Title

Designing Worldwide Clinical Trials with Multiple Objectives using Nature-Inspired Metaheuristic Algorithms

Author

Schepps, Mitchell Aaron

Publication Date

2023

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Designing Worldwide Clinical Trials with Multiple Objectives using Nature-Inspired Metaheuristic Algorithms

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Biostatistics

by

Mitchell Aaron Schepps

2023

ABSTRACT OF THE DISSERTATION

Designing Worldwide Clinical Trials with Multiple Objectives using Nature-Inspired Metaheuristic Algorithms

by

Mitchell Aaron Schepps
Doctor of Philosophy in Biostatistics
University of California, Los Angeles, 2023
Professor Weng Kee Wong, Chair

Problems in healthcare and medicine are worldwide and multifaceted. This dissertation explores optimal and efficient designs in global and other important clinical trials with nature-inspired metaheuristic algorithms. The primary objective of this research is to showcase novel uses of metaheuristics in two real world applications: first, we collaborate with physicians and help patients with bipolar disorder by designing optimal sampling times for sustained-release lithium, a vital medication using pharmacokinetics/pharmacodynamics nonlinear mixed effects models and second, to optimize worldwide clinical trial patient recruitment plans within tight, real world complex regulatory and budgetary constraints. We determine interesting optimal designs using metaheuristics in both single and multiple objective situations with complex constraints. We show systematic analyses which can be used to solve and enhance the implementation and interpretability of the discussed real life clinical trial designs. The results from this dissertation are implemented in current software and offer novel insights into the design of experiments and patient recruitment strategies, contributing to the advancement of medical treatments and healthcare practices.

The dissertation of Mitchell Aaron Schepps is approved.

Zhe Fei

Hongquan Xu

Hua Zhou

Weng Kee Wong, Committee Chair

University of California, Los Angeles

2023

To my family and those who are family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First of all, I am forever thankful for Dr. Wong being my advisor and for his continuous support from day one. I have grown under his leadership, and I would recommend Dr. Wong as an advisor to every student. I could not be here today if he did not support me as much as he did. He has been patient with me, invited me to speak on the biggest stages, and we have worked on important projects I never could have dreamt of. He introduced me to this field of metaheuristics, which astonishes me. I appreciate my committee's enthusiasm, encouragement and permission to go up this research exploration. I also owe an insurmountable amount of gratitude to my supervisors at Amgen's Center for Design and Analysis, Dr. Volodymyr Anisimov and Matt Austin, for sponsoring me, mentoring me, and providing the motivation for my work on the worldwide multiple objective clinical trial patient recruitment problem. Dr. Anisimov, a former professor and head of the applied statistics department at the Kyiv National Taras Shevchenko University, and Matt Austin, Executive Director and Head of Data Science at Amgen, were excellent advisors, meeting with me weekly for three years and are major contributors to this dissertation.

Further, thank you to all the professors in the UCLA Department of Biostatistics. A close knit department, I would like to thank all of the professors who I have been the teaching assistant for: Dr. Hilary Aralis, Dr. Martin Lee, Dr. David Gjertson, Dr. Hua Zhou, and Dr. Rob Weiss. I have learned a lot in these different roles and have learned from you. I would also like to thank Dr. David Elashoff and Dr. Gang Li for mentoring me during my Masters' degree. Thank you also to Dr. Tom Belin for sponsoring and overseeing the completion of this work and for always being curious about the students' perspective. Further thank you to our Student Affairs Officer, Roxy Naranjo.

Prior to coming to UCLA, I am a byproduct of working under Dr. Tammy B. Harris and Dr. Eric Shiroma Jr. my two closest mentors from the NIH and was inspired by Dr. Connie Jo Mulligan from the University of Florida.

Lastly, I would like to thank the friends I have met, my mother, father, sister, and grandparents.

VITA

2009 - 2014 B.S. Statistics, B.A. Mathematics, University of Florida

2014 - 2017 Postbac Fellow, National Institute on Aging, National Institutes of Health

2017 - 2019 M.S. Biostatistics, University of California, Los Angeles

2019 - 2023 Teaching Assistant - Contemporary Health Issues, Foundations in Public Health - A B, Methods in Biostatistics - A, Introduction to Biostatistics - A B, Introduction to Data Science, Modeling Longitudinal Data, Statistical Computing

PUBLICATIONS

Mitchell Aaron Schepps, Weng Kee Wong, Matt Austin, Volodymyr Anisimov, "Optimizing Patient Recruitment in Global Clinical Trials using Nature-Inspired Metaheuristics." (Accepted with revisions).

Mitchell Aaron Schepps, Jérémy Seurat, France Mentré, and Weng Kee Wong, "Optimal Sampling Times for Sustained-Release Lithium in Patients with Bipolar Disorder." (Under review).

Mitchell Aaron Schepps, Weng Kee Wong, Matt Austin, Volodymyr Anisimov, "Using Nature-Inspired Metaheuristics to Generate Multiple Objective Efficient Designs Subject to Multiple Constraints for Clinical Trials." (To be submitted).

PREAMBLE

Biostatistics is a field of study with far reaching impact and implications for medicine and public health. This dissertation focuses on the innovative use of metaheuristic algorithms to optimally design worldwide clinical trials with multiple objectives subject to a set of user-specified complex constraints. Real life clinical trials are involved in multiple countries and sites simultaneously, and they are a taxing endeavor on patients and organizations. We present here, difficult optimization problems with computationally challenging nonlinear constraints and multiple objectives. Metaheuristic algorithms are flexible, easy to understand, and strong optimization algorithms because they solve problems traditional techniques cannot. These algortihms are often based on nature and fathomable equations. The use of metaheuristics opens up new avenues for improving the efficiency and effectiveness of clinical trials; thereby potentially accelerating the development of new and more effective medical treatments. This preamble sets the stage to uncover the myriad of ways that metaheuristics can revolutionize experimentation for clinical trials, and beyond to other areas of biostatistics.

CHAPTER 1

Introduction

1.1 Aims

The main aim of this work is to show the power of metaheuristic algorithms to solve multiple objective worldwide clinical trial design issues. Metaheuristics are often strategies employed when traditional techniques cannot. As the term metaheuristics implies, they are not developed to solve specific problems, but give approximate answers to problems with difficult assumptions. They are often related to equations seen in nature which gives them an advantage in multiple objective solutions. There are appropriate metaheuristic algorithms for our study to solve the optimization problem or select a more appropriate metaheuristic algorithm. Alternatively, the metaheuristic algorithm can be tuned and used appropriately.

We also provide tools to help clinical trial developers design efficiently for real life constraints. We demonstrate how clinical trial designs can benefit from use of nature-inspired metaheuristics with two real life clinical trial design problems. The first clinical trial concerns selection of optimal sampling times to ascertain the effects of lithium in a longitudinal study for a complicated pharmacokinetics/pharmacodynamics (PK/PD) nonlinear mixed effects model (NLMEM). The second demonstration is to develop cost-effective recruitment plans for a worldwide clinical trial when there may be multiple objectives in the trial and several types of constraints imposed. As we delve into the specific experimentation and overarching goals of this work, we set the stage for the implementation of metaheuristic algorithms for finding efficient recruitment plans and designs for various types of clinical trials.

Interestingly, metaheuristics are only used sparingly in biostatistical research. We hope our work will inform clinical trial researchers of the usefulness of metaheuristics for optimization purposes and stimulate further research in metaheuristics.

1.2 Outline of the dissertation

The outline of the dissertation is as follows. Section 1.3 introduces metaheuristic algorithms, and Sections 1.4 and 1.5 motivate the use of metaheuristics for designing two clinical trials. Chapter 2 discusses metaheuristics in greater detail with examples. We explain their allure in various fields and give examples on how they function. In Chapter 3, we demonstrate the utility of single and multiple objective metaheuristics to guide physicians on measuring lithium levels in patients with bipolar disorder. Lithium is often the first line treatment to aid in reducing the suicide potential seen in bipolar patients. It requires recurrent therapeutic drug monitoring due to a narrow therapeutic window with potential toxicities (Yacobi and Ornoy 2008, Grof 2010, Gitlin 2016). We use metaheuristics to design a longitudinal study to optimally estimate parameters in a complicated PK/PD NLMEM under multiple physician specified constraints. To this end, we use and enhance advanced, specialized PK/PD software now in an R package (Couffignal et al. 2019).

In Chapter 4, we apply metaheuristics innovatively to find an optimal recruitment plan for a worldwide clinical trial to meet pre-specified targets under a set of complex constraints. This is an important problem because frequently, recruiting enough patients in a timely manner is crucial; otherwise, the study is under-powered and results become unreliable. In extreme cases, the trials may be cancelled due to gross under enrollment (Senn 1997, Sully et al. 2013, Walters et al. 2017). In addition, there are many time-sensitive statistical complexities to be considered when planning a trial, including the number, location, and demographics of recruitment sites (Senn 1997). Chapter 5 summarizes the work and discusses limitations of our work.

1.3 What are metaheuristics?

Metaheuristics are optimization problem solving techniques capable to solve literally any optimization problem whether the objective function or constraints are linear, nonlinear, discontinuous, non-differentiable, multi-modal or high dimensional, etc. (Ezugwu et al. 2021). In complex problems, standard assumptions do not hold; for instance, the lasso penalty is non-differentiable (Lange et al. 2014). Exact or traditional optimization methods often rely on exploitation of these particular assumptions, such as linear or gradient based methods (Lange 2013, Sallan et al. 2015). Metaheuristics completely relax these assumptions and are able to solve problems only knowing the input and output (Blum et al. 2012). This means that the objective we only need to know knowledge of the objective function and constraints. Metaheuristics are often referred to as nature-inspired algorithms with mathematical equations built on metaphors seen in the natural environment, similar to sophisticated hunting or mating patterns of animals (Yang 2020). Two examples of such algorithms are inspired by evolutionary reproductive principles and the hunting patterns of birds or fish: the Genetic Algorithm (GA) (Holland 1992a) and Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995).

Metaheuristics have been increasingly growing over the last few decades in industry and academia (Whitacre 2011a,b). Dedicated journals to metaheuristics include the *IEEE Transactions on Evolutionary Computation*, *Evolutionary Computation*, *Journal of Heuristics*, *Handbook of Evolutionary Computation* and the *International Journal of Metaheuristics*, as well as conferences like the IEEE Congress on Evolutionary Computation and the Genetic and Evolutionary Computation Conference. Despite their versatility and proven success in various fields like engineering and finance, the application of metaheuristics in medicine and public health remains underutilized (Ezugwu et al. 2021). The potential of metaheuristics to optimize healthcare presents a compelling case for their increased integration into medical and health-related research (Ghaheri et al. 2015, Katoch et al. 2021). Throughout this

dissertation, we will combine novel use of metaheuristic algorithms to best answer pressing needs in clinical trials.

1.4 Optimal designs to help patients with bipolar disorder

Optimal designs in PK/PD are recommended for new drug applications because they help drive efficacy and safety (Barrow and Lindsley 2023). PK/PD studies have used NLMEMs for decades to understand the variability in drug response across individuals or populations (Sheiner et al. 1977, Nyberg et al. 2015, Comets and Mentré 2021). This requires specialized software with advanced, developing methodologies (Comets and Mentré 2021). In Chapter 3, we develop efficient optimal designs enhancing this software to better understand PK/PD studies using an application of sustained-release lithium for patients with bipolar disorder. Patients with bipolar disorder are at a 15-fold increase for completed suicide (American Psychiatric Association et al. 2013). Lithium has been a successful mood stabilizer for decades, and is often attributed to a reduction in suicide (Schulze et al. 2010). Oftentimes, there are "excellent lithium responders" who attribute their lives saved to lithium administration, and there may be a genetic link (Grof 2010). Unfortunately, even though lithium is extremely effective, it has a narrow safe therapeutic window and differences in bioavailability can be life-threatening (McKnight et al. 2012). With the future of precision medicine on the horizon, we can all benefit from efficient designs.

We provide guidance on the optimal sampling times to maximize the information gained from blood samples with multiple physician-specified constraints that include (a) a pre-specified number of sampling time points, (b) a pre-specified time window for measurement, and (c) a pre-specified number of groups in the study. We propose alternative designs which can enable a more holistic understanding of the exact statistical model. By employing metaheuristic algorithms, which are not yet standard in PK/PD software, this study endeavors to improve our understanding of the PK/PD of sustained-release lithium, particularly in the

presence of potential genetic links in patients with bipolar disorder.

1.5 Optimal designs for global clinical trial recruitment

Global clinical trials are increasingly common and are used to recruit a diverse set of patients to ensure the medications work across a broad population. Decision makers must plan for timely recruitment of thousands of patients and pre-register sites used in trials, and come up with a design to recruit patients in the most cost-efficient way and subject to real world constraints (Senn 1997). Most commonly, the main target in the trial is to recruit a pre-specified number of patients by a certain time, but this target is often not met and results in a cancelled or failed trial (Bogin 2022, Sully et al. 2013, Walters et al. 2017). To prevent this costly failure, pharmaceutical companies often use statistical models to predict and forecast recruitment rates (Barnard et al. 2010, Anisimov 2016, Gkioni et al. 2019). The Poisson-gamma (PG) model has emerged to be particularly effective in modeling recruitment rates (Anisimov and Fedorov 2007a,b, Anisimov 2011, Anisimov and Austin 2023). The PG model can be used to formulate an optimization problem with multiple nonlinear objectives and constraints, necessitating the use of metaheuristics (Anisimov and Austin 2023).

In Chapter 4, we showcase the flexibility and usefulness of metaheuristics to generate efficient large scale recruitment plan for thousands of patients spread over hundreds of sites in potentially dozens of countries. Such design problems are very under-researched and there is a pressing need to properly design for such studies. This dissertation provide some answers to such real large scale design problems. We close this chapter by providing a real example of such a trial using the recent COVID-19 vaccine trials which used multiple countries to recruit a diverse patient population at a rapid speed. Table 1.1 shows the recruitment plan of the Moderna Phase III clinical trial which recruited 30,420 patients in 99 clinical trial sites in the United States, the Pfizer Phase III trial which recruited 43,548 patients in 152 sites in six countries, and the Johnson & Johnson Phase III trial which recruited 44,325 in

217 sites in eight countries (Polack et al. 2020, Baden et al. 2021, Sadoff et al. 2021). It is not clear if they used the PG model with these designs to ensure adequate recruitment, but the information is available publicly at https://clinicaltrials.gov/study/NCT04368728, https://clinicaltrials.gov/study/NCT04470427, and https://clinicaltrials.gov/study/NCT04505722.

Table 1.1. Recruitment plans used in the Phase III clinical trials for the COVID-19 vaccine.

