

OPTIMISING ANTIBIOTIC TREATMENTS
USING EVOLUTIONARY ALGORITHMS

MILA GORANOVA

Doctor of Philosophy

Institute of Computing Science and Mathematics

University of Stirling

30 September 2022

DECLARATION

I, Mila Gabrielova Goranova, declare that this thesis titled, “Optimising Antibiotic Treatments using Evolutionary Algorithm” and the work presented in it are my own. I confirm that:

- This work was done wholly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

This dissertation is a record of the work carried out at the University of Stirling between 2018 and 2022, under the supervision of Professor Gabriela Ochoa, Dr Andrew Hoyle and Dr Patrick Maier.

30 September 2022



Mila Goranova

ABSTRACT

Antimicrobial resistance is one of the biggest threats to global health, food security, and development. Antibiotic overuse and misuse are the main drivers for the emergence of resistance. Studies in the medical sphere have indicated that shortened antibiotic treatments can be as effective as standard fixed-dose ones, and have shown that an initial higher dose followed by a lower maintenance dose are more beneficial to patients with critical illnesses. It is crucial to optimise the use of existing antibiotics in order to improve medical outcomes, decrease toxicity and reduce the emergence of resistance. We formulate the design of antibiotic dosing regimens as a continuous optimisation problem, and use several evolutionary algorithms as the search technique. Regimens are represented as vectors of real numbers encoding daily doses, which can vary across the treatment duration. A stochastic mathematical model of bacterial infections with tuneable resistance levels is used to evaluate the effectiveness of evolved regimens. The main objective is to minimise the treatment failure rate, subject to a constraint on the maximum total antibiotic used. We consider simulations with different levels of bacterial resistance; two ways of administering the drug (orally and intravenously); as well as coinfections with two strains of bacteria. The approach produced effective dosing regimens, with an average improvement in lowering the failure rate 30%, when compared with standard fixed-daily-dose regimens with the same total amount of antibiotic. A general pattern of an optimised treatment is found, where if $2 \times x$ is the standard daily dose then the optimised treatment follows the $3 \times x$ mg, followed by several $2 \times x$ mg with a last dose of x mg. A noise handling technique is used to minimise the runtime of the experiments while maintaining the quality of treatments. The results of this work indicate that clinical studies confirming the

effectiveness of this approach could be highly beneficial to future of antibiotic treatments.

ACKNOWLEDGMENTS

Firstly, I would like to express my deepest appreciation to my supervisors Professor Gabriela Ochoa and Dr Andrew Hoyle as well as Dr Patrick Maier. Their continuous guidance, consistent academic support and advice was indispensable throughout the course of my PhD.

I'm extremely grateful to my partner Rory Forster for his endless support, proofreading, for keeping me sane during the ups and downs of the last four years. Thank you for all your love and encouragement!

Thanks should also go to my mother, who provided me with constant optimism, and to my grandparents for inspiring me to pursue a career in science.

I'd like to recognize the rest of the PhD student who I had the pleasure of meeting and a special thanks to Sarah, Scott, and Penny for their friendship and moral support.

Finally, I would like to thank my late pet Misha for the constant source of joy and comfort that she provided.

CONTENTS

| | | |
|----------|---|----------|
| i | INTRODUCTION | 1 |
| 1 | INTRODUCTION | 2 |
| 1.1 | Motivation | 2 |
| 1.2 | Overview | 3 |
| 1.2.1 | Project Background | 3 |
| 1.2.2 | Approach | 4 |
| 1.3 | Publications Produced | 4 |
| 1.3.1 | Optimising Antibiotic Treatments with Multi-objective Population-based Algorithms | 5 |
| 1.3.2 | Evolutionary optimisation of antibiotic dosing regimens for bacteria with different levels of resistance | 5 |
| 1.4 | Chapter Plan | 5 |
| 1.5 | Aim of Thesis | 6 |
| 1.5.1 | Mathematical Model Changes | 6 |
| 1.5.2 | Exploring Different Objective Approaches | 7 |
| 1.5.3 | Exploring Different Algorithms | 7 |
| 1.5.4 | Exploring Noisy Optimisation Function Handling | 8 |
| 2 | BACKGROUND & LITERATURE REVIEW | 9 |
| 2.1 | Biological Background | 9 |
| 2.1.1 | Bacteria | 10 |
| 2.1.2 | Bacterial DNA Mutations | 11 |
| 2.1.3 | Horizontal Gene Transfer | 11 |
| 2.1.4 | Antibiotics | 12 |
| 2.1.5 | Minimum Inhibitory Concentration | 13 |
| 2.1.6 | Other Approaches | 14 |

| | | |
|-------|--|----|
| 2.1.7 | Impact of Antibiotic Resistance | 15 |
| 2.2 | Mathematical Model | 15 |
| 2.2.1 | Paterson et al. Model | 16 |
| 2.2.2 | Differential Equations and Parameters | 17 |
| 2.3 | Mathematical Models in Literature | 19 |
| 3 | COMPUTATIONAL APPROACH | 21 |
| 3.1 | Optimisation and Metaheuristics | 21 |
| 3.1.1 | Evolutionary Optimisation | 23 |
| 3.1.2 | Exploitation and Exploration | 23 |
| 3.1.3 | Single-Objective Optimisation | 24 |
| 3.1.4 | Multi-Objective Optimisation | 24 |
| 3.2 | Problem Formulation | 27 |
| 3.3 | Algorithms Used | 28 |
| 3.3.1 | Non-dominated Sorting Genetic Algorithm II | 28 |
| 3.3.2 | Speed-constrained Multi-objective Particle Swarm Op- timisation Algorithm | 29 |
| 3.3.3 | Differential Evolution | 29 |
| 3.3.4 | Covariance Matrix Adaptation Evolution Strategy | 30 |
| 3.4 | Literature Review | 31 |
| 3.4.1 | Single-Objective Optimisation | 31 |
| 3.4.2 | Multi-objective Optimisation | 32 |
| ii | MY CONTRIBUTION | 35 |
| 4 | EXPLORING MULTI-OBJECTIVE EVOLUTIONARY OPTIMISATION | 36 |
| 4.1 | Introduction | 36 |
| 4.2 | Mathematical Model | 37 |
| 4.3 | Computational Methods | 40 |
| 4.3.1 | Implementation of the objectives | 40 |
| 4.3.2 | Noisy Failure Rate | 41 |
| 4.3.3 | Population-based algorithms | 41 |
| 4.3.4 | Hypervolume Indicator | 42 |

| | | |
|-------|--|----|
| 4.3.5 | Parameter Tuning | 42 |
| 4.4 | Results | 43 |
| 4.4.1 | Hypervolume Comparison of Algorithms | 43 |
| 4.4.2 | Partial Pareto-front of best results for NSGA-II and SMPSO | 47 |
| 4.4.3 | Best Treatments | 47 |
| 4.5 | Conclusion | 50 |
| 5 | SINGE OBJECTIVE ALGORITHM WITH CONSTRAINT APPROACH | 53 |
| 5.1 | Introduction | 53 |
| 5.2 | Methodology | 54 |
| 5.2.1 | Biomedical Background | 54 |
| 5.2.2 | Mathematical Model | 55 |
| 5.3 | Computational Optimisation | 60 |
| 5.4 | Results | 63 |
| 5.4.1 | Single Bacteria Strain and Oral Administration | 63 |
| 5.4.2 | Single Bacteria Strain and Intravenous Administration . | 70 |
| 5.4.3 | Two Bacteria Strains with Oral Administration | 73 |
| 5.4.4 | Extrapolating the Optimised Regimens | 75 |
| 5.5 | Conclusion | 77 |
| 6 | NOISY HANDLING MECHANISM AND RUNTIME OPTIMISATION | 79 |
| 6.1 | Introduction | 79 |
| 6.2 | Noise Handling Implementation — pycma | 80 |
| 6.3 | Experimental Setup | 81 |
| 6.4 | Preliminary Results | 82 |
| 6.4.1 | Evaluations Per Treatment | 82 |
| 6.4.2 | CMA-ES | 84 |
| 6.4.3 | UH-CMA-ES | 84 |
| 6.5 | Results | 87 |
| 6.5.1 | Failure Rate | 88 |
| 6.5.2 | Time to Clear Infections | 90 |
| 6.5.3 | Treatments | 94 |
| 6.6 | Conclusion | 97 |

| | | |
|-------|------------------------------|-----|
| iii | CONCLUSION | 99 |
| 7 | DISCUSSION | 100 |
| 7.1 | Summary of results | 100 |
| 7.1.1 | Chapter 4 | 100 |
| 7.1.2 | Chapter 5 | 101 |
| 7.1.3 | Chapter 6 | 101 |
| 7.2 | Future Work | 102 |
| 7.3 | Concluding Remarks | 104 |

LIST OF FIGURES

| | | |
|------------|--|----|
| Figure 1.1 | Antibiotic resistance occurrence | 3 |
| Figure 2.1 | Structure of a typical bacterial cell | 11 |
| Figure 3.1 | Solutions for multi-objective problem | 26 |
| Figure 4.1 | Boxplot Hypervolume Comparison of Algorithms | 45 |
| Figure 4.2 | Failure rate over total antibiotics scatter plot | 46 |
| Figure 4.3 | Barplot of treatments after re-evaluation | 49 |
| Figure 5.1 | Failure rate convergence over DE iterations | 62 |
| Figure 5.2 | Treatments by MIC, maximum total antibiotic and length | 64 |
| Figure 5.3 | Optimised treatments by MIC and maximum total anti- biotics | 65 |
| Figure 5.4 | Comparison of the best treatments | 66 |
| Figure 5.5 | Distribution of the expected time to clear | 69 |
| Figure 5.6 | Optimised treatments administered intravenously | 71 |
| Figure 5.7 | Optimised treatments compared to fixed-dose treatment | 71 |
| Figure 5.8 | Treatments with two strains of bacteria | 74 |
| Figure 6.1 | Boxplot of failure rates | 83 |
| Figure 6.2 | Pycma library output for CMA-ES runs | 85 |
| Figure 6.3 | Pycma library output for UH-CMA-ES runs | 86 |
| Figure 6.4 | Failure rate boxplot | 90 |
| Figure 6.5 | Distributions of the expected time to clear | 91 |
| Figure 6.6 | Boxplot of the expected time to clear | 93 |
| Figure 6.7 | Scatter plot of the expected time to clear | 94 |
| Figure 6.8 | Comparison of the dosage profiles | 96 |

LIST OF TABLES

| | | |
|-----------|---|----|
| Table 2.1 | Initial mathematical model parameters in [12] | 18 |
| Table 4.1 | Parameters and their values | 38 |
| Table 4.2 | F-race Parameters of NSGA-II | 44 |
| Table 4.3 | F-race Parameters of SMPSO | 44 |
| Table 4.4 | Top 10 treatments by failure rate | 48 |
| Table 5.1 | Mathematical model parameter values and references. . | 56 |
| Table 5.2 | List of all events for simulating the population of bacteria during treatment. | 58 |
| Table 5.3 | Events with description and equation | 59 |
| Table 5.4 | MIC and maximum total antibiotics used | 61 |
| Table 5.5 | Differential evolution control parameter values. | 61 |
| Table 5.6 | Comparison of treatments | 76 |
| Table 6.1 | Local optimum treatments for UH-CMA-ES | 89 |

Part I

INTRODUCTION

INTRODUCTION

In this chapter, the motivations for this work are presented as well as the approach taken, the list of publications, a chapter plan for the entirety of the work followed the main aims of this work are briefly introduced.

1.1 MOTIVATION

Antibiotic resistance is one of the major global health challenges today. The increased use of antibiotics, as well as the misuse of antibiotics, has led to a number of diseases becoming progressively harder to treat — the most commonly reported resistant bacteria are *Escherichia coli*, *Klebsiella pneumoniae*, *Staphylococcus aureus*, and *Streptococcus pneumoniae*, followed by *Salmonella* [1]. It was estimated in 2016 that by 2050 there would be around 10 million deaths a year caused by antibiotic and antimicrobial resistance [2]. According to The Global Leaders Group in a meeting in the United Nations General Assembly in September 2022 antimicrobial resistance contributes to almost 5 million deaths per year already, while disproportionately affecting low- and middle-income countries [3] — the data is provided by Global Antimicrobial Resistance and Use Surveillance System (GLASS) initiative started in October 2015 by the World Health Organisation (WHO). From those 5 million, 1.27 million deaths were the direct result of AMR and the rest died from illnesses in which bacterial AMR played a part. Antibiotic resistance is an urgent problem that needs to be addressed, and a way to minimise antibiotic resistance is to decrease the misuse and overuse of those drugs.

1.2 OVERVIEW

1.2.1 Project Background

As a result of the discovery of antimicrobials in the 1910s, many lives have been saved. Antimicrobials are agents that kill microorganisms or stop their growth, and antibiotics are the type of antimicrobials used in living organisms to fight bacterial infections.

Since the start of mass production of Penicillin in 1945, antibiotics have become one of the most commonly prescribed drugs not only in human medicine but also in agriculture, aquaculture, and food production. Many decades after the discovery of the first antibiotics, bacterial infections have again become a threat [4]. The World Health Organisation has stated that “antimicrobial resistance is a global crisis that threatens a century of progress in health and achievement” [5].

Epidemiological studies have found a direct link between the consumption of antibiotics and the development of resistant bacteria strains. The overuse and misuse of those medications are driving the evolution of resistant bacteria strains. Figure 1.1 gives a summary of how antibiotic resistance occurs and how resistance is passed on to different bacteria species [6] [7] [8].

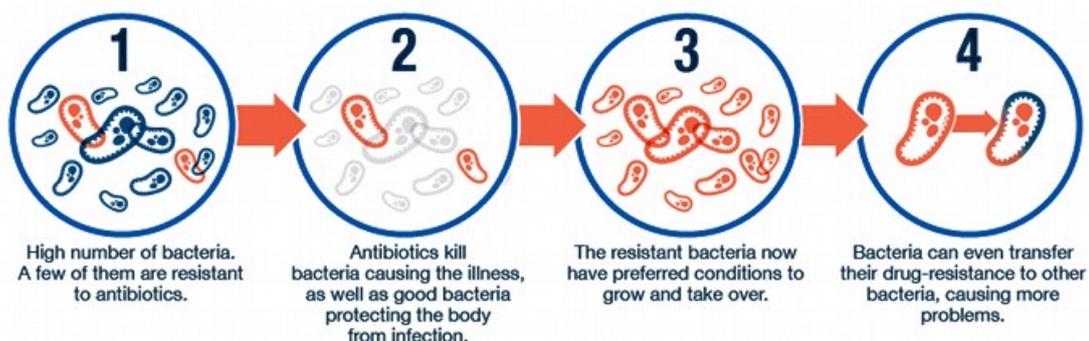


Figure 1.1: How antibiotic resistance occurs. *Source: World Health Organisation*

Due to antibiotic resistance, more bacterial infections become untreatable, causing an increase in deaths. It is estimated that 25 000 people die each year as a result of hospital infections caused by the top five resistant bacteria - Escherichia coli (E. coli), Klebsiella pneumoniae (K.Pneumoniae), Enterococcus faecium, Pseudomonas aeruginosa and Methicillin-resistant Staphylococcus aureus (MRSA), which adds over 1 billion pounds to hospital treatment [9]. Between the 1940s and 1960s, more than 20 new classes of antibiotics were marketed. Since the 1960s only two new classes reached the market [10] until September 2018 when a new class of antibiotics named *optimized arylomycins* was discovered [11].

1.2.2 Approach

In this project, we explore applying different treatment regimens using a mathematical model to analyse how successful each treatment is. We aim to minimise the overall amount of antibiotics prescribed while still having a successful treatment. This will tackle the problem of overuse of antibiotics, as the total number of antibiotics taken by the patient will be minimised. It is important to note that overuse of antibiotics could include several other problems that are not in the scope of this thesis — prescribing antibiotics where antibiotic treatment is not the appropriate choice, giving more antibiotics than needed or needing to take too much as the initial treatment was taken inefficiently. The approach was first proposed by Paterson et al. [12] where both deterministic and stochastic models were explored. More details on the model are provided in Section 5.2.2.

1.3 PUBLICATIONS PRODUCED

Publications produced during the course of this PhD are now mentioned. These are in chronological order, starting with the oldest. For each, a statement about

the authorship of the work is given. In the interest of clarity, no experiments or results obtained by other authors are included in this thesis.

1.3.1 *Optimising Antibiotic Treatments with Multi-objective Population-based Algorithms*

This paper forms Chapter 4 and is published in (the 2020 IEEE Congress on Evolutionary Computation (CEC), track) with authors Mila Goranova, Marco A. Contreras-Cruz, Andrew Hoyle and Gabriela Ochoa. All writing, implementation, experiments, and results were conducted by the author of this thesis, with the exception of the parameter tuning section of the conference paper which is implemented and conducted by Marco A. Contreras-Cruz. Proofreads and small edits were done by all co-authors.

1.3.2 *Evolutionary optimisation of antibiotic dosing regimens for bacteria with different levels of resistance*

Currently, the paper is accepted and in the process of being published in the journal *Artificial Intelligence in Medicine* with authors Mila Goranova, Gabriela Ochoa, Patrick Maier and Andrew Hoyle. All draft writing, experiments and results were conducted by the author of this thesis, other than proofreads by the co-authors. The initial implementation was started by Gabriela Ochoa, with some additions by Patrick Maier. The final version of the implementation was done by the author of this thesis.

1.4 CHAPTER PLAN

Chapter 2 introduces the biological processes that are modelled in our model, then the initial mathematical model is described. Lastly, a literature review

is presented where mathematical models and similar approaches are used to solve problems in similar domains.

Chapter 3 describes the background of the computing science approach we use. Each of the chapters shows a different computational method used, those methods are detailed in this chapter.

Chapter 4 is based on [13] conference paper and shows the difference between two multi-objective algorithms' performance in optimising antibiotic treatments.

Chapter 5 shows a new single-objective approach to the problem, where the formulation is simplified and optimised for running time.

In Chapter 6 further optimisation to use of computation power is made and presented while using a different algorithm and techniques. Runtime comparisons are shown.

Chapter 7 summarises the results in the previous chapters and discusses limitations and future work.

1.5 AIM OF THESIS

This thesis aims to use various evolutionary algorithms and mechanics to identify successful antibiotic treatment regimens and explore different strategies while maintaining computational efficiency.

1.5.1 *Mathematical Model Changes*

One of the central aims of this thesis is to build upon the mathematical model proposed by Paterson et al. in [12] and explore other scenarios of bacterial infection and types of treatment. In Chapter 4 we introduced two new modelling techniques to the mathematical model. Those modelling techniques are used in the approach in the rest of the chapters.

The first modelling technique is to make the model more realistic and is based on how antibiotics are processed by the human body depending on the intake (oral or intravenous). In Chapter 5 we add the exploration of different levels of resistance where the intake is specified as well as if there is a single or multiple bacteria strains present.

The second mathematical modelling technique is an approximate method for the simulation, adding robustness for larger systems and experiments and increasing computational efficiency.

1.5.2 *Exploring Different Objective Approaches*

In this thesis, we want to examine the problem in different formulations. The problem is formulated and explored both as single and multi-objective in this work. In Chapter 4, the problem has multiple objectives - minimising the antibiotic used and minimising the failure rate. In Chapters 5 and 6, the problem is approached as a single-objective where the only objective is to minimise the failure rate — here the maximum antibiotic that could be used is set as a constraint instead.

1.5.3 *Exploring Different Algorithms*

Another aspect which we aim to look into is the different types of evolutionary algorithms and how each of them performs with our problem. In each of the chapters, a different algorithm is used and motivation for the choice is given. In Chapter 4 we look into two different types of multi-objective evolutionary algorithms each inspired by different biological occurrences — one of them is modelled after Darwin's theory of survival of the fittest and natural selection and the other one — the behaviour of bird flocking and fish schooling. Chapters 5 and 6's algorithms are also inspired by Darwin's theory of evolution but implement different aspects of it.

1.5.4 *Exploring Noisy Optimisation Function Handling*

The problem this thesis is looking into is stochastic in nature as similar antibiotic treatments might have a different outcome depending on the person they are administered to, the stage of the infection, the resistance of the bacteria, etc. This means that when running our mathematical model, there is some noise in the evaluation of each one of the possible treatments. An uncertainty-handling technique exploration is essential to the computational approach to cope with the noise in the fitness function (for us this is the mathematical model). In Chapter 6 we look into a specific uncertainty-handling technique and a more robust approach to optimising antibiotic treatments.

BACKGROUND & LITERATURE REVIEW

2.1 BIOLOGICAL BACKGROUND

Antibiotics have been used more and more in medicine, agriculture, and aquaculture since their discovery. The first antibiotic named Salvarsan (also known as Arsphenamine or compound 606) was introduced at the beginning of the 1910s as the first effective treatment for syphilis and African trypanosomiasis. Then in 1928, penicillin was discovered, starting the golden age of natural product antibiotic discovery that peaked in the mid-1950s. This led to a drastic change in modern medicine, extending the average human lifespan by 23 years [14]. Since then, a gradual decline in antibiotic discovery and development, and the evolution of drug resistance in many human pathogens, have led to the current antimicrobial resistance crisis. The development of bacteria strains resistant to antibiotics has been aided by their pervasive use and over-reliance on them. Antibiotics today sometimes find it difficult to kill, or simply cannot kill, resistant bacteria due to its mutation, those bacteria then mutate further, survive and multiply. [15], [16].

Conventional antibiotic treatments apply a constant dose for a set amount of time - for example, take 100 mg per day for 7 days. However, studies in the medical sphere have indicated that shortened treatments can be as effective [17]. Other studies have shown that an initial higher dose followed by a lower maintenance dose is more beneficial to patients with critical illnesses [18].

2.1.1 *Bacteria*

To understand how antibiotics work, it is important to explain the biology of the bacterium organisms. Bacteria are one of the lowest and simplest forms of life. They are single-cell organisms with a prokaryotic structure (lack of nucleus and member-bound cell compartments). Due to their biological simplicity, bacteria are the most numerous of living organisms in terms of the number of species, number of organisms, and total mass of the organisms on Earth [19].

In Fig 2.1 an example structure of a typical bacteria cell is provided. Typical bacteria cells have three main parts [20]:

- cell envelope — comprises the innermost cell membrane, cell wall, periplasmic space between the plasma membrane and cell wall and the outermost layers surrounding the cell wall.
- cytoplasm — contains nucleoids as genetic material, ribosomes as protein synthesis machinery and inclusion bodies dispersed all over the cytoplasmic space. The nucleoid is responsible for controlling the activity of the cell and reproduction and is where transcription and replication of DNA take place. It is important to note that the nucleoid, in contrast to the nucleus of a eukaryotic cell, is not surrounded by a nuclear membrane allowing for easy transfer of DNA between bacteria.
- extracellular appendages (examples are fimbriae, pili and flagella) — serve multiple functions and help cells in conjugation, attachment, and locomotion.

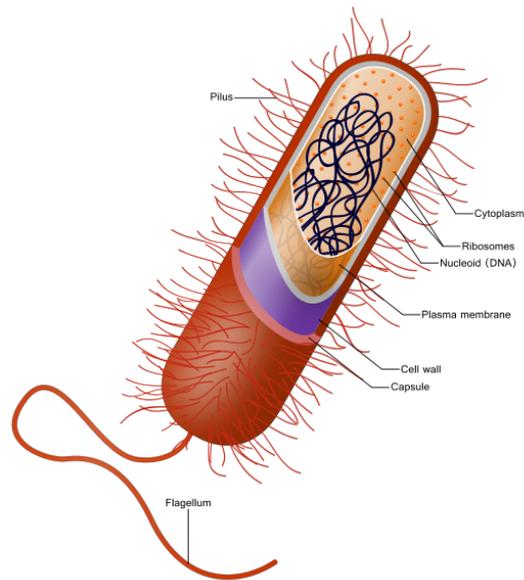


Figure 2.1: Structure of a typical bacterial cell. *Source Wikimedia Commons (Creative Commons Attribution license)*

2.1.2 *Bacterial DNA Mutations*

Mutation of the bacteria's DNA can provide resistance to antibiotics. A bacterial mutation is a change in the nucleotide sequence of a short region of the genome and can create new cellular functionalities or lead to the dysfunction of others. Mutations can occur spontaneously or be caused by exposure to a mutation-inducing environment. Spontaneous mutations occur at a rate of 1 in 10^5 to 10^8 [21] and contribute to random population variation. Mutation-inducing environments include radiation or UV exposure, chemical mutagens, base analog forms, deaminating agents, alkylating agents, etc. Beneficial mutations can be passed on not only through a parent-offspring relationship.

2.1.3 *Horizontal Gene Transfer*

Horizontal gene transfer (HGT) is the sharing of genetic material between organisms that are not in a parent-offspring relationship [22]. After a mutation

has occurred, it can spread through the population through HGT. HGT is responsible for increased propagation of resistance through bacterial populations [23]. If bacteria acquire resistant genes in an environment where they are beneficial, HGT will facilitate the spread of these genes within the population [24], [25].

2.1.4 *Antibiotics*

Antibiotics are a subset of antimicrobials, which are chemical substances used to treat bacterial infections and diseases. Antibiotics act in two main ways: they prevent the growth and reproduction of the bacterial cell (bacteriostatic) or they actively kill the bacterial cell (bactericidal). Their origin could be natural, semisynthetic, or synthetic. For example, some natural antibiotics are metals such as mercury, arsenic, copper, and silver. Semisynthetic antibiotics are derived from natural ones but have been slightly improved to have more beneficial characteristics — for example, fewer side effects or higher bacterial resistance. Synthetic antibiotics are usually chemically related to natural antibiotics, examples of such antibiotics are sulphonamides, diaminopyrimidine, co-trimoxazole, antivirals, antifungals, anticancer drugs, antimalarials, anti-tuberculosis drugs, etc. [26]–[30].

Eighty years of use and misuse have increased the frequency of resistance for the majority of antibiotic and bacterial combinations. In fact, this adaptive evolution of bacteria has been so successful that some bacterial infections are practically resistant to antibiotic treatment. The bacterium has not only altered in its ability to withstand the drug but potentially also in its interaction with the host and environment. [31].

2.1.5 *Minimum Inhibitory Concentration*

Minimum inhibitory concentration (MIC) is the lowest concentration of a chemical, usually a specific antibiotic, which prevents the visible growth of a bacterium or bacteria culture. MIC depends on the microorganism, the affected human being, and the antibiotic itself [32]. In this work, we use MIC as part of our mathematical model to represent the resistance of the bacteria to the antibiotics during the treatment.

