] used a hashing scheme to detect cycles. In our case, we would like to detect cycles in integer appointment times for each patient. Therefore, we have to check the behavior of the Lagrangian multipliers associated with each patient. In their approach, Watson and Woodruff [42] first generated hash weights for each scenario (z_ω). In our algorithm, hash weights are the random numbers between 0 and 1. These hash weights are then multiplied with the Lagrangian multipliers (μ_i^ω). Hash value for each patient is calculated in every iteration as:

$$h_i = \sum_{\omega \in \Omega} z_\omega \mu_i^{\omega(v)} \quad \forall i \in I \tag{4.9}$$

Note that z_ω values are selected different from probability of occurence of each scenario (p^ω). Otherwise, h_i value would be zero. To detect cycling, we compare values of h_i in consecutive iterations. Since the continuous values of hash weights and Lagrangian multipliers may result in non-integer hash values, we compare the hash values of patients in consecutive iterations by using a small threshold value. If the difference between the hash values are smaller than the threshold value over three consecutive iterations, it is considered as the indicator of a cycle. The cycle then needs to be broken to maintain algorithm convergence.

After a cycle is detected, the appointment time for patient i can be fixed to a certain value. There might be different approaches for fixing the appointment time, which include the minimum and maximum appointment times observed among all scenario subproblem solutions for that patient in the previous iteration. However, based on our preliminary experiments, we fix the appointment time to the one that is most repeatedly observed among subproblem solutions, which we call the majority value. We use this approach after the first 50 iterations of the PHA, since there is high variation in the appointment times of a patient across different scenarios in the beginning iterations.

A common observation in our preliminary experiments was that for many patients, the appointment times for a majority of the scenarios converged quickly, whereas those of the remaining scenarios failed to converge for many iterations. To overcome this, we fix the appointment time of a patient if there is convergence in 80% of the scenarios. Although this may reduce the solution quality, it significantly reduces computational times.

4.3.4 Varying CPLEX optimality gap value in the subproblems

In the earlier iterations of the PHA, computation times to solve the subproblems are longer since the algorithm uses lower values of penalty parameter ρ. However, the PHA does not need optimal solutions of the subproblems for the convergence. We may heuristically solve the subproblems to decrease the solution times. For this purpose, we use the idea proposed by Watson and Woodruff [42], where the CPLEX optimality gap value for the subproblems is varied according to the convergence behavior. In particular, it is monotonically decreased by checking the following condition:

$$gap_{v+1} = \min\left\{ \sum_{i\in I, \hat{a_i}^{(v)}\neq 0} \sum_{\omega\in\Omega} \frac{|a_i^{\omega(v)} - \hat{a_i}^{(v)}|}{\hat{a_i}^{(v)}},\ gap_v \right\} \tag{4.10}$$

The gap value is decreased if the appointment times of the patients get closer to the consensus parameter. Otherwise, we keep the gap value equal to that of previous iteration. The initial gap value is determined according to the results of the preliminary experiments.

Using approximate solutions within the PHA may either improve solution quality by making greater amount of iterations or reduce it because of the low-quality solutions particularly at the earlier iterations. The details of this procedure and results are discussed at Chapter 5.

4.3.5 Lower Bound Structure for Nurse Overtime

In the first iteration of the PHA, the scenario subproblems are solved without adding any Lagrangian or quadratic term. Since the durations are known in each subproblem, it is obvious that the waiting times for all patients will be zero in the optimal solution. The only term that contributes to the subproblem objective function in this case is the summation of nurse overtime values. Since the scenario solution obtained in the first iteration is decided in a deterministic environment, the resulting total nurse overtime value constitute a lower bound for the associated scenario subproblem in the subsequent iterations. The summary of this procedure is shown in Figure 4.1.

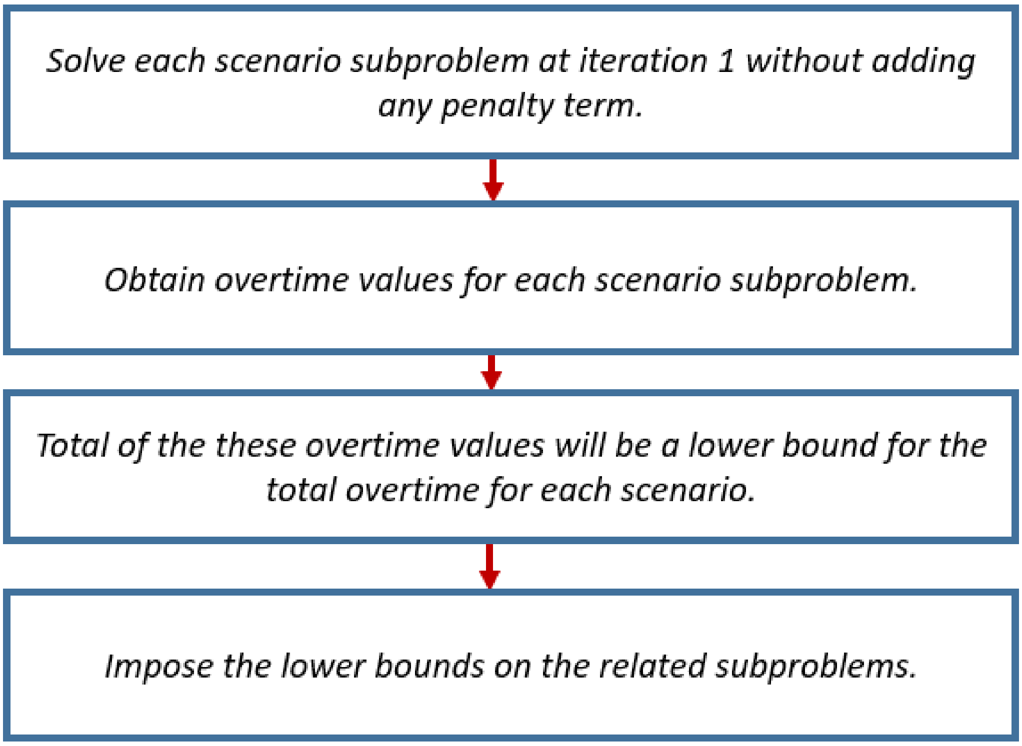

Figure 4.1: Lower Bound for Nurse Overtime within Progressive Hedging Algorithm

CHAPTER 5

COMPUTATIONAL EXPERIMENTS

In this chapter, we discuss the computational experiments for our chemotherapy appointment scheduling problem. We test the methods using data gathered from the Outpatient Chemotherapy Unit at Hacettepe Oncology Hospital from November 2017 to March 2018. The data set includes realized pre-medication and infusion times as well as planned treatment durations for all patients.

In our experiments, Microsoft Visual C++ 2017 is used for the implementation of the algorithm using CPLEX 12.8 Concert Technology. The computations were performed with Intel (R) Core (TM) i7-4790 CPU @3.10 GHz and 16GB RAM.

In Section 5.1, we summarize the generation of instances based on data from Hacettepe Oncology Hospital. The methods proposed to improve scenario subproblem solution times are tested in Section 5.2. The sensitivity of the linearized progressive hedging algorithm (LPHA) to the algorithm parameters is investigated in Section 5.3. The performance of the LPHA is assessed in Section 5.4. Sensitivity analysis on model parameters is conducted to generate managerial insights in Section 5.5. Finally, the value of stochastic solution (VSS) is estimated in Section 5.6.

5.1 Generating Problem Instances

We conduct each experiment on a problem instance set that consists of 10 instances. We generate those problem instances by sampling pre-medication and infusion durations in our data set. The treatment duration of a cancer patient ranges in between 16 and 240 minutes. The duration for a pre-medication phase varies between 0 and

36 minutes, whereas infusion durations change between 16 and 217 minutes. In Hacettepe Outpatient Chemotherapy Unit, the schedules are designed for half-day periods. Therefore, if the estimated duration of a treatment exceeds four hours, the treatment is split into two sessions. To mimic the current practice followed in the unit, we also design half-day schedules.

First of all, data set at hand is clustered according to planned durations for treatments. Four different planned duration classes are formed by specifying lower and upper limits of class intervals as: 20-45, 45-100, 100-150, 150-240 minutes. For each class, the actual pre-medication and infusion durations are analyzed. The intervals for the realized pre-medication and infusion durations for the defined classes as well as the probability of observing the defined classes in the data set are provided in Table 5.1.

Table 5.1: Pre-medication and infusion intervals and the probabilites of observing the defined classes

Duration Interval (min.)	**Probability of Observing the Class**	**Pre-medication Interval**	**Infusion Interval**
[20, 45]	0.10680	[0, 14]	[16, 44]
[45, 100]	0.23301	[6, 35]	[29, 80]
[100, 150]	0.23786	[8, 26]	[74, 132]
[150, 240]	0.42233	[6, 27]	[125, 217]

To generate treatment durations for a patient, we first draw a random number between zero and one in order to determine her class. Next, pre-medication and infusion durations of the patient are drawn from the associated intervals based on uniform distribution.

We consider a set of 10 problem instances in our experimental runs. The instances are labeled as i_j_k where i, j, and k refer to the instance number, number of patients and scenarios, respectively.

5.2 Solving Scenario Subproblems

Several scenario subproblems are solved at each iteration of our algorithm. Therefore, reducing the computational effort spent on the solution of a subproblem would provide significant benefits. In this section, we evaluate different approaches to improve solution times of scenario subproblems. We perform analyses for both strenghtening the model formulation and enhancing the structure of the PHA. For this purpose, we firstly investigate the effects of using different approaches while handling the nonlinear term in the algorithm.

5.2.1 Selection of a technique to solve nonlinear scenario-subproblems

Each scenario subproblem has a nonlinear objective function due to the quadratic term necessary for smooth convergence of the PHA. We test three different approaches to handle the quadratic programming formulation in the subproblem: (1) using CPLEX as nonlinear optimization solver, (2) treating the model as second-order cone program (SOCP), and (3) replacing the quadratic term by its linear approximation. We perform a test to compare the performances of the approaches in small-size problems. The first two approaches are comparable with each other in terms of computation times. On the other hand, the third one outperforms the others in terms of computation times without sacrificing much from the quality of solutions. Therefore, we use the third approach for solving the scenario-subproblems and call this approach as Linearized Progressive Hedging Algorithm (LPHA).

