Efficient solution methods for patient assignment in Medical Day-care Units

by

Mohammadamin Vahedinia

A dissertation submitted to the University of Ottawa
in fulfillment the requirements for
the M.Sc. in Systems Science and Engineering degree

Faculty of Engineering
University of Ottawa

Supervisor by Dr. Onur Ozturk

Abstract

The increasing demand for outpatient cancer treatments has led to the widespread adoption of Medical Day-care Units (MDCUs), where efficient patient scheduling is essential for optimizing resource utilization and ensuring high-quality care. This thesis investigates the patient appointment scheduling problem in MDCUs, formulated as a resource-constrained parallel-machine multi-server job scheduling problem. The objective is to maximize clinic capacity (measured by the number of patients served) while maintaining operational feasibility and, in extended cases, minimizing patient waiting times.

A series of mixed-integer linear programming (MILP) models are developed to address different variants of the problem. The first, a base model, explicitly represents each patient's treatment and resource requirements. To improve scalability, a type-based model aggregates patients by treatment type, significantly reducing computational complexity while preserving scheduling accuracy. A genetic algorithm (GA) is then introduced as a metaheuristic approach for solving large-scale instances efficiently, complemented by a rolling horizon heuristic that dynamically updates solutions across overlapping planning periods. Finally, a deferral-penalized two-stage MILP is proposed for the extended problem: Stage 1 solves the type-based model to fix optimal throughput capacity, and Stage 2 minimizes total waiting time subject to a deferral-penalty term controlled by a tunable coefficient λ that charges accumulated wait for every unscheduled patient, trading throughput against distributional fairness in patient denials.

Simulation results on small, medium, and large problem instances (up to 1000 patients, 6 nurses, 5-day horizons) show that all four solution methods significantly outperform a FIFO baseline, as confirmed by one-sided Wilcoxon signed-rank tests at $\alpha = 0.05$. The type-based model consistently dominates the base MILP model in both solution quality and computational efficiency, and the genetic algorithm and rolling horizon heuristic yield near-optimal results for large-scale scenarios at substantially reduced runtimes. For the extended problem, a moderate penalty of $\lambda \approx 5$ achieves meaningful distributional fairness, halving the concentration of denials among long-waiting patients relative to pure throughput maximization, without imposing the full rigidity of FIFO ordering.

Overall, this research provides an integrated framework of exact and heuristic methods that enhance the efficiency, scalability, and fairness of patient scheduling in outpatient medical settings. Future work includes incorporating fatigue-aware nurse capacity constraints, stochastic programming extensions for patient no-shows and cancellations, and reinforcement learning or column-generation approaches for adaptive large-scale deployment.

Keywords: Patient Scheduling, Outpatient Appointment System, Health system optimization, Genetic Algorithm, Resource-Constrained Parallel Machine Multi Server Job Scheduling

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Cancer remains a leading cause of mortality and morbidity worldwide, with an increasing burden on healthcare systems. In Canada, a growing emphasis on outpatient care and the implementation of medical Day-care units have significantly impacted the cancer treatment process (Brenner et al. (2022)). It is also noteworthy that the healthcare industry takes up to 12.2 percent of the Canadian gross domestic product (cih). Given that both the stakes and expectations are high in this industry, allocating resources efficiently can enhance the quality of service and consumer satisfaction, which is why this field has come to the attention of researchers (Abdalkareem et al. (2021)).

Appointment Systems (AS) have been shown to be an essential aspect of healthcare management solutions in outpatient healthcare settings (Ahmadi-Javid et al. (2017); Gupta and Denton (2008)). In the simplest way, AS is the component that is responsible for the decision of assigning time slots and scheduling patients given the resource (beds, stations, nurses, infusion kits and etc.) constraints (Marynissen and Demeulemeester (2019)), and this decision can be made on different levels, and based on various factors/constraints (Ahmadi-Javid et al. (2017)).

The heart of the AS is an optimization model that is trying to make decisions based on various factors while optimizing the outcome of the system, i.e. the objective. The objective can vary in each context, depending on the definition and metrics used in measuring the *service quality*. The common metrics list includes but is not limited to patient waiting time, resources (nurses, beds, etc.) utilization, resources idle time, overtime, and the number of patients who received services (Berg et al. (2014)). It is trivial that some of the mentioned performance metrics can be transformed into or combined with others; for example, the nurse's idle time can be divided by the total work hours and let the result be α, $1 - \alpha$ represents the utilization of the nurses, or "busy ratio".

This thesis studies a patient appointment scheduling problem in a multi-campus Medical Day-care Unit (MDCU), where designated physicians can make patient referrals. These physicians also handle assigning patients to a specific campus. Once a patient is added to a campus waiting list, they remain there and cannot be transferred to another campus's waiting list. Each campus operates independently, with no shared resources or demand, meaning the scheduling patterns of one campus do not impact the others. Due to this fact, this research focuses on solving the scheduling problem for a single campus.

Each campus is equipped with a set number of nurses and beds, scheduled over a defined planning horizon, which are assumed to be available constantly throughout the planning period, that is, all beds and stations are available all the time, and a certain number of nurses are rostered daily. Precisely, the set of nurses each day might differ but they are assumed to be identical in skills and shift starts, so only the head count matters in the eyes of the AS.

The treatment administered on campus consists of infusion transplants. This process begins with a nurse preparing and attaching the infusion kits to the patient, followed by adjusting the kit's parameters according to the treatment protocol prescribed by the physician. Subsequently, the nurse primarily monitors the procedure to ensure it progresses safely. At the conclusion of the treatment session, the nurse returns to the station to remove the infusion kits, complete the necessary administrative documentation, and replace the linens. This sequence marks the completion of a single appointment.

The remainder of this thesis is structured as follows: initially, a review of the literature is conducted to outline the current trends in appointment systems management (ASM) and decision support tools, followed by a formal definition of the problem in Chapter 3. Subsequently, solution methods and models are developed under various problem assumptions, with the corresponding results and experimental methodology presented in the subsequent chapter. Finally, Chapter 6 provides a conclusive discussion and highlights potential directions for future research.

Chapter 2

Literature Review

Healthcare issues have recently garnered significant interest from researchers, particularly in applying Operations Research within the healthcare domain. Before proceeding with modeling and developing solutions for our problem, it is essential to conduct a literature review to identify existing gaps and gain a deeper understanding of prior research focused on addressing similar challenges. This chapter focuses on reviewing the previous research comprehensively, providing an understanding of the topic and the gaps to be addressed.

2.1 Appointment Systems and Outpatient Clinics

ASMs (Appointment Scheduling Models) have been explored extensively since the 1950s, with foundational work by Bailey (1952) and Lestdley (1952) using single queuing models to reduce patient wait times. Recently, there has been a surge in using advanced analytical techniques to address diverse scheduling challenges (Brailsford and Vissers (2011)). These techniques include queuing models, simulation, mathematical programming, and heuristics (Condotta and Shakhlevich (2014); Deceuninck et al. (2018)). A separate stream of research considers dynamic, system-level decision-making in ambulatory care using Markov decision processes and approximate dynamic programming to cope with uncertain arrivals and capacities (Gupta and Denton (2008); Moosavi et al. (2025)).

Outpatient clinics, particularly infusion clinics, face unique challenges due to varying appointment lengths, patient priorities, procedures, and requirements (Issabakhsh et al. (2020)). The literature offers multiple approaches to manage ASMs in outpatient settings. Some studies use template-based scheduling, which allocates specific time slots for various appointment types and assigns beds and nurses (Condotta and Shakhlevich (2014); Faridimehr et al. (2021); Hesaraki et al. (2019); Huang et al. (2019)). Patients are scheduled in separate decision phases. Alternatively, an open scheduling approach arranges patients on a pre-established waiting list over a planning period (Demir et al. (2020); Heshmat and Eltawil (2019)). Liang et al. (2015) present a matrix-based scheduling method that calculates the probability of assigning a patient to a specific time slot, considering factors such as priority, duration, and resource availability. In this broader context, Moosavi et al. (2025) examine a distributed ambulatory care system for hematology and oncology with multiple campuses and uncertain demand (in stochastic settings), using approximate dynamic programming to derive advance-scheduling policies that manage waiting times and resource usage at a high system-wide level.

2.2 Parallel Machine Common Server Job Scheduling

A similarity can be drawn between this field and the job scheduling field. One can easily see that this problem is similar to the *unrelated non-preemptive multi-server parallel machine job scheduling problem*, considering *loading* and *unloading* times. In this context, the servers are the nurses, the machines are the beds and the patients are the jobs.

The problem typically involves two identical parallel machines and a single server responsible for loading and unloading operations. The main objective is to minimize the makespan, which is the time taken to complete all scheduled jobs. The server also handles setup operations, which can be sequence-dependent, adding complexity to the scheduling problem (Elidrissi et al. (2023)).

The very first implication of this resemblance is that both problems belong to the *NP-HARD* problems class (Hsu and Liao (2020); Kravchenko and Werner (1997); Olteanu et al. (2022)). As a result, it is suggested to employ heuristics and meta-heuristic algorithms for similar problems. Olteanu et al. (2022) has investigated the performance of the Simulated Annealing algorithm for the "*Unrelated Parallel Machine Scheduling with Job and Machine Acceptance and Renewable Resource Allocation*" problem, and concluded that for large instances the simulated annealing can generate favorable results in a reasonable time. Magalhães-Mendes and de Almeida (2013) showed that *Genetic Algorithms (GA)* also are a promising method to solve large-scale problems in this class of problems in respect to the make-span ($C_{\max}$) objective.

Moreover, Lau and Tsang (2001) introduced a novel variant of the GAs hybridized with the *Guided Local Search* algorithm in the *Constraint Satisfaction Problem (CSP)* context, which Abu-Shams et al. (2022) evaluated its performance on the job scheduling context. The results were phenomenal, as the GGA (guided genetic algorithm) enhanced GA's performance tremendously by escaping local minimas.

Zhang and Wirth (2009) investigated multiple heuristics, including the *list scheduling (LS)*, and concluded that in some special problems (*regular equal setup time problem (RESP)*) the solution quality is at most 20 percent worse than the optimal solution. Jiang et al. (2015) argued that for the problem with both setup and unloading times, this metric is $\frac{12}{7}$ (meaning that the $C_{\max}$ calculated using this heuristic is not worse than $\frac{12}{7}$ of the optimal $C^*_{\max}$).

2.3 Problem Features

Moving back to the ASMs context, not every MDCU offer the same treatment or has the same considerations in their AS. Therefore, there exists various approaches in defining the problem scope at ASMs within the literature. In this section, the features studied by other researchers are described and investigated:

- **Appointment Type/Length:** Appointment type indicates that how the treatment is structured, and if their length are distinctive. Issabakhsh et al. (2020); Hesaraki et al. (2019) argued that due to the highly varying treatment plans (frequency, method of delivery and dosage of the drugs) tailored for each patient, it is safer to assume different appointment types in the AS modelling. Since this is the core feature of an ASM, this feature is considered in almost all papers reviewed in this study.

- **Daily Breaks:** At least one break a day in the middle of the shift is essential for each nurse in the real world setting. Three approaches were seen in the literature for dealing with this feature:

a) Nurses can take breaks whenever possible (outside the ASM scope) (Berg et al. (2014); Heshmat and Eltawil (2019); Hahn-Goldberg et al. (2014); Hur et al. (2020); Heshmat et al. (2018)).

b) Fixed time slots for the breaks so that no patients are scheduled at certain times at all (Faridimehr et al. (2021); Hesaraki et al. (2019)).

c) Flexible time slots for each nurse within a *break window* around specific times of day (Condotta and Shakhlevich (2014); Huang et al. (2019); Benzaid et al. (2019)).

- **Nurses' Skill sets:** Under some circumstances, each appointment type may require specific skills that only certain nurses have acquired. For example, Hur et al. (2020); Liang and Turkcan (2016) have included this feature in their studied problem. On contrary, most studies assumed that all nurses are well-trained to perform all of the treatments offered at their MDCU.

- **Nurses' Workload and Fatigueness:** From the resource management aspect of an AS, it is essential to manage the workload of each nurse to ensure their availability (Heshmat and Eltawil (2019)), workload distribution among the nurses (Liang et al. (2015)) or restricting the number of patients being handled at the same time (Condotta and Shakhlevich (2014); Hesaraki et al. (2019); Huang et al. (2019); Hahn-Goldberg et al. (2014); Liang and Turkcan (2016); Leeftink et al. (2017); Ramos et al. (2018)).

- **Overtime:** In some cases, overtime is allowed to wrap-up the shift and previously started or appointed treatments. Faridimehr et al. (2021) has incorporated overtime allowance as a constraint set to limit the maximum overtime of each physician. To the best of our knowledge, this feature has been rarely used in the literature. Besides being among the problem features, it can be also treated as an objective of the problem.

2.4 Performance Metrics

While developing a decision support tool, as in the ASMs, choosing how to quantify the objectives plays a vital role. Ahmadi-Javid et al. (2017) has listed some of these performance metrics in their review:

- **Number of patients:** Maximizing the number of patients that has been scheduled over the planning period is considered one of the most popular performance metrics among the researchers (Huang et al. (2019); Benzaid et al. (2019); Huang et al. (2021)). In a novel approach, Heshmat et al. (2018) tried to maximize the patients' total treatment time, that is, the total appointment lengths of the scheduled patients has been used as a proxy of the number of patients.

- **Wait-times:** Another popular performance indicator is the patients' waiting time. This duration can be partitioned into indirect and direct times, i.e., the time between patient's request for appointment and the planned appointment, and the time between patient's arrival at the clinic and the start of their treatment , respectively (Ahmadi-Javid et al. (2017)). Many studies have considered minimizing the direct or indirect portion of the wait-time as their objective function (Condotta and Shakhlevich (2014); Issabakhsh et al. (2020); Faridimehr et al. (2021); Hesaraki et al. (2019); Heshmat and Eltawil (2019); Agnetis et al. (2019)).

- **Overtime:** Demir et al. (2020) have used nurses' over-time as one of their performance metrics to be minimized in the model.

- **Resource Utilization:** Optimal usage of the available resources such as infusion chairs and beds can impact patient care quality, operational efficiency, and financial performance. Alireza et al. (2021) developed a simulation model that captures the complexities of daily operations at the Mayo Clinic Cancer Center, including detailed resource utilization metrics. This approach allows for a comprehensive analysis of how different appointment scheduling strategies impact resource utilization, patient length of stay, and the possibility of same-day treatment completions. Furthermore, this metric has been employed in the objective function in (Demir et al. (2020); Liang et al. (2015)).

- **Nurse Idleness:** Reducing nurse idleness is crucial for optimizing resource allocation and improving overall clinic efficiency. By minimizing periods of inactivity, clinics can potentially increase patient throughput, reduce waiting times, and improve the quality of care. This metric can be considered as a subset of the previously mentioned metric, if nurses and physicians are looked after as a resource in the oncology clinics.

- **Profit:** The incorporation of profit considerations in mathematical models for oncology infusion clinic appointment systems is an emerging area of research. For example, Berg et al. (2014) aimed at maximizing the expected profits using a previously developed two-stage optimization model, Hahn-Goldberg et al. (2014) aim to balance the potential revenue from overbooking against the costs associated with patient waiting times and potential overtime, and Hadid et al. (2022) presented an integral stochastic discrete simulation-based multi-objective optimization model incorporating profits into the objective.

Also, the aforementioned objectives can be combined and for a multi-objective problem. For instance, Klassen and Yoogalingam (2013) has investigated minimizing waiting-times and maximizing physician utilization (minimizing nurse/physician idleness) in their paper. There exists a large body of literature that incorporating two or more of those metrics in their objective function, mostly by adopting a weighted sum approach (Berg et al. (2014); Demir et al. (2020); Liang et al. (2015); Hur et al. (2020); Leeftink et al. (2017)).

2.5 Research Gap

The literature reviewed in this chapter reveals two complementary streams of work, each relevant to the problem studied in this thesis yet each leaving a distinct gap unaddressed.

