Advanced Stochastic Programming and Machine Learning Models for Healthcare Planning, Scheduling, and Prediction Problems

Soheyl Khalilpourazari

A Thesis

in

The Department

of

Department of Mechanical, Industrial & Aerospace Engineering (MIAE)

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Industrial Engineering) at

Concordia University

Montréal, Québec, Canada

May 2024

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Soheyl Khalilpourazari**

Entitled: **Advanced Stochastic Programming and Machine Learning Models for Healthcare Planning, Scheduling, and Prediction Problems**

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

Doctor of Philosophy (Industrial Engineering)

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

Signed by the Final Examining Committee:

_____ Chair
Dr. Andrea Schiffauerova

_____ External Examiner
Dr. Hamed Samarghandi

_____ External to Program
Dr. Farnoosh Naderkhani

_____ Examiner
Dr. Mingyuan Chen

_____ Examiner
Dr. Onur Kuzgunkaya

_____ Supervisor
Dr. Hossein Hashemi Doulabi

Approved by _____
Dr. Muthukumaran Packirisamy
Graduate Program Director

_____ 2024

_____
Dr. Mourad Debbabi
Dean of Faculty of Engineering and Computer Science

Abstract

Advanced Stochastic Programming and Machine Learning Models for Healthcare Planning, Scheduling, and Prediction Problems

Soheyl Khalilpourazari, Ph.D.

Concordia University, 2024

The increasing demand for global healthcare systems highlights the urgent need for innovative solutions. In response to this challenge, we uses advanced Stochastic Programming and Machine Learning methods to introduce significant improvements in appointment scheduling, operating room planning, and modeling and prediction of the COVID-19 pandemic.

In the first paper, we study the healthcare appointment scheduling problem. The main challenges in appointment scheduling are uncertainties in no-shows, unpunctuality, and service times. We propose a novel stochastic programming model that captures an exponential number of scenarios using a pseudo-polynomial number of variables and constraints without relying on sampling methods. The presented methodology is exact. We show that the generated schedules reduce total costs by 34% on average by incorporating patient-dependent service times, 12% by considering patient-and-time-dependent unpunctuality, and 67% by integrating patient-and-time-dependent no-shows. In addition, we show that personalized reminders have the potential to reduce total costs by 23%.

In the second paper, we study a stochastic operating room planning problem. The unpredictability of surgical durations poses a considerable challenge to efficient OR planning. Existing models often overlook this source of uncertainty. This paper introduces a novel stochastic programming model that effectively manages the uncertainty in surgical times. This model advances the literature by capturing an exponential number of scenarios in a weekly operating room planning problem without sampling, simplifications, or approximations. The results of the computational experiments revealed that our model obtains feasible solutions with an average optimality gap of 0.78% for

instances with 80 surgeries and 1.48E+64 scenarios.

In the third, fourth and fifth papers, we focus on modeling and prediction of the COVID-19 pandemic and aim at developing methodologies that inform and guide public health decisions. In these three papers, we proposed a hybrid reinforcement learning based algorithm as well as two other evolutionary computation based algorithms to forecast the spread of the COVID-19 pandemic. By applying these methods to real-world data from Canada, Quebec, Ontario, France and the U.S., we aim to offer insights into effective pandemic response strategies. We predict the pandemic trajectory as well as the number of different cases with high accuracy.

Acknowledgments

Embarking on this Ph.D. journey has been an extraordinary adventure, illuminated by the support and encouragement of many who have been pillars of strength and wisdom. Their contributions have shaped this journey into a memorable and enriching experience. My heartfelt gratitude goes out to everyone who played a part in this chapter of my life.

Foremost, I extend my deepest appreciation to my supervisor, Dr. Hossein Hashemi Doulabi, for his unwavering commitment to excellence. Dr. Doulabi's patience, expertise, and dedication have been a guiding light throughout this journey. His consistent support and encouragement, rooted in a profound understanding of our field, have been pivotal in my growth as a researcher and individual.

My journey would have been significantly more challenging without the love and support of my girlfriend, Yasaman. Her kindness, supportiveness, and the countless sacrifices she made have been the bedrock of my perseverance. Yasaman, you have been more than a partner; you have been an advisor, mentor, and the greatest supporter, showing unwavering belief in me every step of the way.

The foundational support of my family cannot be overstated. My father, Abdolaziz, and my mother, Aghdas, have been my constant sources of love and encouragement. Their belief in my abilities and their unconditional support have been crucial in overcoming challenges. To my brothers, Saman and Sasan, thank you for your understanding, patience, and the moments of light-hearted relief amidst the pressures of academic life.

I must also express my deep gratitude to my sister-in-law, Nazila. Your encouragement and unwavering support have added another layer of strength to my pursuit. Your presence and belief in me have been sources of comfort and motivation.

Additionally, I want to express my gratitude to my friends - Maryam, Syed, Masoud, and Shabnam. Your friendship, advice, and unwavering support have been invaluable. Whether it was offering a listening ear, sharing insights, or simply being there during times of need, your contributions have been a significant part of my journey.

To Dr. Doulabi, Yasaman, my family, Nazila, and my friends: your influence on my Ph.D. journey has been profound. The collective support, encouragement, and belief from each one of you have not only propelled me through this academic endeavor but have also left lasting imprints on my personal growth. I am eternally grateful for your roles in my life and look forward to carrying forward the lessons learned and the strength gained from your support into future endeavors.

I would like to also express my profound gratitude for the financial supports I received during my Ph.D. studies, which were instrumental in my academic and research endeavors. My heartfelt thanks go to:

The Fonds de Recherche du Québec–Nature et Technologies (FRQNT) for the FRQNT B2X scholarship (Ranked #1 among 15 applicants in Quebec in Engineering Mathematics and Operations Research).

CIRRELT, Polytechnique Montreal, Canada, for their continuous encouragement and recognition of my work through three Doctoral Excellence awards.

Concordia University, Montreal, Canada, for their extensive financial support and numerous awards that significantly contributed to my studies, including Concordia University Stand-Out Graduate Research Award and Concordia University Doctoral Graduate Fellowship.

These scholarships and awards not only provided financial relief but also motivated me to pursue excellence in my research endeavors. I am immensely grateful to each of these institutions for their support and belief in my academic pursuits.

To my beloved family

Contribution of Authors

This dissertation is presented under the *"manuscript-based format"*. It contains five articles where four of them have been published and one is submitted. The first article "Stochastic appointment scheduling with patient-and-time-dependent probability distributions" is submitted to *Manufacturing & Service Operations Management* in March 2024. The second manuscript "Stochastic weekly operating room planning with an exponential number of scenarios" is published in *Annals of Operations Research* in 2022. The Third manuscript "Designing a hybrid reinforcement learning based algorithm with application in prediction of the COVID-19 pandemic in Quebec" is published in *Annals of Operations Research* in 2020. The Fourth manuscript "Robust Modeling and Prediction of the COVID-19 Pandemic in Canada" is published in *International Journal of Production Research* in 2021. The Fifth manuscript "Gradient-Based Grey Wolf Optimizer with Gaussian Walk: Application in Modelling and Prediction of the COVID-19 Pandemic" is published in *Expert Systems with Applications* in 2021. It is worth mentioning that the second to fifth papers have received a total of 203 citations as of March 2024.

The first four manuscripts are co-authored with Dr. Hossein Hashemi doulabi who established research guidelines and reviewed the papers before submission. The last manuscript also includes Prof. Aybike Özyüksel Çiftçioğlu and Prof. Gerhard-Wilhelm Weber who contributed to the editing of the final paper. The author of this thesis acted as the principal researcher with the corresponding duties such as the development of initial ideas, formulations, and algorithms as well as the coding of solution methods and analysis of computational results along with writing the first drafts.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In today's global landscape, healthcare systems are faced with unprecedented challenges. The emergence of health crises, such as the COVID-19 pandemic, coupled with the increasing demands on healthcare infrastructures highlights a pressing need. The main challenge is to offer high-quality healthcare while improving healthcare operations without incurring unsustainable costs. Increasing efficiency, accessibility, and maintaining high quality, present a complex issue that requires creative and new solutions. Healthcare systems that are inefficient, inaccessible, or compromise quality can lead to severe consequences for public health, economic stability, and social equity. Inefficiencies lead to wasted resources and longer wait times that diminish the system's responsiveness to public health emergencies. Limited accessibility increases health disparities, which leaves vulnerable populations without necessary care. Compromises on the quality of care can result in adverse health outcomes that erode public trust in healthcare systems. Therefore, optimizing healthcare delivery is not just a matter of administrative or operational concern but a vital issue that impacts society.

To address these challenges, this dissertation proposes different methods that use advanced Stochastic Programming (SP) and Machine Learning (ML). The aim is to use these advanced methodologies to introduce significant improvements in critical aspects of healthcare delivery including appointment scheduling, operating room planning, and modeling and prediction of the COVID-19 pandemic. By integrating stochastic programming and machine learning this study aims to pave new paths in healthcare optimization and address major challenges healthcare facilities face today.

In the following, initially, we start with an introduction to appointment scheduling and operating room planning problems, which are the focus of the first and second papers that are presented in Chapters 2 and 3, respectively. Subsequently, we provide an introduction to modeling and prediction of the COVID-19 pandemic that covers the subject of the third, fourth, and fifth papers which are presented in Chapters 4, 5, and 6.

1.1 Appointment Scheduling

Appointment Scheduling (AS) focuses on assigning appointment times to a fixed number of patients to provide inclusive, affordable, and timely access to high-quality healthcare, which has become a critical concern in today's society (Dai and Tayur, 2020; Samorani et al., 2022). Rapid developments in the healthcare sector have initiated a transition towards patient-centered care, a model that emphasizes affordable, cost-effective, and quality health services prioritizing patient experience. This is important since the waiting time for healthcare services in Canada are long. Thus, it has become crucial for healthcare providers to continually improve and maintain the highest level of service quality.

The main challenge in AS is that there are three main sources of uncertainties including patient no-shows and unpunctuality, and variations in service times that negatively impact clinics' operations, and reduce clinics' efficiency and revenue. Service time durations in appointment scheduling are uncertain due to the variability in patients' health conditions and the nature of their visits. For instance, new patients might require longer consultations for a comprehensive health assessment, whereas return patients could be visiting for a quick check-up or a complex procedure based on their ongoing treatment. In addition, patient no-show and unpunctuality refer to cases where a patient does not show up for his/her scheduled appointment, or if he/she shows up for the appointment, he/she is not punctual. Moreover, the uncertainty in patient unpunctuality and no-shows is a patient-and-time-dependent factor as their likelihood can vary based on the time of the appointment and individual patient history, such as their past behavior regarding attendance, traffic, and weather conditions.

Traditional models often fall short in capturing the full spectrum of these uncertainties that lead

to inefficiencies and diminished service quality. In the information era and with more data becoming available, incorporating patient-dependent and time-dependent considerations into appointment scheduling models is essential. Using historical data, we can have a deeper understanding of patient behavior, which enables us to generate more accurate schedules. In addition, the dynamic nature of these behaviors, influenced by the time of appointments, necessitates models that can incorporate patient-and-time-dependent considerations. To the best of our knowledge, no study in the literature has simultaneously addressed all these sources of uncertainties with patient-and-time-dependent consideration, which is mainly due to modeling complexity. Addressing this gap, this dissertation introduces a novel stochastic programming model that, for the first time, accurately reflects patient-dependent service times, and patient-and-time-dependent unpunctuality and no-shows simultaneously. This model's innovation lies in its ability to capture an exponential number of scenarios and offer a more detailed and effective approach to appointment scheduling. This paper, presented in Chapter 2, is submitted to *Manufacturing & Service Operations Management*.

1.2 Operating Room Planning

Operating room (OR) planning emerges as a critical area due to its significant impact on hospitals' operational efficiency and finances. Operating rooms contribute substantially to hospital revenues; therefore, optimizing their use is essential for improving service quality and reducing patient wait times. The main challenge in OR optimization lies in the unpredictable nature of surgical times which makes planning a complex task. Research in this area falls into three primary categories based on the approach to uncertainty: deterministic models, stochastic programming models, and robust optimization models. Deterministic models often overlook the variability in surgical times which potentially leads to infeasible plans or unexpected costs. Stochastic programming models consider a limited number of uncertain scenarios for the duration of surgical times, which offer more realistic solutions but can become hard to solve with an increase in the number of scenarios. Robust optimization models consider that the duration of surgeries belongs to an uncertainty set and focus on optimizing the worst-case scenario, which might result in overly conservative plans. We will provide more detail on these approaches in section 3.1.

This dissertation introduces an innovative stochastic programming model for OR planning that addresses the limitations of existing methodologies. Our proposed stochastic programming model can capture an exponential number of scenarios without sampling which is a significant advancement in OR optimization. We demonstrate the effectiveness of our model through extensive computational experiments that highlight its capability in solving large size instances efficiently. This paper, presented in Chapter 3, is published in *Annals of Operations Research*.

1.3 Modeling and Prediction of the COVID-19 pandemic

The unprecedented global impact of pandemics in the past and especially the COVID-19 pandemic in recent years have highlighted the importance of an accurate modeling and prediction approach for future outbreaks. During the COVID-19 pandemic, although most COVID patients recovered independently, some other patients with severe and critical symptoms needed hospitalization (Public Health Agency of Canada, 2020). However, given the capacity limitations of the healthcare system and limited resources, it was not feasible to admit all patients to hospitals (Public Health Agency of Canada, 2020; Government of Canada, 2023). To be prepared for similar problems in the future, it is crucial to develop new methods to model and predict pandemics' growth and determine when to apply public health actions to prevent the spread of viruses/bacteria. Such approaches will also provide policymakers with valuable information to optimize resource allocation in the healthcare system to ensure proper patient care and equipment availability.

Recently, the SIDARTHE model proposed by Giordano et al. (2020) in *Nature Medicine*, shows promising advances in pandemic modeling and prediction. However, the authors have highlighted that the model's complexity presents challenges in solving the set of differential equations (Giordano et al., 2020). This model results in computing vital epidemiological parameters such as reproduction rate that shows how fast a disease is spreading in the population. By solving the SIDARTHE model we can estimate the future number of recovered, death, life-threatening, and infected cases, and values of the parameters associated with the pandemic dynamics.

In this line of research, we have introduced several algorithms to find the optimal parameters for the SIDARTHE model. In the first paper, we propose a new hybrid algorithm by combining

the learning power of reinforcement learning with the problem-solving capabilities of evolutionary computation to solve the SIDARTHE model. Using reinforcement learning as the algorithm's core and various evolutionary computation operators as updating mechanisms, our approach has been able to efficiently solve the SIDARTHE model. In the second paper, we propose a fractal-based algorithm to solve the problem and use Design on Experiments (DOE) to tune the parameters of the algorithm to enhance its performance. In the third paper, we propose a gradient-enhanced evolutionary algorithm that uses the gradient to better guide the search in the solution space to solve the SIDARTHE model more efficiently. These approaches not only facilitate the efficient resolution of complex differential equations associated with pandemic modeling but also help policymakers and healthcare professionals with the tools needed to anticipate and mitigate the impacts of future pandemics. We have implemented our algorithms to real data from Canada, Quebec, Ontario, and the U.S. and have provided several managerial insights about future trends of the COVID-19 pandemic. The results of this line of research are published in three papers in *Annals of Operations Research*, *International Journal of Production Research*, and *Expert Systems with Applications*. These papers are presented in Chapters 4, 5, and 6, respectively.

Chapter 2

Stochastic appointment scheduling with patient-and-time-dependent probability distributions [1]

Abstract

Problem definition: Patient scheduling requires balancing provider idle time, overtime, and patient waiting time. Yet, some factors such as no-shows, unpunctuality, and uncertain service times make scheduling a challenging task resulting in high clinic costs and low patient satisfaction. Addressing these factors in appointment scheduling is a difficult task since they follow patient-and-time-dependent probability distributions. Current models often fail to incorporate patient-and-time-dependent probability distributions and assume identical distributions for all patients in all time slots. This is mainly due to increased modeling and solution complexity which can lead to inefficiencies in scheduling, increased clinic costs, and reduced patient satisfaction. **Methodology:** We propose a novel stochastic programming model that captures an exponential number of scenarios using a pseudo-polynomial number of variables and constraints without relying on sampling methods. We incorporate patient-dependent probability distributions for service times, and patient-and-time-dependent distributions for no-shows and arrival times. To enhance the model,

[1] This paper is submitted to *Manufacturing & Service Operations Management*

we explore the impact of personalized reminders on no-show rates and schedule effectiveness. We investigate strategies to minimize the negative impact of no-shows on patient waiting times and the provider idle time. **Results:** The results demonstrate that our proposed model is able to optimally solve large-scale instances with up to 1.97E+8890 scenarios in a reasonable computational time. We demonstrate that the generated schedules reduce total costs by 34% on average by incorporating patient-dependent service times, 12% by considering patient-and-time-dependent unpunctuality, and 67% by integrating patient-and-time-dependent no-shows. In addition, we show that personalized reminders has the potential to reduce total costs by 23%. **Managerial insights:** Our approach has a significant potential to improve the efficiency of the healthcare services. The generated schedules enhance patient satisfaction, improve clinic operations, and resource utilization by reducing patient waiting time, provider idle time, and overtime costs. Furthermore, our findings highlight the potential of personalized communication strategies in enhancing patient attendance and reducing no-show rates.

2.1 Introduction

Ensuring inclusive, affordable, and timely access to high-quality healthcare has become a critical concern in today's society. To achieve this, continuous improvement in healthcare is essential, particularly in enhancing the patient experience. This includes not only the direct medical care received but also the overall journey patients navigate through the healthcare system (Dai and Tayur, 2020; Samorani et al., 2022). In this context, Appointment Scheduling (AS) is an essential tool for reducing patient wait times, provider idle times, and clinic overtime. However, the effectiveness of appointment scheduling is frequently undermined by uncertainties in service times, patient unpunctuality, and patient no-shows.

The existing literature on the variability in service times has often treated service durations as identically distributed for all patients (Hassin and Mendel, 2008; Kaandorp and Koole, 2007; Li et al., 2016; Tai-Seale et al., 2007). This assumption overlooks the real-world scenarios where service times vary due to multiple patient factors. Although classifications such as inpatient vs. outpatient (Sickinger and Kolisch, 2009) and new vs. return patients (Cayirli et al., 2008) have been

attempted, they fail to fully capture the variability in service times. The primary reason for this is the inherent complexity and diversity of patient needs. Inpatient and outpatient classifications, while useful, do not account for the wide range of medical conditions and associated treatments that can significantly affect service times. Similarly, categorizing patients as new or return offers limited insight into the actual time required for their care. New patients might need extensive initial consultations and evaluations, but this is not always the case. Conversely, return patients could require simple follow-up visits or complex ongoing treatments. Thus, these classifications overlook other crucial factors such as the health status of the patient. All these variables contribute to the variability in service times which makes these classifications insufficient to accurately reflect service time for each patient. In recent research, Salzarulo et al. (2016) showed that a scheduling methodology based on patient-dependent service times results in more accurate schedules. Despite the effectiveness of their approach regarding patient-dependent service times, the authors utilized an identical method to model patient no-shows and unpunctuality, which did not include a patient-and-time-dependent analysis of patient behaviors. This generalized approach potentially overlooks the diverse nature of patient attendance patterns. Secondly, their solution methodology is an approximation rather than an exact approach. This aspect of their research, while practical for certain scenarios, may not properly reflect the complexities present in real-world healthcare environments. The approximation approach could potentially limit the model's accuracy and adaptability in more complex or variable clinical settings. Our model, unlike Salzarulo et al. (2016) can handle patient-dependent stochastic service times, and patient-and-time-dependent unpunctuality and no-shows in a more comprehensive manner. Our model captures an exponential number of scenarios which is achieved through a novel stochastic programming model that has a significantly fewer variables and constraints. The presented methodology is exact thus leading to a more precise and efficient optimization of appointment schedules.

Patient unpunctuality is another source of uncertainty that negatively impacts operations at most healthcare appointment systems. This factor has been mostly oversimplified in existing models to a random variable with little to no consideration of patients' historical behavior or appointment time (Jiang et al., 2019; Zacharias and Yunes, 2020). Our proposed model addresses this gap by incorporating a patient-and-time-dependent level of unpunctuality. We consider several factors such

as individual behavior and external conditions such as traffic and weather to generate more accurate schedules to improve healthcare services.

Another source of uncertainty that disrupts clinics' operations is no-shows (i.e., a patient not showing up for an appointment.). Pesata et al. (1999) showed that 14,000 missed appointments in a children's hospital in a metropolitan area with a population of 220,000 in a midwestern state in a year resulted in a loss of more than one million dollars. Despite the limitations, the existing literature has often addressed no-shows as an identical probability distribution for all patients (Luo et al., 2012; Daggy et al., 2010; Bennett and Baxley, 2009). In the literature, there is a noticeable lack of focus on the patient-and-time-dependent variations in no-show rates. This oversight represents a significant gap in research highlighting the need for more comprehensive models that accurately reflect the dynamic nature of patients' behavior in healthcare scheduling. Our research aims to overcome this limitation by integrating both patient and time dependency of no-shows into our approach. We introduce a novel model that achieves more accurate schedules and results in reduced waiting times, overtime, and higher provider utilization.

One of the main reasons that makes current models in appointment scheduling less efficient is that they often fail to consider patient-and-time-specific factors due to modeling complexity (Li et al., 2019; Kong et al., 2020). To address this gap, our research presents an innovative stochastic programming approach to the problem. Our model uses a pseudo-polynomial number of variables and constraints that are independent of the exponential number of stochastic scenarios. This enables us to simultaneously consider patient-dependent stochastic service times, and patient-and-time-dependent unpunctuality and no-shows. Our approach significantly improves healthcare providers' resource utilization and enhances patient satisfaction by efficiently handling the complex and unpredictable variables in healthcare scheduling.

The other important distinction between this paper and the literature is that the proposed approach considers a case when patients inform the clinic about their availability/unavailability by responding to reminders. This feature allows the clinics to explore new ways to minimize the negative impact of no-shows on waiting times and provider idle time. We examine the cost-effectiveness and potential benefits of implementing our new methodology in clinics and show that the generated schedules reduce total costs by 34% on average by incorporating patient-dependent service

times, and 12% and 67% by considering patient-and-time-dependent unpunctuality and no-shows, respectively. Furthermore, we show that considering incentives to encourage patients to respond to reminders and inform the clinic about their no-show has the potential to reduce the total costs further by 23%. Our research, in addition to the technical contributions, also provides several managerial insights that help healthcare providers improve their services and stay competitive in today's patient-centered healthcare environment. This method will assist clinics in increasing patient satisfaction which will potentially lead to lower patient turnover and higher patient retention rates which are essential for healthcare providers in the competitive market.

The paper is structured as follows: In Section 2.2 we review the literature and identify research gaps. In Section 2.3 we define the problem and introduce a novel stochastic programming model for appointment scheduling considering uncertain service times, no-shows, and unpunctuality. In Section 2.4 , we present computational results and extensive experiments to evaluate the efficiency of our model. This section also studies the impacts of various factors on clinic efficiency. In Section 2.5 we conclude the paper and provide future research avenues.

2.2 Literature Review

Our research draws on four streams of research for addressing no-shows, unpunctuality and variable service times including (i) deterministic models, (ii) stochastic optimization models including stochastic programming and dynamic programming, (iii) robust optimization, and (iiii) queuing theory and simulation approaches.

The extensive literature on deterministic models assumes that the primary parameters of the appointment scheduling models are known and remain unchanged throughout the scheduling horizon. However, these approaches fail to account for potential fluctuations in the key parameters. Consequently, any changes in these parameters will make the solution inapplicable or cause significant costs to the clinic.

Stochastic programming models in the literature explore the modeling of appointment scheduling under uncertain conditions with known probability distributions and study the effects of no-shows, random service times, and unpunctuality. Jiang et al. (2019) study outpatient scheduling

and propose a stochastic programming model for the problem. The authors use a Benders Decomposition within the framework of a sample average approximation algorithm to solve the model. They consider uncertainty in the service time and unpunctuality probabilities (identical distributions for all patients). Zhan et al. (2021) develop a stochastic programming model for an integrated routing and appointment scheduling problem. The research consider stochastic service times and generate 1,000 scenarios and solve the model using an L-shape algorithm within the sample average approximation. LaGanga and Lawrence (2012) propose a heuristic to solve an appointment scheduling problem considering no-shows that could obtain near-optimal overbooked schedules to minimize the effects of no-shows. Erdogan and Denton (2013) study an appointment scheduling problem considering no-shows and present two stochastic linear programming models for the problem. The first two-stage model incorporate no-shows while the second stochastic model focus on assigning appointments to patients dynamically over the planning horizon. They generate 10,000 random scenarios and solve the models using a decomposition-based algorithm. Shehadeh et al. (2021) formulate a multistage stochastic mixed integer program for appointment scheduling problem with patient-dependent arrival times and service times. The authors use a Mont Carlo approach to generate 200 scenarios and evaluate the performance of the model.

Dynamic programming is also another common approach to model and solve appointment scheduling problems. Liu et al. (2010) model appointment scheduling with no-shows as a Markov Decision Process (MDP) and propose heuristics to solve it. Feldman et al. (2014) study the appointment scheduling problem and consider patient preference and no-shows. The authors present two models including a static and a dynamic one. In the static model, different subsets of appointment days is offered to the patient while ignoring the current state of booked appointments. On the other hand, the dynamic model uses the current state of booked appointments to offer the patient a set of appointment days. Due to the complexity of solving the problem using dynamic programming, they use an approximation approach instead to solve the problem. Soltani et al. (2019) propose a discrete-time Markov chain process for appointment scheduling and address stochastic service times and patient no-shows for multiple-provider systems. The objective function focuses on minimizing patient waiting time, and provider idle time and overtime. Their approach use a machine learning approach to capture patterns to guide a heuristic algorithm toward near-optimal solutions. Diamant

et al. (2018) study the appointment scheduling problem with no-shows and formulate the problem as a Markov decision process. They solve the model using an approximate dynamic programming algorithm.

Some research in the literature has used queuing theory to model appointment scheduling. Liu (2016) propose an M/M/1/K queuing model for choosing appointment scheduling window (i.e. patients cannot make appointments beyond a given day from the day when they make appointments) with patient-specific no-show probabilities. Luo et al. (2019) propose an M/M/1/N queuing model to determine the optimal appointment scheduling window to minimize no-shows. Zacharias and Yunes (2020) study the appointment scheduling problem with time-dependent no-shows, unpunctuality, and stochastic service times. The authors propose a single-server queueing model to determine the clinic's workload over time and prove that the objective function is supermodular.

Simulation approaches are also commonly used to address appointment scheduling problems in the literature. Cayirli et al. (2012) introduce a "Dome" appointment rule for appointment scheduling by creating a planning constant, that controls the time intervals between appointments to denote various appointment rules, considering no-shows and uncertain service times. To determine the planning constant, the authors use a nonlinear regression and simulation and create a procedure to minimize the effect of no-shows by adjusting the mean and standard deviation of service times.

Robust optimization is also widely used to handle uncertainties in the appointment scheduling problem that focuses on enhancing scheduling robustness against worst-case scenarios. Mandelbaum et al. (2020) propose a data-driven robust optimization approach for an appointment scheduling problem with uncertain service durations and punctuality. They use a bootstrapping procedure to generate service durations based on observed unpunctuality and service durations in the data. Kong et al. (2020) present a distributionally robust approach for an appointment scheduling problem with random service times and time-dependent no-shows to minimize the worst-case total expected costs. Jiang et al. (2017) propose a distributionally robust optimization for an appointment scheduling problem under no-shows and consider the worst-case expectation/conditional value-at-risk cost. They formulate the problem as a mixed integer nonlinear programming model and solve it using a completely positive decomposition algorithm that use approximation heuristics during the solution process.

Despite the significant advancements offered by these studies, several studies have applied identical probability distributions to patient no-shows and unpunctuality while neglecting the critical aspect of service time variability linked to patient health states. The closest works to our research are the papers that consider an individualized approach toward appointment scheduling. Salzarulo et al. (2016) consider service time as a function of patient health conditions. While their methodology offers a step towards personalized scheduling, it does not fully capture the patient-and-time dependency in unpunctuality and no-shows. In another research, Li et al. (2019) propose a Bayesian nested logit model and provide an estimation of the individual no-show probabilities of patients. In a case study and using a simulation approach, they provide an approximation solution and show that their model improves the clinic's profit, yet their approach may lack the consideration of variations in service times and unpunctuality. The lack of an individualized approach toward appointment scheduling is also evident in Kong et al. (2020) who consider random service times and time-dependent no-shows.

In real-world appointment scheduling problems, service times are patient-dependent, and patient unpunctuality and no-shows are patient-and-time-dependent. Many researchers have studied appointment scheduling with some of these complicating criteria and details separately. However, in none of the studies above, we observe all uncertainty factors with explained details simultaneously (Kong et al., 2020; Salzarulo et al., 2016; Li et al., 2019; Zacharias and Yunes, 2020). This is because considering these factors makes the modeling complex and cannot be solved optimally by available optimization methodologies. As a remedy to these problems, we propose a novel stochastic mixed-integer programming model that simultaneously considers patient-dependent stochastic service times, and patient-and-time-dependent unpunctuality and no-shows. In addition, the proposed model considers arbitrary probability distributions for uncertain parameters without limiting the probability distributions to be of the same type and captures these distributions as they are without any sampling. The model benefits from a pseudo-polynomial number of variables and constraints that enables us to consider an exponential number of stochastic scenarios and solve the problem optimally or with a very low optimality gap in a reasonable time. Through these enhancements, our research offers a more flexible, and comprehensive solution to appointment scheduling challenges and sets a new benchmark for personalized patient care and clinic efficiency.

2.3 Problem Definition and Formulation

We consider an appointment scheduling problem where we have to assign appointment times to a certain number of patients. A single healthcare provider continues visiting patients up to a maximum overtime limit beyond which the remaining appointments are cancelled. Some of the main challenges in this problem are as follows:

(1) Service durations are both stochastic and patient-dependent. This variability in service durations reflects the diversity in patients' health conditions. In our study, we explore a broad setting where service durations are allowed to follow diverse, non-identical independent probability distributions without being confined to well-known distributions.

(2) Patient arrival times are stochastic and patient-and-time-dependent that reflect the uncertainty associated with patient unpunctuality. Patient-dependency means that, for each patient, we have a different set of probability distributions to represent his/her unpunctuality. Also, time-dependency means that the amount of lateness/earliness depends on the allocated appointment time. We consider this assumption to incorporate the effect of traffic in rush hours on patient unpunctuality. This setting allows for greater modeling flexibility by considering patient-and-time-dependency simultaneously.

(3) In some cases, patients may not show up for their appointment. As suggested by Kong et al. (2020), we assume that the probability of no-shows is time-dependent. Our model advances this assumption by incorporating both time-dependent and patient-dependent probabilities for no-shows, thereby enriching the existing framework with a more detailed understanding of patient behavior

The primary goal of this research is to minimize the weighted sum of healthcare provider idle times, patient waiting times, and the penalties corresponding to health provider overtime. In the following, initially, we present a simplified model that mainly focuses on the stochastic patient-dependent service durations. This simplification allows us to establish a fundamental understanding of the scheduling dynamics and the presented model without the added complexities of patient

unpunctuality and no-shows. Building upon the simplified model, we progressively integrate the stochastic patient unpunctuality and no-shows in Section 2.3.2. In Section 2.3.3, we further refine our model to incorporate the impact of appointment reminders and patient notifications on no-show probabilities. This enhancement captures the interactive dynamics between clinic communication strategies and patient behavior and offers insights into how proactive patient notifications in response to reminders can influence scheduling efficiency and resource allocation.

2.3.1 The Basic Stochastic Appointment Scheduling Model

We formulate the problem by a novel SP approach, called state variable model, that results in a closed-form MIP model (Doulabi et al., 2022). The main advantage of this formulation is its ability to handle an exponential number of scenarios efficiently using a pseudo-polynomial number of variables and constraints. This feature is particularly beneficial as it facilitates the solution of large-scale instances with an exponential number of scenarios within a computationally feasible time. To present the proposed model, we define the sets, variables, and parameters as follows:

Sets:

I : The ordered set of patients, .i.e., $I = \{1, 2, \ldots, |I|\}$. Also, we suppose $I_0 = I \cup \{0\}$ where "0" denotes a dummy patient with a service time equal to 0 that is scheduled before the first patient at the beginning of the clinic time.

T : The set of possible appointment times that are multiples of a small time unit (e.g., 2 minutes) denoted by θ, resulting in $T = \{0, \theta, 2\theta, 3\theta \ldots, L + O\}$. We suppose that all stochastic service and arrival times are multiple of θ.

T_i : The set of possible appointment times for patient i. We have

$$T_i = \begin{cases} \{0\}, & i = 1, \\ \left\{t \in T \mid t \geq \sum_{i' \in I: i' < i} t_{i'}^{min}\right\}, & i \in I \setminus \{1\}, \\ \{.\}, & i = |I| + 1. \end{cases}$$

$T_1 = \{0\}$ shows that we always offer the first available time to patient 1. In computation of T_i for $i \in I$, we consider the minimum service time of previous patients denoted by $t_{i'}^{min}$. Also, $i = |I| + 1$ denotes a dummy patient that is always scheduled after the last patient at a dummy time $\{.\}$.

F_i : The set of possible finish times for patient i. We have

$$F_i = \begin{cases} \left\{t \in T \mid t \geq \sum_{i' \in I: i' \leq i} t_{i'}^{min}\right\} \cup \{f^*\}, & i \neq 0, \\ \{0\}, & i = 0. \end{cases}$$

In this relation, f^* represents the case that the appointment of patient i finishes at the time slot after cancellation threshold $L + O$. For simplicity of understanding the model, one can consider $f^* = O + L + 1$.

Parameters:

L : The available clinic time.

O : The maximum available overtime of the clinic. After time L+O, the appointments of all remaining patients (if any are available) are canceled.

t_i^{min} : The minimum possible service time of patient i.

$c^p_{if_it_{i+1}}$: The waiting cost of patient $i+1$ provided that he/she is given an appointment time of t_{i+1} and the service to patient i has finished at the time f_i. We have $c^p_{if_it_{i+1}} = \alpha^p[f_i - t_{i+1}]^+$ where α^p is the penalty for a patient waiting for a time unit and $[x]^+$ is equal to x if it is positive and 0 otherwise.

$c^d_{if_it_{i+1}}$: The health provider idle cost while waiting for patient $i+1$ provided that the patient is given an appointment time of t_{i+1} and the service to the previous patient i has finished at the time f_i. We have $c^d_{if_it_{i+1}} = \alpha^d[t_{i+1} - f_i]^+$ where α^d is the penalty for the health provider waiting for a time unit.

$c^o_{if_it_{i+1}}$: The expected over time cost corresponding to the service to patient $i+1$ provided that the patient is given an appointment time of t_{i+1} and the service to the previous patient i has finished at the time f_i. We have $c^o_{if_it_{i+1}} = \alpha^o \Big(E[\max\Big(L, \max(f_i, t_{i+1}) + \tilde{d}_{i+1}\Big) - \max(L, f_i)]\Big)$ where $\tilde{d}_{i+1}$ is a random variable representing the service duration of the patient $i+1$.

α^c : The cancellation cost of an appointment.

Variables:

w_{it_i} : 1 if we assign time slot t_i to patient i; 0 otherwise.

$z_{if_it_{i+1}}$: The probability that the service to patient i finishes at the time f_i (for $f_i < O + L$) and we have $w_{(i+1)t_{i+1}} = 1$, i.e., the next patient $i+1$ is scheduled at the time t_{i+1}. If $w_{(i+1)t_{i+1}} = 1$ holds, $z_{if_it_{i+1}}$ will be equal to the above probability. Otherwise, it will be equal to 0. In the case of $i = |I| + 1$, t_{i+1} can only take the dummy value "." and we ignore the last condition $w_{(i+1)t_{i+1}} = 1$. Also, for $f_i = O + L$, $z_{if_it_{i+1}}$ computes the probability that the service to patient i finishes at the time f_i or later and we have $w_{(i+1)t_{i+1}} = 1$. Finally, for $f_i = f^*$, $z_{if_it_{i+1}}$ represents the probability that patient *i* is cancelled due to exceeding the cancellation threshold of $O+L$, while for the next patient we have $w_{(i+1)t_{i+1}} = 1$,

Using the given notation, we formulate the simplified appointment scheduling problem as (M1).

$$(M1) \quad \min \sum_{i\in I_0} \sum_{f_i\in F_i} \sum_{t_{i+1}\in T_{i+1}} \Big(c^p_{if_it_{i+1}} + c^d_{if_it_{i+1}} + c^o_{if_it_{i+1}} + \alpha^c \mathbf{1}_{(f_i=f^*)} \Big) z_{if_it_{i+1}} \tag{1}$$

$$\sum_{t_i\in T_i} w_{it_i} = 1, \qquad \forall i \in I, \tag{2}$$

$$\sum_{f_{i-1}\in F_{i-1}} z_{(i-1)f_{i-1}t_i} = w_{it_i}, \qquad \forall i \in I,\ t_i \in T_i, \tag{3}$$

$$\sum_{f_{i-1}\in F_{i-1}} \sum_{t_i\in T_i} z_{(i-1)f_{i-1}t_i} \Pr\Big(\max(t_i, f_{i-1}) + \widetilde{d}_i = f_i\Big) = \sum_{t_{i+1}\in T_{i+1}} z_{if_it_{i+1}}, \qquad \forall i \in I,\ f_i \in F_i \setminus \{f^*\}, \tag{4}$$

$$\sum_{f_{i-1}\in F_{i-1}} \sum_{t_i\in T_i} z_{(i-1)f_{i-1}t_i} \Pr\Big(\max(t_i, f_{i-1}) + \widetilde{d}_i \geq f_i\Big) = \sum_{t_{i+1}\in T_{i+1}} z_{if_it_{i+1}}, \qquad \forall i \in I,\ f_i = f^*, \tag{5}$$

$$w_{it_i} \in \{0,1\}, \qquad \forall i \in I,\ t_i \in T_i, \tag{6}$$

$$0 \leq z_{if_it_{i+1}} \leq 1, \qquad \forall i \in I^0,\ f_i \in F_i,\ t_{i+1} \in T_{i+1}. \tag{7}$$

In equation (1), the objective function minimizes the expected value of the total cost. The function includes an indicator, $\mathbf{1}_{(f_i=f^*)}$ which is set to 1 when the condition $f_i{=}f^*$ holds and 0 otherwise. Constraint (2) ensures that we assign exactly a single appointment time to each patient to avoid any missing or duplicated appointments. Constraint (3) establishes a relationship between variables $z_{(i-1)f_{i-1}t_i}$ and w_{it_i}. This constraint implies that if w_{it_i} is equal to 0, then all corresponding variables $z_{(i-1)f_{i-1}t_i}$ are equal to 0. This is consistent with the definition of variables $z_{(i-1)f_{i-1}t_i}$. When w_{it_i} is equal to 1, the sum of variables $z_{(i-1)f_{i-1}t_i}$ on the left-hand side of constraint (3) must be equal to 1. This reflects that for a fixed $i{\in}I$, and $t_i{\in}T_i$, $z_{(i-1)f_{i-1}t_i}$ variables represent the probability distribution of the finish time of service to the patient $(i{-}1)$. Constraint (4) consecutively compute the values of variables $z_{if_it_{i+1}}$ using conditional probability relations based on variables $z_{(i-1)f_{i-1}t_i}$ and the probability terms $\Pr(\max(t_i, f_{i-1}){+}\tilde{d}_{i\xi}{=}f_i)$. The right-hand side of this constraint calculates the probability that the appointment of patient i finishes at the time f_i. The left-hand side of the constraint (4) also computes the same probability by multiplying $z_{(i-1)f_{i-1}t_i}$,

which represents the probability that the appointment of the patient $(i-1)$ finishes at the time f_{i-1} and patient i is scheduled at the time t_i, by the probability $\Pr(\max(t_i, f_{i-1})+\tilde{d}_{i\xi}=f_i)$. This represents the probability that the duration of the appointment for patient i is such that we finish patient i at the time f_i. Similarly, constraint (5) calculate the values of variables $z_{if_it_{i+1}}$ for $f_i = f^*$ using conditional probability relations based on variables $z_{(i-1)f_{i-1}t_i}$ and the probability term $\Pr(\max(t_i, f_{i-1})+\tilde{d}_{i\xi}\geq f_i)$.

Theorem 1. *Model (1) - (7) is a valid formulation.*

The proof of Theorem 1 is provided in Appendix A. To illustrate the application of the proposed model, particularly in managing service time uncertainty in the Appointment Scheduling (AS) problem, we provide a numerical example in Appendix B.

2.3.2 Incorporating No-Show and Punctuality

In this section, we enhance our model by incorporating stochastic arrival times and no-shows to further align it with the unpredictable nature of real-world healthcare scheduling scenarios. We introduce the random variable $\tilde{a}_{it}$ to represent time perturbations in the arrival of patient *i* from his/her appointment scheduled at time t. A negative value of $\tilde{a}_{it}$ indicates that the patient arrives earlier than expected, whereas a positive value indicates a delay. Moreover, we define parameter π_{it} to represent the probability of a no-show for patient *i* at appointment time *t*. These enhancements are crucial for capturing the dynamic and uncertain aspects of patient attendance behavior. We also note that that these stochastic parameters take into account the patient-and-time-dependency simultaneously.

To integrate these factors into our state-variable model, we modify the transition probabilities in constraints (4) and (5). The revised probabilities account for variations in arrival times, as denoted by $\tilde{a}_{it}$, and the probability of patient no-shows, as represented by π_{it}. Therefore, the updated transition probability in constraint (4), which now incorporates the stochastic nature of patient arrivals and no-show probabilities, is formulated as:

$$\pi_{it}\Pr\left((\max\left(t_i, f_{i-1}\right)\right)=f_i) +(1-\pi_{it})\Pr(\max(t_i+\tilde{a}_{it}, f_{i-1})+\tilde{d}_i=f_i). \tag{8}$$

Similarly, the revised probability for constraint (5) is formulated as:

$$\pi_{it}\Pr\left((\max\left(t_i, f_{i-1}\right)\ ) \geq f_i\right) + (1-\pi_{it})\Pr(\max(t_i+\tilde{a}_{it}, f_{i-1})+\tilde{d}_i \geq f_i). \tag{9}$$

These modifications ensure that our model accurately reflects the stochastic nature of patient arrival times and no-shows when determining the appointment start and finish times. This enhances the model's applicability and reliability in practical healthcare scheduling cases.

2.3.3 Incorporating No-Shows and Punctuality and Reminders

In this section, we augment our model to encompass a realistic scenario in healthcare scheduling that is when patients notifying the clinic in advance if they are unable to attend their appointment. This scenario is particularly relevant for clinics that utilize reminder systems and prompt patients to confirm or cancel their appointments.

To model this new feature, we introduce a probability π^{info} which represents the likelihood of patients notifying the clinic about their no-show in response to reminders. Using this probability, we can evaluate the effectiveness of the reminder system in prompting patient responses regarding their appointment attendance.

The incorporation of π^{info} necessitates modifications to the transition probabilities in constraints (4) and (5). These adjustments are essential to accurately represent the dynamics of the system when patients proactively communicate their attendance or absence. By considering stochastic service times, arrivals, no-shows, and the probability of responding to reminders, the new transition probability for constraint (4) is formulated as:

$$\begin{aligned} \pi_{it}\left[\pi^{info}\mathbf{1}\left(f_{i-1}=f_i\right) + \left(1-\pi^{info}\right)\Pr\left(\max(t_i, f_{i-1}) = f_i\right)\right] \\ +(1-\pi_{it})\Pr\left(\max(t_i+\tilde{a}_{it}, f_{i-1}) + \tilde{d}_i = f_i\right). \end{aligned} \tag{10}$$

Consequently, the probability transition in constraint (5) will be updated to reflect these additional factors. The revised formulation considering stochastic service times, arrivals, no-show, and

the likelihood of patients responding to reminders is formulated as:

$$\pi_{it}\left[\pi^{info}\mathbf{1}\left(f_{i-1}\geq f_i\right)+\left(1-\pi^{info}\right)\Pr\left(\max(t_i,f_{i-1})\geq f_i\right)\right] \\ +(1-\pi_{it})\Pr\left(\max(t_i+\tilde{a}_{it},f_{i-1})+\tilde{d}_i\geq f_i\right). \tag{11}$$

It is important to note that the validity of this enhanced model originates from the validity of the original model as demonstrated in Appendix A. The primary difference in the enhanced model is in the updated transition probabilities which now accommodate the added detail of patient responses to reminders.

2.4 Computational results

Our computational analysis aims to evaluate the performance of the proposed state-variable model through a series of targeted questions, each designed to assess a critical aspect of healthcare appointment scheduling. These computational analyses will aim at answering the following questions:

(1) Does our proposed state-variable model demonstrate computational efficiency?

(2) How does the inclusion of different HS levels influence the appointment scheduling efficiency?

(3) What is the impact of patient-and-time-dependent SA on appointment scheduling efficiency?

(4) In what ways do considering patient-and-time-dependent NS levels affect the appointment schedule?

(5) How does incorporating reminders to encourage patients to disclose their no-show behavior alter the appointment schedule and overall system efficiency?

(6) What are the effects of enhanced patient show-up, for instance through incentives, on the optimal appointment schedule efficiency?

To design our computational experiments, we consider the three key factors on clinic operational

efficiency including service times, stochastic arrivals, and no-shows. As it will be discussed in Section 2.4.1.1, we categorize service times into four levels and both stochastic arrivals and no-shows into five levels to consider the independent and simultaneous effect of patient and time dependency in in these factors. Our computational experiments include the following analyses to answer the above questions:

(1) To answer Question 1, first, we evaluate the computational efficiency of the model by solving several instances using CPLEX in C++. This assessment will provide insights into the practicality and scalability of the model in various settings.

(2) To answer Questions 2, 3, and 4, we separately explore the effects of incorporating patient-dependent service times (HS), and patient-and-time-dependent stochastic arrival times (SA), and no-shows (NS) into the scheduling model.

(3) To answer Question 5, we assess the influence of patient information-sharing through responding to reminders on appointment scheduling. This includes studying how encouraging patients to proactively communicate their attendance intentions, particularly through no-show disclosures in response to reminders, affect scheduling efficiency.

(4) To answer Question 6, we introduce a scaling parameter that adjusts the show-up probability of patients. This parameter is crucial for understanding how fluctuations in patient attendance probabilities affect the appointment schedule efficiency.

In the following, first we explain the instance generation and the different settings for HS, SA, and NS factors in Section 2.4.1. Then in Section 2.4.2, we provide the results of the computational experiments and provide several managerial insights.

2.4.1 Instance generation

To generate instances, we consider session lengths of 150, 210, and 270 minutes to reflect diversity in clinical practices as in (Salzarulo et al., 2016; Klassen and Yoogalingam, 2009). Moreover, we consider a predetermined overtime allowance of 60 minutes. Our model incorporates three cost parameters including α^d, α^p, and α^o which represent the cost per minute for healthcare provider

idleness, patient wait time, and overtime, respectively. In our research, the cost ratio factor (α^d/α^p) represents the relative importance of healthcare provider time to patient waiting time. Following Klassen and Yoogalingam (2009) and Salzarulo et al. (2016), we explore the cost ratio across three distinct levels including 1, 5, and 10, to understand its impact on operational efficiency. Moreover, the overtime cost is set to 1.5 times higher than the idle time cost (Salzarulo et al., 2016). We investigate the above-mentioned settings in instances with 10, 12, and 14 patients to assess the effectiveness of the model across various clinical settings. This approach enables us to thoroughly evaluate the interplay of these diverse factors and their collective impact on the efficiency of clinical appointment scheduling.

Following Klassen and Yoogalingam (2009), we consider no-show rates ranging from 0% to 30% that are generated using a uniform distribution. To generate realistic values for unpunctuality, we assume an average early arrival time of 10 minutes. The data is derived from several normal distributions with means of -5, -10, and -15 minutes and a standard deviation of 1.7 as suggested by (Salzarulo et al., 2016; Cayirli et al., 2008).

2.4.1.1 Service times

As in Salzarulo et al. (2016), in Set 1, we model the service times as probability functions of patients' health states. In the following, we consider four levels of information on patients' health states. Each level introduces increasing complexity in terms of patient-dependent information and needs, as follows:

HS0 (Level 1): This baseline level does not incorporate any specific health information and assumes that appointment durations are uniform and not influenced by individual patient characteristics.

HS1 (Level 2): At this level, patients are classified as either returning or new. To estimate the mean examination time for the ith patient and its standard error, we utilize a logarithmic transformation as follows (Salzarulo et al., 2016):

$$\ln(\mu_i) = 2.329RP + 2.247NP, \tag{12}$$

$$\sigma_i = 0.658. \tag{13}$$

HS2 (Level 3): Here, we incorporate more patient-specific information and categorize patients into four health states including Low Health, Moderate Health, High Health, and Excellent Health. In addition, we consider factors such as prescription medication (PM), physical examination (PE), total chronic conditions (TC), and unspecified visit reasons (NR). The log-transformed mean examination time is computed as follows (Salzarulo et al., 2016):

$$\ln(\mu_i) = 3.088H_i^{Low} + 2.430H_i^{Mid} + 2.341H_i^{High} + 2.184H_i^{Perf} + 0.147TC - 0.106NR + 0.315PE - 0.102PM, \quad (14)$$

with a standard error of $\sigma_i = 0.644$.

HS3 (Level 4): This advanced level includes the main effects of HS2 and additional two-way interactions. At this level, we calculate the log transformation of ith patient's mean examination duration and the standard deviation as follows (Salzarulo et al., 2016):

$$\ln(\mu_i) = 3.112H_i^{Low} + 2.462H_i^{Mid} + 2.388H_i^{High} + 2.245H_i^{Perf} + 0.145TC - 0.116NR + 0.265PE - 0.108PM - 0.105NP - 0.369BC + 0.785NP(BC), \quad (15)$$

$$\sigma_i = 0.642. \quad (16)$$

Table 2.4 presents a concise summary of the key factors considered at each health state level. This Table outlines the presence (+) or absence (-) of each factor across HS0, HS1, HS2, and HS3 (Salzarulo et al., 2016).

To assess the impact of varying levels of health state information levels on appointment scheduling, we generate 10 instances for each combination of clinic time, cost ratio, and the number of patients. This approach yields a substantial dataset comprising 1080 unique instances that offers a rich basis for in-depth analysis. This comprehensive approach allows us to perform a detailed investigation of how health state information affects appointment scheduling in clinical settings.

Table 2.4: Key factors at each health state level

	Health State			
	HS0	HS1	HS2	HS3
Return Patients (RP)	-	+	-	-
New patients (NP)	-	+	+	-
Low Health- (H_i^{Low})	-	-	+	+
Moderate Health- (H_i^{Mid})	-	-	+	+
High Health- (H_i^{Hi})	-	-	+	+
Excellent Health- (H_i^{Perf})	-	-	+	+
Total Chronic (TC)	-	-	+	+
No Reason (NR)	-	-	+	+
Physical Examination (PE)	-	-	+	+
Prescription Medication (PM)	-	-	+	+
Birth Control (BC)	-	-	-	+
New Patient $\times$ Birth Control NP(BC)	-	-	-	+

2.4.1.2 Patient Unpunctuality

In Set 2, we examine different levels of patient unpunctuality categorized as SA0, SA1, SA2, SA3, and SA4 where each represents varying degrees of information about patient unpunctuality.

SA0 (Level 1): This baseline level assumes all patients are punctual and arrive on time.

SA1 (Level 2): Here, we introduce a single unpunctuality distribution for all patients that is uniform across various time slots. This setting represents the case that the clinic collects unpunctuality data regardless of the patient information and the appointment time.

SA2 (Level 3): We consider different unpunctuality distributions for different time slots. We note that the distribution considered for each time slot is independent of the patient information, and it applies to the time slot regardless of the patient information. This setting represents the case that the clinic collects the unpunctuality data considering the appointment time regardless of the patient information.

SA3 (Level 4): At this level, we consider different patient-dependent unpunctuality distributions that apply to all time slots. Essentially, it acknowledges the variance in patient arrival patterns but assumes these patterns are consistent across various times of the day. This setting models a scenario where the clinic gathers unpunctuality data unique to each patient but does not differentiate the data based on appointment times.

SA4 (Level 5): In the most detailed level, SA4, we introduce patient-and-time-dependent unpunctuality distributions. This setup represents the case where the clinic collects detailed unpunctuality data considering both the patient behavior and the appointment times.

In the following, we first explain the formulas for SA4 in which the unpunctuality of patient i is patient-and-time-dependent. Then, we provide information about other levels that are simplified versions of the fifth level. For SA4, we model patient unpunctuality using a truncated normal distribution with a mean μ_{it} and standard deviation $\sigma_{arrival}$. For this purpose, we present the following equation:

$$\mu_{it} = \mu_i + \beta \left(\overline{\mu} \left(\frac{\alpha_t}{\overline{\alpha}} \right) - \overline{\mu} \right), \tag{17}$$

where μ_i stands for the mean unpunctuality of patient i during regular traffic. Besides, the coefficient β is a control parameter to increase or decrease the effect of traffic. Coefficient α_t is the traffic index that reflects the relative traffic volume at time slot t. The average traffic index $\overline{\alpha}$ and average punctuality $\overline{\mu}$ are calculated over all time slots and patients, respectively, as follows.

$$\overline{\alpha} = \frac{\sum_{t \in T} \alpha_t}{|T|}. \tag{18}$$

$$\overline{\mu} = \frac{\sum_{i \in I} \mu_i}{|I|}. \tag{19}$$

In Eq.(18), for the values of α_t we have used the traffic data of arterial roads in urban areas of Ohio in 2016 (Ohio Department of Transportation, 2023). Also, in Eq.(19), we generate patient-specific mean unpunctuality μ_i using a uniform distribution in the range of [-25, -5] minutes. This results in $|I| \times |T|$ distributions for μ_{it} in SA4. In all these truncated normal distributions, we considered the domain of $[\mu_{it} - 3\sigma_{it}, \mu_{it} + 3\sigma_{it}]$ for the random parameter where σ_{it} is the standard deviation of distribution.

To construct the unpunctuality models for SA1, SA2, and SA3, we average μ_{it} over different dimensions. For SA3, we calculate the average of the parameter μ_{it} over time slots yielding a unique truncated normal distribution for each patient. For SA2, we average μ_{it} over patients for each time slot which yields a truncated normal distribution for each time slot. Finally, for SA1, parameter

μ_{it} is averaged over both patients and time slots which creates a single distribution applicable to all patients and time slots.

To assess the impact of varying levels of detail about patient unpunctuality on appointment scheduling, we generate 10 instances for each combination of clinic time, cost ratio, and the number of patients that results in a total of 1350 instances. Such an extensive evaluation is crucial to thoroughly investigate how different levels of unpunctuality information affect appointment scheduling. This will enhance our understanding of the dynamics involved in managing patient arrivals in various healthcare settings.

2.4.1.3 No-shows

In Set 3 of our analysis of the appointment scheduling problem, we have integrated no-shows as a pivotal factor. We consider five distinct levels including NS0, NS1, NS2, NS3, and NS4 to represent varying degrees of information about patient no-shows.

NS0 (Level 1): At this level, we consider a basic scenario with no specific no-show data in which all patients are assumed to attend their appointments.

NS1 (Level 2): Here a single distribution for no-shows applicable to all patients across all time slots. The distribution is valid for all time slots and all patients as we do not have time and patient dependency. This level models a scenario where a clinic collects no-show data regardless of the patient and the appointment time.

NS2 (Level 3): This level introduces time-dependent no-show distributions. We consider a specific distribution for each timeslot that applies to all patients assigned to that timeslot. This setting represents a case in which clinics collect no-show data based on appointment times while ignoring individual patient characteristics.

NS3 (Level 4): Here, we focus on patient- dependent no-show distributions that are consistent across different time slots. This approach represents clinics that collect individual patient no-show data regardless of their appointment times.

NS4 (Level 5): This is the most detailed level, where no-show probabilities are tailored to individual patients and specific time slots. This setting depicts a clinic collecting comprehensive no-show data considering both patient behavior and appointment times.

In the following, we first explain the process to generate NS4, and then we provide information about NS3, NS2, NS1, and NS0 levels. For NS4, the no-show probability for each patient *i* at time slot t is set to $\pi_{it}=\pi_i^{'}+\pi_t^{''}$, where parameter $\pi_t^{''}$ is a time-dependent no-show parameter and parameter $\pi_i^{'}$ is a patient-dependent constant that enables π_{it} to take varying values for different patients at a particular time slot t. For $\pi_t^{''}$, we refer to Figure 1-a in Kong et al. (2020) for time-dependent no-show probabilities. Besides, we uniformly generated $\pi_i^{'}$ in the interval [-0.3, 0.3].

To generate NS3, we calculate the average of π_{it} over different time slots for each patient and obtained a distinct truncated normal distribution for each patient. For NS2, we compute the average of π_{it} across patients for each time slot which yields a unique truncated normal distribution for each time slot. Furthermore, for NS1, we calculate the average of π_{it} across both patients and time slots and determine one truncated normal distribution for all patients and all time slots.

To assess the impact of varying no-show levels on appointment scheduling, we generate 10 instances for each combination of clinic time, cost ratio, and the number of patients that results in a total of 1350 instances.

2.4.2 Computational experiments and results

In this section, we report the computational results to answer questions presented at the beginning of Section 2.4. To solve the proposed model, we use CPLEX 12.10 on a PC with an AMD Rome 7532 @ 2.40 GHz 256M cache L3 CPU and 16 GB of RAM. We considered a time limit of 24 hours for each instance. Remarkably, the majority of instances were resolved well before this limit that shows the effectiveness of the model in handling complex scheduling problems. This performance indicates not only the computational viability of the proposed approach for real-world application but also its potential for scalability to larger or more complex instances in healthcare scheduling.

2.4.2.1 Tractability of the model

In the following, we perform a comprehensive assessment of our proposed appointment scheduling model and focus particularly on its performance and computational efficiency. We use the CPLEX solver and extensively test the model across various settings. Table 2.5 presents the summary of the computational results of the CPLEX on 270 instances. In this table, the first three columns

indicate the settings of instances including the number of patients, clinic time, and cost ratio. In the next columns, we report 1) the number of the stochastic scenarios captured by our model, 2) the number of variables, 3) the number of constraints, 4) the number of nodes exploited in the branch-and-bound tree, 5) lower bound, 6) upper bound, 7) time: computational time of the model in seconds, 8) gap(%): which is computed by gap = $\frac{100(UB - LB)}{LB}$, 9) waiting cost, 10) idle cost, and 11) overtime cost. We note that in all these instances we considered HS, NS, and SA at their highest level of detail that provide in-depth information about the health state, no-show, and unpunctuality of the patients. This means that the unpunctuality and no-show probabilities follow patient-and-time-dependent distributions.

Table 2.5 reveals that the proposed model efficiently solves appointment scheduling problems with an exponential number of stochastic scenarios (Ω = 1.97E+8890) within a reasonable time limit. This represents a significant accomplishment, especially considering the complexity and scale of the addressed scheduling problems, which involve patient-and-time-dependent unpunctuality and no-shows, along with patient-dependent service times. The average optimality gap reported is 0.2 percent with the majority of instances achieving optimal solutions within the allocated time limit. The average number of variables and constraints are 171323 and 12020, respectively, which are relatively low considering the exponential number of stochastic scenarios captured by the proposed model. This highlights the efficiency and innovation of our approach.

A key observation from our analysis is the impact of the number of patients on the clinic's operational costs. Specifically, as the number of patients increases, while keeping clinic time and cost ratio constant, there is a corresponding rise in waiting costs, idle costs, and overtime costs. Furthermore, an increase in the cost ratio also leads to increases in costs across waiting costs, idle costs, and overtime costs. This highlights the need for balancing the workload of healthcare providers and the patient satisfaction to avoid overburdening the former and dissatisfaction in the latter.

Figure 2.1 presents a visual representation of the calculated finish time probabilities as determined by our proposed model for a given instance with 14 patients.

Table 2.5: Computational results of the CPLEX in solving the proposed state-variable model.

No. of patients	Clinic time	Cost ratio	$\|\Omega\|$	No. of variables	No. of constraints	No. of nodes	LB	UB	Time (sec)	Gap (%)	Waiting Cost	Idle Cost	Overtime Cost
10	150	1	9.14E+4050	83031	6880	1551	19.96	19.96	833	0.00	6.31	13.65	0.00
		5	9.14E+4050	83031	6880	2326	51.08	51.08	856	0.00	25.41	25.66	0.01
		10	9.14E+4050	83031	6880	2367	69.57	69.57	894	0.00	41.53	28.02	0.01
	210	1	8.04E+5198	135903	8824	1987	19.31	19.31	1554	0.00	5.59	13.72	0.00
		5	8.04E+5198	135903	8824	2354	49.74	49.74	1454	0.00	24.49	25.24	0.00
		10	8.04E+5198	135903	8824	3070	68.00	68.00	1735	0.00	41.16	26.84	0.00
	270	1	8.28E+6349	201735	10768	1843	19.94	19.94	1834	0.00	6.23	13.71	0.00
		5	8.28E+6349	201735	10768	2736	51.05	51.05	2351	0.00	25.70	25.35	0.00
		10	8.28E+6349	201735	10768	2350	69.54	69.54	2040	0.00	41.53	28.01	0.00
12	150	1	9.06E+4859	101435	9259	8420	31.45	31.45	9336	0.00	10.10	20.97	0.38
		5	9.06E+4859	101435	9259	13114	81.28	81.28	11098	0.00	40.13	39.70	1.45
		10	9.06E+4859	101435	9259	10622	112.21	112.21	8629	0.00	64.24	44.92	3.04
	210	1	6.95E+6239	166043	11875	7638	30.04	30.04	15247	0.00	9.72	20.32	0.00
		5	6.95E+6239	166043	11875	11051	78.30	78.30	17283	0.00	39.42	38.88	0.00
		10	6.95E+6239	166043	11875	15599	106.94	106.94	17161	0.00	63.18	43.76	0.00
	270	1	5.59E+7618	246491	14491	7399	31.19	31.19	21833	0.00	10.16	21.03	0.00
		5	5.59E+7618	246491	14491	6071	80.43	80.43	18945	0.00	40.39	40.04	0.00
		10	5.59E+7618	246491	14491	7860	110.34	110.34	17537	0.00	62.61	47.74	0.00
14	150	1	7.17E+5669	119839	11978	14762	41.12	41.12	21304	0.00	14.91	19.82	6.40
		5	7.17E+5669	119839	11978	22654	107.94	107.94	30993	0.00	47.52	40.32	20.11
		10	7.17E+5669	119839	11978	39844	158.07	158.68	43079	0.49	72.48	47.74	38.47
	210	1	9.69E+7279	196183	15362	15203	32.03	32.15	42442	0.37	10.78	21.37	0.00
		5	9.69E+7279	196183	15362	21997	83.97	84.94	47453	1.13	41.02	43.91	0.00
		10	9.69E+7279	196183	15362	36605	113.69	116.76	62017	2.90	65.43	51.33	0.00
	270	1	1.97E+8890	291247	18746	4502	34.58	34.58	18916	0.00	13.36	21.22	0.00
		5	1.97E+8890	291247	18746	7313	88.29	88.29	22508	0.00	45.79	42.50	0.00
		10	1.97E+8890	291247	18746	16904	120.30	121.05	43601	0.60	71.94	49.11	0.00
Average				171323	12020	10672	68.90	69.11	17886	0.20	34.86	31.66	2.59

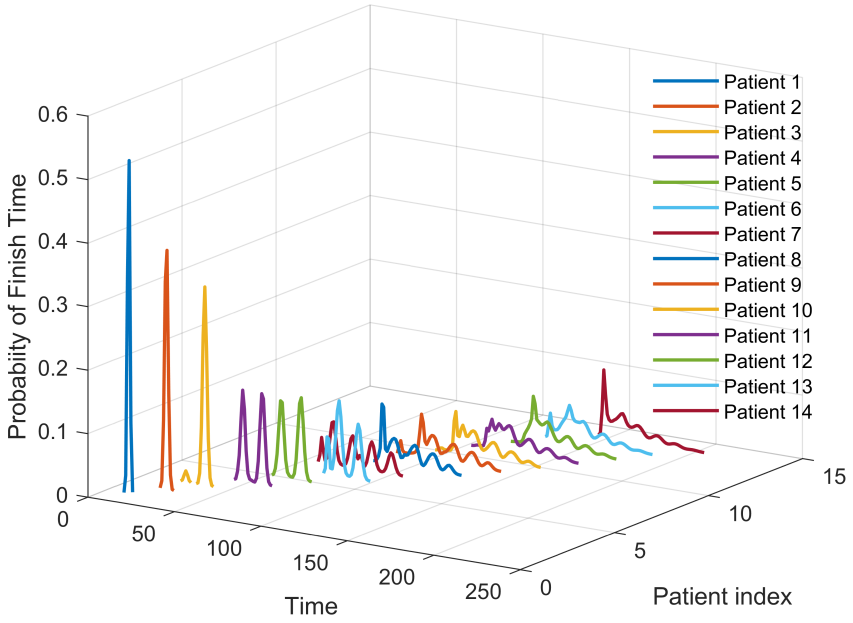

Figure 2.1: Calculated finish time probabilities in 3D.