2.1.5.1 *Levels and Forms of Antibiotic Resistance*

Multi-drug resistance (MDR) is defined by Magiorakos et al. as acquired non-susceptibility to at least one agent in three or more antimicrobial categories, however different definitions for the term exist. Another definition of MDR bacteria is bacteria that are “resistant to one key antimicrobial agent”. Bacteria could also be classified as extreme drug-resistant (XDR). They are significant due not only to their resistance to multiple antimicrobial agents, but also to the likelihood of being resistant to all, or almost all, approved antimicrobial agents. Pandrug-resistant (PDR) bacteria are bacteria resistant to all antimicrobial agents [33].

Antibiotic Heteroresistance happens when a pre-existing subpopulation of resistant cells rapidly replicates in the presence of a given antibiotic, whereas the majority population of susceptible cells is killed. This happens in scenarios where a subculture is collected from the patient and isolated for resistance. In the collected subculture, there is a subpopulation of cells that shows a higher resistance compared to the main population. This sometimes leads to that population being the primary bacterial population and making the bacterial infection more resistant to antibiotics [34], [35].

2.1.6 *Other Approaches*

It is important to note that there are other approaches to treating and preventing bacterial infections rather than antimicrobials [36].

- **Antibodies** that bind to and inactivate a pathogen, its virulence factors, or its toxins were widely considered one of the alternative approaches most likely to have a major clinical impact.
- **Probiotics** are defined as live microorganisms that, when administered in adequate amounts, confer a health benefit to the host organism.
- **Bacteriophages.** Phage lysins are enzymes used by bacteriophages to destroy the cell wall of a target bacterium. They infect and kill bacteria and have the potential to replace antibiotics for some indications. Bacteriophages could be used in small doses because they replicate when their host bacterium is present [37].
- **Immune stimulation.** Successful antimicrobial therapy depends on an appropriate immune response. Immune stimulation has been proposed as a potential adjunct approach in conjunction with antibiotic therapy.
- **Vaccines.** They substantially reduce the incidence of infection and, therefore, the need for antibiotics.
- **Antimicrobial Peptides** are a class of small peptides that exist in nature and are an important part of the immune system of organisms. They have a wide range of inhibitory effects against bacteria, fungi, parasites, and viruses. Antimicrobial peptides constitute one of the most promising alternatives to antibiotics since they could be used to treat bacterial infections, especially those caused by multidrug-resistant pathogens [38].

2.1.7 *Impact of Antibiotic Resistance*

Epidemiological studies have found a direct link between the consumption of antibiotics and the development of resistant bacteria strains [8]. The overuse and misuse of those medications are driving the evolution of resistant bacteria strains. With the increasing growth of the human population and the demand for animal protein, the use of antibiotics in food production continues to rise as well [31], [39], [40].

Due to antibiotic resistance, more bacterial infections become untreatable, causing an increase in deaths. It is estimated that 25 000 people die each year as a result of hospital infections caused by the top five resistant bacteria - Escherichia coli (E. coli), Klebsiella pneumoniae (K.Pneumoniae), Enterococcus faecium, Pseudomonas aeruginosa and Methicillin-resistant Staphylococcus aureus (MRSA), which adds over \$1 billion to hospital treatment [9].

Even though alternatives to antibiotic treatments are being researched, the world urgently needs to change the way it prescribes and uses antibiotics, as it is still the most effective way to fight bacterial infections. However, even if progress is made with new medicine development, without a change in behaviour in the use of antibiotics, antibiotic resistance will still remain a major threat.

2.2 MATHEMATICAL MODEL

This work is looking into optimising antibiotic treatments while ensuring the prolonged effectiveness of the administered drugs. The model explored was first introduced by Paterson et al. [12] where a mathematical model simulates the progression of a bacterial infection. In this section we will explain that initial model, and in the contribution chapters we have taken this model and built upon it.

2.2.1 *Paterson et al. Model*

Paterson et al. [12] propose two variants of the mathematical model — deterministic and stochastic. Here we will first look into the deterministic equations and then discuss the stochastic approach. In the contributing chapters, we chose the stochastic mathematical model as it is closer to the real world, as not all patients will respond the same to the same antibiotic treatment. The stochastic process is achieved by using a Gillespie algorithm [41]. A later work [42] takes the stochastic model and explores several formulations of the problem.

2.2.1.1 *Overview of Model*

The formulation detailed in [12], where a mathematical model of the progression of a bacterial infection, and the effect of antibiotic treatment, is proposed. When antibiotic treatments are designed, there are two key variables — the daily dosages and the treatment duration. This is modelled as a vector of doses $\underline{x} = (x_1, x_2, \dots, x_n)$, where x_i represents the dosage taken on day i , where $0 \leq x_i \leq x$ in works [12] and [42].

The stochastic model follows the steps below for each day and then additional three days when no antibiotics are taken:

- Antibiotic dose for the day is taken. The following steps happen until the next dose needs to be taken:
 - Probabilities for bacteria’s population death and reproduction are calculated.
 - Bacteria population is updated based on the probabilities.
 - Time increases by a time-step.
 - New concentration of the antibiotic in the body is calculated.
- Next dose is taken.

2.2.2 Differential Equations and Parameters

Differential equations in keeping with studies [24], [43]–[45] are used to describe the dynamics of a population of bacteria and how the antibiotics affect them. It is important to note that in the Paterson et al. work there are two types of bacteria — susceptible S and resistant R , but in *Chapter 4* and *Chapter 6* we only explore the model where the bacterial infection is caused by only one type of bacteria. The equations 2.1, 2.2 and 2.3 represent the whole deterministic mathematical model used by Paterson et al. where R stands for resistant bacteria and S for susceptible:

$$\frac{dS}{dt} = \underbrace{rS \left(1 - \frac{S+R}{K}\right)}_{\text{Natural Growth}} - \underbrace{\theta S}_{\text{Horizontal Gene Transfer}} - \underbrace{A_S(C)S}_{\text{Antibiotics Death}} \quad (2.1)$$

$$\frac{dR}{dt} = \underbrace{rR \left(1 - \frac{S+R}{K}\right) (1 - \alpha)}_{\text{Natural Growth}} + \underbrace{\beta SR}_{\text{Horizontal Gene Transfer}} - \underbrace{A_R(C)R}_{\text{Antibiotics Death}} \quad (2.2)$$

$$\frac{dC}{dt} = \underbrace{\sum_{n=1}^{10} D_n \delta(t - \hat{t}_n)}_{\text{Antibiotic Doses}} - \underbrace{gC}_{\text{Degradation}} \quad (2.3)$$

Table 2.1: Initial mathematical model parameters in [12]

| Parameter | Description | Value |
|--------------|---|--------------|
| r | Replication Rate | 2.7726 |
| K | Carrying Capacity | 1000 |
| β | Rate of Transmission of Resistant Plasmid | 0.00001 |
| θ | Natural Death Rate | 0.2 |
| a | Cost of Resistance | 0.2 |
| g | Degradation rate of antibiotic | 0.48 |
| \min | Min net growth at high AB concentrations | 2.1 |
| \max | Max net growth in absence of AB | $r - \theta$ |
| mic | Min inhibitory concentration (MIC) | 16 |
| k | Hill coefficient | 4 |

2.2.2.1 Gillespie Algorithm

Paterson et al. work uses the well-established Gillespie algorithm to obtain a stochastic simulation for the different treatment regimens. By calculating the probability of the individual events occurring, based on rates and parameter values from the deterministic model, the Gillespie algorithm randomly chooses the next event to happen and the time at which it will happen. An example for two events for resistant bacteria, where C is the concentration of the antibiotic, is given below:

$$\text{ReproductionRate}_{(\text{bacteria}_R)} = rB_R \left(1 - \frac{B_R}{K} \right) \quad (2.4)$$

$$\text{DeathRate}_{(\text{bacteria}_R)} = mB_R + \frac{(\max_R - \min_R) \left(\frac{C}{\text{mic}_R} \right)_R^k}{\left(\frac{C}{\text{mic}_R} \right)_R^k \frac{\min_R}{\max_R}} B_R \quad (2.5)$$

The population of the bacteria is adjusted according to the events that have happened, and the process is repeated. As the events are chosen randomly, each simulation will be slightly different. The success rate for each treatment regimen was obtained by calculating the total number of simulations which resulted in the eradication of both susceptible and resistant bacteria.

As asexual reproduction requires energy, it is assumed that the growth rate of bacteria is limited and therefore modelled using the standard logistic growth equation. This choice for using the standard logistic growth equation is made based on previous research such as [24], where a similar population is modelled. A cost, α , is associated with carrying the genes which introduce resistance to antibiotics and results in a reduced growth rate for the resistant strain. Genes can pass from resistant to previously susceptible bacteria through HGT, β , resulting in the loss of susceptible bacteria and the addition of resistant bacteria. There are 3 main mechanisms of HGT: transformation, transduction and conjugation. The mathematical model does not distinguish between the differing modes of HGT. Both susceptible and resistant bacteria die at a natural death rate, θ , and through exposure to antibiotics.

2.3 MATHEMATICAL MODELS IN LITERATURE

Mathematical and bio-mathematical modelling has become an accessible way of testing hypotheses in medicine and assessing the improvement of current processes without a clinical trial. We will look into some prominent and recent studies where different antibiotic treatment models were explored.

Birkegaard et al. [46] explore several mathematical models proposed in publications and a comprehensive review is given where models are grouped based on the approach used. The majority of models explored are population-based models (77%) with a smaller number of agent-based models or individual models as well as one nested model [47] where the main individual are pigs and the bacterial populations inside them are modelled. From the listed pub-

lications, [48] and [49] are concentrated on treatment optimisation and are explored in more detail below.

D'Agata et al. [48] propose an individual-based model (IBM) formulated as a system of stochastically determined events describing the complexities of the transmission dynamics of antibiotic-resistant bacteria, as well as a healthcare worker (HCW) model. Treatment scheduling was part of the IBM model, where there are two types of bacteria modelled - bacteria (N) which are non-resistant to the antibiotic treatment and bacteria (R) which are resistant to the antibiotic treatment. The analysis of both models shows that for the emergence and spread of antibiotic-resistant bacteria, it is crucial that early initiation of treatment and minimisation of its duration is needed for preventing resistance epidemics in hospitals.

COMPUTATIONAL APPROACH

In the following sections of this thesis, we will look into the background of optimisation algorithms - multi-objective optimisation and single-objective optimisation as well as the background and some notable works where such techniques are used in the medical field. Finally, we will present the algorithms and techniques that are used in the following chapters.

3.1 OPTIMISATION AND METAHEURISTICS

Optimisation problems are present in many domains — science, engineering, management and business. Such problems can be defined by the tuple (S, f) , where S represents the set of feasible solutions (search space), and f the objective function to optimise. The individual solutions $s \in S$ each have a different objective function $f(s)$ value [50].

Algorithms for optimisation can be roughly divided into two categories: exact algorithms and heuristics. [51] The exact algorithms are designed in a way where it is guaranteed that the optimal solution is found in a finite amount of time. However, in order to do that, the exact algorithms have to explore the search space and examine possible solutions, which could be very computationally expensive and it does not scale for large problems. Heuristics optimisations are strategies using readily available information to control problem-solving processes and are often problem-dependent. A heuristic approach trades the optimality, completeness, accuracy, or precision of the exact algorithms for speed and might not provide the optimal solution, but will mostly produce a satisfactory one [52].

A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimisation algorithms and are problem independent [53].

Some terminology describing metaheuristic algorithms [50], [54]–[60]:

- Local optima — a solution that is optimal (either maximal or minimal) within a neighbouring set of candidate solutions.
- Global optima — the optimal solution among all possible solutions, not just those in a particular neighbourhood of values.
- Single-solution algorithm — modifications and improvements to a single solution are made (called current or incumbent solution) over a number of iterations. A new solution is obtained in each iteration by a single move of the initial solution from the neighbourhood of that solution. Algorithms using this technique are Simulated Annealing, Iterated Local Search, Variable Neighbourhood Search, Guided Local Search, etc.
- Population-based algorithm — good solutions are found by iteratively selecting and then combining existing solutions from a set, usually called a population. Such algorithms are Artificial Immune System, Genetic Algorithm, Ant Colony Optimisation, Particle Swarm Optimisation, Stochastic Diffusion Search, Artificial Bee Colony, etc.
- Optimization objective — is an effective approach to achieve a "best" solution, where a single objective is maximized or minimised.
- Fitness function — a particular type of objective function that is used to summarise, as a single figure of merit, how close a given design solution is to achieving the set objective.
- Solution space — the set of all possible solutions for the combinatorial optimisation problem.

3.1.1 *Evolutionary Optimisation*

According to Blum et al. there are two ways evolutionary optimisation is defined [61]:

- A specific term devoted to a group of algorithms, including Genetic Algorithms, Genetic Programming, Evolution Strategies, Evolutionary Programming, and to a lesser extent Differential Evolution and Estimation of Distribution Algorithms.
- A general term describing population-based search methods that involve some form of randomness and selection.

Evolutionary optimisation is a type of Artificial Intelligence and is mainly inspired by natural processes, such as natural selection, species migration, bird flocks, human culture, and ant colonies [62].

3.1.2 *Exploitation and Exploration*

Two key elements in evolutionary optimisation are exploitation and exploration, and the tradeoff between the two is critical to the performance of the algorithms [63]. Exploration is key in algorithms like Hill Climber, belonging to the family of local search. Local search algorithms move from solution to solution in the search space by making local changes until an optimal solution is found, or a time limit is reached [64]. Exploitation algorithms like Random Search, where every iteration is not dependent on the prior iteration's candidate solution and is moving to different positions in the search space [65]. Balancing between the two is very important, as too much exploitation of the same space might result in a local optima, and too much exploration might miss an optima altogether.

3.1.3 *Single-Objective Optimisation*

A Single-Objective Problem is defined as minimising (or maximising) $f(s)$ subject to $g_i(s) \leq 0$, $i = \{1, \dots, m\}$, and $h_j(s) = 0$, $j = \{1, \dots, p\}$ $s \in \Omega$. A solution minimises (or maximises) the scalar, $f(s)$ where s is an n -dimensional decision variable vector $s = (s_1, \dots, s_n)$ from some universe Ω . Here, $h_j(s) = 0$ represents a specific constraint that needs to be fulfilled. s could either be continuous or discrete, and f could too be continuous or discrete. The goal of determining the global optima solution is called the global optimisation problem for a single-objective problem [66].

3.1.4 *Multi-Objective Optimisation*

Problems with multiple objectives are present in most disciplines. Many real-world problems are usually presented as non-linear programming problems with multiple conflicting objectives. Usually, mostly due to the lack of a better solution technique, these problems are converted into a single-objective problem and then solved on the basis of that new formulation.

Let us consider a problem in which you have to prescribe antibiotic treatment for a specific infection. You have to minimise the total dosage, take into account the strength of the antibiotic and minimise the treatment length. This is a three-objective problem that cannot be solved as a single-objective problem without introducing constraints on some objectives.

A Multi-Objective problem can be defined as a vector of decision variables which satisfies constraints and optimises a vector function whose elements represent the objective functions. These functions are usually in conflict with one another, and we need to find such a solution that would find the values of all the objective functions acceptable to the decision maker [67].

For a problem to be considered multi-objective, it has to have the following properties:

- Cardinality (the number of elements) of the optimal set has to be more than one.
- There should be at least two different goals of the optimisation.
- The goals should have different search spaces.

In Figure 3.1, seven different solutions to the antibiotics prescription problem are plotted where axis x is the total dosage prescribed in milligrams, axis y is the treatment length in days and axis z is the strength of the class of antibiotic marked from 1 to 5, where 1 has the lowest strength. Between any of the seven solutions, one will always be better in terms of one objective, but this betterment comes only from a sacrifice of other objectives. Such trade-off solutions provide a clear *front* on an objective space plotted with objective values. This front is called the *Pareto-optimal front*, and all such trade-off solutions are called *Pareto-optimal solutions*.

A solution is called non-dominated, Pareto optimal, Pareto efficient or non-inferior if none of the objective functions can be improved in value without degrading some other objective values. Without additional subjective preference information, all Pareto optimal solutions are considered equally good (as vectors cannot be ordered completely).

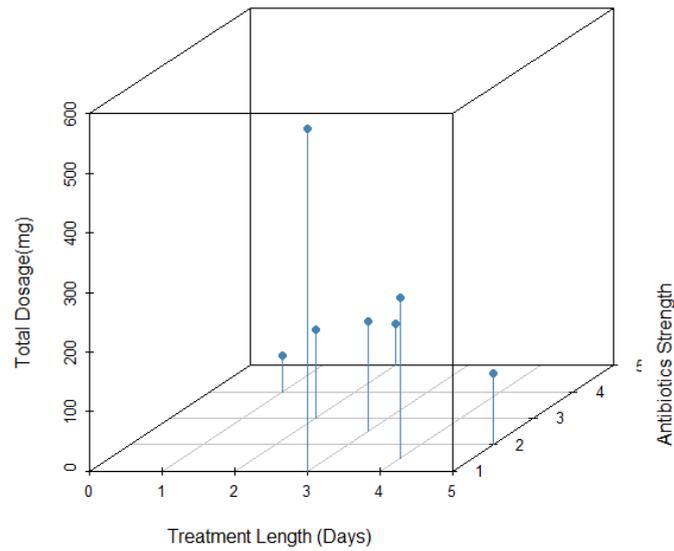


Figure 3.1: Representation of available solutions for a three-objective decision-making problem. Note: These points are arbitrarily taken and lines are added for easier reading.

The primary goals of a Multi-Objective Evolutionary Algorithm (MOEA) are [66]:

- Preserve non-dominated points in objective space and associate solution points in decision space.
- Continue to make algorithmic progress towards the Pareto Front in objective function space.
- Maintain diversity of points on the Pareto front and/or of the Pareto optimal solutions - decision space.
- Provide the decision maker with enough but also a limited number of Pareto points for selection resulting in decision variable values.

3.2 PROBLEM FORMULATION

Conventional antibiotic treatments apply a constant dose for a fixed amount of time — for example, take 60mg per day for 10 days. However, medical studies have indicated that shorter treatments can be more effective [17]. Other studies have shown that an initial higher dose followed by a lower maintenance dose is beneficial to patients with critical illnesses [18]. Therefore, in this work, we will look into optimising antibiotic treatments where different doses can be taken on each day of the treatment.

The treatment is represented as a vector, where each one of the values is a daily dosage and the length of the vector is the duration of treatment in days. An example treatment can be represented like $[45, 50, 30, 55, 30]$ — this indicates a five-day-long treatment where the dose for the first day is 45mg, the dose for the second day is 50mg, etc. The design of antibiotic treatments can be formulated as an optimisation problem where one or several objectives need to be satisfied.

In order to evaluate the fitness of treatments (solutions), we use a stochastic mathematical model of bacterial infection which serves as our primary fitness function. As explained in the previous chapter, a single run of the model returns either 0 (unsuccessful treatment) or 1 (successful treatment). The different result in the outcome depends on the the Gillespie algorithm and is the reason we need to run the mathematical model several times to have an accurate evaluation. In chapter 6 we explore the levels of noise depending on how many times the mathematical model runs. In the following chapters, we call this evaluation failure rate and reevaluation rate, and we aim to minimise it.

A secondary fitness function is A_{total} , which is the sum of all the antibiotics in the treatment. In the case of $s = [45, 50, 30, 55, 30]$, $A_{\text{total}} = 210\text{mg}$. This secondary fitness function is explored in Chapter 4 and then introduced as a constraint on the solutions in the following chapters.

3.3 ALGORITHMS USED

In this thesis, several evolutionary algorithms are explored. It is important to note that the listed ones are not the only ones that have been implemented, but those are the ones that provided the best results from the available libraries. Algorithms like HypE (An Algorithm for Fast Hypervolume-Based Many-Objective Optimization) and IBEA (Indicator-Based Selection in Multiobjective Search) were explored, but SMPSO and NSGA-II suited the problem better from the jMetalPy library. From SciPy it was important that there was a clause for adding constraints and Differential Evolution was chosen over the minimisation algorithms. Covariance Matrix Adaptation Evolution Strategy is chosen specifically for its uncertainty handling mechanics.

3.3.1 *Non-dominated Sorting Genetic Algorithm II*

The multi-objective algorithm chosen to solve our problem formulation first for our model is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) which is one of the benchmark algorithms used in the multi-objective evolutionary computation field proposed in 2002 by Deb et al. [68] which is an extension and improvement of the NSGA proposed by Srinivas and Deb in 1995 [69]. The algorithm is based on an elitism approach and uses evolutionary operators such as selection, genetic crossover and genetic mutation. The population is sorted into a hierarchy of subpopulations based on the ordering of the Pareto dominance, where each subgroup of the population is evaluated on the Pareto Front and the resulting groups. Similarity measures are used so that a diverse population of non-dominated solutions is achieved.

In this work, we use the library jMetalPy [70] for solving our problem, where the implementation follows the original proposal of the algorithm by Deb et al. in [68].

3.3.2 *Speed-constrained Multi-objective Particle Swarm Optimisation Algorithm*

Speed-constrained Multi-objective Particle Swarm Optimisation Algorithm (SMPSO) [71], [72] is a multi-objective particle swarm optimisation algorithm that uses a strategy to limit the velocity of the particles. This strategy allows for producing new effective particle positions when the velocity becomes too high. It also includes polynomial mutation and an external archive to store the non-dominated solutions found during the search. SMPSO produced remarkable results when compared to NSGA-II on a number of standard benchmark functions in [71]. An interesting feature of SMPSO is the use of polynomial mutation as a turbulence factor and an external archive to store the non-dominated solutions found during the search. The library we use for the implementation of SMPSO is jMetalPy [70] which follows the algorithm pseudocode described in the algorithm templates section of the paper [71].

3.3.3 *Differential Evolution*

Differential Evolution (DE) is a population-based stochastic search method, designed to solve continuous optimisation problems, and able to handle non-differentiable, nonlinear and multimodal objective functions [73]. DE is amongst the state-of-the-art evolutionary algorithms for continuous optimisation and has been successfully applied to a variety of problems in science and engineering [74]. There is growing evidence supporting the excellent performance of DE in terms of accuracy, convergence speed and robustness, in domains including electronics, manufacturing, machine learning, bioinformatics and biomedical engineering [74], [75].

A feature of DE, distinguishing it from other evolutionary algorithms, is its *differential* mutation operator. Given a population of candidate solutions in \mathbb{R}^n a new mutant vector \bar{x}' is produced by adding a perturbation vector to an existing one, $\bar{x}' = \bar{x} + \bar{p}$, where the perturbation vector \bar{p} is the scaled

vector difference of two other, randomly chose population members $\bar{p} = F \times (\bar{y} - \bar{z})$. The other reproduction operator is the uniform crossover, subject to a crossover rate parameter $Cr \in [0, 1]$. In general, a DE algorithm has three control parameters, the scaling or mutation factor F , the population size NP and the crossover rate Cr .

The software library used for the experiments running DE in this work is SciPy [76], where the implementation follows [73].

3.3.4 *Covariance Matrix Adaptation Evolution Strategy*

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is an evolutionary algorithm for difficult non-linear non-convex black-box optimisation problems in the continuous domain and is considered a state-of-the-art algorithm. CMA-ES is usually applied to unconstrained or bounded constraint (having an upper and/or lower bound value) problems. The algorithm uses a second-order approach and estimates a positive definite matrix within an iterative procedure — more specifically, a covariance matrix (a square matrix giving the covariance between each pair of elements of a given random vector). According to Hansen et al. this makes the method feasible on non-separable and/or badly conditioned problems, non-smooth and even non-continuous problems, as well as on multimodal and/or noisy problems [77]–[82].

It is beneficial that the CMA-ES implementation in `pycma` [83] used to run the experiments in Chapter 6 does not require a lot of parameter tunings in order to be implemented. This is due to the fact that the developers wanted the strategy parameters to be part of the algorithm design, and not the application — the aim being to have a high-performing algorithm out-of-the-box. The starting population size is always small, allowing a fast convergence. There is an automatic termination criterion implemented, but it could be overwritten based on iterations or fitness function evaluations count [83].

3.4 LITERATURE REVIEW

Evolutionary optimisation techniques have been used to solve real-world problems in the health sector. In the following paragraphs, we will look into some recent studies where single-objective and multi-objective approaches were used for drug treatment optimisation or classification.

3.4.1 *Single-Objective Optimisation*

Other groups of authors have used genetic algorithms to optimise antibiotic dosing regimens. Cicchese et al. [84] use genetic algorithms and surrogate-assisted optimisation to design regimens to treat Tuberculosis infections. Their formulation assumes that doses are fixed across the treatment, and instead vary the frequency of application of multiple drugs. The single objective function has two terms measuring the average time to eradication and the dose size and frequency of antibiotics. Treatments are evaluated using a hybrid, multiscale model that combines agent-based modelling with differential equations, and a pharmacokinetic model.

Colin et al. [85] use a genetic algorithm to optimise a dosing guideline for intermittent infusion of vancomycin in adults. They encode dosing regimens as combinations of discretised loading doses, maintenance doses and dosing intervals. Although the loading and maintenance doses can vary across candidate solutions, a given solution holds the same loading and maintenance dose with varying dosing intervals. The formulation uses a single objective function with several constraints, and only focuses on the pharmacokinetic model (antibiotic concentrations), without explicitly modelling the bacteria infection, to simulate an adult patient population.

3.4.2 *Multi-objective Optimisation*

Petrovski and McCall [86] propose another use of multi-objective optimisation for cancer treatment, where the goal is to minimise the size of the tumour/s and maximise the patient survival time. A Strength Pareto Evolutionary Algorithm (SPEA) approach was used to find the non-dominated chemotherapy schedules, where the final decision-making is left to oncologists. The multi-objective approach looked into multi-drug combinations where each of the drugs has a different dose during the treatment. The study presents that the evolutionary algorithm was able to detect solutions that were missed by other optimisation techniques. A follow-up paper [87] compared the results from a Particle Swarm Optimisation (PSO) and a Genetic Algorithm (GA), where the PSO algorithm produced better treatment schedules.

Petrovic et al. [88] present a multi-objective optimisation model for scheduling radiotherapy treatments for categories of cancer patients. The model was developed for a real-life treatment process in the UK, where the constraints considered staff rota, machine availability, waiting time, etc. Two objectives were defined - minimising the waiting time and minimising the breaching of waiting time targets. Standard-GA, knowledge-based (KB)-GA and weighted-GA are implemented and compared, where the KB-GA was the best performing one.

Bevilacqua et al. [89] present a new approach to artificial neural network topology optimisation using a multi-objective genetic algorithm to find the best network configuration for a breast cancer database classification problem where two classes of tumours are considered - benign and malignant. The first step of the approach uses a genetic algorithm (GA) to find the optimal topology for the problem, and the second step of the research uses a multi-objective GA to refine the topology space. After the space was refined, a neural network was used to classify the tumours.