5.2.2 Effect of symmetry-breaking constraints and bound on nurse overtime

The mathematical programming model for chemotherapy appointment scheduling problem is strenghtened with equations (3.27), (3.28), and (3.29) as given in Chapter 3. These equations make scenario subproblems easier to solve by breaking symmetry within the nurses and by determining a lower bound on overtime level of Nurse 1. Since the feasible region of the mathematical programming model becomes narrower after adding these constraints to the scenario subproblems, we expect lower compu-

Table 5.2: Effect of varying CPLEX optimality gap value on the performance of LPHA

	Initial optimality gap = default			**Initial Optimality Gap = 0.1**			**Initial Optimality Gap = 0.15**		
Instance name	**Objective Value**	**Run Time (s)**	**Number of Iterations**	**Objective Value**	**Run Time (s)**	**Number of Iterations**	**Objective Value**	**Run Time (s)**	**Number of Iterations**
1_8_50	48.53	2866.40	107	49.63	2669.47	128	47.78	2123.33	128
2_8_50	47.78	6651.35	124	48.35	9434.17	128	47.44	7124.02	121
3_8_50	59.92	2103.39	109	60.09	2196.83	118	59.76	3117.67	113
4_8_50	35.35	4187.46	120	34.95	2994.86	109	38.31	2964.76	172
5_8_50	53.72	3555.59	103	54.03	3306.86	127	54.05	3623.33	117
6_8_50	51.50	3521.03	106	49.05	2771.02	121	49.45	2511.32	118
7_8_50	41.96	5682.71	112	45.65	5464.81	123	47.76	6351.17	141
8_8_50	53.82	3084.82	118	53.07	2652.60	126	57.76	3664.83	112
9_8_50	57.72	2225.59	110	57.64	2662.63	123	58.42	3366.93	111
10_8_50	48.25	3614.74	111	48.46	3873.18	123	48.97	3561.55	116
Average Values	***49.85***	***3749.31***	***112***	***50.09***	***3802.64***	***123***	***50.97***	***3840.89***	***125***

tational times in solving them. We apply the LPHA on the formulation including these constraints, and the resulting computation times are compared with those found based on the formulation without these constraints. The computational times are improved after adding the symmetry breaking constraints and bound on overtimes for nurses. The results show that the effect of these constraints becomes more explicit when patient, nurse, and chair numbers are increased.

5.2.3 Tests for varying CPLEX optimality gap value in the subproblems

A technique for solving scenario subproblems approximately based on optimality gap values that vary over iterations was introduced in Chapter 4, subsection 4.3.4. The optimality gap values exhibit monotonically nonincreasing behavior over iterations due to the given rule. In this part, we perform experiments to see the effects of this technique in the computational times and quality of solutions by starting with two different initial optimality gap values of 10% and 15%. The results of the experiments are presented in Table 5.2. During these experiments, we fixed some parameters of the problem. The value of λ, number of patients, nurses and chairs are fixed to 0.3, 8, 2, and 4, respectively.

The technique is compared with the LPHA that uses default CPLEX optimality gap value while solving subproblems (Default CPLEX optimality gap is 0.0001%.). There is no guarantee that the overall computation times decrease when the optimality gap is increased, because the algorithm may converge after larger amount of iterations. Indeed, the one that uses default CPLEX optimality gap value performs the best in terms of solution quality and run time on average. Due to insufficient basis for the use of rule, we do not vary the optimality gap in our further experiments.

5.3 Sensitivity analysis on LPHA parameters

In this part, we conduct experiments to evaluate the sensitivity of the LPHA to the algorithm parameters. As there are many parameters associated with the LPHA, we perform preliminary experiments to fix some of them before elaborating on others. As a result, the initial values of Lagrangian multipliers ($\mu_i^{\omega(v)}$) are fixed to zero since there is no significant effect of changing this parameter. We also fix number of patients, nurses, and chairs in the experiments presented at this section to see the effect of the LPHA parameters clearly. In the experiments presented at this section, we assumed two nurses, four chairs, and eight patients are present in the unit. The value of λ is also fixed to 0.3.

The effects of different values set for α, ρ, ρ_1^u, and $iterlimit$ are tested based on a one-way sensitivity analysis approach. First, we investigate the effect of α on the performance of the algorithm.

5.3.1 Effect of α in the performance of LPHA

The parameter α is the step size that determines the rate of change in the penalty parameter ρ. The parameter ρ may be updated by a factor of α, 1, or $\frac{1}{\alpha}$ according to the changes in the Δ_p and Δ_d, defined in Chapter 4. When α takes higher values, it means the penalty parameter varies significantly during the consecutive iterations. On the other hand, small values of α smoothly change the value of the penalty parameter. When the deviation in the penalty parameter is smaller, we expect convergence in longer computational times to a higher quality solution. We set three different values of α (1.5, 2, 4) to observe the effect of it in both computational times and solution quality. In Table 5.3, the results of the experiments on 10 different instances are presented.

Table 5.3: Effect of α change in the algorithm performance

	α=1.5			**α=2**			**α=4**		
instance name	**Objective Value**	**Run Time (s)**	**Number of Iterations**	**Objective Value**	**Run Time (s)**	**Number of Iterations**	**Objective Value**	**Run Time (s)**	**Number of Iterations**
1_8_50	47.47	3687.45	91	48.52	2866.40	107	49.83	2431.25	112
2_8_50	50.07	9377.71	130	47.78	6651.35	124	47.38	4997.25	107
3_8_50	58.91	3321.66	109	59.92	2103.39	109	59.32	1750.90	104
4_8_50	35.92	6958.45	113	35.35	4187.46	120	35.57	3045.20	104
5_8_50	53.68	6223.86	120	53.72	3555.59	103	56.90	4297.17	124
6_8_50	55.44	5717.47	115	51.45	3521.03	106	55.36	2579.76	75
7_8_50	43.24	12811.90	129	41.96	5682.71	112	43.49	4428.96	105
8_8_50	57.45	7213.58	111	53.82	3084.82	118	51.77	3319.27	107
9_8_50	56.99	5357.53	109	57.72	2225.59	110	61.29	2042.10	112
10_8_50	48.60	5081.09	119	48.25	3614.74	111	49.00	3089.77	110
Average Values	***50.78***	***6575.07***	***115***	***49.85***	***3749.31***	***112***	***50.99***	***3198.16***	***106***

The table also indicates the average values of the objective values of the instances and average computation times in these three different settings. We observe that the case of $\alpha = 2$ performed better than other cases in terms of average objective value. The computation time decreases with increasing values of α as expected. The number of iterations needed until convergence also decreases as α increases. When the value of α is larger, the penalty parameter ρ can reach larger values in earlier iterations. This makes solving scenario subproblems easier, and thus reduces the subproblem solution time. Due to this reason, it is not surprising to have smaller computational times with larger α values.

The value of α is aimed to be fixed to a resonable value where we have a balanced circumstance between solution time and quality. When $\alpha = 4$, the computation times are more promising. On the other hand, the solutions are not consistently strong through instances in terms of quality. We have much worse solutions compared to other α settings for some instances. Therefore, we fix $\alpha = 2$ which gives better and consistently good enough solutions in reasonable amount of time.

5.3.2 Effect of initial value of ρ in the performance of LPHA

There are various factors that affect the behaviour of the penalty parameter value ρ. The first one is the initial value of it in the first iteration. According to the previous literature, the lower values of ρ yields better quality solutions in longer computational times. To test the effect of the initial value of ρ, three different values are used (0.0001, 0.01, and 0.1). The experimental results are provided in Table 5.4.

Table 5.4: Effect of change in initial value of penalty parameter (ρ_1^u) in the performance of LPHA

	ρ=0.0001			**ρ=0.01**			**ρ=0.1**		
instance name	**Objective Value**	**Run Time (s)**	**Number of Iterations**	**Objective Value**	**Run Time (s)**	**Number of Iterations**	**Objective Value**	**Run Time (s)**	**Number of Iterations**
1_8_50	48.526	2866.4	107	63.43	2814.73	124	56.87	2330.27	90
2_8_50	47.78	6651.35	124	47.682	3161.57	107	48.90	3273.86	112
3_8_50	59.916	2103.39	109	58.778	1531.13	79	60.11	2012.09	135
4_8_50	35.346	4187.46	120	34.292	2167.90	109	34.78	1956.05	110
5_8_50	53.722	3555.59	103	60.372	3115.91	108	53.04	3271.82	103
6_8_50	51.496	3521.03	106	50.376	2397.60	170	55.00	2213.27	110
7_8_50	41.96	5682.71	112	53.858	3640.15	123	47.52	3364.41	98
8_8_50	53.822	3084.82	118	56.998	1912.18	120	52.97	1609.74	82
9_8_50	57.722	2225.59	110	63.962	2144.96	139	57.56	1573.98	76
10_8_50	48.25	3614.74	111	54.97	3129.26	93	48.32	3347.27	111
Average Values	***49.85***	***3749.31***	***112***	***54.47***	***2601.54***	***117***	***51.51***	***2495.28***	***103***

The computation times decrease significantly with the increase in the initial value of ρ. Since the increase in the initial value of ρ may create an undesirable increase in the objective values for some instances, we avoid using initial ρ value as 0.01 and 0.1 in the additional experiments. The lower values of the penalty parameters provide consistently good objective function values.

5.3.3 Effect of upper limit on penalty parameter ρ_1^u on the performance of LPHA

As stated in Chapter 4, the value of the penalty parameter ρ is limited using an upper bound ρ_1^u on its value. This prevents faster convergence of the algorithm to premature solutions due to high penalty parameter values. There is a trade-off between solution quality and computational times to be considered while selecting the value of ρ_1^u. For this purpose, we used three different values as 0.1, 0.5, and 0.7 to test the performance of the algorithm. The experimental results are provided in Table 5.5.