Country	Moderna	Pfizer	Johnson & Johnson
United States	99	130	117
South Africa	-	4	25
Turkey	-	9	-
Germany	-	6	-
Brazil	-	2	27
Argentina	-	1	12
Colombia	-	-	13
Peru	-	-	9
Chile	-	-	6
Mexico	-	-	8
Number of sites	99	152	217
Number of countries	1	6	8
Number of patients	30,420	43,548	44,325
Time for recruitment	Jul - Oct '20	Jul - Nov '20	Sep '20 - Jan '21

CHAPTER 2

Nature-inspired metaheuristic algorithms

Metaheuristics are versatile and robust algorithms capable to solve all classes of optimization problems because they do not need any mathematical assumptions like convex, linear, or gradient programming methods (Yang 2020). The world of finance, operations research, and engineering are making use of this innovative line of thinking, but the application of metaheuristics to medicine and public health presents many applications yet to be explored (Alam 2016, Ezugwu et al. 2021). This chapter describes properties, examples, and applications of metaheuristics and draws from many available reviews and books on metaheuristics further discussed within, for instance see Bianchi et al. (2009), Boussaïd et al. (2013), Gogna and Tayal (2013), Zavala et al. (2014), Xiong et al. (2015), Soler-Dominguez et al. (2017), Abdel-Basset et al. (2018), Almufti (2019), Dokeroglu et al. (2019), Hussain et al. (2019), Maier et al. (2019) or Ezugwu et al. (2021).

2.1 What are metaheuristics?

Traditional methods, like gradient descent or linear programming, often require detailed information about the problem structure to perform the optimization and guarantee convergence (Bonnans et al. 2006, Khan et al. 2015). In contrast, metaheuristics does not rely on assumptions and can be used in any class of optimization problem no matter how complex the objective function or constraints become (Ezugwu et al. 2021). Metaheuristics are approximate algorithms, which do not guarantee convergence, but (a) can be used in situations where traditional methods cannot and (b) can still perform well in traditional problems

(Hussain et al. 2019). Metaheuristic algorithms are sets of instructions used to solve complex optimization problems. A "heuristic" is a systematic strategy to either minimize or maximize a value of a particular problem (Romanycia and Pelletier 1985). "Meta" meaning higher, refers to the search of gaining experience with heuristics over time (Sörensen and Glover 2013).

Metaheuristic algorithms are usually defined by simple mathematical equations built from metaphors seen in nature (Yang 2020). The taxonomy for categorizing metaheuristics draws from biological, physical, and even mythological sources (Molina et al. 2020, Ezugwu et al. 2021). This variety in algorithmic foundations means each algorithm can have distinct search characteristics and frequently produce different answers (Rajwar et al. 2023). The No Free Lunch theorem applies to metaheuristics, and each unique problem and scenario may benefit from a different algorithm (Wolpert and Macready 1997, Yang 2012). While much discussion surrounds the origins of algorithmic inspirations, a pivotal distinction in categorizing metaheuristics lies between single-solution based and population-based algorithms (Talbi 2009). Single-solution based algorithms, which focus on iterative improvement of a single candidate solution, are particularly effective at exploiting locally optimal areas (Molina et al. 2020). In contrast, population-based algorithms evolve a group of candidate solutions through communication of social processes and are adept at exploring vast and rugged landscapes. The diversity of the population can cover more area, hence provide a better chance of escaping local optima (Yang 2020). These population-based algorithms generate an initial population of a pre-specified size and during each iteration, the fitness of each candidate is evaluated, and the algorithm selects promising candidates. Subsequently, the algorithm generates new solutions using these promising candidates, gradually bringing the population closer to the optimum over time.

There are many evolutionary or nature-inspired algorithms, but real-life settings can use only a limited number of algorithms effectively. Who is to know which algorithm will yield a higher quality solution and at what computation speed (Silberholz and Golden 2010)? Fur-

thermore, most metaheuristics are stochastic and can vary from one simulation to another; hence, conducting multiple simulations is recommended (Derrac et al. 2011). The choice of which metaheuristic to use can be problem specific (Osaba et al. 2021). For instance, the planning of the optimal design of a large road system can allow for weeks or months of computation, while an air traffic controller needs an answer immediately. Typically before implementation, a few preliminary simulations of each algorithm are run for the problem at hand and then compared according to outcomes like convergence speed, solution quality, robustness, and computational cost (Derrac et al. 2011). Whenever a new algorithm is proposed, it must show improvements in these metrics compared to others Derrac et al. (2011), Carrasco et al. (2020) and (Osaba et al. 2021). Statistical inference methods can be used like the ANOVA or a non-parametric such as the Friedman's F-test, and the Pairwise Wilcoxon Rank Sum can then be used to determine if there are pairwise differences (Carrasco et al. 2020, Osaba et al. 2021). With each focused problem, we will conduct simulations and employ novel analysis techniques to determine the most informative metaheuristic for the distinct clinical scenarios to be discussed.

Oftentimes these algorithms are analyzed for their performance on benchmark problems with known solutions. This often justifies the use of them, however, real life problems may be entirely different. In the next subsection, we will discuss different application areas with a specific focus on statistics and biostatistics.

2.1.1 Where are they used?

The versatile metaheuristics have been found to be useful across various fields including supply chain optimization (Abualigah et al. 2023), finance (Soler-Dominguez et al. 2017, De Almeida-Filho et al. 2021), and engineering (Zavala et al. 2014, Xiong et al. 2015). Metaheuristics are also commonly used in scheduling problems like the vehicle routing or traveling salesman problems routinely used by services like Uber and Amazon delivery drivers (El-Sherbeny 2010, Osaba et al. 2020). Scheduling problems can also be used in applications

such as batch processing in cloud computing (Singh et al. 2021) and airplane trafficking (Erdem et al. 2021). The algorithms' adaptability and efficiency in handling complex problems make them an invaluable tool in the researcher's arsenal, especially when dealing with real-world problems where complexities render traditional optimization techniques insufficient (Blum et al. 2008a).

Throughout this work, we will be applying metaheuristics to complex models with clinical trial applications. However, metaheuristics have only sparingly been used in statistics and biostatistics (Ali and Hassanien 2015, Ghaheri et al. 2015, Kim et al. 2021). This is surprising given metaheuristics ability to solve for noisy and large, complex landscapes (Kononova et al. 2008, Blum et al. 2012). Early optimization methods centered around least squares or maximum likelihood, but the emergence of intricate, nonstandard models, constraints and objective functions has necessitated the use of metaheuristics in some biological settings (Behnamian and Ghomi 2010, Sun et al. 2011, Rahman 2013). For instance, metaheuristics have been used to escape local optima seen during parameter estimation of complicated genetic networks and metabolite models (Gilman and Ross 1995, Tominaga et al. 2000, Higashi and Iba 2003). Metaheuristics have proven useful in feature selection for health prediction (Xue et al. 2015, Vivekanandan and Iyengar 2017, Liu and Wang 2019, Agrawal et al. 2021, Dokeroglu et al. 2022) and cluster analysis (José-García and Gómez-Flores 2016). They have been used in various clinical settings without widespread usage (Ghaheri et al. 2015).

Metaheuristics and machine learning can benefit from each other (Talbi 2021, Akay et al. 2022). Neural networks have historically relied on back propagation with stochastic gradient descent for optimization, but metaheuristic algorithms have recently been proven to improve the accuracy of neural networks and are undergoing implementation in popular software systems such as TensorFlow (Muñoz-Ordóñez et al. 2018, Santoso et al. 2023). Further, metaheuristics can determine the proper framework or hyperparameters of a neural network, e.g. how many layers are optimal (Mohakud and Dash 2022).

2.1.2 How are they used?

An algorithm's performance can largely be based on its implementation like the language it is programmed in and potential parallelization capabilities (Gmys et al. 2020). Some metaheuristic algorithms are so easy to use and understand, that short YouTube videos or large language models like ChatGPT can create functional understandable code. There are numerous classes and educational websites like Udemy, Coursera, and Manning Publications covering metaheuristics. Most commonly, metaheuristics are implemented and maintained in open-source packages in programming languages (Tian et al. 2017, Van Thieu and Mirjalili 2023). One can imagine the YouTube coding metaheuristic may be outperformed by a continuously maintained package in a high-performing language (Gmys et al. 2020). There are a variety of different packages for the same baseline metaheuristic, each coded differently. For instance, in R, there are multiple packages of GA which come with different capabilities. The `GA` package (Scrucca 2013) can handle discrete and continuous functions with an option for parallelization, `gafit` (Tendys 2016) is for one dimensional problems only, `rgenoud` (Mebane Jr. and Sekhon 2011) can use derivative information if given, and `metaheuristicOpt` (Septem Riza et al. 2019) has a GA as well as a variety of other different algorithms. Many can handle multiple objectives like `mco` `ecr`. Similarly, Python, Java, Julia, and MATLAB have a variety of libraries for metaheuristics listed in Table 2.1. The user must decide which one to implement. These population based metaheuristics Researchers have developed metaheuristic toolboxes and frameworks which allow for comparison of algorithms on benchmark problems (Egea et al. 2014, Hansen et al. 2021). Metaheuristic algorithms are flexible, but still need careful planning in how you use them. In this work, we will discuss strategies to best use these packages.

2.2 Examples of metaheuristics

In this section, we will describe GA and PSO and a few other metaheuristics that we use throughout this work: the Grey Wolf Optimizer (GWO) (Mirjalili et al. 2014), Harmony Search (HS) (Geem et al. 2001b), and Moth-Flame Optimizer (MFO) (Mirjalili 2015b). Each of these algorithms has more detailed descriptions including pseudocodes readily available elsewhere (Holland 1992b, Kennedy and Eberhart 1995, Geem et al. 2001a, Mirjalili et al. 2014, Mirjalili 2015c). Other nature-inspired population-based algorithms used briefly in this work are the Ant Lion Optimizer (ALO) based on doodlebugs eating ants (Mirjalili 2015a), Bat Algorithm (BA) inspired by echolocation and varying pulse rates (Yang 2010), Clonal Selection Algorithm (CSA) imitating an immune system (De Castro and Von Zuben 2000), Differential Evolution (DE) (Das and Suganthan 2010), Dragonfly Algorithm based on how they avoid predators (Mirjalili 2016a), Shuffled Frog-Leaping Algorithm (SFL) (Eusuff and Lansey 2003, Eusuff et al. 2006), sine cosine algorithm (Mirjalili 2016b, Abualigah and Diabat 2021), and the whale optimization algorithm based on humpback whales (WOA) (Mirjalili and Lewis 2016, Nadimi-Shahraki et al. 2023). These are by no means the only metaheuristic algorithms available. Others worth noting are the Ant Colony Optimization (ACO) (Dorigo et al. 2006) and Artificial Bee Colony (ABC) (Karaboga 2010). A few locations provide lists of metaheuristics like Fister Jr et al. (2013), Ma et al. (2023) and Wikipedia.

Here, we describe in detail how popular metaheuristics work.

2.2.1 The Genetic Algorithm (GA)

The GA is one of if not the most popular metaheuristics which strategically searches for a solution based on survival of the fittest principles (Katoch et al. 2021). Each parameter in the model is considered a gene, and the entire set of parameters is encoded into a string called a chromosome (Holland 1992b). Each chromosome is evaluated for its objective value, and iteratively, the best chromosomes survive and reproduce until the stopping criteria

is reached. The iterative strategic search process creates offspring chromosomes from the best parent chromosomes through processes known as selection, crossover, and mutation (Goldberg 1989).

Selection, or natural selection, is the choice of parent chromosomes to "mate" to generate the next population (Holland 1992b). This could be a deterministic strategy where the top individuals are selected to mate, or a probabilistic "roulette wheel" strategy places emphasis on those with higher fitness values while still introducing randomness to the search (Zhong et al. 2005). A balanced selection strategy between deterministic and probabilistic is the tournament selection strategy which picks the best out of a randomly selected handful of chromosomes to move onto the next generation (Miller et al. 1995).

Crossover represents a fundamental genetic operator in genetic algorithms, akin to the biological process of recombination, where segments of genetic material are exchanged between paired chromosomes (Goldberg 1989). The strategy and extent of crossover are variable and can be tailored to the problem at hand (Umbarkar and Sheth 2015). In certain scenarios, only a single gene may be interchanged between two chromosomes, while in others, a more complex shuffling may occur. Notably, uniform crossover is a method where genes are exchanged between parent chromosomes at each position with a fixed probability, resulting in offspring that is a mix of both parents' traits (Syswerda et al. 1989). This process can be visualized in Figure 2.1, which depicts the exchange of three genes between two parent chromosomes, leading to the creation of new chromosomes for the subsequent generation. Post-crossover, the fitness of the new generation is assessed and compared to that of its predecessors, ensuring the propagation of advantageous traits and the continuous evolution of the solution.

Mutation is a genetic operator used to maintain genetic diversity within a population of chromosomes in genetic algorithms (Lambora et al. 2019). It represents the spontaneous and random changes to gene values, occurring independently of the crossover process. By altering the genetic information of chromosomes at random, mutation introduces new traits into the

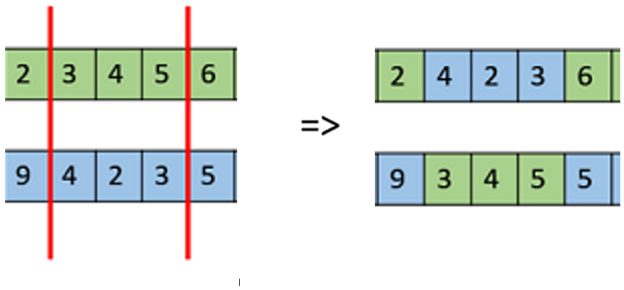


Figure 2.1. An example of genetic algorithm's uniform crossover.

population, some of which may have never been present in the parent chromosomes. This increase in variability is crucial for exploring previously uncharted regions of the search space and prevents the population from becoming too homogeneous, which can lead to premature convergence on suboptimal solutions.

Now, we have described the fundamental principles of how the GA works. Here, we use the terms described above to present a brief description for the generic GA.

1. Encode an initial population.
2. Evaluate each member of the population.
3. Select the chromosomes to mate.
4. Crossover between mates.
5. Mutate some of the offspring genes.
6. Repeat steps 2, 3, 4, and 5 until convergence.

In what is to follow, we present brief descriptions of other algorithms used throughout this work.

2.2.2 The Particle Swarm Optimization (PSO)

The PSO algorithm is inspired by the social behavior of birds flocking or fish schooling (Kennedy and Eberhart 1995). It optimizes a problem by iteratively trying to improve a

candidate solution with regard to a given measure of quality. The algorithm shares information among individual solutions, referred to as particles, to guide their search in the solution space. The velocity is the speed or search length each particle uses to justify its position in the search space and changes based on the search. Each particle has tuning parameters to adjust based on its own experience, c_1, and the experience of neighboring particles, c_2.

1. Initialize a swarm of N particles with certain or random positions and velocities.
2. Update individual and global best positions based on the velocity, c_1 and c_2.
3. Adjust the velocity and position of each particle, and information about the distance from the particle's best position and the global best position.
4. Continue Steps 2-3 until convergence or a maximum number of iterations is reached.