Based on Figure 2.1, we observe that the probability of finish time of the appointment for last patients becomes more uncertain compared to those of first patients. This is mainly due to the fact that the uncertainty in no-show, unpunctuality, and service durations of the first patients affect the start time of the appointment for later patients.

2.4.2.2 Analysis of Health States (HS)

In Table 2.6, we thoroughly examine the impact of varying levels of Health State (HS) on the appointment scheduling problem. In this table, for all instances, we have considered the highest levels of stochastic arrival (SA4) and no-shows (NS4) for all patients. This means we have patient-and-time-dependent unpunctuality and no-show probability distributions. For instances with the setting of HS3, we have reported the results obtained from the proposed model directly in the table. However, the solutions of instances with other health settings (i.e., HS0, HS1, and HS2) must be evaluated in the reference setting of "HS3-SA4-NS4" that represents the real-world situation. This approach aligns with our objective outlined in Question 2 of Section 2.4 that aims to assess the

significance of incorporating detailed patient health information into scheduling effectiveness.

In order to simulate these solutions (HS0, HS1, and HS2) in the context of the most comprehensive health state setting (HS3-SA4-NS4), we first solve the proposed model with "(HS0, HS1, or HS2)-SA4-NS4" and save their appointment schedule presented by w_{it_i} variables. Then, we fix the w_{it_i} variables and resolve the model with the reference setting of "HS3-SA4-NS4" to assess (or equivalently simulate) the solutions obtained in "(HS0, HS1, or HS2)-SA4-NS4". This method allows us to directly compare the outcomes across different health state settings which provides insights into the value of detailed patient health information in optimizing appointment scheduling in complex healthcare environments.

Our results indicate that as we move from HS0 to more detailed levels of HS (HS1, HS2, and HS3), there is a significant decrease in the total cost, waiting cost, and overtime cost for a given instance. This trend highlights the substantial benefits of incorporating a comprehensive range of health related factors into the scheduling model. By integrating more detailed and patient-specific health information, the model benefits from more accurate probability distributions for service times and subsequently generates more efficient schedules. Moreover, the results reveal that an increase in the cost ratio leads to higher total, waiting, idle, and overtime costs across all HS levels.

When comparing HS2 against HS3, we notice only marginal improvements in reducing total costs. This suggests that while advancing from a moderate to a more detailed understanding of patient health states does contribute to efficiency, the gains are relatively slight. Given the costs associated with collecting and integrating more detailed health state data, stakeholders must carefully consider whether the slight improvements justify the additional efforts and used resources. This is especially pertinent in settings where resources are limited, and the incremental cost savings might not offset the investment required for gathering comprehensive health data.

Similarly, the transition from HS0 to HS1 also shows improvements in total cost; however, these improvements are modest. This comparison indicates that moving from a basic to a slightly more detailed health state offers benefits, but similar to the HS2 to HS3 transition, the value of these improvements must be weighed against the costs of data collection and processing.

Table 2.6: Computational results for HS variations.

Instance information			Total Cost				Waiting Cost				Idle Cost				Overtime Cost			
No. of patients	Clinic time	Cost ratio	HS0	HS1	HS2	HS3	HS0	HS1	HS2	HS3	HS0	HS1	HS2	HS3	HS0	HS1	HS2	HS3
10	150	1	46.63	45.61	20.48	19.96	41.15	40.08	5.95	6.31	5.48	5.52	14.53	13.65	0	0	0	0
		5	79.89	79	51.91	51.08	68.83	68.99	24.59	25.41	11.06	10	27.32	25.66	0.01	0.01	0.01	0.01
		10	100.36	100.24	70.97	69.57	87.59	88.27	40.55	41.53	12.75	11.96	30.41	28.02	0.02	0.02	0.01	0.01
	210	1	45.62	45.02	19.84	19.31	40.28	39.62	5.68	5.59	5.34	5.39	14.17	13.72	0	0	0	0
		5	77.44	77.32	50.39	49.74	66.92	67.59	24.45	24.49	10.52	9.74	25.94	25.24	0	0	0	0
		10	99.88	99.95	69.44	68	87.03	87.74	39.86	41.16	12.85	12.21	29.58	26.84	0	0	0	0
	270	1	47.33	45.19	20.46	19.94	41.97	39.67	5.86	6.23	5.36	5.53	14.6	13.71	0	0	0	0
		5	79.88	78.99	51.87	51.05	68.83	68.83	24.57	25.7	11.06	10.15	27.29	25.35	0	0	0	0
		10	100.83	100.95	71.07	69.54	87.23	89.05	40.8	41.53	13.59	11.89	30.27	28.01	0	0	0	0
12	150	1	61.67	63.97	32.05	31.45	53	55.73	11.34	10.1	8.32	7.88	20.34	20.97	0.34	0.35	0.38	0.38
		5	114.97	113.11	82.33	81.28	96.95	96.42	42.05	40.13	16.18	14.85	38.83	39.7	1.85	1.83	1.46	1.45
		10	145.33	144.29	113.23	112.21	123.22	121.96	66.89	64.24	18.22	18.48	43.29	44.92	3.89	3.84	3.05	3.04
	210	1	60.84	62.22	30.52	30.04	52.83	54.47	9.33	9.72	8.01	7.75	21.19	20.32	0	0	0	0
		5	110.33	109.82	79.13	78.3	93.66	95.28	40	39.42	16.67	14.54	39.12	38.88	0	0	0	0
		10	139.9	138.94	108.11	106.94	121.21	121	64.69	63.18	18.69	17.94	43.42	43.76	0	0	0	0
	270	1	61.32	64.45	31.52	31.19	53	56.68	10.12	10.16	8.32	7.77	21.4	21.03	0	0	0	0
		5	113.56	111.94	81.48	80.43	97.22	96.31	39.76	40.39	16.34	15.63	41.72	40.04	0	0	0	0
		10	139.7	139.04	111.91	110.34	120.96	120.23	66.43	62.61	18.73	18.8	45.48	47.74	0	0	0	0
14	150	1	82.8	81.74	42.06	41.12	70.96	69.99	15.41	14.91	7.83	7.76	20.17	19.82	4	3.99	6.48	6.4
		5	156.38	159.39	109.09	107.94	118.3	122.67	44.96	47.52	16.79	15.27	43.58	40.32	21.29	21.45	20.55	20.11
		10	214.91	213.98	160.33	158.68	149.93	150.75	69.13	72.48	20.13	18.56	52.35	47.74	44.85	44.67	38.85	38.47
	210	1	78.82	77.77	32.81	32.15	71.24	70.26	9.6	10.78	7.58	7.5	23.21	21.37	0	0	0	0
		5	133.37	133.02	85.91	84.94	116.95	118.21	39.79	41.02	16.43	14.81	46.12	43.91	0	0	0	0
		10	166.91	166.28	117.82	116.76	147.27	147.96	65.17	65.43	19.64	18.32	52.65	51.33	0	0	0	0
	270	1	80.56	79.85	35.02	34.58	72.8	72.37	11.57	13.36	7.76	7.48	23.45	21.22	0	0	0	0
		5	135.51	137.54	88.93	88.29	118.86	122.08	41.88	45.79	16.66	15.46	47.05	42.5	0	0	0	0
		10	170.05	168.5	123.53	121.05	149.93	150.39	65.51	71.94	20.12	18.11	58.01	49.11	0	0	0	0
Average			105.36	105.11	70.08	69.11	89.56	90.1	34.29	34.86	12.98	12.2	33.17	31.66	2.82	2.82	2.62	2.59

2.4.2.3 Analysis of Stochastic Arrivals (SA)

In Table 2.7, we report the results of computational experiments performed to assess the impact of different levels of Stochastic Arrivals (SA) on the efficiency appointment scheduling problem. These findings are key in addressing Question 3, as outlined at the beginning of Section 2.4. This analysis is instrumental in understanding the extent to which stochastic patient arrival times impact the overall effectiveness and optimization of healthcare scheduling and provide valuable insights for enhancing operational efficiency in clinical settings.

In this table, for all instances, we have considered the highest levels of Health State (HS3) and No-Shows (NS4) for all patients. For instances, we have reported the results obtained from the proposed model in "HS3-SA4-NS4" setting directly in the table. However, the solutions of instances with other stochastic arrival settings (i.e., SA0, SA1, SA2, and SA3) must be evaluated in the reference setting of "HS3-SA4-NS4", as discussed in the previous subsection. This is because we aim at assessing the importance of including different levels of patients' stochastic arrivals in the effectiveness of appointment scheduling.

The outcomes show that SA4, which combines both patient-and-time-dependent unpunctuality factors, significantly outperforms other stochastic arrival settings. This indicates that, by integrating the highest level of detail in unpunctuality, the model allocates appointment slots more effectively that leads to reduced costs and enhanced operational efficiency.

Settings SA2 and SA3 benefit from modeling two different aspects of uncertainty in patients' arrival times and therefore none of them is expected to dominate the other one in all instances. However, interestingly, our results indicate that in most instances SA2 dominates SA3 in terms of the total cost. This observation is reasonable considering that typical traffic patterns are expected to be the main factor affecting the punctuality of patients. However, it is conceivable that patient demographics might emerge as a significant factor influencing scheduling outcomes on a case-by-case basis. Also, our results indicate that SA2 and SA3 outperform SA1, which is a simplified version of both settings considering only one probability distribution for all patients and all time slots. Finally, as expected, we observe SA1 is superior to SA0 that assumes that all patient are punctual.

Table 2.7: Computational results for SA variations.

Instance information			Total Cost					Waiting Cost					Idle Cost					Overtime Cost				
No. of patients	Clinic time	Cost ratio	SA0	SA1	SA2	SA3	SA4	SA0	SA1	SA2	SA3	SA4	SA0	SA1	SA2	SA3	SA4	SA0	SA1	SA2	SA3	SA4
10	150	1	22.6	21.24	20.51	20.99	19.96	6.16	5.63	6.24	5.38	6.31	16.44	15.62	14.27	15.61	13.65	0	0	0	0	0
		5	58.6	55.21	52.18	54.99	51.08	37.64	33.66	28.85	34.1	25.41	20.95	21.55	23.32	20.88	25.66	0.01	0.01	0.01	0.01	0.01
		10	82.02	75.82	70.77	75.47	69.57	61.47	54.07	44.25	54.53	41.53	20.53	21.73	26.51	20.94	28.02	0.01	0.01	0.01	0.01	0.01
	210	1	22.16	20.78	19.77	20.38	19.31	5.59	5.2	5.89	5.3	5.59	16.58	15.58	13.88	15.08	13.72	0	0	0	0	0
		5	58.12	54.6	50.32	54.31	49.74	39.05	33.65	26.24	33.44	24.49	19.07	20.94	24.08	20.88	25.24	0	0	0	0	0
		10	79.67	75.59	69.57	75.26	68	56.68	55.26	42.89	55.02	41.16	22.98	20.34	26.68	20.23	26.84	0	0	0	0	0
	270	1	22.62	21.62	20.7	21.12	19.94	6.22	5.25	6.51	5.3	6.23	16.39	16.37	14.19	15.82	13.71	0	0	0	0	0
		5	59.1	55.48	52.25	55.46	51.05	37.11	34.24	28.27	33.98	25.7	21.99	21.24	23.98	21.48	25.35	0	0	0	0	0
		10	83.17	77.19	71.55	76.41	69.54	64.2	55.31	43.75	57.53	41.53	18.97	21.88	27.8	18.88	28.01	0	0	0	0	0
12	150	1	36.95	34.14	33.24	33.53	31.45	8.8	10.01	11.99	10.22	10.1	27.5	23.63	20.78	22.87	20.97	0.66	0.5	0.47	0.44	0.38
		5	92.3	87.27	83.1	85.88	81.28	54.44	49.6	43.6	46.96	40.13	36.43	36.24	38.03	37.52	39.7	1.43	1.43	1.48	1.4	1.45
		10	126.82	119.99	114.55	117.53	112.21	89.8	80.32	66.67	76.96	64.24	33.85	36.53	44.82	37.46	44.92	3.17	3.14	3.06	3.11	3.04
	210	1	35.46	32.66	31.05	31.88	30.04	8.42	9.27	10.47	8.57	9.72	27.04	23.38	20.58	23.31	20.32	0	0	0	0	0
		5	89.93	83.09	79.94	82.51	78.3	56.02	46.92	41.13	46.5	39.42	33.92	36.18	38.81	36.02	38.88	0	0	0	0	0
		10	120.64	113.53	108.98	112.93	106.94	89.33	79.04	66.17	76.44	63.18	31.31	34.49	42.81	36.49	43.76	0	0	0	0	0
	270	1	36.45	33.74	32.85	33.03	31.19	8.07	10.44	12.25	9.98	10.16	28.38	23.3	20.6	23.05	21.03	0	0	0	0	0
		5	90.43	85.34	81.36	83.93	80.43	54.46	49.65	41.8	47.13	40.39	35.98	35.7	39.56	36.8	40.04	0	0	0	0	0
		10	123.01	115.81	113.49	113.89	110.34	87.77	80.11	66.12	77.15	62.61	35.24	35.69	47.38	36.74	47.74	0	0	0	0	0
14	150	1	46.89	44.87	42.83	44.03	41.12	13.97	12.96	15.78	14.58	14.91	24.91	23.94	20.55	22.27	19.82	8.01	7.97	6.5	7.18	6.4
		5	120.01	117.69	110.24	114.25	107.94	63.04	53.06	47.85	53.67	47.52	37.01	43.85	42.27	40.57	40.32	19.95	20.78	20.12	20.01	20.11
		10	175.15	171.65	162.93	168.52	158.68	99.48	86.2	73.02	87.25	72.48	36.59	46.88	50.92	42.89	47.74	39.07	38.57	38.99	38.38	38.47
	210	1	38.01	35.96	33.72	35.01	32.15	7.11	8.04	12.05	8.69	10.78	30.9	27.92	21.67	26.31	21.37	0	0	0	0	0
		5	95.78	90.7	87.23	90.35	84.94	56.79	46.56	43.91	49.33	41.02	38.99	44.13	43.32	41.02	43.91	0	0	0	0	0
		10	132.12	126.34	119.3	125.91	116.76	94.15	81.64	66.25	82.54	65.43	37.97	44.71	53.05	43.37	51.33	0	0	0	0	0
	270	1	39.45	38.5	36.47	36.28	34.58	8.24	11.21	15.26	9.94	13.36	31.21	27.29	21.21	26.34	21.22	0	0	0	0	0
		5	97.85	93.93	89.92	92.62	88.29	57.29	46.37	46.46	47.08	45.79	40.56	47.57	43.46	45.54	42.5	0	0	0	0	0
		10	134.09	131.3	124.46	128.05	121.05	94.34	85.13	70.8	83.14	71.94	39.76	46.17	53.66	44.92	49.11	0	0	0	0	0
Average			78.5	74.56	70.86	73.54	69.11	46.88	41.79	36.46	41.52	34.86	28.94	30.09	31.78	29.4	31.66	2.68	2.68	2.62	2.61	2.59

We also observe that, an increase in the cost ratio results in higher costs across all SA levels. Similarly, an increase in the number of patients also leads to an increase in costs that highlights the challenges posed by larger patient volumes.

2.4.2.4 Analysis of No-Shows (NS)

In Table 2.8, we have reported the computational results for assessing the impact of different levels of no-show (NS) on the efficiency appointment scheduling problem. These findings contribute to addressing research Question 4 as outlined at the beginning of Section 2.4.

In this table, for all instances, we have considered the highest levels of Health State (HS3) and Stochastic Arrivals (SA4) for all patients. As discussed in subsections 2.4.2.2 and 2.4.2.3, we have evaluated the solutions of instances with NS0, NS1, NS2, and NS3 in the reference setting of "HS3-SA4-NS4". Also, the results of NS4 are directly reported as obtained from the proposed model.

The results indicate that NS4 significantly outperforms other no-show settings thanks to combining patient-and-time-dependent no-show factors. This outcome suggests that more detailed no-show information enables clinics to schedule appointments more effectively thus maximizing resource utilization. Additionally, similar to the SA analysis, we observe NS2 results in lower total cost than NS3 implying that time dependency plays a more critical role than patient dependency in no-shows. This analysis highlights that while our conclusions are generally applicable, variations might occur on a case-by-case basis, where patient demographics could emerge as a more critical factor in no-shows.

Notably, NS2 and NS3 demonstrate superior performance compared to NS1, which can be attributed to NS1's lack of consideration for both time-dependent and patient-dependent no-show probabilities. Furthermore, NS1 outperforms NS0 which shows the importance of acknowledging no-shows in the scheduling process. Across all levels of no-shows, an increase in the cost ratio leads to increased costs. Similarly, as the number of patients increases, so do the total costs, along with specific increases in waiting, idle, and overtime costs.

Table 2.8: Computational results for NS variations.

Instance information			Total Cost					Waiting Cost					Idle Cost					Overtime Cost				
No. of patients	Clinic time	Cost ratio	NS0	NS1	NS2	NS3	NS4	NS0	NS1	NS2	NS3	NS4	NS0	NS1	NS2	NS3	NS4	NS0	NS1	NS2	NS3	NS4
10	150	1	32.25	22.44	21.26	21.04	19.96	0.74	9.6	5.85	10.5	6.31	31.46	12.84	15.41	10.53	13.65	0.05	0	0	0	0
		5	141.46	71.2	58.65	57.16	51.08	1.06	58.64	31.21	44.24	25.41	140.35	12.55	27.43	12.92	25.66	0.05	0.01	0.01	0.01	0.01
		10	268.62	107.69	79.8	85.01	69.57	1.64	98.71	52.98	75.74	41.53	266.89	8.96	26.81	9.25	28.02	0.1	0.01	0.01	0.02	0.01
	210	1	32.03	22.02	20.71	21.11	19.31	0.58	10.15	5.66	11.69	5.59	31.44	11.88	15.05	9.42	13.72	0	0	0	0	0
		5	144.82	73.15	57.5	61.26	49.74	0.86	61.96	30.19	51.55	24.49	143.96	11.19	27.31	9.7	25.24	0	0	0	0	0
		10	273.58	115.26	79.95	91.73	68	1.09	108.59	52.63	85.8	41.16	272.49	6.67	27.32	5.93	26.84	0	0	0	0	0
	270	1	31.62	24.19	21.41	22.5	19.94	0.76	13.58	5.56	13.47	6.23	30.86	10.61	15.85	9.02	13.71	0	0	0	0	0
		5	142.58	82.97	58.93	66.13	51.05	1.06	75.29	30.86	57.82	25.7	141.51	7.68	28.07	8.31	25.35	0	0	0	0	0
		10	268.75	125.16	81.38	96.53	69.54	1.63	120.45	51.87	91.18	41.53	267.13	4.71	29.51	5.35	28.01	0	0	0	0	0
12	150	1	55.12	33.35	32.51	33.28	31.45	1.43	13.54	9.49	15.77	10.1	46.3	19.51	22.59	17.2	20.97	7.39	0.3	0.42	0.31	0.38
		5	231.98	94.9	85.85	93.24	81.28	2.56	70.3	41.32	72.31	40.13	205.91	23.06	43.12	19.36	39.7	23.51	1.54	1.41	1.57	1.45
		10	433.35	140.54	118.07	131.32	112.21	3.09	119.8	68	111.27	64.24	393.12	17.12	47.01	16.51	44.92	37.13	3.61	3.06	3.54	3.04
	210	1	49.55	32.74	30.78	32.62	30.04	0.85	15.05	8.6	17.34	9.72	48.7	17.69	22.17	15.28	20.32	0	0	0	0	0
		5	219.19	98.58	82.6	93.09	78.3	1.43	81.13	39.09	76.31	39.42	217.76	17.45	43.51	16.79	38.88	0	0	0	0	0
		10	415.59	146.7	113.83	134.07	106.94	1.75	134.31	67.3	121.77	63.18	413.84	12.4	46.54	12.3	43.76	0	0	0	0	0
	270	1	50.18	35.48	32.27	35.06	31.19	1.18	19.04	8.97	20.73	10.16	48.99	16.44	23.3	14.33	21.03	0	0	0	0	0
		5	220.63	103.71	84.7	101.17	80.43	1.85	87.29	40.2	87.14	40.39	218.77	16.43	44.49	14.03	40.04	0	0	0	0	0
		10	415.83	158.06	117.01	150.18	110.34	2.41	148.93	71.17	141.45	62.61	413.42	9.13	45.84	8.73	47.74	0	0	0	0	0
14	150	1	69.96	46.89	43.22	42.95	41.12	2.4	21.96	14.2	20.27	14.91	44.92	19.05	22.05	17.18	19.82	22.63	5.87	6.97	5.5	6.4
		5	282.79	130.99	115.73	118.98	107.94	4.11	82.29	46.8	76.11	47.52	192.48	29.18	48.75	23.54	40.32	86.2	19.52	20.18	19.33	20.11
		10	515.33	200.88	173.01	177.47	158.68	5.34	135.45	79.68	114.1	72.48	359.99	24.09	54.5	23.76	47.74	150	41.35	38.83	39.61	38.47
	210	1	49.24	35.87	33.62	34.56	32.15	1.15	14.77	9.48	16.75	10.78	48.03	21.09	24.14	17.8	21.37	0.05	0	0	0	0
		5	209.57	104.49	93.3	97.2	84.94	2.09	75.5	39.92	74.65	41.02	207.42	28.99	53.38	22.56	43.91	0.05	0	0	0	0
		10	399.66	162.42	130.23	138.34	116.76	2.76	144.55	71.26	119.7	65.43	396.79	17.86	58.97	18.64	51.33	0.1	0	0	0	0
	270	1	50.27	40.16	37.29	40.3	34.58	1.67	20.86	11.45	24	13.36	48.6	19.3	25.84	16.3	21.22	0	0	0	0	0
		5	212.56	109.79	95.99	105.1	88.29	2.78	82.87	43.21	85.58	45.79	209.78	26.93	52.78	19.52	42.5	0	0	0	0	0
		10	400.56	174.35	132.51	153.59	121.05	3.69	158.52	73.63	139.63	71.94	396.88	15.83	58.88	13.96	49.11	0	0	0	0	0
Average			208.04	92.37	75.26	82.78	69.11	1.92	73.45	37.43	65.81	34.86	193.99	16.25	35.21	14.38	31.66	12.12	2.67	2.63	2.59	2.59

2.4.2.5 Analysis of reminder systems

In this section, we analyze the impacts of effective communication with patients on the efficiency of the clinic's scheduling system particularly in terms of managing no-shows. We investigate whether sending reminders to patients to inform the clinic about their potential no-shows is economically viable and how it influences overall clinic efficiency. To conduct a thorough investigation, we assess various levels of π^{info} that represents the likelihood of patients revealing their no-shows in response to reminders. We perform this analysis for $\pi^{info} \in \{0,\ 0.2,\ 0.4,\ 0.6,\ 0.8, 1\}$ indicating different scenarios from "no response to reminders" to "absolute certainty in patient communication" regarding no-shows. By examining this spectrum, we aim to understand the effect of different degrees of patient engagement and communication effectiveness on the scheduling process. The primary aim is to address Research Question 5 outlined at the start of Section 2.4 which focus on how different extents of patient responsiveness to reminders influence the overall effectiveness of the scheduling strategy.

We analyze instances with 10, 12, and 14 patients to ensure that our findings are applicable across clinics with different sizes. In all these instances, we consider the highest level of information available for health state (HS), stochastic arrival (SA), and no-shows (NS) to carry out our analysis. The findings presented in Table 2.9 shed light on the potential enhancements in scheduling efficiency that can be realized through the implementation of successful patient communication strategies.

The results show a 23% reduction in the average total cost as the probability of patients informing the clinic about potential no-shows (π^{info}) increases from 0 to 1. This improvement is achieved mainly thanks to substantial decreases in total waiting and idle costs. This significant reduction highlights the importance of effective communication between patients and clinics. Our results show that sending reminders to patients to inform the clinic about their potential no-shows is a highly effective strategy in enhancing the overall efficiency of the healthcare system. Our study also reveals that reminder systems results in more significant savings in total, waiting, and idle costs in clinics with higher cost ratios.

The results also indicate that our model is computationally effective in providing practitioners with the opportunity to assess the economic feasibility of establishing a reminder system. This

Table 2.9: Computational results for information level variations.

Instance information			Average of Total Cost						Average of Waiting Cost						Average of Idle Cost						Average of Overtime Cost					
No. of patients	Clinic time	Cost ratio	0	0.2	0.4	0.6	0.8	1	0	0.2	0.4	0.6	0.8	1	0	0.2	0.4	0.6	0.8	1	0	0.2	0.4	0.6	0.8	1
10	150	1	19.96	19.28	18.44	17.35	15.92	13.84	6.31	6.29	5.89	5.77	5.54	5.23	13.65	12.99	12.55	11.58	10.38	8.61	0	0	0	0	0	0
		5	51.08	49.57	47.49	44.77	40.65	34.25	25.41	24.71	24.48	21.97	20.84	17.52	25.66	24.86	23.01	22.8	19.8	16.72	0.01	0.01	0.01	0.01	0.01	0.01
		10	69.57	67.65	64.98	61.34	55.94	46.61	41.53	40.2	38.56	34.98	32.71	27.52	28.02	27.43	26.4	26.35	23.22	19.08	0.01	0.01	0.01	0.01	0.01	0.01
	210	1	19.31	18.58	17.7	16.48	14.92	12.61	5.59	5.51	5.77	5.48	5.06	4.38	13.72	13.07	11.93	11.01	9.86	8.23	0	0	0	0	0	0
		5	49.74	48.06	45.91	42.96	38.48	31.28	24.49	24.02	23.49	21.33	18.91	15.82	25.24	24.05	22.42	21.63	19.57	15.46	0	0	0	0	0	0
		10	68	65.93	63.14	59.16	53.27	42.77	41.16	40.08	37.31	33.28	30.93	24.68	26.84	25.86	25.82	25.88	22.34	18.09	0	0	0	0	0	0
	270	1	19.94	19.28	18.46	17.34	15.93	14.01	6.23	6.1	5.72	5.76	5.59	5.42	13.71	13.17	12.74	11.57	10.34	8.6	0	0	0	0	0	0
		5	51.05	49.47	47.44	44.71	40.59	34.28	25.7	25.02	24.46	21.95	20.83	17.59	25.35	24.45	22.98	22.76	19.76	16.69	0	0	0	0	0	0
		10	69.54	67.65	64.95	61.31	55.91	46.58	41.53	39.62	38.56	34.97	32.7	27.66	28.01	28.03	26.39	26.33	23.2	18.92	0	0	0	0	0	0
12	150	1	31.5	30.8	30.3	28.9	27.6	25.7	10.1	10.4	10.7	10.1	9.82	9.62	21	20.1	19.2	18.5	17.4	15.8	0.4	0.4	0.4	0.4	0.4	0.4
		5	81.3	79.5	77.2	74.1	69.9	63.2	40.1	38.8	38.2	37.8	35.3	32.1	39.7	39.3	37.5	34.9	33.2	29.7	1.5	1.5	1.4	1.4	1.4	1.4
		10	112	110	107	102	96.3	86.6	64.2	62.3	60	57.9	54.8	49.9	44.9	44.8	43.5	41.4	38.5	33.9	3	3	3	3	2.9	2.8
	210	1	30	29.6	28.3	27.1	25.6	23.4	9.72	9.91	9.41	9.01	8.64	8.52	20.3	19.7	18.9	18.1	17	14.9	0	0	0	0	0	0
		5	78.3	76.1	73.9	70.1	65.3	57.7	39.4	38	36.2	35.4	33.1	29.3	38.9	38.1	37.7	34.7	32.2	28.5	0	0	0	0	0	0
		10	107	104	101	96.2	89.5	78.5	63.2	60.6	60.2	55.6	53	46	43.8	43.6	40.7	40.6	36.5	32.4	0	0	0	0	0	0
	270	1	31.2	31.5	30.1	29.1	27.5	25.5	10.2	12	10.4	11.5	9.92	8.9	21	19.5	19.7	17.6	17.6	16.6	0	0	0	0	0	0
		5	80.4	78.2	76	73.2	68.7	62	40.4	38.8	38	36	35.2	32.6	40	39.4	38	37.2	33.5	29.4	0	0	0	0	0	0
		10	110	108	105	100	94.7	84.1	62.6	64.4	60.1	60.5	53	50.6	47.7	43.7	44.8	39.9	41.7	33.5	0	0	0	0	0	0
14	150	1	41.1	41.3	39.9	38.7	37.4	35.7	14.9	15.7	14.9	14.3	13.6	13.1	19.8	19.3	18.9	18.4	17.8	16.6	6.4	6.3	6.2	6	6	6
		5	108	106	104	101	97.3	91.5	47.5	47.1	46.7	44.6	42.8	41	40.3	39.2	37.6	37	34.9	31.3	20	20	20	20	20	19
		10	159	157	153	149	143	134	72.5	69.5	67.6	67.3	63.9	59.3	47.7	48.9	47.7	43.9	42	38	38	38	38	38	38	37
	210	1	32.2	31.5	30.8	29.9	28.8	26.8	10.9	10.2	10.1	10	10.3	9.25	21.3	21.3	20.7	19.8	18.5	17.6	0	0	0	0	0	0
		5	84.9	83.2	80.8	77.8	73.5	66.9	41	40.8	40	36.5	35.6	34.3	43.9	42.5	40.8	41.3	37.9	32.6	0	0	0	0	0	0
		10	117	114	111	107	101	90.7	65.4	65.9	64.1	61.8	59.2	54.6	51.3	48.4	47	45.1	41.8	36.1	0	0	0	0	0	0
	270	1	34.6	34.3	33.7	32.6	31.3	30.1	13.4	12.2	12.3	12.2	11.8	13	21.2	22.1	21.4	20.4	19.5	17.1	0	0	0	0	0	0
		5	88.3	86.8	84.9	81.8	78.3	72.5	45.8	44.9	41.9	41.8	40	38.4	42.5	41.9	43	40.1	38.3	34.1	0	0	0	0	0	0
		10	121	119	116	112	106	98.3	71.9	69.6	64.6	66.7	61.9	57.4	49.1	49.1	51.3	45.6	44.4	40.9	0	0	0	0	0	0
Average			69.11	67.66	65.6	62.86	59.04	52.95	34.86	34.16	32.95	31.65	29.82	27.17	31.66	30.92	30.1	28.68	26.71	23.31	2.59	2.57	2.55	2.53	2.51	2.47

analysis is also applicable in determining the most effective communication methods (SMS, emails, phone calls) based on cost effectiveness and patient response rates.

2.4.2.6 Analysis of enhanced show-up rate

In this section, we analyze the effect of increasing patients' show-up rates in the efficiency of the appointment scheduling systems. The motivation behind this analysis is that clinics can increase patients' show-up rates through incentivization and penalty mechanisms. Thus, we introduce a new parameter α that represents the adjustment in no-show rates as in $\pi_{it} \coloneqq (1-\alpha)\,\pi_{it}$. The values of $\alpha > 0$ reflect having enhanced show-up rates, while $\alpha < 0$ represents a decreased likelihood of patient attendance, possibly due to negative past experiences. In our analysis, we consider different values of $\alpha \in \{-0.2, -0.1,\ 0,\ 0.1, 0.2\}$. Our approach involves comparing the results of the scheduling model with the "HS3-SA4-NS4" setting before and after adjusting the no-show rates. We have provided the results of this analysis in Table 2.10. These results are crucial as they provide a deep understanding of the implications of adjusting patient show-up probabilities.

According to Table 2.10, the extreme case of enhanced show up, $\alpha = 0.2$, results in around 10% improvement in the average total cost compared to the reference setting of $\alpha = 0$. This indicates that, by employing strategies such as incentivization, clinics can better utilize their resources, reduce idle times, and minimize the waiting time for patients.

On the other hand, the degraded show up rates in the case of $\alpha = -0.2$ results in an average deterioration of 13.58%. This observation highlights the criticality of enhancing patient experience through multifaceted approaches, such as cultivating a congenial environment within healthcare settings. By implementing strategies like incentivization, clinics can optimize resource utilization, diminish idle periods, and curtail patient waiting times and thus elevate the overall efficiency and satisfaction in healthcare delivery.

Table 2.10: Computational results for Show-up adjustment coefficient.

Instance information			Average of Total					Average of Waiting					Average of Idle					Average of Overtime				
			Show-up adjustment coefficient					Show-up adjustment coefficient					Show-up adjustment coefficient					Show-up adjustment coefficient				
No. of patients	Clinic time	Cost ratio	-0.2	-0.1	0	0.1	0.2	-0.2	-0.1	0	0.1	0.2	-0.2	-0.1	0	0.1	0.2	-0.2	-0.1	0	0.1	0.2
10	150	1	25.41	22.8	19.96	17.53	15.67	8.28	6.93	6.31	5.75	4.74	17.12	15.88	13.65	11.78	10.93	0	0	0	0	0
		5	64.5	58.71	51.08	45.42	40.9	33.95	30.89	25.41	22.57	21.17	30.54	27.82	25.66	22.84	19.71	0	0	0.01	0.01	0.02
		10	87.12	80.29	69.57	62	56.59	53.49	48.78	41.53	36.97	33.86	33.63	31.5	28.02	25	22.68	0	0.01	0.01	0.03	0.05
	210	1	24.91	22.28	19.31	16.76	14.74	8.18	6.64	5.59	5.43	4.03	16.72	15.64	13.72	11.34	10.71	0	0	0	0	0
		5	63.76	57.82	49.74	43.95	39.2	34.06	29.68	24.49	21.79	19.33	29.7	28.13	25.24	22.16	19.87	0	0	0	0	0
		10	86.24	79.17	68	60.04	54.36	52.54	47.15	41.16	35.58	30.52	33.7	32.02	26.84	24.47	23.85	0	0	0	0	0
	270	1	25.41	22.84	19.94	17.52	15.58	8.19	7.29	6.23	5.74	4.62	17.22	15.55	13.71	11.77	10.95	0	0	0	0	0
		5	64.56	58.72	51.05	45.37	40.86	33.66	30.71	25.7	22.56	21.15	30.9	28.01	25.35	22.81	19.71	0	0	0	0	0
		10	87.13	80.32	69.54	61.92	56.49	53.9	48.56	41.53	36.46	34.71	33.23	31.76	28.01	25.46	21.79	0	0	0	0	0
12	150	1	36	34.2	31.5	29.2	27.7	13.2	11.4	10.1	10.3	9.83	22.8	22.6	21	18.1	16.1	0	0.1	0.4	0.8	1.7
		5	89	86.8	81.3	76.4	73.1	49.1	44.5	40.1	36.7	35.1	39.7	41.7	39.7	36.6	32.6	0.2	0.6	1.5	3.1	5.5
		10	119	118	112	107	104	75.6	71.4	64.2	59.2	56.1	42.8	45.2	44.9	41.4	36.5	0.5	1.3	3	6.2	11
	210	1	34.9	33	30	26.9	24.5	11.4	10.5	9.72	8.5	7.54	23.5	22.5	20.3	18.4	16.9	0	0	0	0	0
		5	87.8	84.8	78.3	71	64.6	48.8	42.2	39.4	35.1	30.4	39	42.6	38.9	35.9	34.3	0	0	0	0	0
		10	117	115	107	97.9	89.3	74.3	69.6	63.2	56.7	52	42.8	45.6	43.8	41.2	37.2	0	0	0	0	0
	270	1	36.3	34.1	31.2	28.6	25.9	13.4	12.1	10.2	9.47	8.29	22.9	22	21	19.2	17.6	0	0	0	0	0
		5	89.7	86.7	80.4	74.1	67.9	48.6	44.3	40.4	35.3	32.4	41.1	42.4	40	38.8	35.5	0	0	0	0	0
		10	119	118	110	101	92.7	74.8	71	62.6	57.4	55.1	44.3	47	47.7	43.7	37.6	0	0	0	0	0
14	150	1	41.4	40.9	41.1	42.7	44.8	14.9	14.3	14.9	14.3	13.9	25.3	23.2	19.8	17.9	15.6	1.2	3.3	6.4	10	15
		5	106	106	108	117	128	53.9	49.5	47.5	44.5	44.1	48.5	46.8	40.3	37.3	32.1	3.7	9.3	20	35	52
		10	147	151	159	179	204	84	77.6	72.5	69	63.8	55.7	54.8	47.7	43	40.6	7.6	18	38	67	100
	210	1	39.5	36	32.2	28.8	25.9	13.1	11.3	10.9	9.22	7.74	26.3	24.7	21.3	19.6	18.2	0	0	0	0	0
		5	101	94.6	84.9	77.2	70.8	52.6	48.2	41	35.7	34.5	48.6	46.4	43.9	41.4	36.3	0	0	0	0	0
		10	138	130	117	107	98.1	83.6	75.9	65.4	62.3	54.6	54.6	54.2	51.3	44.5	43.5	0	0	0	0	0
	270	1	41.5	38.5	34.6	31.4	28.4	16.1	13.7	13.4	11.1	9.49	25.5	24.8	21.2	20.3	18.9	0	0	0	0	0
		5	104	97.5	88.3	81.3	75.5	52.3	45.7	45.8	38.7	39	51.5	51.8	42.5	42.6	36.4	0	0	0	0	0
		10	143	134	121	112	104	81.6	78.1	71.9	66.8	57.4	61.1	56	49.1	44.8	46.2	0	0	0	0	0
Average			78.5	74.88	69.11	65.14	62.36	42.5	38.82	34.86	31.6	29.09	35.51	34.84	31.66	28.97	26.38	0.49	1.22	2.59	4.56	6.89

2.5 Conclusion

Our research introduces a novel approach to the address the pressing challenges of healthcare appointment scheduling through an efficient stochastic model. This model adeptly navigates the complexities in patient scheduling by considering the intrinsic uncertainties tied to service times, patient unpunctuality, and no-shows. In addition, these uncertainties are characterized by their patient-and-time-dependent distributions which is a critical aspect that previous models have often oversimplified due to the modeling and solution complexity. We showed that our model is efficient in solving large-scale instances optimally within a reasonable computational time. The proposed model captured an exponential number of scenarios while maintaining a polynomial number of variables and constraints for the first time in this field. Our novel integration of uncertain variables at an individualized time-dependent level has revealed significant operational efficiencies, demonstrated by a notable reduction in total clinic costs. Specifically, we observed a comprehensive cost reduction of 34% attributed to the consideration of patient-dependent service times. Furthermore, the model's sensitivity to patient-and-time-dependent unpunctuality and no-show probabilities leaded to additional cost reductions of 12% and 67%, respectively. These findings highlight the critical importance of embracing the complexity of patient behavior patterns in scheduling models to optimize healthcare delivery and operational efficacy. Our findings highlight the vital importance of an optimized resource allocation, which maintains system efficiency without overburdening healthcare providers or compromising patient satisfaction.

We introduced personalized reminders as a strategic approach to reduce no-shows within our model framework that showed the profound impact of personalized communication in mitigating no-show rates. This consideration resulted in a further 23% reduction in total costs. This strategic insight highlights the transformative potential of using data-driven, patient-centric communication strategies to enhance clinic efficiency and patient engagement. Our model not only challenges the conventional paradigms of healthcare scheduling but also sets a new benchmark for integrating patient-specific data to drive operational improvements. The detailed understanding and application of stochastic variables, reflective of real-world patient behaviors, enhance the model's practical relevance and applicability across diverse healthcare settings. By considering a wider array of factors,

our model proved how an patient-and-time-dependent appointment scheduling approach can significantly enhance healthcare service efficiency and result in considerable cost savings and improved patient satisfaction.

Chapter 3

Stochastic weekly operating room planning with an exponential number of scenarios[1]

Abstract

In this paper, we consider a two-stage stochastic weekly operating room planning problem with an exponential number of scenarios. The objective function is to minimize the sum of the fixed opening cost of operating rooms and the expected overtime costs that are computed in the second stage. We propose a state-variable model to formulate the two-stage stochastic operating room planning problem and prove its validity. The main advantage of the proposed state-variable model is that it has a pseudo-polynomial number of variables and constraints that are significantly fewer than the number of variables and constraints in an equivalent scenario-based stochastic programming model. We improve the quality of the proposed model by developing an enhanced model that includes remarkably fewer variables and constraints. We also strengthen the model by developing several valid inequalities, including worst-case scenario and symmetry-breaking cuts. We carried out extensive computational experiments to evaluate the performance of the proposed model. The computational results show that the proposed model is capable of finding optimal solutions of instances with 50

[1]This paper is published in *Annals of Operations Research* in 2022 and has been cited 16 times as of March 2024.

surgeries and 1.55E+40 scenarios that is a significant improvement over the state-of-the-art models. The results revealed that the model finds feasible solutions with an average optimality gap of 0.78% for instances with 80 surgeries and 1.48E+64 scenarios.

3.1 Introduction

The main goal of healthcare managers is to provide high-quality services to patients in an economical manner. In recent decades, there has been an emergent interest in applying operations research methods to optimize service quality and cost in the healthcare sector. Among these applications, operating room planning is one of the most notable examples, which has attracted the attention of many researchers and practitioners in recent years (Cardoen et al., 2010). This is mainly because Operating Rooms (ORs) generate more than 70% of hospitals' total revenue (Bandi and Gupta, 2020; Guerriero and Guido, 2011). Therefore, optimizing the utilization of ORs will reduce costs and increase the revenue and utilization rate of the ORs, resulting in lower waiting times for patients.

Optimizing the utilization of ORs is a complex process in nature. One of the most critical factors that increase the complexity of operating room planning problems is the uncertainty in surgical times. In order to handle these uncertainties, scientists use different methodologies that we can categorize into three main groups. Most papers in the literature neglect the uncertainty of surgical durations and propose deterministic models (Erekat et al., 2020; Park et al., 2021; Ozkarahan, 2000; Ogulata and Erol, 2003; Fei et al., 2008, 2009; Roshanaei et al., 2017; Marques and Captivo, 2017; Mateus et al., 2017). In this category, we consider papers studying integrated operating room planning and scheduling problems (Naderi et al., 2021; Roshanaei and Naderi, 2021; Roshanaei et al., 2020; Marques et al., 2012; Vijayakumar et al., 2013; Molina-Pariente et al., 2015; Hashemi Doulabi et al., 2016; Roshanaei et al., 2017). The main downside of deterministic models is that the obtained plans may turn out to be infeasible or result in a significant unexpected overtime cost in practice due to ignoring the uncertainty of surgical times. Because of the same reason, many researchers studying other applications have proposed various methodologies to model the underlying uncertainties.

In the second category, researchers proposed stochastic models for operating room planning problems (Guo et al., 2021; Atighehchian et al., 2020; Zhang et al., 2020; Lamiri et al., 2008; Denton

et al., 2010; Hans et al., 2008; Min and Yih, 2010; Wang et al., 2014; Choi and Wilhelm, 2014; Jebali and Diabat, 2015; Freeman et al., 2015). Wang et al. (2014) presented a new mathematical model for an operating theater allocation problem under uncertain demand and surgery duration. The authors proposed a Sample Average Approximation (SAA) framework and a Generation-Based Heuristic (CGBH) to formulate the problem and solve it. They showed that the average gap of the proposed methodology is less than 5% in instances with up to 100 patients and 200 scenarios. Choi and Wilhelm (2014) proposed a mathematical model for capacity allocation of the operating rooms. In their computational experiments, they considered 10 surgery specialties and 10 operating rooms. The authors solved instances with up to 250 scenarios using a News-Vendor-based heuristic. Jebali and Diabat (2015) developed a two-stage stochastic programming model for an operating room planning problem. The authors used an SAA approach to solve the model. In the computational experiments, the authors generated 1000 scenarios and considered the scheduling of 30 patients. They reported an average optimality gap of 3% for 87.50% of the generated scenarios. Freeman et al. (2015) proposed a new stochastic model for operating theater scheduling under uncertainty. The authors considered uncertainties in the duration of the elective surgeries and also the arrival time of random emergency patients. They solved the model using a two-step method considering 15 scenarios and six operating rooms. Their proposed algorithm resulted in an average 2.3% to 5% gap for different categories of instances.

Atighehchian et al. (2020) presented a two-step stochastic model for operating room scheduling in a multi-resource environment and offered L-shaped-based solution methods to solve the problem. The authors solved several instances with at most 30 patients and 400 stochastic scenarios. Zhang et al. (2020) presented a new mathematical model for an advance surgery scheduling problem with multiple operating rooms and downstream units. They applied a Column-Generation-based Heuristic to solve the model. They solved the model with up to 80 patients and 150 scenarios for the lower bound problem in each iteration of SAA and obtained solutions with an average optimality gap of around 1%. Guo et al. (2021) proposed a model for the stochastic distributed operating room scheduling problem. They embedded the proposed model in an SAA and developed three versions of Logic-based Benders Decomposition (LBBD) to solve the model. The authors considered instances with at most 75 patients and 100 stochastic scenarios and found solutions with an average

optimality gap of 20%.

Compared to the works in the first category, the papers in the second category are more reliable and applicable in practice because of addressing the uncertainty by scenarios. However, in these works, the main issue is that the number of variables and constraints in proposed models are linear to the number of scenarios. Therefore, such models are solvable only when a few scenarios (e.g., 1000 scenarios) are available. We refer to "1000 scenarios" as "a few numbers" because it accounts only for $100 \times (\frac{1000}{4^{50}})$ =1.27E-25% of all possible scenarios in an operating room planning problem with 50 surgeries and 4 possible durations for each surgery.

In the last category, scientists presented robust optimization models to deal with uncertain surgical durations (Breuer et al., 2020; Shehadeh and Padman, 2021; Denton et al., 2010; Addis et al., 2016; Marques and Captivo, 2017; Neyshabouri and Berg, 2017; Wang et al., 2017). The main advantage of these models is that the obtained solutions are immunized against the worst-case scenarios. However, the obtained schedules can be too conservative against uncertainty and may show poor performance in practice as they do not consider probability distributions and ignore optimizing the expected objective function.

Recently, Hashemi Doulabi et al. (2020) proposed a general modeling approach for a class of two-stage stochastic optimization problems. In these problems, for a given first-stage solution, one can determine the optimal values of recourse decisions sequentially by inspection. Their proposed formulation, referred to as the state-variable model, is capable of modeling problems with an exponential number of scenarios without any need for sampling.

In this paper, we extend their model to formulate stochastic weekly operating room planning where the decision-maker must consider the deadlines of surgeries and assign surgeries to operating rooms over a week. We also present an enhanced model with significantly fewer variables and constraints than the primary model. Moreover, we develop some valid inequalities to improve the quality of the enhanced model. By extensive computational experiments, we demonstrate the significant effect of the proposed enhancements. Our computational results show that the proposed model optimally solves instances with up to 50 surgeries and 1.55E+40 scenarios. Our model also finds solutions with an average optimality gap of 0.78% for instances with 80 surgeries and 1.48E+64 scenarios. One of the main advantages of the proposed model is that it can easily formulate chance

constraints for restricting overtime periods. Hence, we study the computational behavior of the proposed model in the existence of chance constraints. The significance of our contribution to the literature is more evident by noting that the maximum number of scenarios considered in many operating room planning and scheduling problems does not exceed 1500 scenarios (Lamiri et al., 2008; Denton et al., 2010; Hans et al., 2008; Min and Yih, 2010; Wang et al., 2014; Choi and Wilhelm, 2014; Jebali and Diabat, 2015; Freeman et al., 2015).

The remainder of this paper is organized as follows. In Section 3.2, we present the problem definition and a classic scenario-based stochastic model to better understand the problem. In Section 3.3, we develop a primary state-variable model for the problem. In Section 3.4, we first propose an enhanced state-variable model, and then we develop several valid inequalities to further improve the model. In Section 3.5, we present some computational results, and finally, we present the concluding remarks and directions for future studies in Section 3.6.

3.2 Problem definition

In the weekly operating room planning problem, we must assign a set of surgeries, denoted by $\mathcal{I}$, to a set of identical operating rooms on workdays of a week. For each surgery i, we must respect a deadline dd_i, which means the assigned surgical day must be on or before day dd_i. The main difficulty of this operating room planning problem is that surgical times are stochastic. Therefore, if the sum of surgical times assigned to an operating room exceeds the standard available time in operating rooms, we must pay an overtime penalty proportional to the length of overtime. The objective function is to minimize the sum of fixed opening cost of operating rooms, which are determined in the first-stage model, and the expected overtime costs that we compute in the second stage after realization of uncertain surgical times.

To have a better understanding of the problem, we present a classic two-stage stochastic programming model obtained by extending the model proposed by Denton et al. (2010). In this model, we suppose that several stochastic scenarios with known probabilities are available. We introduce the sets, parameters, variables and the model as follows.

Sets:

$\mathcal{I}$: The set of surgeries.

$\mathcal{I}_d$: The set of surgeries with a deadline of larger than or equal to d.

$\mathcal{D}$: The set of days.

$\mathcal{R}_d$: The set of operating rooms available on day d.

$\mathcal{D}_i$: The subset of days $\mathcal{D}$ for which the deadline dd_i is respected ($\mathcal{D}_i = \{d \in \mathcal{D} | d \leq dd_i\}$).

Ω : The set of available uncertain scenarios.

Parameters:

c^f : The fixed cost of opening any operating room on any day.

c^o : The overtime cost per time unit.

L : The normal available time in each operating room.

M : The maximum available allowed overtime in each operating room.

$t_{i\omega}$: The duration of surgery i in scenario ω.

p_ω : The probability of realization of scenario ω.

Variables:

y_{dr} : 1 if we open operating room r on day d; 0 otherwise.

x_{dri} : 1 if we assign surgery i to operating room r on day d; 0 otherwise.

$o_{dr\omega}$: The overtime used in operating room *r* on day *d* in scenario ω.

$$\min \sum_{d\in\mathcal{D}} \sum_{r\in\mathcal{R}} c^f y_{dr} + \sum_{\omega\in\Omega} \sum_{d\in\mathcal{D}} \sum_{r\in\mathcal{R}_d} p_\omega c^o o_{dr\omega} \tag{20}$$

$$\sum_{d\in\mathcal{D}_i} \sum_{r\in\mathcal{R}_d} x_{dri} = 1 \qquad \forall i \in \mathcal{I} \tag{21}$$

$$x_{dri} \leq y_{dr} \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d, i \in \mathcal{I}_d \tag{22}$$

$$o_{dr\omega} \geq \sum_{i\in\mathcal{I}_d} t_{i\omega} x_{dri} - L \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d, \omega \in \Omega \tag{23}$$

$$o_{dr\omega} \leq M y_{dr} \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d, \omega \in \Omega \tag{24}$$

$$x_{dri} \in \{0,1\} \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d, i \in \mathcal{I}_d \tag{25}$$

$$y_{dr} \in \{0,1\} \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d \tag{26}$$

$$o_{dr\omega} \geq 0 \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d, \omega \in \Omega \tag{27}$$

In Model (20)-(27), the objective function computes the sum of opening cost of operating rooms and expected cost of overtimes. Constraint (21) assigns each surgery to an operating room on a day. Constraint (22) implies that x_{dri} can take 1 only if y_{dr} is equal to 1. Constraint (23) computes the amount of overtimes in each operating room over a week. Constraint (24) restricts the maximum available overtime in each operating room. Constraints (25)-(26) state that x_{dri} and y_{dr} are binary variables. The main issue with the classic Model (20)-(27) is that it is a scenario-based model that can solve instances with only a few thousands of scenarios. However, as discussed in Section 3.1, real-world operating room planning and scheduling problems include a significantly larger number of scenarios. The other shortcoming with the above model is that it is inflexible in modeling chance constraints. In the real world, practitioners are interested in controlling the probability of using overtime in each operating room. However, the above model is not able to address these issues. The model we propose in the next section is capable of addressing these issues.

3.3 Proposed state-variable model

We propose the following model for the weekly operating room planning problem defined in Section 3.2. In this model, we suppose that surgical durations are independent of each other, and for each surgery, a separate set of uncertain scenarios are available. We also assume that possible values of stochastic surgical times are multiples of a time unit τ that we set to 5 minutes. In the following, except to Ω, p_ω, and $o_{dr\omega}$, we keep all other sets, variables, and parameters as defined in Section 3.2. Moreover, we introduce some additional notations as follows.

Sets:

Ω_i : The set of available scenarios for the surgical time of surgery i. Using this set, we define the set of scenarios for all surgeries as $\Omega = \Omega_1 \times \Omega_2 \times ... \times \Omega_{|\mathcal{I}|}$.

$\mathcal{I}_d^0$: $\mathcal{I}_d^0 = \mathcal{I}_d \cup \{0\}$. Index 0 refers to a dummy surgery with a trivial duration $t_0 = 0$ that is scheduled at the beginning of each operating room.

$\mathcal{T}_i$: The set of all possible values for the cumulative time used by any subset of surgeries 1 to i assigned to the same operating rooms, i.e.,

$$\mathcal{T}_i = \Big\{0, \tau, 2\tau, \ldots, \min\Big(\sum_{\substack{i' \in \mathcal{I}: \\ i' \leq i}} \max_{\omega \in \Omega_{i'}} (t_{i'\omega}), L + M\Big)\Big\}.$$

$\mathcal{T}_i'$: The set of all possible values for the cumulative used time by any subset of surgeries 1 to i assigned to the same operating rooms provided that there is enough remaining operating room time to schedule surgery $i + 1$ such that this surgery finishes before the end of the overtime period, i.e.,

$$\mathcal{T}_i' = \Big\{0, \tau, 2\tau, \ldots, \min\Big(\sum_{\substack{i' \in \mathcal{I}: \\ i' \leq i}} \max_{\omega \in \Omega_{i'}} (t_{i'\omega}), L + M - \max_{\omega \in \Omega_{(i+1)}} (t_{(i+1)\omega})\Big)\Big\}.$$

The difference between $\mathcal{T}_i$ and $\mathcal{T}_i'$ is that, in $\mathcal{T}_i$ we ignore addition of the future surgeries $i + 1$ to $|\mathcal{I}|$ in the definition of this set, while we consider addition of surgery $i + 1$ to the operating room in definition of $\mathcal{T}_i'$ and make sure that there will enough time available for performing this surgery.

Parameters:

c_{it} : The expected overtime cost corresponding to surgery i if it starts at time t. We compute it by $\mathop{\mathbb{E}}_{\omega \in \Omega_i} \big[c^o \max(0, t + t_{i\omega} - \max(t, L))\big]$.

Variables:

y_{dr}, x_{dri} As defined in Section 3.2.

z_{drit}^1 : The probability that the total surgical times of those surgeries 1 to i allocated to operating room r on day d is equal to t and we have also assigned surgery $(i + 1)$ to the same operating room (i.e., $x_{dr(i+1)} = 1$).

z^0_{drit} : The probability that the total surgical time of those surgeries 1 to i allocated to operating room r on day d is equal to t and we have not assigned surgery $(i+1)$ to the same operating room (i.e., $x_{dr(i+1)} = 0$).

We formulate the weekly operating room planning problem as follows.

$$\min \sum_{d\in\mathcal{D}} \sum_{r\in\mathcal{R}_d} c^f y_{dr} + \sum_{d\in\mathcal{D}} \sum_{r\in\mathcal{R}_d} \sum_{i\in\mathcal{I}^0_d\backslash|\mathcal{I}|} \sum_{t\in\mathcal{T}'_i} c_{(i+1)t} z^1_{drit} \tag{28}$$

$$(21)-(22),(25)-(27) \tag{29}$$

$$\sum_{t\in\mathcal{T}'_i} z^1_{drit} = x_{dr(i+1)} \qquad \forall d\in\mathcal{D}, r\in\mathcal{R}_d, i\in\mathcal{I}^0\backslash|\mathcal{I}| \tag{30}$$

$$\sum_{t\in\mathcal{T}_i} z^0_{drit} = 1 - x_{dr(i+1)} \qquad \forall d\in\mathcal{D}, r\in\mathcal{R}_d, i\in\mathcal{I}^0\backslash|\mathcal{I}| \tag{31}$$

$$\mathbb{1}_{(i\neq|\mathcal{I}|\wedge t\in\mathcal{T}'_i)} z^0_{drit} + z^1_{drit} = \mathbb{1}_{(t\in\mathcal{T}_{(i-1)})} z^0_{dr(i-1)t} + \sum_{\substack{t'\in\mathcal{T}'_{(i-1)}:\\ t'\leq t}} z^1_{dr(i-1)t'} Pr(t'+t_{i\omega}=t) \quad \forall d\in\mathcal{D}, r\in\mathcal{R}_d, i\in\mathcal{I}, t\in\mathcal{T}_i \tag{32}$$

$$0 \leq z^0_{drit} \leq 1 \qquad \forall d\in\mathcal{D}, r\in\mathcal{R}_d, i\in\mathcal{I}^0, t\in\mathcal{T}_i \tag{33}$$

$$0 \leq z^1_{drit} \leq 1 \qquad \forall d\in\mathcal{D}, r\in\mathcal{R}_d, i\in\mathcal{I}^0\backslash|\mathcal{I}|, t\in\mathcal{T}'_i \tag{34}$$

$$x_{dri} = 0 \qquad \forall i\in\mathcal{I}, d\notin\mathcal{D}_i, r\in\mathcal{R}_d \tag{35}$$

In Model (28)-(35), the first part of the objective function represents the opening cost of operating rooms, and the second part computes the expected overtime cost. Constraint (29) is the set of first stage constraints as defined in Model (20)-(27). Constraints (30)-(31) are the linking constraints between first-stage variables $x_{dr(i+1)}$ and the state-variables z^0_{drit} and z^1_{drit}. Constraint (30) implies that if $x_{dr(i+1)}$ is equal to 1 (or 0), then the sum of left-hand side state variables z^1_{drit} must be equal to 1 (or 0). Similarly, Constraint (31) enforces the sum of left-hand side state variables z^0_{drit} must be equal to 1 (or 0) if $x_{dr(i+1)}$ is equal to 0 (or 1). We note that these constraints are valid with respect to the definitions of variables z^1_{drit} and z^0_{drit}. We refer to constraint (32) as the transition constraint that links the state variables of surgery i, (i.e., z^0_{drit} and z^1_{drit}) to state variables

for surgery $i-1$, (i.e., $z^0_{dr(i-1)t}$ and $z^1_{dr(i-1)t}$). In this constraint, $\mathbb{1}_{(.)}$ is an indicator that is equal to 1 if condition $(.)$ is satisfied, and 0 otherwise. Both sides of constraint (32) independently represent the probability that the total surgical times of those surgeries 1 to i assigned to operating room r on day d is equal to t. With respect to the definition of z^0_{drit} and z^1_{drit}, it is clear that the left-hand side computes this probability correctly. Moreover, the right-hand side of constraint (32) properly calculate this probability value by conditional probability relations based on the probabilities $z^0_{dr(i-1)t}$ and $z^1_{dr(i-1)t}$. Considering constraints (30)-(31), either z^0_{drit} or z^1_{drit} will be equal to the probability value computed on the right-hand side of constraint (32).

Constraints (33)-(34) enforce the bounds of variables z^0_{drit} and z^1_{drit}. Constraint (35) implies that if the deadline for surgery i is before day d, then we cannot schedule this surgery on this day. We cannot remove constraint (35) from the model because all variables x_{dri} appear in constraints (30)-(31).

To have an idea about why the proposed model is correct, it is enough to note that for $i = 0$, constraints (30)-(31) set $z^0_{dr00} = 1$ and $z^1_{dr00} = 0$ or $z^0_{dr00} = 0$ and $z^1_{dr00} = 1$ that are valid relations because the dummy surgery 0 starts at time 0 in all operating rooms. Then, constraints (30)-(31) for $i = 1$, and constraint (32) for $i = 0$ guarantees the proper computation of values z^0_{dr1t} and z^1_{dr1t}. Following the same way, by induction we can prove that all values of z^0_{drit} and z^1_{drit} are correctly computed for all $i \in \mathcal{I}^0 \backslash |\mathcal{I}|$. This is the basis of the formal proof that we have presented in Appendix C for the validity of Model (28)-(35)

Theorem 2. *Model (28)-(35) is valid for the weekly operating room planning problem.*

Proof. Appendix C. □

An advantage of the proposed Model (28)-(35) over the classic Model (20)-(27) is that we can easily enforce chance constraint (36) to control the probability of using overtime in operating rooms.

$$\sum_{t \in \mathcal{T}_i : t > L} z^0_{drit} \leq \alpha \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d, i \in \mathcal{I}^0, t \in \mathcal{T}_i \tag{36}$$

In constraint (36), the left-hand side of the inequality computes the probability of using overtime periods. Parameter α represents the maximum probability of using overtime in an operating room.

3.4 Enhancements

We present two enhancements to improve the efficiency of Model (28)-(35). The idea of the first one is to reformulate Model (28)-(35) by eliminating some redundant state-variables z^0_{drit} and z^1_{drit} and their corresponding transition constraint (32). As the second enhancement, we develop some valid inequalities.

3.4.1 Enhanced model

As explained in Section 3.3, we cannot simply remove constraint (35) because all variables x_{dri} including those fixed to zero by constraint (35), appear in constraints (30)-(31). The main issue of having redundant variables x_{dri} $i \in \mathcal{I}, d \in \mathcal{D}_i, r \in \mathcal{R}_d$ is not just about them, but it is about the huge number of corresponding variables z^0_{drit} and z^1_{drit} respectively defined for $i \in \mathcal{I}, d \notin \mathcal{D}_i, r \in \mathcal{R}_d, t \in \mathcal{T}_i$ and $i \in \mathcal{I}, d \notin \mathcal{D}_i, r \in \mathcal{R}_d, t \in \mathcal{T}'_i$. The idea of the modification that we present here is to remove all variables x_{dri}, z^0_{drit} and z^1_{drit} defined for $i \in \mathcal{I}, d \notin \mathcal{D}_i, r \in \mathcal{R}_d$. We suppose that on each day $d \in \mathcal{D}_i$, surgeries in $\mathcal{I}^0_d$ are ordered lexicographically. We define n_{id}, p_{id} and l_d as follows.

Parameters:

- n_{id} : The surgery in $\mathcal{I}^0_d$ after surgery i considering the lexicographic order.
- p_{id} : The surgery in $\mathcal{I}^0_d$ before surgery i considering the lexicographic order.
- l_d : The last surgery in $\mathcal{I}^0_d$ considering the lexicographic order.

We also respectively replace sets $\mathcal{T}_i$ and $\mathcal{T}'_i$ with $\mathcal{T}_{id}$ and $\mathcal{T}'_{id}$ as defined below.

Sets:

$\mathcal{T}_{id}$: The set of all possible values for the cumulative time used by any subset of surgeries 1 to i in $\mathcal{I}_d$ assigned to the same operating rooms, i.e.,

$$\mathcal{T}_{id} = \Big\{0, \tau, 2\tau, ..., \min\Big(\sum_{\substack{i' \in \mathcal{I}_d: \\ i' \leq i}} \max_{\omega \in \Omega_{i'}} (t_{i'\omega}), L + M\Big)\Big\}.$$

The difference of $\mathcal{T}_{id}$from $\mathcal{T}_i$ is that $i' \in \mathcal{I}$ is replaced with $i' \in \mathcal{I}_d$ under the summation.