Table 5.5: Effect of change in first upper limit of penalty parameter (ρ_1^u) on the performance of the LPHA

	ρ_1^u=0.1			ρ_1^u=0.5			ρ_1^u=0.7		
instance name	**Objective Value**	**Run Time (s)**	**Number of Iterations**	**Objective Value**	**Run Time (s)**	**Number of Iterations**	**Objective Value**	**Run Time (s)**	**Number of Iterations**
1_8_50	48.53	2866.40	107	55.58	2803.66	124	60.45	2407.81	131
2_8_50	47.78	6651.35	124	48.92	7049.39	145	48.75	6157.34	147
3_8_50	59.92	2103.39	109	78.58	1923.07	135	81.89	1786.41	110
4_8_50	35.35	4187.46	120	47.70	4295.61	155	53.20	3951.52	177
5_8_50	53.72	3555.59	103	77.79	3594.69	133	85.35	3600.41	131
6_8_50	51.50	3521.03	106	60.24	3342.44	110	66.24	3603.25	203
7_8_50	41.96	5682.71	112	41.83	5552.75	103	41.80	5217.23	129
8_8_50	53.82	3084.82	118	55.79	3088.51	121	55.47	2885.80	118
9_8_50	57.72	2225.59	110	66.17	2246.98	157	74.21	2133.24	152
10_8_50	48.25	3614.74	111	58.06	3175.75	126	57.95	3117.85	123
Average Values	***49.85***	***3749.31***	***112***	***59.07***	***3707.29***	***131***	***62.53***	***3486.09***	***142***

According to the results provided in Table 5.5, the best solutions with respect to the solution quality are obtained with a smaller value of ρ_1^u. When the value of ρ_1^u is increased, the algorithm needs more iterations for convergence. Since the larger penalty parameter values make scenario subproblems easier to solve, the time spent for completing an iteration decreases. The table also reveals that the average computational times for the cases where $\rho_1^u = 0.1$ and $\rho_1^u = 0.5$ are close to each other. However, the second one needs 19 additional iterations for convergence on average. When ρ_1^u is raised to 0.7, the computational times are improved, but the objective value deteriorated by 25% with respect to the case where $\rho_1^u = 0.1$. Since there is no significant difference in the computation times, we use 0.1 for ρ_1^u for further experiments.

5.3.4 Effect of iteration limit for changing ρ_1^u to ρ_2^u on the performance of the LPHA

In the LPHA, we use ρ_1^u as an upper bound for the value of the penalty parameter up to a predetermined iteration whose value is represented by $iterlimit$ in Algorithm 2 presented at Chapter 4. After $iterlimit$ many iterations, the upper bound on penalty limit is increased to ρ_2^u. We set reasonable values for $iterlimit$, because very large values of $iterlimit$ may lead to limited values of ρ even when the iteration number reaches a considerable amount. This in turn may result in excessive computation times until convergence. On the other hand, having a limited value of ρ over a large amount of iterations would improve solution quality. We try three different alternatives for determining the value of $iterlimit$ as 50, 70 and 100, because the computation times increase significantly, once the $iterlimit$ exceeds 100 according to our preliminary experiments. The values of the objective function and computation times are given in Table 5.6.

Table 5.6: Effect of change in iteration limit ($iterlimit$) in the algorithm performance

	iteration limit:50			*iteration limit:70*			*iteration limit:100*		
Instance Name	**Objective Value**	**Run Time (s)**	**Last İter.**	**Objective Value**	**Run Time (s)**	**Last İter.**	**Objective Value**	**Run Time (s)**	**Last İter.**
1_8_50	49.062	2606.36	162	50.68	2977.85	150	48.53	2866.40	107
2_8_50	50.844	6102.86	152	47.07	6468.69	101	47.78	6651.35	124
3_8_50	65.392	1767.93	78	60.78	2024.65	108	59.92	2103.39	109
4_8_50	63.502	3929.76	153	43.95	4337.01	147	35.35	4187.46	120
5_8_50	63.376	3342.84	141	60.20	3513.85	172	53.72	3555.59	103
6_8_50	51.446	3601.86	75	51.50	3585.05	83	51.50	3521.03	106
7_8_50	65.104	5820.1	129	41.95	5729.8	132	41.96	5682.71	112
8_8_50	56.122	3395.29	128	84.89	2764.47	98	53.82	3084.82	118
9_8_50	62.976	2801.51	209	59.88	2151.71	89	57.72	2225.59	110
10_8_50	48.778	4061.35	174	48.62	3538.37	97	48.25	3614.74	111
Average values	***57.66***	***3742.99***	***140***	***54.95***	***3709.15***	***118***	***49.85***	***3749.31***	***112***

The average computation times are close to each other in the three settings of $iterlimit$.

When the $iterlimit$ is set to lower values, the algorithm starts to use higher values of ρ earlier iterations. Both the quality of the solutions and number of iterations spent to achieve convergence worsen. Since the computational times are similar, we use $iterlimit$ as 100 in our experiments.

5.4 Assessment of the LPHA Performance

In this section, we evaluate the performance of the LPHA by conducting comparisons with CPLEX and commonly used scheduling heuristics from the literature, respectively. In the experiments of this section, the value of λ is 0.3.

5.4.1 Comparison with CPLEX

To validate the performance of the LPHA, we first compared our solutions with CPLEX solutions. Since CPLEX cannot find the optimal solutions in three hours of time limit even for problem instances having 8 patients, 5 scenarios, 2 nurses and 4 chairs. We decrease the number of patients to 7 for this experiment. We used five different λ values on 10 different instances. The objective values, run times, and optimality gap of the algorithm for all instances are provided at Table A.1 in Appendix 2.

On the average, CPLEX spends eight times as much time as the LPHA to find the optimal solution. The average optimality gap for the LPHA solutions is 7.28%, while it is less than 0.35% in 25% of the instances.

5.4.2 Comparison with Scheduling Heuristics

We compare the performance of the LPHA with various combinations of sequencing and appointment time setting heuristics from the relevant literature. We sequence patients using four different sequencing heuristics: increasing mean of treatment time (SPT), decreasing mean of treatment time (LPT), increasing variance of treatment time (VAR), and increasing coefficient of variation of treatment time (CoV). We consider *job hedging* heuristic for determining planned lengths of patient appointments.

Before discussing the implementaion details of the heuristics, the structure of the model in terms of first and second stage decisions is investigated. When values for patient precedence b_{ij} and appointment time a_i variables become known in the first stage, it is then easy to settle on the second stage decisions. The optimal nurse and chair assignment decisions at this stage can be made even using a simple rule that checks b_{ij} values. The rule would favor the patient preceding others in the list while assigning the first available nurse and chair. Next, actual treatment start time for each patient can be determined accordingly, and their waiting times are revealed according to the discharge time of the previously scheduled patients. By checking discharge times of the patients treated by a nurse, overtime value for each nurse can be easily calculated. Since solving the second-stage problem is easy, finding a heuristic approach to solve the first-stage problem is a more critical issue. After setting the first-stage variable values, we call CPLEX rather then using the simple rule to solve the second-stage problem, as CPLEX finds the optimal solution within a very short amount of time.

To create schedules, we first choose one of the the four sequencing heuristics discussed above. There are two different durations that may affect the sequencing rules, average pre-medication and infusion durations. We use average treatment time, which is equal to the summation of those two values.

After the sequence is fixed, we utilize the job hedging heuristic to set patient appointment times [49]. We apply the heuristic in accordance with the approach also used in Gul [50], Castaing et al. [6], Gul et al. [51]. We sort durations for the relevant patient class in the data set in non-increasing order, and calculate the k^{th} percentile of the set to use it for determining the planned treatment duration of a patient. Since there are two different chemotherapy durations in the problem, the percentile values associated with them are separately determined, and the summation of those two values is considered while setting patient appointment times. The steps of the scheduling heuristic that combines job hedging with one of the sequencing heuristics, LPT, is demonstrated in Algorithm 3.

We vary the percentile values between 40% and 75% for all sequencing rules. Table

Algorithm 3 Job Hedging Heuristics for LPT Rule

Step 1:

Calculate the average treatment duration represented by $average_i$ for every patient.

$$average_i = \frac{\sum_{\omega \in \Omega} s_i^\omega + t_i^\omega}{|\Omega|} \quad \forall i$$

Step 2:

Sort the values of the $average_i$ according to LPT rule, assign index numbers to patients and sequence the patients according to their index numbers.

Step 3:

Assign b_{ij} values according to index numbers found at Step 2.

Step 4:

Set appointment times according to given percentile level of job hedging heuristic. If the index of the patient is less than or equal to the both of the numbers of chair and nurse, the appointment time of the associated patient is assigned as zero. The appointment times of the other patients are set sequentially according to their index numbers by checking the first estimated available time of nurse and chair simultaneously.

Step 5:

Call CPLEX to obtain second stage decisions and objective function value.

5.7 compares the performances of the heuristics with the LPHA. The table illustrates the average gap between the solutions of LPHA and each combination of sequencing and job hedging heuristic for a specific percentile level.

The results show that the LPHA outperforms all combinations of heuristics significantly. On the average, LPT rule performs better than the other sequencing rules. This result makes sense since the patients with longer treatment durations are assigned to the earlier hours of the day also at the outpatient chemotherapy unit in Hacettepe Oncology Hospital. Note that the LPHA improves the solutions of even the best-performing and commonly used rule, LPT, by 27%. Furthermore, the percentiles 60%, 55%, 60%, and 65% used for assigning appointment times is the best in terms of quality of the solutions on the average for sequencing rules SPT, LPT, VAR, and CoV, respectively. The detailed computational results for different sequencing rules are provided in Tables A.13-A.16 in Appendix.

Table 5.7: Average gap values between the solutions of the LPHA and heuristics using different sequencing and job hedging levels

Percentile used in job hedging	**Average gap for SPT**	**Average gap for LPT**	**Average gap for VAR**	**Average gap for CoV**
45%	37.8%	27.1%	35.9%	31.4%
50%	36.4%	25.4%	34.4%	29.5%
55%	34.8%	25.0%	32.5%	28.2%
60%	34.5%	25.9%	32.2%	28.4%
65%	34.8%	27.3%	32.6%	29.1%
70%	35.6%	29.1%	33.8%	27.2%
75%	37.7%	31.7%	35.2%	29.0%
Average Values	35.9%	27.4%	33.8%	29.3%

5.5 Sensitivity Analysis on Model Parameters

In this section, we present a sensitivity analysis on the model parameters to generate managerial insights. The value of λ and number of nurses and chairs are varied for this purpose.