Cicchese et al. [90] present a new treatment strategy for prescribing antibiotics for treating tuberculosis using an agent-based model capturing tuberculosis

granuloma formation with algorithms for mathematical optimisation aiming to identify the optimum treatment regimens. A Genetic Algorithm was used, as well as a radial basis function (RBF) neural network, where they found that the RBF network surrogate model was more suitable for predicting the optimal treatments. In their previous work [91] a PK/PD model was already implemented that captured the distribution and oral administration of the antibiotics.

In 2007 Ochoa et al. [92] use the employment of evolutionary algorithms as a decision system for designing chemotherapy schedules, where the schedule is formulated as an optimal control problem. A mathematical model is used for simulating the tumour growth during the chemotherapy schedules. The objective of the study is to find effective drug schedules that help eradicate the tumour while maintaining the patient's health above the acceptable level.

In 2013, another study by Ochoa and Villasana [93] uses population-based algorithms for designing combination cancer chemotherapies. The chemotherapy sessions schedule is expressed as an optimisation problem with the main objective of minimising the tumour size while minimising the compromise of the patient's health. A mathematical model similar to the one in [92] was used to describe the tumour's progression and evaluate the solution. Three algorithms were used - a differential evolution (DE) algorithm, Covariance Matrix Adaptation (CMA) evolution strategy and a particle swarm pattern optimisation algorithm.

Villasana et al. [94] published another study using the mathematical model of cancer cytotoxic chemotherapy from [92] where a new drug type was considered - a cytostatic agent. This type of drug has the effect of arresting cells in a phase of their cycle and then being targeted with a cytotoxic agent with the objective of maximising cell killing fraction and minimising normal cell killing. In comparison to treatments only using the cytotoxic agent, the incorporation of the cytostatic agent drastically improved the performance of the model.

Luong et al. use benchmark multi-objective evolutionary algorithms for high-dose-rate brachytherapy planning for prostate cancer treatment [95]. Four different MOEAs are used on the problem - Non-dominated Sorting Genetic Algorithm II (NSGA-II), Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), Multi-Objective Adapted Maximum-Likelihood Gaussian Model Iterated Density-Estimation Evolutionary Algorithm (MAM-aLGaM) and Multi-Objective Real-Valued Gene-pool Optimal Mixing Evolutionary Algorithm (MO-RV-GOMEA). The results show that for this problem, the recently proposed 2015 MO-RV-GOMEA [96] is the best-performing algorithm.

Part II

MY CONTRIBUTION

EXPLORING MULTI-OBJECTIVE EVOLUTIONARY OPTIMISATION

4.1 INTRODUCTION

This chapter explores optimising antibiotic treatments, which are effective while also short and using the least amount of drug possible. We extend a mathematical model that simulates the progression of a bacterial infection, first introduced by Paterson et al. [12] where a single-objective evolutionary algorithm was used to design effective treatments. Ochoa et al. also applied a multi-objective evolutionary algorithm in order to automatically design successful antibiotic treatments where constraints and objectives are combined. [42].

The main contributions of this chapter are as follows:

- To extend the mathematical model used in [12], [42] with a pharmacokinetics/pharmacodynamics (PK/PD) component modelling the antibiotic absorption from the stomach to the blood flow.
- To use a Tau-leaping approach to speed up the simulation time of the stochastic bacteria population model.
- To contrast two population-based algorithms (evolutionary vs. particle swarm optimisation) in the task of optimising a multi-objective formulation of the treatment design problem.
- Analyse the best performing treatments in terms of effectiveness and establish common trends.

4.2 MATHEMATICAL MODEL

We followed the formulation detailed in [12] and [42] and in Chapter 2, where a mathematical (stochastic) model of the progression of a bacterial infection, and the effect of antibiotic treatment, is proposed.

In this chapter, we consider a vector of real numbers to encode treatments, $x_i \in \mathbb{R}$, instead of a vector of integer numbers, $x_i \in \mathbb{Z}$ as was the case in [12] and [42]. This allows us to explore more precise prescriptions, which may be relevant considering the recent trend in personalised medicine. There is also only one type of bacteria instead of the two (resistant and susceptible) in [12].

The model follows similar steps to the ones in Chapter 2:

- Antibiotic dose for the day is taken. The following steps happen until the next dose needs to be taken:
 - Probabilities for bacteria's population death and reproduction are calculated.
 - Bacteria population is updated based on the probabilities.
 - Time increases by a time-step of 15 minutes.
 - New concentration of the antibiotic in the body is calculated.
- Next dose is taken.

The model is detailed in Algorithm 1 where the steps listed above are described further.

4.2.0.1 *Parameters and Equations*

The equations below show how the reproduction rate and the death rate of the bacteria population are calculated. Table 4.1 provides a breakdown with the parameter description and values. All but α , g and p parameter values are taken from [12]. The rest of the parameter values were chosen by Paterson et al. such that as the concentration of antibiotics increases, the death rate

will increase as well until it reaches a saturation point. The concentration of antibiotics naturally decays within a host, so that was taken into account when choosing the parameters' values.

$$\text{ReproductionRate}_{(\text{bacteria})} = rB \left(1 - \frac{B}{K} \right) \quad (4.1)$$

$$\text{DeathRate}_{(\text{bacteria})} = mB + \frac{(\max - \min) \left(\frac{C_{\text{Blood}}}{\text{mic}} \right)^k}{\left(\frac{C_{\text{Blood}}}{\text{mic}} \right)^k \frac{\min}{\max}} B \quad (4.2)$$

Table 4.1: List of parameters and values. All the values except for a , g and p are taken from [12].

| Parameter | Description | Value |
|--------------|--|--------|
| r | Replication rate of bacteria B | 2.7726 |
| K | Carrying capacity | 1000 |
| m | Mortality rate of bacteria | 0.2 |
| a | Degradation rate of antibiotics in the stomach | 0.6 |
| g | Degradation rate of antibiotics in the blood | 0.4 |
| p | Proportion of antibiotics that reaches the blood | 0.54 |
| \max | Max net growth rate in absence of antibiotics | 2.5 |
| \min | Min net growth rate at high antibiotic levels | -2.1 |
| mic | Min inhibitory concentration (MIC) | 16 |
| k | Hill coefficient | 4 |

4.2.0.2 Pharmacokinetics/Pharmacodynamics

Pharmacokinetics/pharmacodynamics (PK/PD) modelling is the basis of modern-day pharmacotherapy. Pharmacokinetics describes the drug concentration over time as it courses in the body/host, while pharmacodynamics observes the effects resulting from the certain concentration of the drug present

in the body/host. In other words, pharmacokinetics answers the question ‘what the body does to the drug’, while pharmacodynamics — ‘what the drug does to the body’ [97] [98].

In our approach, the PK/PD model is used when calculating the concentration of antibiotics when the daily dose is administered orally. We represent this concentration as C_{Stomach} , and its slow decrease during the simulation is calculated by the following equation:

$$\frac{dC_{\text{Stomach}}}{dt} = -aC_{\text{Stomach}} \quad (4.3)$$

However, for the antibiotics to be effective, they need to reach the bloodstream where they can fight the bacterial infection. The equation below calculates the level of antibiotics in the blood during the day:

$$\frac{dC_{\text{Blood}}}{dt} = p a C_{\text{Stomach}} - g C_{\text{Blood}} \quad (4.4)$$

The parameter values for a , g and p were chosen so that the maximum concentration in the blood corresponds to the half-life time of the drug. A curve fitting analysis was done to find the best values for a , g and p , so that the half-life of the antibiotics is about half a day and there are still some antibiotics left in the system when the next dose is taken, when the dose is 60mg (the maximum dosage).

As it could be observed from Equation 4.2 the death rate of the bacteria is correlated with the concentration of the antibiotics — the higher the concentration, the higher the death rate is.

4.2.0.3 *Tau-leaping*

To save computation power and produce faster results, a Tau-leaping approach is taken in this work, as proposed by Andrew Hoyle. Tau-leaping is an approximate method for the simulation of a stochastic system based on the Gillespie algorithm [99]. In Algorithm 1 we use a time-step of 15 minutes for the approximation, as this value gives a good balance for speeding up the process without

losing too much accuracy. For every time-step, an approximation is calculated for how much the bacteria population has decreased (`bacteria_decrease`) and increased (`bacteria_increase`). Then the whole bacteria population is updated by summing the leftover bacteria from the previous time-step, adding the `bacteria_increase` and subtracting the `bacteria_decrease`. In Equation 4.5 below, the mathematical formula is shown for that process, which also corresponds to lines 15 to 17 in Algorithm 1.

$$B(t + \tau) = B(t) + \text{Poisson}(\tau \text{ReproductionRate}_{(\text{bacteria})}) - \text{Poisson}(\tau \text{DeathRate}_{(\text{bacteria})}) \quad (4.5)$$

4.2.0.4 Objectives

We consider a bi-objective formulation of the problem of optimising antibiotic treatments. Specifically, the two objective functions to be minimised are:

- The percentage of simulation runs where the bacteria survives the treatment — *failure rate* f_r .
- The total amount of antibiotics used, as measured by the sum of the entries in the dosage vector A_{total} .

4.3 COMPUTATIONAL METHODS

4.3.1 Implementation of the objectives

The two objectives described in Section 4.2.0.4 — *failure rate* f_r and the total amount of antibiotics taken A_{total} . The failure rate is estimated by running the mathematical model with a fixed number of simulations and returning the number of runs in which the bacteria population was not eliminated. For example, the treatment vector $x = (44.92, 60, 53.17, 39.25, 60, 1.70)$ has a $A_{\text{total}} = 259.04\text{mg}$ and $f_r = 0.0391$ which could be read as $f_r = 3.91\%$.

4.3.2 *Noisy Failure Rate*

The failure rate is determined by the stochastic model and due to the random elements in the approach, one treatment could have different outcomes per run — failure (the bacteria population does not get eradicated) or success (bacteria population is eradicated). The model is run 500 times for the same treatment, and the failure rate is determined by how many times the outcome was a failure. The number of model runs had to be carefully selected to minimise the noise, but also minimise the number needed to reduce the computational effort. After preliminary investigations, we have chosen to use 500 model simulations for estimating the failure rate for the candidate treatment solutions.

4.3.3 *Population-based algorithms*

We considered two population-based algorithms, as described below. Our implementation used the Python library JmetalPy [70].

4.3.3.1 *Non-dominated Sorting Genetic Algorithm II (NSGA-II)*

NSGA-II uses non-dominated sorting of individuals in the population, with a crowding distance penalty applied to individuals to maintain a diverse Pareto-front [68]. As it is one of the best-known and widely used algorithms, and was previously used for optimising antibiotic treatments in [42] and we wanted to see its performance with the PK/PD element added to the mathematical model.

4.3.3.2 *Speed-constrained Multi-objective Particle Swarm Optimisation Algorithm (SMPSO)*

SMPSO [71], [72] is a multi-objective particle swarm optimisation algorithm that uses a strategy to limit the velocity of the particles. SMPSO produced remarkable results when compared to NSGA-II on a number of standard

benchmark functions, and we wanted to test that for our problem. It was one of the most recent additions to the multi-objective algorithms in the JmetalPy library.

4.3.4 *Hypervolume Indicator*

To compare the two algorithms' performance, we use the hypervolume indicator. The hypervolume indicator is a set measure to evaluate the performance of multi-objective algorithms using a reference point. In our case, this is the solution with maximum total antibiotics and failure rate, as we are minimising our objective functions. This solution is a treatment of the maximum 10 days period with the maximum dosage of 60: $\underline{x} = [60, 60, 60, 60, 60, 60, 60, 60, 60, 60]$ that will produce the maximum possible $A_{\text{total}} = 600\text{mg}$ and $f_r = 100\%$. The reference point to the Pareto-front space is measured to produce a single number — the hypervolume indicator. As our two objective functions need to be minimised, the higher the hypervolume indicator, the better the solution is. The implementation of the calculation for the hypervolume used in this work is the one from Fonseca et al. [100].

4.3.5 *Parameter Tuning*

We used the same configuration effort to tune the multi-objective optimisation algorithms in the design of antibiotic treatments. We applied an automatic configurator to find the highest-performing configurations of the algorithms to avoid a bias in the performance of the techniques and to develop a fair comparison.

We selected the software package irace [101]. This software has been applied to a wide variety of configuration tasks, which include not only tuning the numeric parameters of multi-objective optimisations but also designing automatically new multi-objective optimisation algorithms [102]. We only

configured the numerical parameters of the algorithms by using the iterated F-race implemented in irace. It is important to note that the implementation of the parameter tuning was developed by Marco A. Contreras-Cruz.

In iterative F-races, configurations are sampled according to a particular distribution (that evolves with time), and the configurations are evaluated with a racing method to find the optimal configuration. The racing consists of evaluating the performance of the configurations with a sequence of training instances and developing the Friedman test (a statistical test that is used to detect differences in variables across multiple test attempts [103]) to remove statistically worse configurations. This process is repeated until a stopping criterion is met; for example, a maximum number of executions of the algorithm.

During the configuration, the NSGA-II algorithm used the SBX crossover, the polynomial mutation, and the binary tournament selection with the ranking and crowding distance comparator, while the SMPSO algorithm used the polynomial mutation operator and the leader replacement based on the crowding distance archive. For the configuration of the numerical parameters, we used the iterated F-race with 120 executions of the algorithm. Each run of the algorithm executed a maximum number of 5000 function evaluations, with 500 runs of the mathematical model. The real numbers consider four decimal places in the configuration. Tables 4.2 and 4.3 describes the numerical parameters of both algorithms and the results of the tuning procedure.

4.4 RESULTS

4.4.1 Hypervolume Comparison of Algorithms

Figure 4.1 shows the hypervolume results of 30 runs for each of the multi-objective algorithms — NSGA-II with default parameters, NSGA-II with tuned parameters, SMPSO with default parameters and SMPSO with tuned parameters.

Table 4.2: Description of the parameters of NSGA-II and parameters values found by the iterated F-race. *Population size* and *Offspring population* values $\in \mathbb{Z}$ while the rest $\in \mathbb{R}$.

| Parameter | Domain | Default | Tuned |
|------------------------------|--------------|---------|----------|
| Population size | (30,100) | 100 | 62 |
| Offspring population | (30,100) | 100 | 70 |
| Mutation probability | (0.0,1.0) | 0.1 | 0.0971 |
| Mutation distribution index | (5.0, 400.0) | 20 | 306.2005 |
| Crossover probability | (0.0, 1.0) | 1 | 0.5084 |
| Crossover distribution index | (5.0, 400.0) | 20 | 128.7306 |

Table 4.3: Description of the parameters of SMPSO and parameters values found by the iterated F-race. *Swarm size* and *Size of the archive* parameter values $\in \mathbb{Z}$ while *Mutation probability* and *Mutation distribution index* parameters values $\in \mathbb{R}$.

| Parameter | Domain | Default | Tuned |
|-----------------------------|--------------|---------|----------|
| Swarm size | (30,100) | 100 | 59 |
| Mutation probability | (0.0,1.0) | 0.1 | 0.2821 |
| Mutation distribution index | (5.0, 400.0) | 20 | 307.3271 |
| Size of the archive | (30, 100) | 100 | 56 |

4.4.1.1 NSGA-II and SMPSO Performance

From Figure 4.1 we can clearly observe that the solutions produced by NSGA-II performed better in regards to the hypervolume compared to SMPSO. The median for the hypervolume for all 60 runs (both tuned and with default parameters) of NSGA-II is 432.8654 and the one for SMPSO is 429.9859. The p -value = $2.2e - 16$ for Welch Two Sample t-test which is used to test the hypothesis that two populations have equal means is significantly lower than

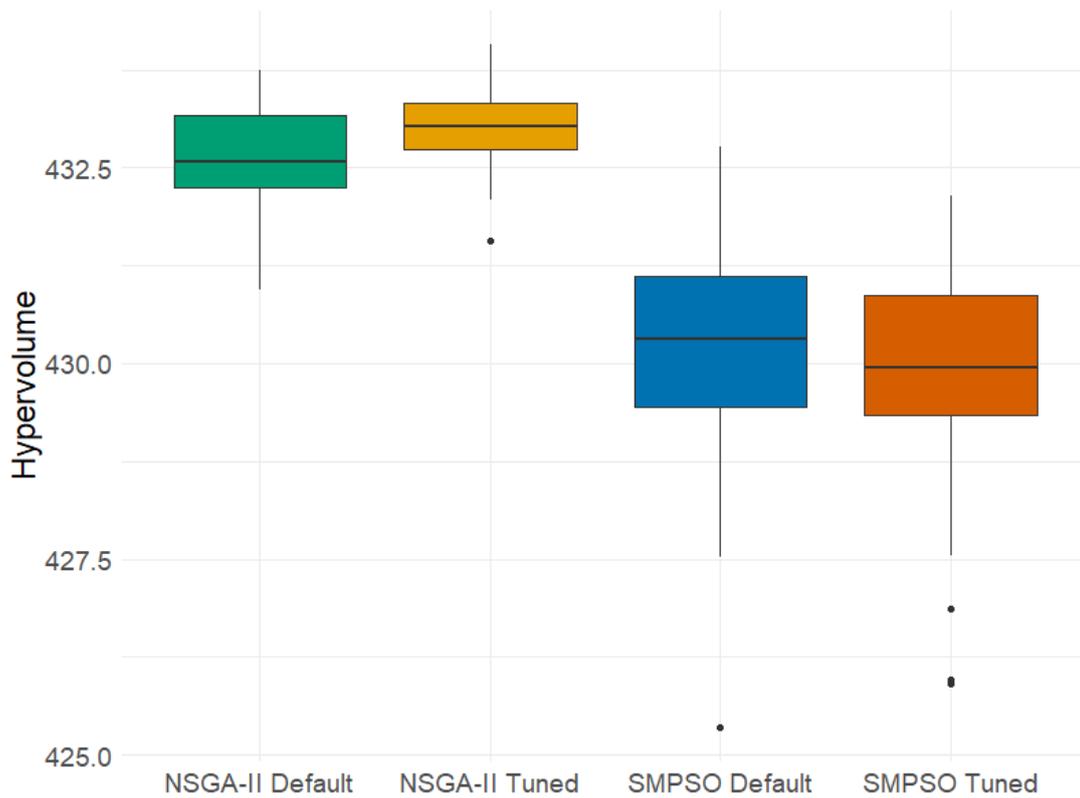


Figure 4.1: A boxplot of the hypervolume of 30 runs for each of the algorithms - NSGA-II with default and tuned parameters and SMPSO with default and tuned parameters.

0.05 showing that the solutions between the two algorithms are significantly different from one another. The p - value = $2.2e - 16$ for Wilcoxon Signed Rank Test that checks whether two samples follow the same distribution also proves the same point. The p - value = $4.441e - 16$ for Kolmogorov-Smirnov Test used to compare the mean of two samples produced proves the same hypothesis.

4.4.1.2 NSGA-II - tuned and with default parameters

It appears that the tuned NSGA-II performs slightly better than the default parameters NSGA-II in terms of hypervolume results. The median of the tuned NSGA-II is 433.00 and the default parameter NSGA-II's median hypervolume is 432.60. The best Pareto-front hypervolume indicator is significantly higher

as well, the outlier is still better than the lowest-performing Pareto-front by the default parameter NSGA-II. This result was expected and according to the $p - \text{value} = 0.01258$ from Welch Two Sample t-test. This proves the hypothesis that the tuned NSGA-II performs better than the default parameter one. Running the Wilcoxon Signed Rank Test the $p - \text{value} = 0.01317$ and Kolmogorov-Smirnov Test the $p - \text{value} = 0.03458$. The $p - \text{values}$ from the two tests are still less than 0.05 so this supports the hypothesis as well.

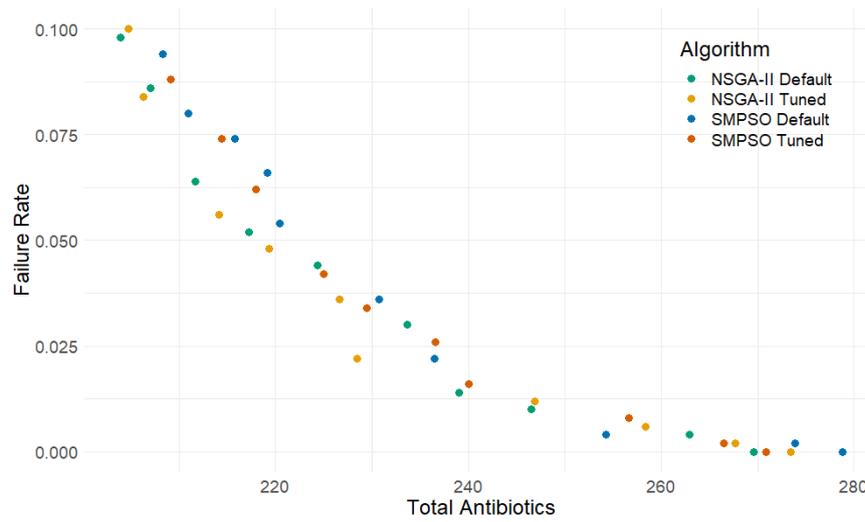


Figure 4.2: A partial view of the Pareto-front. All solutions with Failure Rate $\leq 10\%$ (0.10) and Total Antibiotics $\leq 300\text{mg}$ from 120 runs of the full multi-objective model using the population-based algorithms NSGA-II and SMPSO. The figure combines runs from both default parameters and tuned parameter runs of the two algorithms.

4.4.1.3 SMPSO - tuned and with default parameters

In the case of SMPSO, the tuned algorithm does not perform significantly better than the default parameter. The median of the tuned SMPSO is 429.75 and the median of the default parameter SMPSO is 430.09. The $p - \text{value} = 0.4189$ from Welch Two Sample t-test proves that the tuned SMPSO is not performing better in comparison with the default parameter one. The Wilcoxon Signed Rank Test ($p - \text{value} = 0.7091$) and Kolmogorov-Smirnov Test ($p - \text{value} = 0.808$) show that we cannot prove that there is a big difference in performance

between the two. This indicates that maybe a bigger budget for parameter tuning is needed when tuning a PSO-based algorithm.

4.4.2 *Partial Pareto-front of best results for NSGA-II and SMPSO*

When comparing the performance of the two algorithms from the boxplot in Figure 4.1 we can see that NSGA-II outperforms SMPSO. The median of the hypervolume indicator for all NSGA-II runs is 432.8017 and the hypervolume indicator median for SMPSO is 429.9207.

Figure 4.2 shows only a selected part of the Pareto-front with only the 10 best solutions obtained from the 30 runs of each of the algorithms (NSGA-II with default parameters and with tuned, SMPSO with default parameters). By best, we mean the top 10 ranking solutions according to the lower obtained value of the **Failure Rate (%)**. These solutions all have Failure Rate $\leq 10\%$ and Total Antibiotics (the sum of all dosages in the solution vector) $\leq 280\text{mg}$. The NSGA-II with tuned parameters points coloured in yellow is visibly producing the most non-dominated solutions followed by NSGA-II with default parameters, and then the green points (NSGA-II with default parameters). This again adds evidence for the hypothesis suggested at the beginning of Section 4.4 that NSGA-II outperforms SMPSO.

4.4.3 *Best Treatments*

Table 4.4 shows the top 10 overall best solutions with Failure Rate $\leq 0.10\%$ — we chose this as the threshold to remain consistent with previous research done by Paterson et al. [12] as we are using the same parameters for the mathematical model. The solutions are ordered by Re-Evaluated Failure Rate from lowest to highest. Here, we refer to best as the solutions with the lowest Failure Rate and lowest Total Antibiotics for that rate.

Table 4.4: The best 10 treatments obtained by our approach. They all have a Failure Rate between 0 and 0.10. The treatments are ordered by *Re-eval. Failure Rate* starting from the lowest. The length of the treatment, as well as the algorithm that has produced the result, is provided.

| Treatment Vector | Length (days) | Total Antibiotics | Failure Rate (%) |
|--|---------------|-------------------|------------------|
| 56.46, 46.70, 51.21, 41.14, 31.05, 46.82 | 6 | 273.38 | 0 |
| 59.89, 41.88, 38.62, 56.52, 25, 46.63 | 6 | 268.54 | 0 |
| 44.92, 60, 53.17, 39.25, 60, 1.70 | 6 | 259.04 | 0.4 |
| 58.01, 45.78, 55.60, 40.94, 51.85 | 5 | 252.18 | 0.8 |
| 59.95, 44.53, 52.70, 36, 55.05, 0, 7 | 7 | 255.23 | 0.6 |
| 58.75, 47, 57.36, 37.71, 37.71 | 5 | 238.53 | 1.4 |
| 53, 53, 54.01, 29, 54.47 | 4 | 243.48 | 1.2 |
| 57.14, 50, 45.66, 49.93, 20.67, 13.56 | 6 | 236.96 | 0 |
| 56.18, 44.96, 45.43, 42, 43 | 5 | 231.57 | 1.8 |
| 58, 43.43, 46.99, 52, 25.20 | 5 | 225.62 | 3.2 |

When designed, the treatments could be of lengths between 3 and 10 days, and the upper bound for a single dosage is set to 60mg. In the course of the mutation during the iterations of the multi-objective algorithms, some mutations could occur when a single daily dose is 0.332mg. In cases where there is a dosage below 1mg, it has been rounded down to 0mg as it makes little difference to the failure rate. For example, the treatment $\underline{x} = (59.892, 41.879, 38.623, 56.518, 25, 46.632, 0.558, 0.48, 0, 0.004)$, where the initial length of the treatment is 10 days and the last treatment is so close to 0mg that it has been rounded down. After rounding up the dosages to the second decimal this treatment is listed in the table as $\underline{x} = (59.89, 41.88, 38.62, 56.52, 25, 46.63)$ where the final length is 6 days.

All the solution vectors listed in Table 4.4 were re-evaluated using the mathematical model with 10,000 runs. It was expected for some differences in

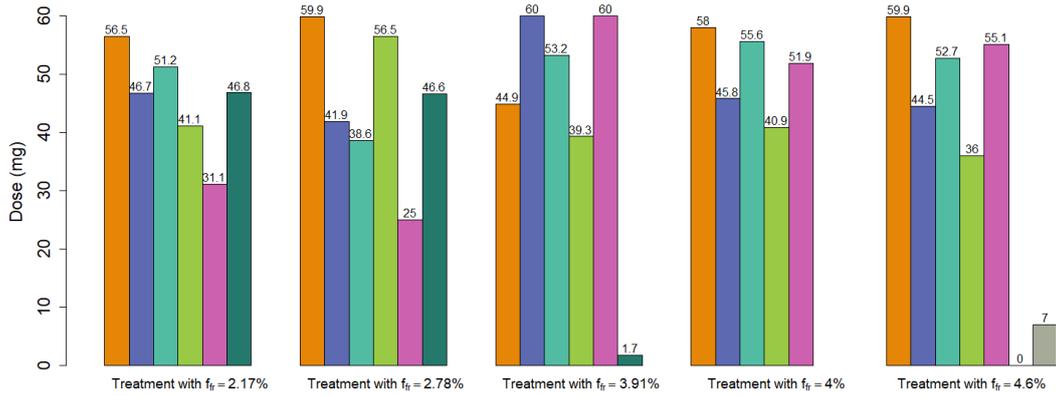


Figure 4.3: Barplot representation of some of the treatments with the lowest failure rate after re-evaluation. Each of the bars corresponds to a single day in the treatment.

the Failure Rate to be present due to the noise produced by the stochastic characteristic of the mathematical model. During the iterations of the two multi-objective algorithms, the mathematical model was run 500 times which could contribute further to the noise in the failure rate evaluation.