$\mathcal{T}'_{id}$: The set of all possible values for the cumulative times used by any subset of surgeries 1 to i in $\mathcal{I}_d$ assigned to the same operating rooms provided that there is enough remaining operating room time to schedule the next surgery n_{id} such that this surgery finishes before the end of the overtime period, i.e.,

$$\mathcal{T}'_{id} = \Big\{0, \tau, 2\tau, \ldots, \min\Big(\sum_{\substack{i' \in \mathcal{I}_d: \\ i' \leq i}} \max_{\omega \in \Omega_{i'}} (t_{i'\omega}), L + M - \max_{\omega \in \Omega_{(i+1)}} (t_{(n_{id})\omega})\Big)\Big\}.$$

The difference of $\mathcal{T}'_{id}$ from $\mathcal{T}'_i$ is that $i' \in \mathcal{I}$ is replaced with $i' \in \mathcal{I}_d$ under the summation.

Based on the above notation, we reformulate model (28)-(35) as follows.

$$\min \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}_d} c^f y_{dr} + \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}_d} \sum_{i \in \mathcal{I}_d^0 \backslash |\mathcal{I}|} \sum_{t \in \mathcal{T}'_{id}} c_{(i+1)t} z^1_{drit} \tag{37}$$

$$(21) - (22), (25) - (26) \tag{38}$$

$$\sum_{t \in \mathcal{T}'_{id}} z^1_{drit} = x_{dr(n_{id})} \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d, i \in \mathcal{I}_d^0 \backslash l_d \tag{39}$$

$$\sum_{t \in \mathcal{T}_{id}} z^0_{drit} = 1 - x_{dr(n_{id})} \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d, i \in \mathcal{I}_d^0 \backslash l_d \tag{40}$$

$$z^0_{drit} + \mathbb{1}_{(i \neq l^t_d \in \mathcal{T}'_{id})} z^1_{drit} = \mathbb{1}_{(t \in \mathcal{T}_{(p_{id})d})} z^0_{dr(p_{id})t} + \sum_{\substack{t' \in \mathcal{T}'_{(p_{id})d}: \\ t' \leq t}} z^1_{dr(p_{id})t'} Pr(t' + t_{i\omega} = t) \quad \forall d \in \mathcal{D}, r \in \mathcal{R}_d, i \in \mathcal{I}_d, t \in \mathcal{T}_{id}[-5pt] \tag{41}$$

$$0 \leq z^0_{drit} \leq 1 \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d, i \in \mathcal{I}^0, t \in \mathcal{T}_{id} \tag{42}$$

$$0 \leq z^1_{drit} \leq 1 \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d, i \in \mathcal{I}^0 \backslash l_d, t \in \mathcal{T}'_{id} \tag{43}$$

Model (37)-(43) is very similar to the Model (28)-(35). The main difference is that we replaced sets $\mathcal{T}_i$ and $\mathcal{T}'_i$, and surgery indices $(i+1)$, $(i-1)$, and $|\mathcal{I}|$ by $\mathcal{T}_{id}$, $\mathcal{T}'_{id}$, n_{id}, p_{id}, and l_d, respectively. We also dropped all redundant variables z^0_{drit} and z^1_{drit} defined for $i \in \mathcal{I}, d \in \mathcal{D}_i, r \in \mathcal{R}_d$ and their corresponding transition constraints (32). In fact, in constraint (41), we link variables z^0_{drit} and z^1_{drit} to $z^0_{dr(p_{id})t}$ and $z^1_{dr(p_{id})t'}$. In constraint (41), a surgery p_{id} may or may not have been scheduled on day d. However, in Constraint (32) in the previous model we know that, if $(i-1) \in \mathcal{I}_d$, surgery $(i-1)$ is definitely not scheduled on day d. Moreover, we can rewrite the chance constraint (36) as follows.

$$\sum_{t \in \mathcal{T}_{id}: t > L} z^0_{dr(l_d)t} \leq \alpha \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d, i \in \mathcal{I}^0, t \in \mathcal{T}_i \tag{44}$$

The difference between the original model (20)-(27) and the enhanced model (37)-(43) is that in the later one we have removed redundant indices based on the deadlines of the surgeries. Therefore, the validity of the enhanced model (37)-(43) originates from the correctness of the original model (20)-(27) as proven in Appendix C.

3.4.2 Valid inequalities

In this subsection, we develop valid inequalities to improve the efficiency of the proposed model. The first two valid inequalities rely on the worst-case realization of surgical times. The following three valid inequalities are symmetry-breaking constraints.

3.4.2.1 Worst-case scenario valid inequalities

The first valid inequality is a knapsack inequality for the worst-case scenarios of surgical times.

$$\sum_{i \in \mathcal{I}_d} \max_{\omega \in \Omega_i} (t_{i\omega}) x_{dri} \leq L + M \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d \tag{45}$$

Valid inequality (45) implies that for the worst-case scenario, where surgical times take their maximum possible values, the total surgical times in each operating room on each day cannot exceed the sum of regular operating room time and the maximum allowed overtime. This constraint is a valid inequality and not an essential constraint to the model because in Model (37)-(43), we have discretized the available time into a limited number of time units. Therefore, the constraint on the maximum available time in operating rooms is implicitly satisfied. However, enforcing this constraint explicitly by Constraint (45) would be more effective.

The subsequent valid inequality enforces a lower bound on the minimum number of required operating rooms during the planning horizon for the worst-case scenarios of surgical times.

$$\sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}_d} y_{dr} \geq \left\lceil \frac{\sum_{i \in \mathcal{I}} \max_{\omega \in \Omega_i} (t_{i\omega})}{L + M} \right\rceil \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d \tag{46}$$

In (46), the left-hand side is equal to the number of opened operating rooms during the planning horizon, and the right-hand side calculates a lower bound on this number for the worst-case realization of scenarios for surgical times.

3.4.2.2 Symmetry breaking valid inequalities

The following valid inequalities guarantee that the Model (37)-(43) does not present the same weekly operating room plan by more than one solution. These valid inequalities are inspired by those developed by Denton et al. (2010). The first set of symmetry-breaking valid inequality is as follows.

$$y_{dr} \geq y_{d(r+1)} \qquad \forall d \in \mathcal{D}, r \in \mathcal{R}_d \backslash \{|\mathcal{R}_d|\} \tag{47}$$

$$y_{d|\mathcal{R}_d|} \geq y_{(d+1)1} \qquad \forall d \in \mathcal{D} \backslash \{|\mathcal{D}|\} \tag{48}$$

Valid inequality (47) states that among all operating rooms available on any day $d \in \mathcal{D}$, we must open the one with the smallest index before the other one. This valid inequality ensures that if the decision-maker decides to use fewer than $|\mathcal{R}_d|$ operating rooms on the day d, then the same daily schedule cannot be presented by swapping the schedules of opened operating rooms with those of the closed ones. Moreover, valid inequality (48) implies that there is not any advantage in using operating rooms on the day $d + 1$ or later days, while there are unused operating rooms on the current day d.

Valid inequalities (47) and (48) partially break symmetry in the model. We can still find the same weekly operating room plan by swapping the schedules of opened operating rooms on the same day. Therefore, to avoid this symmetry, we use the following constraint that enforces surgery i to be scheduled in one of the operating rooms 1 to i within the planning horizon.

$$\sum_{d \in \mathcal{D}_i} \sum_{\substack{r \in \mathcal{R}_d: \\ r \leq i}} x_{dri} = 1 \qquad \forall i \in \mathcal{I} \tag{49}$$

Constraint (49) is valid because it is always possible to satisfy it by swapping the operating rooms' schedules on a day such that surgery i is scheduled in one of the operating rooms 1 to i on that day. Considering the deadlines of surgeries, we can improve the constraint (49) as follows.

$$\sum_{d \in \mathcal{D}_i} \sum_{\substack{r \in \mathcal{R}_d: \\ r \leq o_{id}}} x_{dri} = 1 \qquad \forall i \in \mathcal{I} \tag{50}$$

In valid inequality (50), o_{id} represents the order of surgery i in the set $\mathcal{I}_d$ that is ordered lexicographically. This valid inequality implies that if we cannot schedule m surgeries with indices $i' < i$ on day d because of their earlier deadlines, then it is possible to restrict the potential operating rooms for surgery i on day d to operating rooms 1 to $i - m$. This is valid because we can obtain such a daily schedule by swapping the operating rooms' schedules on a day.

The last symmetry breaking constraint is slightly more complicated than the previous ones. Therefore, we explain it in more detail. As suggested by valid inequality (50), if we want to schedule surgery i on the day d, we can restrict the potential operating rooms for this surgery to rooms 1 to o_{id}. As we discussed before, this restriction eliminates considerable symmetry in the model. However,

in some cases, we may obtain the same daily schedule by swapping the schedule of operating rooms. To explain this issue, let us suppose that we have scheduled surgery i on day d with $o_{id} = 10$, and among surgeries j with $o_{jd} < 10$ only surgery i' with $o_{i'd} = 1$ is scheduled on day d too. Valid inequality (50) suggests that we must schedule surgery i in one of the operating rooms 1 to 10. However, since other surgeries j with $o_{jd} < 10$ and $j \neq i'$ are not scheduled on this day, we can obtain identical solutions by swapping the schedule of the operating room, including surgery i by that of any operating room 2 to 10. Here, we saved operating room 1 for surgery i'. Therefore, to address this symmetry, we propose the following constraint.

$$\sum_{\substack{r' \in \mathcal{R}_d: \\ r \leq r' \leq o_{id}}} x_{dr'i} \leq \sum_{\substack{i' \in \mathcal{I}_d: \\ o_{i'd} \geq r-1 \& \\ o_{i'd} \leq o_{id}-1}} x_{d(r-1)i'} \qquad \forall d \in \mathcal{D}, i \in \mathcal{I}_d, r \in \mathcal{R}_d \backslash 1 : r \leq o_{id} \qquad (51)$$

Valid inequality (51) states that if we schedule surgery i on day d in operating room r or any other operating room with a larger index (i.e., $\sum_{\substack{r' \in \mathcal{R}_d: \\ r \leq r' \leq o_{id}}} x_{dr'i} = 1$), then at least one of the surgeries with $o_{i'd} \geq r-1$ and $o_{i'd} \leq o_{id} - 1$ must have be scheduled in operating room $r-1$, i.e., $\sum_{\substack{i' \in \mathcal{I}_d: \\ o_{i'd} \geq r-1 \& \\ o_{i'd} \leq o_{id}-1}} x_{d(r-1)i'} \geq 1$. On the right-hand side of (51), we have restricted surgeries $i' \in \mathcal{I}_d$ to those with $o_{i'd} \geq r-1$, because other ones with $o_{i'd} < r-1$ cannot be scheduled in operating room $r-1$ with respect to constraint (50).

3.5 Computational results

Here, we present the computational results on the stochastic weekly operating room planning problem. We used CPLEX to solve the MIP models. We run all numerical experiments on a computer with two Intel Xeon X5650 CPUs running at 2,67 GHz. We considered a time limit of 24 hours.

3.5.1 Instance generation

In all instances, we consider five days during the planning horizon. We considered having 8 hours of regular operating room times and a maximum of 2 hours overtime. We generated the deadlines of surgeries by an integer uniform distribution of $U[1, 5]$. We also set the maximum number of operating rooms on each day to the right-hand side of the constraint (46). This value is a quite reasonable

upper bound for the daily number of operating rooms and makes the problem significantly difficult because it results in many variables and constraints in proposed models. In addition, as suggested in Denton et al. (2010), we considered 4437$ as the opening cost of operating rooms. We set the per-minute cost of overtime to $\{1.5, 1.75, 2, 4\}$ times of the per-minute cost of regular operating room time. We refer to this ratio as the overtime factor (OF). Moreover, we set the number of patients to $n = \{20, 30, 40, 50, 60, 70, 80, 90\}$. For each combination of OF and n, we generated 10 instances that resulted in a total number of 320 instances. We generated surgical times from lognormal distributions provided in Table 1 of Vinden et al. (2016), which studied a real case in a teaching hospital in Ontario, Canada. In order to use the lognormal distributions, we considered a truncated version, taking into accounting 99% of the whole probability. Moreover, as our model handles discrete surgical times, we approximated the modified lognormal distributions by discrete probability distributions with a constant step of 5 minutes for surgical times.

3.5.2 Results

In the following computational experiments, we aim at answering the following questions:

(1) How does Model (37)-(43) perform compared to the classic stochastic programming model (20)-(26)?

(2) Are the proposed enhancements in Section 3.4 effective in improving the performance of the proposed model? In other words, how does Model (37)-(43) perform compared to Model (28)-(35)?

(3) What is the largest size of operating room planning problems that we can solve by the proposed model (37)-(43)? How does the model perform on instances with different values of overtime factor (OF)?

(4) How does the proposed Model (37)-(43) perform in the presence of chance constraint (44)?

In all the following tables, each row presents the average computational results for 10 instances. Table 3.4 reports the results of the experiments that we conducted on instances with $OF = 1.5$ to answer the first question. In this table, we compare the classic Model (20)-(27) to Model (37)-(43)

enhanced by valid inequalities. Since the number of scenarios in the classic Model (20)-(27) increases exponentially in the number of scenarios for each surgery, we modified the original instances by considering only 2 equiprobable scenarios for each surgical duration. We randomly generated these two scenarios for each surgery from the lognormal distributions in Table 3.4 of (Vinden et al., 2016). We note that this modification is just because of the inefficiency of the stochastic programming model (20)-(26) in modeling problems with exponential scenarios. For all other computational results presented in Tables 3.5-3.7, we considered the original instances with discretized lognormal distributions, as explained in Section 5.1. In Table 3.4, Column "Instance Info." gives some information about the generated instances. Under this column, "$|\mathcal{I}|$" and "$|\Omega|$" respectively present the number of surgeries and the number of scenarios in the stochastic programming model (20)-(26). Columns "Proposed model with enhancements" and "Stochastic programming model" respectively provides computational results of models (37)-(43) and (20)-(26). Under these two columns, we have the following columns: 1) V : The number of variables, 2) C : The number of constraints, 3) N: The number of nodes exploited in the branch-and-bound tree, 4) *Time*: Computational time of the models in seconds, 5) *Gap(%)*: Optimality gap that we compute by $Gap = 100(UB-LB)/LB$, 6) *OSI*: The number of optimally solved instances out of 10 instances reported in each row of the table.

As we can see in Table 3.4, the classic Model (20)-(27) can barely solve instances with up to 16 surgeries. This model cannot even find any feasible solution for instances with 20 or more surgeries. However, the proposed Model (37)-(43) optimally solves instances with 20 surgeries. For other instances, it finds feasible solutions with an optimality gap of less than 1.42%. The poor performance of the stochastic programming model is due to the exponential number of scenarios that this model cannot handle, while the proposed Model (37)-(43) can quickly solve instances with exponential scenarios.

Table 3.4 also shows that Model (37)-(43) is significantly faster than the classic Model (20)-(27). For instance, the average computational time of the classic Model (20)-(27) in instances with 14 surgeries is 2816 seconds, while the enhanced Model (37)-(43) solved the same instances in an

Table 3.4: Comparison of the enhanced Model (37)-(43) with the classic stochastic programming model (20)-(26) for instances with OF =1.5.

Instance Info.		*Classic Model (20)-(27)*						*Enhanced Model (37)-(43)*					
$\mathcal{I}$	Ω	V	C	N	$Time$	$Gap(\%)$	OSI	V	C	N	$Time$	$Gap(\%)$	OSI
10	1.02E+03	10309	10338	27	10	0.00	10	8119	4971	0	1	0.00	10
11	2.05E+03	20555	20588	254	23	0.00	10	9119	5566	8	1	0.00	10
12	4.10E+03	41045	41084	730	101	0.00	10	10954	6627	173	7	0.00	10
13	8.19E+03	82006	82047	2299	467	0.00	10	11161	6716	179	9	0.00	10
14	1.64E+04	163938	163987	281	2816	0.00	10	13715	8272	0	1	0.00	10
15	3.28E+04	327776	327823	6245	25044	0.03	8	13174	7928	473	36	0.00	10
16	6.55E+04	655465	655519	1362	49223	17.16	7	14870	8893	832	166	0.00	10
20	1.05E+06	-	-	-	-	-	-	19386	11587	14982	4953	0.00	10
30	1.07E+09	-	-	-	-	-	-	47152	27785	60234	44363	0.23	5
40	1.10E+12	-	-	-	-	-	-	65483	38277	33765	86400	0.62	0
50	1.13E+15	-	-	-	-	-	-	113044	66092	25636	86400	0.58	0
60	1.15E+18	-	-	-	-	-	-	131708	76954	10242	86400	0.92	0
70	1.18E+21	-	-	-	-	-	-	195577	113932	8171	86400	1.04	0
80	1.21E+24	-	-	-	-	-	-	230248	134139	4384	86400	1.30	0
90	1.24E+27	-	-	-	-	-	-	287390	167922	1816	86400	1.42	0

Table 3.5: Comparison of the enhanced Model (37)-(43) with the original Model (28)-(35) for instances with OF =1.5.

Instance Info.		*Original Model (28)-(35)*						*Enhanced Model (37)-(43)*					
$\mathcal{I}$	Ω	V	C	N	$Time$	$Gap(\%)$	OSI	V	C	N	$Time$	$Gap(\%)$	OSI
20	1.44E+16	42751	25193	217721	62025	5.45	4	23584	14142	11676	6912	0.00	10
30	1.34E+24	81630	47646	15075	86400	3.70	0	48801	28860	23721	60548	0.26	3
40	1.94E+32	118332	69173	4142	86400	6.16	0	68054	40284	6878	45922	0.09	5
50	1.55E+40	186090	108336	1018	86400	12.19	0	105624	62296	3719	71383	0.78	2
60	1.92E+48	224792	130806	411	86400	16.31	0	133778	78618	1842	86400	0.27	0
70	2.72E+56	328935	191910	0	86400	-	0	193779	114239	1093	86400	1.30	0
80	1.48E+64	377405	219235	2	86400	-	0	216977	127374	739	86401	0.78	0
90	3.05E+72	509603	296910	0	86400	-	0	306616	180278	40	86401	8.00	0
Average		233692	136151	29796	83353	-	-	137152	80761	6214	66296	1.44	-

average of 1 second. We can also observe that the average number of exploited nodes in the branch-and-bound tree for classic Model (20)-(27) is 718.2 versus 72 for enhanced Model (37)-(43). Moreover, for instances with 15 and 16 surgeries, the classic Model (20)-(27) is not able to optimally solve these instances and has an average optimality gap of 8.5%. The reported average computational time for the classic Model (20)-(27) in these instances is 37,133 seconds, while the enhanced Model (37)-(43) solved these instances optimally in on average 101 seconds. It worth mentioning that the average number of variables for classic Model (20)-(27) is 187,870, while the enhanced Model (37)-(43) has on average 11,587 variables. Besides, the enhanced Model (37)-(43) has a significantly lower number of constraints compared to the classic Model (20)-(27).

In Table 3.5, we presents some computational results for the original instances with $OF = 1.5$ to answer the second question about the effect of the proposed enhancements. In these instances, we generated surgical times from lognormal distributions provided in Table 1 in Vinden et al. (2016). In this table, we have compared the original Model (28)-(35) to Model (37)-(43) enhanced by valid inequalities. We observe that the average optimality gap for the enhanced model is 1.44% while we cannot compute this value for the original Model (28)-(35) because of its failure in finding any feasible solution for instances with 70 or more surgeries. However, we can observe that the enhanced Model (37)-(43) significantly outperforms the original Model (28)-(35) in instances with up to 60 surgeries. We also note that the original Model (28)-(35) finds feasible solutions only for 2 instances with 60 surgeries. It is also noteworthy the average optimality gaps of models (28)-(35) to Model (37)-(43) for instances with up to 50 surgeries are 6.88% and 0.28% respectively. Moreover, we observe that the average number of variables and constraints in the enhanced Model (37)-(43) are 70% less than those of the Model (28)-(35). This is because, we have removed a large number of variables and constraints in the enhanced Model (37)-(43). Figure 3.1 shows that, in Model (37)-(43), as the number of surgeries increases, the number of variables and constraints increase polynomially. Also, we can see that the computational time and the number of exploited nodes in the branch-and-bound tree increase exponentially as the number of surgeries increases from 10 to 30. This is reasonable with respect to the NP-hard complexity of the problem. Beyond 30 surgeries, the number of nodes decreases because larger instances are more difficult and the time limit stops exploitation of nodes in the branch-and-bound tree.

In Table 3.6, we assess the performance of our enhanced Model (37)-(43) on instances with different values of OF. In this table, we have three additional columns "LB", "UB", and "VSS". The first two columns represent the values of lower and upper bounds respectively. Column "VSS" provides the values of stochastic solution that we compute by $VSS = 100(UB_{sim} - UB)/UB_{sim}$. In this formula, to compute UB_{sim}, we first solve the expected value problem obtained by replacing the stochastic surgical times by their mean values. Then we fix the obtained surgical allocation from the expected value problem in model (37)-(43) and set UB_{sim} to the obtained optimal objective value of the recent model. We observe that the for different settings of OF, the proposed model is stable and the average optimality gap is less than or equal to 1.50%. It is also noteworthy that

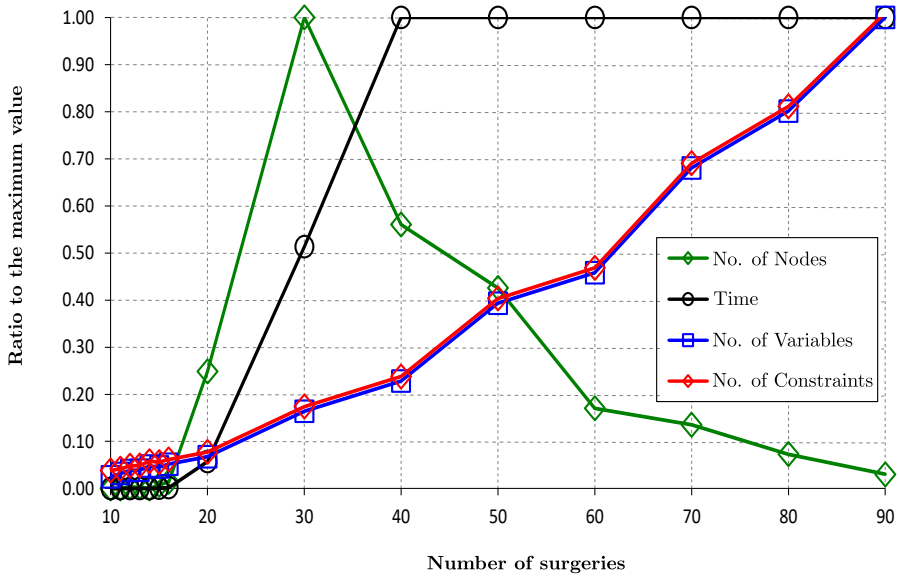

Figure 3.1: Computational time and number of variables, constraint, and nodes versus the number of surgeries.

the average values of VSS varies between 60% to 70%. We expect to see that for larger OF, the average upper bound values are larger. However, the results of Table 3.6 contradicts with this expectation. This is due to the unstable performance of the model on instances with 90 surgeries. Ignoring these instances, we observe that the average values of upper bounds, computational time, and optimality gaps are increasing in OF. In Table 3.6, the largest instances with $OF = 1.5, 1.75, 2,$ and 4 that are optimally solved the enhanced model include 50 surgeries. The number of scenarios in these instances is 1.55E+40.

In Tables 3.7 and 3.8, we evaluate the performance of the enhanced Model (37)-(43) with the chance constraint (44) by setting the chance-constraint factor α to $\{0.01, 0.02, 0.03, 0.04, 0.10, 0.20, 0.30, 1.00\}$. These values show that the maximum probability of using overtime varies from 1% to 100%. In these tables, we presented the computational times and optimality gaps for different values of α. Moreover, for $\alpha \geq 0.02$, we have presented columns "$Imp(\%)$". The values of $Imp(\%)$, show that by increasing the maximum probability of using overtime, how much the objective value improves compared to the case of $\alpha = 0.01$. We have presented the results of Tables 3.7 and 3.8 for instances with up to 60 surgeries, because the proposed model cannot find solutions with reasonable optimality gaps for larger instances. In these tables, we observe that the average optimality gaps are

Table 3.6: Computational results of the enhanced Model (37)-(43) on instances with different values of OF.

Instance Info.			*Enhanced Model (37)-(43)*						
OF	$\|\mathcal{I}\|$	Ω	N	$Time$	LB	UB	$Gap(\%)$	OSI	$VSS(\%)$
1.5	20	1.44E+16	11676	6912	23983	23983	0.00	10	40.08
	30	1.34E+24	23721	60548	32920	33003	0.26	3	50.06
	40	1.94E+32	6878	45922	45712	45750	0.09	5	80.07
	50	1.55E+40	3719	71383	54676	55121	0.78	2	80.01
	60	1.92E+48	1842	86400	65276	65452	0.27	0	60.05
	70	2.72E+56	1093	86400	77819	78814	1.30	0	80.01
	80	1.48E+64	739	86401	86210	86872	0.78	0	80.00
	90	3.05E+72	40	86401	98237	106075	8.00	0	79.59
Average		3.81E+71	6214	66296	60604	61884	1.44	2.86	68.73
1.75	20	1.44E+16	13239	17926	23988	23988	0.00	9	50.14
	30	1.34E+24	24540	60501	32932	33031	0.32	3	70.10
	40	1.94E+32	6757	46168	45713	45758	0.10	5	70.09
	50	1.55E+40	3704	77998	54691	55138	0.78	1	80.05
	60	1.92E+48	1695	86401	65284	65482	0.30	0	50.10
	70	2.72E+56	1185	86401	77852	78879	1.34	0	60.07
	80	1.48E+64	742	86401	86233	86946	0.83	0	60.04
	90	3.05E+72	60	86401	98267	106462	8.34	0	46.55
Average		3.81E+71	6490	68524	60620	61960	1.50	2.57	60.89
2	20	1.44E+16	25900	18234	23991	23992	0.00	9	60.09
	30	1.34E+24	26960	60543	32945	33059	0.36	3	70.07
	40	1.94E+32	7464	48547	45714	45766	0.12	5	90.04
	50	1.55E+40	3444	78448	54706	55153	0.78	1	60.16
	60	1.92E+48	1850	86543	65293	65525	0.35	0	50.10
	70	2.72E+56	1147	86591	77882	78945	1.38	0	39.70
	80	1.48E+64	711	86662	86254	87322	1.25	0	79.60
	90	3.05E+72	96	86693	98295	105318	7.15	0	49.65
Average		3.81E+71	8447	69033	60635	61885	1.42	2.57	62.43
4	20	1.44E+16	23590	20308	24016	24023	0.03	8	50.28
	30	1.34E+24	23070	60554	33061	33284	0.70	3	40.19
	40	1.94E+32	7651	55498	45726	45827	0.23	4	30.48
	50	1.55E+40	3844	78502	54780	55287	0.90	1	50.33
	60	1.92E+48	2313	86400	65364	65817	0.69	0	30.37
	70	2.72E+56	1397	86400	78047	79222	1.51	0	40.34
	80	1.48E+64	584	86401	86402	87677	1.48	0	49.93
	90	3.05E+72	104	86401	98500	103651	5.24	0	16.44
Average		3.81E+71	6214	66296	60604	61884	1.44	2.29	68.73

less than 4.00% and decreasing in α. These tables also show that by increasing the value of α from 0.01 to 1.00 the objective value improves between 2.37% and 8.60%.

3.5.3 Discussion

Considering the limited surgical capacities of hospitals due to the pandemic, an efficient management of operating rooms is vital. Stochastic duration of surgeries is one of the main features of operating room scheduling problems that have been studied in the literature, but there is still a significant research gap in this area. The reason behind the importance of considering the uncertainty

Table 3.7: Computational results of the enhanced Model (37)-(43) with chance constraint (44) for instances with OF=1.5 and $\alpha \in \{0.01, 0.02, 0.03, 0.04\}$.

Instance Info.	$\alpha = 0.01$		$\alpha = 0.02$			$\alpha = 0.03$			$\alpha = 0.04$		
$\|\mathcal{I}\|$	$Time$	$Gap(\%)$	$Time$	$Gap(\%)$	$Imp.(\%)$	$Time$	$Gap(\%)$	$Imp.(\%)$	$Time$	$Gap(\%)$	$Imp.(\%)$
20	5219	0.00	4136	0.00	0.01	5067	0.00	0.01	9188	0.00	0.02
30	32850	2.68	46267	5.40	0.00	45938	5.37	0.00	42265	4.15	1.11
40	49779	2.97	48425	2.97	0.01	47938	2.98	0.01	46130	2.98	0.01
50	79334	5.90	79256	5.24	0.71	75057	4.45	1.48	70220	4.50	1.48
60	86400	7.47	86400	7.47	11.12	86400	6.02	12.47	86400	6.04	12.47
Average	50716	3.80	52897	4.22	2.37	52080	3.76	2.79	50841	3.53	3.02

Table 3.8: Computational results of the enhanced Model (37)-(43) with chance constraint (44) for instances with OF=1.5 and $\alpha \in \{0.10, 0.20, 0.30, 1.00\}$.

Instance Info.	$\alpha = 0.10$			$\alpha = 0.20$			$\alpha = 0.30$			$\alpha = 1.00$		
$\|\mathcal{I}\|$	$Time$	$Gap(\%)$	$Imp.(\%)$	$Time$	$Gap(\%)$	$Imp.(\%)$	$Time$	$Gap(\%)$	$Imp.(\%)$	$Time$	$Gap(\%)$	$Imp.(\%)$
20	3165	0.00	0.02	12510	0.00	0.02	12902	0.00	0.02	6912	0.00	1.61
30	47004	4.21	2.34	57416	2.85	4.75	57865	2.85	4.75	60548	0.26	8.14
40	47087	2.98	0.01	47963	0.09	2.65	47912	0.09	2.65	45922	0.09	2.65
50	71697	3.78	2.25	71013	2.27	3.64	70707	2.27	3.64	71383	0.77	5.11
60	86400	6.65	22.50	86401	5.34	23.77	86401	5.34	23.77	86400	0.27	25.51
Average	51071	3.52	5.42	55061	2.11	6.97	55157	2.11	6.97	54233	0.28	8.60

of surgical durations is that many surgeries are canceled at the end of the day due to the unexpected longer durations of previous surgeries. Our proposed model takes into account the stochastic durations of surgeries and avoid such cancellations by including the expected overtimes of operating rooms.

Our results showed that we can find the near-optimal solutions of large instances with 80 surgeries and 1.48E+64 stochastic scenarios in a reasonable time using a compact integer programming model. The obtained results demonstrated that the proposed model is significantly more efficient than classic scenario-based stochastic programming models. Another advantage of the proposed model in comparison to other models in the literature is that it can be easily implemented in commercial optimization software, without any need for the use of other complicated approaches such as SAA. Therefore, the proposed model has the potential to be used by practitioners for weekly operating room planning problems.

3.6 Conclusion

In this research, we proposed a new state-variable model for a stochastic weekly operating room planning problem with an exponential number of scenarios. The objective function of the mathematical models aimed to minimize the sum of the fixed opening cost of operating rooms and the expected overtime costs that are computed in the second stage. We first proved that our model is valid and showed that the proposed model has several advantages over classical models. The main advantage of the offered model is that it has a pseudo-polynomial number of variables and constraints that are significantly fewer than the number of variables and constraints in an equivalent scenario-based stochastic programming model. After presenting the original state-variable model, we enhanced the model to involve less variables and constraints. We also strengthened the model by developing several valid inequalities, including worst-case scenario and symmetry-breaking cuts.

We evaluated the performance of the proposed models against the classic scenario-based stochastic programming model for the problem in several small, medium, and large size instances. Our results revealed that the proposed original state-variable model provides promising solutions compared to the classic model. The original state-variable model was able to solve instances with up to 20 patients optimally while the classical model could not solve the instances with more than 14 patients optimally. Besides, we showed that the original state-variable model was able to solve instances with up to 90 patients with an average optimality gap of 1.42%. In addition, we demonstrated that our model has significantly fewer variables and constraints. Moreover, we compared the performance of the enhanced state-variable model to the original state variable model and showed that the worst-case scenario and symmetry-breaking cuts significantly improved the model efficiency. The results of the computational experiments disclosed that the enhanced model has even remarkably fewer variables and constraints enabling us to solve large size instances with 50 surgeries and 1.55E+40 scenarios optimally. Furthermore, the results revealed that the enhanced model obtains feasible solutions with an average optimality gap of 0.78% for instances with 80 surgeries and 1.48E+64 scenarios.

For future work, it would be interesting to extend the proposed state-variable model for operating room scheduling problems with the allocation of surgical specialty. Another research direction

is to consider the scheduling of emergency cases together with the elective surgeries. Moreover, extension of the state-variable models for other scheduling applications is another future research avenue.

Chapter 4

Designing a hybrid reinforcement learning based algorithm with application in prediction of the COVID-19 pandemic in Quebec[1]

Abstract

World Health Organization (WHO) stated COVID-19 as a pandemic in March 2020. Since then, 26,795,847 cases have been reported worldwide, and 878,963 lost their lives due to the illness by September 3, 2020. Prediction of the COVID-19 pandemic will enable policymakers to optimize the use of healthcare system capacity and resource allocation to minimize the fatality rate. In this research, we design a novel hybrid reinforcement learning-based algorithm capable of solving complex optimization problems. We apply our algorithm to several well-known benchmarks and show that the proposed methodology provides quality solutions for most complex benchmarks. Besides, we show the dominance of the offered method over state-of-the-art methods through several measures. Moreover, to demonstrate the suggested method's efficiency in optimizing real-world

[1]This paper is published in *Annals of Operations Research* in 2021 and has been cited 70 times as of March 2024.

problems, we implement our approach to the most recent data from Quebec, Canada, to predict the COVID-19 outbreak. Our algorithm, combined with the most recent mathematical model for COVID-19 pandemic prediction, accurately reflected the future trend of the pandemic with a mean square error of 6.29E-06. Furthermore, we generate several scenarios for deepening our insight into pandemic growth. We determine essential factors and deliver various managerial insights to help policymakers making decisions regarding future social measures.

4.1 Introduction

Researchers use optimization in nearly every study area. Optimization remains a fundamental challenge in science and engineering, primarily because of the difficulty of real-world problems and the limitations of traditional methods. Randomization Search Algorithms (RSAs) are among the most flexible and most efficient methodologies to solve complicated problems. These algorithms are mostly polynomial-time algorithms and have significantly lower computational complexity. As one of the most commonly used RSAs, metaheuristics are algorithms that are inspired by natural phenomena to perform optimization. Metaheuristics perform very well in exploring the feasible region and evade local optimum using effective movement processes.

Healthcare science is one of the top research areas in which metaheuristics have been widely applied to. Using these algorithms, scientists can optimize healthcare systems significantly in terms of several objectives, including minimizing cost, waiting time, service time, delivery time, and maximizing reliability or customer satisfaction. In December 2019, a novel strain of coronavirus called SARS-Cov-2 discovered in China. The virus causes COVID-19, a severe respiratory disease. Regardless of primary measures applied by the government of China, the disease spread quickly to many countries leading to 26,795,847 infected cases and 878,963 deaths. Currently, there are no effective medications and vaccines for the disease. However, the effectiveness of some treatment options is under study via clinical trials (Government of Canada, 2023). Although most people with mild COVID-19 symptoms recover independently, some other people with severe and critical symptoms need hospitalization in wards and Intensive Care Units (Public Health Agency of Canada, 2020). However, due to the limited capacity of the healthcare system, it is impossible to admit all

the patients in hospitals.

Efficient Modeling and prediction of the COVID-19 pandemic will meaningfully aid the policymakers and healthcare experts in making decisions to stop the spread of the disease. Besides, by forecasting the upcoming trend of the epidemic, we can also optimally allocate resources to hospitals that will avoid equipment shortages and save patients' lives. Prediction of the COVID-19's trend is challenging because of its uncertain nature and complication. Recently, scientists have provided a novel model to simulate the COVID-19 pandemic called SIDARTHE and was initially offered in research by Giordano et al. (2020) published in *Nature Medicine*. The researchers highlighted the efficiency of the proposed formulation in modeling the pandemic growth. Nevertheless, they emphasized that solving the presented set of differential equations is difficult because of the exceptional characteristics of the model.

In the current research, we offer a novel search methodology that will solve many complex optimization problems very efficiently in a short time. Our algorithm simultaneously benefits from the advantages of Machine Learning (ML) and Evolutionary Computation (EC). In our research, we propose a hybrid algorithm that involves a Reinforcement Learning (RL) technique as the main engine and several ECs as updating operators. The learning process achieved by the RL method accelerates the algorithm and enables us to resolve complicated large-scale problems. The procedure utilizes several operators to enhance the exploration and exploitation capabilities of the algorithm. These features help the proposed process to sidestep local optimum while exploiting the solution space intelligently. We implement the algorithm on several well-known recently developed benchmarks to show its efficiency in solving such complex problems. Moreover, we highlighted significant differences in the performance of the method comparing to other methodologies using robust statistical tests.

Furthermore, we use the proposed algorithm to model and predict the COVID-19 pandemic in Quebec, Canada, that has the most cases of COVID-19 in the country. Our results accurately predict the peak in the number of infected cases of COVID-19 in the province. The outcomes also determine the peak of the number of cases that develop life-threatening symptoms that will require hospitalization. We also perform complex sensitivity analyses to portray future scenarios that enable us to provide detailed information to policymakers and healthcare professionals. In our

study, we also measure the effectiveness of implementing measures such as social distancing and partial lockdown on pandemic growth.

The rest of the current study is prepared as follows: In Section 4.2, we deliver a comprehensive review of existing research on the topic. In Section 4.3, we offer a novel algorithm to resolve many complex problems using RL and EC. In Section 4.4, we apply our algorithm to several well-known benchmark functions. We assess the performance of the suggested approach and compare its efficiency to state-of-the-art methods. In Section 4.5, we implement our method to model and predict the COVID-19 pandemic in Quebec, Canada. In Section 4.6, we perform sensitivity analyses and provide valuable managerial insights to fight the COVID-19 pandemic. In Section 4.7, we conclude the paper.

4.2 Survey on Research Conducted

The core idea behind most EC algorithms is to follow a swarm intelligence that is inspired by animal behavior and natural phenomenon. Mirjalili and Lewis (2016) categorized metaheuristics into four main groups: Evolutionary Algorithms (EAs), Physics-based Algorithms (PAs), Swarm Algorithms (SAs), and Machine Learning-based Algorithms (MLAs). EAs imitate the evolution procedure in nature to solve complex problems. PAs utilize laws of physics that enable this family of algorithms to handle complicated problems. On the other hand, SAs simulate the swarm behavior of many individuals in a group. Also, MLAs use artificial intelligence and machine learning to enhance the performance of the previous families of algorithms in terms of exploration and exploitation. Table 4.1 provides some of the most advanced metaheuristics developed in recent years.

Table 4.1: Classification of the metaheuristics.

EAs	**PAs**	**SAs**	**Other algorithms**	**MLAs**
Genetic Algorithms (GA) (Holland, 1992)	Small-World Optimization Algorithm (SWOA) (Du et al., 2006)	Particle Swarm Optimization (PSO) (Eberhart and Kennedy, 1995)	Stochastic Fractal Search (SFS) (Salimi, 2015)	Hybrid Q-Learning based Algorithm (This research)
Genetic Programming (GP) (Koza, 1992)	Curved Space Optimization (CSO) (Moghaddam et al., 2012)	Grasshopper Optimization Algorithm (Saremi et al., 2017)	Sine Cosine Algorithm (SCA) (Mirjalili, 2016b)	
Degree-Descending Search Strategy (DDS) (Cui et al., 2018)	Charged System Search (CSS) (Kaveh and Talatahari, 2010)	Ant Lion Optimization Algorithm (ALO) (Mirjalili, 2015)	Water Cycle Algorithm (WCA) (Eskandar et al., 2012)	
Biogeography Based Optimizer (BBO) (Simon, 2008)	Multi-Verse Optimization (MVO) Algorithm (Mirjalili et al., 2016)	Crow Search Algorithm (CSA) (Askarzadeh, 2016)	Virus colony search (Li et al., 2016)	
Differential Evolution (DE) (Price, 2013)	Black Hole Mechanics Optimization (BHMO) (Kaveh et al., 2020)	Salp Swarm Algorithm (SSA) (Mirjalili et al., 2017)	Gradient-Based Optimizer (GBO) (Ahmadianfar et al., 2020)	
Estimation of distribution algorithm (EDA) (Wang et al., 2013)	Galaxy-based Search Algorithm (GBSA) (Shah-Hosseini, 2011)	Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014)	Lightning Search Algorithm (LSA) (Shareef et al., 2015)	
Evolution Strategy (ES) (Rechenberg, 1978)	Simulated Annealing (SA) (Kirkpatrick et al., 1983);	Dragonfly Algorithm (DA) (Mirjalili, 2016a)	Coronavirus Optimization Algorithm (COA) (Martínez-Álvarez et al., 2020)	
Evolutionary Programming (EP) (Fogel, 1998)	Gravitational Search Algorithm (GSA) (Rashedi et al., 2009)	Cuckoo Search (CS) (Yang and Deb, 2009)	Sine–Cosine Crow Search Algorithm (SCCSA) (Khalilpourazari and Pasandideh, 2020)	
Central Force Optimization (CFO) (Formato, 2007)		Whale Optimization Algorithm (WOA) (Mirjalili and Lewis, 2016)	Water Cycle Moth Flame Optimization (WCMFO) (Khalilpourazari and Khalilpourazary, 2019)	

Many researchers used these algorithms to solve complex optimization problems in different fields (Hoursan et al., 2020; Sangaiah et al., 2020). Defined by the No Free Lunch (NFL) Theorem, we can logically prove that no single method performs optimally in resolving all problems (Adam et al., 2019; Wolpert and Macready, 1997). Any metaheuristic algorithm may perform well in some benchmarks but weak in others. This theorem makes this field of study highly interesting for researchers searching for an algorithm that performs promising in many benchmarks. In our algorithm, we consider several ECs and operators and let an RL method decide which algorithm to use to relocate each element. Besides, learning during the optimization process will significantly

reduce the computational burden and improve the quality of the results. The learning process accelerates the algorithm due to the fact that using the learning process over iterations, the algorithm adapts its operators to perform the best for each problem.

Moreover, we consider a proper framework to maintain a decent equilibrium between exploration and exploitation of the feasible region over generations of the algorithm to avoid local optima. We show the efficiency of our algorithm on the most complex benchmarks in the literature. Besides, we apply the suggested algorithm to predict the COVID-19 pandemic in Quebec, Canada. Our outcomes express that the designed algorithm robustly predicts the future trends of the pandemic.

4.3 Algorithm Development

Metaheuristics work based on randomization. By randomization, we mean that these algorithms use random stepsizes while updating each particle's position in the solution space. Metaheuristics use unique operators and strategies to update the position of each particle (solution). The efficiency of these operators and algorithms significantly depends on the solution space of the problem. For instance, some algorithms follow a direct updating procedure, such as Water Cycle Algorithm (WCA), while some other encircle the best solution to update the position of a given particle, such as Grey Wolf Optimizer (GWO). Each of these operators and moving strategies has unique advantages that enable the algorithms to perform well in optimizing specific problems. Therefore, developing new algorithms that could efficiently solve a higher number of optimization problems is essential.

In this study, we use several operators (moving strategies) from various algorithms to update the particles' position in the solution space. For updating each particle, we have to choose an operator from the given set of operators. Determining the best strategy and the most efficient operator for any given optimization problem is computationally challenging. Therefore, we use a reinforcement learning method that learns and optimizes the choice of operators during the optimization process to achieve optimal performance. In the following, we first describe all the features of the offered algorithm, and then we define the algorithm in a unique structure.

4.3.1 Q-learning

Reinforcement Learning (RL) that approximates dynamic programming, and neuro-dynamic programming, is a type of machine learning which determines the best actions in a specific environment to maximize a reward (Bertsekas, 2019). One of the main features of reinforcement learning is that the agent receives a reward or punishment after executing an action. The RL continues to interact with the environment to achieve the optimal policy via trial and error.

The Q-Learning is one of the most efficient Reinforcement Learning algorithms that determine an optimal policy by evaluating taken actions using the environment. Q-learning is a way to optimize solutions in a Markov Decision Process (MDP) problem (Akhtar, 2017). The Q-learning algorithm aims to maximize the anticipated reward by determining the optimal state-action pairs. The algorithm uses a $Q\left(s,a\right)$ table where s_t is the state and a_t is the action at time step *t,* and Q is the cumulative reward matrix. The algorithm updates the components of the Q-table $Q\left(s,a\right)$ iteratively using Eq. (52).

$$Q_{(t+1)}(s_t,a_t) = Q_t(s_t,a_t) + \epsilon_t(r_t + \gamma \max(Q_t(s_{t+1},a_{t+1})) - Q_t(s_t,a_t)), \tag{52}$$

In Eq. (52), ϵ_t denotes the learning rate parameter and r_t is the obtained reward/punishment from the current action. Besides, the expression γ is the scaling factor. One of the main challenges in designing an efficient Q-learning algorithm is in determining the importance of the information gained throughout interactions with the environment. For instance, assigning a value close to 1 to the learning rate parameter means that we consider higher importance for the recent information gained. To optimize the value of the learning rate parameter throughout iterations, we use an adaptive methodology that uses Eq. (53) to intelligently tune the learning rate parameter to explore and exploit the search region (Zamli et al., 2018).

$$\epsilon_t = 1 - 0.9\frac{t}{MT}, \tag{53}$$

In Eq. (53), t is the iteration index, and MT is the maximum generation. In this research, we consider a reward value of 1 if the current action improves the solution quality of the particle;

otherwise, we consider a punishment value of -1. Based on the given illustrations, we present the Pseudo-code of the Q-learning in Algorithm 1.

Algorithm 1 Pseudo Code for the Q-Learning Algorithm

```
Input states, actions, gamma, and initial Q(s, a) table;
while stopping criteria not met do
    Select the best action from the Q-table;
    Execute the action and observe the reward and new state;
    Update the current state, s(t) = s(t+1);
end while
return The Q-table; =0
```

In this research, we consider several efficient operators from different algorithms and let the Q-learning algorithm determine the best action throughout the optimization process to modify the location of each particle in the feasible space. In the following subsections, we present the operators in detail.

4.3.2 Grey Wolf Optimizer

Similar to other swarm intelligence-based systems, GWO initially generates a set of primary solutions. Then, it sorts the solutions regarding their fitness and considers the three best solutions as the dominant wolves (Alpha, Beta, and Delta). The following equations imitate the encircling behavior of the grey wolves around prey:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|, \tag{54}$$

$$\bar{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}, \tag{55}$$

In Eqs. (54)-(55), t represents the iteration index and $\vec{A}$ and $\vec{C}$ characterize location vectors of target and other grey wolves. $\vec{X}_p(t)$ and $\vec{X}(t)$ are the position of the prey and grey wolf, respectively. These coefficients are calculated as follows:

$$\vec{A} = 2a_1 \cdot \vec{r}_1 - a_1, \tag{56}$$

$$\vec{C} = 2 \cdot \vec{r}_2, \tag{57}$$

In Eqs. (56)-(57), a_1 decreases over iterations from 2 to 0 and $\vec{r}_1$ and $\vec{r}_2$ are random vectors. After encircling the prey, the wolves start the hunting process. To mathematically express the movements of grey wolves in the hunting process, we consider that the Alpha, Beta, and Delta have superior knowledge of the probable position of the target (possible optimal solution of the problem). In this framework, the following formulas are recommended to mimic the hunting process:

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \qquad \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \qquad \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|, \tag{58}$$

$$\bar{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \qquad \bar{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \qquad \bar{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta, \tag{59}$$

$$\bar{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \tag{60}$$

Figure 4.1 depicts a graphical interpretation of the hunting action in 2D space.

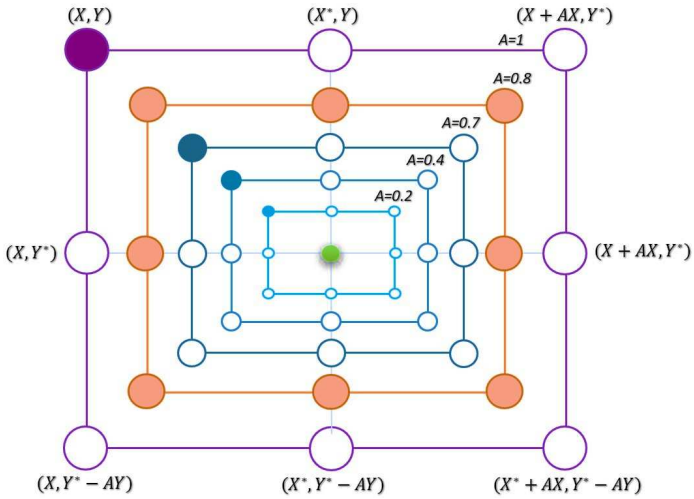

Figure 4.1: Hunting behavior in GWO.

4.3.3 Sine-Cosine Algorithm

SCA is a newly developed search procedure that mimics the sine and cosine like movements in the feasible space to modify the elements using eq. (10):

$$X = X_{\text{best}} + R_1 \cdot \sin(R_2) \cdot \cos(R_3) \cdot (R_4) \tag{61}$$

In Eq. (61), X is the present location, X_{best} is the location of the best particle and R_1, R_2, R_3, R_4

are random numbers in $(0, 1]$. Throughout iterations, R_2 displays the movement path, R_3 controls the moving distance, R_1 guarantees a suitable equilibrium among underline or deemphasize the desalination, and R_4 selects a sine or cosine measure for updating procedure. Figure 4.2 shows the movement behavior of the sine and cosine actions.

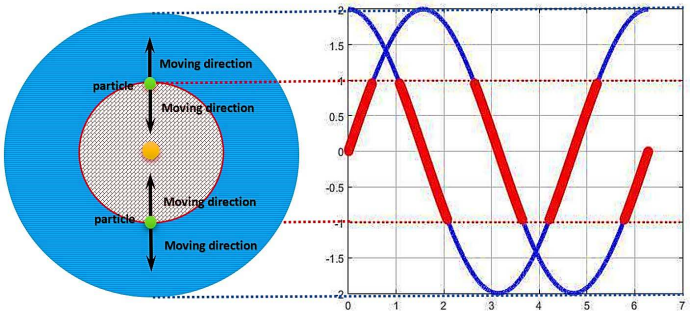

Figure 4.2: Updating procedure in SCA.

SCA uses sine and cosine movements intelligently to evade local optima. In addition to adjusting the particles' movements during the solution process, SCA reduces the value of R_1 parameter using the below formula to sustain a proper equilibrium between exploration and exploitation as follows:

$$R_1 = a_2 - \frac{t \cdot a_2}{T} \tag{62}$$

In Eq. (62), t displays present repetition, a_2 is a constant, and T is the maximum generation.

4.3.4 Moth-Flame Optimization

Like most of the optimization paradigms, MFO initiates the optimization by creating random solutions. Then, it mimics the spiral flying actions of the moths around light sources using a logarithmic spiral function as follows:

$$IP_i^{(X+1)} = |F_i - IP_i^X| \cdot e^{b \cdot t} \cdot \cos(2\pi t) + F_i, \tag{63}$$

In Eq. (63), t is a constant in $[-1, 1]$, and b is a constant for determining the form of the logarithmic spiral. F_i is the flame (the best solution), IP_i^X is the moth, and $|F_i - IP_i^X|$ calculates the distance between the moth and flame. Figure 4.3 shows the spiral movement around the flame.

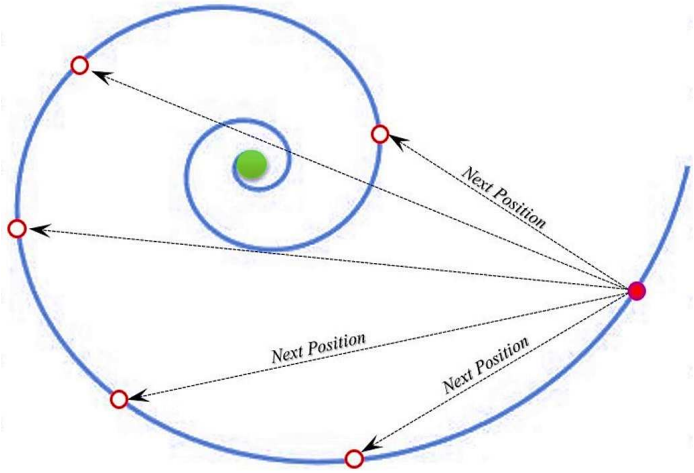

Figure 4.3: The spiral fly path of the moths around the flame.

In order to maintain a suitable balance among its exploration and exploitation, the MFO algorithm reduces the search radius using the following equations:

$$a_3 = -1 + t(\frac{-1}{MT}) \tag{64}$$

$$tt = (a_3 - 1) \times \text{rand} + 1 \tag{65}$$

In Eqs. (64)-(65), t displays present repetition, and MT is the maximum generation.

4.3.5 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is one of the most efficient procedures for optimization, and it performs promisingly in solving many complex problems. PSO is a population-based algorithm that uses the following equation to update the location of a given particle in the solution space:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{66}$$

In Eq. (66), $x_i(t)$ presents the current location of the particle and $v_i(t+1)$ determines the velocity of the particle. The v_i vector is the main component of the updating operator that is calculated as follows:

$$v_i(t+1) = \omega v_i(t) + C_1 r_1(p_i(t) - x_i(t)) + C_2 r_2(G(t) - x_i(t)) \tag{67}$$

In Eq. (67), r_1, r_2 are random numbers in $(0, 1]$, C_1, C_2 are coefficients, $p_i(t)$ is the best solution found by the particle so far, and $G(t)$ is the best solution attained so far.

4.3.6 Water Cycle Algorithm

The Water Cycle Algorithm (WCA) is one of the best algorithms for solving complex problems. WCA is a population-based nature-inspired metaheuristic that mimics the flow of streams to rivers and sea to perform optimization as presented in Eq. (68).

$$x_{\text{current}}^{(i+1)} = x_{\text{current}}^{i} + C(x_{\text{best\_sol}}^{i} - x_{\text{current}}^{i}) \tag{68}$$

In Eq. (68), C is a random value. We use the updating operator in the WCA as one of the means to update the location of a given particle in the feasible space. Figure 4.4 represents the updating procedure in WCA.

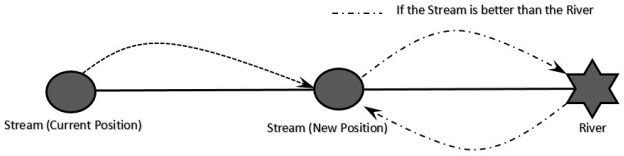

Figure 4.4: Updating procedure in WCA.

4.3.7 Gaussian Walks and Lévy Flight

In this subsection, we use the leading operators of the Stochastic Fractal Search (SFS) offered by (Salimi, 2015). SFS utilizes an important scientific property called "fractal". Fractals are complicated geometric shapes that generally have a "fractional dimension," resulting in self-similarity. SFS follows Diffusion Limited Aggregation (DLA), which is an efficient technique to create fractals. To simulate the DLA, we use the Gaussian walk and Lévy flight. Figure 4.5 presents a fractal shape produced through the DLA scheme.

We use the following equation to simulate the diffusion process in the DLA method.

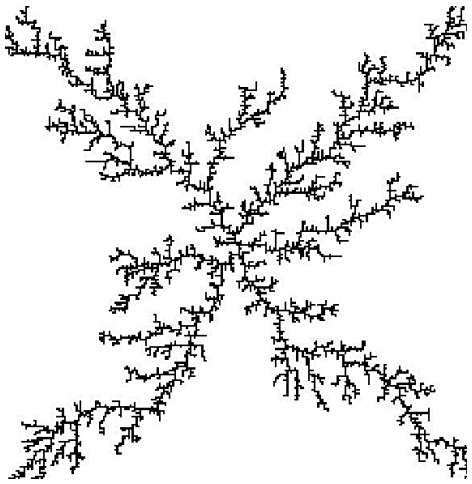

Figure 4.5: A fractal produced through the DLA method.

$$x_i^q = x_i + \beta \times \text{Gaussian}(|BP|, \sigma) - (\varepsilon \times BP - \varepsilon' \times x_i) \tag{69}$$

In Eq. (69), q describes the number of new solutions created through the diffusion of each particle, σ is the standard deviation of the Gaussian walk, and BP is the best solution. Also, x_i^q denotes new particles produced via the diffusion process and x_i is the i-th solution. Besides, ε' and ε are randomly generated numbers in $(0, 1]$. The element σ is defined using the below formula:

$$\sigma = \left| \frac{\log(g)}{g} \times (P_i - BP) \right| \tag{70}$$

where $\frac{\log(g)}{g}$ decreases the length of Gaussian jumps over iterations. Element g is the iteration number and P_i is the current position of the particle.

In order to enhance exploration and randomness in the population, we also apply a Lévy flight-based updating procedure to the particle under consideration as follows:

$$X_{\text{new}}^i = X_c^i + X_c^i \otimes \text{Levy}(D) \qquad \text{for } i = 1, \ldots, m \tag{71}$$

The expression X_{new}^i is the new location and X_c^i is the current location of the particle, respectively. We calculate the Lévy flight using Eq. (72).

$$\text{Levy}(x) = \frac{0.01 \times \sigma \times r_1}{|r_2|^{1/\beta}} \tag{72}$$

where r_1 and r_2 are random numbers. In Eq. (72), σ is obtained as follows:

$$\sigma = \left(\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma((1+\beta)/2)\beta 2^{((\beta-1)/2)}} \right)^{1/\beta} \tag{73}$$

4.3.8 The Developed Hybrid Q-Learning Based Algorithm

In this section, we propose a novel Hybrid Q-Learning based Algorithm (HQLA) to solve complicated optimization problems. The idea behind this algorithm is to design a solution methodology that is capable of solving complicated problems by adapting its operators to any solution space. We designed an optimization procedure that can benefit from the advantages of several algorithms. The process can use any movement strategy based on each updating operator. However, a reinforcement learning based algorithm (Q-learning) selects the best action in each iteration for each particle. When optimization begins, the algorithm performs several random actions to evaluate the efficiency of each type of operator. As the iterations continue, Q-learning learns how to employ different actions to achieve the best possible solution. For each action, we consider a reward equal to 1 if the current operators improve the solution quality; Otherwise, the algorithm assigns a punishment value of -1 to the action. The Pseudo-code of the algorithm is available in Algorithm 2.

Algorithm 2 Pseudo Code for the HQLA

```
Input parameters of the algorithm;
Create a set of randomly generated solutions;
while stopping criteria not met do
    Check for infeasibility of the particles;
    Bring infeasible particles to the feasible solution space;
    Sort the solutions based on their fitness value;
    for each particle do
        if it is the first iteration then
            Select a random action;
        else
            Select the action using the Q-table;
        end if
        if the action is GWO then
            Use GWO operators to update the position of the particle;
        else if the action is SFS then
            Use SFS operators to update the position of the particle;
        else if the action is WCA then
            Use WCA operators to update the position of the particle;
        else if the action is PSO then
            Use PSO operators to update the position of the particle;
        else if the action is MFO then
            Use MFO operators to update the position of the particle;
        else if the action is SCA then
            Use SCA operators to update the position of the particle;
        end if
        Check for infeasibility of the particle;
        Bring infeasible particle to the feasible solution space;
        Calculate the objective function value of the particle;
        Determine the reward/punishment value;
        Update the current state, s(t) = s(t + 1);
    end for
end while
Return the best solution; =0
```

4.4 Results and discussions

Metaheuristics are approximation algorithms that are based on randomized movements. Therefore, their performance may differ from one problem to another. Thus, to show the efficiency of a metaheuristic algorithm, we should apply them to many benchmark functions. For this purpose, we use 29 benchmarks, including Unimodal, multimodal, fixed dimensional multimodal, and hybrid composite functions that are among the most complex benchmarks in the literature. For more details regarding these benchmarks, see Appendix D.

We compare our algorithm to state-of-the-art methods, including Crow Search Algorithm (CSA),

Artificial Bee Colony (ABC), Cuckoo Search (CS), Genetic Algorithm (GA), Moth-Flame optimization (MFO), Gravitational Search Algorithm (GSA), and Dragonfly Algorithm (DA). In order to solve the benchmark functions, we considered 15000 Number of Function Evaluations (NFEs) to perform a reasonable assessment. Besides, to draw a reliable conclusion, we apply each algorithm on each benchmark 30 times, and report mean, standard deviation, worst and best results. Table 4.2 provides more details about the values of the main parameters of the algorithms.

Table 4.2: The values of the parameters of the algorithms.

Algorithm	Parameter	Value	Algorithm	Parameter	Value
HQLA	Number of initial solutions	30	GA	Cross over probability	0.9
	ω	Decreases linearly from 0.9 to 0.4		Mutation probability	0.005
	C_2	2		Number of initial solutions	30
	C_1	2	GSA	Number of initial solutions	30
	a_1	Decreases linearly from 2 to 0		G_0	1
	a_2	Decreases linearly from 2 to 0		α	20
	a_3	Decreases linearly from -1 to -2	MFO	a_3	Decreases linearly from -1 to -2
CSA	Number of initial solutions	30		Number of initial solutions	30
DA	Number of initial solutions	30	CS	Discovery rate of alien solutions	0.25
ABC	Number of initial solutions	30			

In the first step, we assess the performance of the HQLA on unimodal benchmarks (F1-F7). Figure 4.6 depicts a 2D representation of these benchmark functions.

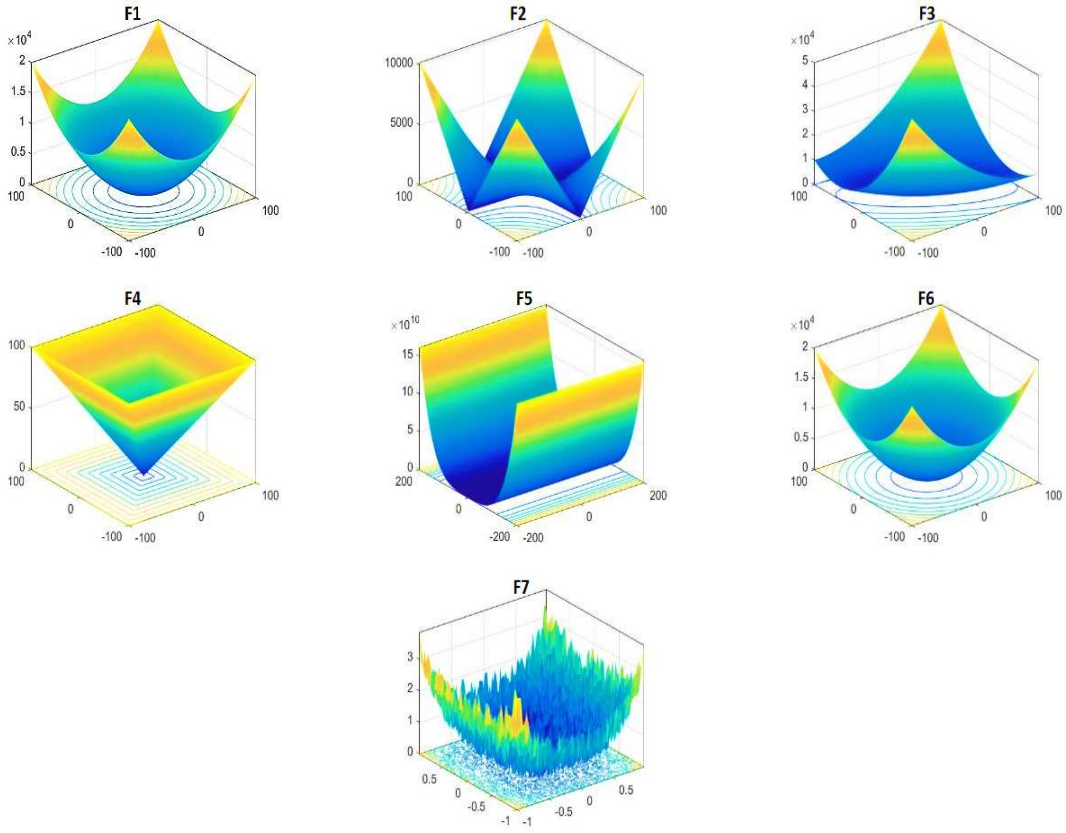

Figure 4.6: 2D representation of F1-F7.