2.2.3 The Grey Wolf Optimizer (GWO)

The GWO algorithm is based on the α, β, δ, and ω hunting hierarchy within packs of grey wolves (Mirjalili et al. 2014). Wolves search for better prey iteratively from these initial hunting patterns based on the hierarchy. When the wolves search, there are scale parameters for obstacles in nature, C, and whether the wolf is converging or diverging from the hierarchy, A.

1. Initialize a hunting position for each of the N wolves.
2. Navigate a new hunting position based on A, C, and wolf hierarchy parameters.
3. Update the α, β, δ wolves.
4. Continue Steps 2-3 until convergence.

2.2.4 The Harmony Search (HS)

The HS algorithm is inspired by the underlying principles of musicians improvisation of harmony (Geem et al. 2001a). The search for the perfect harmony involves trying various possible combinations of the music pitches stored in memory. The HS algorithm begins with a randomly generated Harmony Memory (HM), representing potential solutions. The algorithm generates new music using parameters like the Harmony Memory Considering Rate (HMCR) and the Pitch Adjusting Rate (PAR).

1. Initialize a HM of N harmonies.
2. Improvise a new harmony from HM using the HMCR and PAR.
3. If the new harmony is better than worst harmony in HM, include the new harmony in HM.
4. Continue Steps 2-3 until convergence.

2.2.5 The Moth Flame Optimizer (MFO)

The MFO algorithm draws inspiration from the nocturnal behavior of moths. In nature, moths are naturally attracted to the moon and get distracted by closer light sources, and this behavior forms the basis of the MFO algorithm (Mirjalili 2015c). It operates by simulating the interaction between moths and artificial flames, where the flame represents the optimal solution in the search space. Moths are attracted to the flame in a logarithmic spiral (scale parameter b), and as they attempt to minimize their distance from the flames, they adjust their positions exploring with parameter t and converging based on parameter r.

1. Initialize N moths and select the best moth to become the flame.
2. Move based on a logarithmic spiral pattern with parameters: b, r, and t.

3. Update the best position to become the new flame in the next iteration.

4. Continue Steps 2-3 until convergence.

2.3 Advanced topics in metaheuristics

In this section, we will mention enhancements to baseline metaheuristics. Many variants of each algorithm exist, each attempting to improve some aspect of the baseline metaheuristic. For instance, there are multiple versions of PSO with different search characteristics (Imran et al. 2013, Kumar et al. 2016, Jain et al. 2022). As research progresses, the focus will most likely shift from the creation of new algorithms to best using the already existing algorithms (Kavita and Shinde 2023). Future research other than application is on topics like constraint handling techniques, multiple objective optimization, and hybrid tuned, automatically constructing metaheuristics (Blum et al. 2008b, Singh et al. 2017, Coello 2022).

2.3.1 Constraint handling techniques

In the realm of metaheuristic optimization, handling constraints is a critical challenge to properly adjust for the problem (Smith et al. 1997a, Coello 2022). The penalty functions are the most common methods of constraint handling techniques and repair functions which attempt to salvage infeasible solutions are the second most popular (Rahimi et al. 2023, Lagaros et al. 2023). The various types or penalties including death, static, adaptive, and dynamic are popular because of their ease and effectiveness in managing constraint violations (Smith et al. 1997b).

The death penalty method, known for its simplicity, imposes a severely high cost on any solution that violates the constraints, effectively eliminating it from the pool of feasible solutions. This approach is straightforward to implement but can be overly rigid, as it does not distinguish between degrees of constraint violation. Consequently, it may prematurely

discard potentially valuable solutions that only marginally violate constraints, potentially leading to suboptimal exploration of the solution space because it loses previous information (Kulkarni et al. 2021). A less drastic penalty, the static method assigns a fixed penalty to constraint violations, similar to the death penalty, but not as extreme (Smith et al. 1997a). Unlike the death penalty, the static penalty allows solutions that violate constraints to remain in the pool of potential solutions, albeit with a reduced fitness score. This approach provides a bit more flexibility compared to the death penalty. It enables the exploration of solutions that are near the boundary of feasibility, potentially leading to better overall search performance in the optimization process.

In contrast, the adaptive penalty method offers a more nuanced approach and adjusts the severity of the penalty based on the extent of the constraint violation (Tessema and Yen 2009). By doing so, the adaptive penalty allows for a gradual steering of the search process towards feasible regions of the solution space, without harshly excluding solutions that are near the boundary of feasibility (Rahimi et al. 2023). The dynamic penalty starts with a lower penalty and as the optimization progresses, the penalty increases over iteration, thereby shifting from exploration to exploitation (Smith et al. 1997a). This can be useful in emphasizing the search at different stages of the search process.

Throughout this dissertation, we will be using different methods of penalization. The choice between which method to use depends on the specific characteristics of the optimization problem at hand.

2.3.2 Multiple objective optimization

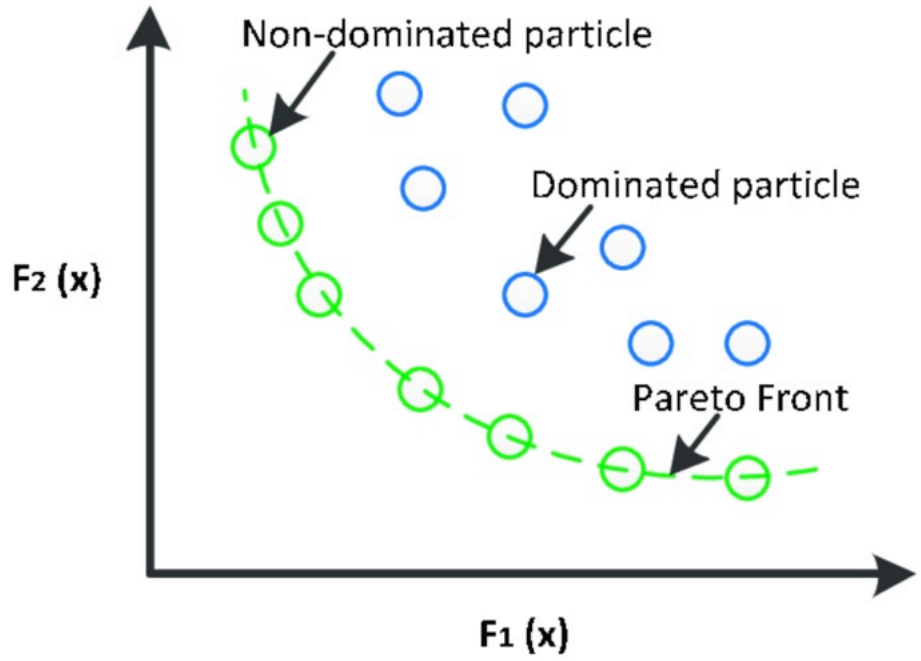
Real life problems are seldom one-dimensional; rather, they are multifaceted, with different goals that need to be achieved simultaneously. Multiple objectives are often conflicting; an improvement in one usually causes a decrease in the other. For instance, a pharmaceutical company could have two simultaneous objectives: minimizing the cost of trial design and maximizing the number of countries to promote diversity. In such problems, optimization

with respect to one objective usually reduces optimality with respect to other objectives. These objectives conflict because the minimal costing solution would probably not involve using every country possible, and, vice versa, the design with the fewest countries may be more expensive.

In multiple objective optimization, compromises are found and are characterized into optimal and suboptimal trade-offs, otherwise known as non-dominated and dominated solutions. A fundamental principle of multiple objective optimization is that there is not a single optimal solution but rather a set of solutions exists, each with its own trade-offs (Edgeworth 1881). The Pareto optimal solutions are the set of non-dominating solutions, when no solution is better in all objectives simultaneously (Pareto 1896). To envision the definitions of dominance and non-dominance, see Figure 2.2 below for two objectives. The green dots represent the Pareto optimal solutions and form the Pareto front. The blue dots represent the dominated solution set which are not optimal in at least one objective and are inferior trade-offs. At its core, non-dominating solutions help us understand how one solution compares to another. By leveraging the Pareto front, we offer a robust choice of solutions that allows for a nuanced balance between objectives. Multiple objective metaheuristics are the most popular way to construct a Pareto front with many different algorithms available (Coello 2006). There is a curated repository of references and software for multiple objective metaheuristics found at https://delta.cs.cinvestav.mx/ ccoello/EMOO/. In the rest of this subsection, we describe how to quantify Pareto fronts and then introduce some of the most commonly used methods to find the Pareto optimal solutions including some multiple objective metaheuristic algorithms.

Visualizing the Pareto front is a helpful way of presenting the solutions to the decision maker. The two- or three-dimensional Pareto fronts are straightforward to visualize, however it gets more challenging with increasing dimension (Coello Coello et al. 2020). A Pareto front need not be continuous and may have gaps which can represent high trade-off solutions (Braun et al. 2015). There have been attempts to visualize the desired aspects of high-

Figure 2.2. An example of a Pareto front and dominated and non-dominated solutions.



dimensional Pareto fronts like boundary points, isolated points, and points with large-trade-offs (Talukder and Deb 2020). Other visualization attempts map one objective at a time (Blasco et al. 2008). There have also been attempts to approximate the high-dimensional front in lower dimensions (Qian and Yu 2017).

The hypervolume is an effective way to measure the value of a Pareto front and is one of if not the most used metric to quantify Pareto fronts (Zitzler and Thiele 1998). The hypervolume metric offers a comprehensive assessment of a Pareto front's performance by measuring both its diversity and quality. This is essential in distinguishing the most dominant Pareto front among multiple options, as a Pareto front that dominates another will exhibit a strictly greater hypervolume (Zitzler et al. 2007). In two-dimensions, the hypervolume is calculated as the sum of different rectangles created from each point on the Pareto front to a reference point, typically, the worst case scenario (Zitzler and Thiele 1998). For more than two-dimensions this value is the sum of the volumes of the hypercubes. Each

individual point on the Pareto front contributes to the overall hypervolume, which takes into account overlapping points (Auger et al. 2012).

In addition to the hypervolume metric, several other multiple objective measurements are widely used in evaluating the quality of Pareto fronts. This list includes the Haussdorff distance, related to the generational distance and inverted generational distance performance metrics, and which measure the average distance from points on the true Pareto front to the nearest points on the approximated Pareto front (Schutze et al. 2012, Bezerra et al. 2017). These metrics are often used when there is knowledge of true non-dominated solutions, which is often only known in benchmark problems and not real life problems or a fake Pareto front can be used for comparison (Bossek 2018). The crowding distance measures diversity among the solutions on the Pareto front, which is essential for a comprehensive exploration of the trade-off coverage in multiple objective optimization problems (Raquel and Naval Jr 2005).

Now that we understand how to compute the value or goodness of a Pareto front, we must discuss the various ways to create Pareto fronts. Multiple objective population based metaheuristics are the most common way to search for the set of solutions found on Pareto fronts due to their population based nature yielding multiple solutions in one simulation (Coello 2007). They benefit over mathematical programming techniques which have difficulty finding any type of Pareto front in any complexity or irregularity of problem like non-convex fronts or discontinuous problems (Miettinen 1999, Coello 2006). There has been much work around multiple objective metaheuristics centered around the core concepts of the GA and how multiple objective evolutionary algorithms approach the Pareto optimal solutions (Konak et al. 2006, Coello Coello et al. 2020). Some base their search process on finding optimal ranks of solutions, and others are indicator-based meaning they are guided by improving metrics such as hypervolume (Deb et al. 2002, Jiang et al. 2014, Falcón-Cardona and Coello 2020). While much emphasis has been around the evolutionary algorithms based on a GA for multiple objective optimization, some metaheuristics have a multiple objective parallel like PSO (Moore and Chapman 2003, Nebro et al. 2009, Valencia-Rodríguez and

Coello 2023).

The naive, yet most traditional way to solve multiple objective optimization algorithms is to weigh the objectives and optimize the weighted solution as a single-objective optimization problem (Marler and Arora 2010). The weights on each objective signify the importance of the objective and are difficult to interpret and properly choose. Each objective must be normalized for the weighting to be non-biased. However, this method has limitations, such as difficulty in identifying solutions in non-convex regions, thus providing an incomplete picture of the Pareto optimal set (Gkiotsalitis 2023). The epsilon constraint method similarly converts multiple objective problems into a series of single-objective optimization problems (Laumanns et al. 2005). One objective function is chosen as the primary objective and is optimized with a single objective optimizer, while the remaining objective functions are converted into constraints with a specified tolerance level denoted by epsilon (Yang et al. 2014). We will see a potential drawback to this method is the necessitated choice of epsilon and need for multiple runs, which can significantly affect the efficiency and diversity of the obtained Pareto-optimal solutions (Laumanns et al. 2005).

The original Non-dominated Sorting Genetic Algorithm (NSGA) was one of the earliest attempts to tackle multiple objective optimization problems using GAs where solutions were grouped based on Pareto dominance levels (Srinivas and Deb 1994). NSGA-II improved upon the computational efficiency of this approach and has quickly become an effective and popular algorithm in multiple objective optimization (Verma et al. 2021). NSGA-III then was modified slightly to best handle problems with more than three objectives (Deb and Jain 2013, Jain and Deb 2013). Generally, the algorithms first generate an initial random population. They then categorize the population into different layers of Pareto fronts, where the true Pareto optimal solutions are given a rank of 1, then the next best Pareto front is given a rank of 2, etc., with each individual assigned a crowding distance to gauge its proximity to others, thus maintaining diversity. Selection for the next generation is performed through a binary tournament based on non-domination and crowding distance. Offspring are produced

via genetic operators of crossover and mutation, and subsequently, the parent and offspring groups are merged. The best candidates from this combined group are then chosen to form a new, stable-sized population. This process iterates until a predefined termination condition as discussed prior. Here, we present a brief description for the NSGA-II algorithm.

1. Encode an initial population.
2. Evaluate each member of the population.
3. Assign rank based on the Pareto front ordering
4. Select the chromosomes to mate.
5. Crossover between mates.
6. Mutate some of the offspring genes.
7. Assign ranks based on the Pareto front ordering
8. Determine crowding distance
9. Select the best individuals based on crowding and ranking distance
10. Repeat steps 2 through 9 until convergence.

2.3.3 Tuning metaheuristics

Metaheuristics have various search strategies parameters, such as population size, iteration count, and algorithm-specific settings, which can be adjusted and significantly impact the effectiveness and efficiency of the algorithm (Birattari and Kacprzyk 2009a, Huang et al. 2019). The ideal settings for these parameters are unknown prior to optimization, with defaults usually set according to the best performance for some benchmark functions. Proper tuning and usage ensures that the algorithm operates at its optimal capacity, tailored to the specific nuances of the problem at hand (Birattari and Kacprzyk 2009a). This section delves into various tuning methods, highlighting their strengths and limitations, providing a comprehensive guide to effectively using metaheuristics.