The first gap concerns the treatment structure. The parallel machine scheduling literature models server involvement as a binary state: the common server is either actively loading or unloading a job, or it is free. Outpatient scheduling models similarly treat appointments as atomic tasks assigned to nurses. Neither stream captures the three-phase treatment process observed in infusion clinics, in which the nurse is exclusively occupied during initialization, passively monitors during the infusion itself, and returns for an active finalization. The passive-monitoring phase is structurally distinctive: it frees the nurse to concurrently serve other patients, a property that has no direct counterpart in classical parallel machine formulations.

The second gap concerns the joint treatment of capacity and wait-time. A broad body of work optimizes one or the other: studies such as Condotta and Shakhlevich (2014); Issabakhsh et al. (2020); Faridimehr et al. (2021); Hesaraki et al. (2019); Agnetis et al. (2019) minimize direct or indirect wait-times, while others such as Huang et al. (2019); Benzaid et al. (2019); Huang et al. (2021) maximize patient throughput. What has not been proposed is a scheduling framework that treats capacity maximization and wait-time minimization as simultaneous, first-class objectives within a single optimization model over a multi-day planning horizon.

This thesis addresses both gaps by developing an integrated appointment scheduling framework that incorporates the three-phase treatment model and jointly optimizes clinic capacity and patient wait-times in a unified formulation.

Chapter 3

Problem Description

This chapter formally introduces the scheduling problem studied in this thesis and describes its two variants in detail. The *base problem* concerns maximizing the number of patients treated over a multi-day planning horizon. The *wait-time extension* augments this with a secondary objective of minimizing the total time patients wait between requesting and receiving an appointment. Both variants share the same clinical setting and resource structure, which are presented first.

In the outpatient MDCU in study, there are multiple nurses, who are responsible for giving prescribed treatments to the patients, and monitoring them while receiving treatment. Each treatment in this clinic has three different phases, represented with discretized time units (slots) of 5 minutes:

a) ***Initialization***: During this period, the nurse is preparing the patient in order to administer the IV treatment; this includes explaining the symptoms the patient may feel, planting infusion equipment in the patient's veins, and setting the transfusion speed. During this phase, the nurse is exclusively occupied with the patient and cannot perform any other tasks concurrently.

b) ***Monitoring***: After the infusion treatment has started, the nurse can leave the station and monitor the treatment process passively. Due to safety concerns, there is theoretically a limit on the number of simultaneous patients that a single nurse can monitor at a time. However, in practice this constraint is not binding: the number of available beds is the effective capacity bottleneck, and any available nurse can take over monitoring responsibility if needed. Therefore, this limit is considered out of scope and is not enforced in the model. It is obvious that the nurse is passively busy with the patient, and is available for other tasks.

c) ***Finalization***: Finally, the nurse comes back to the station in the *finalization* period and prepares the patient to leave. Removing the transfusion set, changing linen, and filling in administrative forms (e.g. documenting the treatment) fall into this period. While completing this period, the nurse is actively involved, and cannot perform any other active tasks, i.e. initialize or finalize other patients.

Additionally, each treatment takes place in a station, i.e. the beds. At this stage of decision-making, it is assumed that the beds are available for all the nurses, that is, all $\bar{n}$ nurses can start treatment for any patient on any of the available stations in the clinic.

This problem bears a close resemblance to the *unrelated non-preemptive multi-server parallel machine job scheduling problem* with *loading* and *unloading* times Elidrissi et al. (2023). In this analogy, the nurses act as the common servers, the beds act as the parallel machines, and the

patients correspond to the jobs. The initialization phase maps to the loading operation (active, exclusive nurse involvement), and the finalization phase maps to the unloading operation (also active and exclusive). The monitoring phase, however, has no direct counterpart in the classical formulation: unlike loading and unloading, it does not require the nurse's active attention, freeing the nurse to serve other patients concurrently. This passive-monitoring property is a structurally distinctive feature of the problem and is what motivates the three-phase treatment model described above. As discussed in Chapter 2, this class of problems is known to be *NP-hard*, which motivates the use of heuristic and meta-heuristic solution approaches.

As of the objective of this study, we chose to maximize the capacity of the clinic, that is, the number of patients that receive their treatments during the planning horizon.

Both problem variants span multiple days, i.e., the planning horizon T covers several consecutive working days. Time is discretized into 5-minute slots across the entire horizon, and overtime is not permitted on any day.

Moreover, in a slightly different problem studied in this thesis, the complexity of handling wait-times has been added. Throughout this thesis, $t = 0$ denotes the start of the planning horizon. In the base problem, all patients are assumed to be available at the start of the planning horizon, i.e., $r_p = 0$ for all patients p. In the wait-time variant, on the contrary, each patient has requested an appointment prior to the start of the planning horizon, so $r_p < 0$; for example, $r_p = -5$ means the patient submitted their request 5 days slots before planning began. If patient p is scheduled to start treatment at day sc_p, their wait time is $sc_p - r_p$.

For the "wait-time" variation of the problem, the problem becomes multi-objective. In these cases, first the maximum number of patients that can be scheduled (p^*) is calculated, then among all the solutions scheduling p^* number of patients, the one with minimal sum of the wait-times is selected.

To summarize, both problem variants are defined over the following inputs:

- **Patients**: either grouped by treatment type or treated as individuals, each patient is characterized by three treatment durations: initialization time, monitoring time, and finalization time. In all instances it is assumed that initialization and finalization each require only one time slot.
- **Planning horizon and time slots**: time is discretized into 5-minute slots spanning multiple working days. Overtime is not permitted, so the number of available slots per day is fixed.
- **Nurses**: a set of nurses responsible for initializing and finalizing treatments. The number of nurses ($\bar{n}$) varies across problem instances.
- **Beds/Stations**: a fixed number of stations that limit the number of patients being treated simultaneously.
- **Release times**: in the base problem, all patients are available at the start of the planning horizon ($r_p = 0$). In the wait-time variant, each patient has a release time $r_p < 0$, indicating they requested an appointment before the planning horizon began.

The objectives differ between the two variants:

- **Base problem — primary objective**: maximize the number of patients who receive treatment within the planning horizon.
- **Wait-time variant — primary objective**: same as above; maximize the number of scheduled patients (p^*).

- **Wait-time variant — secondary objective**: among all schedules achieving p^* patients, minimize the total wait time $\sum_{p \in \mathcal{P}^*}(sc_p - r_p)$, where $\mathcal{P}^*$ is the set of scheduled patients and sc_p is the assigned appointment start time.

Formal problem statement. Given a set of patients $\mathcal{P}$, a set of nurses $\mathcal{N}$, a set of stations $\mathcal{S}$, and a multi-day planning horizon T (discretized into 5-minute slots), find an assignment of a subset $\mathcal{P}^* \subseteq \mathcal{P}$ of patients to nurses, stations, and start times such that:

- each nurse performs at most one active task (initialization or finalization) at any time slot,
- each station is occupied by at most one patient at any time slot,
- no treatment extends beyond the planning horizon,
- (wait-time variant) each patient is scheduled no earlier than their release time r_p,

maximizing $|\mathcal{P}^*|$, and — in the wait-time variant — secondarily minimizing $\sum_{p \in \mathcal{P}^*}(sc_p - r_p)$.

Chapter 4

Models and Solution Methods

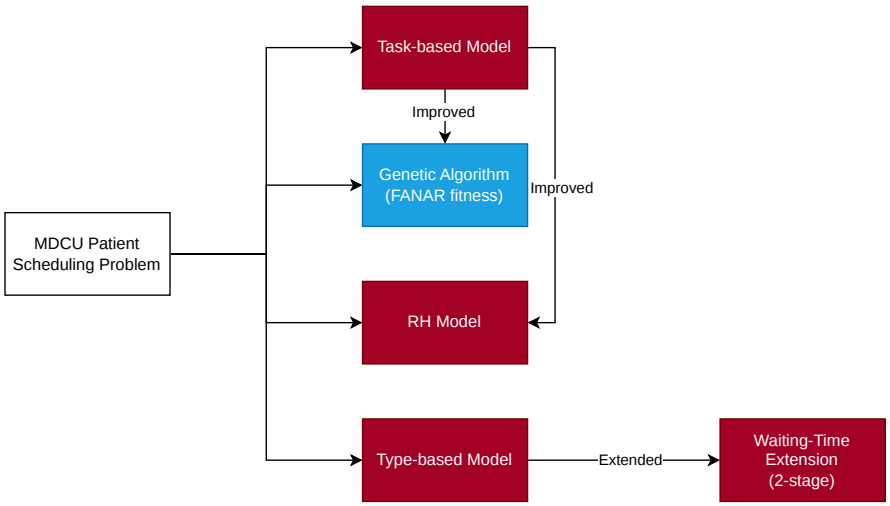

Figure 4.1: Solution methodology roadmap for the MDCU Patient Scheduling Problem.

Figure 4.1 illustrates the progression of solution methods developed in this chapter. Starting from the *Task-based (Base) Model*, an exact MILP formulation, a *Genetic Algorithm* with FANAR fitness is developed as a metaheuristic improvement, enabling scalable solutions for large instances; this GA also seeds the *Rolling Horizon (RH) heuristic*, which extends the base model to dynamic multi-period settings rather than introducing a new formulation, and is covered in Chapter 5. In parallel, the *Type-based Model* reformulates the base model by aggregating patients into treatment types, achieving significant computational gains; it is then extended into the *Waiting-Time Extension* (a two-stage model) to explicitly incorporate patient waiting time objectives. For each method, implementation details, benefits, and limitations are discussed.

4.1 Base Mathematical Model

As discussed in the previous chapter, the problem in this study is isomorphic to the common server parallel-machine non-preemptive scheduling with setup and unloading times or $S, P|s_j = 1, u_j = 1|N$ as per Graham's notation (Graham et al. (1977)). Hence, the base model for this problem is inspired by that problem, with slight modifications to fit our use case. Remainder of this section is dedicated to describe the mathematical model and implementation remarks.

4.1.1 Mathematical Model

First, a detailed description of the sets, parameters, and decision variables used in the base model is presented, as summarized in the Tables 4.1 and 4.2. These elements form the foundational components of the model, defining the variables and constraints that guide its operation.

Table 4.1: Sets and Parameters of the "Base Model"

Name	Definition	Value(s)
	Sets	
$n \in \mathcal{N}$	$\mathcal{N}$ is the set of all nurses	$\mathcal{N} = \{1, 2, \ldots, \bar{n}\}$
$d \in \mathcal{D}$	$\mathcal{D}$ is the set of days in our scheduling horizon	$\mathcal{D} = \{1, 2, \ldots, d\}$
$t \in \mathcal{T}$	$\mathcal{T}$ is the set of time slots in a day	$\mathcal{T} = \{1, 2, \ldots, \bar{t}\}$
$p \in \mathcal{P}$	$\mathcal{P}$ is the set of individual patients	$\mathcal{P} = \{1, 2, \ldots, \bar{p}\}$
	Parameters	
R	Number of beds	13
α	Maximum patients that a nurse can actively handle simultaneously	1
β	Maximum patients that a nurse can passively monitor simultaneously	∞
π_{p1}	Duration of the *initialization* period of patient p in time slots unit	1
π_{p2}	Duration of the *monitoring* period of patient p in time slots unit	$6 \leq \pi_{p2} \leq 18$
π_{p3}	Duration of the *finalization* period of patient p in time slots unit	1

Table 4.2: Decision Variables of the Problem

Variable	Definition	Domain
x_{ndtp}	If nurse n is starting the treatment of patient p at time slot t of day d	$x_{ndtp} \in \{0, 1\}$
r_{1ndt}	Number of the *initialization* tasks that nurse n is actively busy with at time slot t of day d	$\rho_{1ndt} \in \mathcal{N}$
r_{2ndt}	Number of the *monitoring* tasks that nurse n is passively busy with at time slot t of day d	$\rho_{2ndt} \in \mathcal{N}$
r_{3ndt}	Number of the *finalization* tasks that nurse n is actively busy with at time slot t of day d	$\rho_{3ndt} \in \mathcal{N}$

Then objective function (Equation (4.1)) is established as maximizing the number of patients scheduled in the planning horizon, aiming to optimize resource utilization and improve overall scheduling efficiency as discussed previously in Chapter 3.

$$\max Z = \sum_{n \in \mathcal{N}, d \in \mathcal{D}, t \in \mathcal{T}, p \in \mathcal{P}} x_{ndtp} \tag{4.1}$$

The model must ensure that it does not schedule more patients than once; therefore, the Equation (4.2) constraint is incorporated into the model.

$$\sum_{n \in \mathcal{N}, d \in \mathcal{D}, t \in \mathcal{T}} x_{ndtp} \leq 1 \qquad \forall p \in \mathcal{P} \tag{4.2}$$

As per the formal problem statement, each treatment type is divided into three phases: Initialization, Monitoring and Finalization. To take into account the nurses' capacity for each phase,

first we have to count number of patients assigned to the nurse for each time period, grouped by which phase the patient is in during that time slot. Equations (4.3) to (4.5) introduce variables for the aforementioned phases, respectively.

$$\rho_{1ndt} = \sum_{p\in\mathcal{P}} \sum_{t-(\pi_{p1})+1\le t'\le t} x_{ndt'p} \qquad \forall n \in \mathcal{N}, d \in \mathcal{D}, t \in \mathcal{T} \tag{4.3}$$

$$\rho_{2ndt} = \sum_{p\in\mathcal{P}} \sum_{t-(\pi_{p1}+\pi_{p2})+1\le t'\le t-\pi_{p1}} x_{ndt'p} \qquad \forall n \in \mathcal{N}, d \in \mathcal{D}, t \in \mathcal{T} \tag{4.4}$$

$$\rho_{3ndt} = \sum_{p\in\mathcal{P}} \sum_{t-(\pi_{p1}+\pi_{p2}+\pi_{p3})+1\le t'\le t-(\pi_{p1}+\pi_{p2})} x_{ndt'p} \qquad \forall n \in \mathcal{N}, d \in \mathcal{D}, t \in \mathcal{T} \tag{4.5}$$

It is notable that for calculating $\rho_{1ndt}(\forall n \in \mathcal{N}, d \in \mathcal{D}, t \in \mathcal{T})$, the number of patients (p) that their initialization period spans over time slot t are counted. Formally, this includes all the treatments that has been started no later than t and no earlier than $\pi_{p1}-1$ time slots than t. Combining these two criterion, leaves us with interval $[t-\pi_{p1}+1, t]$, inclusive. The same methodology is applied to derive intervals $[t-(\pi_{p1}+\pi_{p2})+1, t-\pi_{p1}]$ and $[t-(\pi_{p1}+\pi_{p2}+\pi_{p3})+1, t-(\pi_{p1}+\pi_{p2})]$ for calculating ρ_{2ndt} and ρ_{3ndt}, respectively.

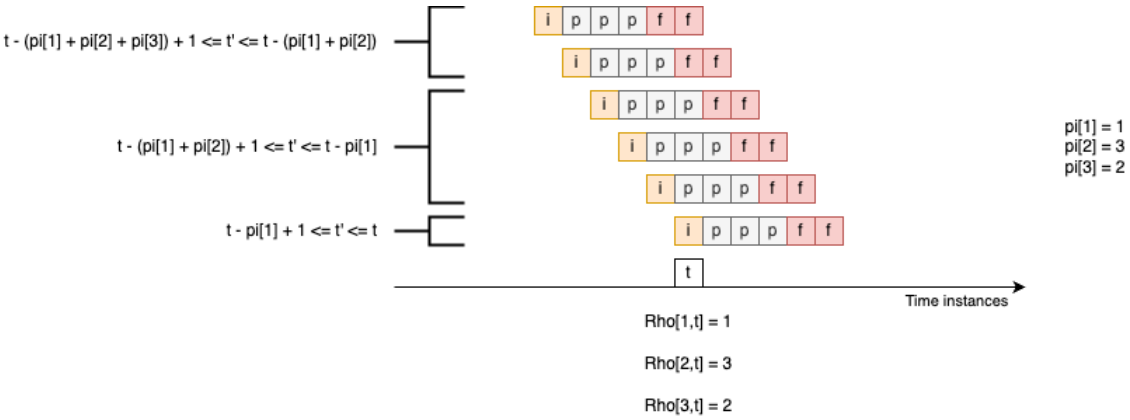

Figure 4.2: Visualization of how constraints Equations (4.3) to (4.5) are defined. Note that this Gannet chart is not necessary a feasible solution.