Unimodal benchmarks do not have several local optima and are considered to assess the exploitation ability of metaheuristic algorithms. This set of test suites are difficult to solve since the algorithms should first locate the global optima approximately and then perform exploitation to provide the best approximation of the location of the global optima. Table 4.3 provides detailed information on the performance of the algorithms in unimodal benchmarks. Based on the results of Table 4.3, we observe that the HQLA performs the best in most of the benchmarks. In F1-F5 and F7, the HQLA outperforms all other algorithms by obtaining the best possible solution for all benchmarks. In F6, HQLA ranks third in providing the best solution for this benchmark. Besides, HQLA provides the lowest standard deviation that shows very low variability in the performance of this algorithm. Moreover, considering the boxplot of the results of Table 4.3 present in Figures 4.7, 4.8, and 4.9, it becomes apparent that HQLA has the lowest and narrowest boxplot among the algorithms.

Table 4.3: Computational outcomes of the algorithms in solving unimodal benchmark functions

		HQLA	CSA	ABC	DA	CS	MFO	GSA	GA
F1	Average	**2.14E-66**	2.25E-05	6.10E-02	6.02E+00	3.68E-05	6.34E-14	2.53E-18	1.21E-02
	Std Dev	**6.92E-66**	2.78E-05	0.056501	16.50088	1.82E-05	1.1E-13	9.7E-19	0.009841
	Worst	**3.20E-65**	1.23E-04	2.37E-01	8.36E+01	9.72E-05	5.61E-13	5.91E-18	3.37E-02
	Best	**8.88E-105**	1.98E-06	5.96E-03	0.00E+00	1.63E-05	6.96E-16	1.35E-18	2.61E-04
F2	Average	**6.95E-37**	3.81E-03	7.54E-02	1.69E+00	1.16E-02	6.67E-01	4.93E-09	1.58E-02
	Std Dev	**2.5E-36**	0.002926	0.035631	2.274098	0.002961	2.494438	1.12E-09	0.009469
	Worst	**1.31E-35**	1.37E-02	1.69E-01	1.09E+01	1.66E-02	1.00E+01	7.85E-09	3.81E-02
	Best	**4.31E-57**	6.10E-04	2.15E-02	1.33E-01	6.33E-03	2.78E-10	3.21E-09	1.99E-03
F3	Average	**5.39E-37**	1.68E-02	1.62E+03	1.96E+02	7.00E-02	1.67E+02	3.31E+00	6.13E+01
	Std Dev	**2.24E-36**	0.025618	396.8719	580.7061	0.025593	897.525	3.383667	27.55541
	Worst	**1.25E-35**	1.34E-01	2.51E+03	3.20E+03	1.42E-01	5.00E+03	1.31E+01	1.32E+02
	Best	**1.61E-44**	7.80E-04	7.52E+02	4.76E-02	3.33E-02	2.05E-05	6.14E-02	1.72E+01
F4	Average	**2.64E-25**	1.08E-02	1.76E+01	1.30E+00	2.59E-01	3.76E-02	1.23E-09	6.21E-01
	Std Dev	**6.16E-25**	0.008445	4.709691	1.290182	0.068563	0.137389	2.11E-10	0.162887
	Worst	**2.98E-24**	3.46E-02	2.44E+01	5.21E+00	4.23E-01	7.47E-01	1.76E-09	9.66E-01
	Best	**1.76E-28**	1.98E-03	8.29E+00	0.00E+00	1.19E-01	1.45E-04	8.47E-10	2.86E-01
F5	Average	**5.34E+00**	6.45E+00	8.06E+01	3.66E+03	5.92E+00	3.18E+03	6.91E+00	3.65E+01
	Std Dev	**0.227996**	2.09398	33.02036	16096.55	2.065037	16131.83	0.19993	38.87984
	Worst	**5.93E+00**	9.38E+00	1.58E+02	9.01E+04	1.00E+01	9.00E+04	7.55E+00	1.33E+02
	Best	**4.88E+00**	3.55E-01	1.74E+01	6.94E+00	3.03E+00	1.16E+00	6.46E+00	2.68E+00
F6	Average	5.36E-06	1.56E-05	5.46E-02	5.47E+00	3.06E-05	7.57E-14	**2.60E-18**	1.40E-02
	Std Dev	1.66E-06	1.16E-05	0.043146	25.99325	1.65E-05	1.51E-13	**8.35E-19**	0.009801
	Worst	1.01E-05	4.10E-05	1.41E-01	1.45E+02	6.71E-05	7.85E-13	**5.09E-18**	3.67E-02
	Best	2.42E-06	1.09E-06	7.47E-03	2.22E-06	7.31E-06	1.91E-16	**1.33E-18**	6.39E-04
F7	Average	**2.87E-04**	2.72E-03	9.01E-02	1.74E-02	1.06E-02	5.82E-03	5.63E-03	2.88E-03
	Std Dev	**0.000182**	0.001551	0.036762	0.013218	0.00359	0.003094	0.002614	0.001706
	Worst	**7.50E-04**	6.82E-03	1.61E-01	5.44E-02	1.72E-02	1.60E-02	1.18E-02	7.37E-03
	Best	**2.15E-05**	8.77E-04	3.30E-02	2.43E-03	3.95E-03	2.02E-03	8.37E-04	6.79E-04

Furthermore, Figures 4.10, 4.11, and 4.12, that present the convergence plots of the methods, disclose that the HQLA can maintain a perfect balance among exploration and exploitation of the

solution space. In Figure 4.10, 4.11, and 4.12, we observe that the HQLA can continually improve the best solution attained in each iteration by choosing the best operator to explore the solution space. The learning process in HQLA enables the algorithm to determine the best operator to change the location of the particles in the solution region by evaluating the efficiency of each updating mechanism. Based on these observations, we conclude that HQLA is a reliable technique for this family of benchmark functions.

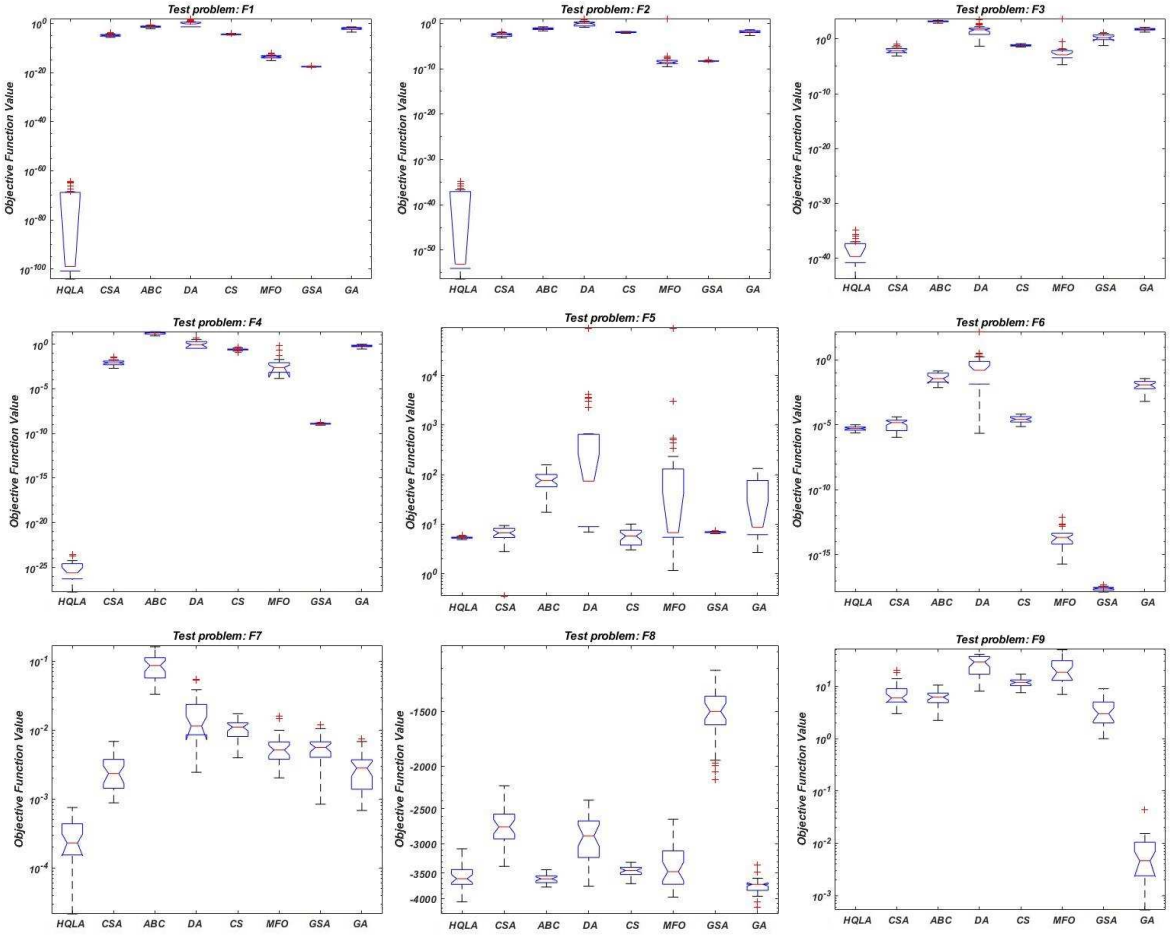

Figure 4.7: Boxplot of the results in F1-F9 benchmarks.

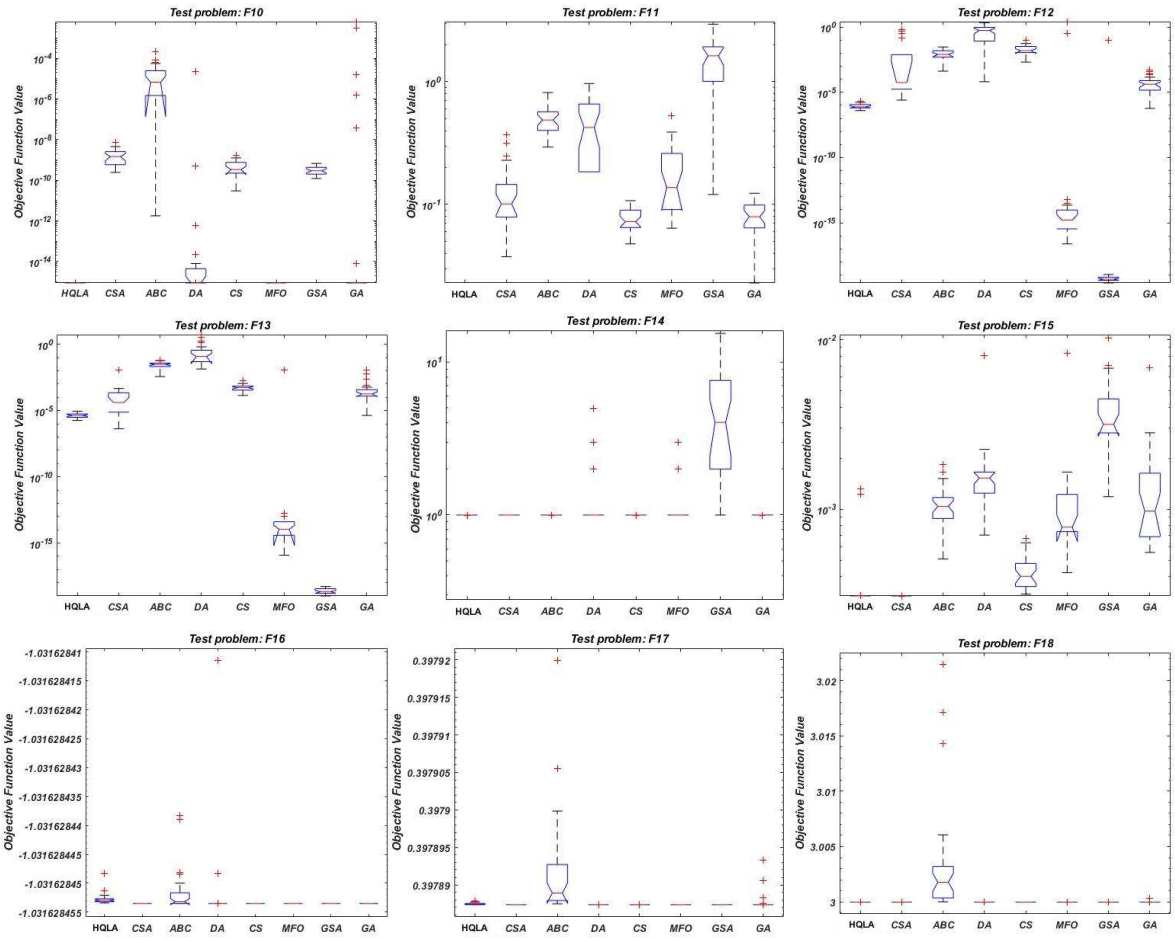

Figure 4.8: Boxplot of the results in F10-F18 benchmarks.

The second and third family of the benchmarks are multimodal and fixed-dimension multimodal benchmarks. These benchmarks contain several local optima that make the solution process a complicated task. To perform well in solving these benchmark functions, the algorithms should maintain an excellent balance among exploration and exploitation. This will help the algorithms avoid local optimum. Figures 4.13-4.14 show a schematic view of these benchmark functions in 2D.

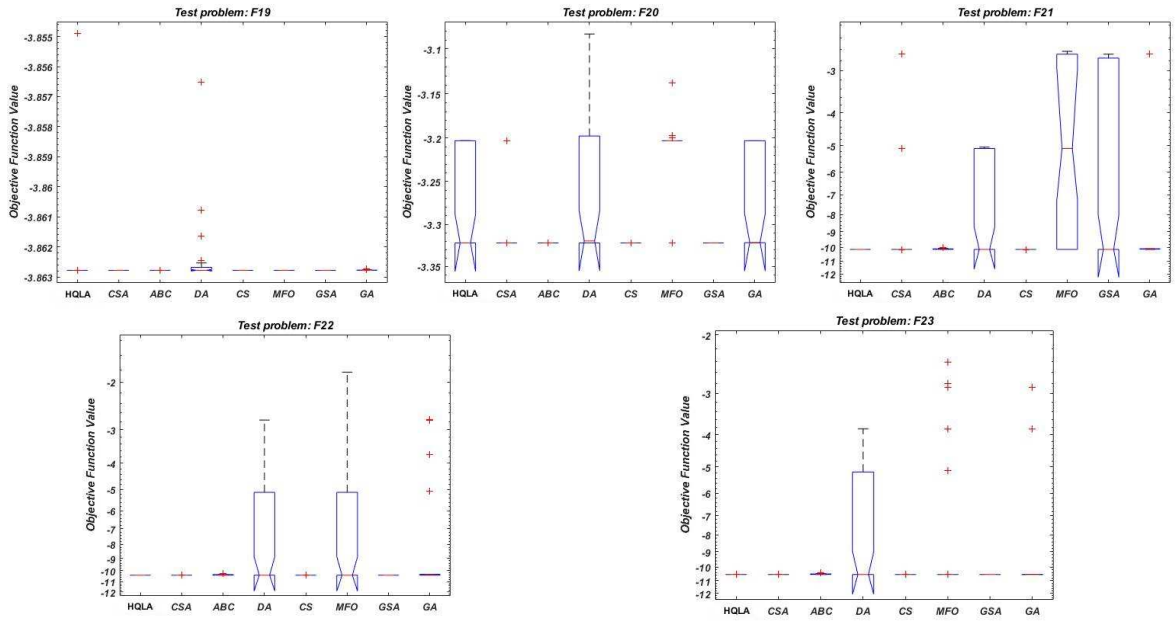

Figure 4.9: Boxplot of the results in F19-F23 benchmarks.

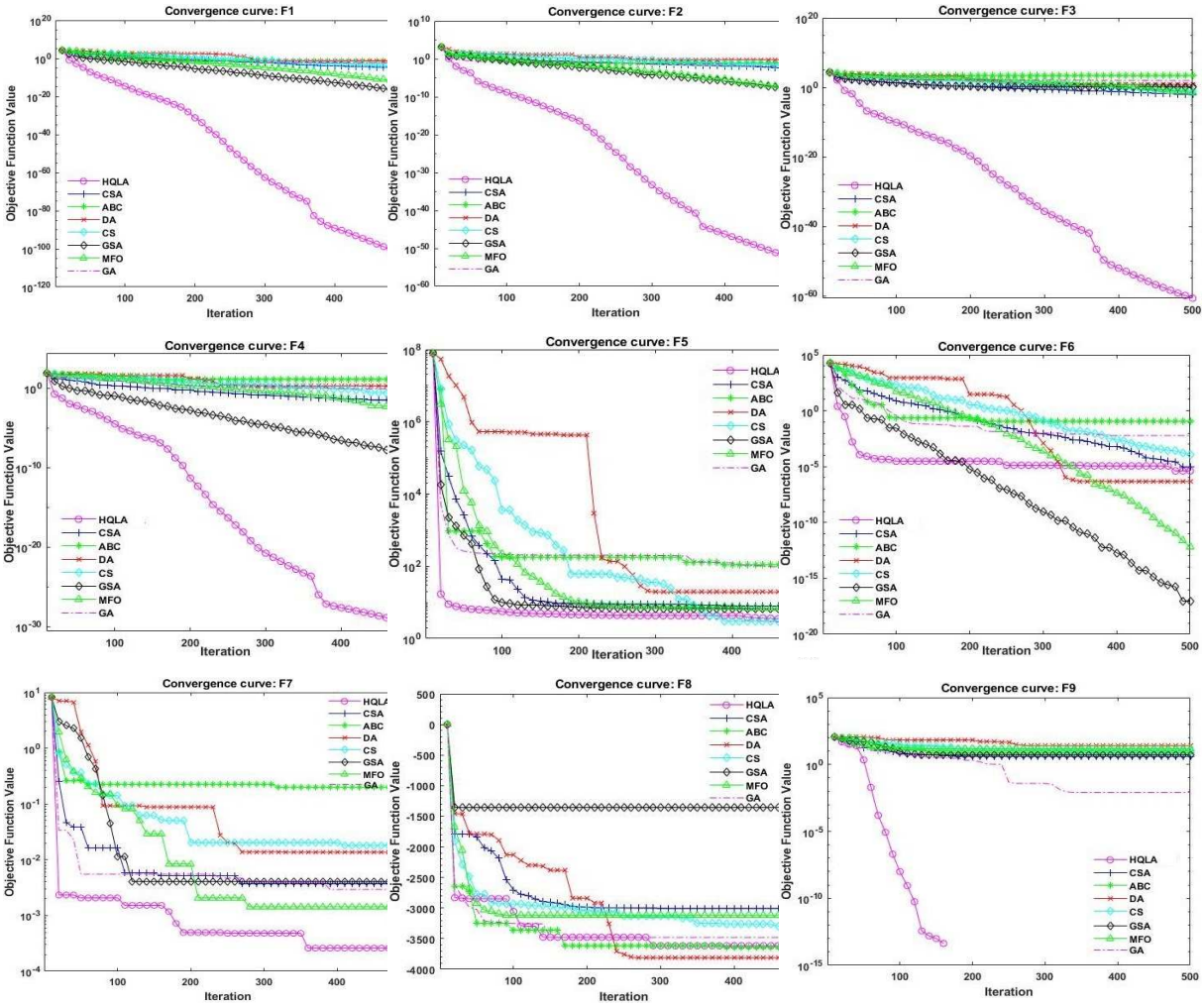

Figure 4.10: Convergence Plot of the algorithms.

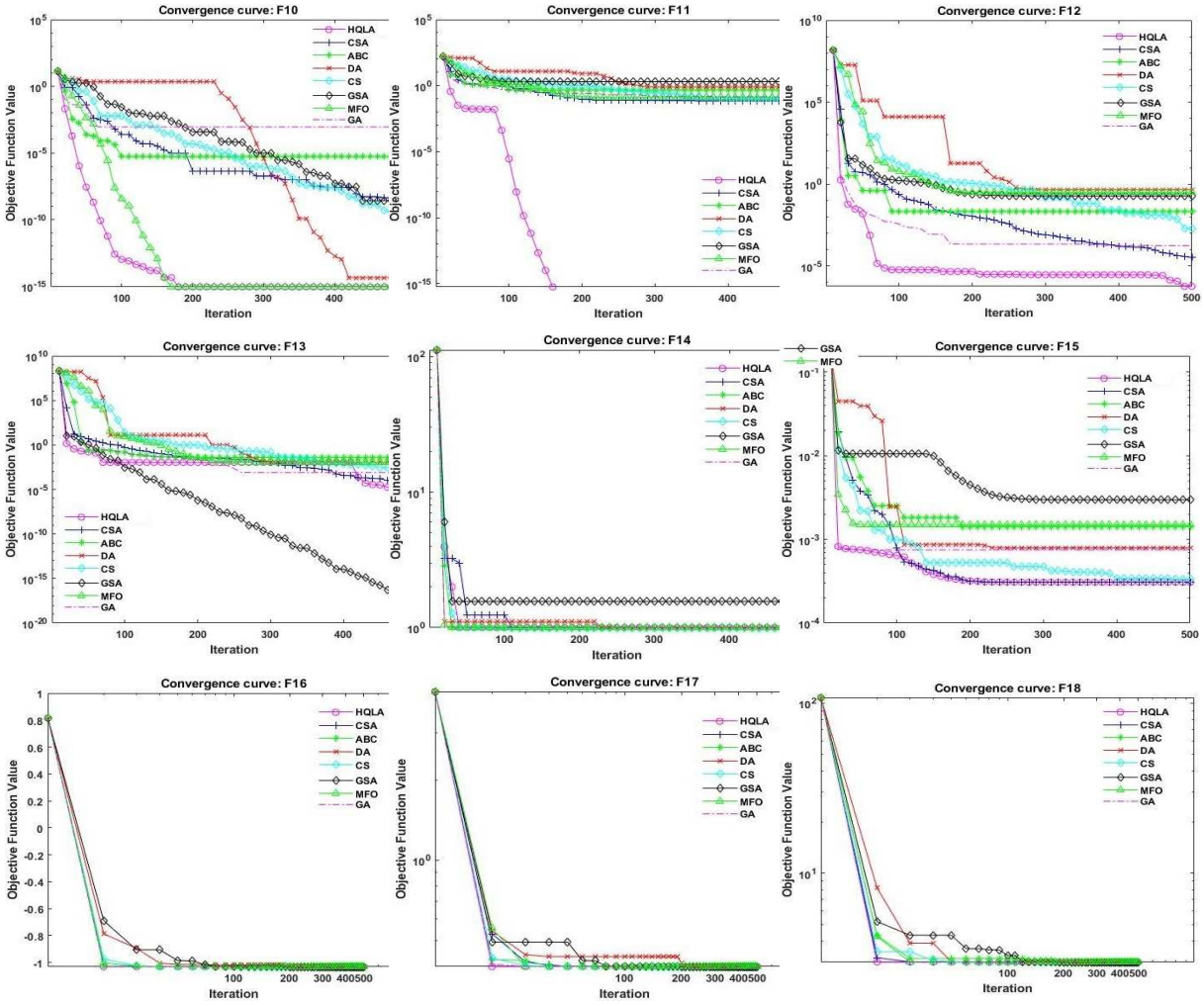

Figure 4.11: Convergence Plot of the algorithms.

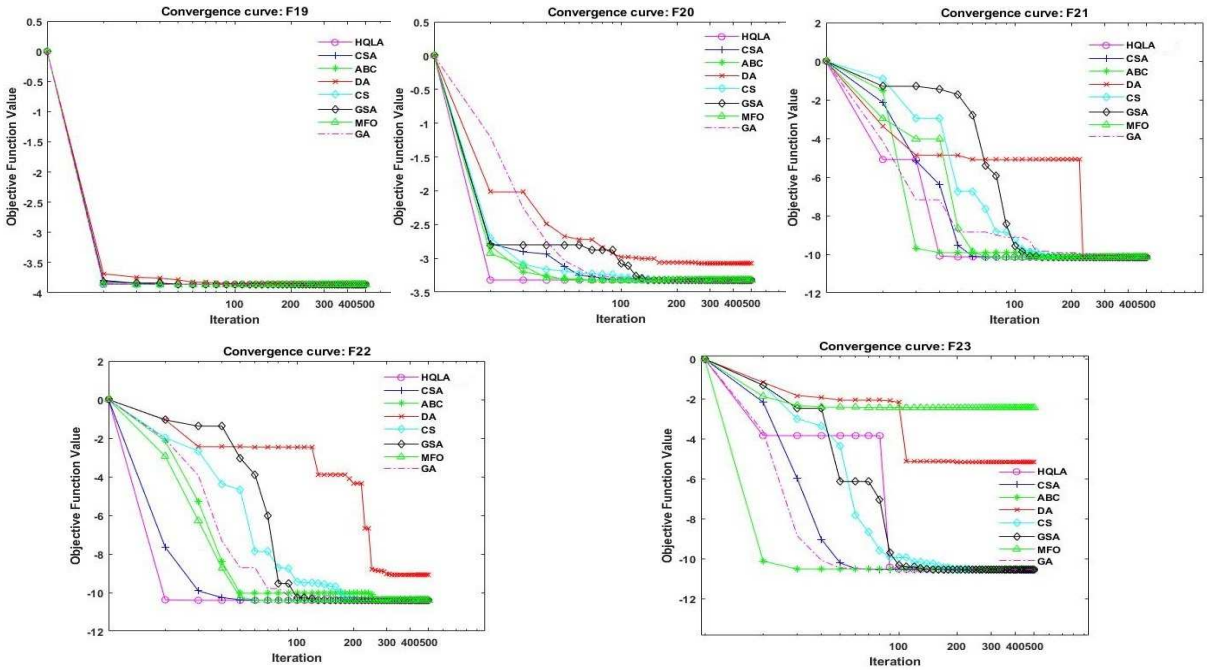

Figure 4.12: Convergence Plot of the algorithms.

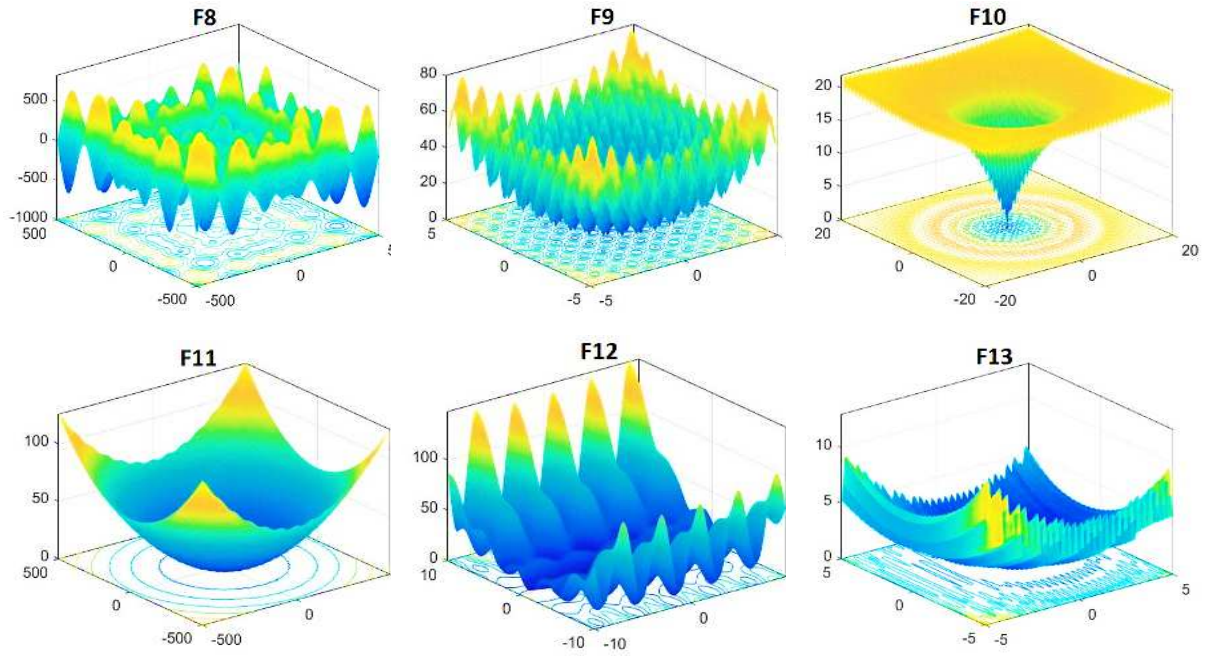

Figure 4.13: 2D representation of F8-F13.

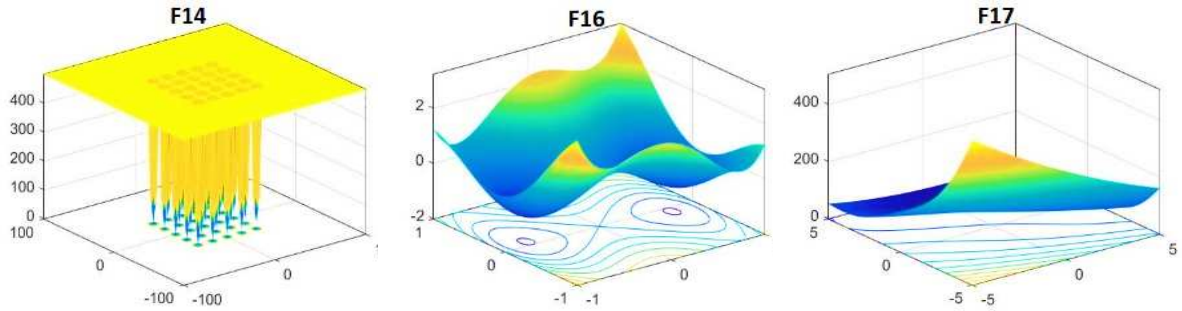

Figure 4.14: 2D representation of F14-F17.

We provide detailed information on the performance of the algorithms on solving multimodal and fixed-dimension multimodal benchmarks in Tables 4.4-4.5. Based on the results, we observe that in F8-F12, F14, and F16-23 (14 out of 16) benchmark functions, the HQLA outperforms other algorithms considering average, standard deviation, best and worst values over 30 repetitions. In some of these benchmarks, such as F9 and F11, the standard deviation of the results provided by HQLA is zero. In these benchmark functions, the HQLA achieves global optima in all the repetitions. These results indicate that HQLA is a robust and reliable algorithm in solving complex optimization problems. HQLA is able to choose between several operators that enable the algorithms to explore and exploit the solution region intelligently. The learning process in the HQLA helps the algorithm evaluate the efficiency of the operators and discover the most efficient operator (action) for any problem. Besides, Figures 4.10, 4.11, and 4.12 depict the perfect balance among exploration and exploitation in the performance of the HQLA throughout iterations. Moreover, in F13 and F15, the HQLA ranked second among all algorithms, which shows its high capability in solving optimization problems. Furthermore, Figures 4.7, 4.8, and 4.9 , disclose that the boxplot of the outcomes of the HQLA is narrower and lower than any other algorithm that highlights the superiority of the proposed methodology over existing approaches. The last family of the benchmark functions is hybrid composite benchmarks that are the most challenging (Liang et al., 2005). Figure 4.15 presents a graphical representation of these benchmark functions in 2D.

Table 4.4: Computational outcomes of the algorithms in solving multimodal benchmark functions

		HQLA	CSA	ABC	DA	CS	MFO	GSA	GA
F8	Average	**-3.59E+03**	-2.74E+03	-3.61E+03	-2.98E+03	-3.46E+03	-3.42E+03	-1.56E+03	-3.75E+03
	Std Dev	**214.4446**	283.4627	90.24338	376.9813	98.34133	375.2893	244.7475	155.5952
	Worst	**-3.08E+03**	-2.22E+03	-3.44E+03	-2.39E+03	-3.31E+03	-2.64E+03	-1.21E+03	-3.36E+03
	Best	**-4.07E+03**	-3.38E+03	-3.77E+03	-3.75E+03	-3.70E+03	-3.97E+03	-2.15E+03	-4.19E+03
F9	Average	**0.00E+00**	7.69E+00	6.16E+00	2.64E+01	1.18E+01	2.18E+01	3.78E+00	7.36E-03
	Std Dev	**0**	4.149784	1.863144	10.72399	2.272461	11.97422	2.06159	0.007922
	Worst	**0.00E+00**	1.99E+01	1.05E+01	4.08E+01	1.71E+01	4.97E+01	8.95E+00	4.34E-02
	Best	**0.00E+00**	2.98E+00	2.22E+00	8.02E+00	7.53E+00	6.96E+00	9.95E-01	5.32E-04
F10	Average	**8.88E-16**	1.83E-09	2.26E-05	7.51E-07	4.88E-10	8.88E-16	3.19E-10	3.10E-04
	Std Dev	**9.86E-32**	1.49E-09	4.01E-05	4.04E-06	3.96E-10	9.86E-32	1.43E-10	0.001217
	Worst	**8.88E-16**	7.16E-09	2.09E-04	2.25E-05	1.66E-09	8.88E-16	6.78E-10	6.10E-03
	Best	**8.88E-16**	2.45E-10	1.78E-12	8.88E-16	3.02E-11	8.88E-16	1.22E-10	8.88E-16
F11	Average	**0.00E+00**	1.28E-01	5.01E-01	4.05E-01	7.53E-02	1.78E-01	1.52E+00	8.03E-02
	Std Dev	**0**	0.077065	0.128705	0.256938	0.015921	0.111694	0.78624	0.024518
	Worst	**0.00E+00**	3.71E-01	8.17E-01	9.68E-01	1.07E-01	5.27E-01	2.95E+00	1.23E-01
	Best	**0.00E+00**	3.75E-02	2.94E-01	0.00E+00	4.77E-02	6.40E-02	1.21E-01	2.29E-02
F12	Average	**9.15E-07**	9.39E-02	1.05E-02	6.54E-01	2.41E-02	9.36E-02	3.18E-03	8.95E-05
	Std Dev	**3.92E-07**	0.187187	0.00722	0.588009	0.020125	0.449848	0.017111	0.00012
	Worst	**1.95E-06**	6.25E-01	3.00E-02	2.17E+00	9.70E-02	2.50E+00	9.53E-02	5.19E-04
	Best	**4.02E-07**	2.52E-06	4.20E-04	6.49E-05	2.07E-03	2.41E-17	2.46E-20	5.89E-07
F13	Average	4.40E-06	1.90E-03	3.14E-02	7.02E-01	5.93E-04	1.83E-03	**2.64E-19**	8.44E-04
	Std Dev	1.68E-06	0.004102	0.013352	1.387596	0.000331	0.004095	**1.13E-19**	0.00221
	Worst	8.88E-06	1.11E-02	6.68E-02	5.22E+00	1.87E-03	1.10E-02	**5.20E-19**	1.13E-02
	Best	1.83E-06	4.27E-07	3.75E-03	1.38E-02	1.40E-04	1.21E-16	**1.01E-19**	4.31E-06
F14	Average	**9.98E-01**	9.98E-01	9.98E-01	1.30E+00	9.98E-01	1.16E+00	5.00E+00	9.98E-01
	Std Dev	**4.61E-12**	3.33E-16	2.41E-09	0.853122	3.07E-15	0.450142	3.794397	1.13E-10
	Worst	**9.98E-01**	9.98E-01	9.98E-01	4.95E+00	9.98E-01	2.98E+00	1.54E+01	9.98E-01
	Best	**9.98E-01**	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01	9.98E-01

Table 4.5: Computational outcomes of the algorithms in solving multimodal benchmark functions

		HQLA	CSA	ABC	DA	CS	MFO	GSA	GA
F15	Average	3.72E-04	**3.07E-04**	1.07E-03	1.73E-03	4.29E-04	1.37E-03	3.74E-03	1.42E-03
	Std Dev	0.000241	**1.89E-13**	0.000279	0.00127	9.41E-05	0.001882	0.001868	0.001202
	Worst	1.32E-03	**3.07E-04**	1.83E-03	8.09E-03	6.75E-04	8.33E-03	1.02E-02	6.85E-03
	Best	3.07E-04	**3.07E-04**	5.09E-04	7.03E-04	3.16E-04	4.22E-04	1.18E-03	5.56E-04
F16	Average	**-1.03E+00**	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Std Dev	**9.35E-10**	0	3.75E-09	7.56E-09	0	0	0	0
	Worst	**-1.03E+00**	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Best	**-1.03E+00**	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
F17	Average	**3.98E-01**	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	Std Dev	**1.45E-07**	1.11E-16	6.8E-06	5.18E-10	1.84E-14	1.11E-16	1.11E-16	1.2E-06
	Worst	**3.98E-01**	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	Best	**3.98E-01**	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
F18	Average	**3.00E+00**	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	Std Dev	**1.28E-07**	4.7E-15	0.005074	1.74E-06	4.97E-15	2.52E-15	3.35E-15	5.38E-05
	Worst	**3.00E+00**	3.00E+00	3.02E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	Best	**3.00E+00**	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
F19	Average	**-3.86E+00**	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
	Std Dev	**0.001966**	2.66E-15	3.84E-09	0.001171	2.66E-15	2.66E-15	2.66E-15	9.07E-06
	Worst	**-3.85E+00**	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
	Best	**-3.86E+00**	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
F20	Average	-3.28E+00	**-3.32E+00**	-3.32E+00	-3.27E+00	-3.32E+00	-3.21E+00	-3.32E+00	-3.28E+00
	Std Dev	0.056061	**0.021345**	2.86E-05	0.075844	2.94E-07	0.038462	1.33E-15	0.05714
	Worst	-3.20E+00	**-3.20E+00**	-3.32E+00	-3.08E+00	-3.32E+00	-3.14E+00	-3.32E+00	-3.20E+00
	Best	-3.32E+00	**-3.32E+00**	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00	-3.32E+00
F21	Average	**-1.02E+01**	-9.74E+00	-1.01E+01	-8.47E+00	-1.02E+01	-6.65E+00	-7.62E+00	-8.64E+00
	Std Dev	**0.00034**	1.592751	0.024798	2.383475	4.19E-07	3.389785	3.414204	2.977379
	Worst	**-1.02E+01**	-2.68E+00	-1.00E+01	-5.06E+00	-1.02E+01	-2.63E+00	-2.68E+00	-2.68E+00
	Best	**-1.02E+01**	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01	-1.01E+01
F22	Average	**-1.04E+01**	-1.04E+01	-1.04E+01	-7.96E+00	-1.04E+01	-8.67E+00	-1.04E+01	-8.74E+00
	Std Dev	**0.000252**	4.39E-13	0.034619	2.851794	7.13E-07	2.936026	0	2.992735
	Worst	**-1.04E+01**	-1.04E+01	-1.02E+01	-2.77E+00	-1.04E+01	-1.84E+00	-1.04E+01	-2.75E+00
	Best	**-1.04E+01**	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01
F23	Average	**-1.05E+01**	-1.05E+01	-1.05E+01	-8.52E+00	-1.05E+01	-8.84E+00	-1.05E+01	-9.58E+00
	Std Dev	**0.000296**	1.65E-11	0.032012	2.661454	2.18E-05	3.106381	8.88E-15	2.447214
	Worst	**-1.05E+01**	-1.05E+01	-1.04E+01	-3.84E+00	-1.05E+01	-2.42E+00	-1.05E+01	-2.87E+00
	Best	**-1.05E+01**	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01

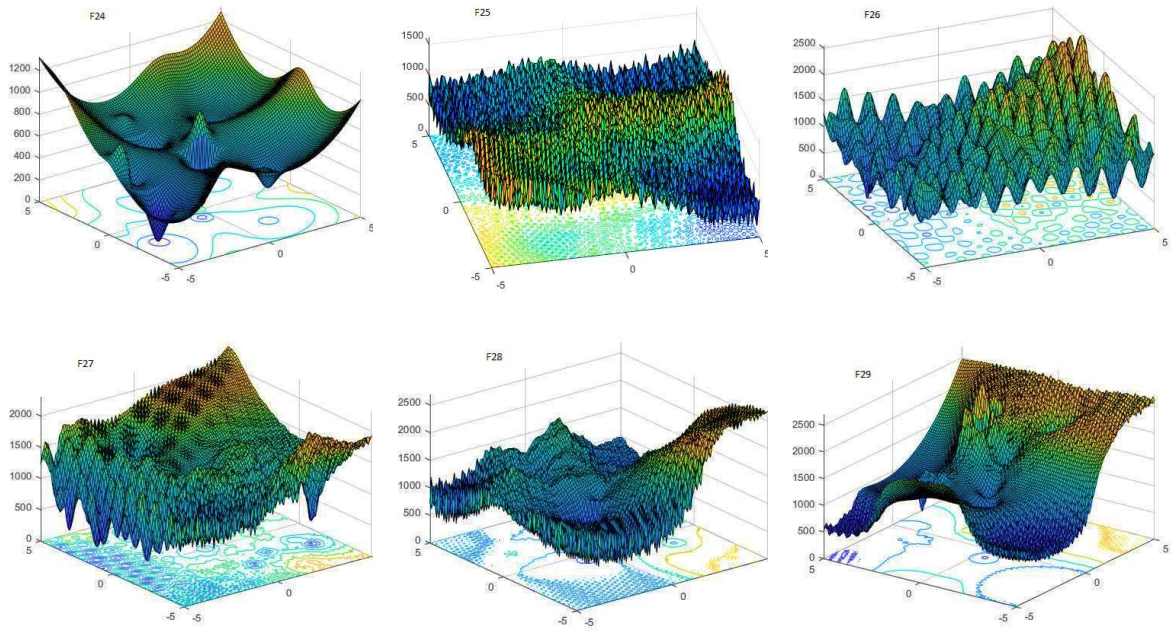

Figure 4.15: 2D representation of F24-F29.

Computational results of solving this family of benchmark functions using each algorithm are available in Table 4.6. Considering the results, we observe that the HQLA provides the best results and outperforms other algorithms in solving hybrid composite benchmarks. Table 4.6 shows that the HQLA obtained the lowest average, standard deviation, and best values for these benchmark functions while maintaining the lowest worst value. Besides, Figure 4.17 approves this statement by showing that the boxplot of the HQLA is narrower and lower than any other algorithms. Moreover, Figure 4.16 shows that the HQLA adjusts exploration and exploitation by intelligently choosing the best operators in each iteration. Based on Figure 4.16, we observe that the HQLA can improve the best solution consistently throughout iterations.

Table 4.6: Computational outcomes of the algorithms in solving composite benchmark functions.

		HQLA	CSA	ABC	DA	CS	MFO	GSA	GA
F24	Average	**6.286974**	89.47798	74.9472	127.8248	11.7614	102.5087	229.0856	129.6113
	Std Dev	**2.737264**	103.3183	14.01152	116.2458	2.422529	67.72289	76.66939	146.3853
	Worst	**9.41E+00**	2.48E+02	1.02E+02	4.62E+02	1.84E+01	1.78E+02	2.81E+02	5.04E+02
	Best	**5.48E-01**	6.91E+00	5.77E+01	4.92E+01	8.76E+00	4.45E+00	2.86E+00	1.65E+01
F25	Average	**4.051293**	41.6118	74.98699	92.44491	34.89759	89.83728	202.1975	64.50657
	Std Dev	**1.305113**	36.35972	12.03202	94.10962	53.20973	87.77274	150.2123	83.70854
	Worst	**5.78E+00**	1.03E+02	9.35E+01	2.34E+02	1.81E+02	2.57E+02	5.34E+02	2.37E+02
	Best	**1.64E+00**	3.22E+00	5.10E+01	6.72E+00	5.45E+00	1.76E-14	1.37E+00	2.91E+00
F26	Average	**5.800523**	89.67129	75.67468	143.2063	11.27073	70.93509	241.6218	188.6492
	Std Dev	**2.535319**	104.9326	12.05648	146.979	2.933045	69.55739	81.02934	162.9257
	Worst	**9.23E+00**	2.54E+02	8.96E+01	5.31E+02	1.71E+01	2.25E+02	2.88E+02	5.15E+02
	Best	**3.27E-01**	1.35E+00	5.12E+01	2.82E+01	7.68E+00	6.34E+00	2.90E+00	2.50E+01
F27	Average	**5.824902**	35.16891	77.9421	161.152	9.838745	76.15984	186.7724	98.29543
	Std Dev	**2.358736**	68.64897	16.40409	151.9523	2.625766	81.88245	123.7938	77.67603
	Worst	**8.89E+00**	2.41E+02	1.00E+02	5.50E+02	1.31E+01	2.26E+02	3.34E+02	2.37E+02
	Best	**1.85E+00**	4.13E+00	4.27E+01	4.24E+01	3.38E+00	5.33E+00	4.40E+00	1.80E+01
F28	Average	**5.907383**	104.8916	67.73344	105.7646	11.92527	102.206	202.6379	85.30539
	Std Dev	**2.128054**	114.5205	14.15686	66.0598	2.57178	75.31781	137.0966	63.38328
	Worst	**9.12E+00**	2.68E+02	9.49E+01	2.37E+02	1.66E+01	2.26E+02	4.31E+02	2.35E+02
	Best	**2.34E+00**	8.17E+00	4.91E+01	4.95E+01	8.13E+00	7.10E+00	5.02E+00	1.69E+01
F29	Average	**5.907383**	104.8916	67.73344	105.7646	11.92527	102.206	202.6379	85.30539
	Std Dev	**2.128054**	114.5205	14.15686	66.0598	2.57178	75.31781	137.0966	63.38328
	Worst	**9.12E+00**	2.68E+02	9.49E+01	2.37E+02	1.66E+01	2.26E+02	4.31E+02	2.35E+02
	Best	**2.34E+00**	8.17E+00	4.91E+01	4.95E+01	8.13E+00	7.10E+00	5.02E+00	1.69E+01

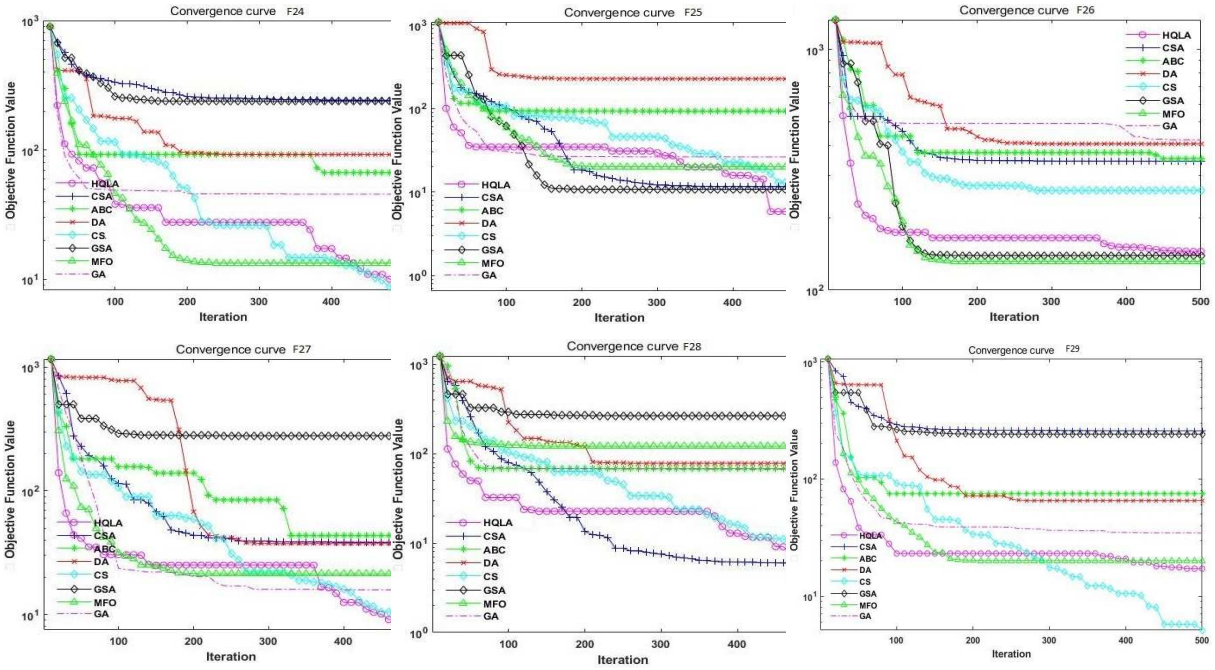

Figure 4.16: Convergence Plots for algorithms in solving composite problems.

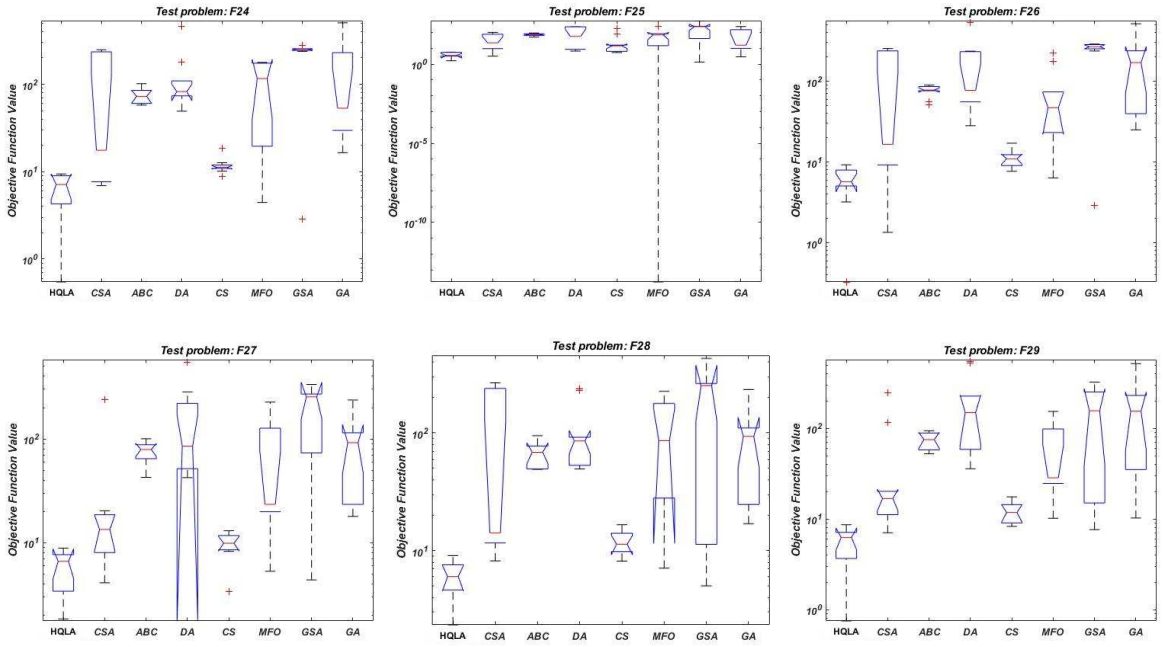

Figure 4.17: Boxplot of the results in Composite benchmarks.

Although we showed that the HQLA outperforms other state-of-the-art methods in terms of solution performance, we apply Friedman's test, which is a powerful statistical test, to show the

statistical superiority of our algorithm to other methods. Table 4.7 presents the results of Friedman's test. Based on the outcomes, the average rank of the HQLA is 3.036207, which is far smaller than that of other algorithms. Therefore, from a statistical point of view, HQLA performs significantly better than all other algorithms. We note that we perform the Friedman's test at a 95 percent confidence level.

Table 4.7: Results of the Friedman's test.

	HQLA	**CSA**	**ABC**	**DA**	**CS**	**MFO**	**GSA**	**GA**
F1	1.033333	4.233333	7.233333	7.266667	4.9	3.033333	2.033333	6.266667
F2	1	4.1	6.966667	7.9	5.3	2.633333	2.7	5.4
F3	1	2.8	7.933333	6.166667	3.933333	2.433333	5.133333	6.6
F4	1.1	3.833333	8	5.966667	5.2	3.433333	2.1	6.366667
F5	2.133333	3.666667	7.1	6.833333	2.8	4.466667	4.033333	4.966667
F6	3.366667	3.966667	7.233333	7.3	4.766667	2	1	6.366667
F7	1	2.833333	7.966667	6.4	6.1	4.333333	4.533333	2.833333
F8	3.266667	6.6	2.8	5.733333	4.166667	3.833333	8	1.6
F9	1	4.833333	4.333333	7.466667	6.166667	6.9	3.3	2
F10	2.25	6.666667	7.8	3.066667	5.533333	2.25	5.233333	3.2
F11	1.016667	3.8	6.666667	5.983333	2.866667	4.733333	7.633333	3.3
F12	2.933333	5.2	6.033333	7.566667	6.533333	2.333333	1.2	4.2
F13	2.933333	4.3	7.066667	7.933333	5.433333	2.633333	1	4.7
F14	5.333333	2.533333	6.7	3.216667	2.766667	3.2	7.966667	4.283333
F15	2.233333	1	5.3	6.4	3.066667	4.9	7.666667	5.433333
F16	7.6	3.45	7.266667	3.883333	3.45	3.45	3.45	3.45
F17	6.9	2.816667	7.9	3.3	5.166667	2.816667	2.816667	4.283333
F18	6.9	4.233333	8	3.916667	3.416667	2.166667	4.466667	2.9
F19	6.466667	2.533333	5.233333	6.666667	2.533333	2.533333	2.533333	7.5
F20	5.366667	2.3	4.833333	6.166667	3.133333	6.516667	1.05	6.633333
F21	4.633333	2.566667	6	4.133333	3.566667	4.8	3.566667	6.733333
F22	4.95	2.866667	6.7	5.233333	4.466667	3.283333	1.566667	6.933333
F23	5.833333	2.4	7.166667	5.5	4.333333	3.166667	1.766667	5.833333
F24	1.2	4.5	5	5.9	2.6	4.9	7.1	4.8
F25	1.6	4	5.6	5.4	4.1	5	6.1	4.2
F26	1.5	3.9	5.4	5.6	2.5	4.2	7	5.9
F27	1.4	3.4	5.9	6.1	2.6	4.7	6.1	5.8
F28	1.1	4.8	4.9	5.5	2.6	5.5	6.2	5.4
F29	1	3.5	5.5	6.4	2.6	5.2	5.6	6.2
Averagee	3.036207	3.711494	6.363218	5.824138	4.02069	3.839655	4.236207	4.968391

4.5 Multi-Criteria Parameter Estimation and Curve Fitting

Quebec is one of Canada's provinces that is dealing with the COVID-19 epidemic triggered by the SARS-CoV-2 virus. The province has been reported the most COVID-19 cases that account for more than 63,713 confirmed cases of COVID-19 and 5,770 death cases by the disease. On April 29, 2020, Quebec hospitals announced that the healthcare system capacity could not respond to the influx of the COVID-19 patients to the hospitals (Weeks and Ha, 2020). On June 28, 2020, Montreal's Emergency Rooms (ERs) reported near capacity status due to limited resources (Fahmy and Ross, 2020). These highlight the need for a methodology that could accurately predict the future trend of the pandemic in the province that enables the policymakers to optimize the resource allocation to avoid loss of lives, as many scientists declared that resource shortages such as ventilator shortages are the difference between life and death for patients (Kliff et al., 2020). Developing new methodologies to predict pandemic growth is essential to optimize resource allocation and determine the optimal time to implement lockdown measures. On the other hand, we could optimize resource allocation for life-threatening cases admitted to ICUs and reduce the disease's fatality rate. In this section, we use the most recent and accurate model called SIDARTHE, presented by Giordano et al. (2020) published in Nature Medicine. The scientists showed that using the model, we could predict the future trend of the pandemic accurately. However, the solution to the problem is a cumbersome task. For more information on the SIDARTHE model, see Appendix E. In this research, we used the SIDARTHE model and applied it to real data from Quebec, Canada. In order to solve the model, we utilize HQLA. Figure 4.18 shows the convergence plot of HQLA throughout iterations. We divided the period (from January 25, 2020 (day 1) to July 19, 2020 (day 176)) into six stages as follows in which the Quebec government applied specific restrictions to control the pandemic:

- **Stage 1** (from January 25, 2020, to March 15, 2020)
- **Stage 2** (from March 15 to March 24)
- **Stage 3** (from March 24 to March 28)
- **Stage 4** (from March 28 to April 2)

- **Stage 5** (from April 2 to April 13)
- **Stage 6** (after April 13)

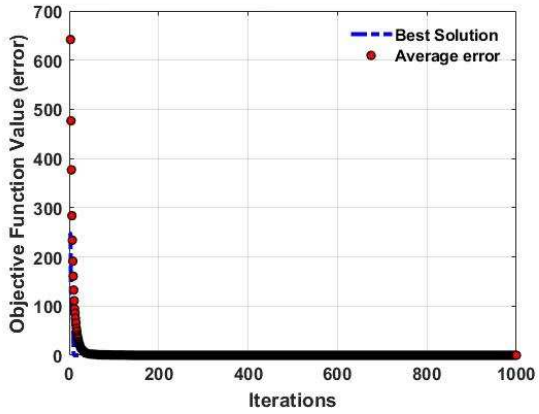

Figure 4.18: Performance of HQLA in solving the SIDARTHE model.

Table 4.8: Results of fitting the model to real-data for Quebec.

Parameters	Stages					
	after Apr 13	Apr 2 to Apr 13	Mar 28, to Apr 2	Mar 24, to Mar 28	Mar 15 to Mar 24	January 25, to March 15
α	0.088024	0.088024	0.08988	0.421243	0.421243	0.11807
β	0.000233	0.000233	0.000233	0.000752	0.000752	0.002927
δ	0.000233	0.000233	0.000233	0.000752	0.000752	0.002927
γ	0.031887	0.031887	0.071847	0.208933	0.208933	0.055155
ε	0.042366	0.013056	0.013056	0.013056	0.039123	0.039123
ζ	0.02182	0.021848	0.021848	0.084829	0.084829	0.084829
η	0.02182	0.021848	0.021848	0.084829	0.084829	0.084829
θ	0.108995	0.108995	0.108995	0.108995	0.108995	0.108995
λ	0.056476	0.056476	0.056476	0.02434	0.02434	0.02434
κ	0.017181	0.017058	0.017058	0.013091	0.013091	0.013091
ξ	0.017181	0.017058	0.017058	0.013091	0.013091	0.013091
ρ	0.017181	0.017058	0.017058	0.02434	0.02434	0.02434
σ	0.000218	0.017058	0.017058	0.013091	0.013091	0.013091
μ	0.00269	0.00269	0.00269	0.003922	0.003922	0.003922
υ	0.028218	0.028218	0.028218	0.031303	0.031303	0.031303
τ	0.009446	0.009446	0.009446	0.009446	0.009446	0.009446

We note that we considered the sum of mean square errors as the objective function value for

our model, and its optimal value for our case study is 6.29×10^{-6}. Based on the outcomes, we observe that the HQLA can solve the problem very efficiently. Table 4.8 shows detailed statistics about the optimized factors of the model. Note that the results of Table 4.8 are the output of the optimization process and solving the SIDARTHE model for Quebec data using HQLA.

In the following, we compared our results with actual data from Quebec to validate our results. Figure 4.19 shows the predicted and actual data from Quebec. In Figure 4.19, we accurately predict the number of infected cases, recovered cases, and cumulative diagnosed cases. Based on the outcomes, we conclude that the HQLA is an efficient solution methodology for the problem.

As mentioned earlier, we separated the planning horizon into six phases in which the Quebec government announced detailed limitations to control the epidemic. The first case of COVID-19 was detected in Quebec, Canada, on February 27, 2020 (Lapierre, 2020). In the first phase, the transmission rate was considered low. Based on our results, we observe that the transmission rates were low at the first stage, and the reproduction rate was $R_0 = 1.0998$. Quebec province first announced a state of emergency on March 12, 2020. We consider March 15, 2020, to March 24, 2020, as the second phase of the pandemic due to a drastic increase in the number of cases. On March 15, the Quebec government ordered the closure of all recreational and entertainment facilities (Gou, 2020).

Following the quick progress in the number of infected cases, on March 27, Montreal declared a local state of emergency, and the Quebec government ordered the closure of all universities and schools. Besides, on March 20, the province banned indoor gatherings. In the second phase of the pandemic, our study estimates a reproduction rate of $R_0 = 3.8028$ for Quebec province. From March 24, 2020, to March 28, 2020, Quebec received more COVID-19 test kits enabling the healthcare authorities to perform more tests and determine the infected cases. In this phase, we approximate the reproduction rate equal to $R_0 = 4.6096$.

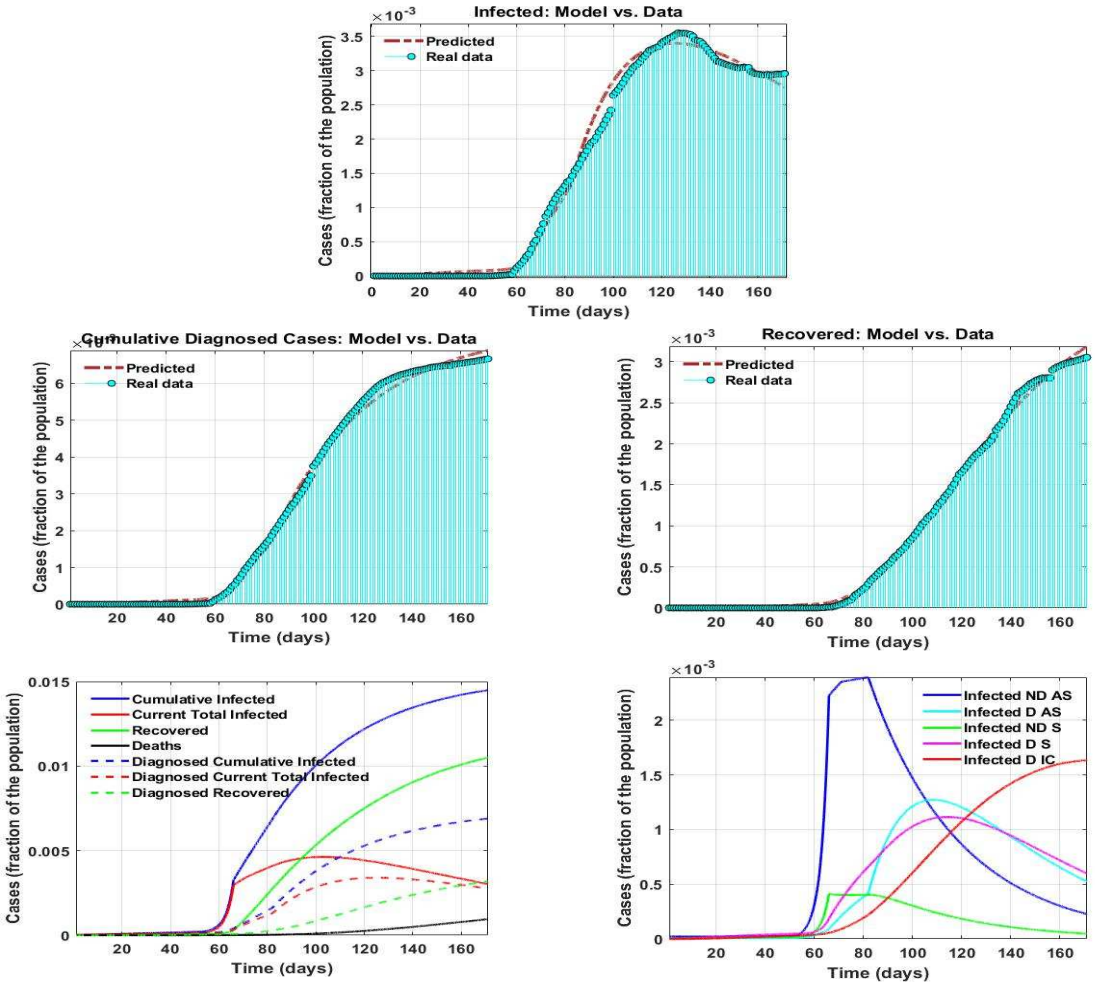

Figure 4.19: Prediction vs. data using HQLA. Non-Diagnosed Asymptomatic (ND AS), Diagnosed Asymptomatic (D AS), Non-Diagnosed Symptomatic (ND S), Diagnosed Symptomatic (DS), and Diagnosed with Life-Threatening Symptoms (D IC).