Parameters in metaheuristics could be categorical or numerical. For instance, the muta-

tion and crossover rates in the GA can range from 0 to 1, the iteration length can matter, while the choice of selection strategy can be categorical like the roulette-wheel and tournament strategies (Holland 1992c). Each of these choices can affect performance. A smaller population size requires less computation than a larger one but then may also result in a poorer solution. Tuning is therefore itself an optimization problem, in which the objective function measures an algorithm's performance across potentially conflicting criteria, such as solution accuracy, computation time, and resource consumption (Birattari and Kacprzyk 2009a). Several methods have been developed for tuning metaheuristics, each with unique strengths and limitations.

Further, tuning of algorithm parameters is not unique to metaheuristics. Neural networks, random forests, and support vector machines also benefit from proper selection of tuning parameters and hyperparameters (Becherer et al. 2019, Mantovani et al. 2015, Probst et al. 2019). Metaheuristics themselves have been used to tune other metaheuristics and other algorithms (Calvez and Hutzler 2005, Grefenstette 1986, Wang 1997). This technique is known as "meta-optimization" (Huang et al. 2019, Mercer and Sampson 1978). Application of meta-optimization has included the use of GAs to tune other GAs (Calvez and Hutzler 2005, Grefenstette 1986, Wang 1997). Many metaheuristics have tuned convolutional neural network hyperparameters like number of layers and nodes (Mohakud and Dash 2021, Dobslaw 2010, Nematzadeh et al. 2022). In the rest of this subsection, we discuss systematic tuning methods applicable to configuring metaheuristics and other algorithms.

Manual tuning can be performed to adjust the algorithm parameters based on experience or trial and error. This allows for immediate adjustments, but can be time consuming, subject to user error and is unscalable. A systematic approach rather, automated tuning, is preferred for many reasons including its reproducibility, consistency, and efficiency. Grid search is the brute-force method that involves testing every possible combination of parameter values to identify the most effective setup. Although this will lead to the optimal tuning parameters, it is often computationally intractable due to the high dimension of the parameter space. A less

computationally intensive procedure is pure random search of parameter combinations, where a random subset of the grid is selected for comparison, and while this is less computationally intense, this is still not an optimal tuning strategy (Probst et al. 2019).

The racing algorithm represents the de factor method to tuning metaheuristics, where different parameter configurations are pitted against each other in a competitive framework (López-Ibáñez et al. 2016). The racing algorithm was first described in 2002 (Birattari et al. 2002) and then extensively studied in a textbook in 2009 (Birattari and Kacprzyk 2009b). The most recently developed racing algorithm ,"irace", stands for iterated racing". The 'irace' R package was developed in 2016, comes with a user-guide, and includes parallelization capability to increase speed (López-Ibáñez et al. 2016, López-Ibáñez et al. 2016). In the racing algorithm, tuning parameter combinations compete against one another (López-Ibáñez et al. 2016). This method aims to efficiently identify the most effective parameter settings for a given optimization problem. Rather than testing a few select parameter settings, the irace procedure globally searches for the best parameters. The process starts with a large set of randomly generated candidate configurations defined within bounds specified by the user for each parameter. The core idea is to iteratively evaluate and eliminate less promising configurations, thus concentrating computational resources on exploring the most promising areas of the parameter space. Only the configurations that perform well enough compared to the others are "survivors" and continue to be evaluated on more instances. The means and standard deviations of the objective function corresponding to all tuning combinations are recorded. After a number of steps, the means and standard deviations are compared using either a t-test or Friedman's non-parametric two-way ANOVA. Those significantly worse than any one configuration are removed from the set under consideration. In the racing algorithm, tuning parameter combinations compete against one another. Those significantly worse than any one configuration are removed from the set. The race continues with the remaining configurations until either a set time expires, a set number of iterations occur, or no other comparisons are possible. To avoid premature convergence, the package

implements a restart strategy to validate previous races. This elitist racing procedure can handle both numeric and categorical parameters. The racing procedure continues until a termination criterion is met, which could be a fixed number of iterations, a time budget, or a convergence criterion.

Effective tuning of metaheuristics is both an art and a science, requiring a deep understanding of the algorithm, the problem, and the interaction between various parameters. By carefully selecting and applying the appropriate tuning methods and considering practical constraints, one can significantly enhance the performance of metaheuristic algorithms, making them powerful tools for solving a wide range of complex optimization problems. Hybrid metaheuristics combine metaheuristics like a GA within a PSO or one following the other (Talbi 2002, Blum et al. 2008b, Talbi et al. 2013). They can be also used in combination with exact algorithms known as matheuristics (Puchinger and Raidl 2005, Fischetti et al. 2018). Future research involves the automatic design of a well-performing algorithm (Bezerra et al. 2015, 2020). This involves the use of irace to test different strategies including constraint handling techniques, population initiation, and how to best combine particular parts of metaheuristics (Zhang et al. 2022).

Table 2.1. A list of metaheuristic packages and their coding languages.

Package Name	Language
jMetal	Java
MOEA Framework	Java
Metaheursitics.jl	Julia
Evolutionary.jl	Julia
GeneticAlgorithms.jl	Julia
ABCoptim	R
DEoptim	R
metaheuristicOpt	R
ppso	R
hydroPSO	R
rgenoud	R
GA	R
gafit	R
microbats	R
EmiR	R
ecr	R
mco	R
Global Optimization Toolbox	MATLAB
PlatEMO	MATLAB
Pyswarms	Python
Opytimizer	Python
Hive	Python
NeverGrad	Python
Pyomo	Python
ParadisEO	C++
EO	C++

CHAPTER 3

Project 1: Optimal sampling times for sustained-release lithium used by patients with bipolar disorder

In this chapter, we will present various types of efficient designs using metaheuristics and a recently proposed PK/PD NLMEM for sustained-release lithium, one of the most popular drugs prescribed for bipolar disorder (Couffignal et al. 2019). Lithium levels are measured routinely to make sure the levels stay within a narrow therapeutic window and hence prevent toxicity (McKnight et al. 2012). This work is part of a broader research grant funded by the French Ministry of Health as part of an ancillary study focusing on the genetics of bipolar disorder and lithium administration.

3.1 Background

Lithium is a successful mood stabilizer, but toxicity must be monitored often because it is easy to overdose due to the narrow therapeutic window (Sheikh et al. 2022). Too high a concentration could cause a range of side effects, including gastrointestinal issues, kidney disorders, hyperthyroidism, and it was once thought to be teratogenic causing a rare birth abnormality named Ebstein's anomaly if taken during the first trimester (Yacobi and Ornoy 2008, Gitlin 2016). In the age of precision medicine, identifying good or poor responders to medication can save lives. Lithium toxicity was unknown during the late 19th and early 20th centuries. Lithium-rich mineral waters were popular as health tonics in Europe and the

United States (Shorter 2009). Lithium was used in the production of the soft drink 7Up, marketed as a mood-enhancing ingredient, similar to cocaine in Coca-Cola (Brown 2019). However, this practice was discontinued in the 1950s due to changing regulations and safety concerns (Marmol 2008). There is also call for increased scrutiny on the direct relation of suicide reduction and lithium because some governments have proposed to add lithium to the drinking water (Memon et al. 2020, Miller and Black 2020).

NLMEMs are a widely used tool to help make informed decisions in the drug development process (Comets and Mentré 2021). The ability to model fixed and random effects enables measurement of the variability in drug response across individuals or populations. Specialized software is often used to handle the complicated optimization of parameters and designs in PK/PD NLMEM studies and could benefit from using metaheuristics (Duffull et al. 2002, Nyberg et al. 2015, Gadkar et al. 2016). In what is to follow, we construct designs to estimate, as accurately as possible, all or some parameters of a recently developed PK/PD NLMEM for sustained-release lithium developed from patients with bipolar disorder who were good responders to the drug for at least 2 years using metaheuristics (Couffignal et al. 2019).

3.2 Optimal designs in PK/PD

Fisher Information Matrix (FIM) based designs measure the worth of the design and designs that optimize the FIM provide the most accurate statistical inference for a given cost (Atkinson et al. 2007). The FIM is the negative of the expectation of the second derivatives of the total log likelihood function with respect to the vector of parameters (Fedorov and Leonov 2014). Its inverse provides a lower bound of the variance-covariance matrix of any unbiased estimated parameters, according to the Cramer-Rao inequality (Rao 1992). This theoretical background underscores the FIM's pivotal role in assessing the reliability and accuracy of parameter estimates (Pukelsheim 2006).

A mapping of the FIM to a scalar is compared to evaluate the worth of different designs

(Nyberg et al. 2012). The most popular scalar method is known as the D-optimal design, which focuses on maximizing the determinant of the FIM, which is a measure of the overall information and volume of the parameter space, ensuring that the design is efficient in terms of the information it extracts about all the parameters collectively (Atkinson et al. 2007). However, some parameters are of more interest than the others. D_s-optimal designs are used when some parameters of the model are of more interest than the others, where the subscript s stands for subset. These two stand out among other criteria, such as A-optimality which focuses on minimizing the average variance of the estimates or E-optimality to maximize the minimum eigenvalue of the FIM (Atkinson et al. 2007).

If the goal is to estimate all model parameters as accurately as possible and Ψ is the vector of nominal values for the parameters, we seek a locally D-optimal design, Ξ_D, that satisfies:

$$\Xi_D = \underset{\Xi \in \Xi^*}{\operatorname{argmax}} \{(|M(\Psi, \Xi)|)^{1/p}\}, \tag{3.1}$$

where Ξ^* is the set of all feasible population designs on the given design space, $|M|$ is the determinant of the FIM M and $p = dim(\Psi)$ is the number of parameters in the model. The normalized FIM determinant is the D-criterion value, and the larger this value is, the more efficient and precise is the design for making inference on the model parameters. We optimize for the criterion and the resulting optimal designs are termed locally optimal designs (Chernoff 1972).

To compare two designs Ξ_1 and Ξ_2, we use their relative D-efficiency ratio:

$$\text{D-efficiency} = \left\{ \frac{|M(\Psi, \Xi_1)|}{|M(\Psi, \Xi_2)|} \right\}. \tag{3.2}$$

As an example, if the ratio is 0.5, Ξ_1 needs to be replicated twice to do as well as the design Ξ_2 for estimating the model parameters. If Ξ_2 is a D-optimal design, the above ratio is the D-efficiency of Ξ_1.

The method for finding the D_s-optimal subset designs parallels that for D-optimality, except we minimize the generalized variance of the estimated parameters of interest only.

Design monographs such as Fedorov (1972) and Silvey (1980) provide details and illustrations on how to partition the full covariance matrix appropriately before maximizing the determinant of the properly partitioned submatrix M_k corresponding to the k parameters of interest. The locally D_s-optimal design focusing on k parameters satisfies:

$$\Xi_{Ds} = \underset{\Xi \in \Xi^*}{\operatorname{argmax}} \{(|M_k(\Psi, \Xi)|)^{1/k}\}, \tag{3.3}$$

D_s-efficiency follows similarly to that of D-efficiency.

The D_s-optimal design will presumably favor the selected parameters in exchange for less information on the other parameters. We propose multiple objective optimal designs to quantify efficiency trade-offs between potentially conflicting objectives. This will enable us to estimate the more interesting parameters with higher efficiency while doing as best as possible to estimate the remaining, less important parameters in the model. We propose to do this by separating the preferred and non-preferred parameters into two separate D_s-optimal designs and attempt to simultaneously maximize both objective functions. The optimal multiple objective design, $\Xi_{D_{multi}}$, satisfies the following equation:

$$\begin{aligned} \Xi_{D_{multi}} &= \underset{\Xi \in \Xi^*}{\operatorname{argmax}} \{f_1(\Psi, \Xi), f_2(\Psi, \Xi)\} \\ f_1 &= \{(|M_k(\Psi, \Xi)|)^{1/k}\} \\ f_2 &= \{(|M_{p-k}(\Psi, \Xi)|)^{1/p-k}\} \end{aligned} \tag{3.4}$$

where M_k and M_{p-k} represent the properly partitioned submatrices for the k parameters of interest and remaining $p-k$ parameters respectively. Then, weighted normalization scores can be used to determine the optimal efficiencies. Such designs have been found in Cook and Wong (1994) and Zhu and Wong (2001), but their method is inapplicable to our design problem because we will have multiple real-world physician placed constraints which make our problem a non-convex optimization problem. We resort to constructing a Pareto front and ultimately choose a solution based on evaluating efficiency trade-offs.

3.2.1 A nonlinear mixed effect model for lithium

This subsection describes the recently proposed PK/PD NLMEM for a once-a-day administration of sustained-release lithium (Couffignal et al. 2019). The two-compartmental model (erythrocyte and serum) seen in Equation (3.5) has 5 parameters represented in vector θ: $\theta^T = (k_a, V_S, CL, CL_{SE}, CL_{ES})$. The absorption process is modeled using a first-order rate constant (k_a), and the other parameters are V_S, the distribution volume in the serum, CL, the total elimination clearance from the serum, CL_{SE}, the clearance from the serum to the erythrocyte, and CL_{ES}, the clearance from the erythrocyte to the serum. After a dose d of lithium is administered to bipolar patients, the mean response f that describes the steady state pharmacokinetics in the serum at time t is given by:

$$\begin{aligned} f(\theta, d, t) = d \times \frac{k_a}{V_S}\Big(&\frac{k_{SE} - k_a}{((B+A)/2 - k_a) \times ((B-A)/2 - k_a)} \times \frac{\exp(-k_a \times t)}{1 - \exp(-k_a \times \tau)} \\ &+ \frac{k_{SE} - (B+A)/2}{(k_a - (B+A)/2) \times ((B-A)/2 - (B+A)/2)} \times \frac{\exp(-(B+A)/2 \times t)}{1 - \exp(-(B+A)/2 \times \tau)} \\ &+ \frac{k_{SE} - (B-A)/2}{(k_a - (B-A)/2) \times ((B+A)/2 - (B-A)/2)} \times \frac{\exp(-(B-A)/2 \times t)}{1 - \exp(-(B-A)/2 \times \tau)}\Big), \end{aligned} \tag{3.5}$$

with $k_{SE} = CL_{SE}/V_S$, $k = CL/V_S$, $k_{ES} = CL_{ES}/57.5$, $B = k + k_{SE} + k_{ES}$, and $A = \sqrt{B^2 - 4 \times k \times k_{SE}}$. The pharmacokinetics was assumed to be at steady state and the dosing regimen was 36 mEq (1Eq = 36.8g) per day (interval between two doses τ is 24h).

We assume an additive heteroscedastic error: $\sigma \times f(\theta, d, t) \times \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, 1)$. The between-subject variability of the parameters is modeled using an exponential model for the random effects b_i with $b_i \sim \mathcal{N}(0, \Omega)$, and Ω is the diagonal variance-covariance matrix *i.e.* log-normal distribution of the parameters. Let λ be the vector of variance parameters containing all unique and nonzero elements of Ω and σ. We investigated designs with and without a genetic covariate effect, β_{CL}, in the model. The genetic covariate effect is included as follows: $log(CL_i) = log(\mu_{CL}) + \beta_{CL} + b_{CL_i}$, with μ_{CL} the fixed effect of CL and b_{CL_i} the random effect of CL for the individual i.