5.5.1 Varying value of λ

The chemotherapy unit manager would typically consider the trade-off between patient waiting time and nurse overtime while designing schedules. To illustrate the scope of this trade-off, we vary the value of λ, which is the objective function coefficient associated with patient waiting time, between 0 and 1. $\lambda = 0$ refers to the extreme case where nurse overtime is the only performance measure for the model. On contrary, $\lambda = 1$ means that patient waiting time is important while nurse overtime is taken into account only through the upper limit imposed in the constraint set. We test different values of λ on an instance set by incrementing it by 0.1 at each time. We fixed the values of the number of patients, nurses, and chairs to 8, 2, and 4, respectively in these experiments. The results of all experiments are provided at Appendix 2 in Tables A.2-A.12. In Table 5.8, the summary of the experimental results is provided. Figure 5.1 clearly illustrates the trade-off between patient waiting time and nurse overtime with reference to λ values.

Table 5.8: The values of the average patient waiting time and nurse overtime in changing λ

λ	**Average Waiting time per scen.**	**Average Overtime per scen.**
0	414.72	41.98
0.1	88.91	46.37
0.2	41.83	55.49
0.3	22.32	61.66
0.4	14.15	64.86
0.5	9.23	68.21
0.6	6.16	75.48
0.7	3.52	84.78
0.8	1.38	96.89
0.9	0.55	110.76
1	0.04	300.00

As can be observed from both Table 5.8 and Figure 5.1, the average patient waiting time decreases with increasing λ value since its importance gets higher. When $\lambda = 1$, most instances find zero waiting times for the patients by allocating the largest possible treatment durations to patients which increases idle time of resources and hence overtime. In some instances, the resulting total patient waiting time is not zero, since feasibility can not be preserved in some cases due to the limit on nurse overtime. Since there is no term to force to minimize nurse overtime in the objective function in this case, overtime of the nurses is equal to maximum allowable overtime amount, which is denoted as L in our mathematical programming formulation presented in Chapter 3. As we set L to 150 minutes and we have two nurses in these experiments, the average total nurse overtime equals 300 minutes.

Another interesting observation is the change in computational times while varying λ. When $\lambda = 1$, the scenario subproblem models directly assign overtime values to their upper limits and focuses on minimizing waiting times. The computation times generally appears to be lower in this case. On the other hand, solving scenario subproblems takes longer in $\lambda = 0$ case. In some instances, the algorithm converges to a solution in one iteration by assigning 0 to appointment times of the all patients.

Note that the cases where λ is 0 and 1 are not realistic for the unit manager. These values are only useful for obtaining information about the structure of the problem.

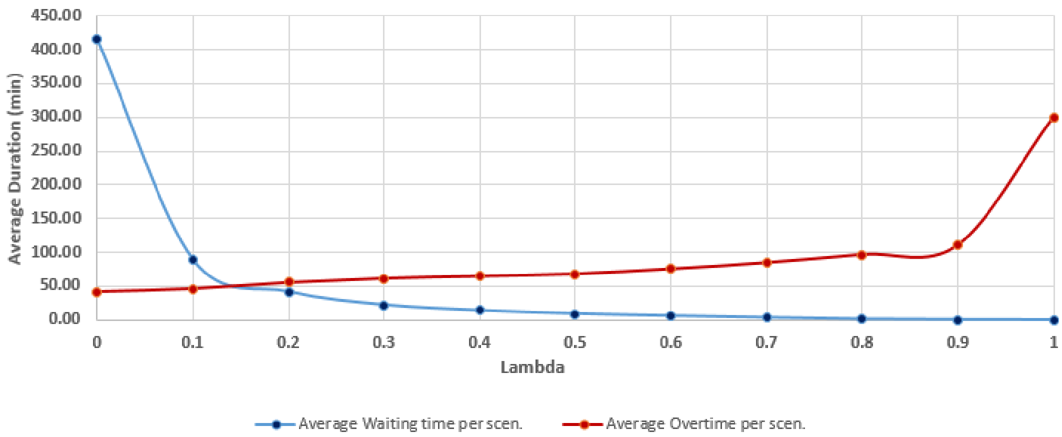

Figure 5.1: Average patient waiting time and nurse overtime values in changing λ

The manager may choose a λ value between those extreme points according to the goals of the units. In our experiments, we fix λ as 0.3, which assigns particular importance to nurse overtime, in the remainder of our experiments.

5.5.2 Varying values of number of nurses and chairs

The decision maker in the chemotherapy unit would also be interested in investigating the effect of using different levels of chairs and nurses on the performance measures. For this purpose, we vary the number of nurses and chairs in our instance sets. The chair and nurse numbers are varied betweeen 4 and 6, and 1 and 3, respectively. We do not consider the case where we have 3 nurses and 4 chairs since the corresponding nurse to chair ratio is not realistic. We repeat the experiments in 10 different instances under fixed λ value as 0.3 and take the average values of objective function values and run times in order to see the effect better. The summary is provided at Table 5.9.

As expected, when the number of chairs or nurses is increased, the average objective value decreases. The marginal benefits associated with an increase in the number of resources is higher when the base value is smaller. As an example, when the number of chairs is changed from 4 to 5, the objective value decreases by approximately 45% when there is only one nurse in the unit. On the other hand, if it is changed from 5 to 6, the decrease in the objective function is 25%.

Table 5.9: Sensitivity of the objective value and computational time to the number of chairs and nurses

		Objective Value			**Run Time (s)**		
		Number of Chairs			**Number of Chairs**		
		4	**5**	**6**	**4**	**5**	**6**
Number of Nurses	**1**	64.95	35.69	26.49	1428.88	1323.04	1338.4
	2	49.85	14.57	5.36	3111.90	2567.85	1357.32
	3	Not solved	9.62	1.49	Not solved	8337.14	4111.17

Another interesting issue is the change in the computation times. An increase in the number of chairs generally makes subproblems easier to solve resulting with a reduction in the solution times. However, if the number of nurses becomes larger, computation times increase significantly. Therefore, the number of nurses is a crucial factor that makes subproblems complicating to solve.

The chemotherapy unit manager should examine the trade-offs between the resource levels and performance measures. In some cases, it might not be a good idea to increase the resource levels in order to increase the service level of the unit since the effect in nurse overtime and patient waiting time might not be significant.

5.6 Estimating the Value of Stochastic Solution (VSS)

A measure for evaluating the success of stochastic programs is the value of stochastic solution. To identify this measure, expected value solution is needed. If all random variables are replaced with their expected values, we have a simple problem called *expected value problem* or *mean value problem*. Expected value solution denoted by $\bar{x}(\bar{\xi})$ is obtained from the mean value problem. Then, the optimal second-stage scenario subproblem solutions are found according to $\bar{x}(\bar{\xi})$ and ξ. The expected objective value associated with mean value solution is denoted as *EEV*. VSS is defined as the difference between EEV and the objective value of *recourse problem*, denoted by RP (i.e. VSS = EEV-RP). It measures the benefit of using stochastic programming solution over the mean value solution. In other words, VSS is a cost of excluding uncertainty while giving the first stage decisions.

In this part, we estimated the value of stochastic solution (VSS) for our instance set for five different λ values under fixed values of number of patients, nurses, and chairs. In Table 5.10, the average values of the VSS under different λ setting is provided.

Table 5.10: Average values of Value of stochastic solution (VSS) in changing λ values

λ	EEV	LPHA	VSS
0.3	60.32	49.85	17.4%
0.4	56.06	44.58	20.5%
0.5	51.88	38.72	25.4%
0.6	47.92	33.89	29.3%
0.7	44.62	27.89	37.5%

As expected, VSS is always positive in our experimental runs. Therefore, considering uncertainty in chemotherapy appointment scheduling problem is valuable to minimize total weighted sum of patient waiting time and nurse overtime. According to results provided at Table 5.10, VSS increases monotonically with increasing λ values in terms of quantity and percentage. The LPHA improves the solutions of mean value problem by 26% on average. The values of VSS for every instance based on different λ values are given at Table A.17 in Appendix.

CHAPTER 6

CONCLUSION

The ability of operations research to help improve the healthcare systems has been increasingly attracting the attention of healthcare practitioners. The need of using scarce resources effectively makes OR tools valuable for chemotherapy clinics, which aim to reduce costs while meeting certain service levels to satisfy the needs of the patients.

Over the recent years, due to the increase in the prevalence of cancer the demand for chemotherapy units has been growing, and these units should have efficient planning and scheduling structures. In this thesis, we focused on the Chemotherapy Appointment Scheduling Problem, where patients are sequenced and assigned appointment times by considering the availability of nurses and infusion chairs at the same time. The uncertainty in both the pre-medication and infusion durations are considered in the study. The aim is to design an appointment schedule in order to minimize the expected weighted sum of patient waiting time and nurse overtime. A two-stage stochastic mixed integer programming formulation is used to formulate the problem.

The uncertain durations of chemotherapy are represented with scenarios in the stochastic programming formulation. Solving the problem even with a small-size scenarios with CPLEX is computationally challenging. Therefore, a scenario decomposition based algorithm, Progressive Hedging, is implemented to solve the problem. The PHA is enhanced through a number of modifications. A penalty parameter update method is proposed, where the parameters are set dynamically and controlled using changing limits. A cycle detection method is used to guarantee convergence of the algorithm. A variable fixing procedure is incorporated to reduce overall computation times. Subproblem solution times are improved through symmetry-breaking constraints and bounds imposed on nurse overtime in the constraint set, and by linearization of the quadratic term in the objective function. The resulting algorithm after enhancements is called as linearized PHA (LPHA).

The proposed method is implemented based on real data from the Hacettepe Outpatient Chemotherapy Unit. To evaluate the performance of algorithm, the LPHA is compared with CPLEX in small-size problems. Moreover, combinations of common sequencing and job hedging heuristics are used to validate the performance of the algorithm. The results show that the LPHA outperforms commonly used heuristics in all cases. At last, VSS is estimated to assess the value of considering uncertainty in our problem. It is found that the LPHA improves the solutions of mean value problem significantly.

Our solution approach can be useful for chemotherapy unit managers to evaluate the effects of appointment schedules on nurse overtime and patient waiting time. The unit managers can also observe the effects of changing the level of resources in the chemotherapy unit.