In Table 4.4, the top 10 of the solutions have a difference of over 5% between the failure rate and the re-evaluated failure rate which is a lot higher than expected but could be explained by the stochastic characteristic of the mathematical model.

Another observation about the solutions listed in Table 4.4 is that only two out of the ten were generated using the SMPSO algorithm (one solution with tuned parameters and one with default parameters). Five out of the ten solutions were generated with NSGA-II with default parameters and three with tuned parameters. This is another indication of the better performance of NSGA-II over SMPSO.

Figure 4.3 shows the top 5 treatments from Table 4.4, which have produced the lowest failure rate after the re-evaluation. The Total Antibiotics for the treatments presented are as follows: 273.4mg, 268.5mg, 259.1mg, 252.2mg and 255.2mg. Each of the treatments is represented by a barplot where each of the

bars is a separate day from the treatment. The dose for the day could be seen at the top of each of the bars, and the failure rate of the treatment is provided below the barplot for that treatment.

What we can observe from Figure 4.3 is that three out of the five treatments alternate between a very high dosage (between 50mg and 60mg) and a lower dosage. The pattern observed in the top solutions in the previous studies [12], [42] was towards tapered doses, where the first dose will be the highest and every dose after it will be lower than the previous. This difference in dose patterns could be explained by the introduction of the PK/PD model, as the concentration of antibiotics in the body is modelled differently, impacting the $\text{DeathRate}_{\text{bacteria}}$ calculation. As a higher dose of antibiotics stays longer in the system due to the PK/PD mechanic, the next dose can be smaller, and the antibiotics could still be effecting throughout the treatment.

4.5 CONCLUSION

The proposed approach looks into the problem of overuse of antibiotics and more specifically optimising the number of antibiotics prescribed and the length of the overall treatment. The automatic design of possible treatments and their evaluation has little constraints at the moment — upper and lower limits on the treatment (3 to 10 days) and upper and lower limits on the daily dosages (0mg to 60mg) where the bacteria levels are always the same.

In our study, we introduced new techniques to the mathematical model — the PK/PD modelling and switched from using the standard Gillespie Algorithm to another variation of the Gillespie Algorithm — Tau-leaping for predicting events. Then two population-based multi-objective algorithms were chosen — NSGA-II and SMPSO for designing the antibiotic treatments. Both of the algorithms were then tuned and the hypervolume for each of the runs of the algorithms was calculated. Those hypervolume indicators were then compared as well as some of the best solutions. What could be concluded from the results

is that the NSGA-II algorithm provided better results than SMPSO. There was also not a significant improvement upon tuning the algorithms in terms of results at the end of the budget of 5000 iterations.

In this model, the patient's profile (overall health, diet, and other possible medical conditions) and correct usage of the antibiotics are not taken into account even though they play a big factor when fighting bacterial infections. These points will be investigated for the future versions of this model as well as including more objectives when designing the treatments.

Algorithm 1 Outline of the simulation model

```
1: treatment = { $x_1, x_2, \dots, x_n$ }
2: bacteria = 1000
3: concentrationblood = 0
4: concentrationstomach = 0
5: time = 0
6: timestep = 15 minutes
7: end_of_day = 1440 minutes
8: for day 1 until the last day of treatment do
9:   concentrationstomach = concentrationstomach + treatment $x_{day}$ 
10:  concentrationstomach0 = concentrationstomach
11:  concentrationblood0 = concentrationblood
12:  while time  $\leq$  end_of_day or bacteria = 0 do
13:    reproduction_rate is calculated using Equation 4.1.
14:    death_rate is calculates using Equation 4.2.
15:    bacteria_increase = Poisson(timestep  $\times$  reproduction_rate)
16:    bacteria_decrease = Poisson(timestep  $\times$  death_rate)
17:    bacteria = bacteria + bacteria_increase - bacteria_decrease
18:    time = time + timestep
19:    concentrationstomach is calculated using Equation 4.3.
20:    concentrationblood is calculated using Equation 4.4.
21:  end while
22: end for
```

SINGLE OBJECTIVE ALGORITHM WITH CONSTRAINT APPROACH

5.1 INTRODUCTION

In this chapter, we use our most recent bacterial infection model [13] presented in Chapter 4, but make two fundamental changes to the optimisation problem formulation. Firstly, the total amount of antibiotics is treated as a constraint rather than an objective. The single objective to be minimised is the regimen failure rate. This is because reducing the failure rate is the prominent aim of any successful treatment while reducing the total amount of antibiotic used is a secondary goal. Therefore, exploring the whole trade-off of these two goals, as it is done by multi-objective evolutionary algorithms, is not interesting in practice. The second fundamental change relies on the representation of candidate dosing regimens. We use real-numbers for representing daily dosage — we argue that this encoding allows the exploration of a wider search space of possible dosing regimens. Moreover, this encoding prompted us to use an evolutionary algorithm specifically tailored to continuous optimisation (differential evolution [73], [75]), rather than standard genetic algorithms. This chapter also departs from previous work as we experiment with varied levels of bacterial resistance, coinfections with two strains of bacteria, and two ways of administering the drug: orally and intravenously. We contrast the optimised dosing regimens against the standard practice of fixed-daily doses with the same total amount of antibiotics. In summary, this chapter is guided by the following research questions.

How do optimised regimens vary according to:

1. *antimicrobial resistance level?*
2. *form of administering antibiotics, orally vs. intravenously?*
3. *presence of a single bacterial strain vs. two strains of bacteria with different resistance levels?*

5.2 METHODOLOGY

5.2.1 *Biomedical Background*

Once an antibiotic is chosen, conventional treatments have three main characteristics: the concentration of each dose, the time interval between doses and the total number of doses given. These characteristics are usually decided by the manufacturer or a health body and usually consist of fixed-sized doses at fixed time intervals. For example, a course of *Amoxicillin* may be 250 mg taken 3 times daily for 5 to 7 days [104]. While these fixed-dose treatments may be effective, they may not be the optimal dose or duration to administer the antibiotic most efficaciously. Although we are taking a theoretical approach, our parameters are 'loosely' based around an *E. coli* UTI infection being treated with *Amoxicillin*. This is in comparison to some of the previous work, where parameters were more arbitrarily chosen [12], [13], [42].

Dose regimens are often based on Pharmacokinetics and Pharmacodynamics studies of target populations. One significant characteristic of the bacterial population is the minimum inhibitory concentration (MIC). This is the lowest concentration (in $\mu\text{g}/\text{mL}$) of an antibiotic that inhibits the population growth of a given strain of bacteria. In this work, we have chosen four MIC values: sensitive, 8 $\mu\text{g}/\text{mL}$, intermediate, 16 $\mu\text{g}/\text{mL}$ and 24 $\mu\text{g}/\text{mL}$, and resistant, 32 $\mu\text{g}/\text{mL}$ [105].

For short-term infections, there is often only one type of bacteria present in the host, which is the case for most healthy people when they are suffering from a bacterial infection. Later in this chapter, we consider immunosuppressed hosts. Here, the body is more susceptible to infection, and this could result in the host having multiple bacterial infections at the same time. Alternatively, two strains can also be present when a mutation occurs to create a more antibiotic-resistant one.

5.2.2 *Mathematical Model*

The mathematical model used follows a similar formulation as in [12], [13], [42], [106], where a population of bacteria is simulated with a Markov chain approach using the Gillespie algorithm [99], and the effect of an antibiotic treatment to eradicate the infection is considered as detailed below. The model simulates the bacteria population through the duration of treatment plus an extra 3 days to allow the antibiotic in the blood to dissipate and to establish if the bacteria population has reached the count of 0 (treatment is successful) or not (treatment is not successful, or failed). A n -day treatment is denoted as a vector $\bar{x} = (x_1, x_2, \dots, x_n)$, where x_i represents the dosage taken on day i , with $x_i \geq 0$. In this formulation, x_i are real positive numbers, $x_i \in \mathbb{R}$. The maximum total antibiotic, $\sum_{i=1}^n x_i$, is selected based on the amount needed to cure the host with a fixed daily-dose regimen, for the specific MIC of the bacteria determined empirically using the model. The time interval between doses is fixed at 24 hours throughout this chapter.

Bacteria modelling (one strain). Where a single type of bacteria is present in the host, there are two events that happen: the birth of bacteria (p_1) and death of bacteria (p_2) shown in Table 5.2. In p_1 , the term rB represents the bacteria's binary fission for the time step, producing exponential growth at rate r - the carrying capacity of the host is removed, as the host is likely to die before the bacteria population reaches the carrying capacity [107]. Instead, the host

dies when a fixed number of bacteria is reached, set as 2000. In p_2 , we sum the natural death of the bacteria, due to the host's immune system m , and antibiotic-induced death rate represented by parameters b_1 , b_2 , mic and k .

The pseudocode, for the one-strain model, can be seen in Algorithm 2, with parameters presented in Table 5.1.

Table 5.1: Mathematical model parameter values and references.

| Parameter | Description and reference | Value |
|-----------|--|---------------------|
| a | Absorption rate of antibiotics in the stomach [108] | 33.27 |
| g | Degradation rate of antibiotics in the blood [108] | 1.11 |
| p | Proportion of antibiotics that reaches the blood [108] | 0.95 |
| m | Immune system response rate [109] | 0.1 |
| r | Replication rate of bacteria [110] | 0.5 |
| b_1 | Maximum kill rate of the antibiotic (as $C_b \rightarrow \infty$) | 2.5 |
| b_2 | Level of antibiotic giving half max kill rate [12] | $1.5137 \times mic$ |
| mic | Min inhibitory concentration (MIC) | 8,16,24 or 32 |
| k | Hill coefficient in antibiotic induced death | 4 |

Algorithm 2 Outline of the stochastic mathematical model with one bacteria strain.

```
1: treatment = { $x_1, x_2, \dots, x_n$ }
2: initial bacteria population  $B = 700$ 
3: initial antibiotic concentrations  $C_B = 0, C_S = 0$ 
4: time = 0 minutes
5: time step  $\tau = 15$  minutes
6: end_of_day = 1440 minutes (= 24 hours)
7: deadly_level_bacteria = 2000
8: for day 1 until the last day of treatment + 3 extra days do
9:    $C_S = C_S + \text{treatment}(\text{day})$     (take dose)
10:  while time  $\leq$  end_of_day and  $0 < B <$  deadly_level_bacteria do
11:    calculate average number of bacteria created  $p_1$  (Table 5.2)
12:    calculate average number of bacteria deaths  $p_2$  (Table 5.2)
13:    update bacteria population:  $B = B + P(\tau p_1) - P(\tau p_2)$ 
14:    update time: time = time +  $\tau$ 
15:    update antibiotic concentrations,  $C_S$  and  $C_B$  (Table 5.2)
16:    if  $B \geq$  deadly_level_bacteria then
17:      Treatment is unsuccessful
18:    end if
19:  end while
20: end for
21: if  $B \leq 0$  then
22:   Treatment is successful
23: else
24:   Treatment is unsuccessful
25: end if
```

Table 5.2: List of all events for simulating the population of bacteria during treatment.

| Events | Description | Bacteria Population (B) Change |
|--------|--|--|
| p_1 | Birth of new bacteria | rB |
| p_2 | Death of bacteria | $mB + \frac{b_1 C_B^k}{C_B^k + b_2^k} B$ |
| C_B | Concentration of antibiotics in the blood | $C_B + \tau(p\alpha C_S - gC_B)$ |
| C_S | Concentration of antibiotics in the stomach | $C_S - \tau\alpha C_S$ |

Bacteria modelling (two strains). When modelling two strains of bacteria, S denotes the bacterial strain with a lower MIC and is more susceptible to the antibiotic, while R denotes the bacterial strain with a higher MIC, being more resistant and requiring a higher dose of antibiotics. In this case, the mathematical equations in Table 5.2 are replaced by those in Table 5.3. There are now five events that take place in the simulation: the birth of new bacteria of each type (p_1 and p_2), death of each type of bacteria (p_3 and p_4) and finally p_5 representing the horizontal gene transfer process (resistance gene from the R bacteria strain is passed on to the S bacteria strain).

Table 5.3: List of all events for simulating the population of bacteria during treatment where two strains of bacteria are present. Here, S bacteria is more susceptible with a lower MIC and R is more resistant with a higher MIC.

| Events | Description | Bacteria Population (S, R) Change |
|--------|--|---|
| p_1 | Birth of new S bacteria strain | $r_S S$ |
| p_2 | Birth of new R bacteria strain | $r_R R$ |
| p_3 | Death of S bacteria | $m_S S + \frac{b_{S1} C_b^{k_S}}{C_b^{k_S} + b_{S2}^{k_S}} S$ |
| p_4 | Death of R bacteria | $m_R R + \frac{b_{R1} C_b^{k_R}}{C_b^{k_R} + b_{R2}^{k_R}} R$ |
| p_5 | S bacteria becomes R due to the horizontal gene transfer process | θSR |

Implementation and technical set up. To speed up the simulation process, an approximation of the Gillespie algorithm is used, known as Tau-leaping [99]. Following preliminary model runs, we settled on a fixed time step of $\tau = 15$ minutes and updated the number of bacteria using the equation in Algorithm 2 (line 14), where $P(\tau p_i)$ is a Poisson distributed random variable with mean τp_i . We chose 15 minutes as the time step as 20 and 30 minutes will result in too much additional stochasticity and anything less than 10 minutes will not provide enough computational power saving.

Our implementation uses Python with the Numba JIT compiler [111] to parallelise the simulation runs on up to 32 computer cores, significantly speeding up the process.

5.3 COMPUTATIONAL OPTIMISATION

Problem formulation. The task at hand is formulated as an optimisation problem. Specifically, as a single objective minimisation problem with a single linear constraint. The objective to minimise is the failure rate f_r measured as the ratio of simulation runs, using the stochastic model described in section 5.2.2, where the bacteria population is not eradicated, that is where the bacteria population size is above zero after three days of the last regimen dose. We used a number of 10 000 simulation runs, and the failure rate is the ratio of the number of runs where the bacterial population is eradicated out of the 10 000 runs.

The constraint accounts for the maximum total antibiotic allowed for treatment. The total antibiotic used by a regimen vector $\bar{x} = (x_1, x_2, \dots, x_n)$, is simply the sum $\sum_{i=1}^n x_i$ of its daily doses. The maximum total antibiotic allowed is modelled as a hard constraint, which means that a regimen vector that exceeds the allowed maximum A_{total} is deemed invalid and thus discarded by the optimisation process. More formally, the optimisation problem can be stated as follows:

Find vector $\bar{x} = (x_1, x_2, \dots, x_n)$, $x_i \in \mathbb{R}^+$

to minimise function f_r

subject to the constraint $\sum_{i=1}^n x_i \leq A_{\text{total}}$

In our experiments, the duration of treatment was set to 10 days, $n = 10$ and no upper bound is imposed on the daily doses. Table 5.4 reports the minimum inhibitory concentration (MIC) values used in our experiments. For each MIC value, two values for the total antibiotic constraint were considered, which we name here in relative terms *lower* and *higher*. The *higher* values were selected in such a way that the best fixed-dose treatments in simulation reach a failure rate

Table 5.4: Values for the minimum inhibitory concentration (MIC), and maximum total antibiotic used.

| MIC [$\mu\text{g}/\text{mL}$] | | 8 | 16 | 24 | 32 |
|---------------------------------|--------|-----|-----|-----|-----|
| Total Antibiotic [mg] | higher | 150 | 300 | 450 | 600 |
| | lower | 125 | 250 | 400 | 550 |

below 1%. The *lower* values allow us to explore the impact on the failure rate of reducing the total amount of antibiotic for both fixed-dose and optimised treatments.

Differential Evolution (DE) is the population-based stochastic search method used in this chapter. Our experiments use dithering for the mutation factor F , as it can help the speed of convergence. Dithering uniformly at random (from a given tuple (min, max)) changes the mutation constant on a generation-by-generation basis. Over the years, several DE variants have been proposed [75]. Here we use the classic ‘rand/1/bin’ strategy, where ‘rand’ indicates that base vectors are randomly chosen, ‘1’ means that only one vector difference is used to form the mutated population, and the term ‘bin’ (from binomial distribution) indicates that uniform crossover is employed when creating the trial population.

Table 5.5: Differential evolution control parameter values.

| Parameter | Description | Value |
|-----------|---------------------------|----------|
| F | Scaling factor (mutation) | (0.7, 1) |
| NP | Population size | 150 |
| Cr | Crossover rate | 0.7 |

Stopping condition. The stopping condition for the DE runs was set as a maximum number of iterations. We used a maximum of 4 000 iterations for experiments with MIC = 8 $\mu\text{g}/\text{mL}$, 8 000 iterations for MIC = 16 and MIC = 24

$\mu\text{g}/\text{mL}$, and 10 000 iterations for $\text{MIC} = 32 \mu\text{g}/\text{mL}$. We needed to scale up the iterations with the MIC value as a larger amount of antibiotic was required the higher the MIC, which resulted in an increased feasible search space. Figure 5.1 shows typical DE failure rate convergence profiles. The failure rate appears to stabilise (within a margin of error) well before the chosen iteration bounds.

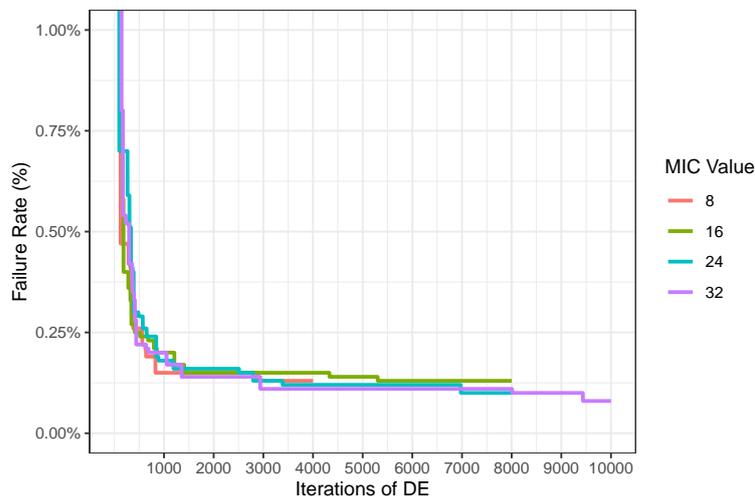


Figure 5.1: Failure rate convergence over DE iterations for experiments with different MIC values and total antibiotic constraint $\text{TA} = A$ as indicated in Table 5.4.

Constraint Handling. A common way of handling constraints within evolutionary algorithms is to apply penalty functions. In its simplest form, the function to be minimised can be computed by penalising the objective function with a weighted sum of constraint violations. A disadvantage of this approach, however, is that one or more additional penalty parameters are expected to be set by the user a priori, which requires additional effort. Therefore, in our experiments we adopted the constraint handling technique proposed in [112], where the replacement rule of the DE algorithm is modified. Specifically, when compared with the corresponding member in the population, a trial (mutant) vector will be selected if: (i) it is feasible and provides a lower or equal objective function value, (ii) it is feasible while the current vector is unfeasible, or (iii) it is infeasible but provides a lower or equal constraint violation. This

method has the advantage in our formulation of not requiring any additional parameter value other than the total antibiotic (A_{total}) constraint value.

Implementation and technical set up. The optimisation process was implemented in Python using NumPy [113] and the Differential Evolution algorithm with its associated constraint handling methods available in SciPy [76]. A total of 10 DE runs were conducted for each MIC and A_{total} constraint values.

Re-evaluation of best-found solutions. As the underlying mathematical model of bacterial infection is stochastic, the evaluation of the failure rate f_r during DE runs is susceptible to noise. This is due to both the mathematical model using a fixed number of 10 000 simulation runs, and the greedy DE selection bias, where noise could produce optimistic estimates of the failure rate. To counter these inaccuracies, all final solutions are re-evaluated by running the stochastic model 1 000 000 times. Binomial confidence intervals (with 95% confidence limit) are then calculated for each of the failure rates f_r , and these confidence intervals are used when comparing solutions in order to establish which one truly performs best.

5.4 RESULTS

Our results are organised into 3 subsections, reporting experiments with a single strain of bacteria and antibiotic administered orally (5.4.1), a single strain of bacteria and antibiotic administered intravenously (5.4.2), and two strains of bacteria and antibiotic administered orally (5.4.3). For all experiments, the antibiotic is administered at fixed 24-hour intervals for the duration of treatment.

5.4.1 *Single Bacteria Strain and Oral Administration*

We start by contrasting the effectiveness of fixed-dose treatments against those optimised by DE. In order to identify the fixed-dose benchmarks, we compute

the failure rates, using the mathematical model, of fixed-dose treatments with a duration between 5 and 10 days, for all the MIC and total antibiotic constraint values. The daily doses of fixed-dose treatments are simply the total antibiotic values divided by the number of treatment days. The resulting failure rates are plotted in Figure 5.2. We have re-scaled the plot to only include treatments with $f_r \leq 8\%$ so some data points are missing from the last two plots.

We observe that treatments of length 6, 7 and 8 days provide the lowest failure rate f_r , while 9 and 10 days regimens produce the highest failure rates f_r , especially for bacteria with MIC 8 and 16 $\mu\text{g}/\text{mL}$.

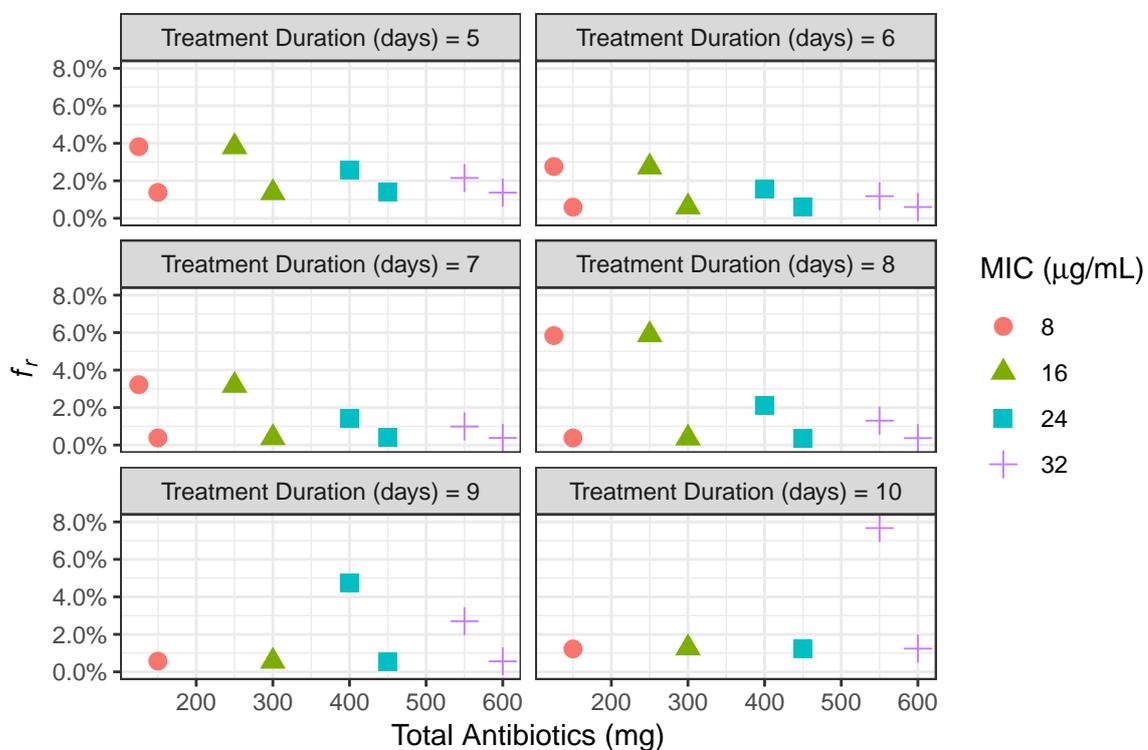


Figure 5.2: Treatments with fixed daily doses by MIC value, maximum total antibiotic and length. The plot has been re-scaled so only solutions with $f_r \leq 8\%$ are presented.

To compare the best fixed-dose treatments against the DE-optimised treatments, we completed 10 runs of DE for every combination of MIC and total antibiotic. The resulting scatter plot of failure rates can be seen in Figure 5.3.

For comparison, the figure also shows the best fixed-dose failure rates (taken from Figure 5.2) as a black-coloured marker. Note that the failure rates f_r of all optimised regimens are based on re-evaluating the mathematical model 1 000 000 times. In addition, the dose was set to 0 for each day when a DE-optimised treatment recommended a dose of less than 5 mg as doses under 5 mg have little effect on the success of the treatment

We observe that the fixed-dose treatments are less effective, that is, have a higher failure rate f_r than any of the optimised treatments. For the treatments where the MIC is 8 and 16 we can see that even a small increase in the total antibiotic results in an improvement from around $f_r = 2.25\%$ to $f_r = 0.3\%$, whereas with MIC at 24 and 32, the improvement is slightly less. This is expected, as a higher MIC requires more antibiotics to kill the bacteria, so adding 50 mg of antibiotic amounts to a lesser relative increase of the total amount of antibiotic.

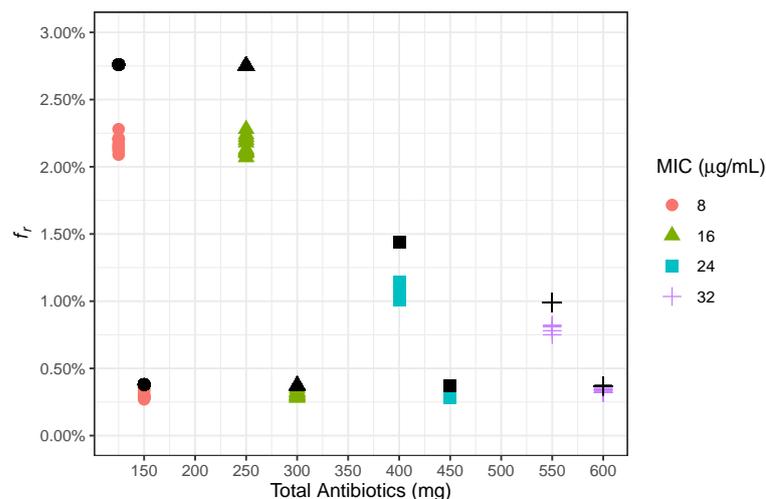


Figure 5.3: Optimised treatments with different daily doses by MIC value and maximum total antibiotics. The black markers represent the fixed treatment with the lowest failure rate f_r for that configuration.