Let $\Psi = (\mu^T, \beta^T, \lambda^T)^T$ be the vector of all population parameters to be estimated. Overall, the model has 11 or 10 nonzero parameters depending upon the inclusion of the genetic covariate. Table 3.1 presents the nominal values for the the fixed effects, μ, and the variance components, ω, used in Equation (3.5) obtained earlier from the study by Couffignal et al. (2019) for the parameters k_a, V_S, CL, CL_{SE}, CL_{ES}, β_{CL}, and σ^2. This data was measured from lithium concentrations retrieved for 17 patients who had at least 2 years of successful sustained-release lithium treatment. For the measurements, patients were instructed to switch from an evening dose to a morning dose for the 15 days prior to studying. Adherence to this switch was high, with only two patients missing one dose each. Blood samples were collected at the hours 0, 1, 4, and 8. Further details and justifications for the model and its assumptions are available in Couffignal et al. (2019).

Table 3.1. Parameter settings of the model describing the lithium pharmacokinetics.

Parameter	**Distribution**	μ	ω	σ^2
k_a (h^{-1})	log-normal	0.93	0.72	-
V_S (L)	log-normal	22.3	0.3	-
CL (L/h)	log-normal	1.24	0.2	-
CL_{SE} (L/h)	-	4.15	-	-
CL_{ES} (L/h)	log-normal	11.1	0.27	-
β_{CL}	-	0.32	-	-
σ^2	-	-	-	0.137

In the context of mixed effects models, the likelihood L of the vector of individual observations (y_i) for parameters Ψ is given by

$$L(\Psi, y_i) = \int_{b_i} p(y_i|b_i, \Psi)p(b_i|\Psi)\, db_i, \tag{3.6}$$

where $p(\Psi, y_i|b_i)$ is the probability density function of y_i given random effects b_i, and $p(b_i|\Psi)$ is the probability density function of the random effects. The FIM for a NLMEM has no

closed form, and it has been shown that FIM evaluation for NLMEM by first-order (FO) linearization is adequate, with expected standard errors of parameters close to empirical standard errors obtained by simulations (Bazzoli et al. 2009, Nyberg et al. 2015). This method is used in popular PK/PD software, including PFIM, NONMEM, and PopED (Nyberg et al. 2015, Dumont et al. 2018, Bauer et al. 2021). We therefore use this method to evaluate the FIM in this work and consider it sufficient. Details for the calculation of the FIM by FO, using an additive or an exponential random effects model, are given in the publication and user manual of the R software PFIM (Dumont et al. 2018).

3.3 The proposed designs

Physician given designs and constraints representative of the real world. We search for a D-optimal design with five time points to place for measurement within an 8 hour window. The time window and the specific number of blood draws are practical constraints put in place to minimize patient and staff workload. Specifically, we set to improve from the naive equispaced design of time points at hours 0, 2, 4, 6, and 8. We will compare three, four, five, six, and seven time points, but with three or four time points, the FIM is often noninvertible, and the scenarios with more than five blood draws are unrealistic due to the increased workload and burden put on staff and patients. Further, there are 3 parameters the physicians have particular interest in: the fixed and random effects corresponding to overall clearance, μ_{CL} and $\dot{\_}CL$, and the genetic covariate, β_{CL}. To focus on and best estimate the clearance and genetic parameters, we search for what we will call D_{s3}-optimal designs corresponding to the 3 parameters of interest. We also consider designs when the genetic covariate is not included in the model and ascertain whether there is a difference in designs. In the scenario without the genetic covariate, we search for $D_{s2}-$optimal designs that only focus on estimating μ_{CL} and b_{CL}. We compare all designs to the locally D- and D_s-optimal designs found by the metaheuristic algorithms and evaluate their D- and D_s-efficiencies

respective to the best design found.

Additionally, at the request of physicians, we investigate these D- and D_{s3}- or D_{s2}-optimal designs in two groups and four time points. When we explored the one group design with four time points, the relative standard errors for the estimates were unreliable and even resulted in noninvertible matrices. Here, with potentially different time points in each group, the two group scenario can provide sufficient information. We seek to optimize the optimal sampling times and the number of patients in each group. For example, if there were $N = 100$ patients, and the proportions were split into two groups at 55% and 45%, then the physicians would assign 55 patients to the first group and 45 patients to the second group, and the two groups may have different sampling time points.

We also find designs which maximize the information for the remaining, less important 8 parameters, called $D_{s8}-$optimality. As this is a secondary goal, we will search for multiple objective optimal designs to simultaneously maximize the information from the interesting parameters using D_{s3}-optimality and the less important parameters using D_{s8}-optimality. We will analyze the Pareto front fo optimal efficiency tradeoffs.

3.3.1 R programs

Nonlinear mixed effects models (NLMEMs) are a widely used tool to help make informed decisions in the drug development process (Comets and Mentré 2021). The ability to model fixed and random effects enables measurement of the variability in drug response across individuals or populations. Specialized software is often used to handle the complicated optimization of parameters and designs in PK/PD NLMEM studies (Nyberg et al. 2015). PFIM is an exemplary, continuously maintained open-source software package in R used to evaluate and optimize designs in NLMEMs (`http://www.pfim.biostat.fr`) (Dumont et al. 2018). PFIM currently only computes D-optimal designs to best estimate all parameters in a model using either the Nelder-Mead simplex (Nelder and Mead 1965) or a modified version of the Fedorov-Wynn algorithm (Retout et al. 2007). The developers of PFIM were

interested in having additional or alternative tools to solve more complex design problems under different criteria, and have algorithms more capable of extracting itself out from a local minima (Duffull et al. 2002, Dumont et al. 2018).

To focus on maximizing the information found from specific parameters of interest, we implemented D_s-optimal and multiple objective optimal designs. The algorithms we use to optimize these designs are (a) the PFIM simplex algorithm implemented in R, (b) the `optim_pso()` function in the `ppso` R package and (c) the `ecr()` function in the `ecr` R package. In our implementation, we leave most inputs to these algorithms as the default in the packages because we saw convergent solutions. We set the initial population for each algorithm to begin their search at the equispaced design and to have lower and upper bounds at 0 and 8 respectively. For `optim_pso()`, we set the population size to 40 and number of iterations to 20 for a total of 800 function calls. From our preliminary testing, this was sufficient from both an objective function and a computation time standpoint. The R package `ecr` has a function `stopOnMaxTime()` so the algorithm will stop after a certain number of seconds. This enabled us to manually set the search time for `ecr` to be similar to the other two algorithms which do not have the capability to designate a time limit.

3.4 Optimal sampling times for lithium

Here we present results for the proposed designs using the physician guided designs with a PK/PD model for sustained-release lithium.

3.4.1 Number of sampling points

The number of sampling points was pre-specified by the physicians to be five. Table 3.2 presents D-optimal designs consisting of three, four, five, six, and seven time points computed by the PFIM simplex. Each of these designs has time points at the beginning and end of the time window (hours 0 and 8). The three and four time point designs often resulted in

noninvertible matrices. Looking only at the five, six, and seven time point designs, each had points at the first half hour (0.4 or 0.5), between the second and third hour (2.0 to 2.7), and near hour 5 (5.0 to 5.4). The measure at the first half hour could be to best estimate the absorption process and then the follow-up points measure the descent from the peak concentration. The six time point design has the same time points as the five time point design with a redundancy at hour 8. This duplicated time point can be interpreted such that the information at hour 8 is important and can possibly be worth weighting in future analyses. The five point design, which the physicians constricted us to, has good performance compared to the seven time point design being 85% as efficient. The seven point design has a similar design to the five and six point designs with an additional time point at hour 3.7 again presumably to measure the descent from the peak. This confirms that the physician placed constraints are feasible and in fact somewhat optimal given the duplicated time points and marginal efficiency gain that comes with increased blood draws.

Table 3.2. A comparison of a different number of design points for D-optimal designs and efficiencies.

Time Points	Design	D-efficiency
3	0, 4.9, 8	6%
4	0, 0.5, 2.7, 8	51%
5	0, 0.4, 2.3, 5.1, 8	85%
6	0, 0.4, 2.3, 5.0, 8, 8	93%
7	0, 0.4, 2.0, 3.7, 5.4, 8, 8	100%

3.4.2 Convergence results of PSO

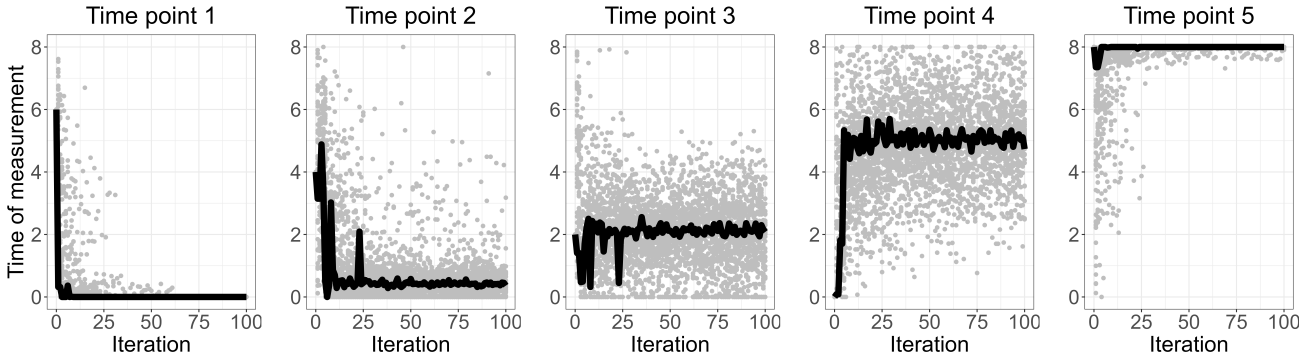
We analyze the convergence trajectory of each design point for the five time points showcasing one long simulation of 100 iterations from the `optim_pso()` function. The function has a built in logging capability to track the search path of each individual in the population (Francke 2020). We use the history of each individual in the population to plot out how each of the five design points converges to their final solution starting from the equispaced design.

Figure 3.1 presents the convergence trajectory of the `optim_pso()` search for each design point after 100 iterations. The gray dots represent each individual particle with 40 particles per iteration. The black line represents the particle which has the highest D-optimal criterion value at each iteration, and thus the final design point at the end of the search. The five design points initially started off at hours 0, 2, 4, 6, and 8 respectively. They finished at time points of 0, 0.5, 2.2, 5.0, and 8. However, where the PSO search started had little to do with where the design points finished. For instance, the time point which originated at hour 6 ended up at time 0, and the original time 0 converged to hour 5.1. The algorithms quickly converged to the best solution sometimes within 10 - 15 iterations as signified by the black line. This justifies our use of 20 iterations to compute the designs found in the Tables 3.3 and 3.4.

3.4.3 D- and D_s-optimal designs

Table 3.3 presents the D- and D_{s3}-optimal time points for the one and two group designs from the simplex, `optim_pso()`, and `ecr()` optimizations. Each row represents an algorithm's optimal design, as well as, the D- and D_{s3}-efficiencies relative to the local optima found and the computation time in seconds. Each algorithm found a more efficient D-optimal design than the equispaced design in under 30 seconds. The naive equispaced design points of 0, 2, 4, 6 and 8 hours had a D-efficiency of 74% when compared with the locally D-optimal

Figure 3.1. Iterative convergence trajectory plots of each individual design point highlighting the optimal solution.



designs found by the simplex and PSO algorithms. The D-optimal sampling time points at hours 0, 0.5, 2.2, 5.0, and 8 may appeal to the physician because they provide the most accurate estimates for all parameters in the model. The D_{s3}-optimal design points give more preference to the clearance and genetic covariate parameters and the time points and efficiencies should be compared with the D-optimal designs. When focused on the clearance and genetic parameters, the $D_{s3}-$optimal five point design was found by PSO and has time points at hours 0, 0.9, 6.8, 8, and 8. Instead, the D_{s3}-optimal designs focus on the later stage of the drug. Again, we see two time points at hour 8, this time at the expense of the 2nd hour measurement. However, these D_{s3}-optimal designs do not perform well under D-optimality. For instance, the D_{s3}-optimal design just mentioned has a D-efficiency of 59%. The D-optimal designs, however, have a high D_{s3}-efficiency at 97% signifying the D-optimal design will estimate the 3 parameters of interest very well.

The designs with two groups have similar if not the same time points as the one group design, e.g. 0, 0.4, 2.2, 8h, but sometimes there is a group whose last time point does not end at the 8h mark. The proportion of patients assigned to each group is displayed next to the design in Table 3.3. Even though we specified to have two groups, there are scenarios when the proportion of patients assigned to each group is so low that it does not make

Table 3.3. D- and D_{s3}-optimal designs and efficiencies for the model with the genetic covariate. PSO stands for the `optim_pso()` function and ECR stands for the `ecr()` function.

Algorithm	Design	Proportion	D-efficiency	D_{s3}-efficiency	CPU (s)
One group five design points D-optimal					
Equispaced	0, 2, 4, 6, 8	-	74%	94%	-
Simplex	0, 0.4, 2.3, 5.1, 8	-	100%	97%	27
PSO	0, 0.5, 2.2, 5.0, 8	-	100%	97%	27
ECR	0, 0.4, 2.4, 5.0, 8	-	99%	97%	30
One group five design points $D_{s3}-$optimal					
PSO	0, 0.1, 6.8, 8, 8	-	59%	100%	123
ECR	0, 1.0, 5.8, 8, 8	-	67%	100%	120
Two group four design points D-optimal					
Simplex	0, 0.8, 4.7, 8	59%	93%	94%	123
	0, 0.4, 2.2, 5.9	41%			
PSO	0, 0.7, 4.4, 8	59%	93%	94%	106
	0, 0.4, 2.2, 5.8	41%			
ECR	0, 2.4, 5.3, 8	54%	85%	94%	120
	0, 0.8, 2.3, 7.3	46%			
Two group four design points $D_{s3}-$optimal					
PSO	0, 1.4, 8, 8	100%	28%	99%	110
	0, 2.5, 8, 8	0%			
ECR	0, 1.9, 3.7, 7.8	91%	57%	98%	120
	0, 0, 2.9, 8	9%			

Table 3.4. D- and D_{s2}-optimal designs and efficiencies for the model without the genetic covariate. PSO stands for the `optim_pso()` function and ECR stands for the `ecr()` function.