From March 28, 2020, to April 2, 2020, strict actions taken by the government decreased the reproduction rate to $R_0 = 1.1193$. The reproduction rate dropped over the next period to $R_0 = 1.0248$ from April 2, 2020, to April 13, 2020. After April 13, 2020, the reproduction rate was considered to be $R_0 = 0.7782$. In order to present the progress of the pandemic in the next few months, we extended our prediction to the next 365 days, as is shown in Figures 4.20 and 4.21.

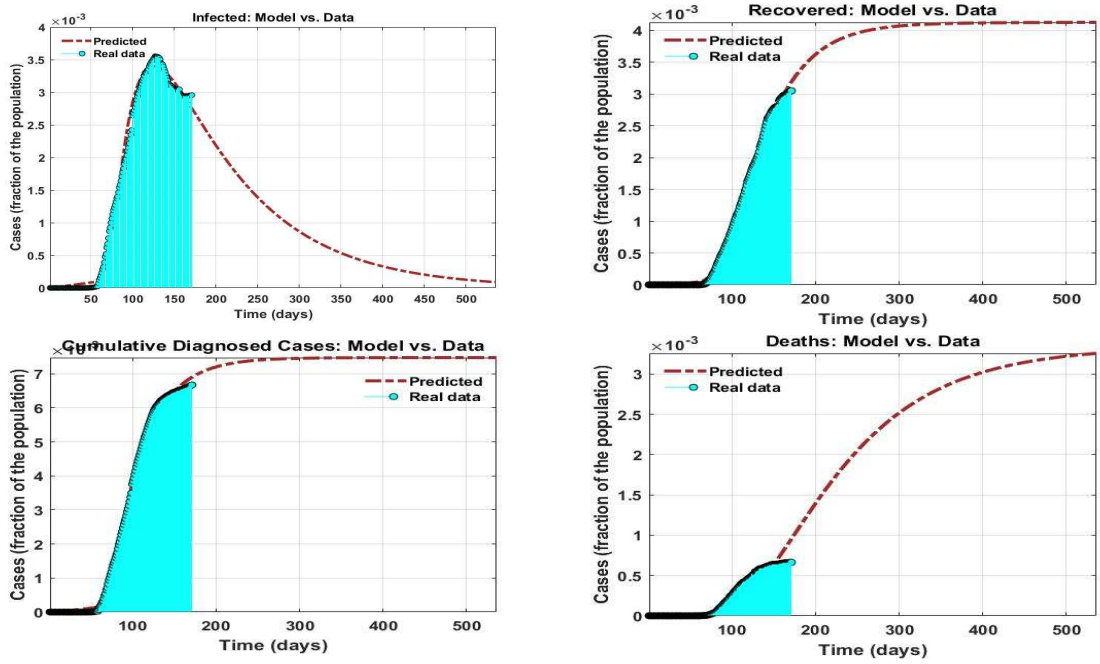

Figure 4.20: Prediction of future cases using SIDARTHE and HQLA.

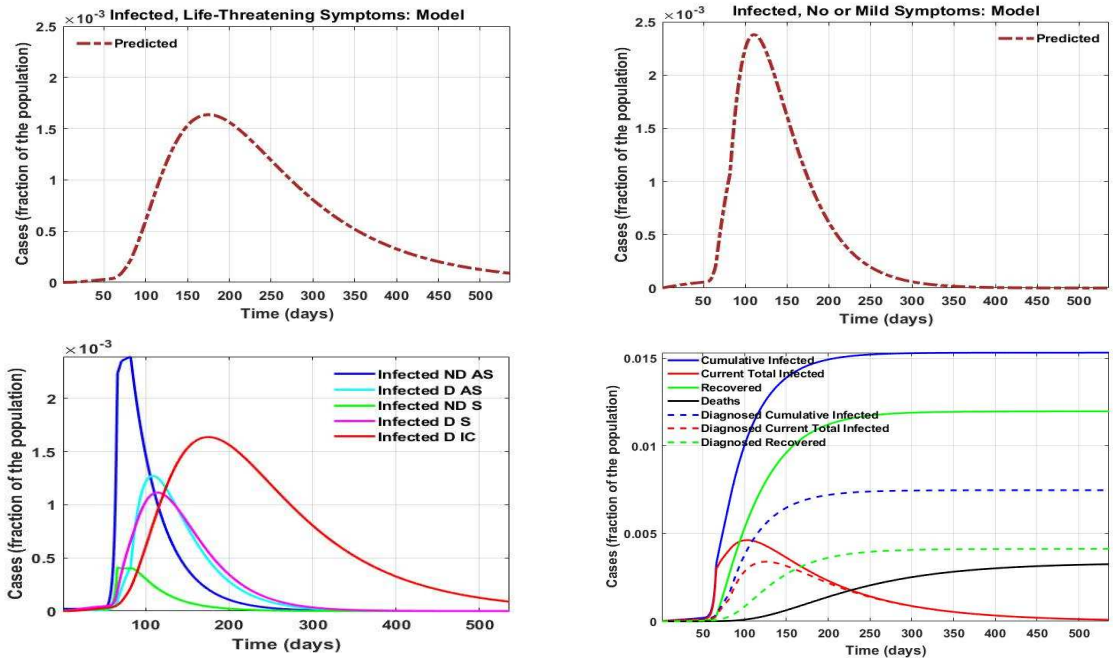

Figure 4.21: Prediction of future cases using SIDARTHE and HQLA. Non-Diagnosed Asymptomatic (ND AS), Diagnosed Asymptomatic (D AS), Non-Diagnosed Symptomatic (ND S), Diagnosed Symptomatic (DS), and Diagnosed with Life-Threatening Symptoms (D IC).

Based on our results, considering the current social distancing and limitations, we will experience a significant decrease in the number of cases in the next few months if and only if strict measures such as partial lockdown remain in place for the next few months.

4.6 Sensitivity Analyses and Managerial Insights

In the previous section, we reflected the pandemic growth over the next few months, considering the current partial lockdown and closure measures. However, in May 2020, the Quebec government ordered a gradual reopening of the businesses. Consequently, it is vital to discover how the reopening of the businesses will impact forthcoming circumstances. In this section, we examine the consequence of variation in transmission rates on the progress of the pandemic. Therefore, we augmented the parameters α, β, γ, δ, and ϵ and explore their impact on the number of infected, recovered, cumulative diagnosed, and death cases. We portray the outcomes in Figures 4.22 - 4.25.

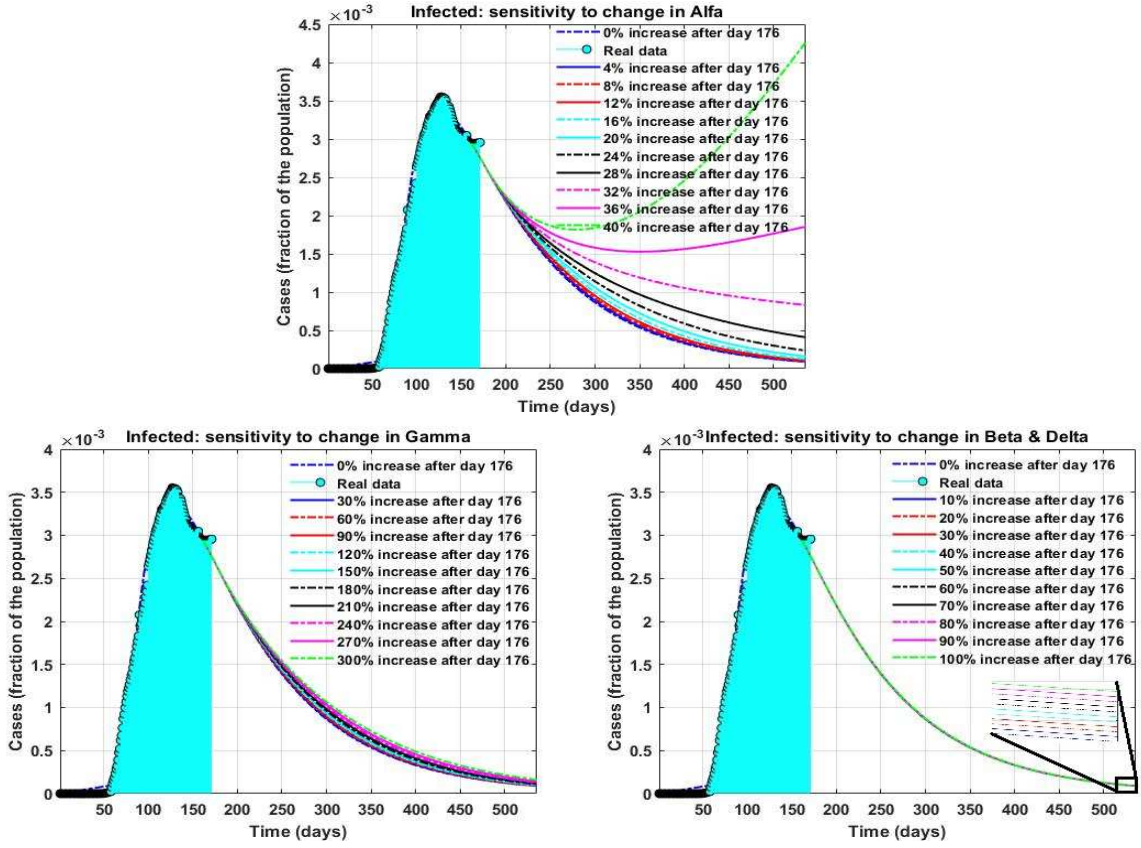

Figure 4.22: Future scenarios of the infected cases in Quebec.

The outcomes prove that the parameter α has a substantial effect on the pandemic growth, and increasing this parameter increases the number of infected, recovered, cumulative diagnosed, and death cases. Besides, increasing β, γ, and δ increases the number of infected, recovered, cumulative diagnosed, and death cases. Moreover, increasing the parameter ϵ remarkably reduces the number of infected, recovered, cumulative diagnosed, and death cases. Therefore, to decrease the spread of the virus and stop the pandemic, we need to reduce the transmission rate of the infection by applying significant social distancing and behavioral measures while increasing the detection rate of asymptomatic cases.

Based on our results, we observe that from day 250, we will see an increase in the number of infected cases from October 1, 2020, in Quebec by limiting the lockdown measures. Therefore, starting October 1, 2020, Quebec will experience a significant increase in the number of infected cases.

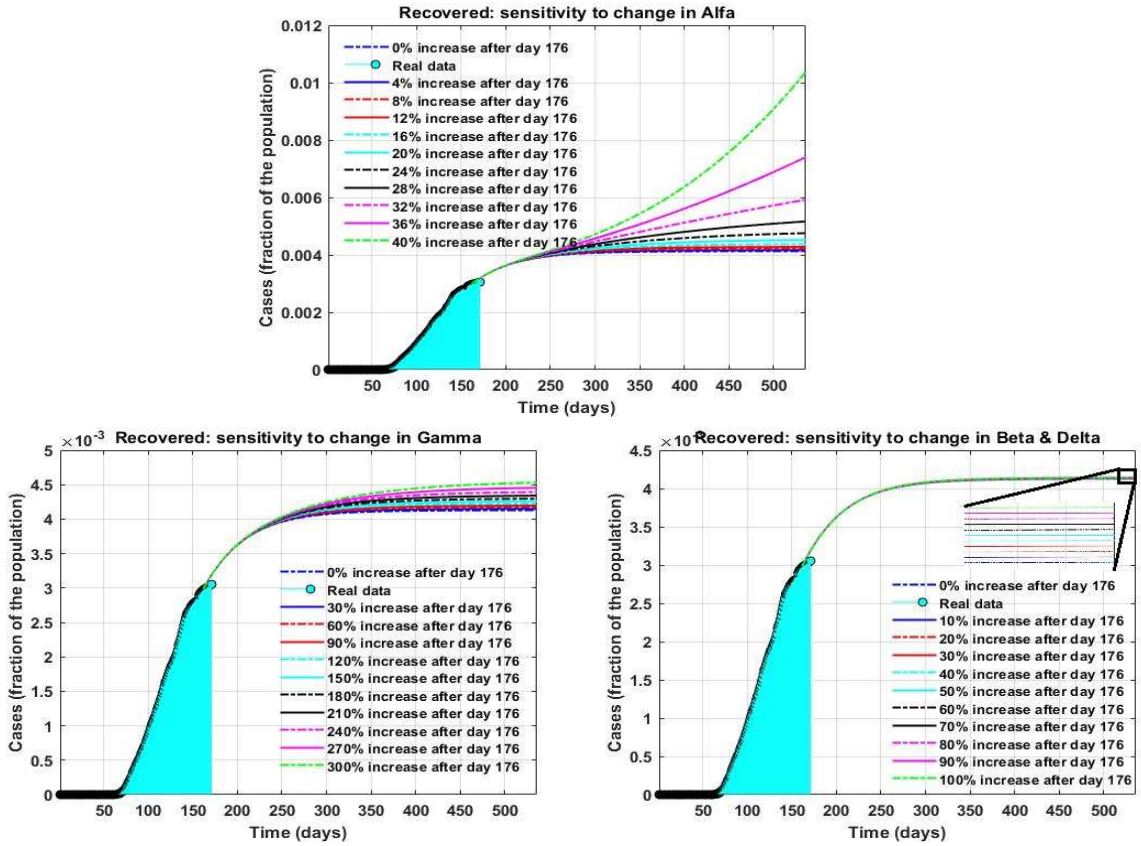

Figure 4.23: Future scenarios of the recovered cases in Quebec.

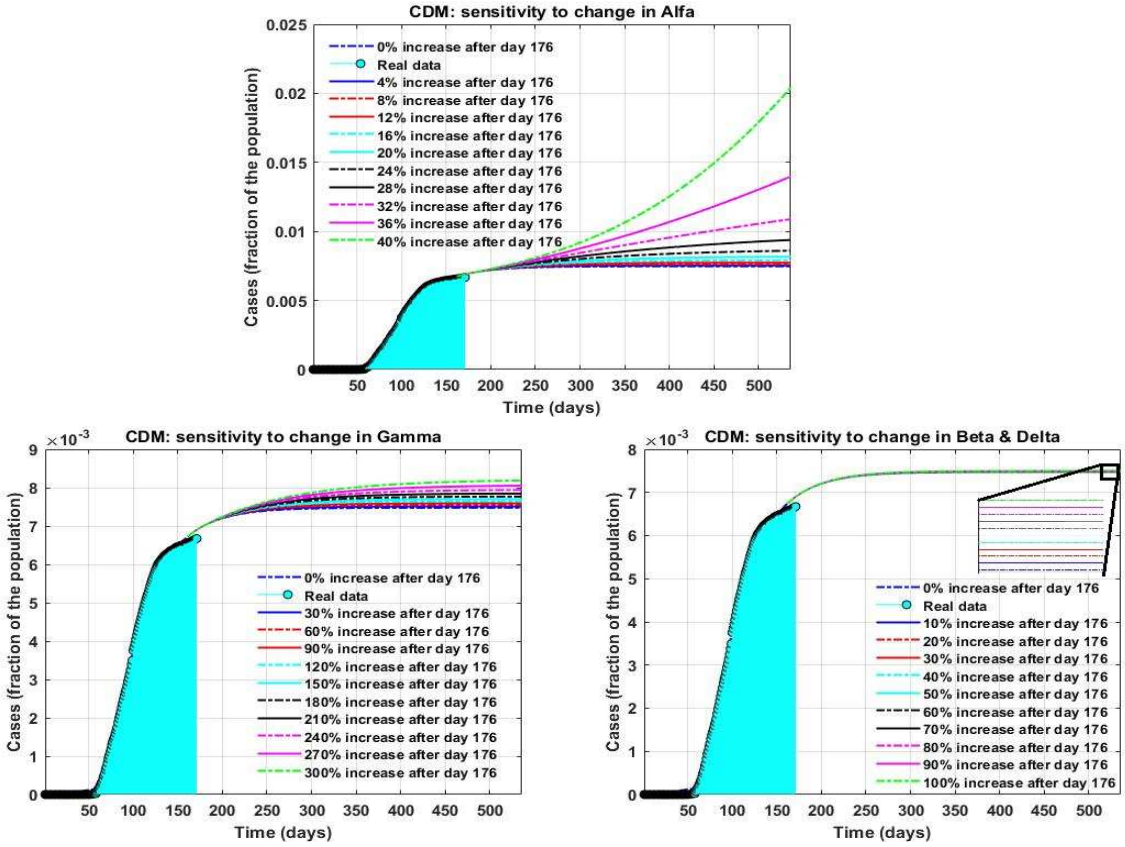

Figure 4.24: Future scenarios of the cumulative diagnosed cases in Quebec.

4.7 Conclusion

COVID-19 is a severe health threat, and the condition is growing every day. Therefore, presenting new procedures to model and predict the COVID-19 pandemic is vital. Prediction of the COVID-19 pandemic will enable policymakers to optimize the use of healthcare system capacity and resource allocation to minimize the fatality rate. In this research, we design a new hybrid reinforcement learning-based algorithm capable of solving complex optimization problems. We applied our algorithm to several well-known benchmarks and show that the presented methodology provides high-quality solutions for the most complex problems in the literature. Besides, we showed the superiority of the offered method to state-of-the-art methods through several measures. Moreover, to demonstrate the efficiency of the proposed methodology in optimizing real-world problems, we implemented our approach to the most recent data from Quebec, Canada, to predict the COVID-19

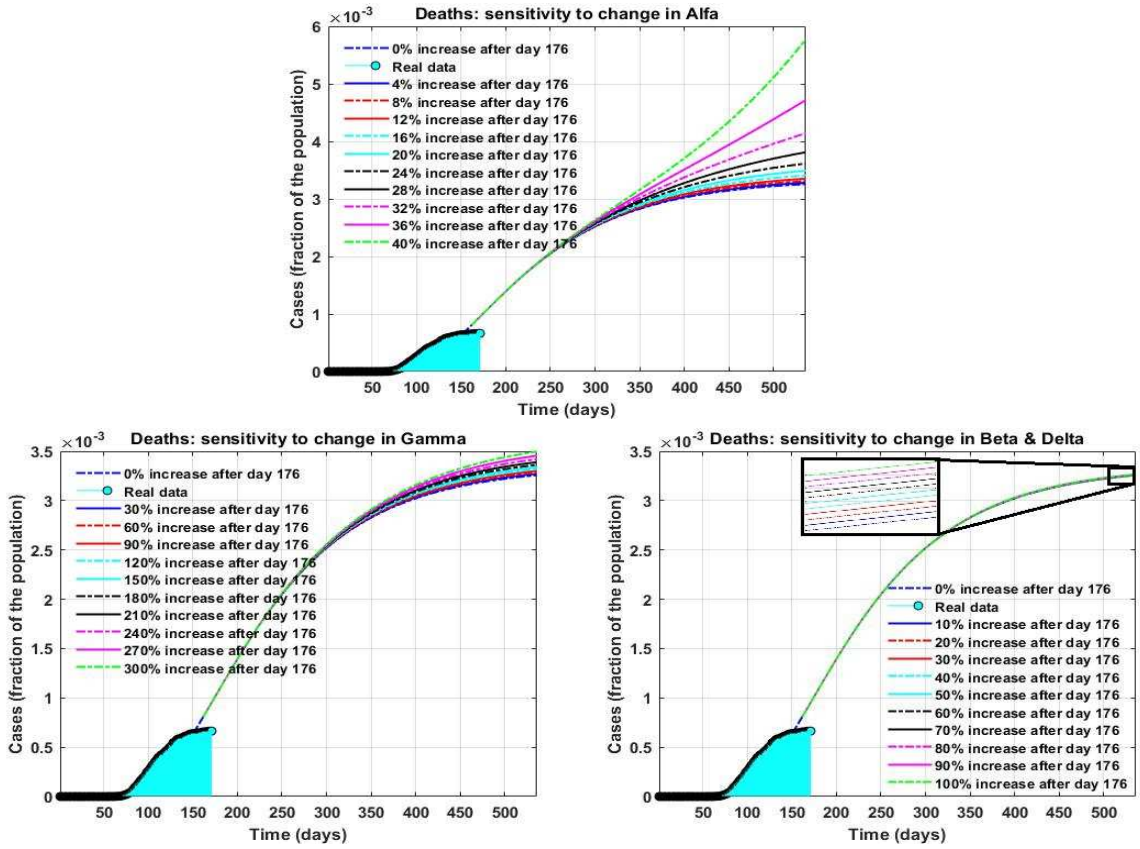

Figure 4.25: Future scenarios of the death cases in Quebec.

outbreak. Our algorithm, combined with the most recent mathematical model for COVID-19 pandemic prediction, accurately reflected the future trend of the pandemic. Furthermore, we analyzed several scenarios for deepening our insight into pandemic growth. We determined essential factors and delivered various managerial insights to help policymakers making decisions regarding future social measures. Our results showed that the transmission rate caused by the interaction of a susceptible case with an asymptomatic case has the most significant effect on future trends. Increasing this parameter can significantly increase the number of infected, recovered, cumulative diagnosed, and death cases. Besides, increasing the transmission rate due to contact of a susceptible case with a diagnosed, ailing, and recognized case increases the number of infected, recovered, cumulative diagnosed, and death cases. Moreover, increasing the parameter ε remarkably reduces the number of infected, recovered, cumulative diagnosed, and death cases. Therefore, to decrease the spread of the virus and stop the pandemic, we need to reduce the transmission rate of the infection by

applying significant social distancing and behavioral measures while increasing the detection rate of asymptomatic cases. As future research, from a mathematical modeling viewpoint, it would be worthwhile to consider stochasticity in the proposed model. Besides, it would be interesting to use the proposed model to plan for resource management during the pandemic to decrease the fatality rate by increasing the healthcare system capacity. Based on the predictions, healthcare managers can plan for testing kit allocation to test centers. Also, it would be worthwhile to consider the age, medical condition, and gender of the infected cases in the model from a modeling perspective. Moreover, it would also be interesting to use the proposed model to plan for managing healthcare resources such as Personal Protective Equipment (PPE) and Ventilators during the pandemic.

Chapter 5

Robust Modeling and Prediction of the COVID-19 Pandemic in Canada[1]

Abstract

Developing new methodologies to predict the COVID-19 pandemic will help policymakers plan to contain the spread of the virus. In this research, we develop a Stochastic Fractal Search algorithm combined with a mathematical model to forecast the pandemic. To enhance the algorithm, we employed a design of the experiments approach for tuning. We applied our algorithm to public datasets to model the COVID-19 pandemic in Canada in the upcoming months. Our algorithm predicts the number of infected, recovered, life-threatening, and death cases. We show that increasing the testing capacity would enhance the detection of asymptomatic cases and limit community transmission. Moreover, we performed sensitivity analyses to discover the effects of changes in transmission rates on pandemic growth. The sensitivity analyses provide a realistic overview of the future number of cases if the transmission rates change due to the emergence of new variants or change in social measures. Considering the outcomes, we provide several managerial insights to minimize community transmission.

[1]This paper is published in *International Journal of Production Research* in 2021 and has been cited 38 times as of March 2024.

5.1 Introduction

In December 2019, scientists announced that they have discovered a new strain of coronavirus in Wuhan, China. The virus known as SARS-Cov-2 causes a severe respiratory illness called COVID-19. Even though the officials in China took early steps to put measures to control the outbreak, the virus has spread worldwide, causing 13,036,550 cases and 571,574 deaths by July 13, 2020. Currently, in July 2020, neither medication nor vaccine is available. However, some medicines and vaccines are under development (Government of Canada, 2023). Although most cases have mild symptoms and recover with no need for hospitalization, some cases with severe symptoms need hospitalization. However, because of the limited resources, all patients cannot be admitted to the hospitals. Therefore, it is crucial to implement some restrictions and measures to contain the virus, such as social distancing or lockdown. In addition, it is essential to model and predict the spread of the virus and the peak time of the pandemic. To predict the peak time of infected and death cases, we need to know some epidemiological parameters of the pandemic, such as the reproduction rate at each stage of the pandemic. The reproduction rate presents the average number of new cases produced by one case in a population. The reproduction rate is affected by many parameters, such as the contact rate and the probability of infection transmission (Milligan and Barrett, 2015). Since COVID-19 is a new disease, epidemiological parameters of the infection and its spread are still not very well known (Teles, 2020). In recent months several researchers aimed to develop mathematical models to predict the future trends of the outbreak.

Many researchers have proposed machine learning methodologies to forecast the future trends of the outbreak in different countries (Abebe, 2020; Alamo et al., 2020; Ardabili et al., 2020; Garcia et al., 2020; Kavadi et al., 2020; Lalmuanawma et al., 2020; Malki et al., 2020; Panwar et al., 2020; Peng and Nagata, 2020; Pinter et al., 2020; Tuli et al., 2020; Wang et al., 2020). Malki et al. (2020) proposed several regressor machine learning models to determine the connection between the COVID-19 pandemic spread and various environmental measures including moisture and temperature. Kavadi et al. (2020) suggested a Partial Derivative Regression method combined with a Nonlinear Machine Learning approach to forecast the pandemic. The authors applied their methodology to data from India and showed that their model could predict future daily new cases. In

another research, Tuli et al. (2020) proposed a learning-based methodology to forecast the COVID-19 outbreak by a generalized statistical distribution. The authors used a cloud computing platform to apply their methodology to real data and obtained real-time predictions of the epidemic growth. All of these machine-learning-based approaches have some limitations that make their results less realistic and applicable; First of all, huge data are required to train these approaches. Besides, the proposed approaches do not predict the number of cases with severe and asymptomatic symptoms. Moreover, they are not based on epidemiological models and therefore they are not able to provide important epidemiological data including the reproduction rate, and different transmission rates.

Another research trend under the category of machine learning is the use of Long-Short Term Memory (LSTM) to predict the COVID-19 pandemic (Alakus and Turkoglu, 2020; Arora et al., 2020; Bouhamed, 2020; Chimmula and Zhang, 2020; Shoeibi et al., 2024; Tian et al., 2020; Yang et al., 2020). Chimmula and Zhang (2020) suggested a deep learning algorithm relied on LSTM to forecast daily new cases and the possible ending point of the outbreak. Another instance in this area is Arora et al. (2020) that used a deep learning-based approach via LSTM to predict India's future cases. Compared to the LSTM of Chimmula and Zhang (2020), the algorithm proposed by Arora et al. (2020) can only predict the number of infected cases over a short period and does not provide any insight into the ending point of the pandemic. The proposed LSTM has the same shortcomings mentioned earlier for the other machine learning algorithms. On top of those issues, the other downside of using LSTM is that it requires a large amount of memory that makes the experiments burdensome.

Another category of prediction methods for COVID-19 is based on epidemiological models (Okuonghae and Omame, 2020; Giordano et al., 2020; Higazy, 2020). Okuonghae and Omame (2020) proposed a model investigating several non-pharmaceutical criteria on the population dynamics and the illness spread. One of the most efficient approaches to predict the COVID-19 pandemic is the model presented by (Giordano et al., 2020). The authors developed a new model that considers daily new cases and predicts the number of infected, death, and recovered cases. The authors called the model SIDARTHE since it takes into account susceptible (S), infected (I), diagnosed (D), ailing (A), recognized (R), threatened (T), healed (H), and extinct (E) cases. The authors implemented the model on real data from Italy and showed that their model could predict future

trends. One of the main challenges in applying the proposed model to real-world data is the computational complexity of solving it (Giordano et al., 2020). Higazy (2020) extended the SIDARTHE model and studied the fractional-order SIDARTHE model with several policies.

This research proposes a Stochastic Fractal Search algorithm combined with the SIDARTHE model to predict the COVID-19 pandemic. The proposed methodology predicts cumulative diagnosed, recovered, death, and infected cases. The developed algorithm addresses the computational complexity of solving the SIDARTHE model for the first time. It finds a high-quality solution for the SIDARTHE model and efficiently determines many epidemiological parameters that provide vital data to policymakers and healthcare experts. We enhance our algorithm's performance using a robust design of experiment approach for parameter tuning. To validate the efficiency of the suggested methodology, we applied our algorithm on recent data from Canada to forecast the COVID-19 pandemic. We show that the algorithm solves the model in a short time with high accuracy. Using our solution approach, we derive all the epidemiological parameters of the pandemic in Canada that will significantly help the policymakers enhance the healthcare system's efficiency while minimizing the spread of the virus. Moreover, we perform sensitivity analyses to generate possible scenarios of future trends.

The remainder of this research is prepared as follows: In Section 5.2, the SIDARTHE model is presented and a Stochastic Fractal Search algorithm is developed to model the COVID-19 outbreak. We also suggest a parameter tuning methodology to enhance the performance of the algorithm. In Section 5.3, we implement the proposed methodology on real data from Canada and obtain a reliable model to predict future cases of COVID-19. In Section 5.4, we perform sensitivity analyses to determine the most critical epidemiological parameters and observe the effect of changes in these parameters on future trends of the pandemic. In Section 5.5, we conclude the paper and provide future research directions.

5.2 Mathematical model and algorithm development

Giordano et al. (2020) first proposed the SIDARTHE model which consists of eight differential equations to model the COVID-19 pandemic. The model takes into account different health states

for cases. This model projects the number of daily new cases with different types of symptoms. For more information about the SIDARTHE model, please refer to Appendix E. Giordano et al. (2020) mentioned that the solution of the suggested model is a challenging task. In the following, we present an efficient algorithm to solve this model and obtain a quality solution for the decision variables.

5.3 Stochastic Fractal Search

Salimi (2015) proposed a new algorithm called Stochastic Fractal Search (SFS). SFS uses a fundamental mathematical concept known as ”fractal”. Fractals contain complex geometric shapes that have self-similarity. SFS uses a Diffusion Limited Aggregation (DLA) to generate fractals. To mimic the DLA and explore and exploit the solution space, SFS utilizes the Gaussian walk and Lévy flight as random walks during the optimization process. Figure 5.1 presents a self-similar shape. SFS starts optimization by initiating a set of particles (random solutions in the solution

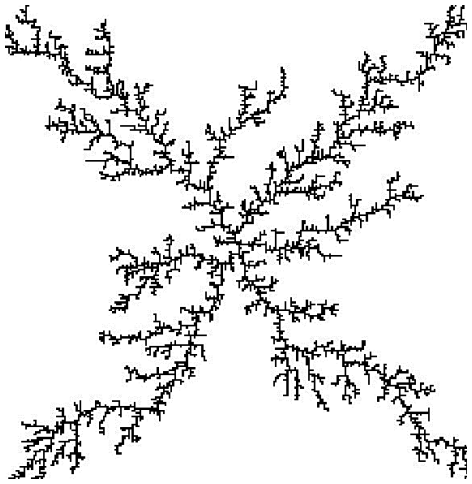

Figure 5.1: A fractal produced through the DLA method.

space). Then, it assigns a specific amount of electric energy to each particle as follows:

$$E_i = \frac{E}{P}. \tag{74}$$

In this relation, P stands for the number of particles, E represents the total initial energy, and E_i shows the amount of electric energy assigned to particle $i \in \{1, \ldots, P\}$. SFS utilized the diffusion process to produce new solutions around each solution in the initial set to create new solutions. The diffusion process is inspired by the dielectric breakdown process in which an electrically insulating material abruptly converts to an electrical conductor under immense voltage. The phenomenon allows the electric current to go through it, creating fractal shapes.

To generate new solutions, the SFS applies Lévy flight and Gaussian walk to imitate DLA development (Salimi, 2015). Although both Lévy flight and Gaussian walks generate fractals, the created clusters vary remarkably in shape. The cluster generated through Lévy flight consists of different islands (sets of short steps) linked by lengthy trips (long steps), which make the Lévy flight proper for exploration of the solution space (Chakrabarti et al., 2006). On the contrary, the Gaussian walk generates a denser and smaller cluster that contains several small steps (Chakrabarti et al., 2006). The behavior of the Gaussian walk makes it an excellent walk to enhance the exploitation of the algorithm. Therefore, SFS swaps randomly between Lévy flight and Gaussian walk to benefit from both (Salimi, 2015). Lévy distribution, denoted by $L(s)$, could be stated as the following formula (Yang, 2010):

$$L(s) = \frac{1}{\pi} \int_0^{\infty} \exp(-\alpha x^{\beta'}) \cos(sx)\, dx, \tag{75}$$

In Eq. (75), s is the input of the Lévy function, β' is a parameter in $(0, 2]$, and α denotes the distribution scale factor. We note that calculating the stepsize using relation Eq. (75) is not straightforward (Mousavirad and Ebrahimpour-Komleh, 2017). Therefore, we used the approximation algorithm proposed by Mantegna (1994) to compute the step size. Then, SFS utilizes the below equations to perform the diffusion procedure (Salimi, 2015):

$$x_i^q = x_i + \beta \times \text{Gaussian}(|BP|, \sigma) - (\epsilon \times BP - \epsilon' \times x_i), \tag{76}$$

$$x_i^q = x_i + \alpha_i^q \text{ stepsize}, \tag{77}$$

$$x_i^q = x_i + \beta \times \text{Gaussian}(|x_i|, \sigma). \tag{78}$$

In equations (76)-(78), parameter q represents the number of generated particles using the diffusion of each initial solution, and parameter σ shows the standard deviation of the Gaussian walk. Also, BP is the position of the best solution, and α_i^q stands for the coefficient of the stepsize that is updated, as we will discuss later. Besides, x_i^q represents the new solutions generated by the diffusion procedure, and x_i shows the i-th particle. In addition, ϵ' and ϵ represent randomly selected values in $(0, 1]$. Furthermore, SFS adjusts the Gaussian parameter as $\sigma = |\beta \times (x_i - BP)|$ and cuts the length of jumps as optimization continues by setting $\beta = \frac{\log(g)}{g}$ where g is the iteration number. Subsequently, the remaining solutions create a set of particles for the next iteration. Then SFS distributes the reduced energy by ignoring less efficient solutions among the new set as follows:

$$E_{\text{new}}^i = E_{\text{old}}^i + \left(\frac{fit_i}{\sum_{i=1}^{P} fit_i} \right) \omega\xi. \tag{79}$$

In equation (79), E_{old}^i and E_{new}^i represent the energy of particle i before and after the procedure. Also, fit_i is the fitness value of particle i and ω denotes the energy obtained from eliminating the low-quality particles. Moreover, ξ is the distribution rate among all particles.

In the design of metaheuristics, two main challenges should be appropriately addressed to achieve high performance: i) the updating process, and ii) exploration-exploitation tuning. To perform the updating process, numerical procedures are employed to move the particles toward the best solution. For this purpose, first, SFS ranks the solutions based on their fitness value. Then, it assigns a probability to every particle as follows:

$$pr_i = \frac{N - \text{Rank}(x_i) + 1}{N}. \tag{80}$$

In equation (80), $\text{Rank}(x_i)$ shows the rank of solution i, and N stands for the total number of solutions. Consequently, better solutions obtain higher probabilities. Then SFS updates the variables in each particle as:

$$x_i^{\text{new}}(j) = \begin{cases} x_r(j) - \xi'(x_t(j) - x_i(j)) & \text{if } \xi' > pr_i \\ x_i(j) & \text{otherwise} \end{cases} \tag{81}$$

In equation (81), x_i^{new} is the new position of x_i where index j refers to the j-th decision variable.

Also, ξ' is a random number generated using a uniform distribution in $(0, 1]$. Also, $x_r(j)$ and $x_t(j)$ are randomly selected solutions. Moreover, in the second updating procedure, SFS increases the exploration by adjusting the positions of particles as:

$$x_i^{\text{new}}(j) = \begin{cases} x_i(j) - \zeta \times (x_t(j) - BP(j)) & \text{if } \xi' \leq 0.5 \\ x_i(j) + \gamma \times (x_t(j) - x_r(j)) & \text{if } \xi' > 0.5 \end{cases} \tag{82}$$

In equation (82), ζ and γ are random numbers generated from a Gaussian Normal distribution. To ensure an appropriate trade-off in exploration and exploitation, we use the following equations that control the stepsize throughout iterations:

$$\alpha_i^q = \frac{\log(\min(E^{\wedge})) \times (UB - LB)}{g \times \log(E_i)}, \tag{83}$$

$$\alpha_i^q = \frac{(UB - LB)}{(g \times \log(E_i))^{3/2}}. \tag{84}$$

In equations (83)-(84), $\min(E^{\wedge})$ is the amount of energy for the worst solution. Also, LB and UB are the lower and upper limits of the variables, respectively. During the optimization process, we randomly use one of the equations (83)-(84) to calculate α_i^q. The pseudo-code of the SFS is presented in Algorithm 3.

5.4 Case study

Canada is one of the countries experiencing the COVID-19 pandemic. The first case of COVID-19 in Canada was reported on January 25, 2020. Canada has confirmed 107,589 COVID-19 cases and 8,783 death cases by July 13, 2020. Presenting a robust prediction model will significantly help the government control daily new cases by wisely planning for restrictions and lockdowns. Besides, the prediction model will help healthcare professionals plan for the upcoming peak by forecasting the future number of infected people with life-threatening symptoms. Using our results, healthcare managers can plan for resources in the most efficient way. We applied our methodology to the public datasets provided by the Canadian Health Authority to predict the COVID-19 outbreak in Canada. The input data used in this research is available online (Statistics Canada, 2020). The data

ranges for a period of 160 days from January 25, 2020 to July 1, 2020. In the following subsection, first, we tune the SFS to achieve the best performance of the algorithm. Then we used the tuned algorithm to model the pandemic in Canada.

Algorithm 3 Pseudocode of SFS

Enter the values of the algorithm parameters;
$It \leftarrow 0$; {Referring to iteration}
$MT \leftarrow$ Maximum iteration;
for $i = 1$ to $Npop$ **do**
  for $j = 1$ to $Nvar$ **do**
Initial solution $(i, j) \leftarrow (UB(j) - LB(j)) \cdot \text{rand}(0, 1) + LB(j)$;
  end for
end for
while $It \leq MT$ **do**
  for $i = 1$ to $Npop$ **do**
    for $j = 1$ to $Ndiff$ **do**
Generate a new particle;
    end for
  end for
Perform the updating;
Order the solutions;
  for $i = 1$ to $Npop$ **do**
    for $j = 1$ to $Nvar$ **do**
Generate a new particle;
      if probability $\leq \text{rand}(0, 1)$ **then**
Update the corresponding element;
      end if
    end for
  end for
Perform the updating;
Order the solutions;
  for each solution found while updating **do**
Generate a new particle;
    if probability $\leq \text{rand}(0, 1)$ **then**
Modify the solution;
    end if
  end for
$It \leftarrow It + 1$;
end while
return the best solution; =0

5.5 Tuning the Stochastic Fractal Search algorithm

Evolutionary-based search algorithms are based on randomization. The parameter setting of these algorithms affects the performance of the algorithm drastically. Although we cannot ensure convergence of any evolutionary algorithm, we can tune the parameters to maximize the algorithm's performance over several runs. In other words, we can minimize the deviation of the algorithm from its best performance. To tune our algorithm and find the best parameter setting, we use factorial designs or Design of Experiments (DOE) and consider the output as the response. We pass the information to the Taguchi Method that is a well-known robust design methodology. Taguchi Method works by adjusting control factors (i.e., parameters of RSFS) to minimize inconsistency in the performance of the algorithm (Roy, 2010; Taguchi et al., 2004). The SFS has four main input parameters, including the number of start points, the maximum number of generations, the number of diffusions, and the probability of Gaussian walks. The number of start points determines the number of solutions in the population during the optimization process. The maximum number of generations specifies the maximum number of iterations in the algorithm. The number of diffusions determines the number of particles generated by the diffusion of every solution. The probability of Gaussian walks shows the probability of using Gaussian walks throughout the optimization process. Table 5.1 presents the parameters of RSFS and their three levels.

Table 5.1: Different levels of the parameters in SFS.

Parameters	**LB-UB**	**Level**		
		1	2	3
Number of start points	30-70	30	50	70
Maximum number of generations	300-1000	300	700	1000
Number of diffusion	1-9	1	5	9
Probability of Gaussian walks	0-0.5	0	0.25	0.5

To design a robust algorithm, we use L^9 orthogonal array. In each row, the experiment determines different levels for each parameter and requires a response. In this research, we consider minimizing the sum of mean square errors between predicted and real data for the number of cumulative diagnosed, recovered, and infected cases as the objective function. In this study, we consider

the normalized sum of mean square errors as the response. The Taguchi method calculates a measure referred to as the Signal to Noise ratio (SN ratio) using the following formula (Najafi et al., 2009; Roy, 2010):

$$S/N = -10 \log \left(\frac{1}{m} \sum_{i=1}^{m} \frac{1}{y_i^2} \right). \tag{85}$$

In Eq. (85), m denotes the number of test runs and y_i is the response in the i-th run. Table 5.2 presents the results of the Taguchi method.

Table 5.2: Results of tuning the algorithm.

Run Order	Number of start points	Maximum number of generations	Number of diffusions	Probability of random walks	Normalized Response	S/N
1	1	1	1	1	0.032353	-29.8017
2	1	2	2	2	0.536585	-5.40722
3	1	3	3	3	1	0
4	2	1	2	3	0.338462	-9.40981
5	2	2	3	1	0.085603	-21.3502
6	2	3	1	2	0.6875	-3.25455
7	3	1	3	2	0.846154	-1.45101
8	3	2	1	3	0.511628	-5.82092
9	3	3	2	1	0.328358	-9.67304

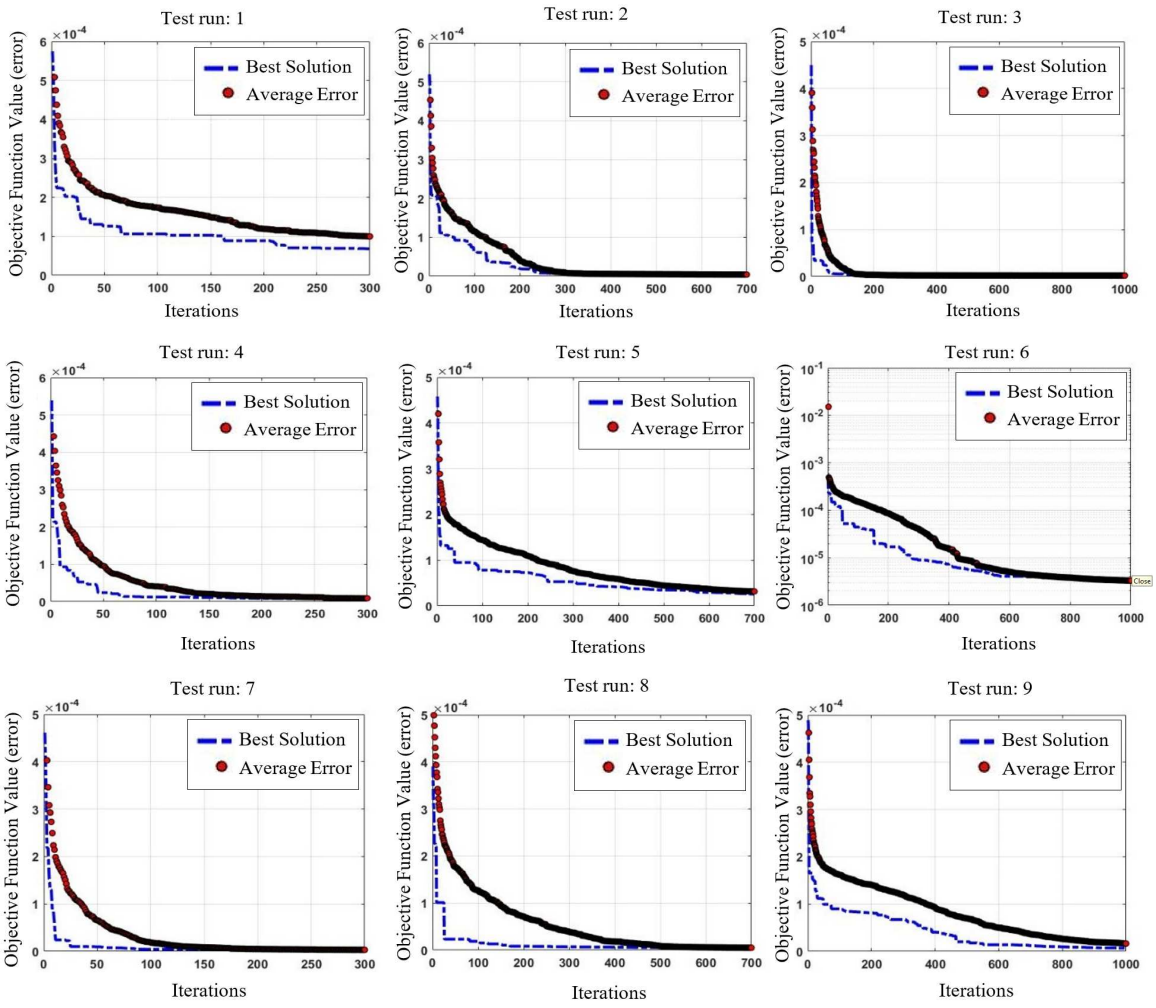

Figure 5.2: Convergence plot of the algorithms.

Figure 5.2 represents the convergence plot and the average value of the objective function (i.e., the sum of mean square errors) among all particles throughout iterations. This figure also shows that the performance of the SFS is significantly affected by the parameter setting, especially those parameters that control the exploration and exploitation ability of the algorithm. Figure 5.3 shows the results of the Taguchi method for the S/N ratio.

In this section, we report the results of our forecasting model for data in Canada based on the high-quality solution obtained by RSFS. Figure 5.4 shows the convergence curve and the sum of the mean square errors throughout iterations. We considered several periods from day 1 to day 160 to reflect the applied social measures by federal and provincial governments.

In Figure 5.4, we observe that the algorithm improves the quality of the initial solutions quickly

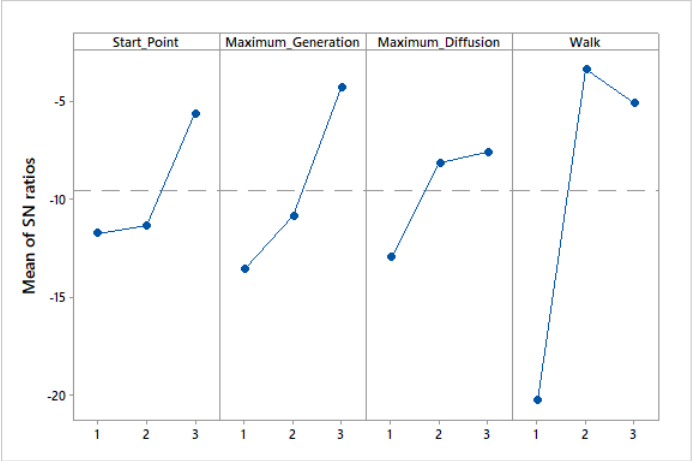

Figure 5.3: Results of parameter tuning (SN ratios).

using its efficient exploration ability. In this phase, the RSFS adjusts the Gaussian parameter as $\sigma = |\beta \times (x_i - BP)|$ and cuts the length of jumps throughout iterations by setting $\beta = \frac{\log(g)}{g}$ where g is the iteration number. Therefore, at the beginning of the optimization process, RSFS focuses more on exploration and tries to search the solution space and avoid trapping in local optima. In Figure 5.4, we can see that around iteration 50, the RSFS finds the first major local optimal, resulting in a lower objective function value for the best solution. After a few iterations, the algorithm finds a better solution and leaves the local optima. After iterations 250, the RSFS cuts the length of the random walks and focuses more on exploitation. Table 5.3 shows Canada's epidemiological parameters for each stage of the pandemic.

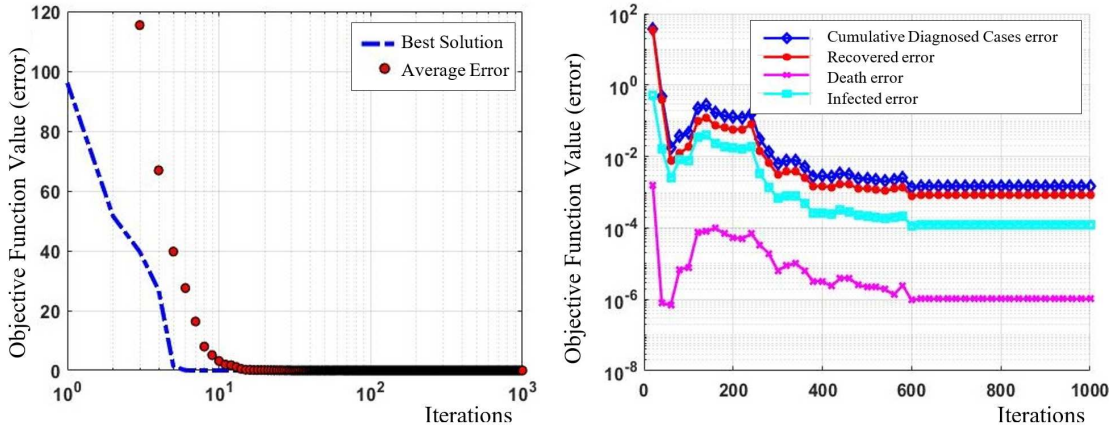

Figure 5.4: Performance of the RSFS in solving the problem.

Table 5.3: The output of the model for different stages.

Stages	after April 13	April 2 to April 13	March 28 to April 2	March 24 to March 28	March 15 to March 24	January 25 to March 15
α	0.1954	0.1954	0.2038	0.4652	0.4652	0.1779
β	0.0019	0.0019	0.0019	0.0034	0.0034	0.0048
δ	0.0019	0.0019	0.0019	0.0034	0.0034	0.0048
γ	0.1563	0.1563	0.1762	0.3037	0.3037	0.3926
ε	0.0922	0.0673	0.0673	0.0673	0.1238	0.1238
ζ	0.0374	0.0383	0.0383	0.0430	0.0430	0.0430
η	0.0374	0.0383	0.0383	0.0430	0.0430	0.0430
θ	0.1799	0.1799	0.1799	0.1799	0.1799	0.1799
λ	0.1162	0.1162	0.1162	0.0499	0.0499	0.0499
κ	0.0359	0.0346	0.0346	0.1480	0.1480	0.1480
ξ	0.0359	0.0346	0.0346	0.1480	0.1480	0.1480
ρ	0.0359	0.0346	0.0346	0.0499	0.0499	0.0499
σ	1.8783e-04	0.0346	0.0346	0.1480	0.1480	0.1480
μ	0.0121	0.0121	0.0121	0.0255	0.0255	0.0255
υ	0.0289	0.0289	0.0289	0.0336	0.0336	0.0336
τ	0.0127	0.0127	0.0127	0.0127	0.0127	0.0127

Our proposed RSFS finds the best parameters of the SIDARTHE model for each stage of the case study. Using these parameters, we have plotted Figure 5.5 that shows the number of different types of cases over time.

In Figure 5.5, we observe that the best solution provided by the RSFS algorithm perfectly fits real data of COVID-19 in Canada. Our results not only are useful to project the future number of different types of cases, but also they provide insight into some data that are unknown for health policymakers. For example, Figure 5.5 estimates the number of cases with mild symptoms or without any symptoms that are not detected but contribute significantly to community transmission. It is noteworthy that the minimum error for results is 5.41E-07.

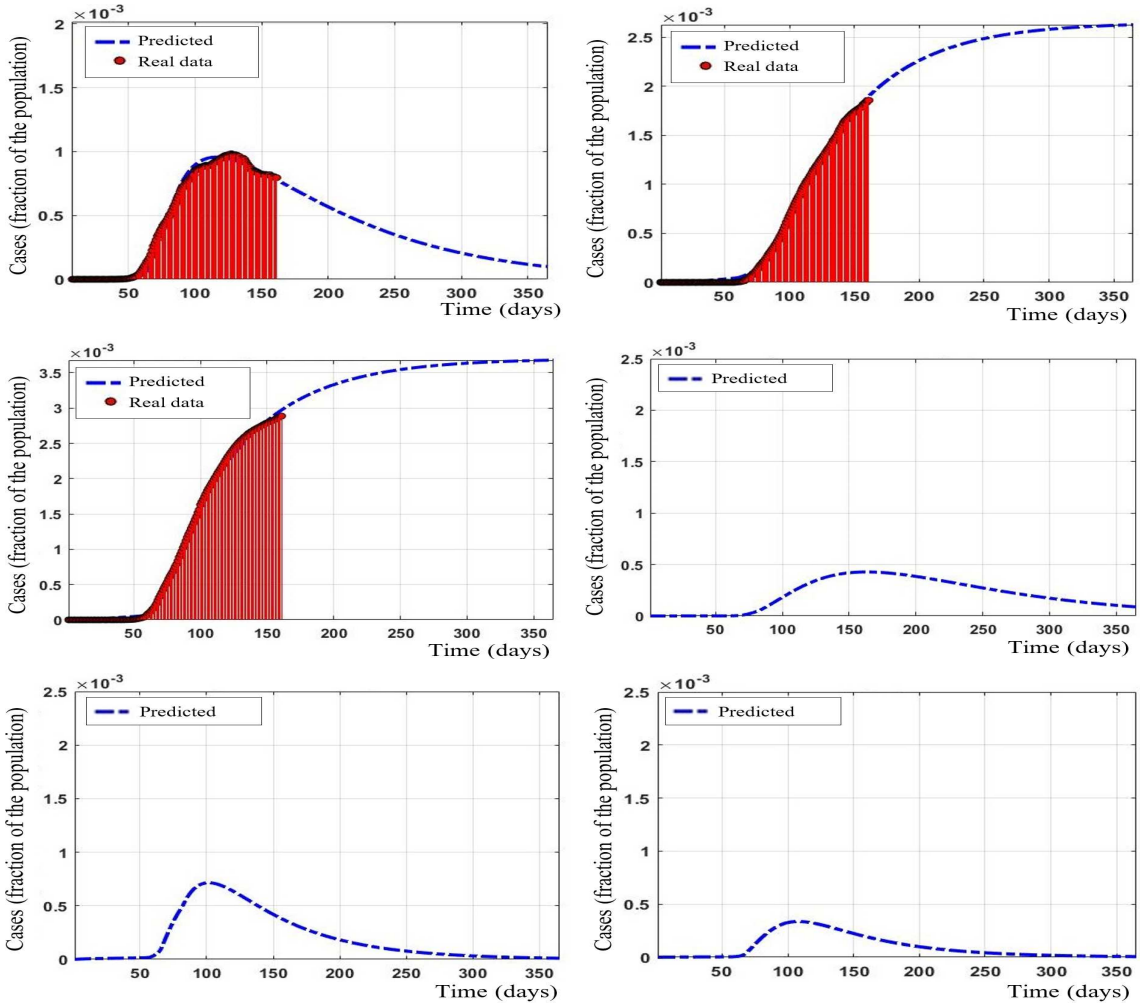

Figure 5.5: Prediction vs. data using RSFS.

Canada reported its first case on January 25, 2020. In the early days of the pandemic, the risk for Canadians was considered negligible by healthcare officials due to the low transmission rate. As our prediction model shows, the transmission rates between cases were low initially, starting January 25, 2020, to March 15, 2020. Our model indicates that the reproduction rate was $R_0 = 1.0812$, which is a low rate for the COVID-19 (Giordano et al., 2020). On March 5, 2020, Canada reported its first community transmission case (Slaughter, 2020). Following rapid increase in the number of detected cases, most of Canada's provinces stated a state of emergency. Quebec was the first province that declared a state of emergency on March 12, 2020. The province was considered as the epicenter of the outbreak in Canada, with the most COVID-19 cases. We consider the second stage from March 15, 2020, to March 24, 2020. The Quebec government called off schools and entertainment

centers on March 13 and 15, respectively (Gouvernement du Québec, nd). On March 16, The federal government of Canada restricted travel to Canada to only Canadian citizens, permanent residents, and U.S. citizens. On March 20, Canada and the United States temporarily closed their border. On March 20, the Quebec government forbid indoor gatherings and also applied mandatory measures for outdoor gatherings. On March 12, the Government of Ontario announced the closure of schools (Ontario.ca, nd). Also, On March 23, all "non-essential" businesses were ordered to close. The same measures were applied by other provinces such as British Columbia and Alberta. During the second stage, our model shows a reproduction rate of R_0 =2.3449. From March 24 to March 28, 2020, higher number of test kits arrived in Canada, leading to better testing and determination of infected people. As a result, the reproduction rate raised to $R_0 = 3.1555$. From March 28 to April 2, 2020, vigorous measures applied by the officials significantly reduced the reproduction rate resulting in $R_0 = 1.0699$. The reproduction rate kept decreasing due to the practical measures to $R_0 = 1.0168$ and $R_0 = 0.9182$ in the last two stages.

Figure 5.6 depicts details regarding the future trends in the next 204 days. In Figure 5.6, we have projected the anticipated number of different cases with different types of symptoms over the next few months. Based on these data, we can observe the upcoming peak in the number of cases, that will help the healthcare policymaker plan for future social measures such as lockdowns. In Figure 5.6, we have plotted the number of Cumulative Infected, Current Total Infected, Recovered, Deaths, Diagnosed Cumulative Infected, Diagnosed Current Total Infected, and Diagnosed Recovered cases. These data help the policymakers plan for resource allocation to avoid resource shortages in the healthcare system. For instance, using these data, the policymakers can estimate the number of COVID-19 tests that they will need to perform per day in the upcoming months.

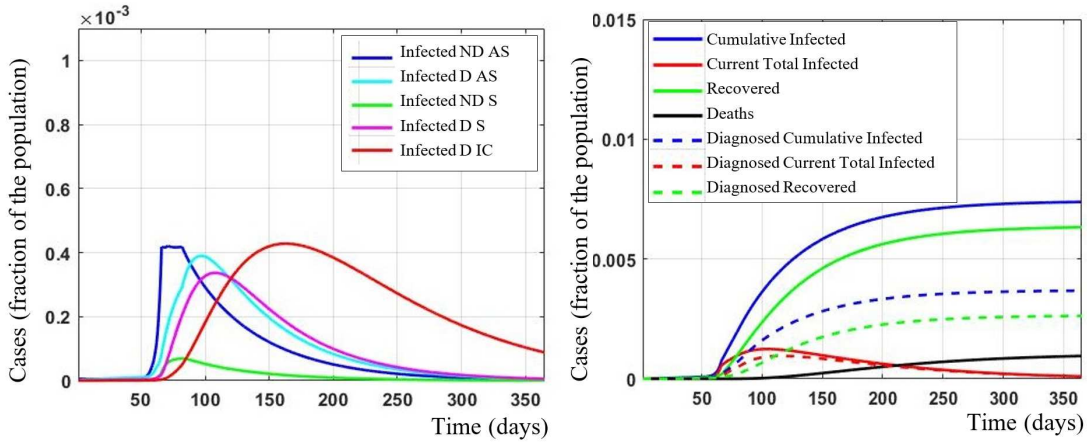

Figure 5.6: Results of RSFS for Canada. Diagnosed Symptomatic (D S), Non-Diagnosed Asymptomatic (ND AS), Non-Diagnosed Symptomatic (ND S), Diagnosed Asymptomatic (D AS), and Diagnosed with Life-Threatening Symptoms (D IC).

5.5.1 Curve Fitting and Estimating the values of Epidemiological Parameters in Quebec

Quebec has been dealing with a surge in daily new cases since January 2020. The province has reported a record high in daily new cases in April 2020 and announced that most hospitals are full. Since the transmission rates in Quebec are different from those of other provinces, we applied our methodology to model the pandemic in the province to project future scenarios. Figure 5.7 presents the output of the model for Quebec that is equivalent to Figure 5.5 we depicted before for Canada. To understand this figure, we refer the readers to the explanation of Figure 5.5. Figure 5.7 shows the prediction of the number of daily new cases, recovered cases, cumulative diagnosed cases, and the cases with severe symptoms in Quebec. The outcomes show that the restrictions applied by the government are effective in reducing community transmission and limit pandemic growth. Considering the results, we note that maintaining the current measures is vital to limit the spread of the virus.

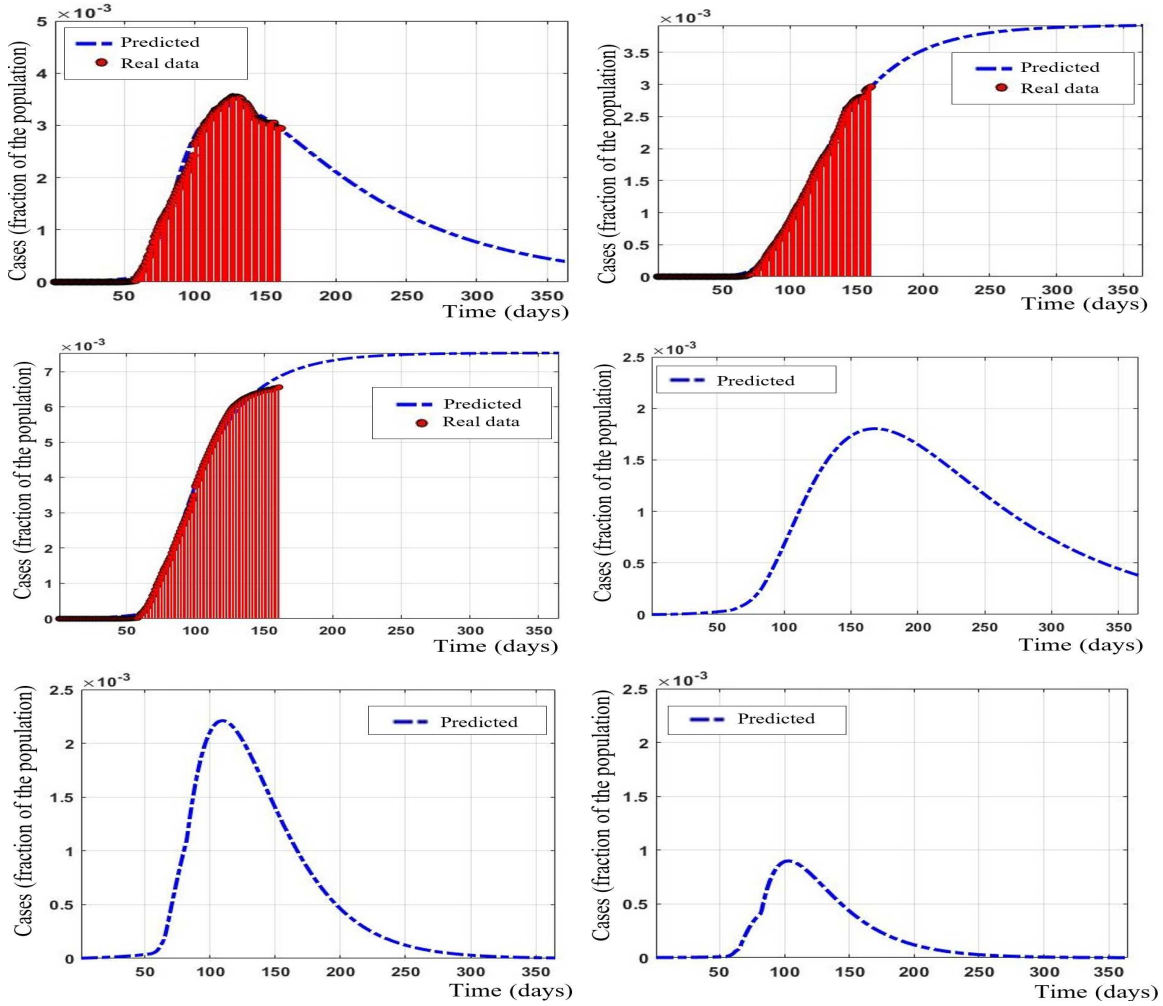

Figure 5.7: Results of the modeling and prediction of the COVID-19 pandemic in Quebec.