For further research, availability of the lab tests and oncologist evaluations can be included in the problem. Furthermore, an integrated model that considers both the assignment of patients to treatment days and appointment time setting under uncertainty can be constructed. Moreover, variation in cancer types can be considered while assigning patients to nurses since the workload associated with different types of cancer patients can be different. In this study, we assumed that chemotherapy drugs are made ready in pharmacy before the associated patient arrives at the clinic. We can also add constraints related to pharmacist availability into our model structure.

Another important direction of further research is to enhance our algorithm performance by solving scenario subproblems using parallel computing. This application may reduce the computation times, which can be beneficial for solving instances of larger size.

REFERENCES

[1] OECD, "Health at a Glance 2017." https://www.oecd-ilibrary.org/content/publication/healthglance-2017-en, 2017. Accessed: November 2018.

[2] H. Ritchie and M. Roser, "Causes of death." https://ourworldindata.org/causes-of-death, 2018. Accessed: November 2018.

[3] Cancer Research UK, "Worldwide cancer statistics." https://www.cancerresearchuk.org/health-professional/cancer-statistics, 2018.

[4] National Cancer Institute, "Cancer Statistics." https://www.cancer.gov/about-cancer/understanding/statistics, 2018. Accessed: November 2018.

[5] Sağlık Bakanlığı, "Kanser İstatistikleri." https://dosyasb.saglik.gov.tr, 2017. Accessed: November 2018.

[6] 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*, vol. 6, no. 3, pp. 111–125, 2016.

[7] C. E. Saville, H. K. Smith, and K. Bijak, "Operational research techniques applied throughout cancer care services: a review," *Health Systems*, pp. 1–22, 2018.

[8] G. Lamé, O. Jouini, and J. S.-L. Cardinal, "Outpatient chemotherapy planning: A literature review with insights from a case study," *IIE Transactions on Healthcare Systems Engineering*, vol. 6, no. 3, pp. 127–139, 2016.

[9] D. Gupta and B. Denton, "Appointment scheduling in health care: Challenges and opportunities," *IIE transactions*, vol. 40, no. 9, pp. 800–819, 2008.

[10] T. Cayirli and E. Veral, "Outpatient scheduling in health care: a review of literature," *Production and operations management*, vol. 12, no. 4, pp. 519–549, 2003.

[11] A. Ahmadi-Javid, Z. Jalali, and K. J. Klassen, "Outpatient appointment systems in healthcare: A review of optimization studies," *European Journal of Operational Research*, vol. 258, no. 1, pp. 3–34, 2017.

[12] M. Heshmat and A. Eltawil, "Comparison between outpatient appointment scheduling and chemotherapy outpatient appointment scheduling," *The Egyptian International Journal of Engineering Sciences & Technology*, vol. 19, no. 2, pp. 326–332, 2016.

[13] A. Turkcan, B. Zeng, and M. Lawley, "Chemotherapy operations planning and scheduling," *IIE Transactions on Healthcare Systems Engineering*, vol. 2, no. 1, pp. 31–49, 2012.

[14] A. Condotta and N. Shakhlevich, "Scheduling patient appointments via multilevel template: A case study in chemotherapy," *Operations Research for Health Care*, vol. 3, no. 3, pp. 129–144, 2014.

[15] M. Heshmat and A. Eltawil, "A new sequential approach for chemotherapy treatment and facility operations planning," *Operations Research for Health Care*, vol. 18, pp. 33–40, 2018.

[16] S. Hahn-Goldberg, J. C. Beck, M. W. Carter, M. Trudeau, P. Sousa, and K. Beattie, "Solving the chemotherapy outpatient scheduling problem with constraint programming," *Journal of Applied Operational Research*, vol. 6, no. 3, pp. 135–144, 2014.

[17] A. Sadki, X. Xie, and F. Chauvin, "Appointment scheduling of oncology outpatients," in *2011 IEEE International Conference on Automation Science and Engineering*, pp. 513–518, IEEE, 2011.

[18] B. Liang and A. Turkcan, "Acuity-based nurse assignment and patient scheduling in oncology clinics," *Health care management science*, vol. 19, no. 3, pp. 207–226, 2016.

[19] P. Santibáñez, R. Aristizabal, M. L. Puterman, V. S. Chow, W. Huang, C. Kollmannsberger, T. Nordin, N. Runzer, and S. Tyldesley, "Operations research methods improve chemotherapy patient appointment scheduling," *The Joint Commission Journal on Quality and Patient Safety*, vol. 38, no. 12, pp. 541–AP2, 2012.

[20] R. Dobish, "Next-day chemotherapy scheduling: a multidisciplinary approach to solving workload issues in a tertiary oncology center," *Journal of Oncology Pharmacy Practice*, vol. 9, no. 1, pp. 37–42, 2003.

[21] 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*, vol. 124, pp. 347–358, 2018.

[22] A. Huggins, D. Claudio, and E. Pérez, "Improving resource utilization in a cancer clinic: an optimization model," in *Proceedings of the 2014 Industrial and Systems Engineering Research Conference*, p. 1444, 2014.

[23] S. Sevinc, U. A. Sanli, and E. Goker, "Algorithms for scheduling of chemotherapy plans," *Computers in biology and medicine*, vol. 43, no. 12, pp. 2103–2109, 2013.

[24] 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*, 2018.

[25] 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*, vol. 17, no. 1, pp. 60–76, 2014.

[26] M. Alvarado and L. Ntaimo, "Chemotherapy appointment scheduling under uncertainty using mean-risk stochastic integer programming," *Health care management science*, vol. 21, no. 1, pp. 87–104, 2018.

[27] T. Tanaka, *Infusion chair scheduling algorithms based on bin-packing heuristics*. State University of New York at Binghamton, 2011.

[28] A. Mandelbaum, P. Momcilovic, N. Trichakis, S. Kadish, R. Leib, and C. A. Bunnell, “Data-driven appointment-scheduling under uncertainty: The case of an infusion unit in a cancer center,”

[29] 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*, vol. 53, no. 24, pp. 7177–7190, 2015.

[30] A. Shapiro and A. Philpott, “A tutorial on stochastic programming,” *Manuscript. Available at www2. isye. gatech. edu/ashapiro/publications. html*, vol. 17, 2007.

[31] C. Mancilla and R. Storer, “A sample average approximation approach to stochastic appointment sequencing and scheduling,” *IIE Transactions*, vol. 44, no. 8, pp. 655–670, 2012.

[32] J. R. Birge and F. Louveaux, *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[33] R. M. Van Slyke and R. Wets, “L-shaped linear programs with applications to optimal control and stochastic programming,” *SIAM Journal on Applied Mathematics*, vol. 17, no. 4, pp. 638–663, 1969.

[34] G. Laporte and F. V. Louveaux, “The integer l-shaped method for stochastic integer programs with complete recourse,” *Operations research letters*, vol. 13, no. 3, pp. 133–142, 1993.

[35] S. Sen and J. L. Higle, “The c3 theorem and a d2 algorithm for large scale stochastic mixed-integer programming: set convexification,” *Mathematical Programming*, vol. 104, no. 1, pp. 1–20, 2005.

[36] Y. Yuan and S. Sen, “Enhanced cut generation methods for decomposition-based branch and cut for two-stage stochastic mixed-integer programs,” *INFORMS Journal on Computing*, vol. 21, no. 3, pp. 480–487, 2009.

[37] S. Gul, B. T. Denton, and J. W. Fowler, “A progressive hedging approach for surgery planning under uncertainty,” *INFORMS Journal on Computing*, vol. 27, no. 4, pp. 755–772, 2015.

[38] L. M. Hvattum and A. Løkketangen, "Using scenario trees and progressive hedging for stochastic inventory routing problems," *Journal of Heuristics*, vol. 15, no. 6, p. 527, 2009.

[39] R. E. Gonçalves, E. C. Finardi, and E. L. da Silva, "Applying different decomposition schemes using the progressive hedging algorithm to the operation planning problem of a hydrothermal system," *Electric Power Systems Research*, vol. 83, no. 1, pp. 19–27, 2012.

[40] R. T. Rockafellar and R. J.-B. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of operations research*, vol. 16, no. 1, pp. 119–147, 1991.

[41] T. G. Crainic, X. Fu, M. Gendreau, W. Rei, and S. W. Wallace, "Progressive hedging-based metaheuristics for stochastic network design," *Networks*, vol. 58, no. 2, pp. 114–124, 2011.

[42] J.-P. Watson and D. L. Woodruff, "Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems," *Computational Management Science*, vol. 8, no. 4, pp. 355–370, 2011.

[43] D. Gade, G. Hackebeil, S. M. Ryan, J.-P. Watson, R. J.-B. Wets, and D. L. Woodruff, "Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs," *Mathematical Programming*, vol. 157, no. 1, pp. 47–67, 2016.

[44] K. Cheung, D. Gade, C. Silva-Monroy, S. M. Ryan, J.-P. W. , R. J.-B. Wets, and D. L. Woodruff, "Toward scalable stochastic unit commitment," *Energy Systems*, vol. 6, no. 3, pp. 417–438, 2015.

[45] J. M. Mulvey and H. Vladimirou, "Applying the progressive hedging algorithm to stochastic generalized networks," *Annals of Operations Research*, vol. 31, no. 1, pp. 399–424, 1991.

[46] S. Atakan and S. Sen, "A progressive hedging based branch-and-bound algorithm for mixed-integer stochastic programs," *Computational Management Science*, pp. 1–40, 2018.

[47] T. Helgason and S. W. Wallace, "Approximate scenario solutions in the progressive hedging algorithm," *Annals of Operations Research*, vol. 31, no. 1, pp. 425–444, 1991.

[48] J.-P. Watson, D. L. Woodruff, and W. E. Hart, "Pysp: modeling and solving stochastic programs in python," *Mathematical Programming Computation*, vol. 4, no. 2, pp. 109–149, 2012.

[49] A. Helseth, "Stochastic network constrained hydro-thermal scheduling using a linearized progressive hedging algorithm," *Energy Systems*, vol. 7, no. 4, pp. 585–600, 2016.

[50] S. Gul, "A stochastic programming approach for appointment scheduling under limited availability of surgery turnover teams," *Service Science*, vol. 10, no. 3, pp. 277–288, 2018.