Using these auxiliary variables, the problem constraints can be defined. Equation (4.6) limits the number of tasks that require the nurse's active attention to $\alpha = 1$, and Equation (4.7) does the same for passive monitoring constraint (β). Finally, Equation (4.8) take into account the number of beds (R) constraint in the problem.

$$\rho_{1ndt} + \rho_{3ndt} \le \alpha \qquad \forall n \in \mathcal{N}, d \in \mathcal{D}, t \in \mathcal{T} \tag{4.6}$$

$$\sum_{n\in\mathcal{N}} \rho_{2ndt} \le \beta \qquad \forall d \in \mathcal{D}, t \in \mathcal{T} \tag{4.7}$$

$$\sum_{n\in\mathcal{N}} (\rho_{1ndt} + \rho_{2ndt} + \rho_{3ndt}) \le R \qquad \forall n \in \mathcal{N}, d \in \mathcal{D}, t \in \mathcal{T} \tag{4.8}$$

Finally, since overtime is not allowed for the nurses, no patient shall be admitted to the clinic if it is known that their treatment cannot finished within the current shift. Equation (4.10) is ensuring that no treatment is initialized for any patient type that we know they does not fit into that day. In other words, the only condition that a patient is not fitting into the current day's schedule is

that being started late and the completion time of that treatment ($c_p = t + \pi_{p1} + \pi_{p2} + \pi_{p3} - 1$) being later than $\bar{t}$ ($c_p \geq \bar{t}$). This implies that

$$t + \pi_{p1} + \pi_{p2} + \pi_{p3} - 1 \geq \bar{t} \Rightarrow t \geq \bar{t} - (\pi_{p1} + \pi_{p2} + \pi_{p3}) + 1 \tag{4.9}$$

$$x_{ndtp} = 0 \qquad \forall n \in \mathcal{N}, d \in \mathcal{D}, p \in \mathcal{P}, t \in \{\bar{t} - (\pi_{p1} + \pi_{p2} + \pi_{p3}) + 1, \ldots, \bar{t}\} \tag{4.10}$$

4.1.2 Implementation Details

To implement this model, there exists various methods, among which, the combination of `PyOmo` Python library and `IBM CPLEX Mathematical Solver` (*CPLEX*) has been chosen. Having the mathematical model developed, it has to be reduced in size for better results, both time-wise and quality-wise, with `CPLEX`.

The very first technique to achieve this goal is to remove redundant variables, that is, the auxiliary variables. In other words, those variables have been defined for the sake of simplicity and intuitiveness of the model, and can be omitted by substituting those in Equations (4.6) to (4.8) with their definition in Equations (4.3) to (4.5). By this optimization in the model, $3 \times |\mathcal{N}| \times |\mathcal{D}| \times |\mathcal{T}|$ variables are removed from the model, as well as the constraint set Equations (4.3) to (4.5), which is a significant reduction in model size given that in practice regarding the number of variables.

Secondly, since β is set to ∞, i.e. it is assumed that a nurse does not have any limits on the number of the patients that they can monitor passively, Equation (4.7) constraint set is rendered redundant as well; therefore, it can be removed from the constraints set.

Finally, the last constraint set (Equation (4.10)) can be compressed into a single constraint by substituting that equations set with the aggregated sum of that set:

$$\sum_{n \in \mathcal{N}} \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}} \sum_{t \in \{\bar{t} - (\pi_{p1} + \pi_{p2} + \pi_{p3}) + 1, \ldots, \bar{t}\}} x_{ndtp} = 0 \tag{4.11}$$

To wrap up all the enhancements made, here is how the implemented model looks like:

$$\max Z = \sum_{n\in\mathcal{N},d\in\mathcal{D},t\in\mathcal{T},p\in\mathcal{P}} x_{ndtp} \tag{4.12}$$

$s.t.:$

$$\sum_{n\in\mathcal{N},d\in\mathcal{D},t\in\mathcal{T}} x_{ndtp} \leq 1 \qquad \forall p\in\mathcal{P} \tag{4.13}$$

$$\sum_{p\in\mathcal{P}} \sum_{t-(\pi_{p1})+1\leq t'\leq t} x_{ndt'p} + \sum_{p\in\mathcal{P}} \sum_{t-(\pi_{p1}+\pi_{p2}+\pi_{p3})+1\leq t'\leq t-(\pi_{p1}+\pi_{p2})} x_{ndt'p} \leq \alpha \qquad \forall n\in\mathcal{N}, d\in\mathcal{D}, t\in\mathcal{T} \tag{4.14}$$

$$\sum_{n\in\mathcal{N}}\sum_{p\in\mathcal{P}} \sum_{t-(\pi_{p1})+1\leq t'\leq t} x_{ndt'p} + \sum_{n\in\mathcal{N}}\sum_{p\in\mathcal{P}} \sum_{t-(\pi_{p1}+\pi_{p2})+1\leq t'\leq t-\pi_{p1}} x_{ndt'p} + \sum_{n\in\mathcal{N}}\sum_{p\in\mathcal{P}} \sum_{t-(\pi_{p1}+\pi_{p2}+\pi_{p3})+1\leq t'\leq t-(\pi_{p1}+\pi_{p2})} x_{ndt'p} \leq R \qquad \forall d\in\mathcal{D}, t\in\mathcal{T} \tag{4.15}$$

$$\sum_{n\in\mathcal{N}}\sum_{d\in\mathcal{D}}\sum_{p\in\mathcal{P}} \sum_{t\in\{\bar{t}-(\pi_{p1}+\pi_{p2}+\pi_{p3})+1,\ldots,\bar{t}\}} x_{ndtp} = 0 \tag{4.16}$$

$$x_{ndtp}\in\{0,1\} \qquad \forall n\in\mathcal{N}, d\in\mathcal{D}, t\in\mathcal{T}, p\in\mathcal{P} \tag{4.17}$$

4.2 Genetic Algorithm

Genetic Algorithms are a powerful tool for large-scale optimization problems, including this paper's problem. This algorithm mimics the natural selection evolutionary process; thus, it has five main components:

a) **Initialization**: A population of candidate solutions is generated, either randomly or using a problem-specific method. Each solution, known as a chromosome, is represented by a permutation notation that has been used for the solution description; the index of a patient in the permutation indicates the priority of the patient.

b) **Fitness Evaluation**: The quality of each solution is evaluated using a fitness function. This function assigns a score to each individual based on how well it solves the problem, that is, the objective function. There are two ways to interpret the chromosome's notation when calculating the objective function, which is explained in detail in Sections 4.2.1 and 4.2.2.

c) **Parent Selection**: This step involves selecting individuals from the current population to serve as parents for creating the next generation. In fact, this paper decided to go with the tournament selection as it provides the algorithm with both exploratory and exploitatory features. This method first creates a tournament of fixed size with randomly selected chromosomes as candidates, then selects the two fittest chromosomes in the tournament as the parents. Algorithm 1 describes the algorithm used for this components.

d) **Genetic Operators**: Genetic operators are used to create new offspring from the selected parents, or modify the current population. The two most common operators are crossover

and mutation:

- **Crossover**: This operator combines two parents, preserving some of the features from each of the parents. One of the most common crossover operators for permutation notation is order crossover (OX1) (Davis (1990)). This operator selects a segment of one of the parents and copied the segment to the child. Afterward, moves the remaining permutation elements from the other parent to the child, while preserving their order in the second parent. Algorithm 2 describes the algorithm used for this components.
- **Mutation**: This operator is inspired by the mutations in the nature. By applying this operator, the population can have random perturbations that lead to more exploration of the solution space. In this problem, swap mutation operator has been employed. That is, two randomly positions in the chromosome are being swapped with a predefined probability. Algorithm 3 describes the algorithm used for this components.

e) **Preservation**: The offspring produced through crossover and mutation replace some or all of the individuals in the current population, forming a new generation. Elitist survival, where some of the fittest individuals are always preserved, has been chosen as the replacement strategy due to its exploitation features. Algorithm 4 describes the algorithm used for this components.

Algorithm 1 Tournament Parent Selection Pseudo Code

```
1: procedure TournamentParentSelection(population)
2:     global tournamentSize
3:     tournament ← SelectRandomChromosomes(population, tournamentSize)
4:     return TwoMostFittest(tournament)
```

Algorithm 2 Order Crossover Pseudo Code

```
1: procedure OrderCrossover(P_0, P_1)
2:     a, b ← RandomSegmentOf(P_0)
3:     child ← empty list of size |P_0|
4:     Copy segment [a, b] from P_0 to child, preserving the positions
5:     Copy the rest of the permutation from P_1 to child while preserving the order from P_1
6:     return child
```

Algorithm 3 Swap Mutation Pseudo Code

```
1: procedure SwapMutation(chromosome)
2:     a, b ← RandomPositionsOf(chromosome)
3:     swap chromosome_a and chromosome_b in place
4:     return chromosome
```

In summary, Algorithm 5 combines all aforementioned components simulating the evolutionary process of finding the fittest solutions for our patient scheduling problem. Moreover, at each stage when a chromosome is created and modified, its fitness value have to be (re-)calculated. The fitness calculation methods are described in the following sections.

4.2.1 Sequence-partitioning into sub-sequences for each nurse (SPISS)

One way to divide the workload among nurses when the sequence of jobs are given is to partition each sequence into continuous sub-sequences, and assign each sub-sequence to one nurse. To accomplish this, we can first calculate the maximum number of patients that can be scheduled

Algorithm 4 Elitist Survival Strategy Pseudo Code

```
1: procedure ElitistSurvival(population, offsprings)
2:     global elitistPopulation, generationPopulation
3:     newPopulation ← GetTopFittest(population, elitistPopulation)
4:     newPopulation ← newPopulation ∪ GetTopFittest(offsprings, generationPopulation − elitistPopulation)
5:     return newPopulation
```

Algorithm 5 Genetic Algorithm Pseudo Code

```
1: procedure GeneticAlgorithm
2:     global generationPopulation, numGenerations, numCrossovers, μ_m
3:     population ← GenerateRandomInitialPopulation(generationPopulation)
4:     for generation ← 1 to numGenerations do
5:         offsprings ← []
6:         for c ← 1 to numCrossovers do
7:             P_0, P_1 ← TournamentParentSelection(population)
8:             offsprings ← append(offsprings, OrderCrossover(P_0, P_1))
9:             offsprings ← append(offsprings, OrderCrossover(P_1, P_0))
10:        for chromosome ∈ [population ∪ offsprings] do
11:            if randomeNumber() ≤ μ_m then
12:                chromosome ← SwapMutation(chromosome)
13:        population ← ElitistSurvival(population, offsprings)
14:    return Fittest(population)
```

using only one nurse for each sub-sequence greedily, then partition the sequence using dynamic programming with goal of maximizing the sum of the number of the patients scheduled for each nurse. In other words, in the first stage we can calculate the objective function for a smaller sub-problem (scheduling a sub-sequence for one nurse) by simply iterating through the sub-sequence, finding the first available time that the patient fits there and stop if we cannot schedule the patient anymore. Let tbl_{ij} be the approximated solutions for the sub-problem on sub-sequence $[a, b)$, then this DP rule can be used for the second stage:

$$dp_{i,0} = 0 \qquad \forall i \leq |chromosomoe| \tag{4.18}$$

$$dp_{i,n} = \max_{1 \leq k \leq i} \{tbl_{ki} + dp_{k,n-1}\} \tag{4.19}$$

Algorithm 6 Sequence Partitioning Fitness Function Pseudo Code

```
1: procedure SequencePartitioningFF(Π)
2:     initialize empty table tbl_{|Π|×|Π|}
3:     for i ← 1 up to |Π| do
4:         for j ← i up to |Π| + 1 do
5:             tbl_ij ← CalculatePartialFF(Π, [i, j))
6:     initialize dp_{|P|+1,n̄} table
7:     for n ← 1 up to n̄ do
8:         for i ← 1 up to |Π| + 1 do
9:             dp_{i,n} ← max_{1≤k≤i} {tbl_ki + dp_{k,n−1}}
10:    return dp_{|Π|+1,n̄}
```

It is trivial that the time complexity of the Algorithm 6 is $\mathcal{O}(\max\{\bar{p}^2(\bar{p}+\bar{dt}), \bar{p}^2\bar{n}\}) \approx \mathcal{O}(\bar{p}^3\bar{dt})$. After scrutinizing the calculation process of tbl, we understood that each row of that table (all the sub-sequences starting at position i of the permutation Π) can be calculated at once; cutting the

time complexity of calculating fitness function for each chromosome by $\bar{p}$, i.e. reducing the time complexity down to $\mathcal{O}(\bar{p}^2\bar{dt})$.

4.2.2 First-available-nurse assigning rule (FANAR)

Another way to evaluate the fitness of a sequence is to use the list scheduling (LS) heuristics for m machines (Jiang et al. (2015)). In detail, we iterate through Π and assign each patient to the first available nurse time-wise. It can be seen that the time-complexity of calculating the fitness function this way is $\mathcal{O}(\bar{p}\max\{\bar{dt}, \bar{n}\})$ which is by far better than the previous heuristic used. The psuedo-code of this method is demonstrated briefly in.

Algorithm 7 First Available Nurse Fitness Function Pseudo Code

```
1: procedure FANFF(Π)
2:     for i ← 1 up to |Π| do
3:         availabilities ← GetFirstAvailableTimeForAllNurses(Π_i)  ▷ Finding first time that fits the patient Π_i without compromising the resource constraints
4:         if cannot schedule on none of the nurses then return i − 1  ▷ Only scheduled i − 1 patients between the nurses
5:         n ← GetBestNurse(availabilities)
6:         Assign Π_i to nurse n and update the nurses scheduling table.
7:     return |Π|  ▷ Successfully scheduled all the patients
```

4.3 Grouping Patients ("Type-based")

In the continuous pursuit of optimizing patient assignment and appointment scheduling within medical Day-care units, the need for efficient and scalable models becomes increasingly apparent. Traditional approaches often treat each patient as a unique entity, considering individual parameters, i.e. processing times, in the modeling process. While this method provides a tailored approach to patient care, it may not always be the most efficient, especially in environments characterized by high patient volumes and limited resources. This chapter introduces the "type-based" model, a novel approach designed to address these challenges by grouping patients according to their treatment types.

Previous models in this domain have emphasized individualized patient data, resulting in larger and often more computationally demanding models. The intrinsic similarity in treatment requirements and parameters among patients receiving the same type of medical care presents an opportunity for optimization. By acknowledging and utilizing these similarities, the type-based model proposes a more computationally efficient framework.

This model leverages the concept that patients undergoing similar treatments can be aggregated into distinct groups or types. This aggregation reduces the complexity of the model without compromising the accuracy and quality of patient care. The primary motivation behind this approach is to improve model performance, reduce computational load, and provide a scalable solution that can be adapted to various operational scales within medical Day-care units.

The following subsections of this section will delve into the mathematical formulation of the type-based model, detailing the assumptions, variables, and constraints that define this approach. Subsequent discussion will cover the implementation of the model using IBM CPLEX Solver.

4.3.1 Mathematical Model

Tables 4.3 and 4.4 describe the sets, parameters and decision variables used in our model.