5.5.2 Curve fitting and estimating the values of epidemiological parameters in Ontario

Ontario is one of the main provinces of Canada that has been dealing with the ongoing viral pandemic caused by the SARS-Cov-2 virus. The province announced a state of emergency on March 17, 2020. During the first two months of the pandemic, the province faced a significant surge in daily new cases. The province considered several measures such as lockdowns and remote workings to limit the pandemic growth. Due to the high number of patients, hospitals faced a shortage of equipment for treating all patients. To be able to control the pandemic and limit the community transmission, we need to model and forecast the upcoming trends. Figure 5.8 presents the output

of the model for Ontario that is equivalent to Figure 5.5. To understand this figure, we refer the readers to the explanation of Figure 5.5. Based on the outcomes, we observe different transmission rates in the different stages of the pandemic. Our methodology approximated that the transmission rates were 0.8170, 8.3268, 10.9885, 2.1213, 1.0901, and 0.9166. from Stages 1 to Stage 6, respectively. It becomes apparent that the transmission rates increased significantly over the first stages of the pandemic. However, the measures applied by the government were effective and reduced the transmission rates and community transmission. We note that it is essential to keep the measures in place for the upcoming months since our model shows that the province will still have COVID-19 cases in the next 200 days, even in the best scenario.

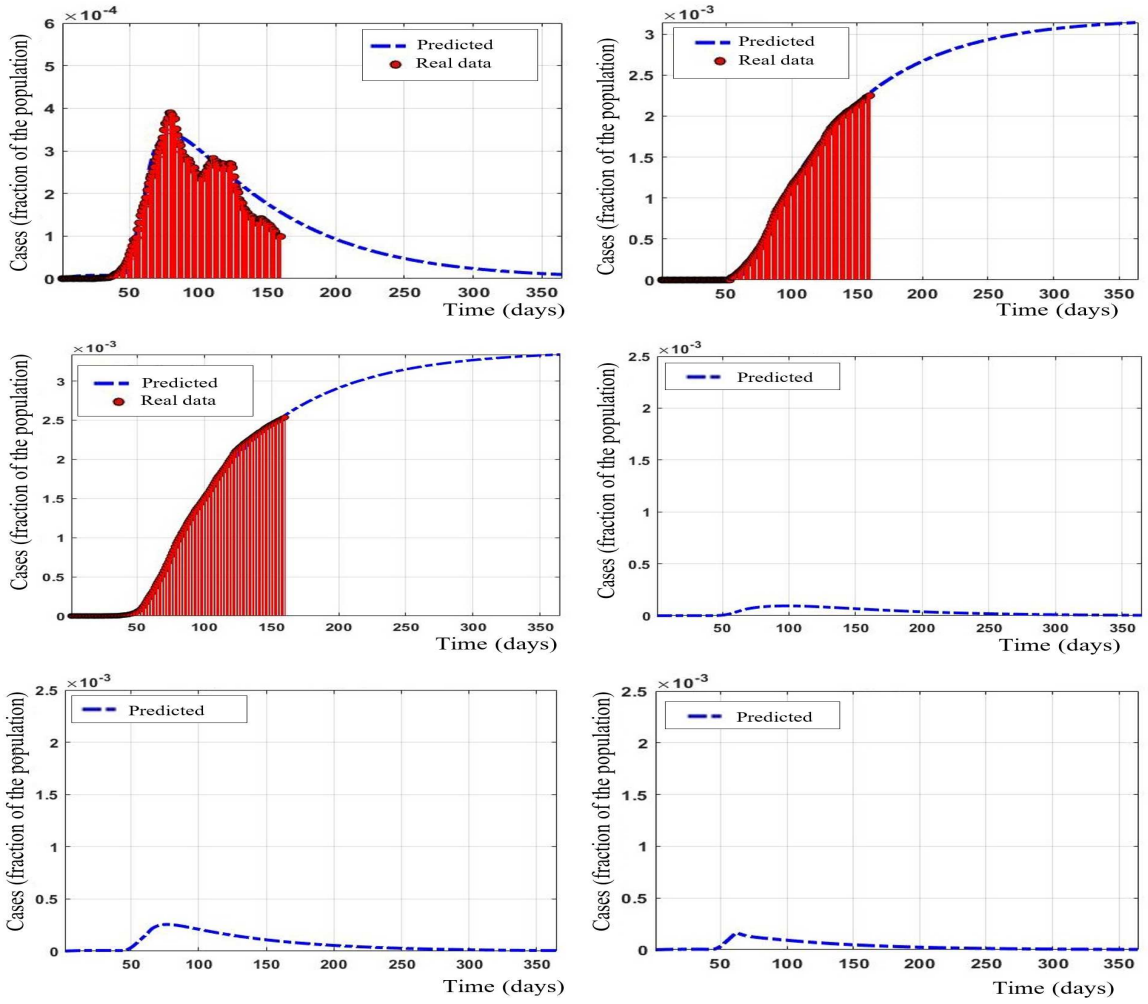

Figure 5.8: Results of the modeling and prediction of the COVID-19 pandemic in Ontario.

5.5.3 Managerial insights and sensitivity analyses

The prediction of future scenarios in the previous section is based on the presence of social distancing measures. Yet, some entertainment centers were allowed to reopen while maintaining social distancing. Therefore, it is essential to explore how restorations would influence future scenarios. This section investigates the effect of changes in transmission rates because of the reopening of businesses and changes in adaptation to behavioral recommendations. Therefore, we increased the values of the parameters α, β, γ, δ, and ε and investigate their effect on the number of different cases. Figures 5.9-5.12 show the results. Increasing α significantly rises the number of cases. In addition, increasing β, γ, and δ increases the number of cases as well.

By comparing Figure 5.9, we can realize that α is the most important parameter in the future number of cases. It makes sense because α represents the transmission rate from an infected case (that is not diagnosed) to a susceptible case. On the other side, in Figure 5.9, we can see that the future number of cases is not sensitive to changes in parameters β and δ because they represent the transmission rates from diagnosed/ailing cases to susceptible cases. This is reasonable because diagnosed cases must legally respect the quarantine conditions or they will be fined. Figure 5.9 shows that, by increasing the number of tests (ε), the number of infected cases increases. This is because those with mild symptoms will be diagnosed more and must respect the quarantine conditions. Also, in Figure 5.9, we observe that the number of cases does not increase as parameter γ increases. Here, γ represents the transmission rate from recognized cases (infected and detected cases) to susceptible cases. This behavior is reasonable because, as we can see in Figure 5.5, the number of recognized cases is significantly less than the number of cases with mild or no symptoms. Therefore, they will not contribute significantly to the community transmissions.

Figures 5.10-5.12 present a sensitivity analysis of the main parameters similar to the one we showed in Figure 5.9. In these three figures, we have performed the analysis for the infected, recovered, and death cases, respectively. Hence, to limit the transmission of the disease, we should limit community transmission by employing social distancing measures, while increasing the testing capacity. This research determined asymptomatic patients as the main drivers of the pandemic.

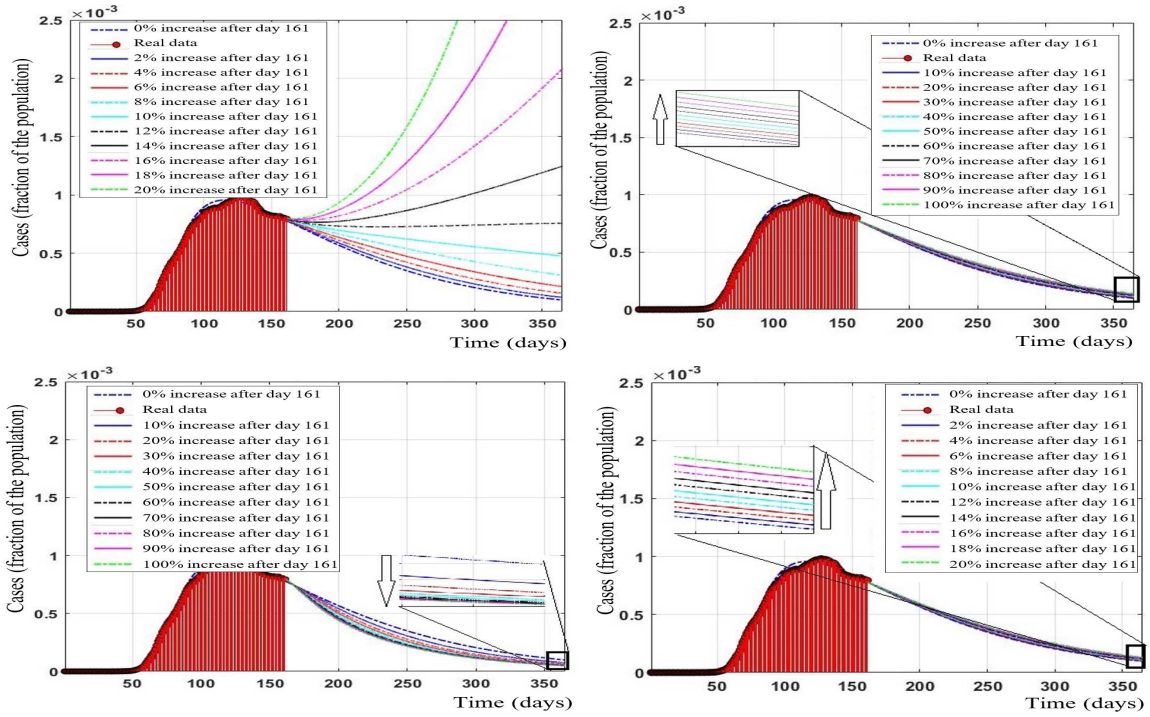

Figure 5.9: Sensitivity analyses for infected cases in Canada.

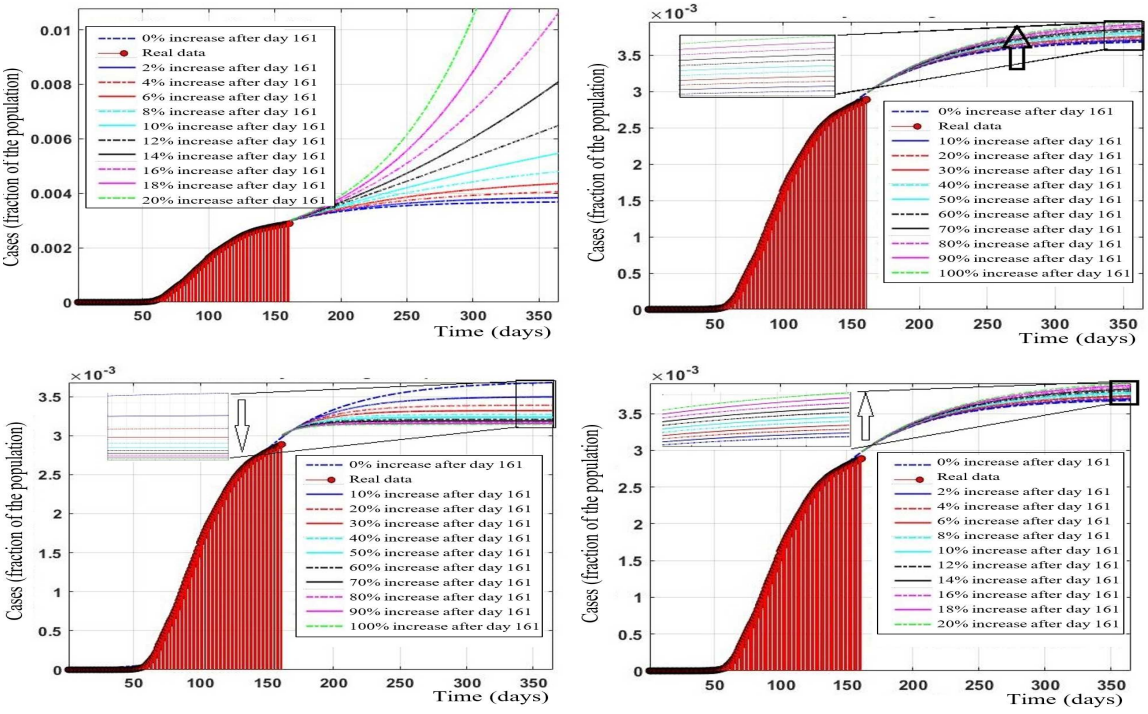

Figure 5.10: Sensitivity analyses for Cumulative Diagnosed Cases (CDC) in Canada.

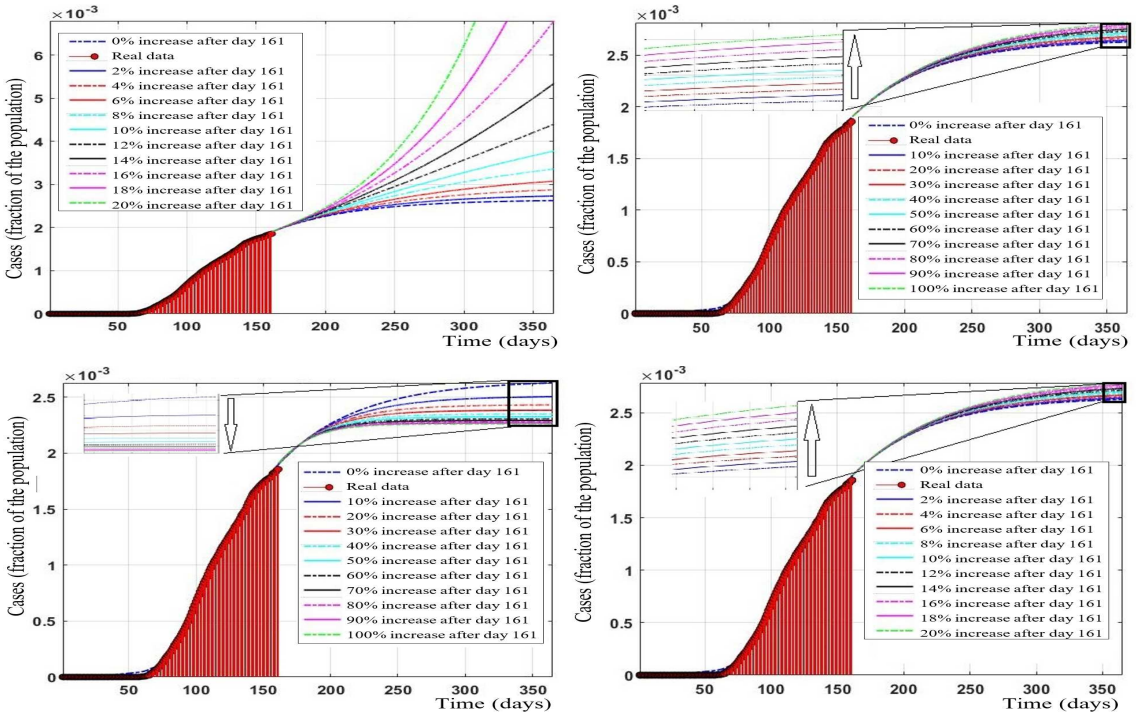

Figure 5.11: Sensitivity analyses for number of recovered cases in Canada.

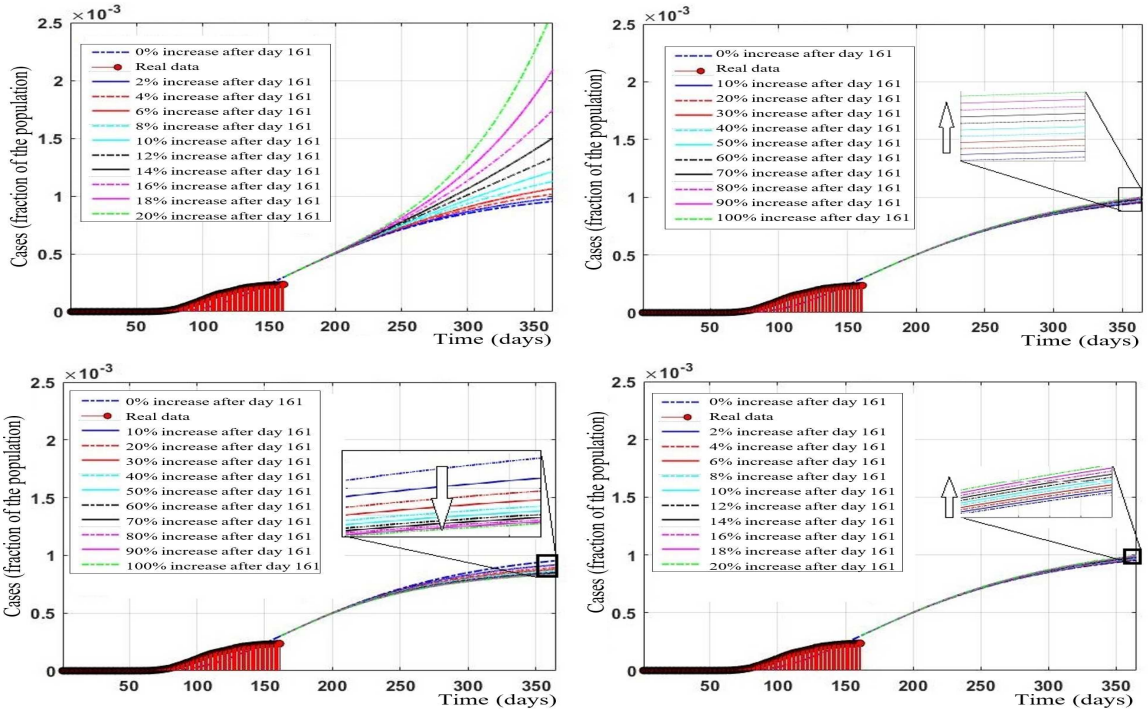

Figure 5.12: Sensitivity analyses for number of death cases in Canada.

5.6 Conclusion and outlook

COVID-19 is a serious disease that is spreading quickly. Hence, developing a methodology to predict the growth of the COVID-19 pandemic is essential. This research presented an efficient Robust Stochastic Fractal Search algorithm combined with a mathematical model to forecast the COVID-19 pandemic. To enhance the performance of our algorithm, we used a robust design of experiment approach for tuning. We used the proposed methodology to model the COVID-19 outbreak in Canada. Our results showed that our algorithm obtains a promising fit function to real data. We calculated epidemiological parameters for Canada and estimated the reproduction rate of the virus in different stages of the pandemic. We showed that the measures applied by the provincial and federal governments effectively reduced the community transmission leading to a sharp decrease in the number of daily new cases. In addition, our methodology estimated the number of symptomatic and non-symptomatic cases as well as the number of cases with mild symptoms and severe symptoms. These results help the policymakers to plan for resource allocation in the upcoming months and optimize the use of the resource in the healthcare system. Such planning and optimization will minimize the probability of equipment shortages in the hospitals and also the number of death cases due to limited equipment. Moreover, we performed sensitivity analyses to estimate the possible future scenarios. We changed the transmission rates to explore the effect of variations in these vital parameters on future trends. Our findings disclosed that the transmission rate from an asymptomatic infected case to a susceptible case is the most important factor. Any increase in this rate raises the number of infected cases. In addition, increasing the transmission rate from other cases to susceptible cases also raised the number of cases. Moreover, increasing testing capacity reduced the number of cases. Therefore, to limit the virus transmission and slow down the pandemic growth, we should apply effective social distancing measures, while increasing the diagnosis rate of asymptomatic cases. As future research directions, researchers can use the results of this study as an input to an optimization model to decide on the optimal allocation of limited resources such as testing kits. Finally, another direction for future studies is to consider the effects of vaccination on the growth of the pandemic in the proposed methodology and find a step-by-step plan for relaxing the social measures.

Chapter 6

Gradient-Based Grey Wolf Optimizer with Gaussian Walk: Application in Modelling and Prediction of the COVID-19 Pandemic[1]

Abstract

This research proposes a new type of Grey Wolf optimizer named Gradient-based Grey Wolf Optimizer (GGWO). Using gradient information, we accelerated the convergence of the algorithm that enables us to solve well-known complex benchmark functions optimally for the first time in this field. We also used the Gaussian walk and Lévy flight to improve the exploration and exploitation capabilities of the GGWO to avoid trapping in local optima. We apply the suggested method to several benchmark functions to show its efficiency. The outcomes reveal that our algorithm performs superior to most existing algorithms in the literature in most benchmarks. Moreover, we apply our algorithm for predicting the COVID-19 pandemic in the US. Since the prediction of the epidemic is a complicated task due to its stochastic nature, presenting efficient methods to solve the problem

[1]This paper is published in *Expert Systems with Applications* in 2021 and has been cited 79 times as of March 2024.

is vital. Since the healthcare system has a limited capacity, it is essential to predict the pandemic's future trend to avoid overload. Our results predict that the US will have almost 16 million cases by the end of November. The upcoming peak in the number of infected, ICU admitted cases would be mid-to-end November. In the end, we proposed several managerial insights that will help the policymakers have a clearer vision about the growth of COVID-19 and avoid equipment shortages in healthcare systems.

6.1 Introduction

Scientists employ optimization in almost every research field. Optimization is a significant challenge in science and engineering, mainly due to the complexity of problems on the one hand and the shortcomings of classical approaches, on the other hand. Random Search Algorithms (RSA) are one of the most efficient means of solving complex real-world problems (Zabinsky et al., 2009; Solis and Wets, 1981; Hong and Nelson, 2007). These algorithms sacrifice optimality to find a high-quality near-optimal solution in a short time. The main feature of these methods is randomness embedded in their framework during the iterations of the algorithm. RSAs are more flexible and easier to apply compared to traditional methods in terms of implementation complexity. Metaheuristics are one group of the main RSAs that have been widely used to resolve complex optimization problems. Some of the most recent metaheuristic algorithms are Grey Wolf Optimizer (GWO), Salp Swarm Algorithm (SSA), and Coronavirus Herd Immunity Optimizer (CHIO).

Healthcare science is one of the main fields in which optimization makes a remarkable improvement. In December 2019, a new virus named SARS-Cov-2 emerged in China that causes severe respiratory disease (COVID-19). The virus spread rapidly to more than 213 countries resulting in 22,185,755 cases and 780,369 deaths. Improvement in modeling the COVID-19 outbreak will significantly help the authorities in decision making. Besides, these insights enable us to optimally distribute resources and side-step equipment shortages in hospitals and save humans' lives. Prediction of the COVID-19 pandemic is challenging due to its stochastic nature and complexity.

Zhang et al. (2020) proposed a piecewise Poisson formulation to study the recent cases of the

COVID-19 pandemic. Using the suggested model, the researchers projected the peak of the epidemic. Chimmula and Zhang (2020) presented a deep learning-based method using Long-Short Term Memory (LSTM) to forecast the progress of the COVID-19 outbreak. The authors also aimed at estimating the possible ending point of the epidemic. The offered methodologies have several limitations that make their outcomes inapplicable. Scientists should provide enormous data to train the LSTM. Moreover, the suggested method cannot forecast either the number of cases with life-threatening symptoms or the number of asymptomatic cases. Furthermore, LSTM cannot estimate essential epidemiological statistics, including the reproduction rate. Arora et al. (2020) utilized a deep learning-based method using LSTM to forecast India's forthcoming COVID-19 cases. Their offered method has the same limitations as the method suggested by Chimmula and Zhang (2020), which makes their forecasts valid for a short period. Many researchers utilized LSTM and machine learning techniques to forecast the future pandemic scenarios in several countries; however, most of them have the same limitations (Abebe, 2020; Alamo et al., 2020; Garcia et al., 2020; Lalmuanawma et al., 2020; Panwar et al., 2020; Peng and Nagata, 2020)

One of the most recent models to define the pandemic is the SIDARTHE model. The model was first presented in a paper by (Giordano et al., 2020). The authors claimed that the formulation is able to project the future trend of the outbreak over a more extended period of time. Besides, the model provides the policymakers and healthcare professionals with vital epidemiological information such as reproduction rate. Although the model is very efficient in predicting future trends, the scientists highlighted that solving the model optimally is complicated due to its unique characteristics.

As mentioned earlier, solving complex optimization problems using metaheuristics is easier compared to classical methods. Grey Wolf Optimizer (GWO) is one of the most recent and efficient metaheuristic algorithms in the literature. GWO is inspired by the hunting behavior of Grey wolves in nature. The GWO performs acceptably in exploration by adapting the search radius of the wolves in the first iterations. It maintains a good diversity among the wolves to avoids local optima. However, we could improve the exploration ability of the GWO to enhance its ability to search the solution space more intelligently (Long et al., 2018). The lack of efficient exploration ability in GWO is apparent from its results in multimodal benchmark functions (Mirjalili et al., 2014). Besides, to enhance the algorithm to efficiently exploit the solution space, we should add

some new operators to the algorithm. The GWO performs average in the exploration of the solution space, considering its results in composite benchmarks in which other algorithms dominate GWO in most benchmarks (Mirjalili et al., 2014). Random movements such as Gaussian and Lévy walks in the exploration phase will remarkably increase the exploration ability of the algorithm.

In the exploitation phase, the GWO uses random movements on a tiny scale that do not necessarily guarantee an improvement in the best solution. Using gradient information that always guarantees improvement in the best solution will significantly improve the performance of GWO. GWO is applied successfully to many optimization problems in different fields such as text document clustering (Rashaideh et al., 2019), feature selection (Abdel-Basset et al., 2020), predicting the strength of concretes (Golafshani et al., 2020), biodiesel production (Samuel et al., 2020), multi-objective flexible job-shop scheduling problem (Zhu and Zhou, 2020), and three-dimensional path planning for UAVs (Dewangan et al., 2019). For more detailed information about applications of GWO, please see (Faris et al., 2018).

In this research, we present a new algorithm called Gradient-based Grey Wolf Optimizer (GGWO) that enables scientists to solve many real-world optimization problems. In our algorithm, we utilize the advantages of the gradient that presents valuable information about the solution space. In many optimization problems, gradient information is available or could be estimated. Using gradient information, we explore the solution space more intelligently by considering the gradient direction in our search process, leading us to the optimal or a good near-optimal solution. Almost all metaheuristic algorithms ignore the gradient information, which increases the probability of getting trapped in local optima. This motivated us to add the gradient in one of the most efficient algorithms to improve the exploration and exploitation abilities of the method. Considering gradient information, we accelerate the algorithm that enables us to solve well-known complex benchmark functions optimally for the first time in the field.

Besides, we use deep mathematical concepts such as Gaussian walk and Lévy flights to improve the search efficiency of our method. The proposed contributions enable the suggested algorithm to avoid local optima. Our computational results on several benchmarks demonstrate the superiority of our algorithm to other algorithms in the literature. Moreover, we apply several statistical tests to determine significant differences in the performance of the algorithm compared to state-of-the-art

methodologies. Moreover, we apply the devised algorithm to forecast the spread of the pandemic in the United States, with most cases of COVID-19. Our results predicted the maximum number of infected and hospitalized cases in the United States that will happen in mid-to-end November 2020. Besides, we perform further analysis to project future scenarios. We also measured the effect of the implemented restrictions by the government.

We have organized the remainder of this paper as follows: Section 6.2 provides a detailed literature review. Section 6.3 proposes a new methodology to solve optimization problems based on gradient information and random walks. In Section 6.4, we carry out computational experiments on challenging benchmarks using our algorithm. Section 6.5 presents an application of our methodology for forecasting the spread of the COVID-19 outbreak. In Section 6.6, we analyzed the uncertainty in the future spread of the pandemic. Section 6.7 concludes the paper, including an outlook on future research avenues.

6.2 Survey on the relevant literature

The underlying idea of most of the metaheuristic algorithms is to mimic a swarm behavior of nature. Mirjalili and Lewis (2016) divided metaheuristics into three categories: Swarm Algorithms (SAs), Evolutionary Algorithms (EAs), and Physics-based Algorithms (PAs). EAs, PAs, and SAs mimic the evolution process, law of physics on particles, and swarm behavior, respectively. The classification of metaheuristic algorithms includes a variety of strategies aimed at solving complex optimization problems. These strategies are broadly categorized into four main groups:

- **Evolutionary Algorithms:** These algorithms are inspired by the process of natural evolution. Examples include Genetic Programming (GP) (Koza, 1992), Biogeography-Based Optimizer (BBO) (Simon, 2008), Evolutionary Programming (EP) (Fogel, 1998), Genetic Algorithms (GA) (Holland, 1992), Evolution Strategy (ES) (Rechenberg, 1978), and Differential Evolution (DE) (Storn and Price, 1997).

- **Physics-based Algorithms:** Inspired by physical phenomena, this category includes Simulated Annealing (SA) (Kirkpatrick et al., 1983), Galaxy-based Search Algorithm (GBSA) (Kaveh and Talatahari, 2010), Central Force Optimization (CFO) (Formato, 2007), Curved

Space Optimization (CSO) (Moghaddam et al., 2012), and Gravitational Search Algorithm (GSA) (Rashedi et al., 2009).

- **Swarm-based Algorithms:** These algorithms model the behavior of social animals. Key examples are Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), Artificial Bee Colony (ABC) (Karaboga and Basturk, 2007), Ant Lion Optimizer (ALO) (Mirjalili, 2015), Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014), and Whale Optimization Algorithm (WOA) (Mirjalili and Lewis, 2016).

- **Other Population-based Algorithms:** This group encompasses various innovative strategies such as Stochastic Fractal Search (SFS) (Salimi, 2015), Sine Cosine Algorithm (SCA) (Mirjalili, 2016b), and Water Cycle Algorithm (WCA) (Eskandar et al., 2012).

Each category represents a unique approach to navigating complex search spaces, leveraging the principles of evolution, physical laws, collective animal behavior, or other innovative concepts to identify optimal or near-optimal solutions to a wide range of problems.

Based on the classification of Mirjalili and Lewis (2016), our algorithm is in the category of the swarm-based algorithms; however, this is not the only classification in the literature. For instance, based on Blum and Roli (2003), metaheuristics could be classified based on different perspectives such as nature-inspired vs. non-nature inspired, population-based vs. single point search, dynamic vs. static objective function, one vs. various neighborhood structures, and memory usage vs. memory-less methods. Based on the latter classification, our algorithm is in the class of population-based nature-inspired algorithms with the static objective function. The readers are referred to Blum and Roli (2003) for more details regarding the latter classifications. Grey Wolf Optimizer (GWO) is one of the most efficient algorithms in solving complex optimization problems. The GWO performs acceptably in exploration by modifying the distance between grey wolves in the first iterations. It maintains a proper distance and diversity between the wolves to avoid local optima. However, the exploration ability of the GWO could be significantly improved (Long et al., 2018). Using random movements like Gaussian and Lévy walks during the exploration phase will remarkably increase the exploration ability of the algorithm. However, GWO suffers from a lack of an efficient exploitation ability (Bansal and Singh, 2021; Long et al., 2018). In the exploitation phase,

the GWO uses random movements on a tiny scale that do not necessarily guarantee an improvement in the best solution. However, gradient information, which always guarantees improvement in the best solution, will significantly improve the performance of GWO. In this research, we enhanced the GWO by adding new operators to search the solution space using the gradient information for the first time. We called the algorithm Gradient-based Grey Wolf Optimizer (GGWO). The gradient provides valuable information about the solution space and enables the GGWO to achieve highly accurate results. Gradient information and new operators meaningfully enhanced the performance of the GGWO in exploiting the neighborhood of the best solution. Moreover, we apply a Gaussian walk and Lévy flight at the end of each iteration to enhance exploration. These features enable GGWO to avoid local optima while maintaining proper exploitation throughout the optimization process. We demonstrate the superiority of our methodology on some benchmarks using robust statistical tests. Furthermore, as an application, we use our proposed algorithm to forecast the spread of the COVID-19 pandemic in the US. Our results show that our algorithm could predict the future trends of the pandemic.

6.3 Designing an Accelerated Grey Wolf Optimizer

We will first illustrate the fundamentals of Grey Wolf Optimizer (GWO), then we will accelerate the GWO using gradient information, Gaussian and Lévy flights.

6.3.1 Grey Wolf Optimizer

GWO, recently proposed by Mirjalili et al. (2014), is inspired by grey wolves' hunting strategies in nature. Generally speaking, grey wolves are hierarchically categorized into four classes: Alpha, Beta, Delta, and Omega (Abdel-Basset et al., 2020). The Alpha is the dominant wolf in the pack. He/she makes all the decisions in the swarm. Other swarm members must comply with his/her decision. Besides, the only wolves that breed in the swarm are Alphas. Beta wolves' assist Alpha and communicate between Alpha wolves and other wolves. Beta wolf is the best nominee for being Alpha if one of the Alpha wolves dies or is too old to manage the swarm. The Beta fulfills the orders of the Alpha but also controls other wolves of the swarm. Omega wolves represent

the lowest-ranked grey wolves (Dhargupta et al., 2020). Omega wolves always follow other high-ranking wolves. Wolves that are not included in the Alpha, Beta, or Omega class are named Delta wolves. The Deltas manage the Omega wolves while assisting Alpha and Beta. Like many other swarm intelligence-based algorithms, GWO starts optimization by initializing a population. Then, after determining the dominant members, the wolves update their location in the solution space around the target. We apply Eqs. (86) to (86) to simulate the encircling process:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|, \tag{86}$$

$$\bar{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}, \tag{87}$$

In Eq. (86) to (86), t represents the iteration index and $\vec{A}$ and $\vec{C}$ characterize location vectors of target and other grey wolves. $\vec{X}_p(t)$ and $\vec{X}(t)$ are the position of the prey and grey wolf, respectively. These coefficients are calculated as follows:

$$\vec{A} = 2a_1 \cdot \vec{r}_1 - a_1, \tag{88}$$

$$\vec{C} = 2 \cdot \vec{r}_2, \tag{89}$$

In Eq. (88)-(89), a_1 decreases over iterations from 2 to 0 and $\vec{r}_1$ and $\vec{r}_2$ are random vectors. After encircling the prey, the wolves start the hunting process. To mathematically express the movements of grey wolves in the hunting process, we consider that the Alpha, Beta, and Delta have superior knowledge of the probable position of the target (possible optimal solution of the problem). In this framework, the following formulas are recommended to mimic the hunting process:

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \qquad \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \qquad \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|, \tag{90}$$

$$\bar{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \qquad \bar{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \qquad \bar{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta, \tag{91}$$

$$\bar{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \tag{92}$$

The GWO performs the above actions repeatedly to find a near-optimal solution for the problem

until a stopping criterion is met.

6.3.2 Accelerated Gradient-based Grey Wolf Optimizer

To perform fine in terms of exploration, an algorithm should maintain an appropriate balance between exploration and exploitation. GWO searches the solution space by updating the position of the dominated wolves regarding the position of Alpha, Beta, and Delta. By reducing the parameter a over iterations, GWO aims at exploration in the first iterations and then focuses on exploiting in the last iterations. Besides, adjusting this parameter helps the GWO avoid trapping in local optima. This paper adds two novel features to GWO to enhance its performance and propose a novel algorithm called Gradient-Based Grey Wolf Optimizer (GGWO). First, we propose a new procedure to use gradient information to improve the algorithm's exploitation and exploration abilities. In many optimization problems, the gradient will provide valuable information about the shape of the solution space by determining the steepest slope at each point in the solution space. We move particles to the nearest local optima using gradient information while maintaining a proper exploration ability. Such updating operators enable GWO to search the solution space more efficiently and enhance the exploration ability of the algorithm to side-step local optima. We propose the following new updating formulations for Omega wolves: For $i = 1, \ldots, m$ and $w = 1, \ldots, n$, the position of the w-th wolf in the i-th dimension at iteration $t + 1$ is updated as:

$$X_W^i(t+1) = \begin{cases} \text{Eqns. (90) to (92)} & \text{if } \gamma \frac{\partial f}{\partial X^i}^{(\text{Min})} \leq \frac{\partial f}{\partial X_W^i}(t) < \gamma \frac{\partial f}{\partial X^i}^{(\text{Max})} \\ X_W^i(t) - \text{rand}(0,1)\lambda^i(t) \left(\frac{\partial f}{\partial X_W^i}(t) \right), & \text{otherwise} \end{cases} \tag{93}$$

where i is the index of decision variables in the optimization problem, and n is the number of grey wolves. The terms $\frac{\partial f}{\partial X^i}^{(\text{Max})}$ and $\frac{\partial f}{\partial X^i}^{(\text{Min})}$ show the largest positive and the smallest negative slopes for each dimension at each iteration of the algorithm. Whereas γ is a continuous parameter determined in $(0, 1]$. In the above formulation, we update λ^i using equation (94) as follows:

$$\lambda^i(t) = 0.1 \frac{(Ub^i - Lb^i)}{\max\left(|\frac{\partial f}{\partial X_k^i}^{(\text{Min})}|, |\frac{\partial f}{\partial X_k^i}^{(\text{Min})}| \right)}. \tag{94}$$

Based on the given illustrations in Pahnehkolaei et al. (2017), it is apparent that:

$$|\lambda^i(t)\left(\frac{\partial f}{\partial X_w^i}^{(t)}\right)| \geq 10(Ub^i - Lb^i). \tag{95}$$

In some optimization problems, the gradient of the problem may be unknown due to the non-differentiability of the objective function or discrete characteristics of the decision variables. In order to handle those problems, we present the following equation:

$$\frac{\partial f}{\partial X} = \frac{f(t) - f(t-1)}{X(t) - X(t-1)}. \tag{96}$$

The second contribution that we have added to GWO is the use of Gaussian walk and Lévy flight. These two are random walks to increase randomness in the GGWO and boost its exploration ability. Lévy flight and Gaussian walks create self-similar clusters (trajectories) but differ significantly in structure. The cluster created by the lévy flight contains several islands (sets of short steps) connected by long excursions (Chakrabarti et al., 2006). However, the Gaussian walk creates a denser and smaller cluster (within the same number of iterations) that consists of many small steps (Mousavirad and Ebrahimpour-Komleh, 2017). Random selection of these two methods enhances the exploration capability of the GGWO by helping the algorithm avoid local optima. Therefore, GGWO switches randomly between Lévy flight and Gaussian walks to use the advantage of both (Salimi, 2015). In the proposed GGWO, we use the following formulations to update the position of Omega wolves in the solution space at the end of each iteration. For $i = 1, \ldots, m$ and $w = 1, \ldots, n$, the new position of the wolf is given by:

$$X^i_{W,\text{new}} = X^i_W + K \times \text{Gaussian}(|\theta_i|, \sigma) - (\xi \times \theta_i - \xi' \times X^i_W) \tag{97}$$

$$X^i_{W,\text{new}} = X^i_W + X^i_W \times \text{Levy}(\eta) \tag{98}$$

where θ_i and $|\sigma|$ represent the best solution and standard deviation of the Gaussian distribution, respectively. GGWO changes the Gaussian parameter as $\sigma = |K \times (x_i - \text{BP})|$ and reduces the length of steps over iterations by setting $K = (\log(l))/l$, where l is the iteration number. The

expression $X^i_{W,\text{new}}$ is the new position of the wolf, and X^i_W is its current position. Besides, ξ' and ξ are random numbers in $(0, 1]$. The Lévy flight is computed by Eq. (99).

$$\text{Levy}(x) = \frac{0.01 \times \sigma \times r_1}{|r_2|^{(1/\beta)}} \tag{99}$$

where r_1 and r_2 are random numbers in $(0, 1]$. β is a constant equal to 1.5. In Eqn. (99), we compute σ by:

$$\sigma = \left(\frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\beta 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \tag{100}$$

Based on the given illustrations, the main framework of the GGWO is the same as GWO; however, with some significant changes. For instance, first, instead of the classical GWO operators, the GGWO uses a combination of the original operators and gradient-based operators to update the position of the wolves. Besides, we need to add a new feature to the GGWO (a function) to calculate the gradient of the objective function at each point of the solution space. Moreover, we use Gaussian walk and levy flight to increase randomness at the end of each iteration, which significantly improved the exploration and exploitation by using both long and short steps to move the particles in the solution space. The pseudo-code of the GGWO is presented as follows:

6.4 Results and Discussion

In order to evaluate the efficiency of the offered GGWO, we compare it with well-known algorithms in the literature, including Grey Wolf Optimizer (GWO), Gradient-based Water Cycle Algorithm (GWCA), Artificial Bee Colony (ABC), Gravitational Search Algorithm (GSA), hybrid Particle Swarm Optimization Gravitational Search Algorithm (PSOGSA), Particle Swarm Optimization (PSO), Salp Swarm Algorithm (SSA), Sine-Cosine Algorithm (SCA), and Moth-Flame Optimization (MFO). Table 6.1 provides the values of the parameters of the algorithms. We implement the experiments considering two different dimensions, 30 and 50, to enhance the benchmark functions' complexity. In all the tests, we consider a maximum NFEs of 30,000 as the stopping condition. We also set the control parameter γ to 0.9 in all tests. We repeat the solution process by each algorithm

Algorithm 4 Grey Wolf Optimizer (GGWO)

Input the parameters of GGWO
for $i = 1$ to npop **do**
  Create a random solution
  Calculate the fitness
end for
Sort the solutions based on the fitness values
Set the best three solutions as Alpha, Beta, and Delta, respectively.
Set the remaining wolves as Omegas
$it = 1$
while stopping criterion is not met **do**
  for $i = 1$ to npop **do**
    Calculate and update A and C
    Calculate the value of $\frac{\partial f}{\partial X_W^i}(t)$
    Update the position of wolves using Eq. (8)
  end for
  Calculate the fitness values of all wolves
  Update the Alpha, Beta, and Delta
  for $i = 1$ to number of Omegas **do**
    Update the position of Omega wolves using Eqs. (12) to (13)
  end for
  Decrease $\vec{a}$
  $it = it + 1$;
end while=0

50 times to enhance the accuracy of the results. Besides, we report the average, standard deviation, best, and worst values of the objective function for each test problem and each algorithm. Table 6.2 presents the used benchmarks. These benchmarks are known as complex benchmarks in the literature (Lozano et al., 2011; Liao et al., 2015).

Table 6.1: Main parameters of the algorithms.

Algorithm	**Parameter**	**Value**	**Algorithm**	**Parameter**	**Value**
GWCA	parameter γ	0.9	**PSO**	parameter c_1	2
	dmax	0.001		parameter c_2	2
	Nsr	4		Inertial weight	Linearly decreases from 0.6 to 0.3
ABC	# of onlookers	0.5*pop	**GSA**	Rnorm	2
	# of employed bees	0.5*pop		Rpower	1
	# of scouts	1		Alpha and G_0	20 and 100
GGWO	parameter γ	0.9	**MFO**	parameter a	Linearly decreases from -1
	parameter a	Linearly decreases from 2 to 0		parameter b	1
GWO	parameter a	Linearly decreases from 2 to 0	**SCA**	parameter a	Linearly decreases from 2 to 0
PSOGSA	parameter c_1	0.5	**SSA**	parameter c_1)	No parameter
	parameter c_2	1.5			

Table 6.2: Benchmark functions.

Function	Formulation	Range	D
Ackley	$f_1(x) = -20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N}x_i^2}\right) - \exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right) + 20 + e$	$[-32,32]^N$	30,50
Rastrigin	$f_2(x) = \sum_{i=1}^{N} x_i^2 - 10\cos(2\pi x_i) + 10$	$[-5.12,5.12]^N$	30,50
Sphere	$f_3(x) = \sum_{i=1}^{N} x_i^2$	$[-100,100]^N$	30,50
Griewank	$f_4(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600,600]^N$	30,50
High Conditioned Elliptic	$f_5(x) = \sum_{i=1}^{N}\left(10^6\right)^{\frac{i-1}{D-1}} x_i^2$	$[-10,10]^N$	30,50
Rosenbrock	$f_6(x) = \sum_{i=1}^{N-1} 100(x_{i+1} - x_i)^2 + (x_i - 1)^2$	$[-30,30]^N$	30,50
Shifted Ackley	$f_7(x) = -20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N}z_i^2}\right) - \exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi z_i)\right) + 20 + e$	$[-32,32]^N$	30,50
Shifted Rastrigin	$f_8(x) = \sum_{i=1}^{N} z_i^2 - 10\cos(2\pi z_i) + 10$	$[-5.12,5.12]^N$	30,50
Shifted Sphere	$f_9(x) = \sum_{i=1}^{N} z_i^2$	$[-100,100]^N$	30,50
Shifted Griewank	$f_{10}(x) = \frac{1}{4000}\sum_{i=1}^{N} z_i^2 - \prod_{i=1}^{N}\cos(\frac{z_i}{\sqrt{i}}) + 1$	$[-600,600]^N$	30,50
Shifted High Conditioned Elliptic	$f_{11}(x) = \sum_{i=1}^{N}\left(10^6\right)^{\frac{i-1}{D-1}} z_i^2$	$[-10,10]^N$	30,50
Shifted Rosenbrock	$f_{12}(x) = \sum_{i=1}^{N-1}[100(z_{i+1} - z_i)^2(z_i - 1)^2]$	$[-30,30]^N$	30,50

In the first four benchmark functions (F1, F2, F3, and F4) in dimension 30, the outcomes in Tables 6.3 and 6.4 show that our proposed method, GGWO performs significantly better than all the other algorithms in D30. The results in Tables 6.5 and 6.6 show the superiority of GGWO over

other algorithms in these benchmarks in D50 as well. Our proposed algorithm provides considerably better solutions in F1-F4 than any other algorithm due to its advanced operators to maximize exploration and exploitation abilities. Figure 6.1 shows that the GGWO avoids trapping in local optima and rapidly reaches the optimal solution for the problems. Besides, GGWO offers significantly lower average, best, worst, and standard deviation of objective function value for these benchmarks compared to other algorithms.

Figures 6.2, 6.3, 6.4, and 6.5 show the boxplots of the results in which GGWO presents significantly lower and narrower charts. The offered GGWO performs very well in F5, F6, and F7 benchmarks in dimension 30 considering average, best, worst, and standard deviation of results compared to other methods based on Tabls 6.3 and 6.4. In addition, in dimension 50, GGWO achieves the third optimum (lowest) results, as shown in Tables 6.5 and 6.6. Moreover, in F5 from Figure 6.1, GGWO's convergence curve shows its exploration and exploitation capabilities and efficiency in avoiding local optima.

The results in Tables 6.3 and 6.4 show that our designed algorithm performs meaningfully better than all the other methods in F8 and F9 for dimension 30. This is because GGWO has a significantly lower average compared to the other algorithms. Besides, based on the best, worst, and standard deviation of the objective function, we could conclude that the GGWO is the best solution approach for this benchmark. In addition, Tables 6.5 and 6.6 shows that for F8 and F9 in dimension 50, GGWO has significantly lower results considering average, best, worst, and standard deviation of the objective function. Therefore, GGWO is a reliable and robust algorithm since it has consistent performance and could find a promising solution in all repetitions. Besides, in F8 and F9, GGWO makes a perfect trade-off amid both exploration and exploitation based on the data provided in Figure 6.1. Although the other algorithms got trapped in local optima, GGWO could achieve global optima quicker without trapping in local optima.

Considering F10 and F12 in dimension 30, the GGWO performs the best. Besides, for dimension 50, GGWO performs well in terms of average, best, worst, and standard deviation of the objective function. In F11 for dimension 30, GGWO performs outstanding comparing to all the other methods. It has significantly lower average, standard deviation, best, and worst values than the other algorithms based on results in Tables 6.3 and 6.4. Besides, in F11 for dimension 50,

GGWO's performance is much promising than other algorithms in terms of average, best, worst, and standard deviation from Tables 6.5 and 6.6. Furthermore, based on Figure 6.1, in F11, GGWO ensures the right balance amid exploration and exploitation. In contrast to the other algorithms that get trapped in local optima, GGWO reaches the global optima.

To draw a reliable conclusion and demonstrate the superiority of the offered algorithm, statistical tests are conducted in this section. For this purpose, we apply Tukey's multiple comparison tests to discover significant differences in the performance of the algorithms. Figures 6.2, 6.3, 6.4, and 6.5 show the results of Tukey's multiple comparison tests schematically. Based on Figures 6.2 and 6.3 for dimension 30 and 50, the results of comparing the boxplot of the GGWO to other algorithms for the first four benchmarks (F1, F2, F3, and F4) show that the boxplot of the GGWO is significantly lower and thinner than all the other algorithm. In F5, F6, F7, F8, F10, and F12, the boxplots of the GGWO are lower than most of the algorithms, especially in F9 and F11, the box plot of the GGWO is a line at zero. This is because GGWO obtained the global optima of the benchmarks in all repetitions. These results show that the proposed algorithm not only performs remarkably but also performs significantly robust and reliable.

Table 6.3: Results of the simulations in 30 dimensions.

Benchmark	Statistics	Algorithms									
		GGWO	GWO	GWCA	ABC	GSA	PSOGSA	PSO	SSA	SCA	MFO
F1	Average	0	8.64E-15	1.07E-15	7.78298	6.23E-09	11.6611	6.07E-07	1.520216	14.29482	14.15397
	StdDev	0	2.75E-15	5.84E-15	1.02873	1.25E-09	8.345575	2.64E-06	0.923497	8.284824	8.396961
	Best	0.00E+00	7.11E-15	0	4.727577	4.85E-09	2.11E-10	2.79E-11	1.76E-05	3.08E-08	8.71E-08
	Worst	0.00E+00	1.42E-14	3.20E-14	9.587644	1.14E-08	19.38025	1.41E-05	3.222505	2.02E+01	2.00E+01
F2	Average	0	0.253528	0	70.24234	24.24192	131.5994	39.599863	58.238599	3.6824603	149.93231
	Std Dev	0	9.75E-01	0	10.32227	7.713473	41.05358	6.66E+00	1.64E+01	8.55E+00	3.21E+01
	Best	0	0	0	41.16908	13.9294	61.68735	27.85883	2.79E+01	1.01E-11	98.57154
	Worst	0	4.34E+00	0	92.06869	44.773	209.9356	5.37E+01	8.95E+01	2.83E+01	2.27E+02
F3	Average	0.00E+00	2.28E-62	2.36E-29	1.51191	8.67E-09	2.666667	5.87E-08	1.79E-05	1.59E-05	2.666667
	Std Dev	0.00E+00	3.68E-62	1.29E-28	6.15E-01	1.79E-09	6.914918	2.50E-07	1.72E-06	2.70E-05	6.914918
	Best	0	2.71E-64	1.23E-45	0.593387	5.80E-09	2.75E-10	2.02E-11	1.39E-05	6.81E-08	2.98E-08
	Worst	0	1.59E-61	7.04E-28	2.68743	1.31E-08	20	1.38E-06	2.11E-05	0.000108	20
F4	Average	0	0.002502	0	1.461505	0.082105	33.15244	0.011078	0.008941	0.08646	21.09083
	StdDev	0	0.0084466	0	0.268137	0.190495	50.22736	0.010589	0.0094284	0.2018043	45.48769
	Best	0	0	0	1.076298	0	0	0	1.52E-08	3.02E-12	4.47E-14
	Worst	0	0.044127	0	2.169038	1.025695	180.4868	0.049282	0.039202	0.850527	180.2163
F5	Average	1.83E-44	7.46E-121	9.45E-66	7706.8	241.9223	41346.57	1.44E-11	23856.1263	8.57E-06	542387.86
	Std Dev	7.04E-44	1.67E-120	5.18E-65	5264.083	159.92	124323	7.60E-11	11888.77	4.68E-05	542451.75
	Best	2.09E-48	8.71E-125	1.56E-154	804.9848	40.904	3.21E-06	1.26E-18	2781.57	6.83E-18	70214.2
	Worst	3.81E-43	7.99E-120	2.84E-64	20380.17	719.373	529831.7	4.17E-10	48046.444	0.000291	2706358.3
F6	Average	34.85498	26.43985	1.62E-24	5105.301	36.1634	3151.067	50.051	333.7611	28.36875	2680335
	Std Dev	27.24732	0.62575	7.46E-24	2839.1764	40.7528	16416.77	27.905481	604.0928	1.409943	14592674.1
	Best	23.46658	25.09341	0	974.2829	24.0738	14.20298	15.03268	23.15309	27.11928	1.189283
	Worst	147.9126	27.93068	4.08E-23	12450.14	233.399	90023.83	85.36001	2309.699	33.39472	7994325

Table 6.4: Results of the simulations in 30 dimensions.

Benchmark	Statistics	Algorithms									
		GGWO	GWO	GWCA	ABC	GSA	PSOGSA	PSO	SSA	SCA	MFO
F7	Average	1.942059	1.77E+00	3.13E-02	6.87411	6.38E-09	1.27E+01	1.37E-08	2.20E+00	3.8256	8.24452
	Std Dev	0.435974	4.09E-01	1.71E-01	1.10026	1.01E-09	7.72E+00	3.46E-08	9.17E-01	0.263284	7.647263
	Best	4.16E-06	1.09E+00	1.56E-07	4.57E+00	4.52E-09	2.09E-10	3.60E-11	2.07E-05	3.18E+00	5.38E-08
	Worst	2.50E+00	2.73E+00	9.37E-01	9.122293	9.15E-09	19.4627	1.88E-07	4.298275	4.24E+00	1.94E+01
F8	Average	2.653285	14.50399	6.007273	70.35375	27.22869	130.1343	41.32392	62.81495	80.68202	147.561
	Std Dev	3.308464	6.75E+00	7.630498	9.115818	5.514884	3.17E+01	1.03E+01	1.91E+01	2.17E+01	3.14E+01
	Best	0	4.87896	0	42.18886	1.69E+01	7.36E+01	2.19E+01	2.89E+01	4.61E+01	55.71759
	Worst	12.9344	3.03E+01	25.86894	81.53446	41.78826	197.9958	5.97E+01	1.05E+02	1.31E+02	1.93E+02
F9	Average	0.00E+00	7.36E-01	1.72E-15	1.337828	8.39E-09	1.36E+00	1.23E-08	1.72E-05	2.18E+00	2.02E+00
	Std Dev	0.00E+00	2.67E-01	2.63E-15	4.39E-01	1.28E-09	5.177062	4.69E-08	2.10E-06	1.95E-01	6.17E+00
	Best	0	2.99E-01	5.55E-17	0.441513	5.83E-09	2.82E-10	2.48E-11	1.23E-05	1.81E+00	1.72E-08
	Worst	0	1.45E+00	1.40E-14	2.456872	1.14E-08	20.712	2.58E-07	2.14E-05	2.71688	20.77365
F10	Average	0.015744	1.142719	0.064927	1.549616	0.078723	32.66406	0.006484	0.013608	2.048136	26.3487
	Std Dev	0.014757	0.176141	0.183418	0.33096	0.103276	43.75126	0.008487	0.016441	0.200311	46.67469
	Best	1.56E-11	0.748066	9.21E-11	1.089729	0	0	0	1.92E-08	1.71E+00	8.99E-15
	Worst	0.04916	1.52756	1.0161	2.3278	0.505	99.19931	0.03196	0.07847	2.5636	171.0315
F11	Average	0.00E+00	1.55E+03	7.67E-27	6974.676	243.352	70235.01	4.67E-11	27888.19	1.13E+04	526409.4
	Std Dev	0.00E+00	1.14E+03	3.38E-26	5005.006	164.8371	198201.4	1.93E-10	15809.66	4.92E+03	788901.1
	Best	0.00E+00	1.70E+02	0.00E+00	523.054	27.1468	1.07E-06	1.10E-19	5121.041	4.36E+03	7121.163
	Worst	0.00E+00	5.52E+03	1.86E-25	23425.73	658.5043	920733.9	1.02E-09	62981.94	25798.35	3677252
F12	Average	32.40043	253.9273	4.31E+01	4974.308	35.59113	2251077	44.5655	118.949	1579.494	24584.68
	Std Dev	17.36383	242.061	2.52E+01	3151.212	46.26933	1229469	30.6383	260.1012	729.624	40160.65
	Best	17.90934	36.21686	27.43943	1626.709	20.27133	17.17134	7.104455	24.5781	619.937	6.028516
	Worst	84.31829	1292.285	1.10E+02	13565.01	264.3985	6734709	103.615	1404.456	4580.251	97478.38

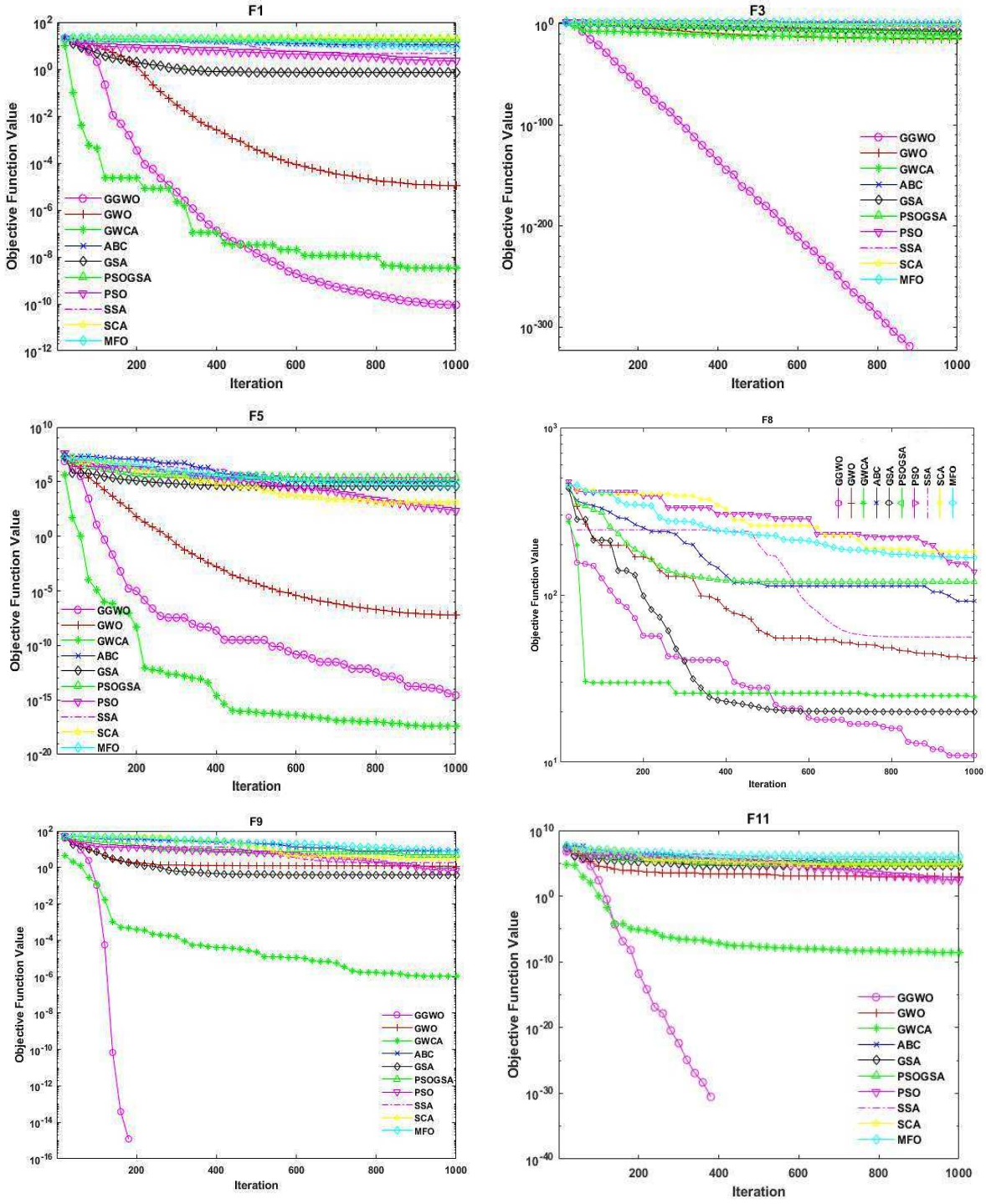

Figure 6.1: Convergence plot of the algorithms in dimension 30.

Tables 6.7, 6.8, and 6.9 present the outcomes of Tukey's multiple comparison tests for objective function values of the benchmarks in dimension 30 and 50. Based on the results, in F1 and F2 for

dimension 30 and 50, all the tests show p-values less than 0.05, except for the second row. This indicates that there are significant differences between the performances of the compared algorithms. Therefore, our proposed algorithm performs significantly better than all other algorithms (GWO, ABC, GSA, PSOGSA, PSO, SSA, SCA, and MFO) in terms of objective function value at 95% confidence level except for GWCA. However, based on the average, best, worst, and standard deviation values, we observe that the GGWO performs much better than GWCA in F1 and F2 (Tables 6.3 and 6.4). Our proposed method accomplishes outstanding results in F3 and F5 for dimensions 30 and 50 compared to other solution methods.

In F4 for dimension 30, GGWO performs statistically better than the other algorithms. Likewise, in the same benchmark for dimension 50, GGWO achieves better results than other methods. Based on the average, best, worst, and standard deviation values in this benchmark, we determine that the GGWO performs much better than GWO and GWCA in F4 (in Tables 6.3 and 6.4). In F6 and F7 for dimension 30, GGWO performs significantly better than all the other algorithms. For dimension 50, GGWO performs better than all the other algorithms except SSA. However, based on the average, best, worst, and standard deviation values, the GGWO outperforms SSA in solving F7 (in Tables 6.3 and 6.4). In F8 for dimension 30 and 50, GGWO outperforms most of the other algorithms. Besides, considering the average, standard deviation, best, and worst cases, GGWO beats GWCA (in Tables 6.3 and 6.4). In F9, F10, F11 and F12, in dimensions 30 and 50, GGWO outperforms other state-of-the-art algorithms.

In this section, we perform more in-depth statistical tests, such as Friedman's test, to make a consistent conclusion. Friedman's test discovers extensive differences among algorithms at a 95% confidence level. It is one of the most famous and widely used statistical tests to compare algorithms in the literature. Tables 6.10 and 6.11 show the Friedman tests' scores for each algorithm considering algorithms' performance in all dimensions. In Friedman's test, the lower the score, the more effective the method is. In Tables 6.12 and 6.13, we assigned a rank for each algorithm in each benchmark function based on the scores obtained in Tables 6.10 and 6.11. Results of Tables 6.12 and 6.13 disclosed that for both dimensions 30 and 50, the proposed algorithm ranked first in most of the benchmark functions, including F1, F2, F3, F4, F8, F9, and F11. Considering F5, F6, and F10, GGWO ranked third. In F7, GGWO performs better than five algorithms for both dimensions

Table 6.5: Results of the simulations in 50 dimensions.

Benchmark	Statistics	Algorithms									
		GGWO	GWO	GWCA	ABC	GSA	PSOGSA	PSO	SSA	SCA	MFO
F1	Average	1.18E-16	1.43E-14	4.74E-16	1.10E+01	3.80E-09	1.58E+11	9.81E-02	2.41E+00	1.56E+01	18.86403
	Std Dev	6.49E-16	2.18E-15	1.23E-15	1.29E+00	3.88E-10	5.48E+00	3.73E-01	6.76E-01	8.77E+00	3.129202
	Best	0	1.07E-14	0.00E+00	7.86E+00	2.97E-09	4.11E-10	1.18E-06	3.24E-05	4.50E-04	2.85E+00
	Worst	3.55E-15	2.13E-14	3.55E-15	1.27E+01	4.78E-09	1.96E+01	1.47E+00	3.57E+00	20.4345	2.00E+01
F2	Average	0	1.260281	0	168.097	30.0477	2.29E+02	9.26E+01	8.50E+01	2.57E+01	273.0153
	Std Dev	0	4.796155	0.00E+00	20.5388	6.719748	3.38E+01	2.03E+01	2.76E+01	3.52E+01	5.43E+01
	Best	0	0	0	116.4862	18.90422	1.56E+02	6.27E+01	3.38E+01	2.60E-04	1.52E+02
	Worst	0	18.9042	0.00E+00	204.64	41.78827	293.5112	1.53E+02	1.40E+02	1.33E+02	3.59E+02
F3	Average	0	1.06E-54	4.21E-30	5.25E+00	7.04E-09	2.00E+00	2.56E-05	2.99E-05	8.21E-02	1.15E+01
	Std Dev	0	2.10E-54	1.28E-29	1.01E+00	5.68E-10	6.10E+00	3.56E-05	1.22E-06	1.33E-01	1.29E+01
	Best	0	5.16E-56	2.71E-44	3.93E+00	5.95E-09	5.70E-10	3.35E-07	2.79E-05	5.81E-05	6.32E-04
	Worst	0	8.67E-54	4.38E-29	7.45E+00	8.11E-09	2.00E+01	0.000149	3.28E-05	5.48E-01	28.28427
F4	Average	0	0.001723	0	5.93384	1.44335	60.13315	0.004186	0.009848	0.247561	57.22312
	Std Dev	0	0.005711	0	2.449968	0.72796	64.11064	0.008477	0.00998	0.32939	64.881
	Best	0	0.00E+00	0	1.53E+00	0.350652	0	4.30E-13	5.92E-08	7.83E-05	1.87E-06
	Worst	0	0.022141	0	11.31384	3.30963	180.336	0.03934	0.036919	0.952744	270.9139
F5	Average	2.34E-36	3.88E-105	2.90E-58	9.08E+04	244.6193	299933.2	2.98E-07	4.27E+04	0.38588	2.57E+06
	Std Dev	1.09E-35	1.37E-104	1.06E-57	3.87E+04	151.415	1141383	8.91E-07	2.05E+04	0.980396	1.89E+06
	Best	3.50E-40	2.39E-109	8.07E-91	3.03E+04	78.73644	0.134512	2.05E-09	1.77E+04	8.27E-06	1.15E+05
	Worst	5.99E-35	5.42E-104	4.17E-57	1.63E+05	761.5141	4498815	4.91E-06	9.83E+04	3.5341	6491725
F6	Average	62.18671	46.75566	8.95E-24	4.32E+04	44.75936	2667924	94.89612	76.9411	4952.241	800426
	Std Dev	51.01123	0.702115	2.60E-23	2.49E+04	0.528158	1461248	55.11365	109.9943	6071.786	2439969
	Best	44.42791	46.11286	0	7121.636	44.04607	37.49836	27.04189	44.79358	49.06581	80.781
	Worst	279.2602	48.56676	1.28E-22	8.23E+04	46.45274	800359	249.9203	643.8471	23217.09	8003304

Table 6.6: Results of the simulations in 50 dimensions.