Algorithm	Design	Proportion	D-efficiency	D_{s2}-efficiency	CPU (s)
One group five design points D-optimal					
Equispaced	0, 2, 4, 6, 8	-	71%	25.5	-
Simplex	0, 0.4, 2.3, 5.1, 8	-	100%	96%	20
PSO	0, 0.4, 2.1, 5.2, 8	-	100%	96%	30
ECR	0, 0.6, 2.3, 5.1, 8	-	99%	95%	30
One group five design points $D_{s2}-$optimal					
PSO	0, 1.2, 8, 8, 8	-	0%	100%	23
ECR	0, 1.2, 7.8, 8, 8	-	0%	100%	30
Two group four design points D-optimal					
Simplex	0, 1.2, 5.1, 8	54%	78.5%	93%	90
	0, 0.4, 2.3, 8	46%			
PSO	0, 0.4, 4.3, 8	51%	79%	91%	41
	0, 0.5, 2.0, 5.7	49%			
ECR	0, 0.7, 5.1, 8	58%	78%	91%	60
	0, 0.4, 1.7, 5.6	42%			
Two group four design points $D_{s2}-$optimal					
PSO	0, 2.9, 6, 8	100%	2%	93%	41
	0, 1.3, 7.0, 8	0%			
ECR	0, 0.6, 2.3, 6	54%	74%	86%	60
	0, 1.1, 4.5, 8	46%			

sense to have a multigroup scenario. For instance, the two group D_{s3}-optimal designs have groups of 0% and 9% respectively. Despite the fact that we have one less sampling time per individual compared to the one group five time point D-optimal design, the two group and four sampling times are sufficiently informative to identify and estimate parameters with acceptable precision and perhaps interesting designs with a D-efficiency between 85 - 93% and a D_{s3}-efficiency of 94%. The efficiency of the multigroup design is expected to be lower because there will be fewer samples taken overall, i.e. 400 observations for this multigroup design compared to 500 with the five point one group design if 100 patients are measured.

The CPU times are consistently longer for computing $D_{s3}-$optimal designs than D-optimal designs and two group designs take longer to compute than one group, but each design ran in under 2.5 minutes which is manageable in practice.

Table 3.4 displays the designs when the genetic covariate is not included in the model. In this table, we present results for D- and D_{s2}-optimal designs focusing on the fixed and random clearance parameters. We can compare the design points between Table 3.3 and 3.4, but the D- and D_s-optimality and efficiency values should not be compared due to partitioning and dimensionality. The one group five time point D-optimal design did not differ between Table 3.3 and Table 3.4 and took about the same amount of time to compute. The one group D_{s2}-optimal designs again had two time points at hour 8. The two group D_{s2}-optimal designs also did not differ practically from that of the D_{s3}-optimal designs potentially suggesting the design does not differ whether the genetic covariate is included in the model or not.

The physician has to make a decision whether to choose the D-optimal design to estimate all parameters well or the D_s-optimal design which tends to focus on the later time points to best estimate the parameters of interest. We next present multiple objective results to help make a balanced, calculated decision on which design points are best for sustained-release lithium monitoring in the 8 hour window.

3.4.4 Robust designs

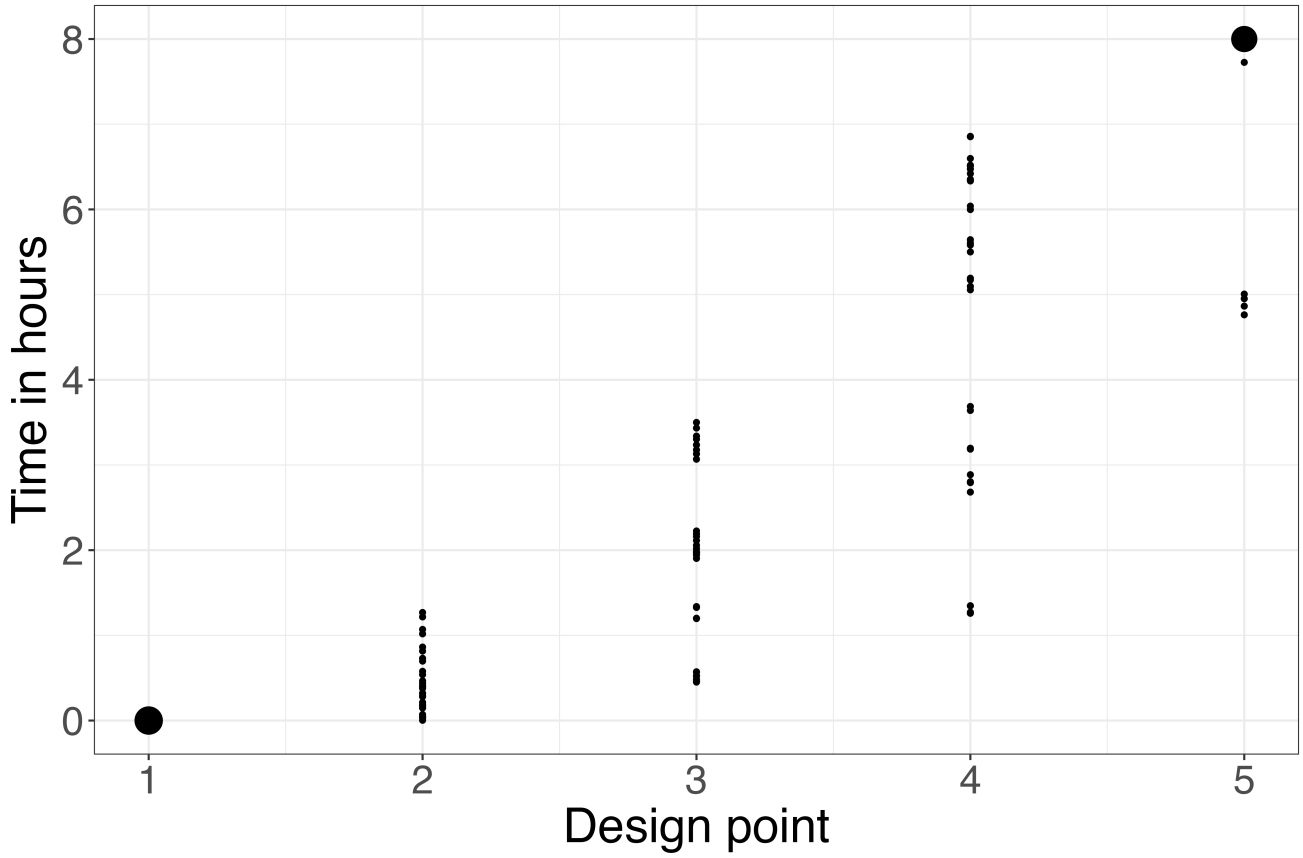
We restrict our analysis to find and discuss locally optimal designs using nominal values physicians believe are reliable, and the values seen in Table 3.1 were the only values the physicians were willing to use. We have a single set of nominal values from the earlier lithium study with 17 patients (Couffignal et al. 2019) and there were no other sets of nominal values available. In our case, physicians were also not willing to use other sets of nominal values. However, our methodology is general and applies to any set of nominal values. Additionally, we discuss pseudo-bayesian designs, which are robust to misspecifications in the nominal values in Table 3.2.

If additional nominal values were available, we would have constructed designs robust to misspecifications in the original values given to use. They include minimax types of designs and fully Bayesian optimal designs. Minimax types of optimal designs allow the user to specify a range of plausible values for each model parameter, and they minimize the maximal inefficiency across all possible nominal values (Chen et al. 2015, 2017). The design problem is harder to find because the design criterion is non-differentiable and has at least 2 nested optimization problems to solve. Alternatively, one may use a Bayesian paradigm and find Bayesian optima designs to estimate model parameters; see Chen et al. (2015, 2017) and Masoudi et al. (2019) who also used metaheuristics and found Bayesian optimal designs for a variety of models.

A simpler method is to implement a hypercube D-optimal design (HCD-optimal design) that optimises a pseudo-Bayesian robust criterion (Foo and Duffull 2010). To find such a design, we used the bootstrapped confidence intervals in Couffignal et al. (2019) for each parameter as prior information. The criterion uses every combination of the 2.5th and 97.5th percentiles from the intervals and the HCD-optimal design maximizes the sum of the log-determinants of the information matrices for each set of parameter combinations; with 5 fixed effects, there are $2^5 = 32$ summands. We optimized the criterion and found that

the HCD-optimal design is similar to our locally D-optimal designs, suggesting that for our problem, the designs are relatively robust to misspecifications in the nominal values. Figure 3.2 shows the D-optimal design time points from each of the 32 individual designs. At each design point, there are 32 points signifying each optimal answer. The spread of each design point is relatively small, signifying there would not be a drastic change in designs if the nominal values were different.

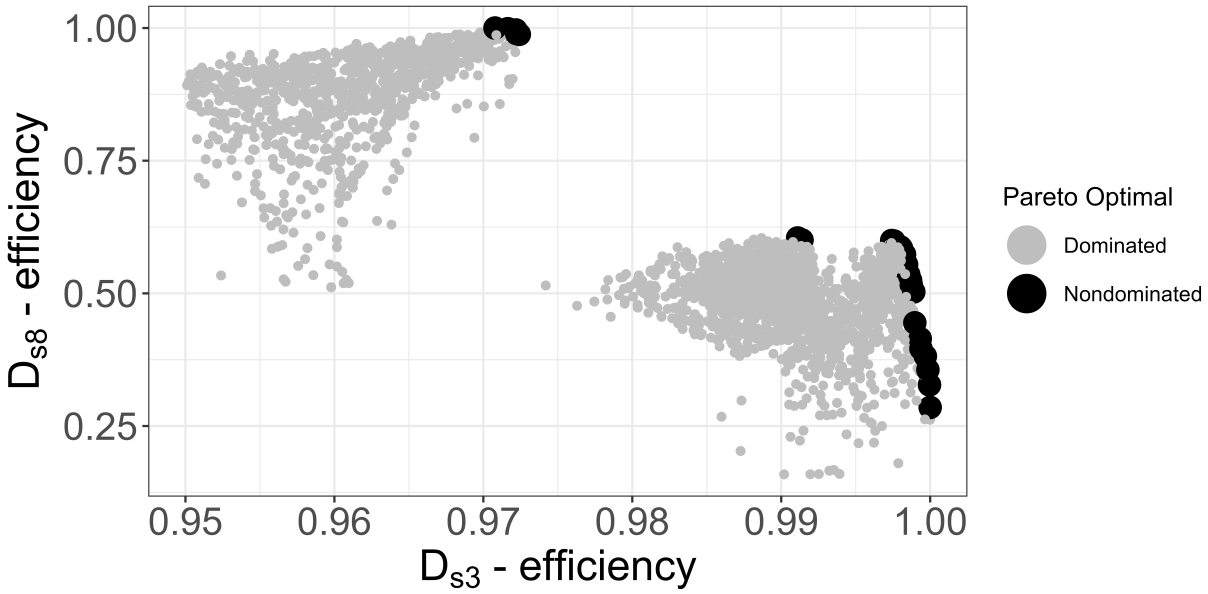
Figure 3.2. Different design points seen with the hypercube D-optimal designs.



3.4.5 Multiple objective results

We compare the efficiency trade-offs when the two objectives $D_{s3}-$optimality and D_{s8}-optimality are optimized simultaneously. The Pareto front in Figure 3.3 was created from 50 simulations of the `ecr()` function in R. Each point on the plot represents a unique design found in the simulations. The black circles represent the Pareto front and are the designs the physician should be interested in. In contrast, the gray circles represent designs with dominated solutions, and so they are inferior or sub-optimal solutions.

Figure 3.3. Pareto front plot for a dual-objective design that balances the efficiencies for estimating the clearance and genetic parameters (D_{s3}-optimality) and the remaining eight parameters (D_{s8}-optimality).



There are three separate clusters of solutions with a discontinuity in between each. Discontinuities in Pareto fronts are possible, and may present good candidates or high trade-off points to analyze further (Rachmawati and Srinivasan 2009). From visual inspection, the left-cluster maximizes the remaining 8 parameters best while the right-cluster maximizes the 3 parameters of interest best. If we are only interested in the 3 parameters, we may be more

inclined to select black dots from the right cluster which estimate the parameters of interest at 99.5 - 100% efficiency, but have a simultaneous trade-off of between 27.5 - 60% efficiency for estimating the remaining 8 parameters. The top point of the right-cluster is preferred because for less than a 1% trade-off in D_{s3}-efficiency, we gain 32.5% in D_{s8}-efficiency. The middle cluster does not gain too much D_{s8}-efficiency at the expense of about 1% loss in D_{s3}-efficiency. A potentially better trade-off can be to choose the left-cluster's black dots because these selected points are designs with about 97% efficiency for estimating the 3 parameters of interest while achieving near maximum efficiency for the remaining 8 parameters as well. There may be distinctions in designs between each cluster.

Further inspection into the time points in each cluster can help the physician make a more informed decision. The right-cluster has time points similar to those found in the D_s-optimal designs where every design in the right cluster has two of the five time points at 8h. The designs in the left-cluster find 97% efficiency for the 3 parameters of interest and simultaneously 100% efficiency for the remaining 8 parameters and are actually like the D-optimal designs found in Table 3.3, e.g. 0, 0.4, 2.1, 5.1, and 8h. This same efficiency calculation happened to be performed in Table 3.3, and the multiple objective optimization confirmed it to be a good decision for the physician to choose the the D-optimal design because it best estimates all parameters simultaneously and only loses 3% of information from the parameters of interest.

Ultimately, the physicians need to make a decision on which time points to measure patients with bipolar disorder who are taking lithium. We provide design efficiencies based on optimizing the estimated parameters of a complicated PK/PD NLMEM model by using designs and metaheuristic algorithms not seen in current softwares. Our results show the time points of 0, 0.4, 2.1, 5.1, and 8h may provide the best trade-off between optimizing the parameters of interest and the remaining, less important parameters. This multiple objective efficiency method helped quantify the value of each design to particular parameters and is generalizable to any PK/PD model with any amount of objectives.

CHAPTER 4

Project 2: Optimal global clinical trial recruitment plans

In this chapter, we show the importance, capability and utility of metaheuristics to optimize complex recruitment designs for global clinical trials using the PG model as an exemplary model. The PG model for patient recruitment assumes that the patients arrive at different sites according to doubly stochastic Poisson processes with rates that are gamma distributed random variables (Anisimov and Fedorov 2007a, Anisimov 2011, 2020). The model is the industry standard and has been adopted by multiple large pharmaceutical companies (Anisimov and Austin 2023). For example, Best et al. (2022) used the PG model to predict COVID-19 pandemic impact on clinical trial recruitment, and Perperoglou et al. (2022) used the PG model to model time-varying recruitment rates in multi-center clinical trials. The PG model not only can forecast patient recruitment, it can also be extended for centralized statistical monitoring of clinical site's recruitment performance, drug supply chain modeling, and predicting events in event-driven trials (Anisimov and Austin 2020, Anisimov et al. 2021, 2022).

As we will discuss, the model can be used to formulate an optimization problem with the goal of minimizing the total monetary cost ($ millions) for a global recruitment plan. For a small scale clinical trial (< 10 countries), there are so few potential recruitment plans that it is possible to computationally evaluate each possible design and choose the lowest costing recruitment plan. A linear version of the problem used the optimization technique developed in Anisimov and Austin (2023) for large clinical trials, a step-wise linear approximation at

each step and the traditional linear simplex. Anisimov and Austin (2023) updated the model to include non-linear constraints and alluded to the use of evolutionary algorithms and other biologically inspired metaheuristic algorithms to handle the non-linear functions, which the simplex and other conventional methods are not designed to handle, hence the motivation for this work. We extend this analysis to multiple-objective clinical trials using metaheuristics for a complete view on clinical trial recruitment plans.