Table 4.3: Sets and Parameters of the "Type-Based Model"

Name	Definition	Value(s)
	Sets	
$n \in \mathcal{N}$	$\mathcal{N}$ is the set of all nurses	$\mathcal{N} = \{1, 2, \ldots, \bar{n}\}$
$d \in \mathcal{D}$	$\mathcal{D}$ is the set of days in our scheduling horizon	$\mathcal{D} = \{1, 2, \ldots, d\}$
$t \in \mathcal{T}$	$\mathcal{T}$ is the set of time slots in a day	$\mathcal{T} = \{1, 2, \ldots, \bar{t}\}$
$j \in \mathcal{J}$	$\mathcal{J}$ is the set of patient types	$\mathcal{J} = \{1, 2, \ldots, \bar{j}\}$
$p \in \mathcal{P}$	$\mathcal{P}$ is the set of individual patients	$\mathcal{P} = \{1, 2, \ldots, \bar{p}\}$
	Parameters	
R	Number of beds	13
ν_j	Number of patients of type j available at time $r = 0$	$0 \leq \nu_j \leq \lvert\mathcal{P}\rvert$ and $\sum_{j \in \mathcal{J}} \nu_j = \bar{p}$
α	Maximum patients that a nurse can actively handle simultaneously	1
β	Maximum patients that a nurse can passively monitor simultaneously	∞
π_{j1}	Duration of the *initialization* period of patient j in time slots unit	1
π_{j2}	Duration of the *monitoring* period of patient j time slots unit	$6 \leq \pi_{j2} \leq 18$
π_{j3}	Duration of the *finalization* period of patient j time slots unit	1

Table 4.4: Decision Variables of the Problem

Variable	Definition	Domain
x_{ndtj}	If nurse n is starting the treatment of a patient of type j at time slot t of day d	$x_{ndtj} \in \{0, 1\}$
r_{1ndt}	Number of the *initialization* tasks that nurse n is actively busy with at time slot t of day d	$\rho_{1ndt} \in \mathcal{N}$
r_{2ndt}	Number of the *monitoring* tasks that nurse n is passively busy with at time slot t of day d	$\rho_{2ndt} \in \mathcal{N}$
r_{3ndt}	Number of the *finalization* tasks that nurse n is actively busy with at time slot t of day d	$\rho_{3ndt} \in \mathcal{N}$

Like previous models discussed in this thesis, this model's goal is to optimize the number of patients appointed, i.e. planned capacity, of the Medical Day-care Unit. Hence, Equation (4.20) defines the objective function.

$$\max Z = \sum_{n \in \mathcal{N}, d \in \mathcal{D}, t \in \mathcal{T}, j \in \mathcal{J}} x_{ndtj} \tag{4.20}$$

First, the model should guaranty that it will not schedule more than the existing patients available; hence, the Equation (4.21) constraint is introduced to the model.

$$\sum_{n \in \mathcal{N}, d \in \mathcal{D}, t \in \mathcal{T}} x_{ndtj} \leq \nu_j \qquad \forall j \in \mathcal{J} \tag{4.21}$$

Similarly to the Base model (Section 4.1), each treatment type is divided into three distinct phases: Initialization, Monitoring, and Finalization. To accurately account for the nurses' capacity during each of these phases, we first need to calculate the number of patients assigned to each nurse in a given time period, categorized by the phase the patient is in during that specific time slot.

The variables representing these phases are introduced in Equations (4.22) to (4.24).

Furthermore, the intrinsic problem constraints (regarding the nurses' active and passive attention, the number of beds and no over-time allowance) are formed identical to those from the *Base model* (Equations (4.25) to (4.28), respectively).

$$\rho_{1ndt} = \sum_{j\in\mathcal{J}} \sum_{t-(\pi_{j1})+1\le t'\le t} x_{ndt'j} \qquad \forall n\in\mathcal{N}, d\in\mathcal{D}, t\in\mathcal{T} \tag{4.22}$$

$$\rho_{2ndt} = \sum_{j\in\mathcal{J}} \sum_{t-(\pi_{j1}+\pi_{j2})+1\le t'\le t-\pi_{j1}} x_{ndt'j} \qquad \forall n\in\mathcal{N}, d\in\mathcal{D}, t\in\mathcal{T} \tag{4.23}$$

$$\rho_{3ndt} = \sum_{j\in\mathcal{J}} \sum_{t-(\pi_{j1}+\pi_{j2}+\pi_{j3})+1\le t'\le t-(\pi_{j1}+\pi_{j2})} x_{ndt'j} \qquad \forall n\in\mathcal{N}, d\in\mathcal{D}, t\in\mathcal{T} \tag{4.24}$$

$$\rho_{1ndt} + \rho_{3ndt} \le \alpha \qquad \forall n\in\mathcal{N}, d\in\mathcal{D}, t\in\mathcal{T} \tag{4.25}$$

$$\sum_{n\in\mathcal{N}} \rho_{2ndt} \le \beta \qquad \forall d\in\mathcal{D}, t\in\mathcal{T} \tag{4.26}$$

$$\sum_{n\in\mathcal{N}} (\rho_{1ndt} + \rho_{2ndt} + \rho_{3ndt}) \le R \qquad \forall d\in\mathcal{D}, t\in\mathcal{T} \tag{4.27}$$

$$x_{ndtj} = 0 \qquad \forall n\in\mathcal{N}, d\in\mathcal{D}, j\in\mathcal{J}, t\in\{\bar{t}-(\pi_{j1}+\pi_{j2}+\pi_{j3})+1,\dots,\bar{t}\} \tag{4.28}$$

4.3.2 Implementation Details

The enhancements made in the model are identical to those applied in the base model (Section 4.1). In both cases, the same process of reducing auxiliary variables, removing unnecessary constraints, and compressing constraint sets has been utilized. The use of CPLEX is consistent throughout, benefiting from these reductions to optimize the solution time and quality.

To wrap up all the enhancements made, here is how the implemented model looks like

$$\max Z = \sum_{n\in\mathcal{N},d\in\mathcal{D},t\in\mathcal{T},j\in\mathcal{J}} x_{ndtj} \tag{4.29}$$

$s.t.:$

$$\sum_{n\in\mathcal{N},d\in\mathcal{D},t\in\mathcal{T}} x_{ndtj} \leq \nu_j \qquad \forall j\in\mathcal{J} \tag{4.30}$$

$$\sum_{j\in\mathcal{J}} \sum_{t-(\pi_{j1})+1\leq t'\leq t} x_{ndt'j} + \sum_{j\in\mathcal{J}} \sum_{t-(\pi_{j1}+\pi_{j2}+\pi_{j3})+1\leq t'\leq t-(\pi_{j1}+\pi_{j2})} x_{ndt'j} \leq \alpha \qquad \forall n\in\mathcal{N}, d\in\mathcal{D}, t\in\mathcal{T} \tag{4.31}$$

$$\sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{J}} \sum_{t-(\pi_{j1})+1\leq t'\leq t} x_{ndt'j} + \sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{J}} \sum_{t-(\pi_{j1}+\pi_{j2})+1\leq t'\leq t-\pi_{j1}} x_{ndt'j} + \sum_{n\in\mathcal{N}}\sum_{j\in\mathcal{J}} \sum_{t-(\pi_{j1}+\pi_{j2}+\pi_{j3})+1\leq t'\leq t-(\pi_{j1}+\pi_{j2})} x_{ndt'j} \leq R \qquad \forall d\in\mathcal{D}, t\in\mathcal{T} \tag{4.32}$$

$$\sum_{n\in\mathcal{N}}\sum_{d\in\mathcal{D}}\sum_{j\in\mathcal{J}} \sum_{t\in\{\bar{t}-(\pi_{j1}+\pi_{j2}+\pi_{j3})+1,\ldots,\bar{t}\}} x_{ndtj} = 0 \tag{4.33}$$

$$x_{ndtj}\in\{0,1\} \qquad \forall n\in\mathcal{N}, d\in\mathcal{D}, t\in\mathcal{T}, j\in\mathcal{J} \tag{4.34}$$

4.4 Incorporating waiting-times

As mentioned previously in the Chapter 1, the efficiency of appointment scheduling systems in healthcare directly influences patient satisfaction, resource utilization, and the overall effectiveness of healthcare delivery. Waiting times stand as a critical metric within these systems, serving as a principal indicator of both accessibility and quality of care. Extended waiting periods can precipitate patient dissatisfaction, diminished adherence to treatment plans, and potentially adverse health outcomes, underscoring the optimization of waiting times as a paramount objective for healthcare administrators intent on enhancing service delivery and patient experience.

In Canada, access to specialist care and medical treatments is commonly evaluated using a two-stage waiting-time framework. **Wait1** measures the time between a general practitioner (GP) referral and the initial specialist consultation, while **Wait2** measures the time from the specialist's decision to treat to the patient's first appointment for receiving treatment (gov (2025); hqo (2025); cih (2025)). National reporting standards, such as those used by the Fraser Institute (sta (2023)) and provincial health authorities, rely on these two intervals to summarize delays in the care pathway. For example, in Prince Edward Island, the combined median waiting time exceeded 64 weeks in 2023, while the national average reached 41.7 weeks, with an additional median wait of 23 weeks between specialist consultation and treatment (Liddy et al. (2024)).

However, by construction, this two-part metric captures only the waiting times of patients who have been scheduled; that is, those who have already secured either a specialist appointment (**Wait1**) or a treatment appointment (**Wait2**). Patients who remain in the backlog without an assigned appointment are not reflected in these measures, even though they may constitute a significant share of the total demand for care. This limitation has motivated complementary

metrics that explicitly quantify backlog volume or "queue length," such as the number of patients waiting without an appointment, the proportion exceeding a target wait threshold, or the ratio of arrivals to completions. The health-operations literature has addressed these queue-based metrics extensively (gov (2025)), emphasizing that backlog size is a separate but essential dimension of access performance.

In the variation of the model developed in this section, we adopt the standard Canadian convention of tracking only **Wait2** for scheduled patients and assume a fixed planning horizon in which the backlog is fully known at the outset. Patients who are not scheduled within this horizon are carried forward as backlog to the next planning cycle. This assumption differs from the earlier formulation in Section 4.3, in which all patients were assumed to be available at time $r = 0$. In the present model, decision-making requires information at earlier time points $r < 0$, including the number of patients who have already checked into the clinic and are ready to be scheduled.

The remainder of this section is dedicated to introducing the model under group scenario in detail and mathematical formulation of assumptions.

4.4.1 Mathematical Model for Treatment Types - Deferral-Penalized Formulation

In contrary to the *base problem*, in this case we can not rely on the information of arrival and scheduling for each patient individually, since they are grouped by their treatment types. Therefore, a more sophisticated 2-stage approach to both problem definition and modeling is required. This section presents a MILP that takes the optimal $Capacity^*$ as a parameter from the *type-based model* as the first stage, and optimizes slot allocation and patient selection, incorporating a deferral penalty to discourage leaving long-waiting patients unscheduled. The parameters of the problem are first defined, then the second stage of the mathematical model is presented.

This problem (Scheduling grouped patients by their treatment types with consideration of waiting times) is an extension of the Section 4.3 problem, hence most of the parameters are inherited from the model introduced there, and only the extended parameters and decision variables and equations are explained in detail in Tables 4.5 and 4.6.

Table 4.5: Extended Parameters of the "Type-Based Model with waiting-times"

Name	Definition	Value(s)
	Sets	
$r \in \mathcal{R}$	$\mathcal{R}$ is the set of arrival days	$\mathcal{R} = \{-\bar{r}, \ldots, -2, -1\}$
	Parameters	
Cap^*	Optimal capacity from the type-based model (derived from the first stage)	$Cap^* \leq \sum_{j \in \mathcal{J}} \nu_j$
ζ_{jr}	Number of patients with treatment type j arriving at day r (relative to the start of the planning horizon)	$0 \leq \zeta_{jr} \leq \nu_j$ and $\sum_{j \in \mathcal{J}, r \in \mathcal{R}} \zeta_{jr} = \sum_{j \in \mathcal{J}} \nu_j$
λ	Deferral-penalty weight: penalizes each unscheduled patient proportionally to their accumulated waiting time	$\lambda \geq 0$
ϵ	percentage of the patients that we can skip relative to the max theoretical capacity (Cap^*)	$0 \leq \epsilon \leq 1$

The model for the second stage, i.e. the optimization of slot allocation and patient selection, directly extends the MILP from Section 4.3 by augmenting the objective with a deferral penalty, adding linking constraints between x_{ndtj} and y_{jr}, and allowing to deviate from the best possible capacity (Cap^*) by a percentage ϵ.

Table 4.6: Extended Decision Variables of the "Type-Based Model with waiting-times"

Variable	Definition	Domain
y_{jr}	Number of treatment types j that have arrived to our system on day r and have been assigned a time slot in the planning period	$y_{jr} \in \{0, 1, 2, \ldots, \bar{p}\}$

$$\begin{aligned} \min TotWT = & \sum_{j\in\mathcal{J}, r\in\mathcal{R}} (-r)\cdot y_{jr} \\ & + \sum_{n\in N, d\in D, t\in T, j\in J} d \cdot x_{ndtj} \\ & + \lambda \sum_{j\in\mathcal{J}, r\in\mathcal{R}} (-r)\cdot(\zeta_{jr} - y_{jr}) \end{aligned} \tag{4.35}$$

Equation (4.35) defines the objective function for this problem instance and is composed of three terms:

a) $\sum_{j\in\mathcal{J}, r\in\mathcal{R}}(-r)\cdot y_{jr}$: aggregates the pre-horizon portion of each scheduled patient's waiting time; the duration between appointment request (arrival day $r < 0$) and the start of the planning horizon.

b) $\sum_{n\in N, d\in D, t\in T, j\in J} d \cdot x_{ndtj}$: sums the in-horizon portion; the number of days from the start of the planning period to the actual appointment.

c) $\lambda \sum_{j\in\mathcal{J}, r\in\mathcal{R}}(-r)\cdot(\zeta_{jr} - y_{jr})$: **deferral penalty**, for each patient left unscheduled, a penalty proportional to their accumulated wait $|r|$ is incurred, weighted by $\lambda \geq 0$.

Setting $\lambda = 1$ gives equal weight to deferral and marginal scheduling benefit and is used throughout this work; $\lambda \to \infty$ forces pure FIFO selection.

The MILP optimizer jointly determines x_{ndtj} and y_{jr} subject to the constraints in Equations (4.36) and (4.37).

$$\sum_{r\in\mathcal{R}} y_{jr} = \sum_{n\in N, d\in D, t\in T} x_{ndtj} \qquad \forall j \in \mathcal{J} \tag{4.36}$$

$$y_{jr} \leq \zeta_{jr} \qquad \forall j \in \mathcal{J}, r \in \mathcal{R} \tag{4.37}$$

$$\sum_{n\in N}\sum_{d\in D}\sum_{t\in T}\sum_{j\in\mathcal{J}} x_{ndtj} \geq \epsilon Cap^* \tag{4.38}$$

Equation (4.36) ensures that the total number of patients assigned across all arrival cohorts equals the total number of scheduled slots for treatment type j. Equation (4.37) enforces that the number of patients assigned from cohort r does not exceed the cohort size ζ_{jr}. Equation (4.38) ensures that the total number of scheduled slots is at least ϵ percent of the maximum capacity.

Chapter 5

Simulation

In this chapter, the methodology used to generate randomised problem instances is described first, followed by the construction of the FIFO baseline and the statistical framework used to evaluate the solution methods. Each solution method is then assessed by comparing its improvement over the FIFO baseline, visualised as paired-difference confidence-interval plots and summarised in a Wilcoxon signed-rank test table. The chapter closes with an analysis of the waiting-time trade-off.

5.1 Instance Generation

As stated in the previous section, a set of parameters are needed in order to define the problem developed in this thesis. All of which, have been generated following a discrete uniform distributed distribution, where the parameters (lower bound, upper bound and steps of quantization) are specified based on the experiment size.