5.4.1.1 Dosage Profile of Optimised Treatments

Figure 5.4 plots the dosage profiles of the three best-optimised treatments per each scenario in relation to MIC and A_{total} . The best treatments for

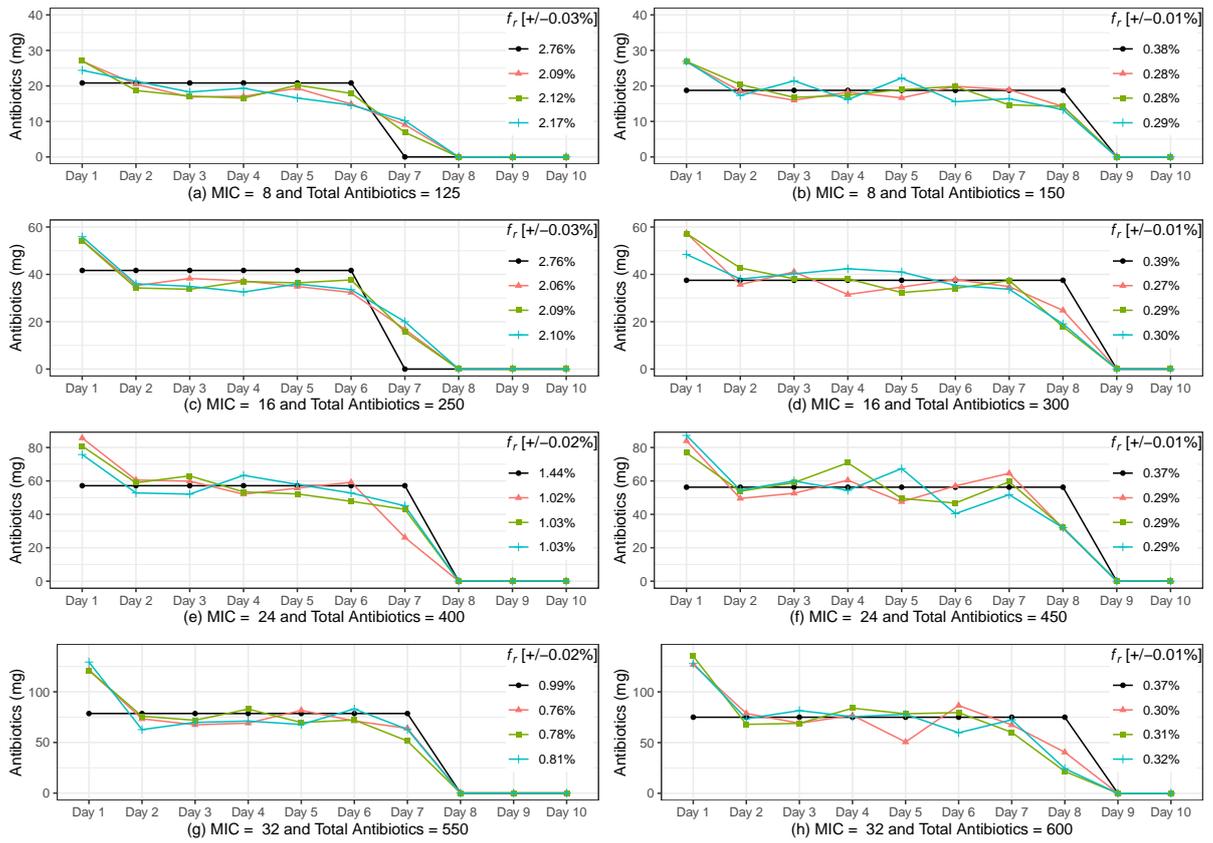


Figure 5.4: Comparison of the dosage profile of the three best-optimised treatments against the best fixed-dose treatment (coloured in black and shaped with a circle). Failure rates f_r are listed on the right-hand side, with confidence intervals in square brackets.

each scenario are those with the lowest failure rates after re-evaluation. For comparison, the (constant) dosage profile of the best fixed-dose treatment for each scenario (as determined in Figure 5.2) is also shown in black. Failure rates and confidence intervals are listed on the right-hand side of each plot.

Across all experiments, we observe that the failure rates of the best-optimised treatments are approximately between 20% and 35% lower than the failure rate of the corresponding best fixed-dose treatment for the scenarios. The failure rate reduction appears to diminish with higher MIC and higher total antibiotic values. For instance, the lowest failure rate reduction of 21% is found for the experiment with MIC = 32 $\mu\text{g}/\text{mL}$ and the higher total antibiotic constraint of 600 mg (Figure 5.4(h)).

None of the experiments produce a clear *best* optimised treatment, as the confidence intervals of f_r of several optimised treatments overlap each other. In addition to being virtually indistinguishable by failure rate, the dosage profiles of the three best-optimised treatments appear to follow a similar pattern.

- All optimised treatments for a given MIC value and antibiotic constraint agree on the treatment duration. In most cases, this is the same as the length of the corresponding fixed-dose treatment (except for the experiments with MIC = 8 or 16 $\mu\text{g}/\text{mL}$ and the lower antibiotic constraint, where optimised treatments take one day longer).
- All optimised treatments start with a high dose on the first day, followed by $n - 2$ doses that are roughly similar to the corresponding fixed-dose treatment, and tapering off with a lower dose on the final day, where n is the duration of the treatment. The first and last doses vary across experiments. In most cases, the first dose is approximately 150% of the second dose, and the final dose is about 50% of the second dose.

5.4.1.2 *Distribution of Time to Clear Infections*

In addition to treatment failure rates, we investigate the *time to clear the infection* of successful treatments by counting the number of days it takes for the bacterial population to drop to zero. Figure 5.5 plots the distributions of the time to clear for each experiment, both for the best fixed-dose treatment (the left-most column of each plot) and for the three best-optimised treatments. Distributions are presented as colour-coded columns, where the height of each colour block corresponds to the number of hosts (out of 1 000 000) that cleared the infection on the given day of the treatment. Shown on top of each column is the *expected time to clear* the infection in days. Note that even though treatments are at most 8 days long, hosts may clear the infection after the last day of treatment. Failed treatments, that is, cases where the infection is not cleared within 13 days, are excluded from the distributions.

Across most experiments, we observe that optimised treatments clear infections faster. In particular, most optimised treatments clear significantly more infections on or before day 4 than the corresponding fixed-dose treatments, resulting in a reduction of the expected time to clear by between 0.4 and 0.8 days. The exception is the experiments with MIC levels 8 and 16 $\mu\text{g}/\text{mL}$ and lower total antibiotic constraint, where the distributions of time to clear of the optimised treatments are very similar to the distributions of the corresponding fixed dose treatments. However, the optimised treatments in these two experiments are one day longer than the fixed-dose treatments, which explains why we do not observe improvements in the time to clear infections in these cases. (Note that we were only optimising the failure rate of treatments, not the time to clear infections.)

5.4.1.3 *Discussion*

Our results suggest an optimal treatment duration of 7 days if optimised against the lower total antibiotics constraint and 8 days if optimised against

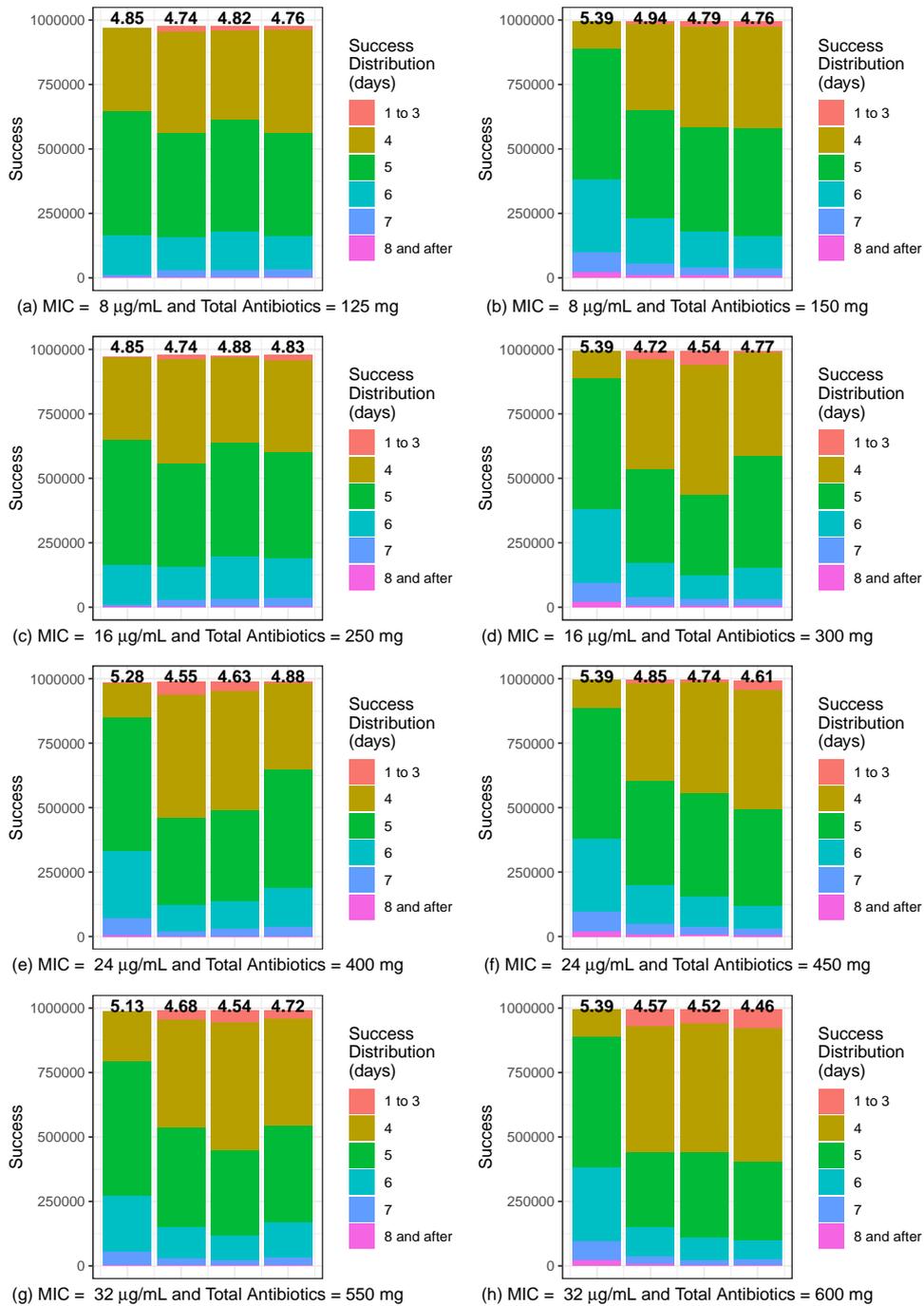


Figure 5.5: Distributions of the time to clear the infection, comparing the best fixed-dose treatment (left-most column of each plot) to the three best-optimised treatments. The expected time to clear is shown on top of each column. Each treatment was evaluated 1 000 000 times.

the higher constraint. This is broadly in line with clinical practice, where most of the treatments prescribed are 5 or 7 days long.

We find that some optimised treatments are slightly longer than the fixed-dose ones, but perform better. We see a bigger improvement in failure rate when the MIC levels of the bacteria are at susceptible and intermediate resistance levels (8 and 16 $\mu\text{g}/\text{mL}$) to the antibiotics than when they are more resistant (24 and 32 $\mu\text{g}/\text{mL}$). As these are the majority of bacterial infections in hospitals, the optimised treatments would reduce the number of cases where bacteria survive after the end of the treatment, thereby reducing the risk of resistant strains emerging.

We also observe that optimised treatments clear infections faster than the corresponding fixed-dose treatments. This effect appears stronger when MIC levels are at the more resistant end (24 and 32 $\mu\text{g}/\text{mL}$). Thus, optimised treatments confer a second advantage, particularly for infections with resistant bacteria, by helping more patients recover quickly, thereby potentially reducing the burden on hospitals.

We attribute both the improvements in the failure rate and in the time to clear infections to the higher first-day dose of optimised treatments.

5.4.2 *Single Bacteria Strain and Intravenous Administration*

When antibiotics are injected intravenously, they go directly into the bloodstream rather than through the stomach as in the previous Section 5.4.1. In terms of the mathematical model (described in Section 5.2.2) C_S , the concentration of antibiotics in the stomach, is set to zero, and p , the proportion of antibiotics that reaches the blood, is set to one (instead of the previous value of 0.95). In order to keep results comparable, the experiments reported here explore the same combinations of MIC value and total antibiotics constraint as in Section 5.4.1.

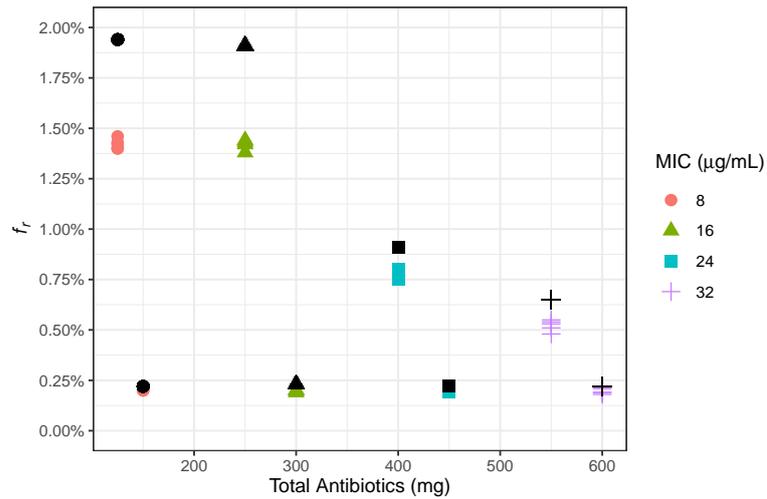


Figure 5.6: Optimised treatments administered intravenously by MIC value and maximum total antibiotics. The black markers represent the fixed-dose treatment with the lowest failure rate f_r for that configuration.

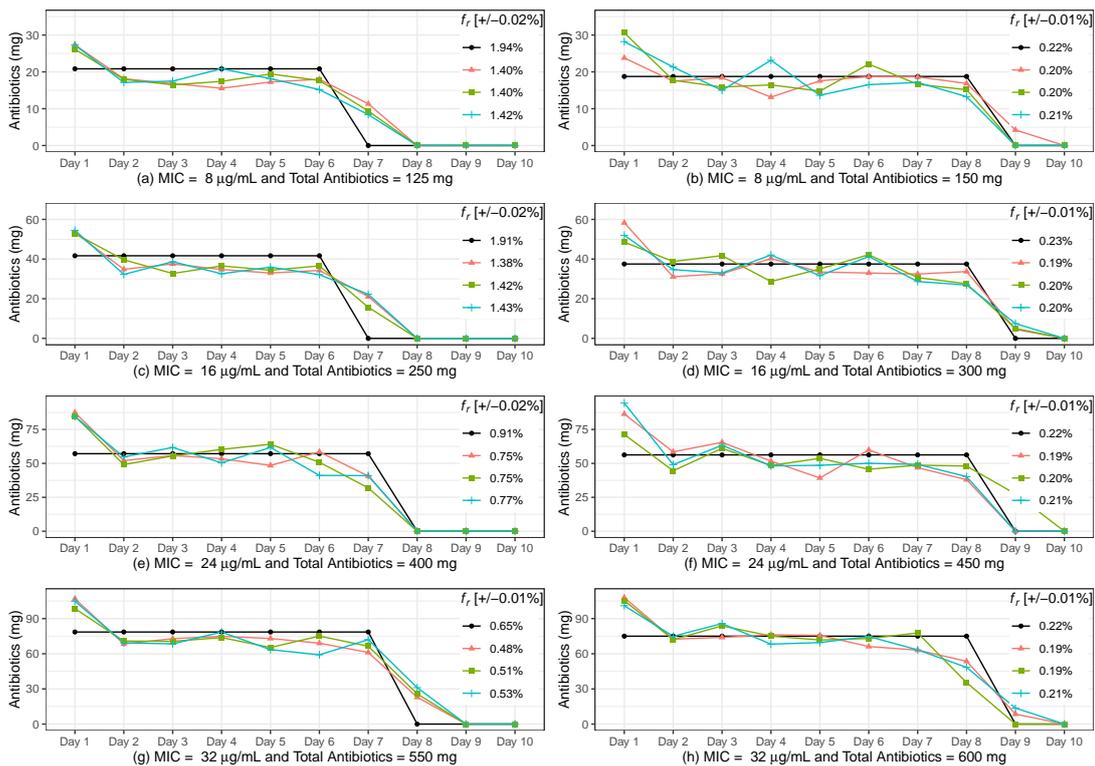


Figure 5.7: Comparison of the dosage profile of the three best optimised intravenous treatments against the best fixed-dose treatment (in black). Failure rates f_r are listed on the right-hand side, with confidence intervals in square brackets.

The scatter plot in Figure 5.6 compares the failure rates of the best fixed-dose treatments against ten DE-optimised treatments. We observe a similar picture as for orally administered treatments, that is, DE-optimised treatments tend to have lower failure rates, and the difference between lower and higher total antibiotics constraints diminishes the higher the MIC value. We also observe that failure rates are lower than for the respective orally administered treatments in Figure 5.3, which confirms that administering the drug intravenously increases effectiveness.

Figure 5.7 plots the dosage profiles of the three best-optimised treatments. For comparison, the profile of the best fixed-dose treatment is also shown in black colour. Failure rates and confidence intervals are listed on the right-hand side of each plot.

Across experiments with the lower total antibiotics constraint, we observe that the failure rates of the best-optimised treatments are approximately between 15% and 30% lower than the failure rate of the corresponding best fixed-dose treatment. However, we see almost no improvement in failure rates for experiments with the higher total antibiotics constraint. In fact, the confidence intervals of many of the DE-optimised treatments overlap the confidence interval of the best fixed-dose treatment.

The general shape of the dosage profiles is similar to the shape of the orally administered treatments: a high first dose, followed by roughly constant doses and tapering off on the final day. However, more than half of the optimised treatments are a day longer than the best fixed-dose treatment. In contrast, most orally administered treatments matched the fixed-dose treatment in duration.

5.4.2.1 *Discussion*

As observed in Figure 5.7, DE barely manages to improve on the failure rate of the fixed-dose treatment in experiments with the higher total antibiotics constraint. There are two hypotheses for this — due to the stochastic nature

of the mathematical model, the fitness function might be too noisy, making it difficult for the DE to find the optimal solution; or the fixed-dose failure rate is already near the optimal value.

To check whether the failure of DE in finding better solutions is down to the noisy fitness function, we performed additional experiments, increasing the number of runs of the mathematical model from 10 000 to 100 000, thereby reducing the noise on the fitness function by an order of magnitude (yet increasing the computational cost by an order of magnitude). The best failure rates found in these experiments ranged from $0.17\% \pm 0.01\%$ to $0.19\% \pm 0.01\%$. This is an improvement on the best fixed-dose failure rates of $0.22\% \pm 0.01\%$, although the relative improvement of about 15% to 20% is smaller than observed in other experiments.

The additional experiments suggest that noise on the fitness function may prevent DE from converging to the optimum. However, the modest improvements despite reducing the noise by an order of magnitude also suggest that there is not a single optimal treatment but a wide basin of treatments with very similar near-optimal failure rates.

5.4.3 *Two Bacteria Strains with Oral Administration*

In cases when people are immunocompromised, it is common that they could carry multiple types of bacteria or several strains of the same type of bacteria. In this set of results, we are modelling the case when 95% of the bacterial population has a resistance of MIC = 8 $\mu\text{g}/\text{mL}$ and 5% of the bacterial population make up a strain with a more resistant MIC (16, 24 or 32 $\mu\text{g}/\text{mL}$). Antibiotics are administered orally.

We first examined treatments using the same total amount of antibiotics as in the experiments with only one strain of bacteria in Section 5.4.1. We observed failure rates around 10% when the more resistant strain has MIC = 16 $\mu\text{g}/\text{mL}$. However, the failure rates rise to an average of 97% for MIC = 24 $\mu\text{g}/\text{mL}$,

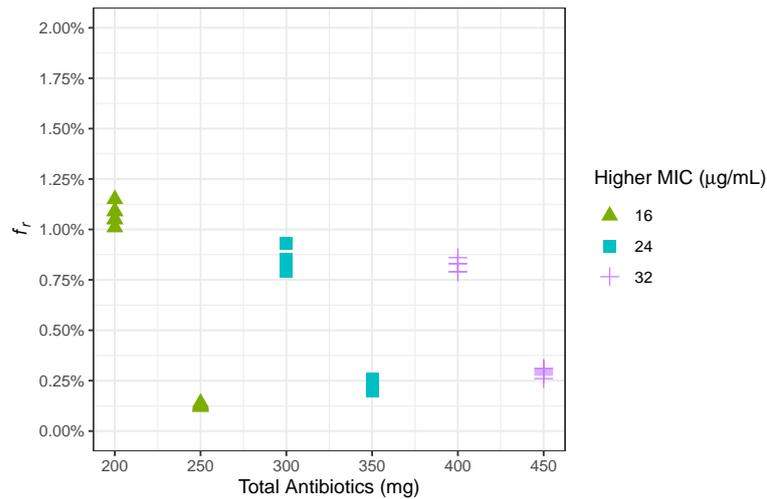


Figure 5.8: Treatments with $f_r \leq 1\%$ when there are two strains of bacteria — one with MIC = 8 $\mu\text{g/mL}$ that makes 95% of the initial bacteria population and one that corresponds to the MIC shown in the plot that makes 5% of the initial bacteria population.

and to 100% for MIC = 32 $\mu\text{g/mL}$. This shows that if a patient is treated for less resistant bacteria when a more resistant strain is present even in small amounts, the treatment will likely fail.

It seems plausible that treating a multi-strain infection will require more antibiotics than would be required for the less susceptible strain on its own and less than would be required to treat the most resistant strain on its own. This is confirmed by the findings in Figure 5.8, which plots failure rates for combinations of the total antibiotics constraint and the MIC value of the more resistant strain. We are showing the result of five runs of the DE with 10 000 iterations for each scenario. For each MIC value, the figure shows two treatments that differ by 50 mg in the total amount of antibiotics used. The treatments with the lower total antibiotics constraint are 7 days long, and the treatments with the higher constraint are 8 days. The shape of the treatments is not shown but follows the same pattern we observed before — high first dose, roughly constant middle doses, and tapering off with a smaller final dose.

5.4.3.1 *Discussion*

We saw that even if a small percentage of the bacteria population develops a mutation-increasing resistance, the treatment could become unsuccessful. Thus, more antibiotics are needed where a multi-strain infection is suspected. In our model, we found that the amount of antibiotics needed to guarantee a failure rate well below 1% is quite close to the number of antibiotics required to treat the more resistant strain on its own — even though that more resistant strain makes up only 5% of the initial bacteria population.

5.4.4 *Extrapolating the Optimised Regimens*

As observed in figures 5.4 and 5.7, the shape of optimised treatments always follows the same pattern: a high first dose, followed by roughly constant doses, and tapering off on the final day. The ratio of first to second doses varies across experiments, but is often close to 1.5. This leads us to extrapolate the following simple way of formulating an optimised treatment without running the DE algorithm.

Suppose the standard fixed-dose regime is a daily dose of $2 \times x$ mg over n days. Then the extrapolated optimised dose regime consists of a first dose of $3 \times x$ mg, followed by $n - 2$ doses of $2 \times x$ mg, followed by a final dose of x mg. Table 5.6 contrasts the failure rates of the best DE-optimised treatments (orally administered) against the failure rates of treatments of the same length but using the extrapolated dosage regime. This shows a slight further improvement (around 5 to 10%) of failure rates across the board. (We see a similar improvement for intravenously administered drugs.)

The extrapolated dose regime could easily be implemented in a real-life scenario where x mg is the dosage of a single pill, and the patient takes 3 pills on the first day of treatment, followed by $n - 2$ days of 2 pill doses, and 1 pill on the final day. Our modelling predicts that such a dose regime would

Table 5.6: Comparison of failure rates f_r of the best fixed-dose treatments, the best DE-optimised treatments, and the corresponding extrapolated treatments.

| | | | | |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|
| MIC [$\mu\text{g}/\text{mL}$] | 8 | 16 | 24 | 32 |
| total antibiotics [mg] | 125 | 250 | 400 | 550 |
| treatment length [days] | 7 | 7 | 7 | 7 |
| f_r fixed dose [%] | 2.76 ± 0.03 | 2.74 ± 0.03 | 1.41 ± 0.02 | 1.00 ± 0.02 |
| f_r best DE [%] | 2.09 ± 0.03 | 2.06 ± 0.03 | 1.02 ± 0.02 | 0.76 ± 0.02 |
| f_r extrapolated [%] | 2.00 ± 0.03 | 1.98 ± 0.03 | 0.98 ± 0.02 | 0.71 ± 0.02 |
| MIC [$\mu\text{g}/\text{mL}$] | 8 | 16 | 24 | 32 |
| total antibiotics [mg] | 150 | 300 | 450 | 600 |
| treatment length [days] | 8 | 8 | 8 | 8 |
| f_r fixed dose [%] | 0.36 ± 0.01 | 0.38 ± 0.01 | 0.38 ± 0.01 | 0.38 ± 0.01 |
| f_r best DE [%] | 0.28 ± 0.01 | 0.27 ± 0.01 | 0.29 ± 0.01 | 0.30 ± 0.01 |
| f_r extrapolated [%] | 0.25 ± 0.01 | 0.25 ± 0.01 | 0.25 ± 0.01 | 0.25 ± 0.01 |

significantly improve treatment failure rates compared to the standard fixed-dose treatment. However, these predictions need to be clinically validated.

5.5 CONCLUSION

Antimicrobial resistance is a growing global threat to healthcare and food production. To deal with this complex challenge, a range of approaches are required, critically including novel strategies to optimise the use of existing antibiotics. This chapter uses mathematical modelling and state-of-the-art evolutionary algorithms for optimising dosing regimes tailored to bacterial infections with different levels of resistance. We also explored two forms of administering antibiotics (orally and intravenously), as well as infections with a single strain and two strains of bacteria. Our formulation encodes dosing regimens as vectors of real numbers and uses a linear constraint on the total antibiotic used.