4.1 Background

The National Institute of Health Research Health Technology Assessment Programme (HTA) calculated that between 2002 - 2016, only 55% of their clinical trials reached their original target sample size (Sully et al. 2013, Walters et al. 2017). Trials that do not recruit well have reduced power and may give misleading conclusions (Harrison 2016, Sun et al. 2022). A new drug in 2023 can cost roughly between $40 millions and about $5 billions, so there is little room for error once the drug gets to the clinical stage (Schlander et al. 2021). Clinical trial organizers and decision makers plan for recruitment and must incorporate factors like incidence rates of diseases in the various countries, healthcare infrastructure, regulatory environments, as well as, marketing availability and drug distribution systems (Senn 1997). To address the complicated issues in recruitment planning, decision makers use statistical models and often employ Mixed Poisson types of models to forecast patient recruitment in large and small clinical trials taking into account potential variations (Barnard et al. 2010, Anisimov 2016, Gkioni et al. 2019).

While the PG model is the industry standard adopted by large pharmaceutical companies, there have been other models developed and used, each with different capabilities and limitations. We briefly mention them here. For a full discussion on recruitment models, please see recent review Barnard et al. (2010), Anisimov (2016), Gkioni et al. (2019). Simple, deterministic methods for patient recruitment do not account for real-world variability

(Carter 2004, Carter et al. 2005, Comfort 2013). Mixed Poisson types of models are widely used to estimate the expected number of patients recruited over time (Barnard et al. 2010, Anisimov 2016, Gkioni et al. 2019). Williford et al. (1987) proposed a Poisson process with a gamma distributed rate to model the overall recruitment for a trial without considering specific sites. Several other papers fix the Poisson recruitment rate, λ, often as the average of all sites (Senn 1997, 1998, Carter et al. 2005). In multi-center clinical trials, a fixed rate is often an oversimplification, as sites vary in recruitment capabilities. Each site within each country has a different recruitment base and varied costs. For multi-center trials, these numbers are aggregated in hopes of reaching the necessary sample size calculated in the statistical analysis plan.

The Bayesian paradigm incorporates all prior information into the design and analysis and is useful at the interim analysis of a trial. Two other examples of Bayesian approaches for recruiting patients into a global or multi-center trial are (i) Gajewski et al. (2008) was interested in global recruitment, but not at individual sites, and Jiang et al. (2016) implemented the method in the R package 'accrual', and (ii) Liu et al. (2020) proposed a Bayesian model to predict recruitment rates for a multi-center trial with varying site activation times. These methods are interesting but may be unsuitable at or near the start of a trial when there is a limited number of active sites and patients recruited.

4.2 Poisson-gamma patient recruitment model

This section focuses on the PG model, originally developed by Anisimov and Fedorov (2007a) and further developed by Anisimov (2011) and Anisimov and Austin (2023).

The Poisson recruitment model assumes that an individual site i recruits patients at a rate λ_i, i.e. the probability that there are k patients recruited in site i at time t if the site is initiated at time 0 is defined by the Poisson distribution:

$$P\left(\Pi_\lambda(t) = k\right) = \frac{e^{-\lambda_i t}(\lambda_i t)^k}{k!}, k = 0, 1, ... \tag{4.1}$$

where $\Pi_\lambda(t)$ is an ordinary Poisson process with rate λ. Denote also by $\Pi(\lambda)$ a Poisson random variable with parameter λ.

The rate of recruitment in the PG model is the gamma distributed parameter used to represent real life variability between sites and countries. This means λ_i is viewed as a random variable which has a gamma distribution with probability density function:

$$\lambda_i \xrightarrow{D} \Gamma(\alpha, \beta) = \frac{e^{-\beta x}\beta^\alpha x^{\alpha-1}}{\Gamma(\alpha)} \tag{4.2}$$

where $\Gamma(\alpha) = \int_0^\infty e^{-x}x^{\alpha-1}dx$ is a gamma function.

Assume now that the recruitment rate λ has a gamma distribution with parameters (α, β) and introduce a mixed (doubly stochastic) Poisson process $\Pi_\lambda(t)$. According to Bernardo and Smith (2004), $\Pi_\lambda(t)$ is a Poisson-gamma (PG) process with parameters (t, α, β) and

$$P(\Pi_\lambda(t) = k) = \frac{\Gamma(\alpha + k)}{k!\ \Gamma(\alpha)}\ \frac{t^k\beta^\alpha}{(\beta + t)^{\alpha+k}},\ k = 0, 1, .. \tag{4.3}$$

For convenience also denote $\mathbf{PG}(t, \alpha, \beta)$ as a PG random variable that has the same distribution as $\Pi_\lambda(t)$. For $t = 1$, $\Pi_\lambda(1)$ has the same distribution as $\Pi(\lambda)$ (mixed Poisson variable). In this case, for simplicity, we use the notation $PG(\alpha, \beta)$ instead of $PG(1, \alpha, \beta)$.

According to Johnson et al. (2005), the distribution of $\Pi_\lambda(t)$ in (4.3) for any $t > 0$, can be also described as a negative binomial distribution with

$$P(\Pi_\lambda(t) = k) = P(NB(\alpha, \frac{\beta}{\beta + t}) = k), k = 0, 1, ... \tag{4.4}$$

Here, $NB(\alpha, p)$ is a random variable which has a negative binomial distribution with parameters α and probability p. In the standard R programming environment (R Core Team 2021), there are functions 'dnbinom' and 'pnbinom' to facilitate the calculation of the PDF and CDF of the negative binomial distribution.

4.2.1 Modeling unrestricted recruitment

In this section we consider first unrestricted and competitive recruitment, that means, each site continues to recruit patients until the total sample size (overall planned number of patients) is reached. Each site i is recruiting with the rate λ_i. However, the amount each site can recruit depends on the time of site initiation and duration of recruitment.

The start date of each clinical site is not the same. Denote u_i as the time of initiation of site i. Then, the duration of active recruitment up to a given interim time t for a site activated at time u_i can be defined by the function $x(t, u_i) = max(0, t-u_i)$. Correspondingly, the cumulative recruitment rate in time interval $[0, t]$ has the form

$$\Lambda_i(u_i, t) = \lambda_i x(t, u_i) \tag{4.5}$$

Note that in Anisimov et al. (2007), the model was considered where the start dates of each site have a uniform distribution within some interval $[a_i, b_i]$. Other types of distributions for u_i (gamma and beta) were considered in Anisimov (2020). However, in this paper we restrict our attention to the case where the times of site initiation are given deterministic values that are provided at the time of recruitment planning based on historical data and expert knowledge.

An entire country's cumulative recruitment rate is the summation of each site's rate, i.e. for an indicator set of sites $\mathbf{I_s}$ for the number of sites in country s, N_s:

$$\Lambda(\mathbf{I_s}, t) = \sum_{i \in \mathbf{I_s}} \lambda_i x(t, u_i) \tag{4.6}$$

It follows that the recruitment process, how many patients will be recruited by time t in country s, can be modelled as a mixed Poisson process with the country cumulative rate $\Lambda(\mathbf{I_s}, t)$.

Let $\bar{\mathbf{N}}$ represent the vector of the number of sites in each country that defines the global recruitment plan, i.e. $\bar{\mathbf{N}} = (N_1, ..., N_s)$, and it is assumed that the times for initiation of

these sites are known, e.g. for country s, $(u_1^s, ..., u_{N_s}^s)$. The global cumulative recruitment rate at time t is a sum of each country's recruitment rates, i.e.

$$\Lambda(t, \bar{\mathbf{N}}) = \sum_{s=1}^{S} \sum_{i \in \mathbf{I_s}} \lambda_i x(t, u_i) \tag{4.7}$$

Correspondingly, the global recruitment process, $n(t)$, as the sum of country processes $n(\mathbf{I_s}, t)$, can be represented as a mixed Poisson process with the global cumulative rate $\Lambda(t, \bar{\mathbf{N}})$. The global rate is represented as a sum of gamma distributed variables with different parameters and in general it does not have a gamma distribution. Therefore, the global recruitment process in general is not a PG process.

Thus, to calculate different predictive characteristics for the global recruitment process $n(t)$, we can consider first a normal approximation. For this purpose, we need to calculate the mean and the variance of the global rate and use the following property of a mixed Poisson distribution: if λ is a random variable, then

$$E[\Pi(\lambda)] = E[\lambda] \text{ and } \ Var[\Pi(\lambda)] = E[\lambda] + Var[\lambda]. \tag{4.8}$$

Let $m_i = E[\lambda_i]$ and let $\sigma_i^2 = Var[\lambda_i]$ be, respectively, the mean and the variance of the recruitment rate λ_i for site i. There is a simple relation between the parameters of a gamma distribution and the mean and the variance: $m_i = \alpha_i/\beta_i$ and $\sigma_i^2 = \alpha_i/\beta_i^2$. Assume for simplicity that the mean and the variance of the recruitment rates are the same for all sites in each country and denote for country s,

$$m(s) = m_i, \ \sigma^2(s) = \sigma_i^2, \ i \in \mathbf{I_s}.$$

The mean and the variance of the cumulative country rates at time t are, respectively,

$$E(\mathbf{I_s}, t) = m(s) \sum_{i \in \mathbf{I_s}} x(t, u_i) \tag{4.9}$$

and

$$S^2(\mathbf{I_s}, t) = \sigma^2(s) \sum_{i \in \mathbf{I_s}} x^2(t, u_i). \tag{4.10}$$

Consequently, the mean and variance of the global cumulative rate are calculated as the sum of each country's mean and variance, i.e.

$$E(t, \bar{\mathbf{N}}) = \sum_{s=1}^{S} E(\mathbf{I_s}, t) \tag{4.11}$$

and

$$S^2(t, \bar{\mathbf{N}}) = \sum_{s=1}^{S} S^2(\mathbf{I_s}, t). \tag{4.12}$$

This implies that by (4.8), for any $t > 0$,

$$E[n(t)] = E(t, \bar{\mathbf{N}}) \text{ and } \; Var[n(t)] = E(t, \bar{\mathbf{N}}) + S^2(t, \bar{\mathbf{N}}) \tag{4.13}$$

However, to obtain a good accuracy of the approximation, it is recommended to apply the normal approximation to trials with at least $15 - 20$ countries.

Anisimov (2020) and Anisimov and Austin (2023) showed that the country recruitment process could be well-approximated by a PG process with some aggregated parameters. This approximation works well for any number of sites and can be used to create predictions of the recruitment process with mean and predictive bounds using the quantiles of a PG distribution. Correspondingly, we can also use a PG approximation of the global process using approximative PG processes in the individual countries.

4.2.2 Probability of success

The likelihood of recruiting the target number of patients by the pre-specified time frame is what we call the probability of success (PoS). The PoS of a recruitment plan is a defining feature and when a recruitment plan's PoS is less than satisfactory, the PoS can be recalculated after more sites are included.

Let $\tau(n)$ be the time the target number of patients n is reached. We denote the PoS of reaching n patients by time T as $P(\tau(n) \leq T)$. For global recruitment process $n(t)$

$$P(\tau(n) \leq T) = P(n(T) \geq n). \tag{4.14}$$

Using a normal approximation and relation (4.14), it is straightforward to prove that the clinical trial with recruitment plan $\bar{\mathbf{N}}$ will complete recruitment up to time T with probability P if the following condition is satisfied:

$$\frac{E(T,\bar{\mathbf{N}})-n}{\sqrt{E(T,\bar{\mathbf{N}})+S^2(T,\bar{\mathbf{N}})}} \geq z_P \tag{4.15}$$

where z_P is the P-quantile of a normal distribution.

If we use a PG approximation, then PoS can be derived using relation (4.14) and a CDF of a corresponding PG process approximating the global recruitment process $n(T)$.

4.2.3 Restricted recruitment with caps on each country's number of patients

In global clinical trials, restrictions are variously imposed on countries by regulatory agencies and may include the patient sample size and recruitment duration in each country or even requiring a reasonable diversity in the patient population. Modeling recruitment with restrictions at the country level was investigated in Anisimov and Austin (2022, 2023). Such restrictions, among others, lead to constraints where there are upper bound restrictions on the recruitment process, e.g. in country s, the recruitment in this country is stopped when the number of patients recruited reaches a defined cap $L(s)$. For countries with few sites to recruit, we will use a PG approximation of the country recruitment processes.

Let the sites in country s be indexed by $\mathbf{I_s}$. Using notation (4.9) and (4.10), let us introduce the variables for country s:

$$A(\mathbf{I_s},t)=E^2(\mathbf{I_s},t)/S^2(\mathbf{I_s},t) \quad \text{and} \quad B(\mathbf{I_s},t)=E(\mathbf{I_s},t)/S^2(\mathbf{I_s},t). \tag{4.16}$$

The following lemma was proved in Anisimov and Austin (2023): The distribution of the unrestricted recruitment process $n(\mathbf{I_s},t)$ can be well approximated by the distribution of a PG random variable

$$PG(A(\mathbf{I_s},t),B(\mathbf{I_s},t)) \tag{4.17}$$

.

According to the above Lemma, the distribution of the unrestricted recruitment process $n(\mathbf{I_s}, t)$ in country s can be approximated by the PG distribution with parameters $(1,\ A(\mathbf{I_s}, t),\ B(\mathbf{I_s}, t))$. This means that, in (4.3) we designate $\alpha = A(\mathbf{I_s}, t),\ \beta = B(\mathbf{I_s}, t),$ and $t = 1$. For computational purposes we can use the negative binomial distribution in (4.4) and put $\alpha = A(\mathbf{I_s}, t),\ \beta = B(\mathbf{I_s}, t),$ and $t = 1$.

Define now in country s with restricted cap $L(s)$ the capped restricted recruitment process $n^{L(s)}(\mathbf{I_s}, t)$ by

$$n^{L(s)}(\mathbf{I_s}, t) = \begin{cases} n(\mathbf{I_s}, t) & as \quad n(\mathbf{I_s}, t) < L(s) \\ L(s) & as \quad n(\mathbf{I_s}, t) \geq L(s). \end{cases} \tag{4.18}$$

Then the distribution of the restricted process $n^{L(s)}(\mathbf{I_s}, t)$ can be calculated directly via the distribution of the unrestricted process $n(\mathbf{I_s}, t)$:

$$P(n^{L(s)}(\mathbf{I_s}, t) = k) = \begin{cases} P(n(\mathbf{I_s}, t) = k) & as \quad 0 \leq k < L(s) \\ 1 - \sum_{k=0}^{L-1} P(n(\mathbf{I_s}, t) = k) & as \quad k = L(s) \\ 0 & \text{otherwise.} \end{cases} \tag{4.19}$$

Thus, we can use (4.4) with $\alpha = A(\mathbf{I_s}, t),\ \beta = B(\mathbf{I_s}, t),$ and $t = 1$ to calculate the distribution of the restricted recruitment process in country s.