[51] S. Gul, B. T. Denton, J. W. Fowler, and T. Huschka, "Bi-criteria scheduling of surgical services for an outpatient procedure center," *Production and Operations management*, vol. 20, no. 3, pp. 406–417, 2011.

APPENDIX A

APPENDIX

A.1 Deterministic Equivalent Model

$$min \quad \left(\lambda \sum_{i \in I} \sum_{\omega \in \Omega} p^{\omega} * w_i^{\omega} + (1-\lambda) \sum_{n \in N} \sum_{\omega \in \Omega} p^{\omega} * O_n^{\omega}\right) \tag{A.1}$$

$$\sum_{n \in N} x_{in}^{\omega} = 1 \qquad \forall i, \forall \omega \tag{A.2}$$

$$\sum_{c \in C} y_{ic}^{\omega} = 1 \qquad \forall i, \forall \omega \tag{A.3}$$

$$b_{ij}^{\omega} + b_{ji}^{\omega} = 1 \qquad \forall i, j > i, \forall \omega \tag{A.4}$$

$$a_i^{\omega} + w_i^{\omega} + s_i^{\omega} + t_i^{\omega} = d_i^{\omega} \qquad \forall i, \forall \omega \tag{A.5}$$

$$a_j^{\omega} + w_j^{\omega} \geq a_i^{\omega} + w_i^{\omega} + s_i^{\omega} - M * (3 - b_{ij} - x_{in}^{\omega} - x_{jn}^{\omega}) \qquad \forall i, j \neq i, \forall n, \forall \omega \tag{A.6}$$

$$a_j + w_j^{\omega} \geq d_i^{w} - M * (3 - b_{ij} - y_{ic}^{\omega} - y_{jc}^{\omega}) \qquad \forall i, j \neq i, \forall c, \forall \omega \tag{A.7}$$

$$a_j + w_j^{\omega} \geq a_i + w_i^{\omega} - M * (1 - b_{ij}) \qquad \forall i, j \neq i, \forall \omega \tag{A.8}$$

$$a_j^{\omega} \geq a_i^{\omega} - M(1 - b_{ij}^{\omega}) \qquad \forall i, \forall j, j \neq i \tag{A.9}$$

$$O_n^{\omega} \geq d_i^{\omega} - H - M * (1 - x_{in}^{\omega}) \qquad \forall i, \forall n, \forall \omega \tag{A.10}$$

$$a_i^{\omega} = a_i \qquad \forall i, \forall \omega \tag{A.11}$$

$$b_{ij}^{\omega} \in \{0, 1\} \qquad \forall i, \forall j, \forall \omega \tag{A.12}$$

$$a_i^{\omega} : integer \qquad \forall i, \forall \omega \tag{A.13}$$

$$d_i^{\omega}, w_i^{\omega} \geq 0 \qquad \forall i, \forall \omega \tag{A.14}$$

$$O_n^{\omega} \geq 0 \qquad \forall n, \forall \omega \tag{A.15}$$

$$x_{in}^{\omega} \in \{0, 1\} \qquad \forall i, \forall n, \forall \omega \tag{A.16}$$

$$y_{ic}^{\omega} \in \{0, 1\} \qquad \forall i, \forall c, \forall \omega \tag{A.17}$$

A.2 Computational Results

Table A.1: The comparison of CPLEX and LPHA in terms of run times and solution quality

Instance Name	λ	CPLEX Run Time (s)	LPHA Run Time (s)	Optimality Gap	Instance Name	λ	CPLEX Run Time (s)	LPHA Run Time (s)	Optimality Gap
1_7_5	0.3	1170.83	57.24	0.00%	1_7_5	0.4	362.52	109.89	0.00%
2_7_5	0.3	3240.29	93.90	1.99%	2_7_5	0.4	2156.81	61.45	2.46%
3_7_5	0.3	128.08	87.60	1.71%	3_7_5	0.4	933.56	89.58	5.60%
4_7_5	0.3	1844.45	189.23	0.11%	4_7_5	0.4	1432.53	153.41	20.38%
5_7_5	0.3	296.37	76.53	5.92%	5_7_5	0.4	151.23	88.38	6.76%
6_7_5	0.3	1700.64	62.11	0.21%	6_7_5	0.4	182.79	91.95	0.00%
7_7_5	0.3	2346.25	105.45	3.88%	7_7_5	0.4	1362.48	153.93	0.32%
8_7_5	0.3	1260.36	119.79	4.60%	8_7_5	0.4	380.14	144.58	10.23%
9_7_5	0.3	67.43	82.39	0.00%	9_7_5	0.4	42.72	75.85	15.03%
10_7_5	0.3	1438.54	170.56	11.17%	10_7_5	0.4	4056.88	143.27	6.47%
1_7_5	0.5	177.53	69.60	0.00%	1_7_5	0.6	506.65	68.48	22.36%
2_7_5	0.5	1761.27	92.76	0.00%	2_7_5	0.6	423.79	83.69	13.73%
3_7_5	0.5	166.52	70.96	7.30%	3_7_5	0.6	123.64	103.29	10.96%
4_7_5	0.5	714.66	121.44	10.22%	4_7_5	0.6	1045.98	242.22	7.68%
5_7_5	0.5	204.49	94.31	3.18%	5_7_5	0.6	30.45	74.90	6.60%
6_7_5	0.5	98.00	74.48	0.35%	6_7_5	0.6	78.27	81.95	17.33%
7_7_5	0.5	3245.87	101.99	13.67%	7_7_5	0.6	1030.65	80.51	13.05%
8_7_5	0.5	139.54	120.72	9.34%	8_7_5	0.6	85.60	95.44	12.12%
9_7_5	0.5	18.02	18.02	0.00%	9_7_5	0.6	11.40	87.79	3.88%
10_7_5	0.5	250.64	151.71	9.65%	10_7_5	0.6	252.95	252.95	19.93%
1_7_5	0.7	28.52	81.32	0.19%	6_7_5	0.7	89.42	125.24	9.31%
2_7_5	0.7	82.45	89.53	4.76%	7_7_5	0.7	569.01	118.51	10.48%
3_7_5	0.7	50.29	102.48	15.07%	8_7_5	0.7	260.97	125.25	17.04%
4_7_5	0.7	571.25	155.46	2.80%	9_7_5	0.7	6.39	62.71	3.92%
5_7_5	0.7	98.81	144.71	5.43%	10_7_5	0.7	404.12	158.14	16.72%

Table A.2: The objective function values and computational times in λ=0

Instance Name	**λ**	**Objective Value**	**Average Waiting time per scen.**	**Average Overtime per scen.**	**Run Time (s)**	**Number of Iterations**
1_8_50	0	40.64	330.34	40.64	527.2	1
2_8_50	0	26.26	370.14	26.26	4952.4	1
3_8_50	0	53.32	562.92	53.32	41931.0	87
4_8_50	0	26.54	505.62	26.54	5946.2	75
5_8_50	0	43.72	405.22	43.72	9928.2	73
6_8_50	0	69.74	205.82	69.74	14743.3	141
7_8_50	0	25.96	385.52	25.96	33398.2	45
8_8_50	0	41.76	412.54	41.76	25366.1	88
9_8_50	0	48.32	582.86	48.32	10391.1	79
10_8_50	0	43.52	386.22	43.52	2090.1	45

Table A.3: The objective function values and computational times in λ=0.1

Instance Name	**λ**	**Objective Value**	**Average Waiting time per scen.**	**Average Overtime per scen.**	**Run Time (s)**	**Number of Iterations**
1_8_50	0.1	48.01	71.46	45.40	2864.67	79
2_8_50	0.1	43.32	69.10	40.46	4754.1	89
3_8_50	0.1	62.94	82.78	60.74	2452.78	97
4_8_50	0.1	33.77	67.34	30.04	4086.91	107
5_8_50	0.1	56.39	142.16	46.86	6063.13	83
6_8_50	0.1	51.23	83.36	47.66	3186.1	94
7_8_50	0.1	47.48	127.58	38.58	8484.27	81
8_8_50	0.1	53.78	75.92	51.32	3099.45	106
9_8_50	0.1	59.59	96.04	55.54	3673.18	102
10_8_50	0.1	49.73	73.38	47.10	5140.55	102

Table A.4: The objective function values and computational times in λ=0.2

Instance Name	λ	Objective Value	Average Waiting time per scen.	Average Overtime per scen.	Run Time (s)	Number of Iterations
1_8_50	0.2	51.96	40.12	54.92	3417.39	120
2_8_50	0.2	46.61	46.66	46.60	8083.15	123
3_8_50	0.2	65.50	35.58	72.98	2630.93	81
4_8_50	0.2	37.11	37.48	37.02	4172.93	123
5_8_50	0.2	57.12	40.24	61.34	3845.47	109
6_8_50	0.2	52.03	38.62	55.38	2744.92	104
7_8_50	0.2	46.79	39.46	48.62	7130.74	109
8_8_50	0.2	56.92	48.34	59.06	3259.41	107
9_8_50	0.2	63.35	53.96	65.70	2884.88	114
10_8_50	0.2	50.17	37.82	53.26	4002.62	109

Table A.5: The objective function values and computational times in λ=0.3

Instance Name	λ	Objective Value	Average Waiting time per scen.	Average Overtime per scen.	Run Time (s)	Number of Iterations
1_8_50	0.3	48.53	22.08	59.86	2866.4	107
2_8_50	0.3	47.78	25.38	57.38	6651.35	124
3_8_50	0.3	59.92	24.86	74.94	2103.39	109
4_8_50	0.3	35.35	18.98	42.36	4187.46	120
5_8_50	0.3	53.72	14.34	70.60	3555.59	103
6_8_50	0.3	51.50	20.92	64.60	3521.03	106
7_8_50	0.3	41.96	21.94	50.54	5682.71	112
8_8_50	0.3	53.82	26.76	65.42	3084.82	118
9_8_50	0.3	57.72	25.20	71.66	2225.59	110
10_8_50	0.3	48.25	22.70	59.20	3614.74	111