Benchmark	Statistics	Algorithms									
		GGWO	GWO	GWCA	ABC	GSA	PSOGSA	PSO	SSA	SCA	MFO
F7	Average	2.44568	2.245981	0.39543	10.8829	3.89E-09	15.9417	2.21E-05	2.538247	4.521862	17.96136
	Std Dev	0.192152	0.245903	0.704873	1.22773	4.05E-10	3.092674	3.27E-05	0.53835	0.546814	3.045411
	Best	2.0781	1.69206	1.49E-07	8.23459	3.28E-09	2.85818	7.98E-07	1.374312	4.047869	2.57947
	Worst	2.769214	2.621921	2.140674	12.7309	4.58E-09	19.29966	0.000131	3.333027	6.377384	19.41094
F8	Average	3.681349	39.81563	8.68934	167.5958	28.45582	231.3938	99.33005	82.51514	193.6459	280.7398
	Std Dev	4.181849	13.17458	14.20356	16.70806	4.671152	41.25106	17.65016	23.03776	33.18219	54.5312
	Best	0	12.74205	0	129.7736	18.90422	159.193	68.6521	33.82857	136.1962	207.1563
	Worst	12.9344	69.08735	36.81349	191.6545	38.80337	313.4414	136.309	136.309	258.4204	423.467
F9	Average	0	1.1938	1.58E-15	5.11762	7.16E-09	7.22E-10	2.58E-05	3.04E-05	3.36659	9.82966
	Std Dev	0	0.33094	1.32E-15	1.1258	1.11E-09	4.51E-11	5.52E-05	1.96E-06	0.362941	12.70863
	Best	0	0.60971	2.04E-16	3.12082	5.59E-09	6.45E-10	1.40E-06	2.68E-05	2.84272	0.00037
	Worst	0	2.080546	5.95E-15	7.55719	1.11E-08	8.38E-10	0.000287	3.44E-05	4.44995	35.5646
F10	Average	0.03344	1.41276	0.05428	5.907881	1.901848	51.6887	0.00394	0.004104	3.468887	48.3729
	Std Dev	0.09759	0.18717	0.11744	2.50211	0.89989	57.03139	0.005661	0.006345	0.4469	61.78711
	Best	5.75E-11	1.140895	8.49E-07	1.57466	0.078681	1.11E-16	5.76E-14	4.17E-08	2.68606	1.32E-05
	Worst	0.52315	1.843354	0.578953	10.73243	4.461654	185.81	0.017226	0.022126	4.6253	186.1236
F11	Average	0	6132.125	3.75E-25	104725.1	327.9037	9289.844	1.92E-06	46914.21	50080.83	2069598
	Std Dev	0	3531.791	1.60E-24	58102.98	208.638	50629.72	6.67E-06	22315.88	18612.17	1356685
	Best	0	1200.19	1.08E-32	14737.57	61.2343	0.064453	7.99E-10	19700.93	16444.15	398587.8
	Worst	0	12604.24	8.78E-24	228529	798.1094	277356.2	3.61E-05	92357.54	86449.89	7605752
F12	Average	54.44438	669.3124	65.17494	57467.26	45.61572	54.55684	91.01528	131.4435	31723.78	7821421
	Std Dev	30.41612	227.4713	28.98188	48491.88	5.515901	24.25274	38.9426	166.5185	95236.72	32471414
	Best	44.70562	346.0705	47.47152	8973.947	44.35261	38.37738	39.82111	41.48509	3659.787	2.032239
	Worst	214.3801	1187.163	193.4626	227428.3	74.80706	131.1459	172.8936	640.1873	531852	1.67E+08

30 and 50. Besides, GGWO ranked fourth and third in dimensions 30 and 50, respectively. The results in Table 6.12 also show that the average ranking of the GGWO is 2.083333 and 2 regarding all the benchmark functions in dimensions 30 and 50, respectively. The outcomes rank the GGWO first among all other algorithms. Considering the best case, the GGWO is better than all the other algorithms. In addition, the worst case of GGWO is significantly lower than all the other algorithms. In other words, it obtained the best rank among all solution methods in terms of the best worst-case rank, which shows the robustness of the offered methodology. The results indicate that the GGWO can achieve very competitive outcomes compared to the other novel metaheuristic methods and perform better for most benchmark functions.

Table 6.7: Results of Tukey's multiple comparison test for dimensions 30 and 50.

Benchmark	Comparison P-value	Dimension 30		Dimension 50	
		Significant diff	P-value	Significant diff	
F1	GGWO-GWO	2.56E-13	Yes	1.81E-13	Yes
	GGWO-GWCA	0.33371	No	0.1694	No
	GGWO-ABC	1.21E-12	Yes	1.71E-12	Yes
	GGWO-GSA	1.21E-12	Yes	1.71E-12	Yes
	GGWO-PSOGSA	1.21E-12	Yes	1.71E-12	Yes
	GGWO-PSO	1.21E-12	Yes	1.71E-12	Yes
	GGWO-SSA	1.21E-12	Yes	1.71E-12	Yes
	GGWO-SCA	1.21E-12	Yes	1.71E-12	Yes
	GGWO-MFO	1.21E-12	Yes	1.71E-12	Yes
F2	GGWO-GWO	0.16074	Yes	0.1608	Yes
	GGWO-GWCA	Nan	No	Nan	No
	GGWO-ABC	1.20E-12	Yes	1.21E-12	Yes
	GGWO-GSA	1.18E-12	Yes	1.21E-12	Yes
	GGWO-PSOGSA	1.20E-12	Yes	1.21E-12	Yes
	GGWO-PSO	1.20E-12	Yes	1.21E-12	Yes
	GGWO-SSA	1.20E-12	Yes	1.21E-12	Yes
	GGWO-SCA	1.20E-12	Yes	1.21E-12	Yes
	GGWO-MFO	1.20E-12	Yes	1.21E-12	Yes
F3	GGWO-GWO	1.21E-12	Yes	1.20E-12	Yes
	GGWO-GWCA	1.21E-12	Yes	1.20E-12	Yes
	GGWO-ABC	1.21E-12	Yes	1.20E-12	Yes
	GGWO-GSA	1.21E-12	Yes	1.20E-12	Yes
	GGWO-PSOGSA	1.20E-12	Yes	1.20E-12	Yes
	GGWO-PSO	1.21E-12	Yes	1.20E-12	Yes
	GGWO-SSA	1.21E-12	Yes	1.20E-12	Yes
	GGWO-SCA	1.21E-12	Yes	1.20E-12	Yes
	GGWO-MFO	1.20E-12	Yes	1.20E-12	Yes
F4	GGWO-GWO	0.041926	Yes	0.081493	No
	GGWO-GWCA	Nan	No	Nan	No
	GGWO-ABC	1.21E-12	Yes	1.20E-12	Yes
	GGWO-GSA	3.45E-07	Yes	1.20E-12	Yes
	GGWO-PSOGSA	5.76E-11	Yes	5.72E-11	Yes
	GGWO-PSO	1.70E-08	Yes	1.20E-12	Yes
	GGWO-SSA	1.21E-12	Yes	1.20E-12	Yes
	GGWO-SCA	1.21E-12	Yes	1.20E-12	Yes
	GGWO-MFO	1.21E-12	Yes	1.20E-12	Yes

Table 6.8: Results of Tukey's multiple comparison test for dimensions 30 and 50.

Benchmark	Comparison	Dimension 30		Dimension 50	
		P-value	Significant diff	P-value	Significant diff
F5	GGWO-GWO	3.02E-11	Yes	2.98E-11	Yes
	GGWO-GWCA	3.02E-11	Yes	2.98E-11	Yes
	GGWO-ABC	3.02E-11	Yes	2.98E-11	Yes
	GGWO-GSA	3.02E-11	Yes	2.98E-11	Yes
	GGWO-PSOGSA	3.02E-11	Yes	2.98E-11	Yes
	GGWO-PSO	3.02E-11	Yes	2.98E-11	Yes
	GGWO-SSA	3.02E-11	Yes	2.98E-11	Yes
	GGWO-SCA	3.02E-11	Yes	2.98E-11	Yes
	GGWO-MFO	3.02E-11	Yes	2.98E-11	Yes
F6	GGWO-GWO	1.75E-05	Yes	0.000167	Yes
	GGWO-GWCA	2.11E-11	Yes	1.94E-11	Yes
	GGWO-ABC	3.02E-11	Yes	2.98E-11	Yes
	GGWO-GSA	0.40354	No	0.063459	No
	GGWO-PSOGSA	0.83026	No	0.5394	No
	GGWO-PSO	0.22823	No	0.051812	No
	GGWO-SSA	3.83E-06	Yes	8.62E-05	Yes
	GGWO-SCA	9.51E-06	Yes	4.57E-10	Yes
	GGWO-MFO	0.007959	Yes	9.65E-10	Yes
F7	GGWO-GWO	0.012111	Yes	0.003828	Yes
	GGWO-GWCA	1.04E-10	Yes	3.60E-11	Yes
	GGWO-ABC	2.86E-11	Yes	2.95E-11	Yes
	GGWO-GSA	2.86E-11	Yes	2.95E-11	Yes
	GGWO-PSOGSA	0.000222	Yes	2.95E-11	Yes
	GGWO-PSO	2.86E-11	Yes	2.95E-11	Yes
	GGWO-SSA	0.013165	Yes	0.3552	No
	GGWO-SCA	2.86E-11	Yes	2.95E-11	Yes
	GGWO-MFO	0.006316	Yes	7.21E-11	Yes
F8	GGWO-GWO	6.44E-10	Yes	3.09E-11	Yes
	GGWO-GWCA	0.17834	No	0.2345	No
	GGWO-ABC	2.31E-11	Yes	2.80E-11	Yes
	GGWO-GSA	2.31E-11	Yes	2.78E-11	Yes
	GGWO-PSOGSA	2.31E-11	Yes	2.80E-11	Yes
	GGWO-PSO	2.31E-11	Yes	2.80E-11	Yes
	GGWO-SSA	2.31E-11	Yes	2.80E-11	Yes
	GGWO-SCA	2.31E-11	Yes	2.80E-11	Yes
	GGWO-MFO	2.31E-11	Yes	2.80E-11	Yes

Table 6.9: Results of Tukey's multiple comparison test for dimensions 30 and 50.

Benchmark	Comparison	Dimension 30		Dimension 50	
		P-value	Significant diff	P-value	Significant diff
F9	GGWO-GWO	1.21E-12	Yes	1.21E-12	Yes
	GGWO-GWCA	1.21E-12	Yes	1.21E-12	Yes
	GGWO-ABC	1.21E-12	Yes	1.21E-12	Yes
	GGWO-GSA	1.21E-12	Yes	1.21E-12	Yes
	GGWO-PSOGSA	1.21E-12	Yes	1.21E-12	Yes
	GGWO-PSO	1.21E-12	Yes	1.21E-12	Yes
	GGWO-SSA	1.21E-12	Yes	1.21E-12	Yes
	GGWO-SCA	1.21E-12	Yes	1.21E-12	Yes
	GGWO-MFO	1.21E-12	Yes	1.21E-12	Yes
F10	GGWO-GWO	3.02E-11	Yes	0.000587	Yes
	GGWO-GWCA	0.20095	No	3.02E-11	Yes
	GGWO-ABC	3.02E-11	Yes	4.50E-11	Yes
	GGWO-GSA	0.004215	Yes	0.002052	Yes
	GGWO-PSOGSA	0.023234	Yes	0.001114	Yes
	GGWO-PSO	0.000182	Yes	0.099258	No
	GGWO-SSA	0.44642	No	3.02E-11	Yes
	GGWO-SCA	3.02E-11	Yes	1.87E-05	Yes
	GGWO-MFO	0.039167	Yes	0.000587	Yes
F11	GGWO-GWO	1.21E-12	Yes	1.21E-12	Yes
	GGWO-GWCA	4.57E-12	Yes	1.21E-12	Yes
	GGWO-ABC	1.21E-12	Yes	1.21E-12	Yes
	GGWO-GSA	1.21E-12	Yes	1.21E-12	Yes
	GGWO-PSOGSA	1.21E-12	Yes	1.21E-12	Yes
	GGWO-PSO	1.21E-12	Yes	1.21E-12	Yes
	GGWO-SSA	1.21E-12	Yes	1.21E-12	Yes
	GGWO-SCA	1.21E-12	Yes	1.21E-12	Yes
	GGWO-MFO	1.21E-12	Yes	1.21E-12	Yes
F12	GGWO-GWO	4.08E-11	Yes	3.02E-11	Yes
	GGWO-GWCA	8.88E-06	Yes	3.32E-06	Yes
	GGWO-ABC	3.02E-11	Yes	3.02E-11	Yes
	GGWO-GSA	1.87E-05	Yes	1.17E-09	Yes
	GGWO-PSOGSA	0.030317	Yes	0.000225	Yes
	GGWO-PSO	0.29047	No	0.004033	Yes
	GGWO-SSA	0.05012	No	0.43764	No
	GGWO-SCA	3.02E-11	Yes	3.02E-11	Yes
	GGWO-MFO	2.39E-08	Yes	1.85E-08	Yes

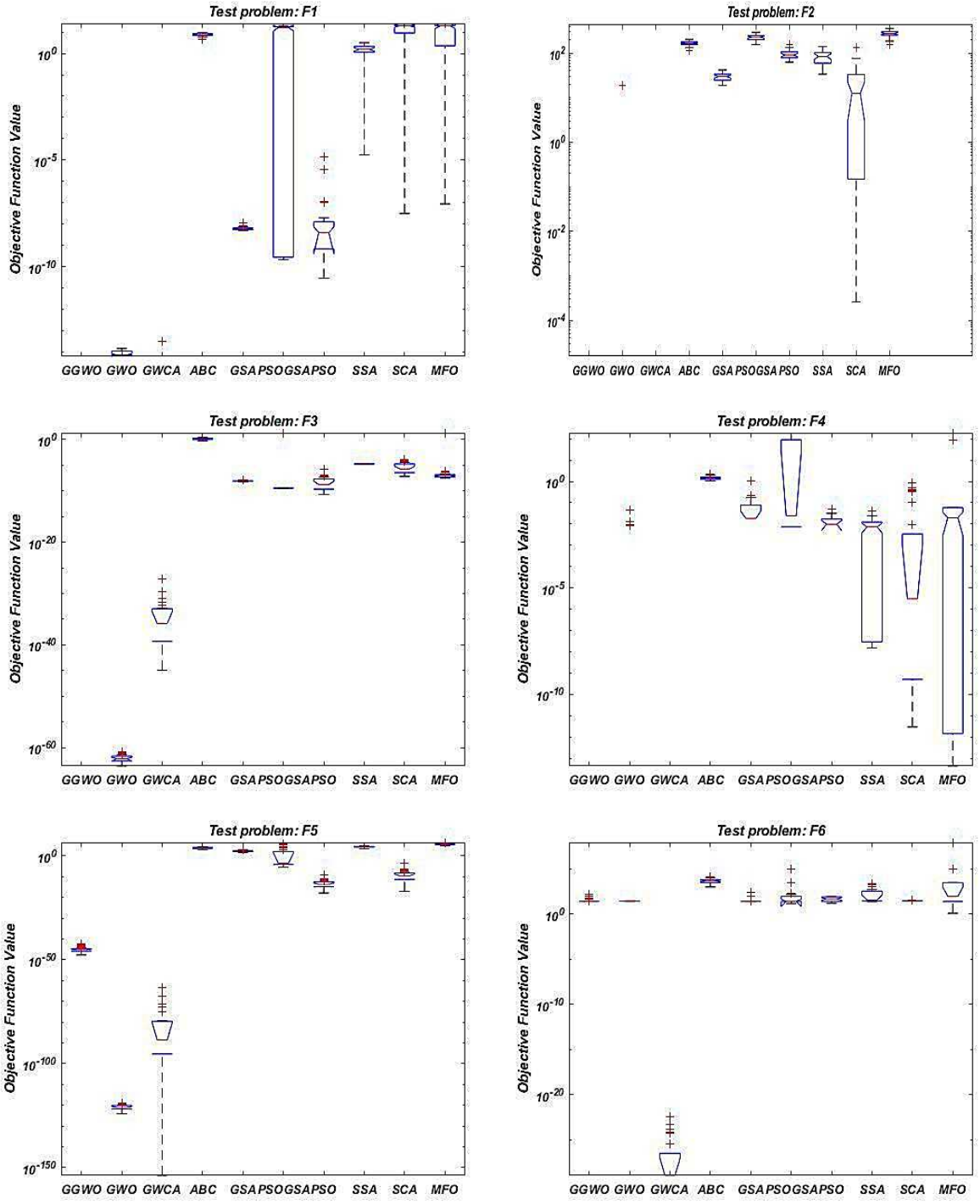

Figure 6.2: Dimension 30 boxplots.

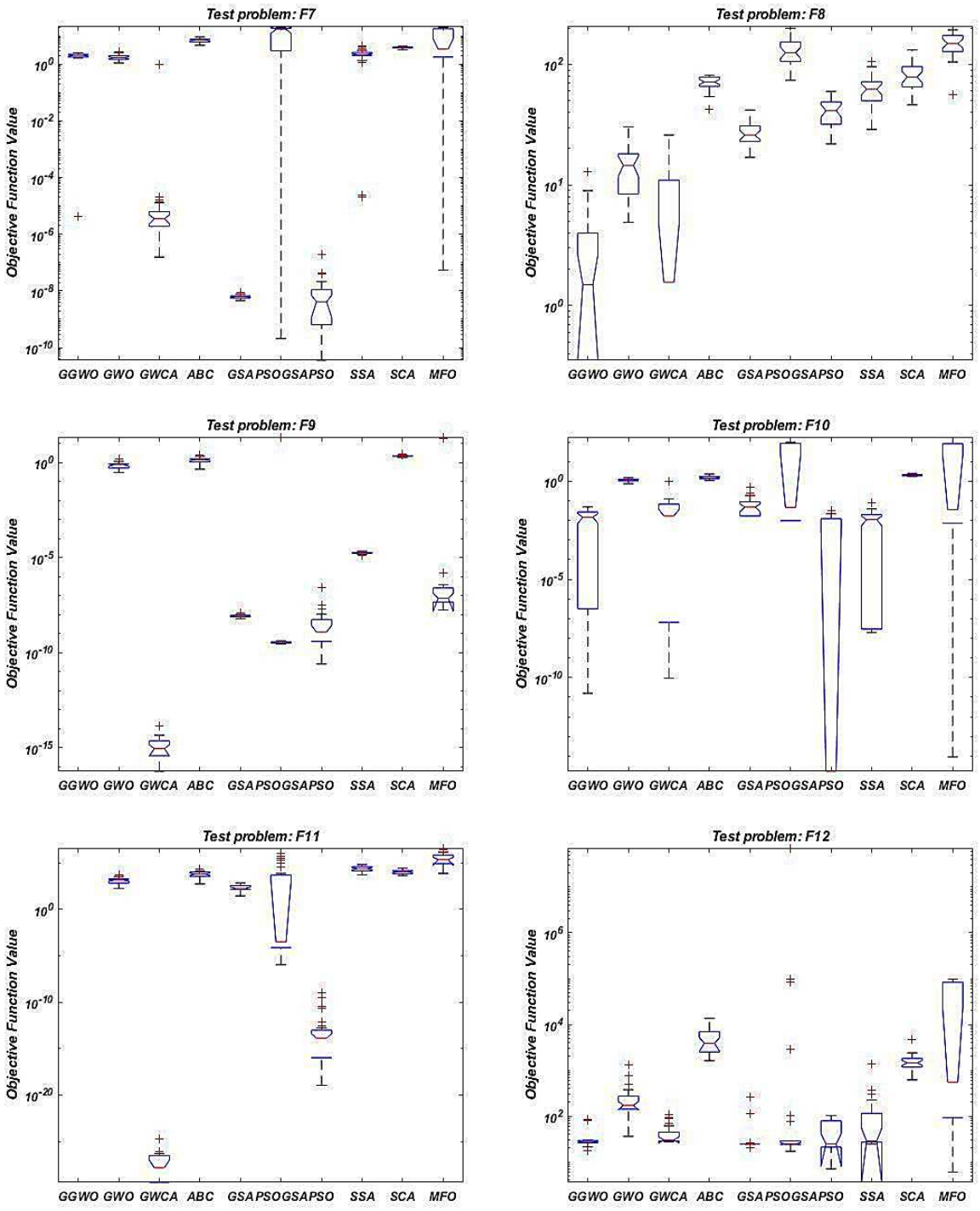

Figure 6.3: Dimension 30 boxplots.

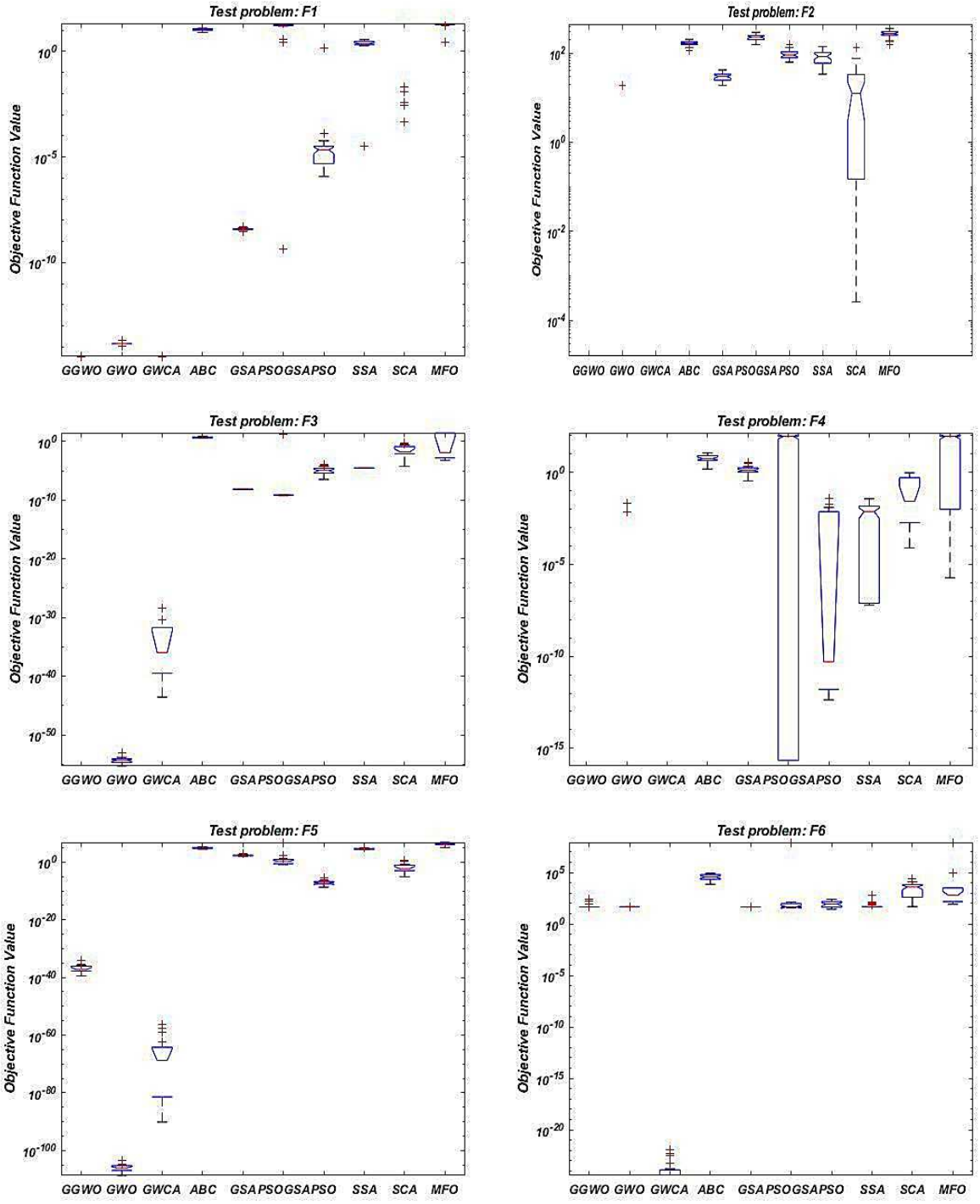

Figure 6.4: Dimension 50 boxplots.

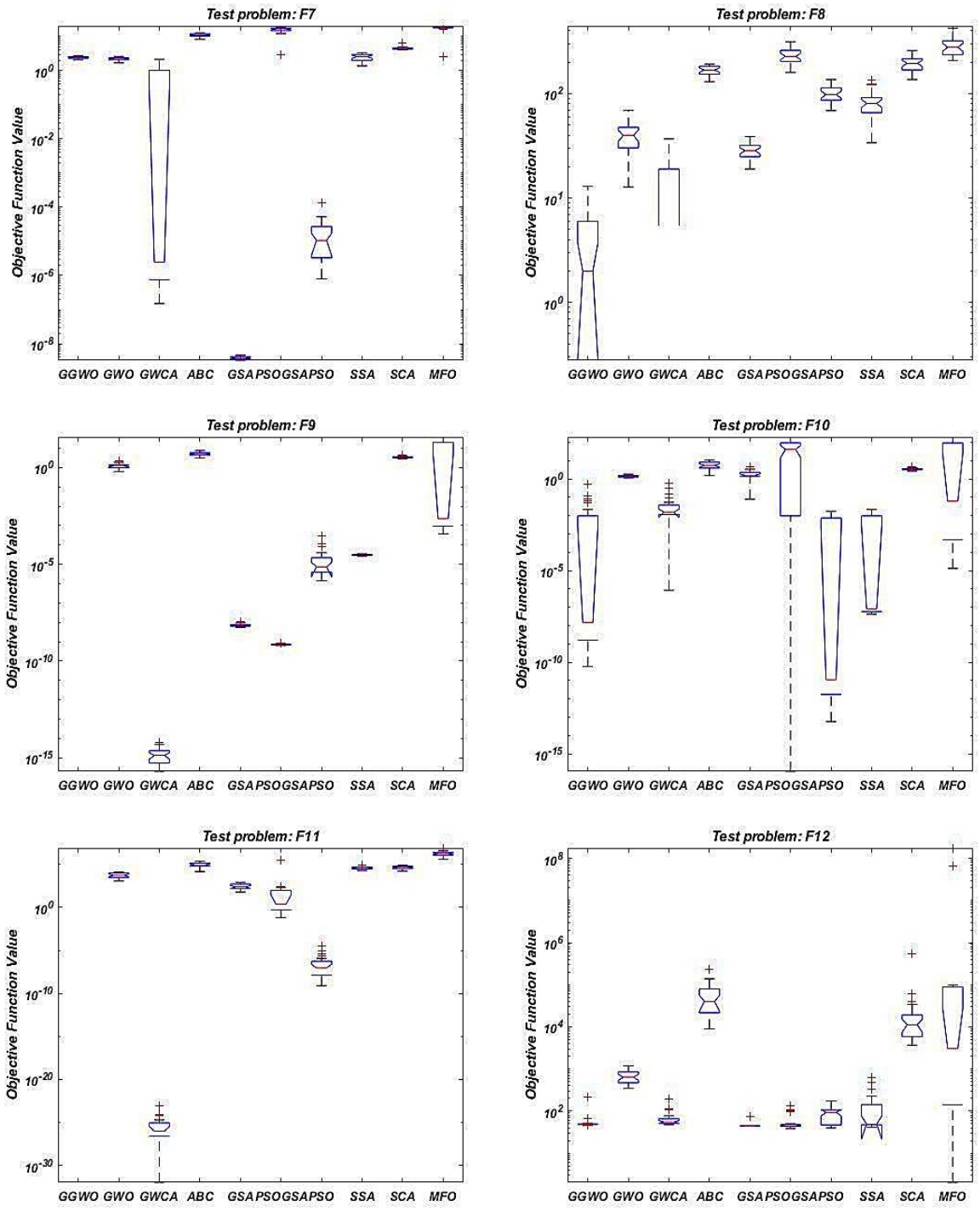

Figure 6.5: Dimension 50 boxplots.

Table 6.10: Friedman's test for dimension 30.

	GGWO	GWO	GWCA	ABC	GSA	PSOGSA	PSO	SSA	SCA	MFO
F1	10.33	25.17	11.00	73.80	44.03	69.53	42.27	62.40	84.77	81.70
F2	15.17	16.57	15.17	76.40	42.40	87.20	60.63	58.90	39.90	92.67
F3	5.50	15.50	25.50	92.83	50.77	46.00	45.90	80.33	74.50	68.17
F4	18.50	24.53	18.50	90.17	54.98	69.47	50.15	57.60	54.27	66.83
F5	25.50	5.93	15.07	74.83	63.03	61.37	36.57	83.10	44.63	94.97
F6	37.76	47.30	5.50	92.60	35.57	46.83	49.73	67.37	59.17	63.17
F7	50.33	46.47	29.10	83.87	13.13	73.93	11.57	55.67	72.83	68.10
F8	9.17	24.73	13.97	66.03	35.73	87.57	46.50	60.47	69.97	90.87
F9	5.50	75.13	15.50	82.97	43.43	31.97	34.70	63.83	93.40	58.57
F10	32.53	70.23	37.90	79.17	43.08	54.58	17.33	30.23	88.07	51.87
F11	5.67	52.97	15.33	65.80	42.57	49.17	25.50	81.87	72.77	93.37
F12	32.30	65.60	45.43	90.73	20.13	29.37	27.00	42.30	80.10	72.03

Table 6.11: Friedman's test for dimension 50.

	GGWO	GWO	GWCA	ABC	GSA	PSOGSA	PSO	SSA	SCA	MFO
F1	10	25.5	11	69.5	35.83	76.4	46.03	58.26	86.23	86.23
F2	15.16	16.7	15.16	73.53	46.06	88.06	56	67.06	35.76	91.46
F3	5.5	15.5	25.5	89.83	44.5	41.03	56.86	62.16	78.73	85.36
F4	15.5	19.68	15.5	84.5	74.76	66.21	42.33	52.33	58.43	75.73
F5	25.5	5.5	15.5	83.53	64.7	57.4	35.5	76.26	46.33	94.76
F6	36	44.66	5.5	93.16	27.1	40.36	50.73	50.56	78.5	78.4
F7	47.03	41.46	19.33	76.26	5.5	86.26	22.03	47.93	66.16	93
F8	11.35	33	12.58	68.86	25.63	84.7	52.83	47.7	75.8	92.53
F9	5.5	71.5	15.5	90.76	35.5	25.5	47.2	53.8	82.23	77.5
F10	23.93	59.1	38.96	84.2	64.43	59.36	15.36	23.2	76.83	59.6
F11	5.5	55.16	15.5	81.16	44.46	37.86	25.5	71.43	72.9	95.5
F12	32.36	69.63	42.63	90.7	14.46	18.06	42.8	39.36	82.93	72.03

Table 6.12: Ranking of the algorithms based on Friedman's test for dimension 30.

	GGWO	GWO	GWCA	ABC	GSA	PSOGSA	PSO	SSA	SCA	MFO
F1	1	3	2	8	5	7	4	6	10	9
F2	1	3	2	8	5	9	6	7	4	10
F3	1	2	3	10	6	5	4	9	8	7
F4	1	3	2	10	6	9	4	7	5	8
F5	3	1	2	8	7	6	4	9	5	10
F6	3	5	1	10	2	4	6	9	7	8
F7	5	4	3	10	2	9	1	6	8	7
F8	1	3	2	7	4	9	5	6	8	10
F9	1	8	2	9	5	3	4	7	10	6
F10	3	8	4	9	5	7	1	2	10	6
F11	1	6	2	7	4	5	3	9	8	10
F12	4	7	6	10	1	3	2	5	9	8
Average	2.08	4.41	2.58	8.83	4.33	6.33	3.66	6.83	7.66	8.25
Best	1	1	1	7	1	3	1	2	4	6
Worst	5	8	6	10	7	9	6	9	10	10

Table 6.13: Ranking of the algorithms based on Friedman's test for dimension 50.

	GGWO	GWO	GWCA	ABC	GSA	PSOGSA	PSO	SSA	SCA	MFO
F1	1	3	2	7	4	8	5	6	9	10
F2	1	3	2	8	5	9	7	6	4	10
F3	1	2	3	10	5	4	6	7	8	9
F4	1	3	2	10	8	7	4	5	6	9
F5	3	1	2	9	7	6	4	8	5	10
F6	3	5	1	10	2	4	7	6	9	8
F7	5	4	2	8	1	9	3	6	7	10
F8	1	4	2	7	3	9	6	5	8	10
F9	1	7	2	10	4	3	5	6	9	8
F10	3	5	4	10	8	6	1	2	9	7
F11	1	6	2	9	5	4	3	7	8	10
F12	3	7	5	10	1	2	6	4	9	8
Average	2	4.166	2.41	9	4.41	5.91	4.75	5.66	7.58	9.08
Best	1	1	1	7	1	2	1	2	4	7
Worst	5	7	5	10	8	9	7	8	9	10

6.5 A case study of the COVID-19 Pandemic in the United States

We use one of the most recently developed models called SIDARTHE to forecast the COVID-19 pandemic (Giordano et al., 2020). The model reflects eight states, including susceptible, infected, diagnosed, ailing, recognized, threatened, healed, extinct cases. This formulation takes into account several health states for patients. The recommended formulation consists of several differential equations to demonstrate the outbreak. For more information about this model please refer to Appendix E.

he United States is part of the COVID-19 pandemic created by acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The country announced its first community transmission case of

COVID-19 in January 2020. Up to date, the US has reported more than 4,918,420 COVID-19 cases and 160,290 death cases, making it the country with the most COVID-19 cases. To optimize the limited resources of the healthcare systems, it is crucial to forecast the pandemic's future trends. This approach will enable managers to estimate the peak of the outbreak and plan for the worst-case scenario. Since COVID-19 is a novel virus, the epidemiological parameters are unknown (Ahamad et al., 2020). Thus, we need to present novel methodologies to model the outbreak.

In the following, we resolve the sum of the mean square error model using the GGWO and attain the optimum result for the model. Figure 6.6 displays the convergence of GGWO and the average objective value of the grey wolves over the iterations.

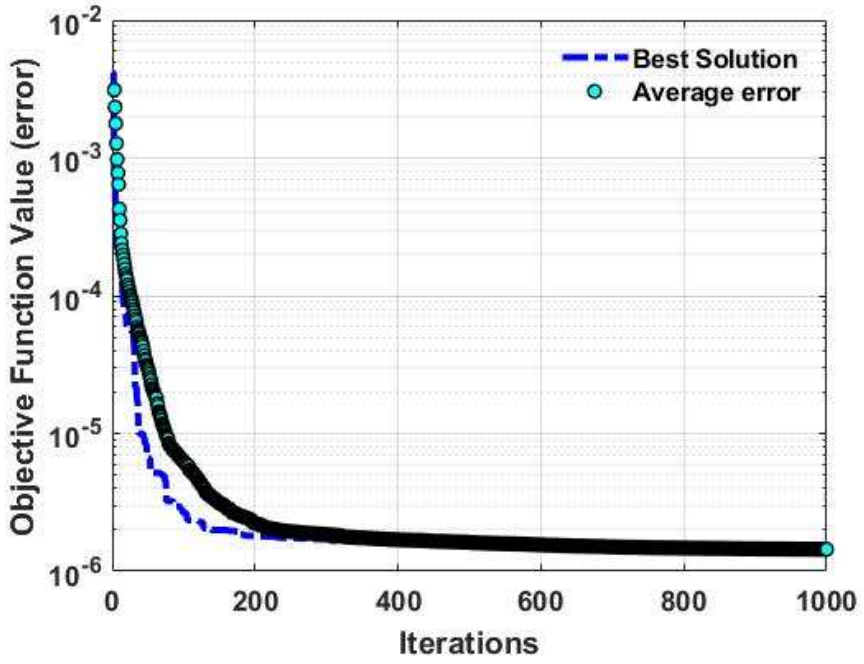

Figure 6.6: Convergence plot of the GGWO.

Table 6.14: Results of fitting the model to real-data for the U.S.

Parameters	Stages					
	Jan 22, to Mar 13	Mar 13 to Mar 22	Mar 22, to Mar 26	Mar 26, to May 22	May 22 to May 25	after May 25
α	0.13946	0.442191	0.442191	0.088807	0.126069	0.126069
β	0.002902	0.004443	0.004443	0.004443	7.22E-05	7.22E-05
δ	0.036517	0.285809	0.285809	0.02874	0.033251	0.033251
γ	0.002902	0.004443	0.004443	0.004443	7.22E-05	7.22E-05
ε	0.019209	0.019209	0.017224	0.017224	0.017224	0.02025
ζ	0.054364	0.054364	0.054364	0.022792	0.022792	0.000193
η	0.054364	0.054364	0.054364	0.022792	0.022792	0.000193
θ	0.070311	0.070311	0.070311	0.070311	0.070311	0.070311
λ	0.013641	0.013641	0.013641	0.067394	0.067394	0.067394
κ	0.009172	0.009172	0.009172	0.008573	0.012395	0.012395
ξ	0.009172	0.009172	0.009172	0.008573	0.012395	0.012395
ρ	0.013641	0.013641	0.013641	0.008573	0.012395	0.012395
σ	0.009172	0.009172	0.009172	0.008573	0.008573	0.0003
μ	0.005856	0.005856	0.005856	0.005631	0.005631	0.005631
υ	0.031166	0.031166	0.031166	0.029214	0.029214	0.029214
τ	0.004884	0.004884	0.004884	0.004884	0.004884	0.004884

We note that we did not reflect the mean square error of the death cases since the data might be profoundly affected by patients' age, health state, and gender. The data used in this research is available at Humanitarian Data Exchange (2020). In Figure 6.6, we can observe a perfect trade-off between exploration and exploitation in the performance of the GGWO. Table 6.14 presents the results of the case study in the US. Figure 6.7 describes the accuracy of the predicted model versus real-data. We observe that the offered procedure forecast future trends precisely. It is also noteworthy that the sum of square errors for suggested parameters is 1.44E-06.

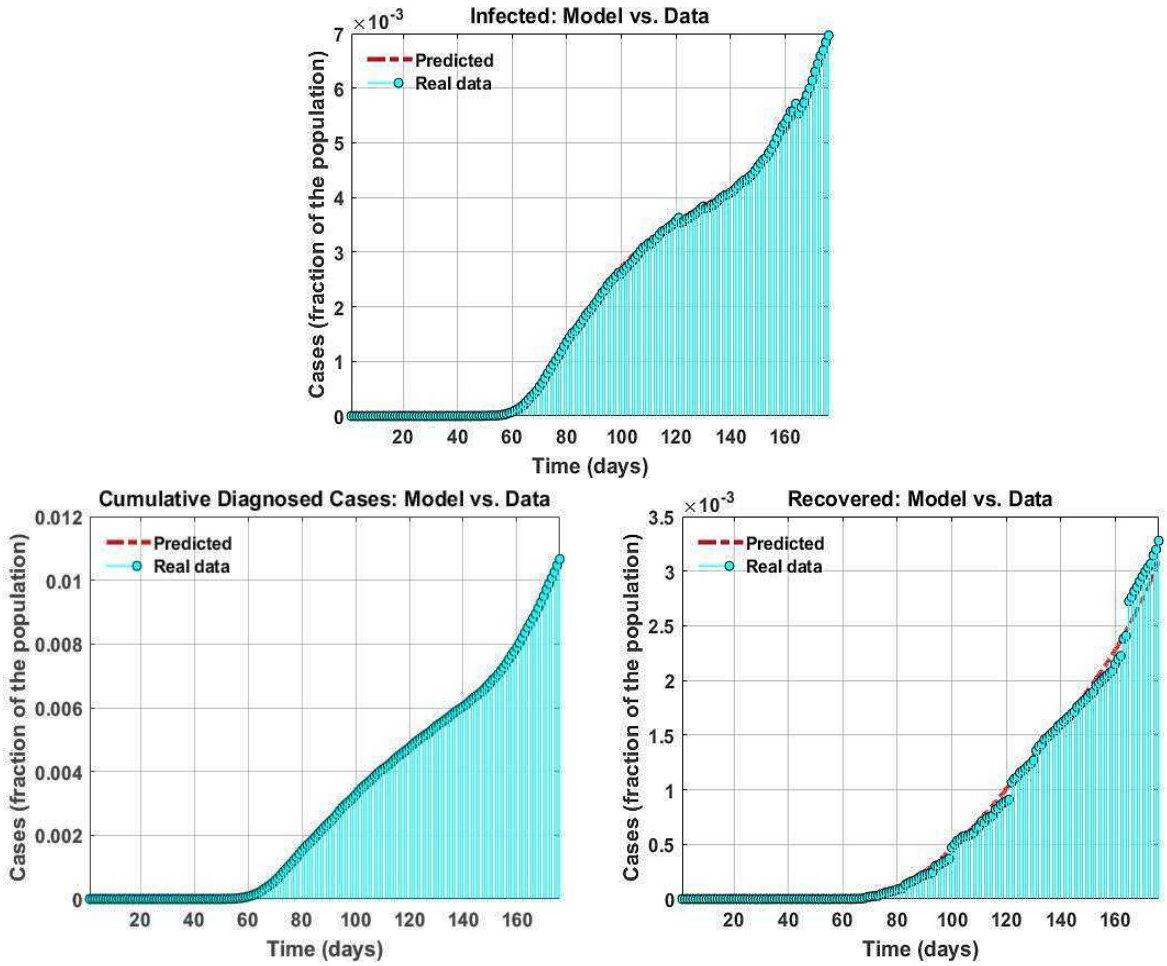

Figure 6.7: Prediction vs. real-data from the US.

Figure 6.8 shows the predicted number of different types of individuals that will develop life-threatening symptoms. Based on the outcomes of our study, we predict that the US will experience the peak of the pandemic in terms of infected cases during mid-November 2020. Our model forecasts that the number of infected cases in the US could reach 16 million by that time. Figure 6.8 depicts an accurate prediction on the number of infected cases that develop life-threatening symptoms in the future so that the policymakers and healthcare professionals, and managers could plan for ICU and ventilator allocation.

During the first stage of the outbreak in the US, the transmission rates were low from January 22 to March 13. Based on our results, the reproduction rate was $R_0 = 1.9249$, for this stage. On February 26, 2020, the first community case of the US was reported by the Centers for Disease

Control and Prevention (CDC). On March 2, 13, and 16, the US government applied some travel restrictions from 26 European countries and the UK and Ireland, respectively, to contain the spread of the virus. On March 11, 2020, the World Health Organization (WHO) stated the outbreak to be a pandemic. In our study, we consider March 13-22 as the second stage due to the fact that a rapid increase in the number of cases was reported in the country. The reproduction rate was approximately $R_0 = 7.2482$ for this stage. As becomes evident, the reproduction number significantly increased in this stage due to community transmission.

On March 19, the testing capacity remarkably increased to detect more infected cases. Since, in this critical time interval, the testing capacity increased, so the detection rate changed from the previous stages. Therefore, we considered March 22-26 as the third stage in our study. During this stage, the reproduction number was $R_0 = 7.4134,$ which shows exponential growth in this time interval. During the fourth stage, March 26 to May 22, the transmission rates were considerably reduced due to dynamic lockdown and social distancing measures. That is why we observe a meaningful reduction in the reproduction rate for this stage $R_0 = 0.8999$. In the fifth stage, May 22 to May 25, we observe an increase in the transmission rates of the virus since other states start to experience exponential growth in the number of COVID-19 cases. During this stage, the reproduction rate reported $R_0 = 1.2581$. In the last stage, after May 25, we observe an increase in the ε parameter due to a significant increase in the number of everyday tests resulting in $R_0 = 1.4374$. In the case of continuing the current measures and restrictions, we predicted the future trends of the pandemic in the US over the next 362 days. Figure 6.8 presents more details about the model and predictions.

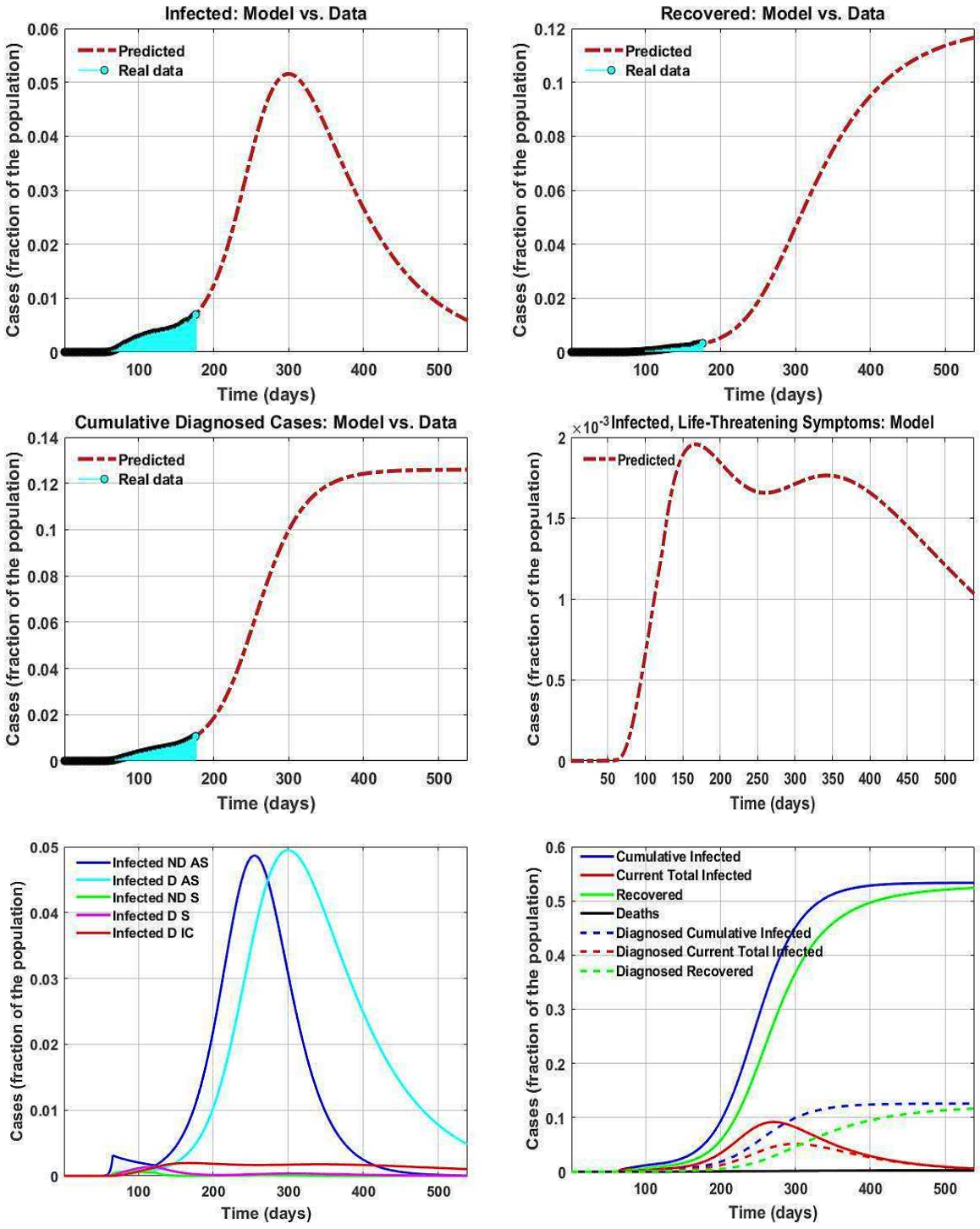

Figure 6.8: Prediction of future pandemic trends.

Based on the outcomes, we determined that keeping the current restrictions such as social distancing and partial lockdowns in place will significantly help to slow down the spread of the virus. It

worth mentioning that any deviation in the future parameters could significantly affect the predicted trends. Therefore, it is crucial to study the effect of changes in the main parameters of the pandemic on future outcomes.

6.6 Sensitivity Analyses and Managerial Insights

The provided forecast of future pandemic growth in the earlier section considers strict social distancing, behavioral recommendations, and preventing gatherings. However, some businesses are permitted to reopen while continuing social distancing. On the other hand, the pandemic has been started to evolve in more states in the country. Hence, it is vital to discover the effects of reopenings and pandemic growth that create variations in the transmission rates on the upcoming situations. Therefore, we augmented the values of the parameters $\alpha,\ \beta,\ \gamma,\ \delta,\ and\ \varepsilon$, and explored the outcomes. Such deviations could meaningfully influence the number of cases. We portray the results in Figures 6.9 to 6.11. Our results show that the parameter α plays a prominent role in the number of infected cases. Increasing parameter α surge the number of infected people considerably.

Increasing α also rises the number of infected people who develop life-threatening symptoms significantly. Therefore, it becomes disclosed that strict measures such as social distancing are the only factors that can decrease this parameter. Increasing α more than four percent will result in a new higher peak in the number of patients who need ICU admission. Our results show that the US will experience its peak in the number of infected people from November 1, 2020, to January 10, 2021. Any increase in this parameter of more than 4 percent will create another peak in the number of infected cases who need ICU admission. Therefore, social distancing, wearing masks, and avoiding gatherings are the most critical factors that will help the country pass the peaks. Increasing other parameters such as β and δ have the same effect; however, their influence on the number of infected cases is lower than those of parameter α.

Moreover, increasing the value of ε considerably decreases the portion of infected, recovered, cumulative diagnosed, and death cases. Hence, to contain the virus and stop the pandemic, we can increase testing capacity by at least 40 percent to avoid experiencing another surge of infection who need ICU admission. We discovered that increasing parameter ε by 100 percent would reduce the

total infected case in the upcoming peak by 50 percent. Our study discovered that asymptomatic cases play the most substantial role in spreading the virus.

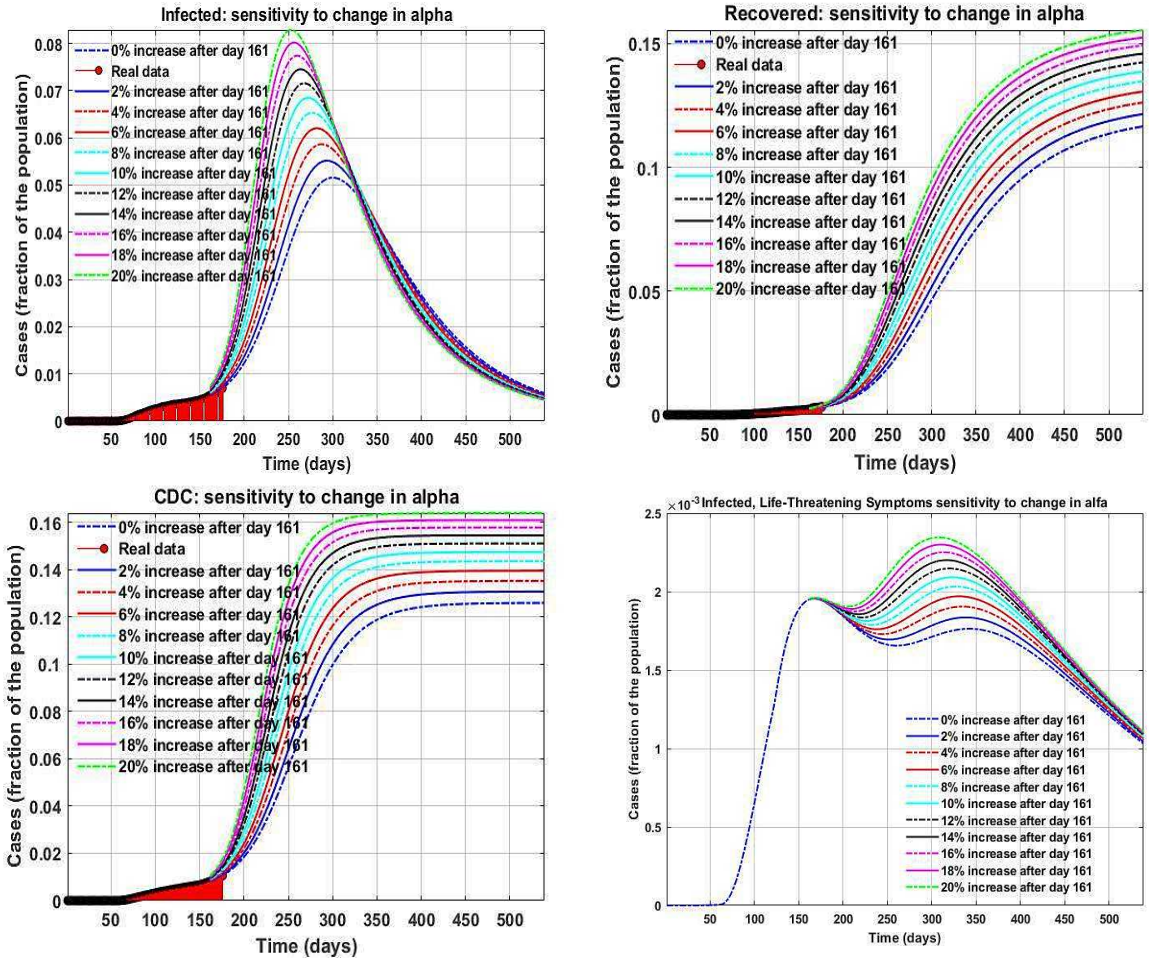

Figure 6.9: Prediction of the COVID-19 cases in the U.S. as Parameter α increases.

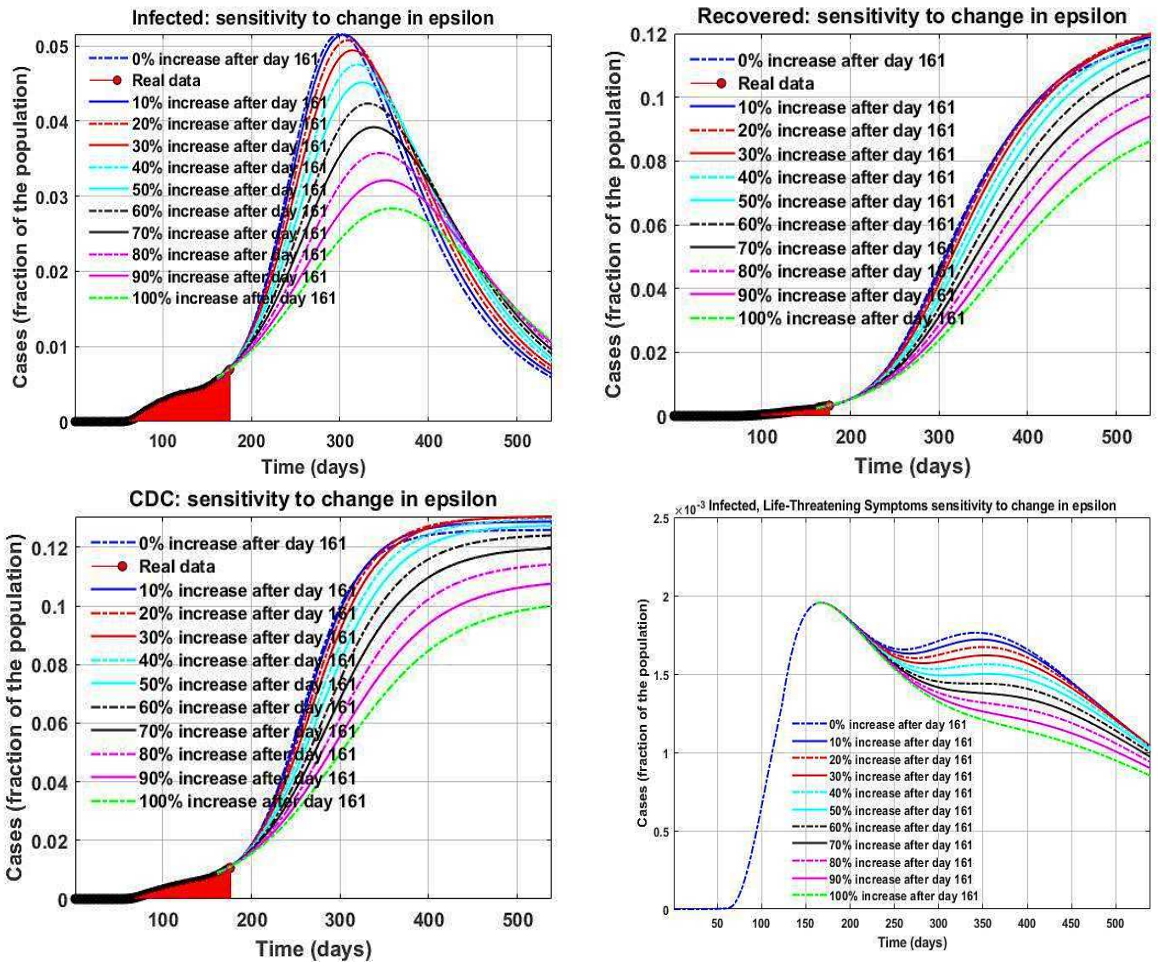

Figure 6.10: Prediction of the COVID-19 cases in the U.S. as Parameter ϵ increases.

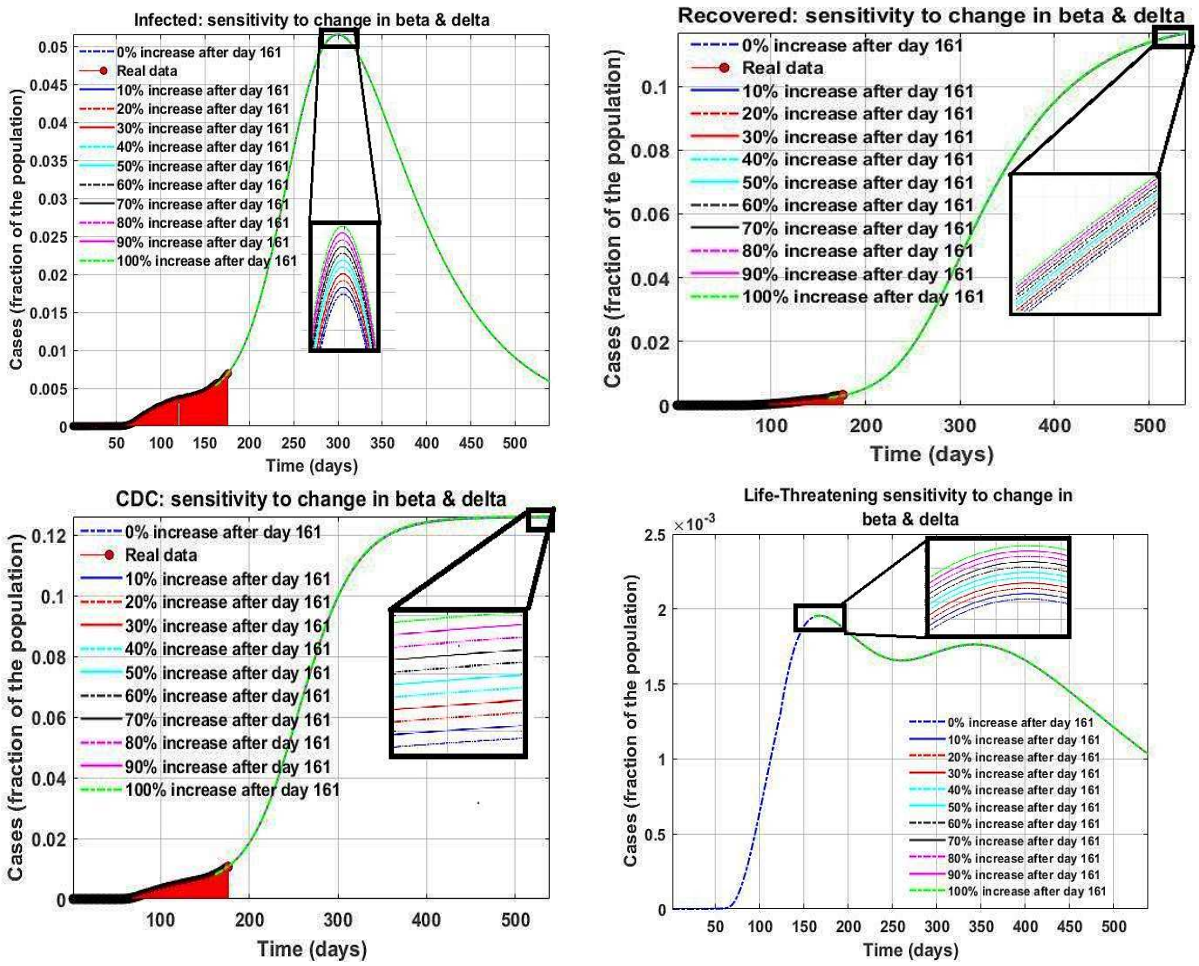

Figure 6.11: Prediction of the COVID-19 cases in the U.S. as Parameter β and δincreases.

6.7 Conclusion and outlook

The original GWO algorithm cannot maintain a proper balance between exploration and exploitation. In this research, we address this issue by presenting a new version of this algorithm, called GGWO, that enables us to solve optimization problems accurately. Our algorithm used the advantages of the gradient that provides valuable information about the solution space. Using gradient information, we accelerated the algorithm that enables us to solve many well-known complex benchmark functions optimally for the first time in the field. Besides, we used deep mathematical concepts such as Gaussian walk and Lévy flight to improve the search efficiency of our methodology. These contributions enabled the proposed algorithm to avoid trapping in local optima. Our

computational results on several benchmarks demonstrated the superiority of our algorithm to other algorithms in the literature. Moreover, we applied several robust statistical tests to determine significant differences in the performance of the algorithm compared to state-of-the-art methodologies. Our outcomes revealed that our algorithm is able to solve most benchmarks optimally without trapping in local optima for the first time. Moreover, in instances with dimension 50, Friedman's test showed that our algorithm's average rank is 2, which is the best average rank among the analyzed algorithms. In 7 out of 12 benchmarks, the proposed algorithm was ranked first.

Moreover, we applied our algorithm for predicting the COVID-19 pandemic in the US. Our results projected the highest number of infected individuals in the United States in mid-November 2020. The results also determined the peak of the number of hospitalized cases. Besides, we performed several analyses to depict upcoming scenarios of the pandemic to help the authorities. The results showed that the transmission rate from an infected person to a susceptible case is the most critical factor in future trends. A surge in this constant would meaningfully raise the total number of cases. Besides, rising the transmission rate from a diagnosed or recognized person to a susceptible case causes a surge in the total number of cases. Moreover, any increase in the value of ε decreases the total number of cases. Thus, to contain the virus, governments should reduce the infection transmission rate by applying more restrictions on social activities and simultaneously increasing daily tests. Our study revealed that asymptomatic cases have the most significant role in spreading the virus.

As one of the potential research avenues, it would be interesting to take stochasticity and uncertainty into account. In addition, considering the effect of information sharing and spread in pandemic growth would be interesting (Belen et al., 2011). Besides, considering other factors such as age, sex, race, and health condition would significantly increase the accuracy of the model. From an algorithmic perspective, presenting a multi-objective version of the proposed algorithm could solve many-objective optimization problems. Moreover, the authorities could use the proposed methodology to optimize resource allocation during the outbreak. Furthermore, healthcare managers could plan for testing kit allocation to test centers using the offered prediction methodology.