Three experiment sizes have been defined for this problem as followed:

- **Small**: Small problem instances with planning horizon of 1 day for a single nurse, and their sole purpose is to verify and validate the model and its implementation.
- **Medium**: Medium instances are the problem instances with planning horizon of less than a week and 2-4 nurses. These instances provide insights on the solution quality of the proposed solutions and heuristics compared to conventional mathematical solvers.
- **Large**: Large problem instances are referring to the instances with planning horizons longer than a week with 4-6 nurses. These problem instances benchmark the solution method's performance on large-scale problems similar to real life examples.

The uniform distribution parameters for each instance size are shown in Table 5.1.

Table 5.1: Uniform distribution parameters for the Task based and Type Based problems

	Uniform Distribution Parameters ($[lb, ub, steps]$)		
Name	Small	Medium	Large
$\bar{n}$	$[1, 1, 1]$	$[2, 4, 1]$	$[4, 6, 1]$
d	$[1, 1, 1]$	$[2, 3, 1]$	$[5, 5, 5]$
$\bar{t}$	$[72, 72, 1]$	$[60, 60, 1]$	$[66, 72, 1]$
$\bar{p}$	$[50, 80, 5]$	$[300, 400, 10]$	$[800, 1000, 50]$
ν_j	Uniformly drew so that always $\sum_{j \in \mathcal{J}} \nu_j = \bar{p}$ holds true		
π_{j2} and π_{p2}	$[12, 15, 1]$	$[6, 10, 1]$	$[12, 18, 1]$

The rest of the parameters are set to their constant values. In the subsequent sections, the solution methods proposed in Chapter 4 are going to be put into testing against the problem instances generated.

5.2 FIFO Baseline

To provide a meaningful reference point for evaluating the solution methods, we construct a *FIFO* (first-in, first-out) baseline schedule: patients are assigned to slots in the order they arrive, with no reordering or combinatorial optimisation. The same greedy decoder used to convert GA chromosomes into feasible schedules is applied here, ensuring that any performance difference reflects the quality of the assignment rather than decoding artefacts. Fifty problem instances are generated for each size class (small, medium, large), matching the experimental setup described in Section 5.1. The FIFO schedule serves as a practical lower bound on throughput and as the paired reference for the hypothesis tests described in the following section.

5.3 Statistical Hypothesis Testing

5.3.1 Hypotheses

For each instance i and solution method m, let $\Delta_i^{(m)} = \mathrm{OF}_i^{(m)} - \mathrm{OF}_i^{\mathrm{FIFO}}$ be the difference in the number of patients scheduled relative to the FIFO baseline. We test

$$H_0 : \mathrm{median}(\Delta^{(m)}) = 0 \quad \text{(the method is not better than FIFO)}, \tag{5.1}$$

$$H_1 : \mathrm{median}(\Delta^{(m)}) > 0 \quad \text{(the method schedules more patients than FIFO)}, \tag{5.2}$$

at significance level $\alpha = 0.05$. The one-sided alternative is chosen because we are interested specifically in improvements, not deteriorations.

5.3.2 Choice of Test: Wilcoxon vs. Paired t-Test

The paired t-test is the natural first candidate: it is the uniformly most powerful unbiased test for normally distributed differences and is widely understood. However, three properties of the scheduling objective make it a poor fit here.

- **Bounded, integer-valued outcomes.** The number of patients scheduled is a non-negative integer bounded above by the pool size $\bar{p}$. Such distributions are inherently discrete and right-skewed, not approximately normal, especially at the extreme ends of the scale.
- **Moderate sample size.** With $n = 50$ paired observations per group, the Central Limit Theorem provides only partial protection; a mild departure from normality in Δ_i can inflate the type I error of the t-test.
- **Heavy tails and outliers.** Instances in which the solver hits a time limit or returns an infeasible solution produce very large negative differences that are not representative of the method's typical behaviour but strongly influence the t-statistic.

The Wilcoxon signed-rank test Wilcoxon (1945) addresses all three issues. It is non-parametric: the only distributional assumption is that the paired differences Δ_i are drawn from a continuous, symmetric distribution, a far weaker requirement than normality. The test ranks the absolute

differences $|\Delta_i|$, sums the ranks of the positive and negative differences separately, and bases inference on the smaller sum; outliers therefore affect the result only through their rank, not their magnitude.

5.3.3 Wilcoxon Signed-Rank Test

The paired design (each FIFO schedule is generated from the same instance as the corresponding method schedule) exploits the within-instance correlation and increases statistical power relative to an independent-samples test. Each group contains $n = 50$ paired observations, except for the Rolling Horizon heuristic, for which only medium and large instances are available (small instances reduce to a single-day, single-nurse problem equivalent to an exact solve).

5.4 IBM CPLEX

The very first runs of the optimization model as stated in the previous sections are solving the whole MILP model using IBM ILOG CPLEX optimization studio through the python libraries and Application Programming Interface (API) provided by the IBM itself. The idea behind this set of experiments is to capture the efficiency and effectiveness of the MILP models developed on large-scale problem instances, such as planning the visits for over a week.

The improvement of the Task-Based and Type-Based CPLEX models over the FIFO baseline is shown in Section 5.4. Each box shows the distribution of paired differences Δ_i across 50 instances; the diamond marker and error bars indicate the mean ± 95% confidence interval. The detailed run-time results are provided in Appendices A and B.

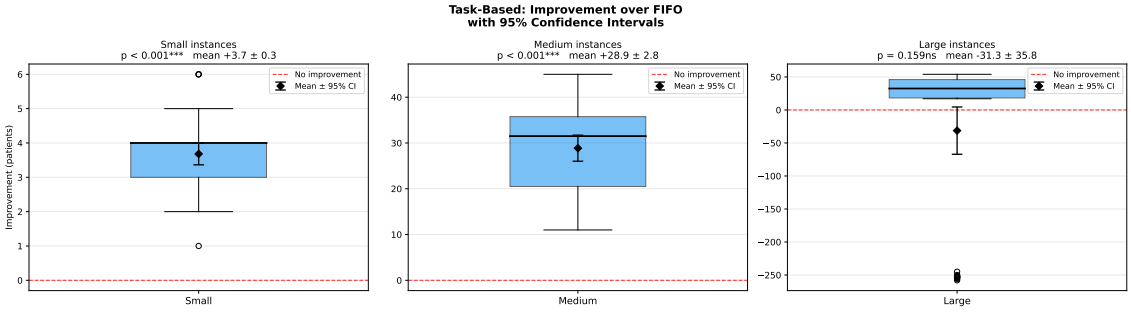

Figure 5.1: Task-Based CPLEX model: improvement in patients scheduled over the FIFO baseline, with 95% confidence intervals, per instance size.

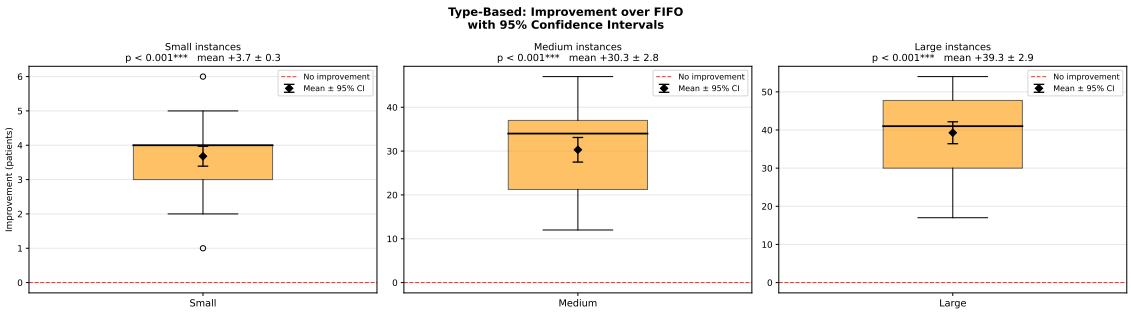

Figure 5.2: Type-Based CPLEX model: improvement in patients scheduled over the FIFO baseline, with 95% confidence intervals, per instance size.

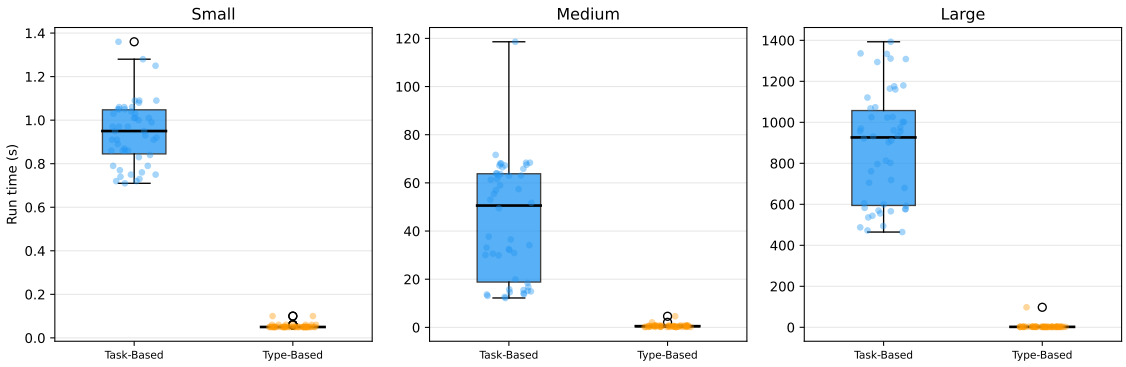

Figure 5.3: Run times for 50 instances per size, IBM CPLEX.

As shown in Section 5.4, the Type-Based model outperforms the FIFO baseline significantly on all instance sizes, while the Task-Based model fails to do so on large instances ($p = 0.159$) because the solver frequently hits the time limit and returns incomplete solutions. The run-time boxplots (Figure 5.3) confirm that the Type-Based model is also considerably faster, reflecting the tighter LP relaxation of its aggregated formulation.

5.5 Rolling Horizon

It can be seen from the runs at Section 5.4 that the model size affects the performance of the MILP model tremendously. To address this issue in the Base model, without compromising the advantages of this model, such as having patient-specific information inside the model, we can apply a heuristic based on a rolling horizon (RH) approach.

This approach has emerged as a crucial decision-making framework in production planning, offering a dynamic and adaptive approach to managing complex manufacturing environments. The rolling horizon heuristic is a decision-making framework used in dynamic and complex environments, particularly in production planning and scheduling. It involves breaking down a long-term planning problem into a series of shorter, overlapping planning periods. At each stage, decisions are made for the immediate period while considering future periods, but only the decisions for the current period are implemented (Clark (2005)).

Section 5.5 shows the improvement of the Rolling Horizon heuristic over the FIFO baseline on medium and large instances (small instances are omitted because the RH decomposition is equivalent to an exact solve for a single-day, single-nurse problem).

The RH heuristic outperforms FIFO significantly on both medium and large instances ($p < 0.001$), with improvement comparable to the Type-Based exact model at a fraction of the computation time, as shown by the run-time boxplots in Figure 5.5. Appendix C contains the detailed raw data of these experiments.

5.6 Genetic Algorithms

Another way to solve the aforementioned problem is to employ Genetic Algorithms (GAs) as a trusted meta-heuristic approach in the literature. In this section, only FANAR objective function is implemented as it is proven being computationally superior efficient-wise compared to the SPISS method. Section 5.6 shows the improvement of the GA over the FIFO baseline across all instance sizes. Moreover, Appendix D presents detailed output of these runs in a tabular format.

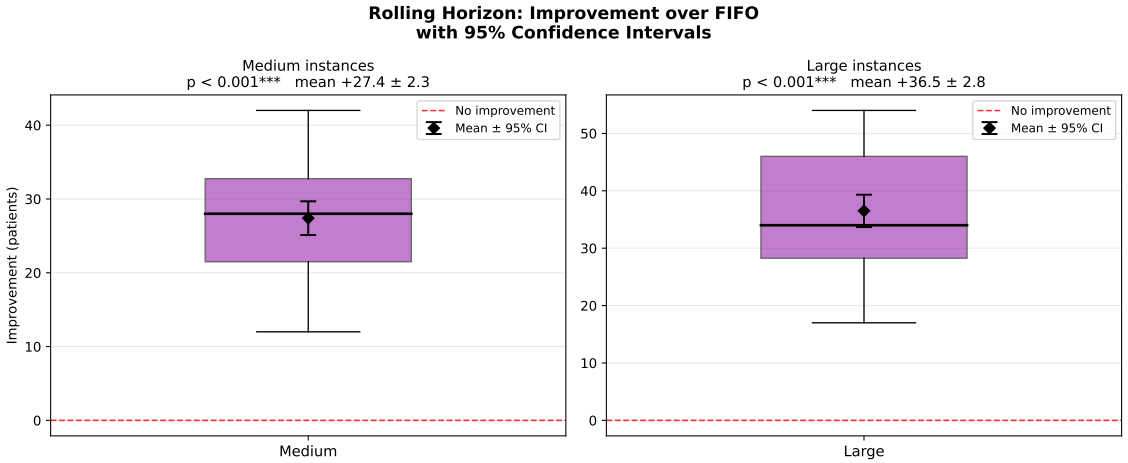

Figure 5.4: Rolling Horizon heuristic: improvement in patients scheduled over the FIFO baseline, with 95% confidence intervals, for medium and large instances.

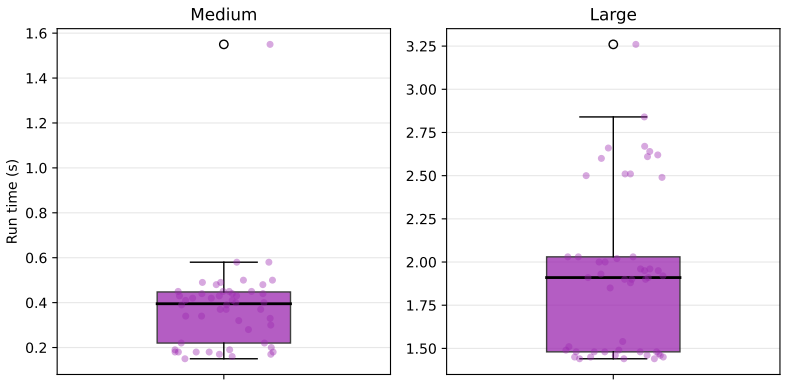

Figure 5.5: Run times for 50 instances per size, Rolling Horizon.

The GA achieves improvements over FIFO that are statistically significant on all instance sizes, verifying the implementation on small instances (where it matches the exact solution quality) and demonstrating robust, sub-optimal but reliable gains on medium and large instances.

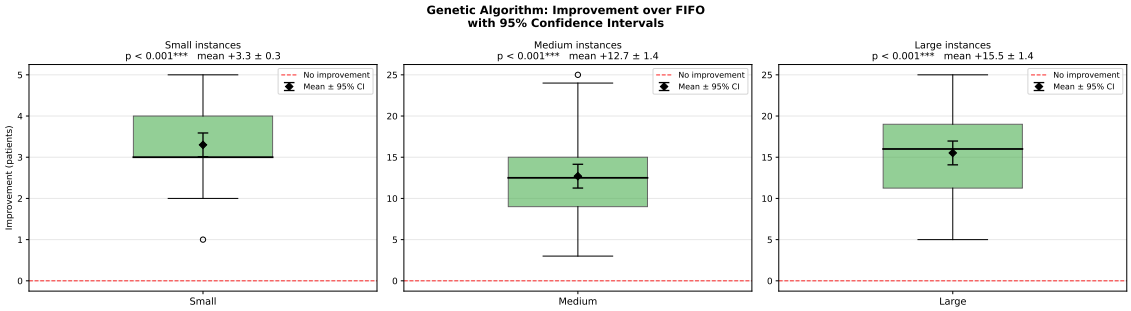

Figure 5.6: Genetic Algorithm: improvement in patients scheduled over the FIFO baseline, with 95% confidence intervals, per instance size.