Table A.6: The objective function values and computational times in λ=0.4

Instance Name	λ	Objective Value	Average Waiting time per scen.	Average Overtime per scen.	Run Time (s)	Number of Iterations
1_8_50	0.4	43.56	12.18	64.48	2530.1	111
2_8_50	0.4	42.28	17.64	58.70	5844.23	104
3_8_50	0.4	53.04	15.20	78.26	2183.64	106
4_8_50	0.4	31.87	12.72	44.64	5560.51	114
5_8_50	0.4	48.53	15.82	70.34	2973.45	113
6_8_50	0.4	45.85	12.36	68.18	2380.83	105
7_8_50	0.4	37.15	9.14	55.82	4851.53	92
8_8_50	0.4	46.64	13.02	69.06	2337.24	106
9_8_50	0.4	52.73	16.38	76.96	2352.67	116
10_8_50	0.4	44.12	17.04	62.18	3513.1	107

Table A.7: The objective function values and computational times in λ=0.5

Instance Name	λ	Objective Value	Average Waiting time per scen.	Average Overtime per scen.	Run Time (s)	Number of Iterations
1_8_50	0.5	38.03	7.16	68.90	2451.82	102
2_8_50	0.5	35.36	10.18	60.54	3944.89	106
3_8_50	0.5	48.09	14.60	81.58	1986.8	124
4_8_50	0.5	27.73	6.38	49.08	3110.06	116
5_8_50	0.5	43.15	9.18	77.12	2796.64	122
6_8_50	0.5	38.07	7.32	68.82	1572.54	109
7_8_50	0.5	32.49	8.80	56.18	4815.07	109
8_8_50	0.5	40.44	7.34	73.54	3117.24	110
9_8_50	0.5	46.39	13.82	78.96	2134.49	119
10_8_50	0.5	37.44	7.48	67.40	3690.67	113

Table A.8: The objective function values and computational times in λ=0.6

Instance Name	λ	**Objective Value**	**Average Waiting time per scen.**	**Average Overtime per scen.**	**Run Time (s)**	**Number of Iterations**
1_8_50	0.6	33.03	2.88	78.26	2408.78	118
2_8_50	0.6	31.21	7.96	66.08	5257.42	106
3_8_50	0.6	40.32	9.72	86.22	2791.85	123
4_8_50	0.6	25.04	3.18	57.84	3818.79	101
5_8_50	0.6	34.82	7.08	76.44	2426.51	111
6_8_50	0.6	38.74	6.60	86.96	1813.71	106
7_8_50	0.6	32.09	5.80	71.52	3790.68	109
8_8_50	0.6	33.01	5.06	74.94	2228.72	84
9_8_50	0.6	38.23	6.06	86.48	1607.98	106
10_8_50	0.6	32.39	7.28	70.06	4147.56	98

Table A.9: The objective function values and computational times in λ=0.7

Instance Name	λ	**Objective Value**	**Average Waiting time per scen.**	**Average Overtime per scen.**	**Run Time (s)**	**Number of Iterations**
1_8_50	0.7	27.11	1.42	87.06	1644.6	105
2_8_50	0.7	28.24	6.00	80.14	3080.84	117
3_8_50	0.7	31.17	5.42	91.26	1713.83	107
4_8_50	0.7	23.78	2.26	73.98	3241.36	109
5_8_50	0.7	33.81	3.20	105.24	1474.77	121
6_8_50	0.7	25.38	3.80	75.74	1490.14	184
7_8_50	0.7	21.57	3.72	63.22	4996.25	109
8_8_50	0.7	33.17	0.50	109.40	1993.38	132
9_8_50	0.7	29.89	4.88	88.26	1550.67	90
10_8_50	0.7	24.81	3.96	73.46	4183.31	114

Table A.10: The objective function values and computational times in λ=0.8

Instance Name	$\boldsymbol{\lambda}$	**Objective Value**	**Average Waiting time per scen.**	**Average Overtime per scen.**	**Run Time (s)**	**Number of Iterations**
1_8_50	0.8	22.34	0.90	108.10	1330.26	104
2_8_50	0.8	19.74	1.28	93.60	3986.52	109
3_8_50	0.8	23.16	0.94	112.06	1233.73	117
4_8_50	0.8	18.20	1.94	83.22	4466	137
5_8_50	0.8	20.60	1.40	97.42	1348.25	120
6_8_50	0.8	17.44	2.36	77.78	1430.71	108
7_8_50	0.8	16.28	2.24	72.44	3069.5	116
8_8_50	0.8	24.26	0.80	118.12	1268.36	114
9_8_50	0.8	23.17	0.60	113.46	1326.47	138
10_8_50	0.8	19.62	1.36	92.68	1988.29	127

Table A.11: The objective function values and computational times in λ=0.9

Instance Name	$\boldsymbol{\lambda}$	**Objective Value**	**Average Waiting time per scen.**	**Average Overtime per scen.**	**Run Time (s)**	**Number of Iterations**
1_8_50	0.9	10.72	0.10	106.28	891.30	119
2_8_50	0.9	11.86	0.98	109.76	1435.51	114
3_8_50	0.9	11.58	1.42	103.04	854.67	107
4_8_50	0.9	11.22	0.10	111.30	2739.09	114
5_8_50	0.9	12.60	0.26	123.66	1217.01	113
6_8_50	0.9	13.72	1.80	121.00	1296.95	114
7_8_50	0.9	9.69	0.34	93.88	2106.68	105
8_8_50	0.9	9.80	0.34	94.90	877.36	99
9_8_50	0.9	11.26	0.08	111.84	969.25	90
10_8_50	0.9	13.28	0.10	131.92	1502.72	107

Table A.12: The objective function values and computational times in λ=1

Instance Name	λ	Objective Value	Average Waiting time per scen.	Average Overtime per scen.	Run Time (s)	Number of Iterations
1_8_50	1	0.06	0.06	300.00	1102.28	85
2_8_50	1	0.00	0.00	300.00	903.70	101
3_8_50	1	0.02	0.02	300.00	832.04	87
4_8_50	1	0.00	0.00	300.00	1091.66	63
5_8_50	1	0.16	0.16	300.00	814.09	72
6_8_50	1	0.00	0.00	300.00	1407.37	87
7_8_50	1	0.00	0.00	300.00	1656.73	92
8_8_50	1	0.10	0.10	300.00	1238.44	90
9_8_50	1	0.02	0.02	300.00	1010.90	100
10_8_50	1	0.02	0.02	300.00	1241.01	85

Table A.13: Objective values and run times of job hedging heuristics for SPT rule

Instance Name	Percentile	Objective Value	Run Time (s)	LPHA Objective Value	Instance Name	Percentile	Objective Value	Run Time (s)	LPHA Objective Value
1_8_50	40%	73.86	4.94	48.53	6_8_50	40%	89.40	3.40	51.50
	45%	72.78	5.30			45%	86.20	3.69	
	50%	71.15	3.82			50%	84.77	3.00	
	55%	70.38	2.23			55%	84.19	3.07	
	60%	69.55	3.08			60%	85.69	2.17	
	65%	71.41	2.92			65%	86.08	1.90	
	70%	72.41	2.00			70%	88.95	1.85	
	75%	73.93	2.11			75%	90.19	2.03	
2_8_50	40%	60.44	2.22	47.78	7_8_50	40%	79.50	6.19	41.96
	45%	57.44	2.10			45%	78.61	4.04	
	50%	54.48	1.86			50%	76.40	3.05	
	55%	53.50	1.86			55%	76.008	3.92	
	60%	54.75	1.80			60%	76.47	3.12	
	65%	56.41	1.45			65%	78.46	2.12	
	70%	60.73	1.54			70%	81.44	2.70	
	75%	62.68	1.57			75%	82.63	2.47	
3_8_50	40%	86.88	5.23	59.92	8_8_50	40%	91.48	5.35	53.82
	45%	85.86	5.19			45%	90.05	2.72	
	50%	84.29	5.67			50%	88.03	2.74	
	55%	84.56	2.85			55%	87.73	3.00	
	60%	85.69	1.98			60%	88.22	2.09	
	65%	85.68	2.47			65%	88.43	1.94	
	70%	88.11	2.02			70%	93.49	1.82	
	75%	89.08	2.11			75%	95.21	1.77	
4_8_50	40%	73.53	6.44	35.35	9_8_50	40%	90.31	6.95	57.72
	45%	72.54	6.12			45%	88.37	6.04	
	50%	71.22	4.91			50%	86.76	3.22	
	55%	71.21	4.41			55%	86.71	3.44	
	60%	70.95	3.43			60%	86.80	2.88	
	65%	71.61	2.96			65%	86.72	2.06	
	70%	73.09	2.08			70%	89.32	1.81	
	75%	75.76	2.00			75%	89.86	1.95	
5_8_50	40%	83.17	4.68	53.72	10_8_50	40%	77.12	4.76	48.25
	45%	81.88	3.72			45%	75.74	6.29	
	50%	79.81	3.25			50%	74.57	3.07	
	55%	79.57	4.31			55%	74.40	4.21	
	60%	79.62	2.31			60%	74.08	3.27	
	65%	80.29	2.47			65%	74.44	2.11	
	70%	82.06	1.72			70%	76.12	2.49	
	75%	84.12	1.84			75%	77.12	1.41	

Table A.14: Objective values and run times of job hedging heuristics for LPT rule