The main goal was to design optimised regimens with lower failure rates than the standard fixed-daily dose regimens for the same amount of antibiotics. The resulting optimised regimens have varying daily doses and achieve an improved lower failure rate of between 20% and 35% when compared to fixed-dose regimens with the same amount of drug, demonstrating a relative improvement. All optimised regimens, for n days in duration, start with a high dose on the first day, followed by $n - 2$ doses that are roughly similar to the corresponding fixed-dose regimen, and tapering off with a lower dose on the final day. The first and last doses vary across experiments. In most cases, the first dose is approximately 150% of the second dose, and the final dose is about 50% of the second dose.

A general pattern can thus be extrapolated of how treatments could be optimised, where the first dose is $3 \times x$ mg, followed by $2 \times x$ mg and the last dose of x mg, where $2 \times x$ mg is the standard daily fixed dose currently prescribed.

It is important to note that different antibiotics have different levels of toxicity, however, taking an extra dose of antibiotics is unlikely to cause serious harm.

NOISY HANDLING MECHANISM AND RUNTIME OPTIMISATION

6.1 INTRODUCTION

While many studies have explored different techniques for handling problems with noisy objective functions [114]–[116] as well as specific implementations for established algorithms [117]–[119] not many direct comparisons between the approaches have been made. We have chosen CMA-ES over the rest of the solutions as it showed promising results in several applications with noisy objective functions [120], [121]. [122] describes CMA-ES less vulnerable to noise because it applies a population-based approach, averaging in the recombination process, and a rank-based, non-elitist selection. In Chapter 3 an overview of the algorithm has been presented and later on in this chapter, an overview of the noise-handling approach is described.

We employ the covariance matrix adaptation evolution strategy (CMA-ES) [78], [123]–[127]. This choice is made because of several reasons:

- CMA-ES is a non-elitist continuous domain evolutionary algorithm. Non-elitism avoids systematic fitness overvaluation on noisy objective functions [117] because even solutions with superior fitness values survive only one generation. This is applicable to the fitness function we use, as the mathematical model is stochastic, and is the reason we evaluate each solution multiple times.
- The selection of CMA-ES is only based on the ranking of solutions. This provides additional robustness in a noisy environment. Ranking-

based selection (sorts the population first according to fitness value and ranks them) is particularly necessary to the strictly order-preserving transformations of the fitness function.

- The CMA-ES provides an effective adaptation of the search distribution to the landscape of the objective function.
- The CMA-ES can be reliably used with small population sizes, allowing for a fast adaptation in an online application or live systems. In future work, the model is to be used by medical professionals to optimise and personalise antibiotic treatment, so it is important for the results to be delivered quickly.
- There is a UH-CMA-ES mechanism that is used for noise handling and uncertainty handling.

A comparison between using CMA-ES and UH-CMA-ES is done, where the number of mathematical runs is decreased in a few steps.

6.2 NOISE HANDLING IMPLEMENTATION — PYCMA

Hansen, Nikolaus, et al. in [126] propose a re-evaluation technique that provides a quantification of the uncertainty for any ranking-based search algorithm. As a ranking-based search algorithm changes the ordering of the solutions, degrees of uncertainty are brought into the objective function. The uncertainty quantification uses rank changes that happen as a result of the reevaluation of the solutions. To counteract this, before the reevaluation is carried out, a small perturbation is applied to count for the noisiness of the fitness function. The base uncertainty handling mechanism reevaluates each solution at most once, and that solution is chosen at random. There is an option to change the solution to be reevaluated to the best solution so far, but the paper does not indicate major differences in the results between the two approaches. After the reevaluation process, the number of ranks changes.

Finally, the measured rank changes are normalised, providing us with the uncertainty measurement.

Another feature of the uncertainty handling mechanism is the increase in the population variance. This is beneficial as it makes the population more diverse, the population escapes search-space regions with too low a signal-to-noise ratio, and premature convergence is prevented.

6.3 EXPERIMENTAL SETUP

We have chosen the `pycma` library to implement the CMA-ES algorithm. [83]. For the experiments we have set the maximum budget for fitness function evaluations to 40 000 and the starting population is the same for both sets of experiments which is set to $4 + 3 * \log(N)$ according to the library's documentation, where $N = 10$ which is the starting length of the treatment. The initial global step size $\sigma = 0.5$ and the seed is set to 234 for all experiments. After the run of the algorithm is finished, the result is reevaluated with 1 000 000 runs of the mathematical model, and we refer to that value as failure rate f_r . The difference in the experiments is whether the uncertainty handling mechanism was enabled or not and how many times the mathematical model was running during the experiment. The number of mathematical model runs is referred to as $r_{\text{mathmodel}}$ in the following sections. It is important to note that the population size increases when the noise handling mechanism is used, and it is optimised during each run.

We have chosen $\text{MIC} = 8$ with $A_{\text{total}} = 150$ scenario, where a single type of bacteria is present, and the antibiotic is taken orally for all experiments. This means the best fixed-dose treatment is the eight-day treatment $\underline{x} = (18.75\text{mg}, 18.75\text{mg}, 18.75\text{mg}, 18.75\text{mg}, 18.75\text{mg}, 18.75\text{mg}, 18.75\text{mg}, 18.75\text{mg})$. The failure rate for that treatment is $f_r = 0.38\%$ with the average time to clear infection $t_{\text{cure}} = 5.39$ days (in most treatments the patient will be cured during day 5 of the treatment).

Here, we have used a constraint on the treatments where the $A_{\text{total}} \leq 150$. Preliminary experiments were carried out where a weighted sum of the failure rate and the A_{total} was used as a fitness function with various weights with the failure rate being the primary function, but the results were not satisfactory when UH-CMA-ES was used with higher than expected failure rates after reevaluation.

The values chosen for how many times the mathematical model runs during the experiment are 10 000, 7500, 5000, 2500, 1000, 500, 250 and 100. For each of those values, we have run 30 experiments with CMA-ES and 30 with UH-CMA-ES. The aim of this chapter is to find the minimum number of mathematical model runs $r_{\text{mathmodel}}$ that is required while maintaining a median failure rate that is better than the fixed-dose rate.

6.4 PRELIMINARY RESULTS

6.4.1 *Evaluations Per Treatment*

In the work [13] which corresponds to Chapter 4 in this thesis, each solution was evaluated using the mathematical model 500 times, and it was established that this number was too low as the reevaluated solutions had noise in the failure rate about 5% in comparison to the one calculated during the run of the multi-objective algorithm. This led to a further analysis of how many evaluations would provide a good estimate of the failure rate without sacrificing too much computational power which was then used in Chapter 5.

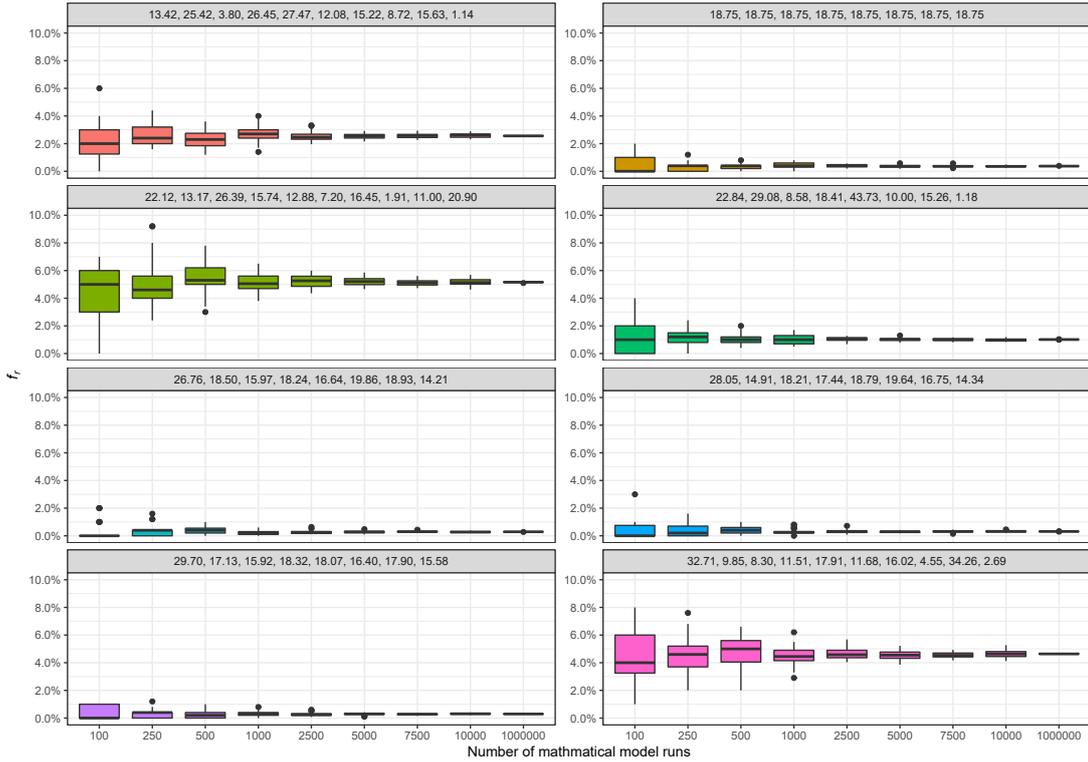


Figure 6.1: Boxplots representing the f_r calculated for various treatments with different numbers of runs of the mathematical model. Each boxplot represents 30 runs.

In Figure 6.1 we have presented six different treatments and their failure rate when running the mathematical model a different number of times - 100, 250, 500, 1000, 2500, 5000, 7500, 10 000 and 1 000 000. For each of the values, 30 runs have been made where the MIC = 8, the A_{total} is around 150 and the length of the treatments is between 8 and 10 days. We have added, 1 000 000 as a reference point to the real value of the f_r which we use for establishing the reevaluated f_r throughout the rest of the chapter. What could be observed for treatments where the $f_r \geq 1\%$ is that 100 runs results is approximately 7% noise; 250 runs — 5%; 500 runs — around 5% as well; 1000 — 3 to 2 %; 2500 — 1.5%; from 5000 to 10000 — or less than 1%. When the treatment has $f_r \leq 1\%$, the noise is much smaller for all values. However, it is important to note that when we start running the evolutionary algorithm, the treatments would be closer to the ones with higher f_r .

6.4.2 CMA-ES

Some preliminary experiments were run to explore the behaviour of the CMA-ES algorithm using the `pycma` library. In Figure 6.2 we have provided the plot presented at the end of the run by the `pycma` library. On the top graph, we can see on the y-axis coloured in blue the best value of f_T for this run, in green the `axisratio`, in red each iteration's best value of f_T and global step-size σ in orange all in \log_{10} scale. On the x-axis, we have the function evaluations. As CMA-ES uses ranking of the solutions, this graph represents the ranking of the solutions during runtime. In the middle graph, we can see the principal axes lengths for each of the 10 variables that are representing our antibiotic treatment. This represents the variance in the covariance matrix in relation to the values of the 10 variables during the runs of the algorithm. On the bottom graph, we can see the change in the actual 10 variables of the best solutions during the run of the algorithm.

We can see from all three panels that there is a sharp jump in the values around function evaluation 27 500 which is due to a change in the evolution path (accumulates historical search directions in successive generations [128]) in the global step size control. When exploring the output files, this also stands for a sudden decrease in the value of σ . As in this algorithm, σ is the global step size starting with the initial value (we set that to 0.5) given and is updated each iteration, this explains the jump in the graphs on Figure 6.2.

6.4.3 UH-CMA-ES

We can see in the caption of Figure 6.3 that the population size for this run was set by the uncertainty handling mechanism, as it differs from the initial one to increase the population diversity [126]. Comparing this run with CMA-ES one from Figure 6.2, we can see that the reevaluated failure rate is much closer to the true value than when using uncertainty handling.

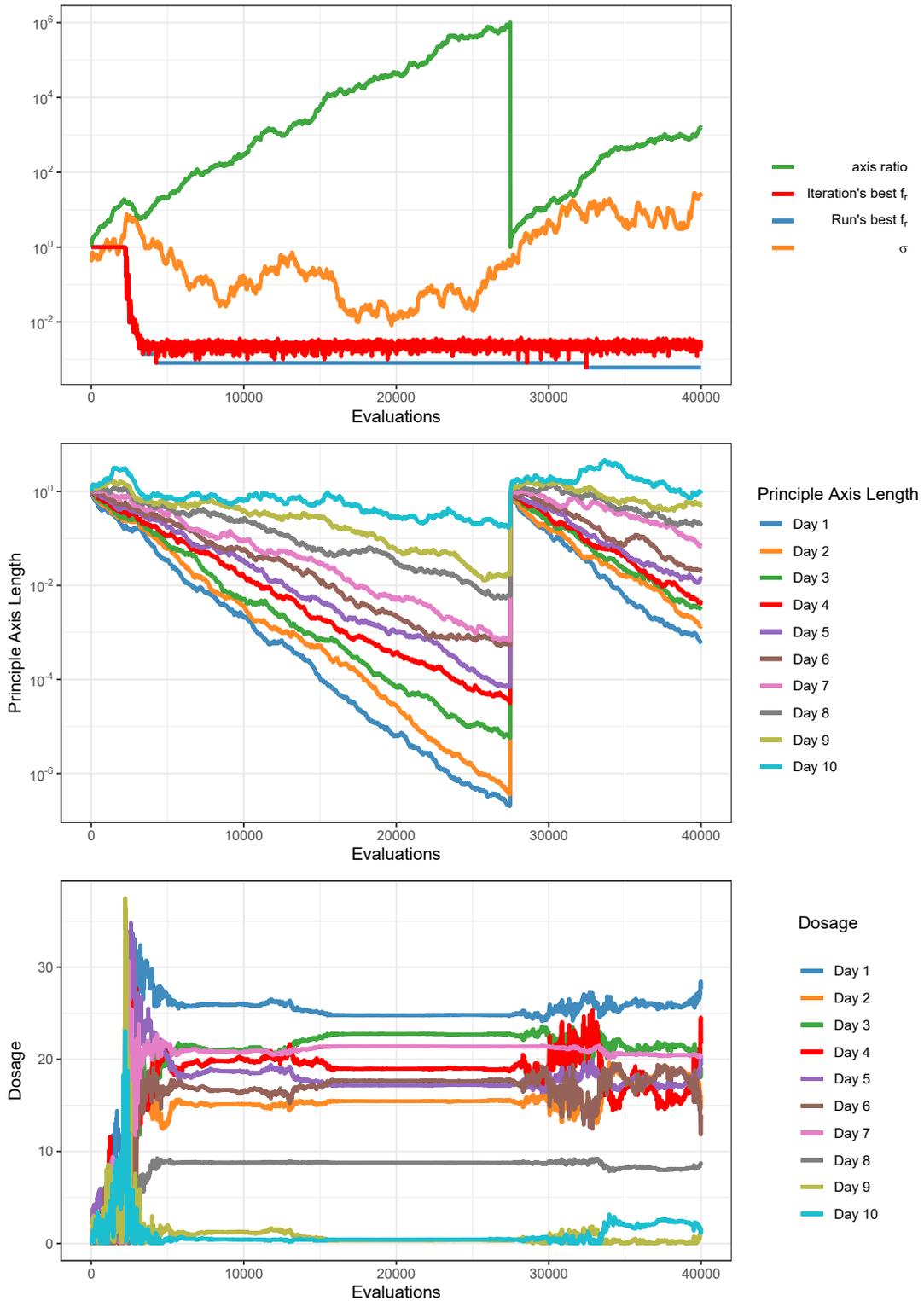


Figure 6.2: Example plot from the pycma library from CMA-ES run with the number of mathematical model runs is set to 5000. The failure rate of this run is 0.35% after reevaluation, and the failure rate is 0.06% during the CMA-ES runtime. The population size is approximately 10. In the top and the middle plots, the y-axis is presented on a logarithmic scale of 10.

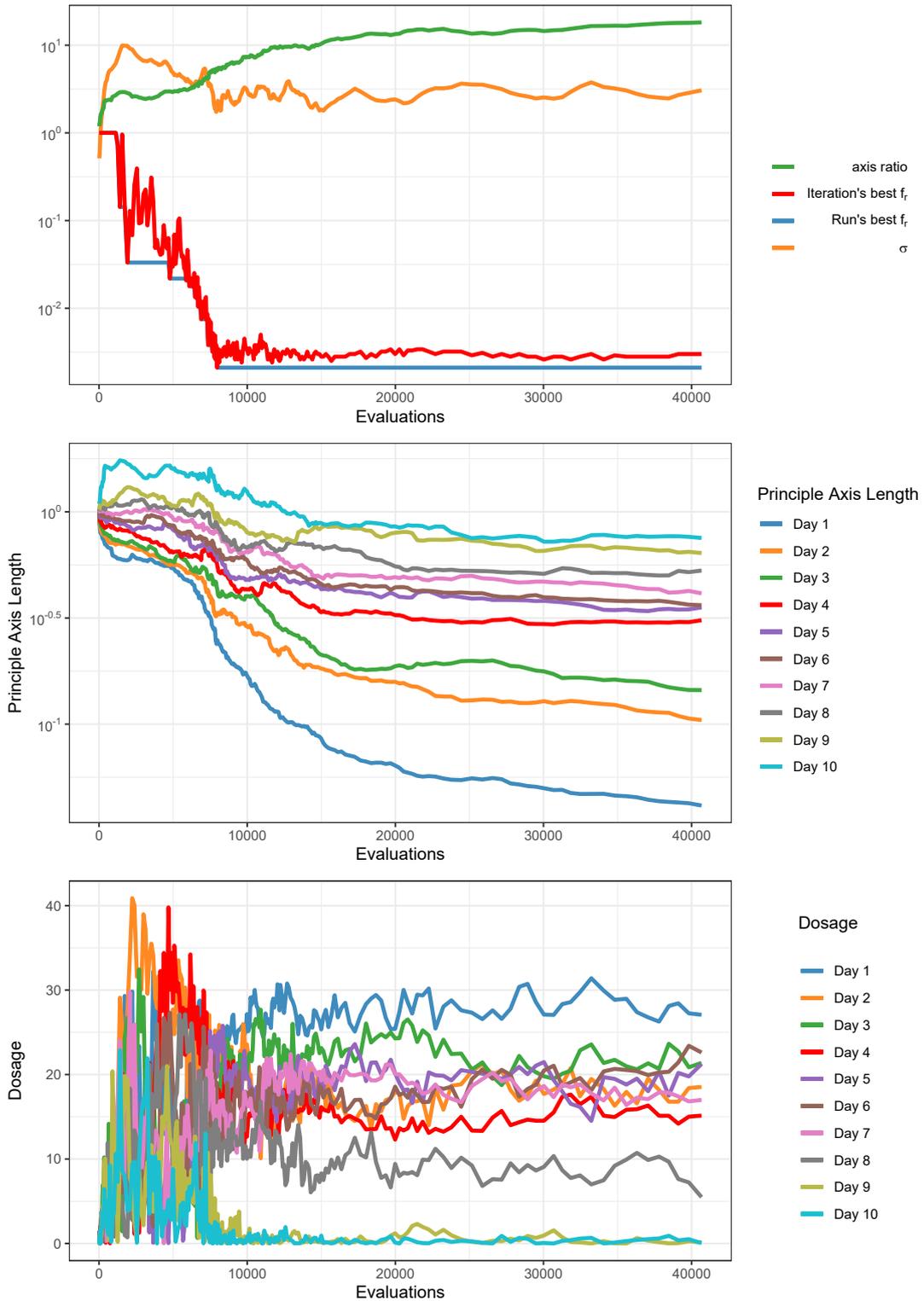


Figure 6.3: Example plot from the pycma library from UH-CMA-ES run with the number of mathematical model runs set to 5000. The failure rate of this run is 0.32% after reevaluation and the algorithm evaluated it at 0.20% during runtime. The population size for this run is set by the noise handling is approximately 858. In the top and middle plots, the y-axis is presented on a logarithmic scale of 10.

On the top graph, we can see that the run's best f_r coloured in blue reaches the minimum value at about 9000 function evaluations in this example run. Then in the middle plot, the change in value is not as steep as the one in Figure 6.2, but we also never reach a sudden change in the value of σ giving us the jump in the top plot. This is explained by the large population and the smaller number of iterations during the run of the algorithm, as in this particular example there were 268 iterations (the run in Figure 6.2 has 4000 iterations). The bottom graph represents the values of the 10 variables, and the noise in the values decreases near the same point where the f_{best} is at its minimum. This shows a different behaviour to CMA-ES where the noise in the variables was present only at the first few thousand evaluations and then after the jump in the value of σ .

Noise handling does change the behaviour of CMA-ES, however, it does not appear to significantly change the result of the optimisation as the difference in the final solution's reevaluated f_r 0.15%. Nevertheless, the final non-reevaluated f_r of the UH-CMA-ES solution is much closer to the real value of f_r .

6.5 RESULTS

When running the mathematical model on a powerful machine with processor Intel Core i5-10400 CPU @ 2.90GHz with 12 threads and 8.00 GB of RAM, a single run of the mathematical model takes around 0.0000075 seconds while using JIT and Numba to optimise the Python runtime. In Chapter 5 the experiments where the MIC = 8, the maximum iterations of the DE algorithm are 4000, the population size of each iteration is 150 and each one of the individuals in each population runs the mathematical model 10 000 times. This means that for a single experiment (out of a total of 240), the mathematical model is run 6 000 000 000 times, taking around 13 hours. While these experiments were run on a more powerful machine with multiple cores

with parallelised processes, here we aim to minimise that time. In the results of this chapter, we will look into decreasing the mathematical model runs in the individuals in the population during the time the algorithm runs. The values we have chosen are 10 000, 7500, 5000, 2500, 1000, 500, 250 and 100, so we have a base for comparison. We will refer to that number as $r_{\text{mathmodel}}$ (mathematical model runs) in the following sections. 30 runs of each implementation (CMA-ES and UH-CMA-ES) are run for each of the values — 480 experiments in total.

6.5.1 *Failure Rate*

In Figure 6.4 are presented all experiments by the number of mathematical runs used in each individual in the population indicated on the top of each boxplot during the algorithm run. The grey line represents the best fixed-dose treatment with $f_r = 0.38\%$ with $A_{\text{total}} = 150$.

6.5.1.1 *CMA-ES*

The median on all the experimental setups falls above the f_r value of the fixed-dosed treatment, except for when the $r_{\text{mathmodel}} = 1000$. However, all the experimental setups except when $r_{\text{mathmodel}} = 100$ and 250 produced a result better than the fixed-dose treatment within the 30 runs. The results for CMA-ES are consistent in their failure rate, as there are not many outliers with abnormal distances from the other values, showing that the algorithm produces consistent results. This algorithm behaviour is already seen in Figure 6.2 where we can see there is little change in the values of the variables over the run of the algorithm.

6.5.1.2 *UH-CMA-ES*

While the UH-CMA-ES performs better in terms of the median of the failure rate, the outlier with abnormal distance from the other values is higher. On

| Treatment | Failure Rate f_r |
|--|--------------------|
| 17.96, 14.92, 17.70, 4.22, 29.34, 4.55, 16.92, 13.31, 11.43, 8.12 | 8.27% |
| 39.02, 14.65, 23.18, 7.36, 7.02, 23.36, 7.68, 6.94, 11.16, 5.47 | 4.74% |
| 22.12, 13.17, 26.39, 15.74, 12.88, 7.20, 16.45, 1.91, 11.00, 20.90 | 5.74% |
| 12.32, 27.91, 11.67, 6.12, 20.42, 19.42, 2.76, 17.06, 24.86, 2.28 | 8.19% |

Table 6.1: Treatments that resulted in local optimums during UH-CMA-ES experiments.

the boxplot, we have limited the results to where the $f_r \leq 3\%$, however, there are several runs which resulted in a failure rate higher than that. Interestingly, all of those runs occurred when $r_{\text{mathmodel}}$ is 10 000, 7500, 5000 or 2500, the population size is very high (around 4000), and the iterations are low (50 to 60). Those runs resulted in the treatments shown in Table 6.1. This could be explained by the population variance mechanism. Nevertheless, the only three experimental setups where the UH-CMA-ES median failure rate is significantly above the fixed-dosed are for 100, 250 and 500 which is to be expected as the noise in the fitness function is much higher as observed in Figure 6.1.

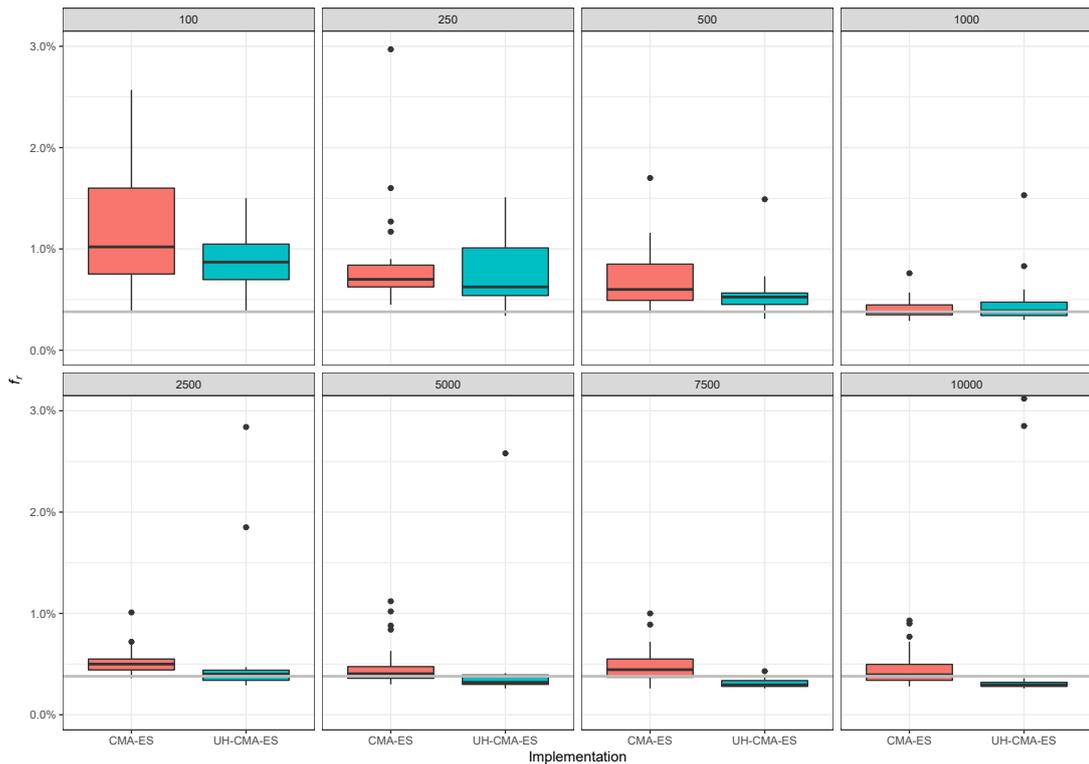


Figure 6.4: Failure rate by implementation and $r_{\text{mathmodel}}$. A gray line representing the failure rate of the optimal fixed-dose treatment is added for comparison. Please note that the plot has been restricted to show only results from 0% to 3%.