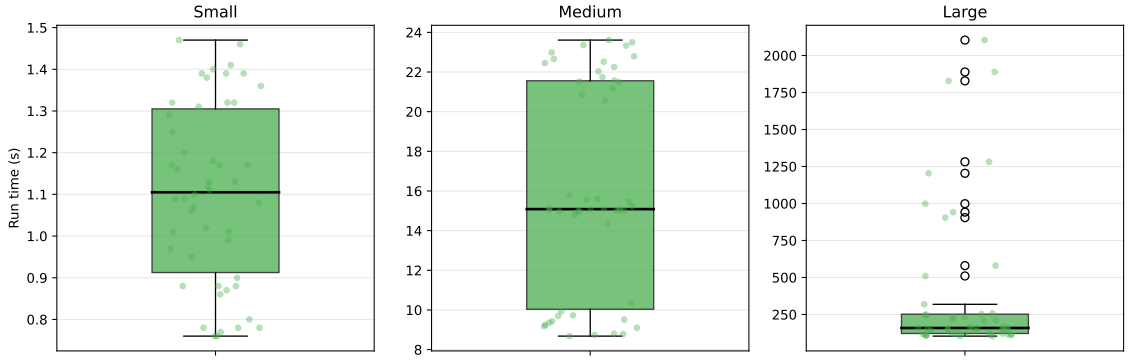

Figure 5.7: Run times for 50 instances per size, Genetic Algorithm.

5.7 Results Summary

Table 5.2 summarises the Wilcoxon signed-rank test results for all four methods across all available instance sizes. Each cell reports the p-value and the mean improvement ± half-width of the 95% confidence interval.

Table 5.2: Wilcoxon signed-rank test results: improvement over FIFO baseline ($n = 50$, $\alpha = 0.05$).

Model		**Small**	**Medium**	**Large**
Task-Based	p-value	$p < 0.001^{***}$	$p < 0.001^{***}$	$p = 0.159$ ns
	mean (CI)	+3.7±0.3	+28.9±2.9	−31.3±35.8
Type-Based	p-value	$p < 0.001^{***}$	$p < 0.001^{***}$	$p < 0.001^{***}$
	mean (CI)	+3.7±0.3	+30.3±2.8	+39.3±2.9
GA	p-value	$p < 0.001^{***}$	$p < 0.001^{***}$	$p < 0.001^{***}$
	mean (CI)	+3.3±0.3	+12.7±1.4	+15.5±1.4
Rolling Horizon	p-value	—	$p < 0.001^{***}$	$p < 0.001^{***}$
	mean (CI)	—	+27.4±2.3	+36.5±2.8

All methods outperform FIFO significantly on the instance sizes for which they produce complete solutions. The sole exception is Task-Based on large instances ($p = 0.159$), where the solver

frequently hits the time limit. Type-Based achieves the largest absolute gains among the exact methods, reflecting the tighter LP relaxation of its formulation. The Rolling Horizon heuristic matches the Type-Based quality on medium and large instances while running in a fraction of the time. GA improvements are smaller in absolute terms but are statistically robust across all sizes, consistent with its sub-optimal heuristic nature.

5.8 Waiting Time

This section evaluates the deferral-penalized waiting-time model (defined in Section 4.4) on large problem instances. The central question is: how does the deferral-penalty coefficient λ affect *which* patients are denied service, and how equitably are denials distributed across arrival cohorts?

5.8.1 Experiment Setup

The experiment follows the two-stage solve procedure from Section 4.4.

Stage 1. The Type-Based model that maximises PatientNum is solved with CPLEX (10-minute timeout, 5% optimality gap), yielding an optimal capacity Cap^*.

Stage 2. The deferral-penalized MILP (objective Equation (4.35)) is solved four times per instance, once for each $\lambda \in \{0, 1, 5, 1000\}$. In each solve the capacity constraint Equation (4.38) is set to target $= \varepsilon \cdot \text{Cap}^*$ with $\varepsilon = 0.9$, so that at most a fraction of patients are deferred (compared to the maximum capacity).

The instance set consists of **50 large instances** (4–6 nurses, 5-day horizon, 800–1000 patients) generated by the same procedure described in Section 5.1.

Defining yp_r. After Stage 2 we define the *unscheduled patient count per cohort* as

$$yp_r = \sum_j (\zeta_{jr} - y_{jr}), \qquad \forall\, r \in R, \tag{5.3}$$

where y_{jr} is the Stage-2 decision variable (number of type-j patients from cohort r who are scheduled) and ζ_{jr} is the total backlog of type-j patients from cohort r (see Table 4.5). The vector $\{yp_r\}$ characterises the distribution of deferral across arrival cohorts, with $r \in \{-\bar{r}, \ldots, -1\}$: $r = -1$ is the most-recent cohort (shortest wait); $r = -\bar{r}$ is the cohort that has waited the longest. The aggregate $N = \sum_r yp_r$ gives the total number of denied patients for the instance.

5.8.2 Fairness Metrics

Two complementary metrics quantify the distributional equity of denials.

Metric 1: Weighted-mean cohort index μ_{norm}. Let $N = \sum_r yp_r$ be the total number of denied patients. Define

$$\mu = \frac{\sum_r r \cdot yp_r}{N}, \qquad \mu_{\text{norm}} = \frac{\mu - (-\bar{r})}{\bar{r} - 1} \in [0, 1].$$

$\mu_{\text{norm}} = 0$ means all denials fall on the earliest cohort (most unfair to long-waiters); $\mu_{\text{norm}} = 1$ means all denials fall on the most-recent cohort. Instances with $N = 0$ (all patients served) are excluded. The weighted mean is a standard summary statistic for discrete distributions over ordered categories and directly reflects the notion of distributional equity used in appointment-scheduling literature Gupta and Denton (2008). Normalising by $(\bar{r} - 1)$ allows comparison across instances with differing backlog lengths.

Metric 2: Early fraction f_{early}.

$$f_{\text{early}} = \frac{\sum_{r \le r_{\text{med}}} yp_r}{N},$$

where r_{med} is the median cohort index present in the instance. A high f_{early} means the majority of denied patients are from the older half of the queue, providing a threshold-based view of inequity that complements μ_{norm}. The clinical motivation is that extended waits impose substantial quality-of-life costs on patients Liddy et al. (2024), making it especially important to detect whether long-waiting patients are systematically the ones denied.

5.8.3 Results

Figure 5.8 shows the mean (± 1 std) unscheduled patient count per cohort for each λ, across the 50 large instances. Each panel corresponds to one penalty level; cohort index $r = -1$ denotes the most-recent arrivals.

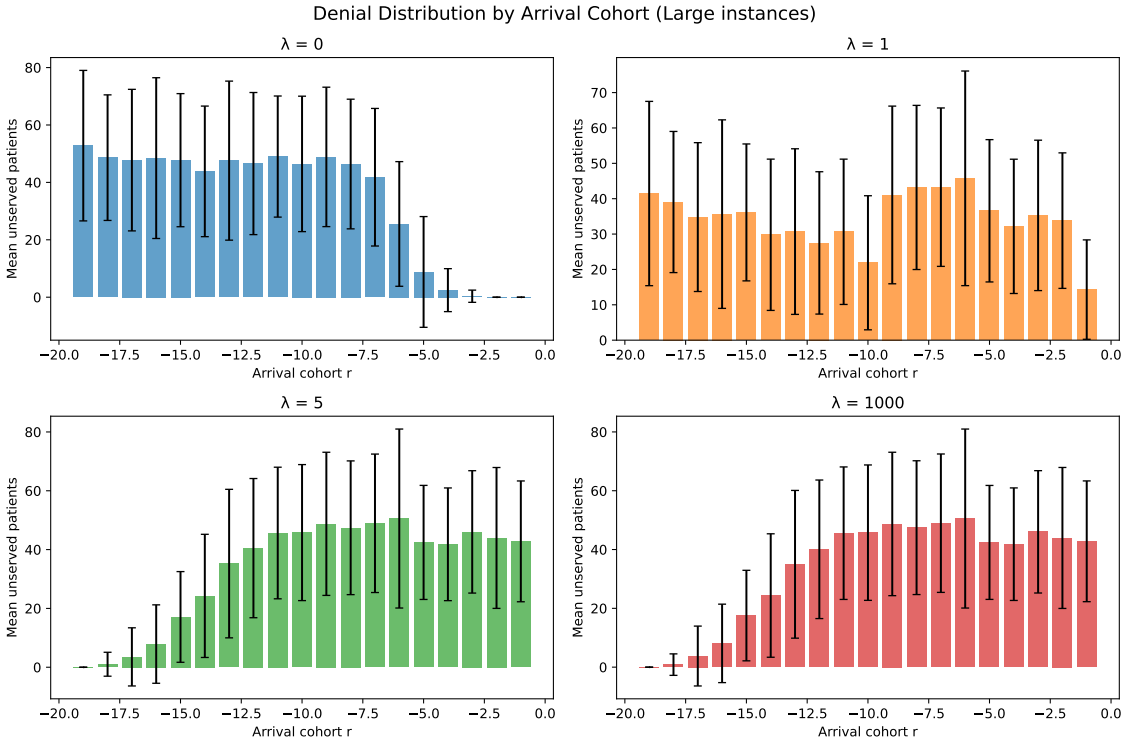

Figure 5.8: Mean unscheduled patients per arrival cohort, by penalty level.

Figure 5.9 plots both fairness metrics as a function of λ. The left axis shows μ_{norm} (blue) and the right axis shows f_{early} (red); error bars indicate one standard deviation.

Figure 5.10 displays the per-instance μ_{norm} values across all λ levels (rows: instances, columns: λ). Darker shading indicates denials concentrated among long-waiting cohorts.

The key findings from the 200-row dataset (50 instances $\times$ 4 λ values) are as follows.

$\lambda = 0$. Pure wait-time minimisation consistently sacrifices long-waiting patients: $\mu_{\text{norm}} \approx 0.32$–$0.43$ and $f_{\text{early}} \approx 0.66$–$0.83$. The bar charts in Figure 5.8 show higher mean denials in the most-negative cohorts, confirming that the unconstrained objective tends to defer patients who have already waited the longest.

$\lambda = 1, 5$. The deferral penalty progressively redistributes denials toward more-recent arrivals:

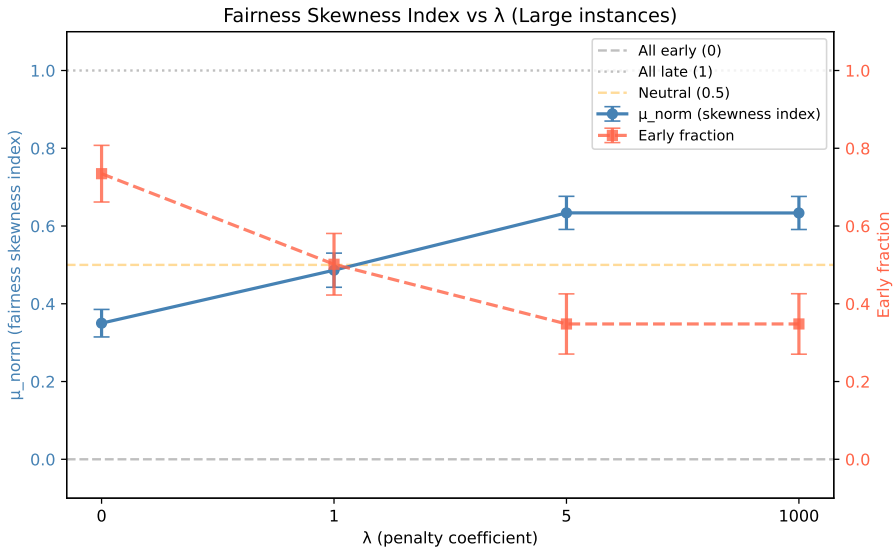

Figure 5.9: Fairness metrics μ_{norm} and f_{early} as a function of λ.

μ_{norm} shifts toward 0.50–0.73 and f_{early} drops markedly. Both metrics improve monotonically in λ.

$\lambda = 1000$. Metrics converge with those at $\lambda = 5$, exhibiting diminishing returns. This is consistent with the theoretical result in Section 4.4 that $\lambda \to \infty$ forces FIFO selection Bailey (1952).

The per-instance heatmap (Figure 5.10) confirms that this shift is systematic rather than driven by outliers: virtually every instance exhibits the same monotone trend across λ.

Hence, a moderate penalty of $\lambda \approx 5$ achieves meaningful distributional fairness without imposing the full rigidity of a FIFO discipline.

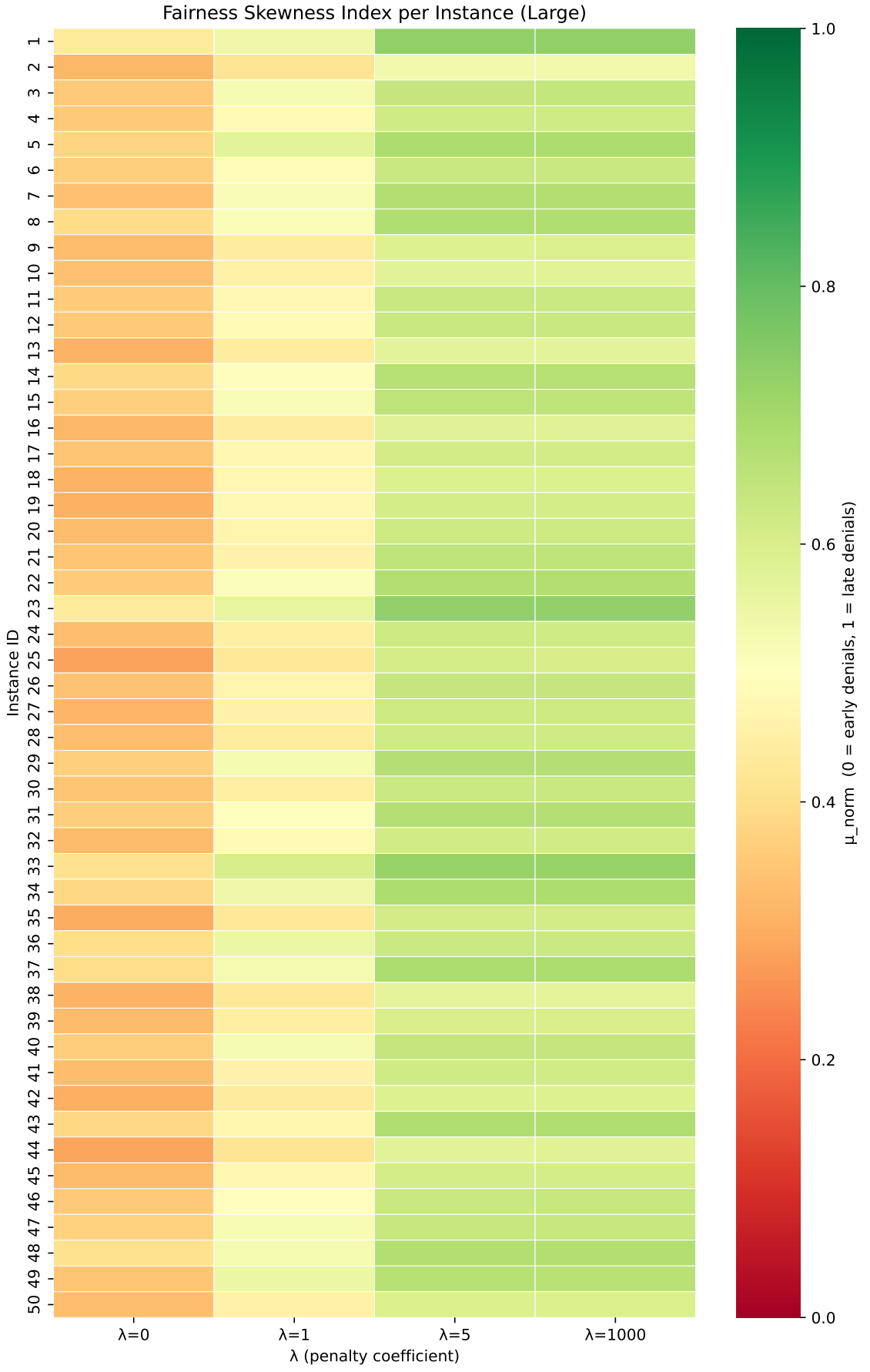

Figure 5.10: Per-instance heatmap of μ_{norm} across λ values.