Instance Name	Percentile	Objective Value	Run Time (s)	LPHA Objective Value	Instance Name	Percentile	Objective Value	Run Time (s)	LPHA Objective Value
1_8_50	40%	68.20	7.08	48.53	6_8_50	40%	66.62	6.21	51.50
	45%	66.67	6.71			45%	67.17	6.43	
	50%	67.02	6.26			50%	65.77	3.66	
	55%	71.18	5.63			55%	65.30	3.43	
	60%	71.68	4.75			60%	69.83	2.54	
	65%	70.58	2.57			65%	69.48	3.04	
	70%	78.53	2.51			70%	72.36	1.77	
	75%	78.95	1.46			75%	76.47	2.29	
2_8_50	40%	63.27	5.37	47.78	7_8_50	40%	55.98	8.16	41.96
	45%	60.47	5.50			45%	52.42	7.10	
	50%	59.59	4.19			50%	52.87	7.56	
	55%	59.80	4.32			55%	54.20	3.76	
	60%	62.52	4.84			60%	54.61	3.06	
	65%	69.01	2.22			65%	53.81	3.20	
	70%	72.66	2.58			70%	55.63	1.58	
	75%	74.37	2.29			75%	57.62	2.25	
3_8_50	40%	82.47	5.50	59.92	8_8_50	40%	72.23	3.92	53.82
	45%	82.47	4.65			45%	69.39	3.60	
	50%	84.10	3.86			50%	68.97	3.67	
	55%	83.77	2.43			55%	72.58	3.64	
	60%	83.85	5.27			60%	73.22	3.26	
	65%	85.79	2.66			65%	74.41	2.37	
	70%	83.05	2.54			70%	78.74	1.45	
	75%	88.82	1.48			75%	78.63	1.73	
4_8_50	40%	51.01	6.66	35.35	9_8_50	40%	77.17	7.54	57.72
	45%	48.15	6.76			45%	78.21	6.34	
	50%	45.94	6.34			50%	76.53	4.29	
	55%	44.69	6.78			55%	77.20	2.73	
	60%	43.53	6.43			60%	81.02	1.68	
	65%	47.91	3.29			65%	83.19	2.01	
	70%	48.37	2.34			70%	82.63	1.96	
	75%	49.84	2.17			75%	86.89	1.41	
5_8_50	40%	74.90	7.87	53.72	10_8_50	40%	72.10	6.12	48.25
	45%	74.53	7.50			45%	70.74	6.11	
	50%	73.76	7.62			50%	74.16	5.37	
	55%	73.12	6.47			55%	75.64	2.33	
	60%	73.52	3.50			60%	78.25	3.11	
	65%	74.67	2.28			65%	78.37	2.34	
	70%	80.73	1.68			70%	82.46	1.21	
	75%	84.83	1.82			75%	89.32	1.18	

Table A.15: Objective values and run times of job hedging heuristics for VAR rule

Instance Name	Percentile	Objective Value	Run Time (s)	LPHA Objective Value	Instance Name	Percentile	Objective Value	Run Time (s)	LPHA Objective Value
1_8_50	40%	69.09	6.17	48.53	6_8_50	40%	87.45	5.98	51.50
	45%	68.11	5.96			45%	85.23	6.67	
	50%	66.57	3.75			50%	83.72	2.08	
	55%	65.91	3.49			55%	82.86	2.10	
	60%	65.05	3.04			60%	83.91	2.22	
	65%	67.08	2.98			65%	85.38	2.00	
	70%	68.16	2.06			70%	86.19	1.94	
	75%	69.64	2.70			75%	87.26	1.73	
2_8_50	40%	59.68	4.55	47.78	7_8_50	40%	77.14	5.70	41.96
	45%	56.90	4.25			45%	76.26	4.91	
	50%	54.03	4.34			50%	74.12	3.93	
	55%	53.33	3.94			55%	73.58	3.53	
	60%	55.16	2.69			60%	73.87	2.64	
	65%	56.91	1.53			65%	75.95	2.17	
	70%	61.19	1.79			70%	78.62	1.92	
	75%	63.34	2.47			75%	79.90	1.77	
3_8_50	40%	87.06	5.06	59.92	8_8_50	40%	88.48	5.86	53.82
	45%	85.83	6.32			45%	86.06	3.14	
	50%	84.01	5.18			50%	82.25	3.15	
	55%	84.09	3.34			55%	81.28	3.48	
	60%	85.14	2.47			60%	80.70	2.02	
	65%	85.11	2.30			65%	82.37	2.29	
	70%	87.34	1.93			70%	81.74	2.08	
	75%	88.51	1.93			75%	83.51	1.71	
4_8_50	40%	71.17	5.57	35.35	9_8_50	40%	90.00	5.75	57.72
	45%	70.30	5.73			45%	88.19	4.07	
	50%	69.16	5.47			50%	86.58	3.31	
	55%	69.30	3.60			55%	86.50	3.28	
	60%	69.62	2.84			60%	86.77	3.11	
	65%	70.51	2.58			65%	86.71	2.36	
	70%	71.72	2.58			70%	89.27	1.90	
	75%	72.45	2.67			75%	89.75	2.07	
5_8_50	40%	78.68	4.44	53.72	10_8_50	40%	74.62	6.27	48.25
	45%	76.47	5.03			45%	73.32	5.81	
	50%	74.27	3.12			50%	72.19	6.12	
	55%	74.01	3.70			55%	72.39	3.22	
	60%	74.57	2.60			60%	71.86	2.61	
	65%	77.14	2.30			65%	72.66	2.62	
	70%	76.26	2.40			70%	73.21	2.56	
	75%	74.12	1.74			75%	74.47	1.40	

Table A.16: Objective values and run times of job hedging heuristics for increasing CoV rule

Instance Name	Percentile	Objective Value	Run Time (s)	LPHA Objective Value	Instance Name	Percentile	Objective Value	Run Time (s)	LPHA Objective Value
1_8_50	40%	67.95	7.793	48.53	6_8_50	40%	66.84	3.934	51.50
	45%	66.67	6.50			45%	67.22	3.97	
	50%	67.24	6.18			50%	66.27	3.56	
	55%	71.68	6.43			55%	65.76	3.61	
	60%	72.06	5.01			60%	70.98	3.01	
	65%	70.37	1.83			65%	70.72	2.48	
	70%	78.50	2.87			70%	72.05	2.23	
	75%	68.56	1.82			75%	76.35	2.03	
2_8_50	40%	63.74	6.17	47.78	7_8_50	40%	56.92	8.00	41.96
	45%	60.95	6.66			45%	54.08	6.74	
	50%	61.01	7.93			50%	52.41	3.69	
	55%	60.96	7.31			55%	53.13	7.18	
	60%	64.00	3.48			60%	53.85	3.75	
	65%	69.48	3.71			65%	53.61	2.96	
	70%	73.58	1.62			70%	55.69	2.66	
	75%	75.40	3.20			75%	57.18	2.34	
3_8_50	40%	93.90	5.85	59.92	8_8_50	40%	73.24	4.66	53.82
	45%	92.46	3.87			45%	71.79	3.88	
	50%	91.14	3.73			50%	69.18	3.41	
	55%	90.83	2.83			55%	69.08	2.83	
	60%	91.04	2.89			60%	68.53	2.53	
	65%	91.63	3.08			65%	70.72	2.06	
	70%	92.13	2.23			70%	72.90	1.83	
	75%	93.81	2.12			75%	76.28	1.72	
4_8_50	40%	68.64	6.37	35.35	9_8_50	40%	82.69	4.20	57.72
	45%	65.07	4.02			45%	81.16	4.10	
	50%	64.09	3.38			50%	80.26	3.57	
	55%	63.58	3.29			55%	79.17	2.54	
	60%	64.26	2.64			60%	77.91	2.52	
	65%	65.21	2.27			65%	77.59	2.44	
	70%	66.89	1.84			70%	78.21	1.82	
	75%	68.18	2.57			75%	79.16	2.27	
5_8_50	40%	83.18	5.47	53.72	10_8_50	40%	73.25	4.99	48.25
	45%	80.75	4.48			45%	70.96	4.60	
	50%	79.02	3.37			50%	67.66	3.22	
	55%	78.71	2.90			55%	66.66	2.64	
	60%	78.64	2.55			60%	64.34	2.36	
	65%	56.92	2.23			65%	64.60	2.87	
	70%	54.08	2.10			70%	67.59	2.13	
	75%	52.41	1.86			75%	68.56	2.58	

Table A.17: VSS values for 10 different instances in varying λ values

Instance Name	**lambda**	**EEV**	**LPHA**	**VSS**	**Instance Name**	**lambda**	**EEV**	**LPHA**	**VSS**
1_8_50	0.3	63.36	48.53	23.4%	1_8_50	0.4	56.43	43.56	12.87
2_8_50	0.3	49.95	47.78	4.3%	2_8_50	0.4	48.88	42.28	13.5%
3_8_50	0.3	72.00	59.92	16.8%	3_8_50	0.4	64.49	53.04	17.8%
4_8_50	0.3	39.48	35.35	10.5%	4_8_50	0.4	37.63	31.87	15.3%
5_8_50	0.3	66.74	53.72	19.5%	5_8_50	0.4	62.10	48.53	21.8%
6_8_50	0.3	64.84	51.50	20.6%	6_8_50	0.4	61.96	45.85	26.0%
7_8_50	0.3	49.27	41.96	14.8%	7_8_50	0.4	47.56	37.15	21.9%
8_8_50	0.3	64.23	53.82	16.2%	8_8_50	0.4	59.50	46.64	21.6%
9_8_50	0.3	69.84	57.72	17.4%	9_8_50	0.4	64.09	52.73	17.7%
10_8_50	0.3	63.47	48.25	24.0%	10_8_50	0.4	57.97	44.12	23.9%
1_8_50	0.5	52.08	38.03	27.0%	1_8_50	0.6	46.50	33.03	29.0%
2_8_50	0.5	47.45	35.36	25.5%	2_8_50	0.6	45.94	31.21	32.1%
3_8_50	0.5	58.87	48.09	18.3%	3_8_50	0.6	54.57	40.32	26.1%
4_8_50	0.5	36.41	27.73	23.8%	4_8_50	0.6	34.22	25.04	26.8%
5_8_50	0.5	57.20	43.15	24.6%	5_8_50	0.6	51.94	34.82	32.9%
6_8_50	0.5	55.22	38.07	31.1%	6_8_50	0.6	50.84	38.74	23.8%
7_8_50	0.5	43.41	32.49	25.2%	7_8_50	0.6	41.24	32.09	22.2%
8_8_50	0.5	54.45	40.44	25.7%	8_8_50	0.6	49.67	33.01	33.5%
9_8_50	0.5	61.98	46.39	25.2%	9_8_50	0.6	55.50	38.23	31.1%
10_8_50	0.5	51.73	37.44	27.6%	10_8_50	0.6	48.74	32.39	33.5%
1_8_50	0.7	40.97	27.11	33.8%					
2_8_50	0.7	44.07	28.24	35.9%					
3_8_50	0.7	48.74	31.17	36.0%					
4_8_50	0.7	33.74	23.78	29.5%					
5_8_50	0.7	46.31	33.81	27.0%					
6_8_50	0.7	50.58	21.57	57.4%					
7_8_50	0.7	42.06	33.17	21.1%					
8_8_50	0.7	48.13	29.89	37.9%					
9_8_50	0.7	49.66	24.81	50.0%					
10_8_50	0.7	41.90	25.38	39.4%					