Chapter 7

Conclusion

This dissertation focused on addressing complex challenges in appointment scheduling, operating room planning, and modeling and prediction of COVID-19 pandemic. This study significantly advanced research in these healthcare areas by introducing innovative stochastic programming models, and machine learning and evolutionary computation based algorithms. We provided a new framework for appointment scheduling that comprehensively addressed uncertainties including no-shows, unpunctuality, and service time variability. In operating room planning, we proposed a novel state-variable model that efficiently handled surgical time uncertainty, that demonstrated substantial operational improvements. The study on pandemic prediction employed advanced algorithms to enhance forecasting accuracy that contributed valuable insights for healthcare policymakers and healthcare professionals. Each area benefits from innovative methodologies that promote efficiency, accuracy, and resilience in healthcare systems.

In chapter 2 we proposed a novel stochastic programming model for appointment scheduling problem considering uncertainties in service times, unpunctuality and no-shows. These uncertainties were characterized by their patient-and-time-dependent distributions which is critical aspect that previous models have often oversimplified due to modeling complexity. We showed that our model is efficient in solving large-scale instances optimally within a reasonable computational time. The proposed model captured an exponential number of scenarios while maintaining a pseudo-polynomial number of variables and constraints for the first time in this field. Our approach showed significant improvement in appointment scheduling that is demonstrated by a notable reduction

in total clinic costs. Specifically, we observed a comprehensive cost reduction of 34% attributed to the consideration of patient-dependent service times. Furthermore, the model's capability to consider patient-and-time-dependent unpunctuality and no-show probabilities resulted in additional cost reductions of 12% and 67%, respectively. These findings highlighted the critical importance of integrating patient behavior and time dependency in appointment scheduling models to optimize healthcare delivery and operational efficacy.

Our research on personalized reminders as a strategic approach to reduce no-shows within our model revealed the profound impact of personalized communication in reducing no-show rates. This consideration resulted in a further 23% reduction in total costs. This strategic insight highlighted the transformative potential of using data-driven, patient-centric communication strategies to enhance clinic efficiency and patient engagement. Our model not only challenged the conventional paradigms of healthcare scheduling but also set a new benchmark for integrating patient-and-time-specific data to drive operational improvements.

In chapter 3, we proposed a new state-variable model for a stochastic weekly operating room scheduling problem with an exponential number of scenarios. We first proved the validity of the proposed model. The main advantage of the offered model was that it maintained a pseudo-polynomial number of variables and constraints that were significantly fewer than the number of variables and constraints in an equivalent scenario-based stochastic programming model. In addition, we further enhanced the model to involve fewer variables and constraints and added several valid inequalities, including worst-case scenario and symmetry-breaking cuts to strengthen the model. We evaluated the performance of the proposed models against the classic scenario-based stochastic programming formulation in several small, medium, and large size instances.

The results demonstrated the superior efficacy of the proposed state-variable model over traditional approaches. Our model could optimally address up to 20 patient instances, which was significantly more than the classic model's benchmark of 14. Our evaluations extended to scenarios involving up to 90 patients in which an average optimality gap of 1.42% was reported. The model's formulation distinguished by a significant reduction in the number of variables and constraints further enhanced computational efficiency. Comparative evaluations with the enhanced version that incorporated worst-case scenario and symmetry-breaking cuts indicated substantial improvements

in model performance. These improvements enabled us to find optimal solutions for complex instances involving 80 with an optimality gap of 0.78% 1.48E+64 scenarios.

In Chapters 4, 5, and 6 we proposed a novel hybrid reinforcement learning based algorithm, as well as two other evolutionary computation-based algorithms to solve complex problems. We applied our algorithms to several complex benchmarks in the literature and through in-depth computational experiments we showed that our algorithm outperforms other state-of-the-art methods. We applied our algorithms to solve SIDARTHE model on real-data from Canada, Ontario, Quebec, and the U.S. to model the COVID-19 pandemic's progression. We have provided a detailed analysis and insights into the transmission dynamics and the impact of government-imposed measures on controlling the spread of the virus. By categorizing the pandemic into different phases, based on governmental measures and public health responses, we aimed to model and pandemic growth at each stage. For instance, for Quebec case, we have observed a significant variation in the reproduction rates (R_0) throughout the course of the pandemic. Initially, with low transmission rates and a reproduction rate of $R0 = 1.0998$, the situation seemed to be manageable. However, as the pandemic progressed a sharp increases in the reproduction rate were observed peaking at $R_0 = 4.6096$ before strict measures helped to decrease it substantially to $R_0 = 0.7782$. Our hybrid reinforcement learning-based algorithm has not only accurately reflected these trends but has also projected the future trajectory of the pandemic under various scenarios. Through our analysis, we revealed the critical importance of maintaining and adapting social distancing measures combined with enhanced testing efforts for asymptomatic cases to control the spread of COVID-19 virus. Our study provided valuable insights for policymakers and healthcare administrators to combat the pandemic. The sensitivity analyses conducted revealed the significant influence of social behavior and governmental policies on the pandemic's trajectory. The parameter adjustments in our model demonstrated how varying levels of social distancing, business closures, and the reopening of businesses could potentially alter the course of the pandemic.

Appendix A

Proof of Theorem 1

To establish the validity of Model (M1), we need to confirm the accuracy of the following statements:

(1) Variables w_{it_i} and $z_{if_it_{i+1}}$ are well-defined by the proposed model.

(2) Objective function (1) computes the expected total cost correctly.

Proof of Statement 1: Constraints (2) and (6) ensure the well-definition of the variables w_{it_i}. The proof of consistency between the values of variables $z_{if_it_{i+1}}$ and their formal definitions is established through an induction method. We first study the case $i{=}\ 1$ as the basis of induction. In this case, constraint (3) results in $z_{000}{=}w_{10}\ =\ 1$ that coincides with the definition of z_{000} as we expect the appointment of dummy patient 0 to finish at 0 and the next patient $i\ =\ 1$ to be scheduled at the beginning of the scheduling horizon. We note that we have $w_{10}\ =\ 1$ based on constraint (2). As the inductive step, we suppose that for a fixed $i{-}1$, the model has computed all variables $z_{(i-1)f_{i-1}t_i}\ \forall f_{i-1}\ \in\ F_{i-1}, t_i\ \in\ T_i$ properly. In the following, supposing that $\tilde{y}_i$ is a random variable representing the finish time of appointment for patient i, we first show that all variables $z_{if_it_{i+1}}\ \forall f_i \in F_i\backslash\{f^*\}, t_{i+1} \in T_{i+1}$ are well-defined and then we discuss how it can be justified for $f_i = f^*, t_{i+1} \in T_{i+1} = \{.\}$

$$
\begin{aligned}
z_{if_it_{i+1}} &= \Pr\left(\left(w_{(i+1)t_{(i+1)}} = 1\right) \wedge (\tilde{y}_i = f_i)\right) \\
&= \Pr\left(w_{(i+1)t_{(i+1)}} = 1\right) \Pr\left(\tilde{y}_i = f_i\right) \\
&= w_{(i+1)t_{(i+1)}} \Pr\left(\tilde{y}_i = f_i\right) \\
&= w_{(i+1)t_{(i+1)}} \sum_{t_i \in T_i} \Pr\left(w_{it_i} = 1\right) \Pr\left(\tilde{y}_i = f_i \mid w_{it_i} = 1\right). \qquad \text{(A.1)}
\end{aligned}
$$

In the recent equations, we used the assumption that the two events $\left(w_{(i+1)t_{(i+1)}} = 1\right)$ and $(\tilde{y}_i = f_i)$ are independent. This assumption holds because $w_{(i+1)t_{(i+1)}}$ is not subject to uncertainty and is determined by the decision maker's choice. According to the latest formula, if $w_{(i+1)t_{(i+1)}} = 0$ then it follows that $z_{if_it_{i+1}} = 0$ which aligns with constraint (3). Consequently, we can exclude

$w_{(i+1)t_{(i+1)}}$ from the final expression and proceed as below:

$$z_{if_it_{i+1}} = \sum_{t_i \in T_i} \Pr\left(w_{it_i} = 1\right) \Pr\left(\tilde{y}_i = f_i \mid w_{it_i} = 1\right) \tag{A.2}$$

$$= \sum_{t_i \in T_i} w_{it_i} \left[\sum_{f_{i-1} \in F_{i-1}} \Pr\left(\tilde{y}_{(i-1)} = f_{(i-1)}\right) \Pr\left(\tilde{y}_i = f_i \mid (w_{it_i} = 1) \wedge \left(\tilde{y}_{(i-1)} = f_{(i-1)}\right)\right) \right] \tag{A.3}$$

$$= \sum_{t_i \in T_i} \left[\sum_{f_{i-1} \in F_{i-1}} w_{it_i} \Pr\left(\tilde{y}_{(i-1)} = f_{(i-1)}\right) \Pr\left(\tilde{y}_i = f_i \mid (w_{it_i} = 1) \wedge \left(\tilde{y}_{(i-1)} = f_{(i-1)}\right)\right) \right] \tag{A.4}$$

$$= \sum_{t_i \in T_i} \left[\sum_{f_{i-1} \in F_{i-1}} w_{it_i} \Pr\left(\tilde{y}_{(i-1)} = f_{(i-1)}\right) \Pr\left(\max(t_i, f_{i-1}) + \tilde{d}_i = f_i\right) \right] \tag{A.5}$$

$$= \sum_{t_i \in T_i} \left[\sum_{f_{i-1} \in F_{i-1}} \Pr\left(w_{it_i} = 1\right) \Pr\left(\tilde{y}_{(i-1)} = f_{(i-1)}\right) \Pr\left(\max(t_i, f_{i-1}) + \tilde{d}_i = f_i\right) \right] \tag{A.6}$$

$$= \sum_{t_i \in T_i} \left[\sum_{f_{i-1} \in F_{i-1}} \Pr\left(\left(w_{it_i} = 1\right) \wedge \left(\tilde{y}_{(i-1)} = f_{(i-1)}\right)\right) \Pr\left(\max(t_i, f_{i-1}) + \tilde{d}_i = f_i\right) \right] \tag{A.7}$$

$$= \sum_{t_i \in T_i} \left[\sum_{f_{i-1} \in F_{i-1}} z_{(i-1)f_{i-1}t_i} \Pr\left(\max(t_i, f_{i-1}) + \tilde{d}_i = f_i\right) \right] \tag{A.8}$$

$$= \sum_{t_i \in T_i} \sum_{f_{i-1} \in F_{i-1}} z_{(i-1)f_{i-1}t_i} \Pr\left(\max(t_i, f_{i-1}) + \tilde{d}_i = f_i\right) \tag{A.9}$$

Also, relation A.6 is valid as $w_{it_i} = \Pr\left(w_{it_i} = 1\right)$ holds, given that w_{it_i} is a binary variable determined by the decision maker and lacks any uncertain nature. Relation A.7 is valid because the events $w_{it_i} = 1$ and $\Pr\left(\tilde{y}_{(i-1)} = f_{(i-1)}\right)$ are independent. Finally, the validity of relation A.8 aligns with the definition of $z_{(i-1)f_{i-1}t_i}$ (the basis of induction), ensuring that constraints (3)-(4) correctly define variables $z_{if_it_{i+1}}$ for $f_i \in F_i \backslash \{f^*\}, t_{i+1} \in T_{i+1}$. To complete the proof of validity of Statement 1, it remains to demonstrate that variables $z_{if_it_{i+1}}$ for $f_i = f^*, t_{i+1} \in T_{i+1} = \{.\}$

are well-defined. This can be established through a process similar to that outlined in relations A.2-A.9 with the sole distinction being the substitution of $f_i = f^*$ and $t_{i+1} =$., and also replacing $\Pr\left(\max(t_i, f_{i-1}) + \tilde{d}_i = f_i\right)$ with $\Pr(\max(t_i, f_{i-1}) + \tilde{d}_i \geq f_i)$. This completes the proof for the validity of Statement 1.

Proof of Statement 2: To verify that the objective function is accurately computed, we focus on the representation of random cost variables for patient i including $\widetilde{\mathrm{c}}_i^{waiting}$, $\widetilde{\mathrm{c}}_i^{idle}$, and $\widetilde{\mathrm{c}}_i^{overtime}$. These variables are contingent upon the scheduled appointment times and the uncertain parameters of service time.

$$
\begin{aligned}
\text{Total expected cost} &= \sum_{i \in I_0} E\left(\tilde{c}_i^{waiting} + \tilde{c}_i^{idle} + \tilde{c}_i^{overtime} + \tilde{c}_i^{cancellation}\right) \\
&= \sum_{i \in I_0} \left(E\left[\tilde{c}_i^{waiting}\right] + E\left[\tilde{c}_i^{idle}\right] + E\left[\tilde{c}_i^{overtime}\right] + E\left[\tilde{c}_i^{cancellation}\right]\right) \\
&= \sum_{i \in I_0} \sum_{f_i \in F_i} \sum_{t_{i+1} \in T_{i+1}} \Pr\left((\tilde{y}_i = f_i) \wedge \left(w_{(i+1)(t_{i+1})} = 1\right)\right) \\
&\quad \times \Big(E\left[\tilde{c}_i^{waiting} \mid (\tilde{y}_i = f_i) \wedge \left(w_{(i+1)(t_{i+1})} = 1\right)\right] + \\
&\quad E\left[\tilde{c}_i^{idle} \mid (\tilde{y}_i = f_i) \wedge \left(w_{(i+1)(t_{i+1})} = 1\right)\right] + \\
&\quad E\left[\tilde{c}_i^{overtime} \mid (\tilde{y}_i = f_i) \wedge \left(w_{(i+1)(t_{i+1})} = 1\right)\right] + \\
&\quad E\left[\tilde{c}_i^{cancellation} \mid (\tilde{y}_i = f^*) \wedge \left(w_{(i+1)(t_{i+1})} = 1\right)\right]\Big) \\
&= \sum_{i \in I_0} \sum_{f_i \in F_i} \sum_{t_{i+1} \in T_{i+1}} z_{i f_i t_{i+1}} \\
&\quad \times \left(c^p_{i f_i t_{i+1}} + c^d_{i f_i t_{i+1}} + c^o_{i f_i t_{i+1}} + \alpha^c \mathbb{1}_{(f_i = f^*)}\right)
\end{aligned} \tag{A.10}
$$

Therefore, Statement 2 holds too and Theorem 1 is valid.

Appendix B

Appointment Scheduling Example

Let us consider a practical example. This example involves two patients both of whom have already been assigned their appointment slots. Thus, Constraint (2) regarding single appointment assignments per patient is satisfied and we have $w_{10}= 1$, $w_{2(30)}= 1$. The service times for Patient 1 are (20, 40) minutes with corresponding probabilities of (0.5, 0.5), and for Patient 2 we have service times of (20, 30) minutes and corresponding probabilities of (0.5, 0.5). The model captures the probability of different finish times for each patient considering their respective service time distributions. This information is presented in Table B.1 which outlines the probability of each patient finishing their appointment at various times.

Table B.1: Probability of finish times for each patient

Line Number	Patient	Finish time	Probability
1	0	0	1
2	1	20	0.5
3		40	0.5
4	2	50	0.25
5		60	0.5
6		70	0.25

First, we schedule Patient 1 in the first appointment slot. Following Constraint (3) we derive the following set of constraints ($\theta = 10$):

$$z_{000} = 1 \tag{B.1}$$

$$z_{00(t_1)} = 0 \quad \text{for } t_1 \in T_1 \backslash \{0\} \tag{B.2}$$

which validates the first line of Table 1.

Considering Constraint (4), we also derive the following constraint:

$$\begin{aligned} z_{000} \Pr(\max(0,0) + \tilde{d}_1 = 20) = {} & z_{1(20)(20)} + z_{1(20)(30)} + z_{1(20)(40)} + z_{1(20)(50)} \\ & + z_{1(20)(60)} + z_{1(20)(70)} + z_{1(20)(80)} + z_{1(20)(90)}. \end{aligned} \tag{B.3}$$

Upon simplification, we obtain:

$$\begin{aligned} 0.5 = {} & z_{1(20)(20)} + z_{1(20)(30)} + z_{1(20)(40)} + z_{1(20)(50)} \\ & + z_{1(20)(60)} + z_{1(20)(70)} + z_{1(20)(80)} + z_{1(20)(90)} \end{aligned} \tag{B.4}$$

From Constraint (3) for Patient 2, we have: Therefore, from constraints, we have:

$$\sum_{f_1 \in F_1} z_{1 f_1(t_1)} = 0 \quad \text{for } t_1 \in T_1 \setminus \{30\} \tag{B.5}$$

Thus, from (B.5) and (B.4) we have:

$$0.5 = z_{1(20)(30)} \tag{B.6}$$

which validates the second line of Table 1.

Considering Constraint (4) for Patient 1 and $f = 40$, we have:

$$\begin{aligned} z_{000} \Pr(\max(0,0) + \tilde{d}_1 = 40) = {} & z_{1(40)(20)} + z_{1(40)(30)} + z_{1(40)(40)} + z_{1(40)(50)} \\ & + z_{1(40)(60)} + z_{1(40)(70)} + z_{1(40)(80)} + z_{1(40)(90)}. \end{aligned} \tag{B.7}$$

Upon simplification, we obtain:

$$0.5 = z_{1(40)(20)} + z_{1(40)(30)} + z_{1(40)(40)} + z_{1(40)(50)} + z_{1(40)(60)} + z_{1(40)(70)} + z_{1(40)(80)} + z_{1(40)(90)}. \tag{B.8}$$

Also, by considering Constraint (3) for Patient 2, we have $\sum_{f_1 \in F_1} z_{1f_1(t_1)} = 0$ for $t_1 \in T_1 \setminus \{30\}$. Now, based on (B.5) and (B.8), we can show that:

$$0.5 = z_{1(40)(30)} \tag{B.9}$$

This validates the third line of Table B.1. Furthermore, considering Constraint (4) for Patient 2 and $f = 50$, we derive:

$$z_{1(20)(30)} \Pr(\max(30, 20) + \tilde{d}_2 = 50) + \sum_{t_i \in T_i \setminus \{30\}} z_{1(20)(t_i)} \Pr(\max(t_i, 20) + \tilde{d}_2 = 50) + \sum_{t_i \in T_i} z_{1(40)(t_i)} \Pr(\max(t_i, 40) + \tilde{d}_2 = 50) = z_{2(50)(\cdot)} \tag{B.10}$$

In Constraint (B.10), the expression $\sum_{t_i \in T_i \setminus 30} z_{1(20)(t_i)} \Pr(\max(t_i, 20) + \tilde{d}2 = 50)$ is equal to zero considering (B.5). The expression $\sum_{t_i \in T_i} z_{1(40)(t_i)} \Pr(\max(t_i, 40) + \tilde{d}_2 = 50)$ is equal to zero since $\max(t_i, 40) + \tilde{d}_2 = 50$ is not possible. Therefore, we have:

$$0.5 \cdot 0.5 = z_{2(50)(.)} \tag{B.11}$$

This validates the fourth line of Table B.1. Further, considering Constraint (B.4) for Patient 2 and $f = 60$ we have:

$$z_{1(20)(30)} \Pr(\max(30,20) + \tilde{d}_2 = 60) + \sum_{t_i \in T_i \setminus \{30\}} z_{1(20)(t_i)} \Pr(\max(t_i,20) + \tilde{d}_2 = 60)$$
$$+ z_{1(40)(30)} \Pr(\max(30,40) + \tilde{d}_2 = 60)$$
$$+ \sum_{t_i \in T_i \setminus \{30\}} z_{1(40)(t_i)} \Pr(\max(t_i,40) + \tilde{d}_2 = 60) = z_{2(60)(.)} \quad \text{(B.12)}$$

Given (B.12), expressions $\sum_{t_i \in T_i \setminus \{30\}} z_{1(20)(t_i)} \Pr(\max(t_i,20) + \tilde{d}_2 = 60)$ and $\sum_{t_i \in T_i \setminus \{30\}} z_{1(40)(t_i)} \Pr(\max(t_i,40) + \tilde{d}_2 = 60)$ are equal to zero considering (B.12). Thus, we have:

$$0.5 \cdot 0.5 + 0.5 \cdot 0.5 = z_{2(60)(.)} \quad \text{(B.13)}$$

which validates the fifth line of Table B.1. Considering Constraint (4) for Patient 2 and $f = 70$ we have:

$$\sum_{t_i \in T_i \setminus \{30\}} z_{1(20)(t_i)} \Pr(\max(t_i,20) + \tilde{d}_2 = 70) + z_{1(20)(30)} \Pr(\max(30,20) + \tilde{d}_2 = 70)$$
$$+ z_{1(40)(30)} \Pr(\max(30,40) + \tilde{d}_2 = 70) + \sum_{t_i \in T_i \setminus \{30\}} z_{1(40)(t_i)} \Pr(\max(t_i,40) + \tilde{d}_2 = 70)$$
$$+ \sum_{t_i \in T_i} z_{1(f \neq 40)(t_i)} \Pr(\max(t_i, f \neq 40) + \tilde{d}_2 = 70) = z_{2(50)(.)} \quad \text{(B.14)}$$

In above equality, $\sum_{t_i \in T_{i \setminus 30}} z_{1(20)(t_i)} \Pr(\max(t_i,20) + \tilde{d}_2 = 70)$ is equal to zero considering (B.5). In addition, $z_{1(20)(30)} \Pr(\max(30,20) + \tilde{d}2 = 70)$ is equal to zero since $\max(30,20) + \tilde{d}2 = 70$ is not possible. Moreover, $z_{1(40)(30)} \Pr(\max(30,40) + \tilde{d}2 = 70)$ is equal to zero considering (B.5). Finally, $\sum_{t_i \in Ti \setminus 30} z_{1(40)(t_i)} \Pr(\max(t_i,40) + \tilde{d}_2 = 70)$ is equal to zero since constraint (3) for Patient 2 shows $z_{1(20)(30)} + z_{1(40)(30)} + \sum_{t_i \in T_i} z_{1(f \neq 20)(30)} = 1$. Which together with $0.5 = z_{1(20)(30)}$ and $0.5 = z_{1(20)(40)}$ proves $\sum_{t_i \in T_i} z_{1(f \neq 20)(30)} = 0$. Thus we have:

$$0.5 = z_{2(50)(.)} \tag{B.15}$$

which validates the last line of Table B.1.

Appendix C

Proof of Theorem 2

To prove the validity of Model (28)-(35), by induction we show that Model (28)-(35) computes the values of variables z^0_{drit} and z^1_{drit} correctly. We first consider the case $i = 0$ as the basis of induction. In constraint (30)-(31), we have $\mathcal{T}_0 = \{0\}$. Therefore, we can rewrite these two constraints for $i = 0$ as follows.

$$z^1_{dr00} = x_{dr1} \qquad d \in \mathcal{D}, r \in \mathcal{R}_d, i \in \mathcal{I}^0 \backslash |\mathcal{I}| \qquad \text{(C.1)}$$

$$z^0_{dr00} = 1 - x_{dr1} \qquad d \in \mathcal{D}, r \in \mathcal{R}_d, i \in \mathcal{I}^0 \backslash |\mathcal{I}| \qquad \text{(C.2)}$$

Constraints (C.1) and (C.2) show that for $i = 0$, z^0_{drit} and z^1_{drit} are defined properly. Now, let us suppose that variables $z^0_{dr(i-1)t}$ and $z^1_{dr(i-1)t}$ are already computed correctly. As the inductive step, we have to show that the model calculates the values of variables z^0_{drit} and z^1_{drit} properly.

The left-hand side of constraint (32) represents the probability that the cumulative used time in operating room r on day d by surgeries 1 to i is equal to t. Also with respect to constraint (30)-(31) it is clear that one of the variables z^0_{drit} and z^1_{drit} will be equal to 0. Therefore, only the other variable appears on the left-hand side of constraint (32). Consider the following notation.

A_{drit} : The event that the cumulative used time in operating room r on day d by surgeries 1 to i is equal to t.

$Pr(.)$: Probability of event (.).

To complete the proof, we have to show that $Pr(A_{drit})$ is equal to right-hand side of constraint

(32).

$$
\begin{aligned}
Pr(A_{drit}) &= \sum_{\substack{t' \in \mathcal{T}_{i-1}: \\ t' \leq t}} Pr(A_{dr(i-1)t'}) Pr(A_{drit} | A_{dr(i-1)t'}) \\
&= \sum_{\substack{t' \in \mathcal{T}_{i-1}: \\ t' \leq t}} Pr(A_{dr(i-1)t'}) \Big[Pr(x_{dri} = 0) Pr(A_{drit} | A_{dr(i-1)t'} \wedge x_{dri} = 0) \\
&\quad + Pr(x_{dri} = 1) Pr(A_{drit} | A_{dr(i-1)t'} \wedge x_{dri} = 1) \Big] \qquad \text{(C.3)}
\end{aligned}
$$

It is clear that $Pr(A_{drit} | A_{dr(i-1)t'} \wedge x_{dri} = 0) = 1$ holds for $t' = t$ if $t \in \mathcal{T}_{(i-1)}$. For other values of t' we have $Pr(A_{drit} | A_{dr(i-1)t'} \wedge x_{dri} = 0) = 0$. Also $Pr(A_{drit} | A_{dr(i-1)t'} \wedge x_{dri} = 1) = Pr(t' + t_{i\omega} = t)$ holds. Considering these points, we can rewrite the above relation as follows.

$$
\begin{aligned}
Pr(A_{drit}) &= \mathbb{1}_{(t \in \mathcal{T}_{(i-1)})} Pr(A_{dr(i-1)t}) Pr(x_{dri} = 0) \\
&\quad + \sum_{\substack{t' \in \mathcal{T}_{i-1}: \\ t' \leq t}} \Big[Pr(x_{dri} = 1) Pr(A_{dr(i-1)t'}) Pr(t' + t_{i\omega} = t) \Big] \\
&= \mathbb{1}_{(t \in \mathcal{T}_{(i-1)})} Pr(A_{dr(i-1)t} \wedge x_{dri} = 0) \\
&\quad + \sum_{\substack{t' \in \mathcal{T}_{i-1}: \\ t' \leq t}} \Big[Pr(A_{dr(i-1)t'} \wedge x_{dri} = 1) Pr(t' + t_{i\omega} = t) \Big] \\
&= \mathbb{1}_{(t \in \mathcal{T}_{(i-1)})} z^0_{dr(i-1)t} + \sum_{\substack{t' \in \mathcal{T}_{i-1}: \\ t' \leq t}} z^1_{dr(i-1)t'} Pr(t' + t_{i\omega} = t) \qquad \text{(C.4)}
\end{aligned}
$$

The last equality is valid with respect to the definition of $z^0_{dr(i-1)t}$ and $z^1_{dr(i-1)t'}$ variables. We observe that the right-hand side of constraint (32) is equal to $Pr(A_{drit})$ and therefore the proof is complete.

Appendix D

Benchmark Functions

Table D.1: The unimodal benchmark functions.

Function	Range	$f_{\min}$
$f_1(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]^{10}	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10] ^{10}	0
$f_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	[-100,100] ^{10}	0
$f_4(x) = \max_i \{\lvert x_i \rvert, 1 \leq i \leq n\}$	[-100,100] ^{10}	0
$f_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	[-30,30] ^{10}	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	[-100,100] ^{10}	0
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0, 1)$	[-1.28,1.28] ^{10}	0
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	[-500,500] ^{10}	-418.9829×5
$F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	[-5.12,5.12] ^{10}	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	[-32,32] ^{10}	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600] ^{10}	0
$F_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}\{(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})]$ $+(y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	[-50,50] ^{10}	0
$y_i = 1 + \frac{x_i+1}{4}$ $\quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m x_i < a \end{cases}$		
$F_{13}(x) = 0.1\{\sin^2(3\pi x_i) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)]$ $+(x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	[-50,50] ^{10}	0

Table D.2: The unimodal benchmark functions.

Function	Range	$f_{\min}$
$F_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	[-65,65] ^{2}	1
$F_{15}(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}]^2$	[-5,5] ^{4}	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 = \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	[-5,5] ^{2}	-1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	[-5,5] ^{2}	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	[-2,2] ^{2}	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	[1,3] ^{3}	-3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	[0,1] ^{6}	-3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10] ^{4}	-10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10] ^{4}	-10.4028
$F_{22}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10] ^{4}	-10.5363

Appendix E

SIDARTHE model

In this appendix, we describe the SIDARTHE model proposed by Giordano et al. (2020). The model considers eight states for people, including Susceptible, Infected, Diagnosed, Ailing, Recognized, Threatened, Healed, Extinct and provide the following definitions (see Figure E.1):
1) Susceptible: uninfected case, 2) Infected: undetected asymptomatic or pauci-symptomatic case, 3) Diagnosed: detected asymptomatic case, 4) Ailing: undetected symptomatic case, 5) Recognized: detected symptomatic case, 6) Threatened: detected with life-threatening symptoms, 7) Healed: recovered, 8) Extinct: dead. We use the notation presented below to represent the model.

Indices	
t	State index
Parameters	
α	Transmission rate due to contact of a susceptible case with an infected case.
β	Transmission rate due to contact of a susceptible case with a diagnosed case.
δ	Transmission rate due to contact of a susceptible case with an ailing case.
γ	Transmission rate due to contact of a susceptible case with a recognized case.
ε	The rate of detecting asymptomatic cases.
ζ	The probability that an infected case is aware of being infected.
	The probability that an infected case is unaware of being infected. η
θ	The detection rate of symptomatic cases.
$\lambda, \kappa, \xi, \rho\ , \sigma$	The recovery rate of the five categories of infected cases.
μ	The probability that an undetected/detected infected case shows life-threatening symptoms.
υ	The probability that a detected infected case develops life-threatening symptoms.
τ	Mortality rate.
State variables	
$S(t)$	The fraction of susceptible (not infected) cases in the population.
$I(t)$	The fraction of infected (infected and undetected cases without symptoms) cases in the population.
$D(t)$	Fraction of diagnosed (infected and detected cases without symptoms) cases in the population.
$A(t)$	The fraction of ailing (infected and undetected cases with symptoms) cases in the population.
$R(t)$	The fraction of recognized (infected and detected cases with symptoms) cases in the population.
$T(t)$	The fraction of threatened (infected detected cases that developed life-threatening symptoms) cases in the population.
$H(t)$	The fraction of recovered cases in the population.
$E(t)$	The fraction of death cases in the population.

Based on the notations, Giordano et al. (2020) represented the following model:

$$\dot{S}(t) = -S(t)(\alpha I(t) + \beta D(t) + \gamma A(t) + \delta R(t)) \tag{E.1}$$

$$\dot{I}(t) = S(t)(\alpha I(t) + \beta D(t) + \gamma A(t) + \delta R(t)) - (\epsilon + \zeta + \lambda)I(t) \tag{E.2}$$

$$\dot{D}(t) = \epsilon I(t) - (\eta + \rho)D(t) \tag{E.3}$$

$$\dot{A}(t) = \zeta I(t) - (\theta + \mu + \kappa)A(t) \tag{E.4}$$

$$\dot{R}(t) = \eta D(t) + \theta A(t) - (\upsilon + \xi)R(t) \tag{E.5}$$

$$\dot{T}(t) = \mu A(t) + \upsilon R(t) - (\sigma + \tau)T(t) \tag{E.6}$$

$$\dot{H}(t) = \lambda I(t) + \rho D(t) + \kappa A(t) + \xi R(t) + \sigma T(t) \tag{E.7}$$

$$\dot{E}(t) = \tau T(t) \tag{E.8}$$

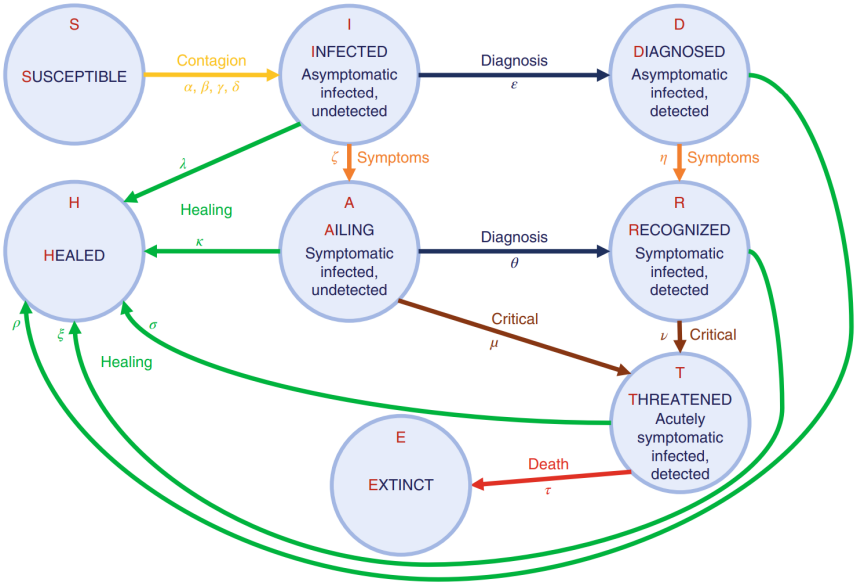

Figure E.1: "Graphical scheme representing the interactions among different stages of infection in the mathematical model SIDARTHE: S, susceptible (uninfected); I, infected (asymptomatic or pauci-symptomatic infected, undetected); D, diagnosed (asymptomatic infected, detected); A, ailing (symptomatic infected, undetected); R, recognized (symptomatic infected, detected); T, threatened (infected with life-threatening symptoms, detected); H, healed (recovered); E, extinct (dead)." (Giordano et al., 2020)

Bibliography

(2020). Coronavirus disease (covid-19) in québec. https://www.quebec.ca/en/health/health-issues/a-z/2019-coronavirus/.

Abdel-Basset, M., El-Shahat, D., El-Henawy, I., De Albuquerque, V. H. C., and Mirjalili, S. (2020). A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Systems with Applications*, 139:112824.

Abebe, T. H. (2020). Forecasting the number of coronavirus (covid-19) cases in ethiopia using exponential smoothing times series model. *medRxiv*, pages 2020–06.

Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M., and Vrahatis, M. N. (2019). No free lunch theorem: A review. *Approximation and optimization: Algorithms, complexity and applications*, pages 57–82.

Addis, B., Carello, G., Grosso, A., and Tànfani, E. (2016). Operating room scheduling and rescheduling: a rolling horizon approach. *Flexible Services and Manufacturing Journal*, 28(1-2):206–232.

Ahamad, M. M., Aktar, S., Rashed-Al-Mahfuz, M., Uddin, S., Liò, P., Xu, H., Summers, M. A., Quinn, J. M., and Moni, M. A. (2020). A machine learning model to identify early stage symptoms of sars-cov-2 infected patients. *Expert systems with applications*, 160:113661.

Ahmadianfar, I., Bozorg-Haddad, O., and Chu, X. (2020). Gradient-based optimizer: A new metaheuristic optimization algorithm. *Information Sciences*, 540:131–159.

Akhtar, E. S. F. (2017). *Practical Reinforcement Learning: Develop self-evolving, intelligent agents with OpenAI Gym, Python and Java*. Packt publishing.

Alakus, T. B. and Turkoglu, I. (2020). Comparison of deep learning approaches to predict covid-19 infection. *Chaos, Solitons & Fractals*, 140:110120.

Alamo, T., Reina, D. G., and Millán, P. (2020). Data-driven methods to monitor, model, forecast and control covid-19 pandemic: Leveraging data science, epidemiology and control theory. *arXiv preprint arXiv:2006.01731*.

Ardabili, S. F., Mosavi, A., Ghamisi, P., Ferdinand, F., Varkonyi-Koczy, A. R., Reuter, U., Rabczuk, T., and Atkinson, P. M. (2020). Covid-19 outbreak prediction with machine learning. *Algorithms*, 13(10):249.

Arora, P., Kumar, H., and Panigrahi, B. K. (2020). Prediction and analysis of covid-19 positive cases using deep learning models: A descriptive case study of india. *Chaos, solitons & fractals*, 139:110017.

Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & structures*, 169:1–12.

Atighehchian, A., Sepehri, M. M., Shadpour, P., and Kianfar, K. (2020). A two-step stochastic approach for operating rooms scheduling in multi-resource environment. *Annals of Operations Research*, 292(1):191–214.

Bandi, C. and Gupta, D. (2020). Operating room staffing and scheduling. *Manufacturing & Service Operations Management*, 22(5):958–974.

Bansal, J. C. and Singh, S. (2021). A better exploration strategy in grey wolf optimizer. *Journal of Ambient Intelligence and Humanized Computing*, 12:1099–1118.

Belen, S., Kropat, E., and Weber, G.-W. (2011). On the classical maki–thompson rumour model in continuous time. *Central European Journal of Operations Research*, 19:1–17.

Bennett, K. J. and Baxley, E. G. (2009). The effect of a carve-out advanced access scheduling system on no-show rates. *Family medicine*, 41(1):51.

Bertsekas, D. (2019). *Reinforcement learning and optimal control*. Athena Scientific.

Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3):268–308.

Bouhamed, H. (2020). Covid-19 cases and recovery previsions with deep learning nested sequence prediction models with long short-term memory (lstm) architecture. *Int. J. Sci. Res. in Computer Science and Engineering*, 8(2).

Breuer, D. J., Lahrichi, N., Clark, D. E., and Benneyan, J. C. (2020). Robust combined operating room planning and personnel scheduling under uncertainty. *Operations research for health care*, 27:100276.

Cardoen, B., Demeulemeester, E., and Beliën, J. (2010). Operating room planning and scheduling: A literature review. *European journal of operational research*, 201(3):921–932.

Cayirli, T., Veral, E., and Rosen, H. (2008). Assessment of patient classification in appointment system design. *Production and Operations Management*, 17(3):338–353.

Cayirli, T., Yang, K. K., and Quek, S. A. (2012). A universal appointment rule in the presence of no-shows and walk-ins. *Production and Operations Management*, 21(4):682–697.

Chakrabarti, B. K., Chakraborti, A., and Chatterjee, A. (2006). Econophysics and sociophysics: trends and perspectives.

Chimmula, V. K. R. and Zhang, L. (2020). Time series forecasting of covid-19 transmission in canada using lstm networks. *Chaos, solitons & fractals*, 135:109864.

Choi, S. and Wilhelm, W. E. (2014). On capacity allocation for operating rooms. *Computers & Operations Research*, 44:174–184.

Cui, L., Hu, H., Yu, S., Yan, Q., Ming, Z., Wen, Z., and Lu, N. (2018). Ddse: A novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks. *Journal of Network and Computer Applications*, 103:119–130.

Daggy, J., Lawley, M., Willis, D., Thayer, D., Suelzer, C., DeLaurentis, P.-C., Turkcan, A., Chakraborty, S., and Sands, L. (2010). Using no-show modeling to improve clinic performance. *Health informatics journal*, 16(4):246–259.

Dai, T. and Tayur, S. (2020). Om forum—healthcare operations management: A snapshot of emerging research. *Manufacturing & Service Operations Management*, 22(5):869–887.

Denton, B. T., Miller, A. J., Balasubramanian, H. J., and Huschka, T. R. (2010). Optimal allocation of surgery blocks to operating rooms under uncertainty. *Operations research*, 58(4-part-1):802–816.

Dewangan, R. K., Shukla, A., and Godfrey, W. W. (2019). Three dimensional path planning using grey wolf optimizer for uavs. *Applied Intelligence*, 49:2201–2217.

Dhargupta, S., Ghosh, M., Mirjalili, S., and Sarkar, R. (2020). Selective opposition based grey wolf optimization. *Expert Systems with Applications*, 151:113389.

Diamant, A., Milner, J., and Quereshy, F. (2018). Dynamic patient scheduling for multi-appointment health care programs. *Production and Operations Management*, 27(1):58–79.

Doulabi, H. H., Ahmed, S., and Nemhauser, G. (2022). State-variable modeling for a class of two-stage stochastic optimization problems. *INFORMS Journal on Computing*, 34(1):354–369.

Du, H., Wu, X., and Zhuang, J. (2006). Small-world optimization algorithm for function optimization. In *Advances in Natural Computation: Second International Conference, ICNC 2006, Xi'an, China, September 24-28, 2006. Proceedings, Part II 2*, pages 264–273. Springer.

Eberhart, R. and Kennedy, J. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, volume 4, pages 1942–1948.

Erdogan, S. A. and Denton, B. (2013). Dynamic appointment scheduling of a stochastic server with uncertain demand. *Informs journal on computing*, 25(1):116–132.

Erekat, A., Servis, G., Madathil, S. C., and Khasawneh, M. T. (2020). Efficient operating room planning using an ensemble learning approach to predict surgery cancellations. *IISE Transactions on Healthcare Systems Engineering*, 10(1):18–32.

Eskandar, H., Sadollah, A., Bahreininejad, A., and Hamdi, M. (2012). Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 110:151–166.

Fahmy, G. and Ross, S. (2020). Montreal ers are near capacity with delayed health problems, a worrying sign as second wave looms. `https://montreal.ctvnews.ca/montreal-ers-are-near-capacity-with-delayed-health-problems-a-worrying-sign-5003710?cache=?clipId=68597`.

Faris, H., Aljarah, I., Al-Betar, M. A., and Mirjalili, S. (2018). Grey wolf optimizer: a review of recent variants and applications. *Neural computing and applications*, 30:413–435.

Fei, H., Chu, C., and Meskens, N. (2009). Solving a tactical operating room planning problem by a column-generation-based heuristic procedure with four criteria. *Annals of Operations Research*, 166(1):91.

Fei, H., Chu, C., Meskens, N., and Artiba, A. (2008). Solving surgical cases assignment problem by a branch-and-price approach. *International Journal of Production Economics*, 112(1):96–108.

Feldman, J., Liu, N., Topaloglu, H., and Ziya, S. (2014). Appointment scheduling under patient preference and no-show behavior. *Operations Research*, 62(4):794–811.

Fogel, D. B. (1998). *Artificial intelligence through simulated evolution*. Wiley-IEEE Press.

Formato, R. (2007). Central force optimization: a new metaheuristic with applications in applied electromagnetics. prog electromagn res 77: 425–491.

Freeman, N. K., Melouk, S. H., and Mittenthal, J. (2015). A scenario-based approach for operating theater scheduling under uncertainty. *Manufacturing & Service Operations Management*, 18(2):245–261.

Garcia, L. P., Goncalves, A. V., de Andrade, M. P., Pedebos, L. A., Vidor, A. C., Zaina, R., de Luca Canto, G., de Araujo, G. M., and Amaral, F. V. (2020). Estimating underdiagnosis of covid-19 with nowcasting and machine learning: Experience from brazil (preprint).

Giordano, G., Blanchini, F., Bruno, R., Colaneri, P., Di Filippo, A., Di Matteo, A., and Colaneri, M. (2020). Modelling the covid-19 epidemic and implementation of population-wide interventions in italy. *Nature medicine*, 26(6):855–860.

Golafshani, E. M., Behnood, A., and Arashpour, M. (2020). Predicting the compressive strength of normal and high-performance concretes using ann and anfis hybridized with grey wolf optimizer. *Construction and Building Materials*, 232:117266.

Gouvernement du Québec (n.d.). Coronavirus disease (covid-19) in québec. `https://www.quebec.ca/en/health/health-issues/a-z/2019-coronavirus`. Accessed: [Insert current date here].

Government of Canada (2023). Covid-19 pandemic in canada. `https://health-infobase.canada.ca/covid-19/`.

Guerriero, F. and Guido, R. (2011). Operational research in the management of the operating theatre: a survey. *Health care management science*, 14(1):89–114.

Guo, C., Bodur, M., Aleman, D. M., and Urbach, D. R. (2021). Logic-based benders decomposition and binary decision diagram based approaches for stochastic distributed operating room scheduling. *INFORMS Journal on Computing*.

Hans, E., Wullink, G., Van Houdenhoven, M., and Kazemier, G. (2008). Robust surgery loading. *European Journal of Operational Research*, 185(3):1038–1050.

Hashemi Doulabi, H., Ahmed, S., and Nemhauser, G. (2020). State-variable modeling for a class of two-stage stochastic optimization problems. *Accepted in INFORMS Journal on Computing, available on https://bit.ly/3hMEwTO*.

Hashemi Doulabi, H., Rousseau, L.-M., and Pesant, G. (2016). A constraint-programming-based branch-and-price-and-cut approach for operating room planning and scheduling. *INFORMS Journal on Computing*, 28(3):432–448.

Hassin, R. and Mendel, S. (2008). Scheduling arrivals to queues: A single-server model with no-shows. *Management science*, 54(3):565–572.

Higazy, M. (2020). Novel fractional order sidarthe mathematical model of covid-19 pandemic. *Chaos, Solitons & Fractals*, 138:110007.

Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1):66–73.

Hong, L. J. and Nelson, B. L. (2007). A framework for locally convergent random-search algorithms for discrete optimization via simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 17(4):19–es.

Hoursan, H., Farahmand, F., and Ahmadian, M. T. (2020). A three-dimensional statistical volume element for histology informed micromechanical modeling of brain white matter. *Annals of biomedical engineering*, 48:1337–1353.

Humanitarian Data Exchange (2020). Novel coronavirus (2019-ncov) case data. `https://data.humdata.org/dataset/novel-coronavirus-2019-ncov-case`. Accessed: [Insert current date here].

Jebali, A. and Diabat, A. (2015). A stochastic model for operating room planning under capacity constraints. *International Journal of Production Research*, 53(24):7252–7270.

Jiang, B., Tang, J., and Yan, C. (2019). A stochastic programming model for outpatient appointment scheduling considering unpunctuality. *Omega*, 82:70–82.

Jiang, R., Shen, S., and Zhang, Y. (2017). Integer programming approaches for appointment scheduling with random no-shows and service durations. *Operations research*, 65(6):1638–1656.

Kaandorp, G. C. and Koole, G. (2007). Optimal outpatient appointment scheduling. *Health care management science*, 10:217–229.

Karaboga, D. and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39:459–471.

Kavadi, D. P., Patan, R., Ramachandran, M., and Gandomi, A. H. (2020). Partial derivative nonlinear global pandemic machine learning prediction of covid 19. *Chaos, Solitons & Fractals*, 139:110056.

Kaveh, A., Seddighian, M., and Ghanadpour, E. (2020). Black hole mechanics optimization: a novel meta-heuristic algorithm. *Asian Journal of Civil Engineering*, 21:1129–1149.

Kaveh, A. and Talatahari, S. (2010). A novel heuristic optimization method: charged system search. *Acta mechanica*, 213(3-4):267–289.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. ieee.

Khalilpourazari, S. and Khalilpourazary, S. (2019). An efficient hybrid algorithm based on water cycle and moth-flame optimization algorithms for solving numerical and constrained engineering optimization problems. *Soft Computing*, 23:1699–1722.

Khalilpourazari, S. and Pasandideh, S. H. R. (2020). Sine–cosine crow search algorithm: theory and applications. *Neural Computing and Applications*, 32:7725–7742.

Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

Klassen, K. J. and Yoogalingam, R. (2009). Improving performance in outpatient appointment services with a simulation optimization approach. *Production and Operations Management*, 18(4):447–458.

Kliff, S., Satariano, A., Silver-Greenberg, J., and Kulish, N. (2020). There aren't enough ventilators to cope with the coronavirus. `https://www.nytimes.com/2020/03/18/business/coronavirus-ventilator-shortage.html`.

Kong, Q., Li, S., Liu, N., Teo, C.-P., and Yan, Z. (2020). Appointment scheduling under time-dependent patient no-show behavior. *Management Science*, 66(8):3480–3500.

Koza, J. (1992). On the programming of computers by means of natural selection. *Genetic programming*.

LaGanga, L. R. and Lawrence, S. R. (2012). Appointment overbooking in health care clinics to improve patient service and clinic performance. *Production and operations management*, 21(5):874–888.

Lalmuanawma, S., Hussain, J., and Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for covid-19 (sars-cov-2) pandemic: A review. *Chaos, Solitons & Fractals*, 139:110059.

Lamiri, M., Xie, X., Dolgui, A., and Grimaud, F. (2008). A stochastic model for operating room planning with elective and emergency demand for surgery. *European Journal of Operational Research*, 185(3):1026–1037.

Lapierre, M. (2020). Quebec's first case of covid-19 has been confirmed.

Li, M. D., Zhao, H., Weng, X. W., and Han, T. (2016). A novel nature-inspired algorithm for optimization: Virus colony search. *Advances in engineering software*, 92:65–88.

Li, Y., Tang, S. Y., Johnson, J., and Lubarsky, D. A. (2019). Individualized no-show predictions: Effect on clinic overbooking and appointment reminders. *Production and Operations Management*, 28(8):2068–2086.

Liang, J.-J., Suganthan, P. N., and Deb, K. (2005). Novel composition test functions for numerical global optimization. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 68–75. IEEE.

Liao, T., Molina, D., and Stützle, T. (2015). Performance evaluation of automatically tuned continuous optimizers on different benchmark sets. *Applied Soft Computing*, 27:490–503.

Liu, N. (2016). Optimal choice for appointment scheduling window under patient no-show behavior. *Production and Operations Management*, 25(1):128–142.

Liu, N., Ziya, S., and Kulkarni, V. G. (2010). Dynamic scheduling of outpatient appointments under patient no-shows and cancellations. *Manufacturing & Service Operations Management*, 12(2):347–364.

Long, W., Jiao, J., Liang, X., and Tang, M. (2018). An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Engineering Applications of Artificial Intelligence*, 68:63–80.

Lozano, M., Molina, D., and Herrera, F. (2011). Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems. *Soft computing*, 15:2085–2087.

Luo, J., Kulkarni, V. G., and Ziya, S. (2012). Appointment scheduling under patient no-shows and service interruptions. *Manufacturing & Service Operations Management*, 14(4):670–684.

Luo, L., Zhou, Y., Han, B. T., and Li, J. (2019). An optimization model to determine appointment scheduling window for an outpatient clinic with patient no-shows. *Health care management science*, 22:68–84.

Malki, Z., Atlam, E.-S., Hassanien, A. E., Dagnew, G., Elhosseini, M. A., and Gad, I. (2020). Association between weather data and covid-19 pandemic predicting mortality rate: Machine learning approaches. *Chaos, Solitons & Fractals*, 138:110137.

Mandelbaum, A., Momčilović, P., Trichakis, N., Kadish, S., Leib, R., and Bunnell, C. A. (2020). Data-driven appointment-scheduling under uncertainty: The case of an infusion unit in a cancer center. *Management Science*, 66(1):243–270.

Mantegna, R. N. (1994). Fast, accurate algorithm for numerical simulation of levy stable stochastic processes. *Physical Review E*, 49(5):4677.

Marques, I. and Captivo, M. E. (2017). Different stakeholders' perspectives for a surgical case assignment problem: Deterministic and robust approaches. *European Journal of Operational Research*, 261(1):260–278.

Marques, I., Captivo, M. E., and Pato, M. V. (2012). An integer programming approach to elective surgery scheduling. *OR spectrum*, 34(2):407–427.

Martínez-Álvarez, F., Asencio-Cortés, G., Torres, J. F., Gutiérrez-Avilés, D., Melgar-García, L., Pérez-Chacón, R., Rubio-Escudero, C., Riquelme, J. C., and Troncoso, A. (2020). Coronavirus optimization algorithm: a bioinspired metaheuristic based on the covid-19 propagation model. *Big data*, 8(4):308–322.

Mateus, C., Marques, I., and Captivo, M. E. (2017). Local search heuristics for a surgical case assignment problem. *Operations Research for Health Care*.

Milligan, G. N. and Barrett, A. D. (2015). *Vaccinology: an essential guide*. John Wiley & Sons.

Min, D. and Yih, Y. (2010). Scheduling elective surgery under uncertainty and downstream capacity constraints. *European Journal of Operational Research*, 206(3):642–652.

Mirjalili, S. (2015). The ant lion optimizer. *Advances in engineering software*, 83:80–98.

Mirjalili, S. (2016a). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural computing and applications*, 27:1053–1073.

Mirjalili, S. (2016b). Sca: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 96:120–133.

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., and Mirjalili, S. M. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software*, 114:163–191.

Mirjalili, S. and Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 95:51–67.

Mirjalili, S., Mirjalili, S. M., and Hatamlou, A. (2016). Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*, 27:495–513.

Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69:46–61.

Moghaddam, F. F., Moghaddam, R. F., and Cheriet, M. (2012). Curved space optimization: a random search based on general relativity theory. *arXiv preprint arXiv:1208.2214*.

Molina-Pariente, J. M., Fernandez-Viagas, V., and Framinan, J. M. (2015). Integrated operating room planning and scheduling problem with assistant surgeon dependent surgery durations. *Computers & Industrial Engineering*, 82:8–20.

Mousavirad, S. J. and Ebrahimpour-Komleh, H. (2017). Human mental search: a new population-based metaheuristic optimization algorithm. *Applied Intelligence*, 47:850–887.

Naderi, B., Roshanaei, V., Begen, M. A., Aleman, D. M., and Urbach, D. R. (2021). Increased surgical capacity without additional resources: Generalized operating room planning and scheduling. *Production and Operations Management*.

Najafi, A. A., Niaki, S. T. A., and Shahsavar, M. (2009). A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations. *Computers & Operations Research*, 36(11):2994–3001.

Neyshabouri, S. and Berg, B. P. (2017). Two-stage robust optimization approach to elective surgery and downstream capacity planning. *European Journal of Operational Research*, 260(1):21–40.

Ogulata, S. N. and Erol, R. (2003). A hierarchical multiple criteria mathematical programming approach for scheduling general surgery operations in large hospitals. *Journal of Medical Systems*, 27(3):259–270.

Ohio Department of Transportation (2023). Hourly percent by vehicle. `https://www.transportation.ohio.gov/programs/technical-services/traffic-monitoring/hourly-percent-by-vehicle`.

Okuonghae, D. and Omame, A. (2020). Analysis of a mathematical model for covid-19 population dynamics in lagos, nigeria. *Chaos, Solitons & Fractals*, 139:110032.

Ontario.ca (n.d.). How ontario is responding to covid-19. `https://www.ontario.ca/page/how-ontario-is-responding-covid-19`. Accessed: [Insert current date here].

Ozkarahan, I. (2000). Allocation of surgeries to operating rooms by goal programing. *Journal of Medical Systems*, 24(6):339–378.

Pahnehkolaei, S. M. A., Alfi, A., Sadollah, A., and Kim, J. H. (2017). Gradient-based water cycle algorithm with evaporation rate applied to chaos suppression. *Applied Soft Computing*, 53:420–440.

Panwar, H., Gupta, P., Siddiqui, M. K., Morales-Menendez, R., and Singh, V. (2020). Application of deep learning for fast detection of covid-19 in x-rays using ncovnet. *Chaos, Solitons & Fractals*, 138:109944.

Park, J., Kim, B.-I., Eom, M., and Choi, B. K. (2021). Operating room scheduling considering surgeons' preferences and cooperative operations. *Computers & Industrial Engineering*, 157:107306.

Peng, Y. and Nagata, M. H. (2020). An empirical overview of nonlinearity and overfitting in machine learning using covid-19 data. *Chaos, Solitons & Fractals*, 139:110055.

Pesata, V., Pallija, G., and Webb, A. A. (1999). A descriptive study of missed appointments: families' perceptions of barriers to care. *Journal of Pediatric Health Care*, 13(4):178–182.

Pinter, G., Felde, I., Mosavi, A., Ghamisi, P., and Gloaguen, R. (2020). Covid-19 pandemic prediction for hungary; a hybrid machine learning approach. *Mathematics*, 8(6):890.

Price, K. V. (2013). Differential evolution. In *Handbook of optimization: From classical to modern approach*, pages 187–214. Springer.

Public Health Agency of Canada (2020). Public health agency of canada. `https://www.canada.ca/en/public-health.html`.

Rashaideh, H., Sawaie, A., Al-Betar, M. A., Abualigah, L. M., Al-Laham, M. M., Al-Khatib, R. M., and Braik, M. (2019). A grey wolf optimizer for text document clustering. *Journal of Intelligent Systems*, 29(1):814–830.

Rashedi, E., Nezamabadi-Pour, H., and Saryazdi, S. (2009). Gsa: a gravitational search algorithm. *Information sciences*, 179(13):2232–2248.

Rechenberg, I. (1978). Evolutionsstrategien. In *Simulationsmethoden in der Medizin und Biologie: Workshop, Hannover, 29. Sept.–1. Okt. 1977*, pages 83–114. Springer.

Roshanaei, V., Booth, K. E., Aleman, D. M., Urbach, D. R., and Beck, J. C. (2020). Branch-and-check methods for multi-level operating room planning and scheduling. *International Journal of Production Economics*, 220:107433.

Roshanaei, V., Luong, C., Aleman, D. M., and Urbach, D. (2017). Propagating logic-based benders' decomposition approaches for distributed operating room scheduling. *European Journal of Operational Research*, 257(2):439–455.

Roshanaei, V. and Naderi, B. (2021). Solving integrated operating room planning and scheduling: Logic-based benders decomposition versus branch-price-and-cut. *European Journal of Operational Research*, 293(1):65–78.

Roy, R. K. (2010). *A primer on the Taguchi method*. Society of manufacturing engineers.

Salimi, H. (2015). Stochastic fractal search: a powerful metaheuristic algorithm. *Knowledge-based systems*, 75:1–18.

Salzarulo, P. A., Mahar, S., and Modi, S. (2016). Beyond patient classification: Using individual patient characteristics in appointment scheduling. *Production and Operations Management*, 25(6):1056–1072.

Samorani, M., Harris, S. L., Blount, L. G., Lu, H., and Santoro, M. A. (2022). Overbooked and overlooked: machine learning and racial bias in medical appointment scheduling. *Manufacturing & Service Operations Management*, 24(6):2825–2842.

Samuel, O. D., Okwu, M. O., Oyejide, O. J., Taghinezhad, E., Afzal, A., and Kaveh, M. (2020). Optimizing biodiesel production from abundant waste oils through empirical method and grey wolf optimizer. *Fuel*, 281:118701.

Sangaiah, A. K., Goli, A., Tirkolaee, E. B., Ranjbar-Bourani, M., Pandey, H. M., and Zhang, W. (2020). Big data-driven cognitive computing system for optimization of social media analytics. *Ieee Access*, 8:82215–82226.

Saremi, S., Mirjalili, S., and Lewis, A. (2017). Grasshopper optimisation algorithm: theory and application. *Advances in engineering software*, 105:30–47.

Shah-Hosseini, H. (2011). Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. *International journal of computational science and engineering*, 6(1-2):132–140.

Shareef, H., Ibrahim, A. A., and Mutlag, A. H. (2015). Lightning search algorithm. *Applied Soft Computing*, 36:315–333.

Shehadeh, K. S., Cohn, A. E., and Jiang, R. (2021). Using stochastic programming to solve an outpatient appointment scheduling problem with random service and arrival times. *Naval Research Logistics (NRL)*, 68(1):89–111.

Shehadeh, K. S. and Padman, R. (2021). A distributionally robust optimization approach for stochastic elective surgery scheduling with limited intensive care unit capacity. *European Journal of Operational Research*, 290(3):901–913.

Shoeibi, A., Khodatars, M., Jafari, M., Ghassemi, N., Sadeghi, D., Moridian, P., Khadem, A., Alizadehsani, R., Hussain, S., Zare, A., et al. (2024). Automated detection and forecasting of covid-19 using deep learning techniques: A review. *Neurocomputing*, page 127317.

Sickinger, S. and Kolisch, R. (2009). The performance of a generalized bailey–welch rule for outpatient appointment scheduling under inpatient and emergency demand. *Health care management science*, 12:408–419.

Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6):702–713.

Slaughter, G. (2020). Canada confirms first 'community case'of covid-19: Here's what that means. *CTV News*, 5(3).

Solis, F. J. and Wets, R. J.-B. (1981). Minimization by random search techniques. *Mathematics of operations research*, 6(1):19–30.

Soltani, M., Samorani, M., and Kolfal, B. (2019). Appointment scheduling with multiple providers and stochastic service times. *European Journal of Operational Research*, 277(2):667–683.

Statistics Canada (2020). Canada's national statistical agency. `https://www.statcan.gc.ca/en/start`.

Storn, R. and Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11:341–359.

Taguchi, G., Chowdhury, S., and Wu, Y. (2004). Taguchi's quality engineering handbook. *(No Title)*.

Tai-Seale, M., McGuire, T. G., and Zhang, W. (2007). Time allocation in primary care office visits. *Health services research*, 42(5):1871–1894.

Teles, P. (2020). Predicting the evolution of sars-covid-2 in portugal using an adapted sir model previously used in south korea for the mers outbreak. *arXiv preprint arXiv:2003.10047*.

Tian, Y., Luthra, I., and Zhang, X. (2020). Forecasting covid-19 cases using machine learning models. *MedRxiv*, pages 2020–07.

Tuli, S., Tuli, S., Tuli, R., and Gill, S. S. (2020). Predicting the growth and trend of covid-19 pandemic using machine learning and cloud computing. *Internet of things*, 11:100222.

Vijayakumar, B., Parikh, P. J., Scott, R., Barnes, A., and Gallimore, J. (2013). A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. *European Journal of Operational Research*, 224(3):583–591.

Vinden, C., Malthaner, R., McGee, J., McClure, J. A., Winick-Ng, J., Liu, K., Nash, D. M., Welk, B., and Dubois, L. (2016). Teaching surgery takes time: the impact of surgical education on time in the operating room. *Canadian Journal of Surgery*, 59(2):87.

Wang, P., Zheng, X., Li, J., and Zhu, B. (2020). Prediction of epidemic trends in covid-19 with logistic model and machine learning technics. *Chaos, Solitons & Fractals*, 139:110058.

Wang, S., Li, J., and Peng, C. (2017). Distributionally robust chance-constrained program surgery planning with downstream resource. In *Service Systems and Service Management (ICSSSM), 2017 International Conference on*, pages 1–6. IEEE.

Wang, S.-y., Wang, L., Liu, M., and Xu, Y. (2013). An effective estimation of distribution algorithm for solving the distributed permutation flow-shop scheduling problem. *International Journal of Production Economics*, 145(1):387–396.

Wang, Y., Tang, J., and Fung, R. Y. (2014). A column-generation-based heuristic algorithm for solving operating theater planning problem under stochastic demand and surgery cancellation risk. *International Journal of Production Economics*, 158:28–36.

Weeks, C. and Ha, T. T. (2020). Quebec hospitals struggling with influx of covid-19 patients even as province moves to reopen. `https://www.theglobeandmail.com/canada/article-quebec-hospitals-struggling-with-influxof-covid-19-patients-even-as`

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82.

Yang, X.-S. (2010). *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons.

Yang, X.-S. and Deb, S. (2009). Cuckoo search via lévy flights. In *2009 World congress on nature & biologically inspired computing (NaBIC)*, pages 210–214. Ieee.

Yang, Z., Zeng, Z., Wang, K., Wong, S.-S., Liang, W., Zanin, M., Liu, P., Cao, X., Gao, Z., Mai, Z., et al. (2020). Modified seir and ai prediction of the epidemics trend of covid-19 in china under public health interventions. *Journal of thoracic disease*, 12(3):165.

Zabinsky, Z. B. et al. (2009). Random search algorithms. *Department of Industrial and Systems Engineering, University of Washington, USA*.

Zacharias, C. and Yunes, T. (2020). Multimodularity in the stochastic appointment scheduling problem with discrete arrival epochs. *Management science*, 66(2):744–763.

Zamli, K. Z., Din, F., Ahmed, B. S., and Bures, M. (2018). A hybrid q-learning sine-cosine-based strategy for addressing the combinatorial test suite minimization problem. *PloS one*, 13(5):e0195675.

Zhan, Y., Wang, Z., and Wan, G. (2021). Home service routing and appointment scheduling with stochastic service times. *European Journal of Operational Research*, 288(1):98–110.

Zhang, J., Dridi, M., and El Moudni, A. (2020). Column-generation-based heuristic approaches to stochastic surgery scheduling with downstream capacity constraints. *International Journal of Production Economics*, 229:107764.

Zhu, Z. and Zhou, X. (2020). An efficient evolutionary grey wolf optimizer for multi-objective flexible job shop scheduling problem with hierarchical job precedence constraints. *Computers & Industrial Engineering*, 140:106280.