Chapter 6

Conclusive Discussion and Future Work

This chapter synthesises the key findings of the thesis and situates them within the broader appointment-scheduling literature. Section 6.1 discusses the base problem: the four solution methods (Base Model, Type-Based Model, Genetic Algorithm, and Rolling Horizon heuristic) are evaluated and compared in terms of solution quality and scalability. Section 6.2 discusses the extended problem with waiting times, focusing on how the deferral-penalty coefficient λ controls the equity of patient denials and on directions for future model development. The chapter closes with Section 6.3, which identifies the practical implications of this work and the steps required to bridge the gap between the mathematical models and real-world clinical deployment.

6.1 Base Problem

For the base problem, three solution methods have been developed in the Chapter 4, that is, the base mode, type-based model and the Genetic Algorithm, and a new heuristic was tested out in Chapter 5 (Rolling Horizon). The results has been presented in the previous chapter, and they looked promising.

First, all four solution methods resulted in the same results for small instances, indicating that they were implemented correctly and are targeting the same problem. The only difference was their performance metric, i.e., run times. It is noticeable that the Type-based model has outperformed the other methods even in small instances with a huge gap. The same results have been observed for medium instances.

Even though the base model was outperformed by the Type-based model at the cost of losing critical individual-based information about each patient, the RH heuristic proved to be effective in scaling up the base model and enabling it to compete with other models with minimal impacts on the quality of the solution.

To sum up, our finding is that if the individual patient information (preferences, arrival time, and patient's history and etc) are not considered in the problem, the best model to be used for providing appointments is the Type-based model. However, this is not always the case. Therefore, RH heuristic is suggested for such cases where it makes scaling up the detail-oriented model (base model) possible.

As per the future directions, using RH might interfere with other problem structures (e.g. considering nurses' fatigue-ness, or waiting times); thus, future studies should concern these dynamics between each rolling planning period. Furthermore, since the structure of the problem also resem-

bles the multi-knapsack problem to a great extent, applying approaches from that literature, such as reinforcement-learning methods or column generation techniques, seems promising.

Finally, plenty of problem aspects were relaxed to ensure the efficiency of the solution methods for this problem. The length of *initialization* and *finalization* periods (which is assumed to be 1 time slot, or 5 minutes) is such assumption. A future gap to be filled is to propose an effective approach to account for various *initialization* and *finalization* period durations.

6.2 Extended Problem with Waiting Times

The Type-Based model maximizes throughput (the number of patients scheduled within the planning horizon) but is entirely agnostic about *which* patients are deferred when capacity is exhausted. Without an explicit equity objective, the optimizer naturally defers those patients whose scheduling cost is highest, which in practice tends to mean patients who have already waited the longest. Long-waiting patients can therefore be systematically denied service in consecutive planning cycles, worsening their health outcomes and undermining the fairness of the appointment system.

To address this limitation, the deferral-penalized two-stage MILP introduced in Section 4.4 augments the Type-Based model with an explicit equity mechanism. In Stage 1, the Type-Based model is solved to obtain the optimal throughput capacity Cap^*. In Stage 2, the deferral-penalized MILP is solved with Cap^* as a fixed parameter: the objective (Equation (4.35)) minimizes total waiting time while adding a penalty term $\lambda \sum_{j,r}(-r)(\zeta_{jr} - y_{jr})$ that charges λ times the accumulated wait for every patient left unscheduled. The capacity constraint (Equation (4.38)) ensures that at least $\varepsilon \cdot \text{Cap}^*$ patients are served, so throughput does not collapse as λ grows.

The simulation experiments (Section 5.8) evaluated this two-stage formulation across 50 large instances (800–1000 patients, 4–6 nurses, 5-day horizon) for $\lambda \in \{0, 1, 5, 1000\}$. The results are consistent and informative. At $\lambda = 0$, the model collapses to pure wait-time minimisation and concentrates denials heavily on the longest-waiting cohorts ($\mu_{\text{norm}} \approx 0.32$–$0.43$, $f_{\text{early}} \approx 0.66$–$0.83$), confirming that an unconstrained throughput objective is inherently inequitable. Increasing λ to 1 and then to 5 progressively redistributes denials toward more-recent arrivals: μ_{norm} shifts toward 0.50–0.73 and f_{early} drops. At $\lambda = 1000$ the metrics converge with those at $\lambda = 5$, exhibiting diminishing returns consistent with the theoretical expectation that $\lambda \to \infty$ recovers a pure FIFO rule. The per-instance heatmap confirms that this trend is systematic across all 50 instances rather than being driven by outliers.

The central practical conclusion is that a moderate penalty of $\lambda \approx 5$ achieves meaningful distributional fairness, halving the concentration of denials among long-waiters relative to $\lambda = 0$, without imposing the full rigidity of FIFO ordering. This represents a principled, tunable trade-off between throughput and equity that is absent from the base formulation.

Several directions remain open for future work. First, the capacity tolerance ε was fixed at 0.9 throughout these experiments; a sensitivity analysis varying ε would clarify how much throughput must be sacrificed to achieve a given equity target, and whether (λ, ε) interact in ways that allow joint calibration. Second, the current approach selects λ offline by a grid search; an adaptive or dynamic tuning strategy; for example, a reinforcement-learning agent that adjusts λ at each planning cycle in response to observed queue dynamics, would allow the system to self-calibrate without manual intervention. Third, the fairness objectives used here (μ_{norm} and f_{early}) are intuitive but not the only possibilities; richer objectives such as the Gini coefficient of the deferral distribution or a max-min equity criterion (which minimizes the maximum denial experienced by any single cohort) could capture different clinical priorities. Finally, the deferral-penalty term currently lives

only in Stage 2 of the MILP; integrating an analogous equity signal into the fitness function of the Genetic Algorithm would extend the fairness mechanism to the heuristic solver and allow it to be applied to instances where CPLEX times out.

6.3 Practical Implications

The models developed in this thesis are necessarily abstractions of clinical reality. The most significant simplification is the treatment of nurses as interchangeable, stationary resources available throughout the full planning horizon. In practice, nurses experience fatigue over long shifts, are subject to scheduled breaks and mandatory rest periods, and may possess heterogeneous skill mixes that restrict which treatment types they can perform. Incorporating fatigue-aware capacity constraints, shift-structure limitations, and skill-mix requirements is an important prerequisite for deploying these models in an operational setting.

Further real-world gaps include patient no-shows and last-minute cancellations, which introduce stochastic demand that is not captured by the deterministic MILP formulations. Emergency insertions (patients who must be accommodated within the current planning horizon regardless of backlog position) similarly require priority-preemption mechanisms that are absent from the current models. Addressing these factors would likely require either stochastic programming extensions or a robust optimisation framework that hedges against worst-case demand realisations.

Despite these simplifications, the thesis demonstrates that combinatorial optimization can produce substantial and statistically solid improvements in appointment-system throughput and equity over a naive FIFO discipline. The Type-Based model consistently schedules more patients within the planning horizon than FIFO across all tested instance sizes, and the deferral-penalized extension shows that the throughput gain can be achieved while simultaneously controlling the distributional fairness of patient denials. These findings are consistent with the broader appointment-scheduling literature, which has long argued that principled resource optimisation is a game-changer for healthcare delivery efficiency (Ahmadi-Javid et al. (2017); Gupta and Denton (2008)).

Looking forward, one the most promising direction for large-scale deployment is the adoption of data-driven, sequential decision-making methods that can learn scheduling policies from historical data and adapt to evolving patient populations. Reinforcement-learning approaches (in particular deep Q-networks or policy-gradient agents that treat each day's scheduling decision as a state-action pair) are natural candidates, as they can implicitly encode both throughput and fairness objectives in a reward signal without requiring an explicit MILP formulation. For very large instances where exact solvers are intractable, column-generation techniques offer a structured decomposition that exploits the block structure of the nurse-slot assignment problem, potentially reducing solution times by orders of magnitude.

Beyond algorithmic development, clinical validation is essential. A pilot study in which the optimised schedule recommendations are presented to scheduling coordinators, and their acceptance, modification, and override decisions are recorded; would provide ground-truth data on both practical feasibility and the gap between model assumptions and clinical workflow. Integration with Electronic Health Record (EHR) systems would further enable real-time data feeds (actual arrivals, cancellations, emergencies) to replace the static instance generation used here, closing the loop between the optimisation model and the live scheduling environment.

Chapter 7

Citations

7.1 References

D. R. Brenner, A. E. Poirier, R. R. Woods, L. F. Ellison, J.-M. Billette, A. A. Demers, S. X. Zhang, C. Yao, C. J. Finley, N. R. Fitzgerald, N. Saint-Jacques, L. G. Shack, D. Turner, and E. Holmes, "Projected estimates of cancer in canada in 2022," *CMAJ : Canadian Medical Association Journal*, vol. 194, pp. E601 – E607, 2022.

"National health expenditure trends," https://www.cihi.ca/en/national-health-expenditure-trends, accessed: 2023-05-01.

Z. A. Abdalkareem, A. M. Amir, M. A. Al-Betar, P. Ekhan, and A. I. Hammouri, "Healthcare scheduling in optimization context: a review," *Health and Technology*, vol. 11, pp. 445 – 469, 2021.

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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221716305239

D. Gupta and B. Denton, "Appointment scheduling in health care: Challenges and opportunities," *IIE Transactions*, vol. 40, no. 9, pp. 800–819, 2008. [Online]. Available: https://doi.org/10.1080/07408170802165880

J. Marynissen and E. Demeulemeester, "Literature review on multi-appointment scheduling problems in hospitals," *European Journal of Operational Research*, vol. 272, no. 2, pp. 407–419, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221718302108

B. P. Berg, B. T. Denton, S. A. Erdogan, T. Rohleder, T. R. Huschka, and huschka. todd, "Optimal booking and scheduling in outpatient procedure centers," *Comput. Oper. Res.*, vol. 50, pp. 24–37, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:8040611

N. T. J. Bailey, "A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting-times," *Journal of the royal statistical society series b-methodological*, vol. 14, pp. 185–199, 1952. [Online]. Available: https://api.semanticscholar.org/CorpusID:86551061

B. D. V. Lestdley, "The theory of queues with a single server," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 48, pp. 277 – 289, 1952. [Online]. Available: https://api.semanticscholar.org/CorpusID:119676458

S. C. Brailsford and J. M. H. Vissers, "Or in healthcare: A european perspective," *Eur. J. Oper. Res.*, vol. 212, pp. 223–234, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:1432279

A. Condotta and N. V. Shakhlevich, "Scheduling patient appointments via multilevel template: A case study in chemotherapy," *Operations research for health care*, vol. 3, pp. 129–144, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:40329746

M. Deceuninck, D. Fiems, and S. D. Vuyst, "Outpatient scheduling with unpunctual patients and no-shows," *Eur. J. Oper. Res.*, vol. 265, pp. 195–207, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:207693189

A. Moosavi, O. Ozturk, and J. Patrick, "Dynamic distributed ambulatory care scheduling," *Production and Operations Management*, vol. 34, no. 10, pp. 3173–3192, 2025. [Online]. Available: https://doi.org/10.1177/10591478251331143

M. Issabakhsh, S. Lee, and H. Kang, "Scheduling patient appointment in an infusion center: a mixed integer robust optimization approach," *Health Care Management Science*, vol. 24, pp. 117 – 139, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:222300944

S. Faridimehr, S. Venkatachalam, and R. B. Chinnam, "Managing access to primary care clinics using scheduling templates," *Health Care Management Science*, vol. 24, pp. 482 – 498, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:230658084

A. F. Hesaraki, N. P. Dellaert, and T. G. de Kok, "Generating outpatient chemotherapy appointment templates with balanced flowtime and makespan," *Eur. J. Oper. Res.*, vol. 275, pp. 304–318, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:59605172

Y.-L. Huang, S. Bach, and S. Looker, "Chemotherapy scheduling template development using an optimization approach," *International Journal of Health Care Quality Assurance*, vol. 32, pp. 00–00, 01 2019.

N. B. Demir, S. Gul, and M. Çelik, "A stochastic programming approach for chemotherapy appointment scheduling," *Naval Research Logistics (NRL)*, vol. 68, pp. 112 – 133, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:210838735

M. Heshmat and A. B. Eltawil, "Solving operational problems in outpatient chemotherapy clinics using mathematical programming and simulation," *Annals of Operations Research*, vol. 298, pp. 289 – 306, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:214020701

B. Liang, A. Turkcan, M. E. Ceyhan, and K. E. Stuart, "Improvement of chemotherapy patient flow and scheduling in an outpatient oncology clinic," *International Journal of Production Research*, vol. 53, pp. 7177 – 7190, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:14065659

A. Elidrissi, R. Banmansour, K. Hasani, and F. Werner, "Scheduling on parallel machines with a common server in charge of loading and unloading operations," 2023. [Online]. Available: https://arxiv.org/abs/2306.16669

C.-L. Hsu and J.-R. Liao, "Two Parallel-Machine Scheduling Problems with Function Constraint," *Discrete Dynamics in Nature and Society*, vol. 2020, p. 2717095, May 2020, publisher: Hindawi. [Online]. Available: https://doi.org/10.1155/2020/2717095

S. Kravchenko and F. Werner, "Parallel machine scheduling problems with a single server," *Mathematical and Computer Modelling*, vol. 26, no. 12, pp. 1–11, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0895717797002367

A.-L. Olteanu, M. Sevaux, and M. Ziaee, "Unrelated parallel machine scheduling with job and machine acceptance and renewable resource allocation," *Algorithms*, vol. 15, no. 11, 2022. [Online]. Available: https://www.mdpi.com/1999-4893/15/11/433

J. Magalhães-Mendes and A. B. de Almeida, "A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem," 2013.

T. Lau and E. Tsang, "Guided genetic algorithm and its application to radio link frequency assignment problems," *Constraints*, vol. 6, pp. 373–398, 01 2001.