6.5.2 Time to Clear Infections

Even though the failure rate is the main feature of our solutions, we investigate the time to clear the infection of successful treatments by counting the average number of days it takes for the bacterial population to drop to zero.

6.5.2.1 Distribution

Figure 6.5 graphs the distribution of the time to clear for each experimental setup, where the fixed-dose treatment is the one in the leftmost column of each plot and the other three are the best-optimised treatments. Similar to Chapter 5, the distributions are colour-coded and the percentage represents

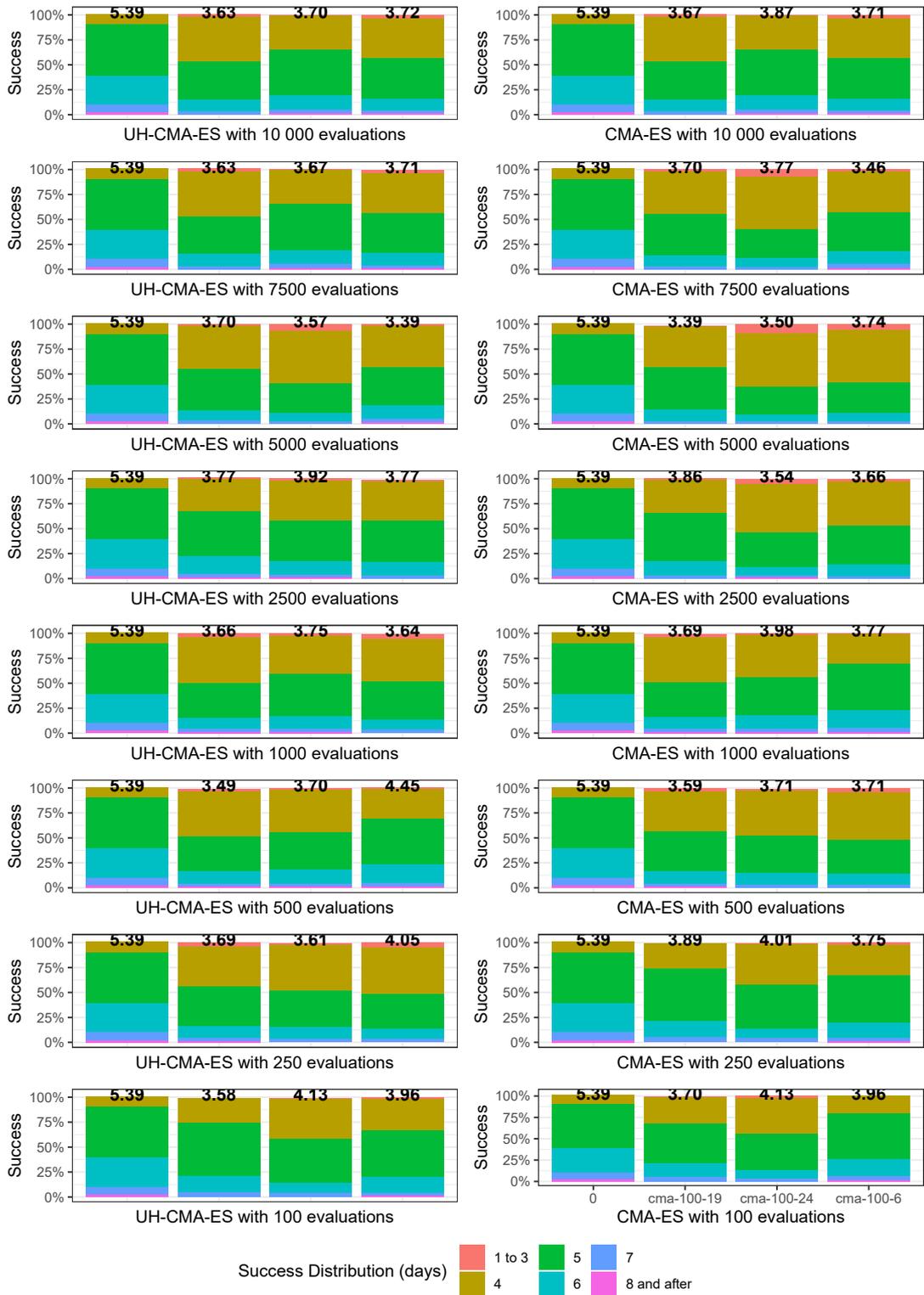


Figure 6.5: Distributions of the expected time to clear the infection, comparing the best fixed-dose treatment (left-most column of each plot) to the three best-optimised treatments. The expected time to clear is shown on top of each column. Each treatment was evaluated 1 000 000 times.

what part of the 1 000 000 evaluations in which the infection was cleared. At the top, we have noted the expected time to clear the infection in days. As the mathematical model is the same as in Chapter 5, the mathematical model runs for 13 days as the infection can be cleared in the days after the antibiotic course is over. As some treatments fail, the distributions do not sum up to 100%.

We can clearly notice that the optimised treatments clear the infection faster than the fixed-dose ones. More specifically, almost all the presented treatments clear the infection in the second half of day 3, which reduces the expected time to clear by almost 2 days. This is a better result than the one we had in Chapter 5, where the three best-optimised treatments have the estimated time to clear at 4.94, 4.79 and 4.76 — reduction with CMA-ES is just over a day. Depending on if the uncertainty handling mechanism was used or not, there is not a significant difference in the results as well as the number of mathematic model runs based on the top three runs for each experimental setup. Considering that the optimisation is not focused on the expected time to clear the infection, the CMA-ES algorithm naturally found solutions that clear the infection faster are more beneficial for achieving a lower failure rate.

6.5.2.2 *Expected Time To Clear*

Figure 6.6 presents the expected time to cure across all treatments, not just the top three for each experimental setup. The data is split into panels depending on the $r_{\text{mathmodel}}$ value, which is noted in the grey bar at the top of each panel. While the best treatments perform similarly, overall UH-CMA-ES has lower values in terms of median value but produces more outliers outside the fixed-dose treatment value. If our goal is to produce a treatment that clears the infection the fastest, we can even use the $r_{\text{mathmodel}} = 100$ with UH-CMA-ES mechanism setup, and we have a high probability to find a treatment with a lower value than the fixed-dose treatment expected time to clear. However, it is important to note that our primary objective is producing a low failure rate.

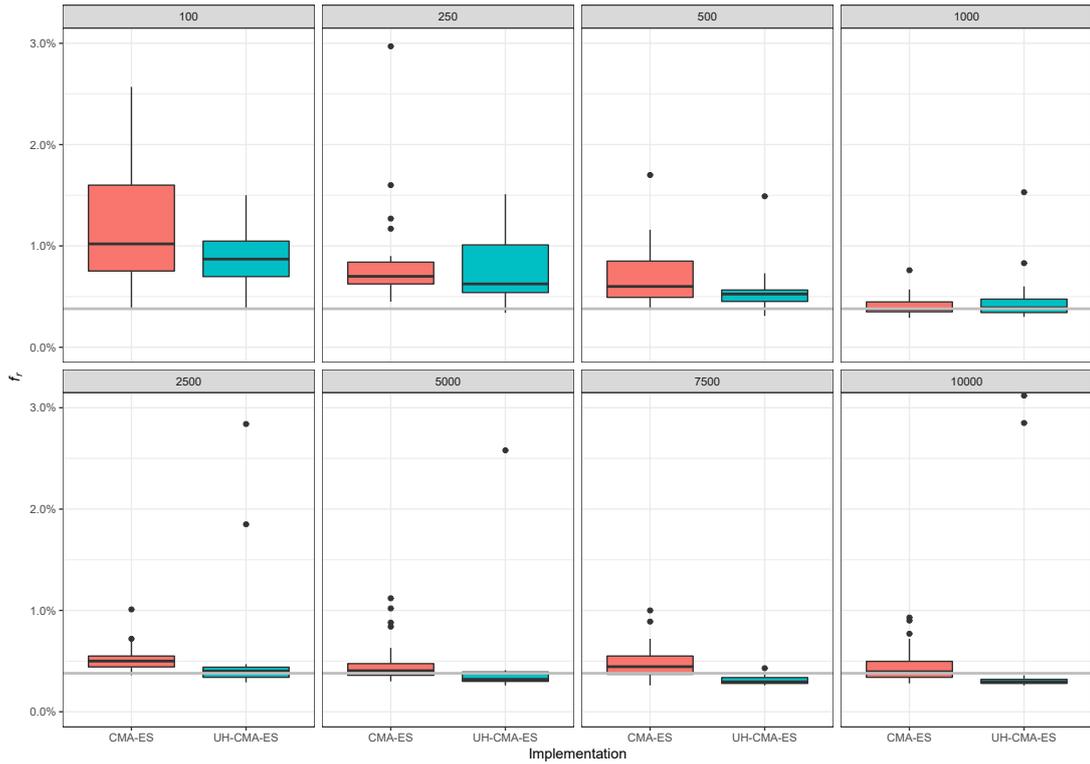


Figure 6.6: Expected time to clear the infection by implementation and mathematical model runs. A grey line representing the corresponding value of the optimal fixed-dose treatment is added for comparison. Treatments with failure rates over 3% are omitted from the plot.

In Figure 6.7 we have presented a scatterplot where on the x-axis we have the expected time to clear and on the y-axis, we have f_r . When looking at the UH-CMA-ES points we can see that with $r_{\text{mathmodel}}$ ranging from 100 to 500, the distance between the points is low, so they form a cluster and the covariance between them is positive. When using CMA-ES the points are not that close in proximity with bigger distance, and we have a higher noise even when $r_{\text{mathmodel}} = 10\ 000$. In relation to the aim of this chapter, using UH-CMA-ES with $r_{\text{mathmodel}} = 1000$ would be the optimal choice where the failure rate and the expected time to clear infection is lower than the fixed-dose one, and the computational time is 10 times faster. As there are outliers present, even if the experiment needs to be run multiple times it will still be more efficient than running a single experiment without UH-CMA-ES with $10\ 000\ r_{\text{mathmodel}}$.

When $r_{\text{mathmodel}} = 1000$ just over half of the final reevaluated treatments have a $f_r \leq 0.38\%$ (the failure rate of the best fixed-dose treatment). Therefore, if the experiment was run 4 times, the chance of a solution better than the best fixed-dose one is $\geq 95\%$ or 3 times for a change of 89.8%.

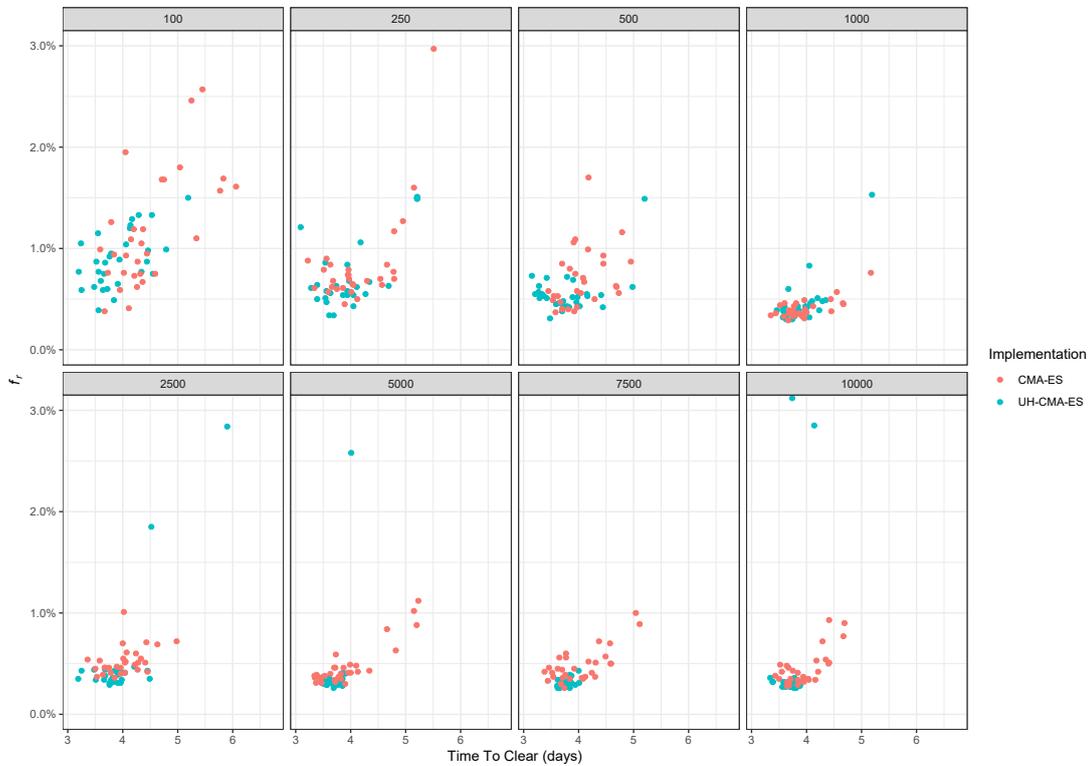


Figure 6.7: Expected time to clear the infection and failure rate scatter plot by implementation and mathematical model runs.

6.5.3 Treatments

In Figure 6.8 the same treatments from Figure 5.5 are plotted with the fixed-dose treatment shown in black. The failure rates and the confidence intervals are presented on the top right-hand side of each panel. In Chapter 5 in the experiments where there is a single bacteria present, and the antibiotics are taken orally (same model as experiments here) the best failure rate with MIC=8 and $A_{\text{total}} = 150$ is 0.28%. In the set of results here, we achieve $f_r = 0.27\%$ with the same computational budget as in Chapter 5 and achieve $f_r = 0.26\%$ with

the smaller budget of $r_{\text{mathmodel}} = 7500$ and half the budget of $r_{\text{mathmodel}} = 5000$. Even when taking into account the confidence interval of all treatments being, $[+/- 0.01\%]$ we are producing a statistically better result using UH-CMA-ES.

When it comes to the treatments themselves, we have similar patterns to the one observed in Chapter 5 — high first dose, similar middle doses and low last dose. Most of the treatments have a length of 8 days, with the exceptions being the ones where the $r_{\text{mathmodel}} = 100$ and 250 . This treatment length of 8 days aligns with the discoveries from the last chapter.

When we use UH-CMA-ES with $r_{\text{mathmodel}} \geq 5000$ we can see that the treatments converge to the same solution - first dose between 25mg and 30mg, middle doses at around 20mg, and last dose at day 8 at around 10mg. This is slightly different to the treatment we observed in Chapter 5 as there the last dose at day 8 is closer to 15mg. However, the treatments generated by UH-CMA-ES are in line with our hypothesis of the extrapolated optimised dose regime — first dose of $3 \times x$ mg, followed by $n - 2$ doses of $2 \times x$ mg, followed by a final dose of x mg.

When comparing UH-CMA-ES and CMA-ES, there is a lot more fluctuation when using CMA-ES in the doses - for example CMA-ES with 10 000 evaluations's first doses are ranging from 25mg to 30mg, and then the same can be observed for the rest of the doses in the treatment. In comparison, the solutions for UH-CMA-ES with 10 000 evaluations are almost identical. Similar observation can be made for UH-CMA-ES with 7500 and 5000 evaluations and CMA-ES with 7500 and 5000 evaluations — here for CMA-ES with 7500 evaluations for one of the treatments we even have a higher dose of 25mg at day 6. For UH-CMA-ES with 2500 and 1000 evaluations, the treatments

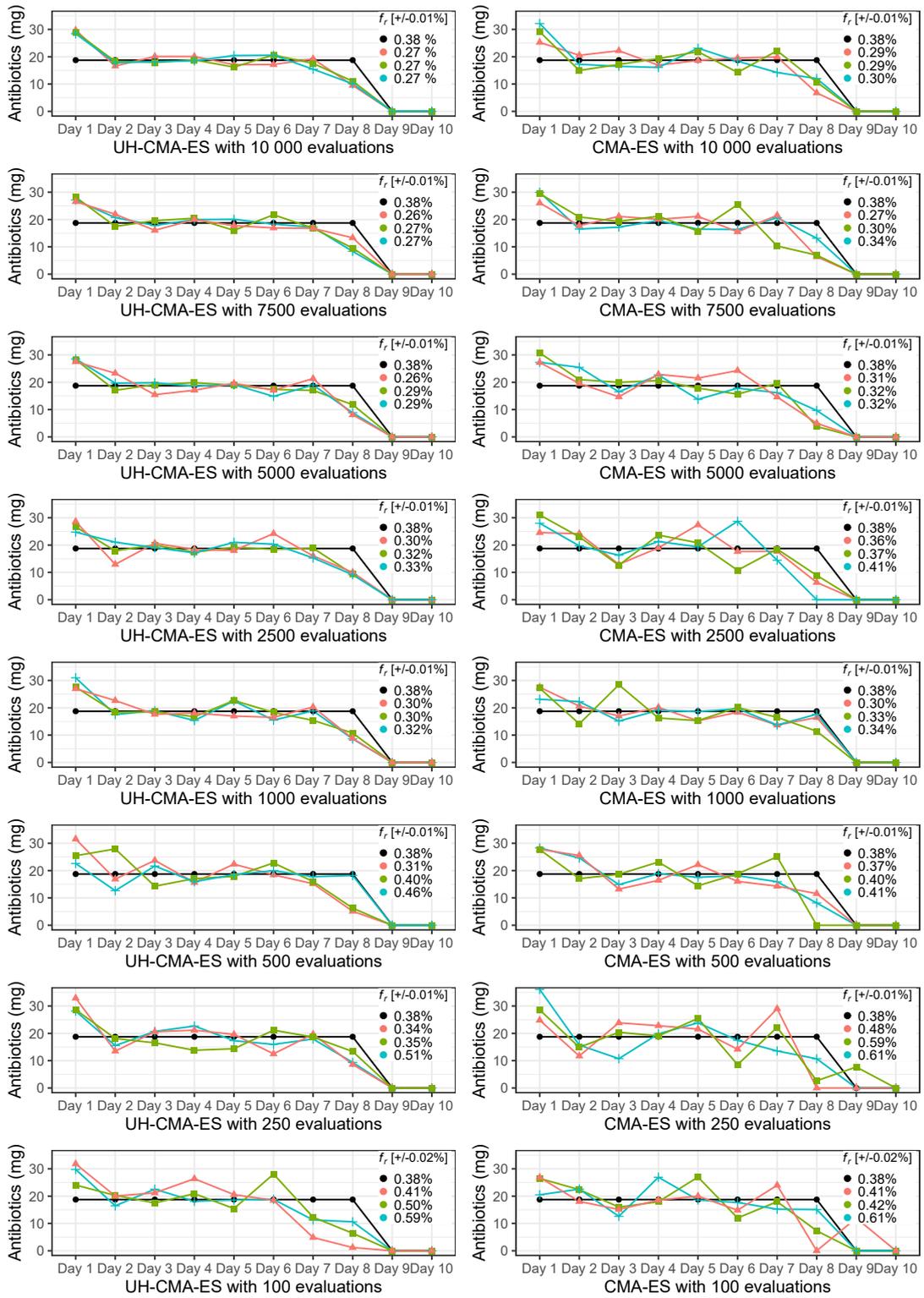


Figure 6.8: Comparison of the dosage profiles of the three of the best treatments against the best fixed-dose treatment (in black). Failure rates f_r are listed on the right-hand side, with confidence intervals in square brackets.

are still aligned with one another, but we start seeing even more different from one another when using the CMA-ES algorithm. When we go into lower evaluation count than 500, the treatments generated with both UH-CMA-ES and CMA-ES become less aligned to one another, with f_r being worse than the fixed-dose treatment's one on average.

As suggested in the previous subsection, selecting $r_{\text{mathmodel}} = 1000$ and using an uncertainty handling mechanism for the optimal experimental setup, we can see in the panel that the treatments have a high correlation with one another between $r_{\text{mathmodel}} = 10\,000$ to $r_{\text{mathmodel}} = 1000$. This shows that we can achieve similar successful treatments that produce f_r better than the best fixed-dose one with a smaller computational budget.

6.6 CONCLUSION

The main goal of this chapter is to investigate the effectiveness of noise handling, at different noise levels, and compare it to experiments without the noise handling mechanism. We implemented the CMA-ES algorithm with the uncertainty handling UH-CMA-ES and applied it to the mathematical model from Chapter 5, where there is only one type of bacteria present and the drugs are taken orally. The specific experimental setup we chose was MIC with value 8 and maximum total antibiotics $A_{\text{total}} \leq 150$.

As the mathematical model is the bottleneck in the runtime of our experiments, we aim to minimise the number of mathematical model runs needed for each experiment. A selection of eight values for the number of mathematical model runs was made, and the results were tested in terms of that as well as if the uncertainty handling was implemented or not with the CMA-ES algorithm. Noise ranges from 10% down to about 1%; computational cost varies by a factor of 100.

It is important to mention that noise handling has different effects in different levels of noise. When the noise is low ($r_{\text{mathmodel}}$ is between 5000 and 10 000)

and moderate ($r_{\text{mathmodel}}$ is 1000 or 2500) it performs well and is on average better than the best fixed-dose treatment. However, when the noise is high ($r_{\text{mathmodel}}$ is from 100 to 500) the fixed-dose treatment is better on average. In addition, UH-CMA-ES produces more outliers than CMA-ES suggesting the higher robustness of CMA-ES, however, UH-CMA-ES produces better treatments in comparison.

We found that with only 1000 runs of the mathematical model (in comparison to 10 000 in the previous chapter), we can achieve results that are better than the best fixed-dose treatment available in terms of failure rate, and with 5 000 evaluations we can produce a better result than the best treatment for this experimental setup in comparison with Chapter 5. As the experiments are already very computationally heavy, even a reduction in half of the runtime is significant.

Part III

CONCLUSION

DISCUSSION

This chapter summarises the results obtained from the work in this thesis and discusses how these results relate to the global context. The limitations of this work and possible future work will also be discussed.

7.1 SUMMARY OF RESULTS

7.1.1 *Chapter 4*

In Chapter 4 we explored the addition of the PK/PD modelling as well as the approximate method Tau-leaping based on the Gillespie algorithm. The PK/PD modelling introduced more detailed mechanics on how the host processes the antibiotic, depending on the intake type — oral or intravenous. In Chapter 4 we only explored the scenario where the antibiotics are given orally and then absorbed in the blood, where they kill the bacteria.

In addition to that, two different types of multi-objective population-based evolutionary algorithms (NSGA-II and SMPSO) were tuned and applied. The analysis showed that the NSGA-II algorithm provided better results than SMPSO based on the hypervolume indicator. No statistical significance in the performance of the algorithms was found in terms of whether they used tuned parameters or not.

The treatments' characteristics were explored and found that the best-performing ones alternate between a very high dose and a lower dose, where the first dose was in most cases the highest one. This differed from previous studies

that found the optimum solution to have tapered doses instead. However, the change in the treatment's dose characteristic could be explained by the addition of the PK/PD modelling.

7.1.2 *Chapter 5*

The single-objective algorithm DE was used in Chapter 5, where the main objective was to produce treatments with a minimal failure rate and the available amount of total antibiotics was set as a constraint. The parameters of the mathematical model were revisited, following different laboratory studies and other mathematical models. We explored different levels of resistance (low, medium, high, very high) of the bacteria for the three scenarios — single bacteria strain with oral administration, single bacteria strain and intravenous administration, and two bacteria strains with oral administration. The best standard fixed-dose treatment was produced for each level of resistance. The resulting optimised regimes have varying daily doses and achieve an improved, lower failure rate of between 20% and 35% when compared to fixed-dose regimens with the same amount of drug. All optimised regimens, for n days in duration, started with a high dose on the first day, followed by $n \times 2$ doses that are roughly similar to the corresponding fixed-dose regimen, and tapering off with a lower dose on the final day. A general pattern was observed, where the first dose is $3 \times x$ mg, followed by $2 \times x$ mg and the last dose of x mg, where $2 \times x$ mg is the standard daily fixed dose from the standard fixed-dose treatment. This differed from the treatments from Chapter 4, but it can be explained by the change in the parameter values in the mathematical model.

7.1.3 *Chapter 6*

Chapter 6 explored another algorithm — CMA-ES, and the uncertainty handling technique UH-CMA-ES implemented for it. The main aim of this chapter

was to analyse the effectiveness of the noise (uncertainty) handling at different levels of noise in the objective function (failure rate) and compare that to experiments without noise handling. Only one of the scenarios from Chapter 5 is used for the experiments — infection with a single bacteria strain with oral administration of the antibiotics, where the bacteria have low levels of resistance to the antibiotic used. The same mathematical model and parameters are used.

We also aim to minimise the number of mathematical model runs required for each experiment. Eight values were chosen, and the results were examined — low levels of noise with a high number of runs of the mathematical model, moderate levels of noise where the mathematical model is run around half of the times in comparison with Chapter 5 and high levels of noise with a low number of runs of the mathematical model. We found that with moderate levels of noise, we can produce treatments that are better on average than the best fixed-dose treatment, no matter if the noise handling is enabled or not. With noise handling the probability of a treatment being better than the best fixed-dose treatment is higher, however, the chance for outliers is also higher. Nevertheless, we produced better treatments in terms of minimal failure rate in comparison with Chapter 5. The treatments' characteristics were similar to those in Chapter 5 with the same pattern of the first dose is $3 \times x$ mg, followed by $2 \times x$ mg and the last dose of x mg, where $2 \times x$ mg is the standard daily fixed dose from the standard fixed-dose treatment.

7.2 FUTURE WORK

While personalised medicine is already used in life-threatening diseases like cancer [129], [130], similar approaches are not generally established for bacterial and viral infections. The standard prescription issued by the pharmaceutical companies producing the drugs is often the one advised by medical staff. As we are discovering, the overall patient's health plays a role in how the

treatment progresses, and the standard prescription might not be optimum for all.

The results from Chapters 5 and 6 show an easy-to-implement real-world scenario approach to administering antibiotic treatments. However, it is important to note that in order for that to happen, extensive laboratory and hospital studies need to be conducted first. This will entail in an in vivo and/or in vitro testing of the hypothesis presented in the two chapters. While this is out of the scope of this thesis, such studies can evaluate the approach we have taken to make a valuable impact on how drugs are prescribed not only in relation to antibiotics but other types of drug prescriptions as well.