M. Abu-Shams, S. Ramadan, S. Al-Dahidi, and A. Abdallah, "Scheduling large-size identical parallel machines with single server using a novel heuristic-guided genetic algorithm (das/ga) approach," *Processes*, vol. 10, no. 10, 2022. [Online]. Available: https://www.mdpi.com/2227-9717/10/10/2071

L. Zhang and A. Wirth, "On-line scheduling of two parallel machines with a single server," *Computers & Operations Research*, vol. 36, no. 5, pp. 1529–1553, 2009, selected papers presented at the Tenth International Symposium on Locational Decisions (ISOLDE X). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054808000361

Y. Jiang, Q. Zhang, J. Hu, J. Dong, and M. Ji, "Single-server parallel-machine scheduling with loading and unloading times," *Journal of Combinatorial Optimization*, vol. 30, no. 2, pp. 201–213, Aug. 2015. [Online]. Available: https://doi.org/10.1007/s10878-014-9727-z

S. Hahn-Goldberg, M. W. Carter, J. C. Beck, M. Trudeau, P. Sousa, and K. Beattie, "Dynamic optimization of chemotherapy outpatient scheduling with uncertainty," *Health Care Management Science*, vol. 17, pp. 379 – 392, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:13382835

Y. Hur, J. F. Bard, and D. J. Morrice, "Appointment scheduling at a multidisciplinary outpatient clinic using stochastic programming," *Naval Research Logistics (NRL)*, vol. 68, pp. 134 – 155, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:216215421

M. Heshmat, K. Nakata, and A. B. Eltawil, "Solving the patient appointment scheduling problem in outpatient chemotherapy clinics using clustering and mathematical programming," *Comput. Ind. Eng.*, vol. 124, pp. 347–358, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:52815406

M. Benzaid, N. Lahrichi, and L.-M. Rousseau, "Chemotherapy appointment scheduling and daily outpatient–nurse assignment," *Health Care Management Science*, vol. 23, pp. 34 – 50, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:58583948

B. Liang and A. Turkcan, "Acuity-based nurse assignment and patient scheduling in oncology clinics," *Health Care Management Science*, vol. 19, pp. 207–226, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:1113029

A. G. Leeftink, I. M. H. Vliegen, and E. W. Hans, "Stochastic integer programming for multi-disciplinary outpatient clinic planning," *Health Care Management Science*, vol. 22, pp. 53 – 67, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:42356962

C. Ramos, A. J. Cataldo, and J.-C. Ferrer, "Appointment and patient scheduling in chemotherapy: a case study in chilean hospitals," *Annals of Operations Research*, vol. 286, pp. 411–439, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:125752186

Y.-L. Huang, I. Sikder, and G. Xu, "Optimal override policy for chemotherapy scheduling template via mixed-integer linear programming," *Optimization Letters*, vol. 16, pp. 1549 – 1562, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:239662824

A. Agnetis, C. Bianciardi, and N. Iasparra, "Integrating lean thinking and mathematical optimization: A case study in appointment scheduling of hematological treatments," *Operations Research Perspectives*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:133017739

B. Alireza, F. John, M. Srimathy, G. Mohan, A. BetcherJeffrey, M. AltmanKristin, G. CollinsJames, H. BryceAlan, and M. Ruben, "Analysis of workflow in chemotherapy clinical practice," *International Journal of Healthcare Systems Engineering*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:239641360

M. Hadid, A. Elomri, R. Padmanabhan, L. Kerbache, O. Jouini, A. E. Omri, A. H. Nounou, and A. Hamad, "Clustering and stochastic simulation optimization for outpatient chemotherapy appointment planning and scheduling," *International Journal of Environmental Research and Public Health*, vol. 19, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:253873252

K. J. Klassen and R. Yoogalingam, "Appointment system design with interruptions and physician lateness," *International Journal of Operations & Production Management*, vol. 33, pp. 394–414, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:154108449

R. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey," *Annals of discrete mathematics*, vol. 5, pp. 287–326, 1977. [Online]. Available: https://api.semanticscholar.org/CorpusID:61033710

L. Davis, "Handbook of genetic algorithms," 1990.

"Measuring Manitoba's Progress on Wait Times - Diagnostic and Surgical Recovery Task Force — Province of Manitoba — gov.mb.ca," https://www.gov.mb.ca/health/dsrecovery/progress.html, 2025, [Accessed 01-12-2025].

"Measuring Wait Times for Other Surgeries and Procedures &x2013; Health Quality Ontario — hqontario.ca," https://www.hqontario.ca/System-Performance/Measuring-System-Performance/Measuring-Wait-Times-for-Other-Surgeries-and-Procedures, 2025, [Accessed 01-12-2025].

"Wait Times for Radiation Therapy (Percentiles) — CIHI — cihi.ca," https://www.cihi.ca/en/indicators/wait-times-for-radiation-therapy-percentiles, 2025, [Accessed 01-12-2025].

"Median waiting times for medical care, from general practitioner to treatment, in canada as of 2023, by province (in weeks waited) [graph]," 2023, accessed: 2024-09-10. [Online]. Available: https://www.statista.com/statistics/649600/medical-treatment-wait-times-canada-province/

C. Liddy, L. Cooper, G. Bellingham, T. Deyell, P. Ingelmo, I. Moroz, P. Poulin, A. Singer, G. S. Logan, R. Visca, A. Zahrai, and N. Buckley, "Patient-reported wait times and the impact of living with chronic pain on their quality of life: A waiting room survey in chronic pain clinics in ontario, manitoba, and quebec," *Canadian Journal of Pain*, vol. 8, no. 1, p. 2345612, 2024. [Online]. Available: https://doi.org/10.1080/24740527.2024.2345612

F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

A. Clark, "Rolling horizon heuristics for production planning and set-up scheduling with backlogs and error-prone demand forecasts," *Production Planning & Control - PRODUCTION PLANNING CONTROL*, vol. 16, pp. 81–97, 01 2005.

Appendix A

Base Model Run Results on IBM CPLEX

Table A.1: Base model run results - Small instances

Case No.	Objective Function	Run Time (s)
01	33	0.86
02	33	0.75
03	32	0.93
04	32	0.83
05	33	0.89
06	32	1.05
07	33	0.79
08	32	0.99
09	32	1.08
10	33	0.95
11	32	0.86
12	32	0.92
13	32	0.84
14	32	0.74
15	33	0.97
16	33	1.06
17	32	1.05
18	32	1.01
19	32	0.75
20	32	0.87
21	32	0.73
22	33	0.91
23	33	1.06
24	33	0.97
25	33	1.06
26	33	0.79
27	33	0.77
28	33	1.09
29	33	1.00
30	32	0.97
31	33	1.09
32	32	1.36
33	32	1.03
34	32	1.25
35	32	1.09
36	32	1.01
37	33	0.71
38	33	0.95
39	33	1.28
40	33	1.04
41	33	0.72
42	32	1.01
43	32	0.91
44	32	0.91
45	32	0.86
46	32	0.76
47	33	0.86
48	32	1.03
49	32	0.72
50	33	1.05

Table A.2: Base model run results - Medium instances

Case No.	Objective Function	Run Time (s)
01	164	33.13
02	117	19.92
03	235	58.95
04	114	15.81
05	114	16.82
06	238	64.02
07	117	12.63
08	238	63.05
09	240	63.84
10	164	37.70
11	164	49.48
12	164	30.60
13	159	34.14
14	116	14.15
15	234	118.59
16	238	67.52
17	236	65.83
18	229	55.44
19	118	18.45
20	160	36.52
21	114	15.50
22	117	15.17
23	237	68.06
24	229	53.06
25	229	56.95
26	117	12.21
27	116	13.76
28	244	68.37
29	164	30.07
30	164	32.14
31	244	67.25
32	241	71.62
33	239	61.33
34	241	68.06
35	239	68.38
36	238	63.64
37	118	14.59
38	227	57.43
39	241	66.47
40	230	51.69
41	117	14.94
42	237	61.58
43	164	32.49
44	234	67.20
45	164	29.91
46	114	13.65
47	164	30.88
48	238	62.98
49	115	13.09
50	238	63.02

Table A.3: Base model run results - Large instances

Case No.	Objective Function	Run Time (s)
01	281	801.85
02	281	921.39
03	260	472.41
04	0	1002.20
05	260	902.63
06	260	487.49
07	260	582.64
08	260	910.88
09	260	956.90
10	0	1120.89
11	260	812.73
12	260	1026.39
13	260	565.96
14	284	1067.80
15	280	960.82
16	260	760.86
17	284	1073.55
18	0	1160.53
19	0	1392.94
20	283	974.73
21	259	718.64
22	0	1333.77
23	260	604.98
24	0	1294.09
25	260	544.09
26	259	1025.00
27	259	592.89
28	260	569.51
29	260	464.69
30	0	1164.54
31	260	934.50
32	259	598.80
33	259	1023.57
34	258	494.24
35	259	705.50
36	0	1175.88
37	260	931.15
38	259	969.52
39	0	1311.07
40	260	536.05
41	259	679.99
42	260	576.30
43	0	1179.92
44	260	796.55
45	0	1336.24
46	259	1001.07
47	260	555.59
48	0	1308.79
49	260	576.69
50	260	955.61

Appendix B

Type-based Model Run Results on IBM CPLEX

Table B.1: Type-based model run results - Small instances

Case No.	Objective Function	Run Time (s)
01	33	0.05
02	33	0.05
03	32	0.05
04	33	0.06
05	32	0.05
06	32	0.10
07	32	0.05
08	32	0.05
09	32	0.05
10	33	0.05
11	33	0.05
12	32	0.05
13	32	0.05
14	32	0.06
15	33	0.05
16	33	0.05
17	32	0.05
18	32	0.05
19	32	0.05
20	33	0.06
21	32	0.05
22	33	0.06
23	33	0.05
24	33	0.05
25	32	0.05
26	32	0.05
27	33	0.05
28	33	0.10
29	32	0.05
30	32	0.05
31	33	0.05
32	32	0.05
33	32	0.05
34	33	0.06
35	32	0.05
36	32	0.05
37	33	0.05
38	33	0.05
39	32	0.05
40	32	0.05
41	33	0.05
42	32	0.05
43	32	0.05
44	33	0.05
45	33	0.06
46	33	0.06
47	33	0.05
48	32	0.05
49	32	0.05
50	33	0.05

Table B.2: Type-based model run results - Medium instances

Case No.	Objective Function	Run Time (s)
01	164	0.55
02	118	0.18
03	243	0.50
04	118	0.16
05	116	0.17
06	246	0.67
07	118	0.16
08	241	0.69
09	242	0.68
10	156	0.29
11	156	0.36
12	158	0.28
13	160	0.50
14	118	0.17
15	232	0.53
16	245	0.70
17	245	0.69
18	240	0.68
19	117	0.18
20	156	0.35
21	118	0.17
22	118	0.17
23	246	0.69
24	229	0.53
25	228	0.49
26	117	2.09
27	114	0.18
28	246	0.63
29	159	0.35
30	156	0.28
31	248	0.67
32	249	0.64
33	246	0.86
34	246	0.66
35	242	0.73
36	239	4.61
37	118	0.17
38	229	0.50
39	245	0.67
40	234	0.68
41	117	0.17
42	242	0.67
43	161	0.35
44	230	0.53
45	157	0.28
46	118	0.16
47	158	0.28
48	242	0.66
49	117	0.17
50	247	0.67

Table B.3: Type-based model run results - Large instances

Case No.	Objective Function	Run Time (s)
01	285	2.03
02	281	2.55
03	260	1.50
04	284	3.04
05	260	97.64
06	260	1.60
07	260	1.54
08	260	2.09
09	260	2.19
10	301	2.64
11	260	2.02
12	260	2.17
13	260	1.59
14	285	2.22
15	282	2.53
16	260	1.73
17	285	2.32
18	281	3.11
19	299	2.80
20	283	2.58
21	260	1.58
22	299	2.89
23	260	2.01
24	286	3.26
25	260	1.59
26	260	2.10
27	260	1.57
28	260	1.52
29	260	1.56
30	286	3.24
31	260	2.05
32	260	1.56
33	260	2.10
34	260	1.56
35	260	1.57
36	287	3.30
37	260	2.07
38	260	1.99
39	283	3.28
40	260	1.63
41	260	1.58
42	260	1.57
43	300	3.19
44	260	1.97
45	304	3.27
46	260	2.24
47	260	1.55
48	300	3.16
49	260	1.56
50	260	2.06

Appendix C

Base Model Run Results with Rolling Horizon Heuristic

Table C.1: RH model run results - Medium instances

Case No.	Objective Function	Run Time (s)
01	158	0.45
02	118	0.19
03	236	0.48
04	118	0.22
05	116	0.15
06	240	0.45
07	118	0.19
08	236	0.49
09	237	0.43
10	158	0.34
11	158	0.41
12	158	0.32
13	158	0.28
14	116	0.16
15	232	1.55
16	237	0.42
17	236	0.49
18	228	0.37
19	118	0.18
20	157	0.33
21	118	0.18
22	118	0.17
23	238	0.45
24	226	0.44
25	226	0.39
26	118	0.18
27	117	0.22
28	237	0.45
29	158	0.30
30	158	0.37
31	243	0.58
32	242	0.50
33	237	0.43
34	238	0.43
35	237	0.44
36	236	0.40
37	118	0.18
38	226	0.44
39	241	0.58
40	226	0.48
41	118	0.17
42	237	0.40
43	158	0.34
44	226	0.42
45	158	0.37
46	118	0.18
47	161	0.41
48	236	0.39
49	118	0.20
50	239	0.50

Table C.2: RH model run results - Large instances

Case No.	Objective Function	Run Time (s)
01	275	2.03
02	272	2.02
03	260	1.48
04	281	2.67
05	260	1.90
06	260	1.44
07	260	1.44
08	260	1.90
09	260	1.95
10	292	3.26
11	260	1.90
12	260	2.00
13	260	1.48
14	275	1.96
15	272	2.03
16	260	1.49
17	275	1.93
18	281	2.84
19	290	2.49
20	272	2.03
21	257	1.54
22	290	2.60
23	260	1.46
24	281	2.61
25	260	1.46
26	260	1.85
27	260	1.48
28	260	1.45
29	260	1.51
30	279	2.64
31	260	1.95
32	260	1.45
33	260	1.92
34	260	1.49
35	260	1.48
36	281	2.62
37	260	1.91
38	260	1.91
39	281	2.66
40	260	1.48
41	260	1.47
42	260	1.44
43	292	2.50
44	260	1.88
45	295	2.51
46	260	2.00
47	260	1.45
48	291	2.51
49	260	1.46
50	260	1.96

Appendix D

Genetic Algorithm Run on Base Problem

Table D.1: Genetic Algorithm model run results - Small instances

Case No.	Objective Function	Run Time (s)
01	32	1.01
02	32	0.88
03	32	1.17
04	32	1.01
05	32	0.90
06	32	1.36
07	32	0.76
08	32	1.31
09	32	1.39
10	32	1.07
11	32	1.11
12	31	1.09
13	32	0.97
14	31	0.78
15	32	1.17
16	32	1.32
17	32	1.38
18	33	1.20
19	32	0.88
20	32	0.95
21	33	0.86
22	32	0.88
23	32	1.41
24	32	1.10
25	32	1.08
26	33	0.78
27	32	0.77
28	33	1.39
29	32	1.12
30	33	1.06
31	32	1.39
32	32	1.46
33	32	1.29
34	32	1.47
35	32	1.25
36	32	1.17
37	33	0.80
38	32	1.13
39	32	1.40
40	32	1.16
41	32	0.76
42	32	1.18
43	32	1.09
44	32	1.13
45	32	1.02
46	32	0.87
47	32	0.99
48	32	1.32
49	32	0.78
50	32	1.32

Table D.2: Genetic Algorithm model run results - Medium instances

Case No.	Objective Function	Run Time (s)
01	142	15.08
02	110	9.51
03	220	20.57
04	111	9.70
05	112	9.36
06	222	22.52
07	111	9.25
08	220	22.04
09	219	23.50
10	141	15.59
11	139	14.97
12	140	15.08
13	138	15.56
14	108	8.74
15	210	15.22
16	221	21.52
17	216	22.79
18	209	15.45
19	111	9.93
20	141	15.07
21	110	9.43
22	110	9.18
23	222	22.99
24	209	14.33
25	207	15.14
26	111	9.73
27	113	8.78
28	223	22.45
29	143	15.03
30	143	15.78
31	227	22.66
32	224	23.61
33	221	21.74
34	222	23.33
35	220	21.17
36	218	21.49
37	111	8.68
38	204	14.98
39	221	22.25
40	208	15.09
41	113	9.10
42	219	20.86
43	136	14.99
44	207	15.01
45	139	14.80
46	111	10.35
47	143	15.02
48	220	23.36
49	110	8.80
50	219	21.57

Table D.3: Genetic Algorithm model run results - Large instances

Case No.	Objective Function	Run Time (s)
01	257	150.03
02	256	143.65
03	228	120.32
04	263	209.66
05	253	147.70
06	225	120.47
07	227	118.44
08	255	159.16
09	254	169.55
10	263	250.19
11	246	151.06
12	256	145.05
13	228	115.53
14	259	158.19
15	257	160.20
16	238	104.37
17	261	904.33
18	262	1828.83
19	263	252.66
20	261	131.27
21	226	103.77
22	262	258.65
23	225	108.74
24	262	233.24
25	231	117.41
26	247	165.30
27	233	125.54
28	226	147.54
29	236	129.65
30	261	1204.13
31	254	143.82
32	228	113.46
33	255	136.37
34	234	109.31
35	231	510.23
36	262	207.56
37	253	318.63
38	252	580.08
39	262	2104.21
40	234	244.24
41	238	109.88
42	225	112.49
43	262	233.49
44	248	215.35
45	264	1888.86
46	254	163.90
47	226	112.87
48	262	1281.73
49	230	941.74
50	248	998.57