The final part of this research project would be to implement a computer program system that could be directly used by medical professionals to treat antibiotic infections. Using this system, we aim to increase the cured patients' rate with initial treatment, which will therefore minimise the antibiotics being prescribed as additional treatments would not be necessary. This tackles the overuse and misuse of antibiotics scenario that contributes to a number of resistance cases. The system would present the medical professional with several options, but leave decision-making to them. However, it is important that the main limitations of this approach be listed:

- Computational budget — not every medical facility might have the necessary equipment to run our system, however a solution for that could be a lookup table of pre-generated treatments that the medical staff can use instead.
- Patient compliance — patients sometimes forget to take their medicine or stop taking them shortly after they feel better with standard treatments. Having different daily doses might be challenging for some patients, however, medical professionals can ensure the treatment is taken appropriately or is packaged in a way which it is easy to follow. Further studies on missed doses can provide interesting insight into such scenarios.

- Establishing the level of resistance the bacteria has to the prescribed antibiotic — while hospitals often do such tests, local clinics might not have the opportunity to do so.
- Mathematical Model Parameters — a more robust way of collecting the parameter values in a lab setting would benefit greatly from the performance of this approach.

7.3 CONCLUDING REMARKS

While new antimicrobial and antibiotic drugs are researched and discovered, bacteria eventually mutate and develop resistance to them. Therefore, it is important that we use the available medicine in the most optimal manner. A new approach to prescriptions is presented in this work that can maximise the efficiency of treatments and thus minimise the overall antibiotics prescribed.

BIBLIOGRAPHY

- [1] W. H. Organisation. "High levels of antibiotic resistance found worldwide, new data shows". (2018), [Online]. Available: <https://www.who.int/news/item/29-01-2018-high-levels-of-antibiotic-resistance-found-worldwide-new-data-shows> (visited on 30/08/2022).
- [2] T. Thompson, "The staggering death toll of drug-resistant bacteria.", *Nature*, 2022.
- [3] W. H. Organisation. "The global leaders group host side event at un general assembly on antimicrobial resistance (amr)". (2022), [Online]. Available: [https://www.who.int/news/item/22-09-2022-the-global-leaders-group-host-side-event-at-un-general-assembly-on-antimicrobial-resistance-\(amr\)](https://www.who.int/news/item/22-09-2022-the-global-leaders-group-host-side-event-at-un-general-assembly-on-antimicrobial-resistance-(amr)) (visited on 22/09/2022).
- [4] B. Spellberg, J. G. Bartlett and D. N. Gilbert, "The future of antibiotics and resistance", *New England Journal of Medicine*, vol. 368, no. 4, pp. 299–302, 2013.
- [5] W. H. Organization *et al.*, "No time to wait: Securing the future from drug-resistant infections", Apr. 2019.
- [6] R. I. Aminov, "A brief history of the antibiotic era: Lessons learned and challenges for the future", *Frontiers in microbiology*, vol. 1, p. 134, 2010.
- [7] M. McKenna, "Antibiotic resistance: The last resort", *Nature News*, vol. 499, no. 7459, p. 394, 2013.
- [8] C. L. Ventola, "The antibiotic resistance crisis: Part 1: Causes and threats", *Pharmacy and therapeutics*, vol. 40, no. 4, p. 277, 2015.

- [9] U. G. Public Health England, *Health matters: Antimicrobial resistance*, 2015. [Online]. Available: <https://www.gov.uk/government/publications/health-matters-antimicrobial-resistance/health-matters-antimicrobial-resistance> (visited on 30/09/2010).
- [10] A. R. Coates, G. Halls and Y. Hu, "Novel classes of antibiotics or more of the same?", *British journal of pharmacology*, vol. 163, no. 1, pp. 184–194, 2011.
- [11] P. A. Smith, M. F. Koehler *et al.*, "Optimized arylomycins are a new class of gram-negative antibiotics", *Nature*, vol. 561, no. 7722, p. 189, 2018.
- [12] I. K. Paterson, A. Hoyle, G. Ochoa, C. Baker-Austin and N. Taylor, "Optimising antibiotic usage to treat bacterial infections", *Scientific Reports*, vol. 6, p. 37853, Nov. 2016. DOI: 10.1038/srep37853.
- [13] M. Goranova, M. A. Contreras-Cruz, A. Hoyle and G. Ochoa, "Optimising antibiotic treatments with multi-objective population-based algorithms", in *2020 IEEE Congress on Evolutionary Computation*, IEEE, 2020, pp. 1–7. DOI: 10.1109/CEC48606.2020.9185489.
- [14] L. Katz and R. H. Baltz, "Natural product discovery: Past, present, and future", *Journal of Industrial Microbiology and Biotechnology*, vol. 43, no. 2-3, pp. 155–176, 2016.
- [15] W. H. O. *et al.*, "Global action plan to control the spread and impact of antimicrobial resistance.", 2017.
- [16] M. I. Hutchings, A. W. Truman and B. Wilkinson, "Antibiotics: Past, present and future", *Current opinion in microbiology*, vol. 51, pp. 72–80, 2019.
- [17] C. M. Parry, V. A. Ho, P. V. B. Bay *et al.*, "Randomized controlled comparison of ofloxacin, azithromycin, and an ofloxacin-azithromycin combination for treatment of multidrug-resistant and nalidixic acid-

- resistant typhoid fever”, *Antimicrobial agents and chemotherapy*, vol. 51, no. 3, pp. 819–825, 2007.
- [18] C. McKenzie, “Antibiotic dosing in critical illness”, *Journal of antimicrobial chemotherapy*, vol. 66, no. suppl_2, pp. ii25–ii31, 2011.
- [19] D. Mara and N. J. Horan, *Handbook of water and wastewater microbiology*. Elsevier, 2003.
- [20] R. Gupta and N. Gupta, “Prokaryotic cell structure and function”, in *Fundamentals of Bacterial Physiology and Metabolism*, Springer, 2021, pp. 43–79.
- [21] S. Watford and S. J. Warrington, “Bacterial dna mutations”, 2017.
- [22] S. M. Soucy, J. Huang and J. P. Gogarten, “Horizontal gene transfer: Building the web of life”, *Nature Reviews Genetics*, vol. 16, no. 8, pp. 472–482, 2015.
- [23] Y. Nakamura, T. Itoh, H. Matsuda and T. Gojobori, “Biased biological functions of horizontally transferred genes in prokaryotic genomes”, *Nature genetics*, vol. 36, no. 7, pp. 760–766, 2004.
- [24] F. Svara and D. J. Rankin, “The evolution of plasmid-carried antibiotic resistance”, *BMC evolutionary biology*, vol. 11, pp. 1–10, 2011.
- [25] C. M. Thomas and K. M. Nielsen, “Mechanisms of, and barriers to, horizontal gene transfer between bacteria”, *Nature reviews microbiology*, vol. 3, no. 9, pp. 711–721, 2005.
- [26] A. D. Russell, “Types of antibiotics and synthetic antimicrobial agents”, *Hugo and Russell’s Pharmaceutical Microbiology*, pp. 152–186, 2004.
- [27] C. Roose-Amsaleg and A. M. Laverman, “Do antibiotics have environmental side-effects? impact of synthetic antibiotics on biogeochemical processes”, *Environmental Science and Pollution Research*, vol. 23, no. 5, pp. 4000–4012, 2016.

- [28] L. M. WILSON, "Etiology of bacterial endocarditis: Before and since the introduction of antibiotics", *Annals of Internal Medicine*, vol. 58, no. 6, pp. 946–952, 1963.
- [29] J. L. Hobman and L. C. Crossman, "Bacterial antimicrobial metal ion resistance", *Journal of medical microbiology*, vol. 64, no. 5, pp. 471–497, 2015.
- [30] E. N. Olsufyeva and V. S. Yankovskaya, "Main trends in the design of semi-synthetic antibiotics of a new generation", *Russian Chemical Reviews*, vol. 89, no. 3, p. 339, 2020.
- [31] D. I. Andersson, "Persistence of antibiotic resistant bacteria", *Current opinion in microbiology*, vol. 6, no. 5, pp. 452–456, 2003.
- [32] B. Kowalska-Krochmal and R. Dudek-Wicher, "The minimum inhibitory concentration of antibiotics: Methods, interpretation, clinical relevance", *Pathogens*, vol. 10, no. 2, p. 165, 2021.
- [33] A.-P. Magiorakos, A. Srinivasan, R. B. Carey *et al.*, "Multidrug-resistant, extensively drug-resistant and pandrug-resistant bacteria: An international expert proposal for interim standard definitions for acquired resistance", *Clinical microbiology and infection*, vol. 18, no. 3, pp. 268–281, 2012.
- [34] V. I. Band and D. S. Weiss, "Heteroresistance: A cause of unexplained antibiotic treatment failure?", *PLoS pathogens*, vol. 15, no. 6, e1007726, 2019.
- [35] D. I. Andersson, H. Nicoloff and K. Hjort, "Mechanisms and clinical relevance of bacterial heteroresistance", *Nature Reviews Microbiology*, vol. 17, no. 8, pp. 479–496, 2019.
- [36] L. Czaplewski, R. Bax, M. Clokie *et al.*, "Alternatives to antibiotics—a pipeline portfolio review", *The Lancet infectious diseases*, vol. 16, no. 2, pp. 239–251, 2016.

- [37] G. F. Hatfull, R. M. Dedrick and R. T. Schooley, "Phage therapy for antibiotic-resistant bacterial infections", *Annual Review of Medicine*, vol. 73, pp. 197–211, 2022.
- [38] M. Rima, M. Rima, Z. Fajloun, J.-M. Sabatier, B. Bechinger and T. Naas, "Antimicrobial peptides: A potent alternative to antibiotics", *Antibiotics*, vol. 10, no. 9, p. 1095, 2021.
- [39] G. G. Khachatourians, "Agricultural use of antibiotics and the evolution and transfer of antibiotic-resistant bacteria", *Cmaj*, vol. 159, no. 9, pp. 1129–1136, 1998.
- [40] N. D. Friedman, E. Temkin and Y. Carmeli, "The negative impact of antibiotic resistance", *Clinical Microbiology and Infection*, vol. 22, no. 5, pp. 416–422, 2016.
- [41] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions", *The journal of physical chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.
- [42] G. Ochoa, L. A. Christie, A. E. Brownlee and A. Hoyle, "Multi-objective evolutionary design of antibiotic treatments", *Artificial Intelligence in Medicine*, vol. 102, p. 101759, 2020.
- [43] C. T. Bergstrom, M. Lo and M. Lipsitch, "Ecological theory suggests that antimicrobial cycling will not reduce antimicrobial resistance in hospitals", *Proceedings of the National Academy of Sciences*, vol. 101, no. 36, pp. 13285–13290, 2004.
- [44] A. Teillant, S. Gandra, D. Barter, D. J. Morgan and R. Laxminarayan, "Potential burden of antibiotic resistance on surgery and cancer chemotherapy antibiotic prophylaxis in the usa: A literature review and modelling study", *The Lancet infectious diseases*, vol. 15, no. 12, pp. 1429–1437, 2015.
- [45] L. N. Lili, N. F. Britton and E. J. Feil, "The persistence of parasitic plasmids", *Genetics*, vol. 177, no. 1, pp. 399–405, 2007.

- [46] A. C. Birkegård, T. Halasa, N. Toft, A. Folkesson and K. Græsbøll, “Send more data: A systematic review of mathematical models of anti-microbial resistance”, *Antimicrobial Resistance & Infection Control*, vol. 7, no. 1, p. 117, 2018.
- [47] K. Græsbøll, S. S. Nielsen, N. Toft and L. E. Christiansen, “How fitness reduced, antimicrobial resistant bacteria survive and spread: A multiple pig-multiple bacterial strain model”, *PLoS One*, vol. 9, no. 7, e100458, 2014.
- [48] E. M. DâAgata, P. Magal, D. Olivier, S. Ruan and G. F. Webb, “Modeling antibiotic resistance in hospitals: The impact of minimizing treatment duration”, *Journal of theoretical biology*, vol. 249, no. 3, pp. 487–499, 2007.
- [49] H.-R. Sun, X. Lu and S. Ruan, “Qualitative analysis of models with different treatment protocols to prevent antibiotic resistance”, *Mathematical biosciences*, vol. 227, no. 1, pp. 56–67, 2010.
- [50] E.-G. Talbi, *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.
- [51] K. Sørensen, “Metaheuristicsâthe metaphor exposed”, *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, 2015.
- [52] J. Pearl, *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Longman Publishing Co., Inc., 1984.
- [53] K. Sørensen and F. Glover, “Metaheuristics”, *Encyclopedia of operations research and management science*, vol. 62, pp. 960–970, 2013.
- [54] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptual comparison”, *ACM computing surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.
- [55] S. Lim and J. Zhu, “Integrated data envelopment analysis: Global vs. local optimum”, *European Journal of Operational Research*, vol. 229, no. 1, pp. 276–278, 2013.

- [56] Z. Beheshti and S. M. H. Shamsuddin, "A review of population-based meta-heuristic algorithms", *Int. J. Adv. Soft Comput. Appl*, vol. 5, no. 1, pp. 1–35, 2013.
- [57] A. Lazar, "Heuristic knowledge discovery for archaeological data using genetic algorithms and rough sets", in *Heuristic and optimization for knowledge discovery*, IGI Global, 2002, pp. 263–278.
- [58] K. Sörensen, M. Sevaux and F. Glover, "A history of metaheuristics", in *Handbook of heuristics*, Springer, 2018, pp. 791–808.
- [59] T. Upcraft and M. Guo, "Phytoremediation value chains and modeling", *Sustainable remediation of contaminated soil and groundwater*, pp. 325–366, 2020.
- [60] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*. Springer, 2015.
- [61] C. Blum, R. Chiong, M. Clerc *et al.*, "Evolutionary optimization", in *Variants of evolutionary algorithms for real-world applications*, Springer, 2012, pp. 1–29.
- [62] D. Simon, *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.
- [63] J. Sun, H. Zhang, Q. Zhang and H. Chen, "Balancing exploration and exploitation in multiobjective evolutionary optimization", in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018, pp. 199–200.
- [64] B. Selman and C. P. Gomes, "Hill-climbing search", *Encyclopedia of cognitive science*, vol. 81, p. 82, 2006.
- [65] L. Rastrigin, "The convergence of the random search method in the extremal control of a many parameter system", *Automaton & Remote Control*, vol. 24, pp. 1337–1342, 1963.
- [66] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen *et al.*, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007, vol. 5.

- [67] A. Osyczka, "Multicriteria optimization for engineering design", in *Design optimization*, Elsevier, 1985, pp. 193–227.
- [68] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii", *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [69] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms", *Evolutionary computation*, vol. 2, no. 3, pp. 221–248, 1995.
- [70] A. Benitez-Hidalgo, A. J. Nebro, J. Garcia-Nieto, I. Oregi and J. Del Ser, "Jmetalpy: A python framework for multi-objective optimization with metaheuristics", *arXiv preprint arXiv:1903.02915*, 2019.
- [71] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. C. Coello, F. Luna and E. Alba, "Smpso: A new pso-based metaheuristic for multi-objective optimization", in *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*, IEEE, 2009, pp. 66–73.
- [72] A. J. Nebro, J. J. Durillo, J. García-Nieto *et al.*, "Extending the speed-constrained multi-objective pso (smpso) with reference point based preference articulation", in *International Conference on Parallel Problem Solving from Nature*, Springer, 2018, pp. 298–310.
- [73] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces", *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [74] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art", *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 4–31, 2010.
- [75] S. Das, S. S. Mullick and P. Suganthan, "Recent advances in differential evolution — an updated survey", *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016, ISSN: 2210-6502. DOI: <https://doi.org/10.1016/j.swevo.2016.01.004>.

- [76] P. Virtanen, R. Gommers and T. e. a. Oliphant, “Scipy 1.0: Fundamental algorithms for scientific computing in python”, *Nature Methods*, vol. 17, 261â272, 2020. DOI: 10.1038/s41592-019-0686-2.
- [77] N. Hansen and A. Auger, “Cma-es: Evolution strategies and covariance matrix adaptation”, in *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, 2011, pp. 991–1010.
- [78] N. Hansen and S. Kern, “Evaluating the cma evolution strategy on multimodal test functions”, in *International conference on parallel problem solving from nature*, Springer, 2004, pp. 282–291.
- [79] P. N. Suganthan, N. Hansen, J. J. Liang *et al.*, “Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization”, *KanGAL report*, vol. 2005005, no. 2005, p. 2005, 2005.
- [80] N. Hansen, A. Auger, R. Ros, S. Finck and P. Pošík, “Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009”, in *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, 2010, pp. 1689–1696.
- [81] A. Auger and N. Hansen, “Performance evaluation of an advanced local search evolutionary algorithm”, in *2005 IEEE congress on evolutionary computation*, IEEE, vol. 2, 2005, pp. 1777–1784.
- [82] N. Hansen, “The cma evolution strategy: A tutorial”, *arXiv preprint arXiv:1604.00772*, 2016.
- [83] N. Hansen, Y. Akimoto and P. Baudis, *CMA-ES/pycma on Github*, Zenodo, DOI:10.5281/zenodo.2559634, Feb. 2019. DOI: 10.5281/zenodo.2559634. [Online]. Available: <https://doi.org/10.5281/zenodo.2559634>.
- [84] J. M. Cicchese, E. Pienaar, D. E. Kirschner and J. J. Linderman, “Applying optimization algorithms to tuberculosis antibiotic treatment regimens”, *Cellular and Molecular Bioengineering*, vol. 10, no. 6, pp. 523–535, Dec. 2017.

- [85] P. J. Colin, D. J. Eleveld and A. H. Thomson, "Genetic algorithms as a tool for dosing guideline optimization: Application to intermittent infusion dosing for vancomycin in adults", *CPT: pharmacometrics and systems pharmacology*, vol. 9, no. 5, pp. 294–302, 2020.
- [86] A. Petrovski and J. McCall, "Multi-objective optimisation of cancer chemotherapy using evolutionary algorithms", in *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, 2001, pp. 531–545.
- [87] A. Petrovski, B. Sudha and J. McCall, "Optimising cancer chemotherapy using particle swarm optimisation and genetic algorithms", in *International Conference on Parallel Problem Solving from Nature*, Springer, 2004, pp. 633–641.
- [88] D. Petrovic, M. Morshed and S. Petrovic, "Multi-objective genetic algorithms for scheduling of radiotherapy treatments for categorised cancer patients", *Expert Systems with Applications*, vol. 38, no. 6, pp. 6994–7002, 2011.
- [89] V. Bevilacqua, G. Mastronardi, F. Menolascina, P. Pannarale and A. Pedone, "A novel multi-objective genetic algorithm approach to artificial neural network topology optimisation: The breast cancer classification problem", in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, IEEE, 2006, pp. 1958–1965.
- [90] J. M. Cicchese, E. Pienaar, D. E. Kirschner and J. J. Linderman, "Applying optimization algorithms to tuberculosis antibiotic treatment regimens", *Cellular and molecular bioengineering*, vol. 10, no. 6, pp. 523–535, 2017.
- [91] M. C. Kjellsson, L. E. Via, A. Goh *et al.*, "Pharmacokinetic evaluation of the penetration of antituberculosis agents in rabbit pulmonary lesions", *Antimicrobial agents and chemotherapy*, vol. 56, no. 1, pp. 446–457, 2012.

- [92] G. Ochoa, M. Villasana and E. K. Burke, "An evolutionary approach to cancer chemotherapy scheduling", *Genetic Programming and Evolvable Machines*, vol. 8, no. 4, pp. 301–318, 2007.
- [93] G. Ochoa and M. Villasana, "Population-based optimization of cytostatic/cytotoxic combination cancer chemotherapy", *Soft Computing*, vol. 17, no. 6, pp. 913–924, 2013.
- [94] M. Villasana, G. Ochoa and S. Aguilar, "Modeling and optimization of combined cytostatic and cytotoxic cancer chemotherapy", *Artificial intelligence in medicine*, vol. 50, no. 3, pp. 163–173, 2010.
- [95] N. H. Luong, T. Alderliesten, A. Bel, Y. Niatsetski and P. A. Bosman, "Application and benchmarking of multi-objective evolutionary algorithms on high-dose-rate brachytherapy planning for prostate cancer treatment", *Swarm and Evolutionary Computation*, vol. 40, pp. 37–52, 2018.
- [96] A. Bouter, N. H. Luong, C. Witteveen, T. Alderliesten and P. A. Bosman, "The multi-objective real-valued gene-pool optimal mixing evolutionary algorithm", in *Proceedings of the Genetic and Evolutionary Computation Conference*, ACM, 2017, pp. 537–544.
- [97] M. E. Levison and J. H. Levison, "Pharmacokinetics and pharmacodynamics of antibacterial agents", *Infectious Disease Clinics*, vol. 23, no. 4, pp. 791–815, 2009.
- [98] B. Meibohm and H. Derendorf, "Basic concepts of pharmacokinetic/pharmacodynamic (pk/pd) modelling", *International journal of clinical pharmacology and therapeutics*, vol. 35, no. 10, pp. 401–413, 1997.
- [99] D. T. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems", *The Journal of Chemical Physics*, vol. 115, no. 4, pp. 1716–1733, 2001.

- [100] C. M. Fonseca, L. Paquete and M. López-Ibáñez, “An improved dimension-sweep algorithm for the hypervolume indicator”, in *2006 IEEE international conference on evolutionary computation*, IEEE, 2006, pp. 1157–1163.
- [101] M. López-Ibáñez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari and T. Stützle, “The irace package: Iterated racing for automatic algorithm configuration”, *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.
- [102] L. C. Bezerra, M. López-Ibáñez and T. Stützle, “Automatic component-wise design of multiobjective evolutionary algorithms”, *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 403–417, 2015.
- [103] W. J. Conover and W. J. Conover, “Practical nonparametric statistics”, 1980.
- [104] N. I. for Health and C. Excellence, *Summary of antimicrobial prescribing guidance â managing common infections*, Nov. 2020.
- [105] I. IDEXX Laboratories, *Microbiology guide to interpreting minimum inhibitory concentration (mic)*, Mar. 2019. [Online]. Available: <https://www.idexx.com/files/microbiology-guide-interpreting-mic.pdf>.
- [106] A. Hoyle, D. Cairns, I. Paterson, S. McMillan, G. Ochoa and A. P. Desbois, “Optimising efficacy of antibiotics against systemic infection by varying dosage quantities and times”, *PLoS computational biology*, vol. 16, no. 8, e1008037, 2020.
- [107] R. Sender, S. Fuchs and R. Milo, “Revised estimates for the number of human and bacteria cells in the body”, *PLoS biology*, vol. 14, no. 8, e1002533, 2016.
- [108] P. Erah, A. Goddard, D. Barrett, P. Shaw and R. Spiller, “The stability of amoxicillin, clarithromycin and metronidazole in gastric juice: Relevance to the treatment of helicobacter pylori infection.”, *The Journal of antimicrobial chemotherapy*, vol. 39, no. 1, pp. 5–12, 1997.

- [109] A. Handel, E. Margolis and B. R. Levin, “Exploring the role of the immune response in preventing antibiotic resistance”, *Journal of theoretical biology*, vol. 256, no. 4, pp. 655–662, 2009.
- [110] P. Wang, L. Robert, J. Pelletier *et al.*, “Robust growth of escherichia coli”, *Current biology*, vol. 20, no. 12, pp. 1099–1103, 2010.
- [111] S. K. Lam, A. Pitrou and S. Seibert, “Numba: A llvm-based python jit compiler”, in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 2015, pp. 1–6.
- [112] J. Lampinen, “A constraint handling approach for the differential evolution algorithm”, in *Congress on Evolutionary Computation. CEC’02*, vol. 2, 2002, pp. 1468–1473. DOI: 10.1109/CEC.2002.1004459.
- [113] C. R. Harris, K. J. Millman, S. J. van der Walt and et. a; “Array programming with NumPy”, *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2.
- [114] P. Rakshit, A. Konar and S. Das, “Noisy evolutionary optimization algorithms—a comprehensive survey”, *Swarm and Evolutionary Computation*, vol. 33, pp. 18–45, 2017.
- [115] H.-G. Beyer, “Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice”, *Computer methods in applied mechanics and engineering*, vol. 186, no. 2-4, pp. 239–267, 2000.
- [116] D.-C. Dang and P. K. Lehre, “Efficient optimisation of noisy fitness functions with population-based evolutionary algorithms”, in *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, 2015, pp. 62–68.
- [117] D. V. Arnold and H.-G. Beyer, *Noisy optimization with evolution strategies*. Springer Science & Business Media, 2002, vol. 8.
- [118] T. Then and E. K. P. Chong, “Genetic algorithms in noisy environment”, in *Proceedings of 1994 9th IEEE International Symposium on Intelligent Control*, IEEE, 1994, pp. 225–230.

- [119] S. Das, A. Konar and U. K. Chakraborty, "Improved differential evolution algorithms for handling noisy optimization problems", in *2005 IEEE Congress on Evolutionary Computation*, IEEE, vol. 2, 2005, pp. 1691–1698.
- [120] V. Heidrich-Meisner and C. Igel, "Uncertainty handling cma-es for reinforcement learning", in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, 2009, pp. 1211–1218.
- [121] J. W. Kruisselbrink, E. Reehuis, A. Deutz, T. Bäck and M. Emmerich, "Using the uncertainty handling cma-es for finding robust optima", in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp. 877–884.
- [122] T. Glasmachers and C. Igel, "Uncertainty handling in model selection for support vector machines", in *International Conference on Parallel Problem Solving from Nature*, Springer, 2008, pp. 185–194.
- [123] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation", in *Proceedings of IEEE international conference on evolutionary computation*, IEEE, 1996, pp. 312–317.
- [124] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies", *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [125] N. Hansen, S. D. Müller and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)", *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.
- [126] N. Hansen, A. S. Niederberger, L. Guzzella and P. Koumoutsakos, "A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion", *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 180–197, 2008.

- [127] N. Hansen, yoshihikoueno, ARF1 *et al.*, *Cma-es/pycma: R3.2.2*, version r3.2.2, Mar. 2022. DOI: 10.5281/zenodo.6370326. [Online]. Available: <https://doi.org/10.5281/zenodo.6370326>.
- [128] Z. Li and Q. Zhang, "What does the evolution path learn in cma-es?", in *International Conference on Parallel Problem Solving from Nature*, Springer, 2016, pp. 751–760.
- [129] I. A. Cree, "Designing personalised cancer treatments", *Journal of Controlled Release*, vol. 172, no. 2, pp. 405–409, 2013.
- [130] A. Kerr, C. Key Chekar, E. Ross, J. Swallow and S. Cunningham-Burley, *Personalised cancer medicine: Future crafting in the genomic era*. Manchester University Press, 2021.