To calculate the distribution of the global recruitment process with restrictions, we use a distributional approach. The distribution of the global restricted recruitment process is a convolution of the distributions of countries' restricted processes and can be calculated numerically in R using the function convolve() which uses a discrete Fourier transform (R Core Team 2021).

The PoS can be derived using relation (4.14) and the distribution of the global recruitment process with restrictions $n(T, L(1), .., L(S))$. Alternatively, PoS can be calculated using a normal approximation. Anisimov and Austin (2022, 2023) derived closed form expressions

for the mean and the variance of the restricted process in each country:

$$E[n^{L(s)}(\mathbf{I_s},t)] = E(\mathbf{I_s},t)P((A(\mathbf{I_s},t)+1, B(\mathbf{I_s},t)) \leq L(s)-2)+L(s)\Big(1-P((A(\mathbf{I_s},t), B(\mathbf{I_s},t)) \leq L(s)-1)\Big) \tag{4.20}$$

$$E[(n^{L(s)}(\mathbf{I_s},t))^2] = (E^2(\mathbf{I_s},t)+S^2(\mathbf{I_s},t))P(PG(A(\mathbf{I_s},t)+2, B(\mathbf{I_s},t)) \leq L(s)-3)+E(\mathbf{I_s},t)P(PG(A(\mathbf{I_s},t)+1, B \tag{4.21}$$

$$Var[n^{L(s)}(\mathbf{I_s},t)] = E[(n^{L(s)}(\mathbf{I_s},t))^2] - (E[n^{L(s)}(\mathbf{I_s},t)])^2 \tag{4.22}$$

The relation (4.14) is also valid for the restricted process and so the same condition (4.15) should be satisfied to complete recruitment up to time T with probability P where the mean and the variance are calcuated using the sums of the means and variances of restricted processes in the countries defined in (4.20) - (4.22). This computational approach is simpler and takes much less time to complete the calculations. The only restriction is that, as normal approximation is used here and at the country level, it is recommended that this approach be used for trials with at least $15-20$ countries.

4.2.4 Optimal trial design

This subsection defines additional notation for factors like monetary costs and practical constraints. Following Anisimov and Austin (2023), suppose that n patients are to be recruited into a global clinical trial. There are up to S given countries to include in the trial plan. Let the vectors $\bar{\mathbf{H}} = (H_1, .., H_S)$ and $\bar{\mathbf{U}} = (U_1, .., U_S)$ be, respectively, the given lower and upper bounds for the number of sites allowed in each country. The number of sites within each country must fit within minimal and maximal bounds ($\bar{\mathbf{H}}$ and $\bar{\mathbf{U}}$). Let $T = T_{plan}$ be the fixed target recruitment timeline. Let the PoS to recruit n patients by time T using recruitment plan $\bar{\mathbf{N}}$ be denoted as $P(n, T, \bar{\mathbf{N}})$. Assume, for simplicity, that the mean and the variance $(m(s), \sigma^2(s))$ of the recruitment rates in any country s are the same for all sites in this country. Further, suppose that all sites are planned to be activated before the target time T, and for any given country s with sites indexed by set $\mathbf{I_s}$, let the corresponding number of

sites be labeled as N_s. Let the vector of site activation times for country s be $(u_1^s, .., u_{N_s}^s)$. These site activation times may be generated from a uniform grid on some interval $[a_s, b_s]$ or on a piece-wise uniform grid.

There are many cost components in the study and they include

1. the vector of costs per one site in each country, $\bar{\mathbf{C}} = (C_s, s = 1, .., S)$;
2. the vector of costs per one recruited patient in each country, $\bar{\mathbf{c}} = (c_s, s = 1, .., S)$;
3. the vector of costs per including country s with a non-zero number of sites, $\bar{\mathbf{Q}} = (Q_s, s = 1, .., S)$;

Let $C(T, \bar{\mathbf{N}}, \bar{\mathbf{C}}, \bar{\mathbf{c}}, \bar{\mathbf{Q}})$ be the total mean cost of the trial for the time period $[0, T]$ for the above setup. This cost is

$$C(T, \bar{\mathbf{N}}, \bar{\mathbf{C}}, \bar{\mathbf{c}}, \bar{\mathbf{Q}}) = \sum_{s=1}^{S} c_s E[n^{L(s)}(\mathbf{I_s}, T)] + \sum_{s=1}^{S} C_s N_s + \sum_{s=1}^{S} Q_s I(N_s > 0) \tag{4.23}$$

where the expression $E[n^{L(s)}(\mathbf{I_s}, T)]$ is defined in (4.20) and shows the mean number of patients recruited in country s up to time T for the restricted process.

4.2.5 Objective functions for an optimal recruitment plan

The decision makers ultimately need to choose one recruitment plan to rollout for the Phase III trial. The above discussion suggests that we want to find a recruitment plan for a clinical trial that (a) minimizes the overall cost and (b) stays within certain real-life constraints. The recruitment plan must not use more or less sites than the upper and lower bounds in each country (H_s, U_s), must stop recruiting when the patient cap limit per country L_s is reached, and the PoS constraint must be satisfied. This PoS constraint can be set at any nominal value, sometimes 90% to nearly guarantee success, but also 50% to provide a cost-benefit trade-off analysis. The optimal recruitment plan discussed above is one that abides by the following conditions:

$$
\begin{aligned}
&\text{minimize:} && C(T, \bar{\mathbf{N}}, \bar{\mathbf{C}}, \bar{\mathbf{c}}, \bar{\mathbf{Q}}) && \text{Total Cost} \\
&\text{subject to:} && P(n, T, \bar{\mathbf{N}}) \geq P && \\
& && H_s \leq N_s \leq U_s && s = 1, 2, .., S \\
& && n(\mathbf{I_s}, T) \leq L(s) && s = 1, 2, .., S
\end{aligned} \tag{4.24}
$$

In the setting of the optimization problem, the expressions for the mean number of patients recruited in country s up to time T, $E[n^{L(s)}(\mathbf{I_s}, T)]$, defined in (4.20), are essentially non-linear subject to the vector of main variables $\bar{\mathbf{N}}$, and the expression for PoS, $P(n, T, \bar{\mathbf{N}})$, is also essentially non-linear and can be calculated via the distribution of the global recruitment process at time T using either a convolution of restricted country processes as described in Section 4.2.3, or using a normal approximation by using relations (4.20) - (4.22) for the mean and the variance of restricted country processes. Therefore, we may expect the objective function to have many local minimums subject to non-linear constraints, and conventional methods may not work properly or may not converge to the point of the global minimum. This motivates the use of metaheuristics to solve the above optimization problem, (4.24).

Further, the flexibility of metaheuristics allows for the construction of similar, alternative optimization problems without difficult adjustments in how the algorithms are used. Instead of minimizing the total cost of the trial, decision makers can plan to minimize the overall number of sites used or the overall number of countries used, shown by Equations (4.25) and (4.26) respectively subject to the same constraints. We also present (4.27), an optimization problem focused on maximizing the PoS given some upper bound cost constraint. Metaheuristics can handle each of these optimization problems with little to no adjustments. In what is to follow, we will use these objective functions to holistically inform the design of a clinical trial's recruitment plan.

$$\begin{aligned} \text{minimize:} \quad & \sum_{s=1}^{S} I(N_s > 0) && \text{Number of Countries} \\ \text{subject to:} \quad & P(n, T, \bar{\mathbf{N}}) \geq P \\ & H_s \leq N_s \leq U_s && s = 1, 2, .., S \\ & n(\mathbf{I_s}, T) \leq L(s) && s = 1, 2, .., S \end{aligned} \tag{4.25}$$

$$\begin{aligned} \text{minimize:} \quad & \sum_{s=1}^{S} N_s && \text{Number of Sites} \\ \text{subject to:} \quad & P(n, T, \bar{\mathbf{N}}) \geq P \\ & H_s \leq N_s \leq U_s && s = 1, 2, .., S \\ & n(\mathbf{I_s}, T) \leq L(s) && s = 1, 2, .., S \end{aligned} \tag{4.26}$$

$$\begin{aligned} \text{maximize:} \quad & P(n, T, \bar{\mathbf{N}}) && \text{Probability of Success} \\ \text{subject to:} \quad & C(T, \bar{\mathbf{N}}, \bar{\mathbf{C}}, \bar{\mathbf{c}}, \bar{\mathbf{Q}}) \leq \$ \\ & H_s \leq N_s \leq U_s && s = 1, 2, .., S \\ & n(\mathbf{I_s}, T) \leq L(s) && s = 1, 2, .., S \end{aligned} \tag{4.27}$$

4.3 Algorithm selection and implementation

The PG model used throughout this work is implemented in a proprietary R package created by the Amgen Center for Design and Analysis, consisting of a team of software engineers,

statisticians, and business experts. It is used in a business setting where output could allow for some time delay because the planning stage is so paramount and takes place months in advance. In deciding which metaheuristic to implement, it made sense to measure the quality and speed of different metaheuristic packages in R to maintain consistency with the package which implements the PG model.

4.3.1 Case study

First, we will provide an illustrative example of data from an earlier large-scale Phase III clinical trial recruitment plan. Suppose a global trial needs to recruit 5000 patients spanning up to 49 countries in 24 months with a budget of $73 millions. The large scale and resource-intensiveness of the trial underscore the necessity to find cost-effective recruitment plans that will result in a full and timely recruitment.

The dataset used includes information for each country estimated for each trial, sometimes based on historical data or expert knowledge. This includes a country's minimum and maximum number of allowable sites, recruitment rates, schedule of site initiation times, costs, and limits on patients. Four countries (China, Japan, United Kingdom, and United States) are required to be in the recruitment plan and the remaining 45 countries may or may not be included in the trial, depending on the optimal plan. The maximum number of sites for each country ranges from 5-200. The average recruitment rate for each country's sites ranges between 0.3 - 2.5 patients recruited per month. The cost per patient for each country ranges from $500 to $7000.

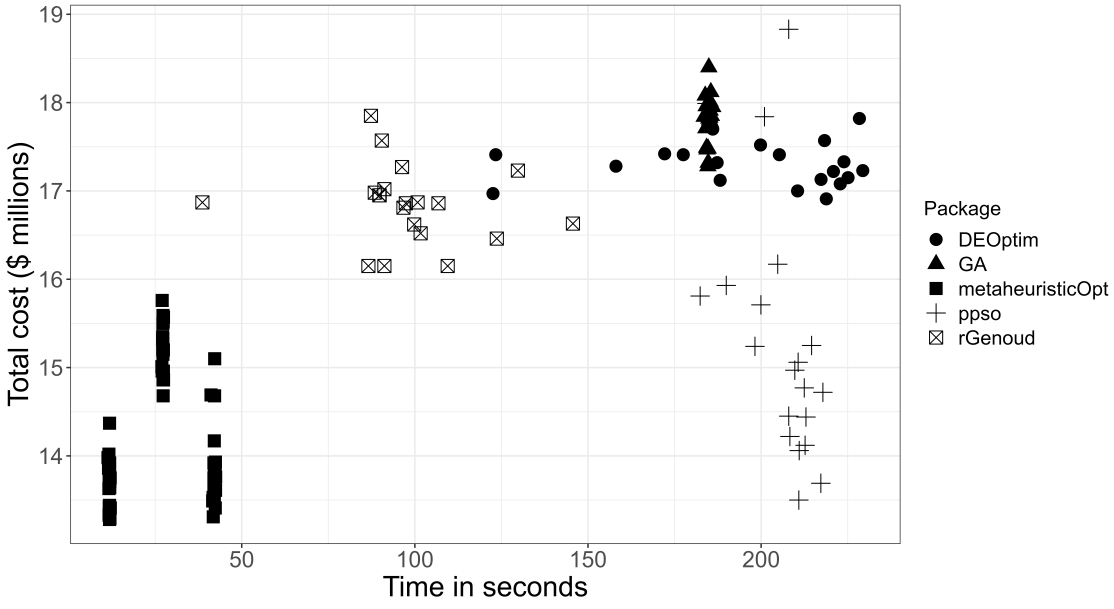
4.3.2 Speeds and qualities of different R packages

The best metaheuristic we seek to install (a) is available, (b) performs well in solution quality and (c) is quick in computation time. First, we decided to analyze different R packages to optimize the model. There are a number of optimization packages available at

https://cran.r-project.org/web/views/Optimization.html. The data science team previously used multiple packages, specifically, `DEoptim`, `GA`, and `rgenoud`. We further tested `ppso` and `metaheuristicOpt` and found the package `metaheuristicOpt` to outperform the rest. In the rest of this subsection, we present preliminary results for this conclusion, which then led us to further inspect this package.

First, we compared the different packages at default settings to see their performance in optimizing (4.24) at a 90% PoS constraint. Figure 4.1 graphs each shape which represent different packages against the solution quality and time to compute. Each package either took longer to perform or performed strictly worse than the `metaheuristicOpt` or both. In any usage case, this motivates our choice of furthering experimentation into the package containing 21 metaheuristic algorithms, `metaheuristicOpt` (Septem et al. 2019).

Figure 4.1. Total cost ($ millions) by time to compute for 20 simulations from different R packages.



The R package `metaheuristicOpt` has 21 diverse metaheuristic algorithms all of which

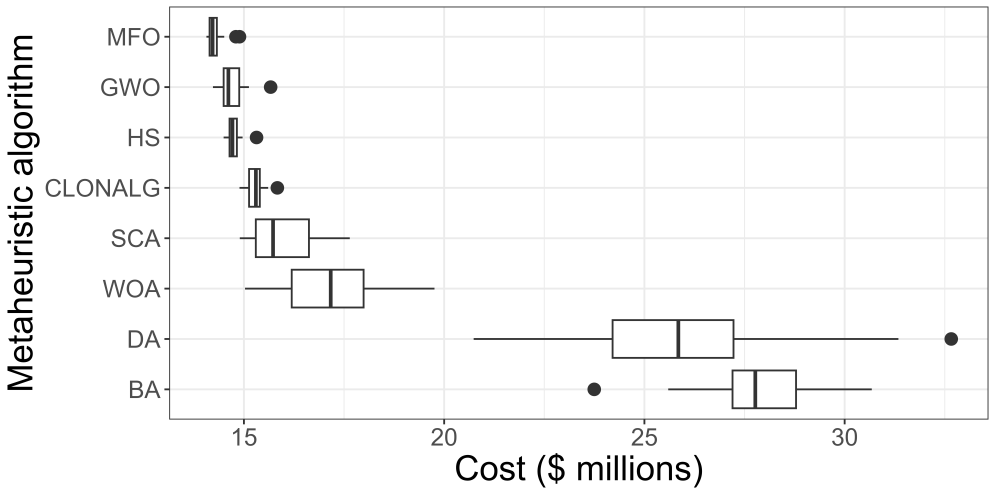
could result in a different answer, and it may or may not be worthwhile to run them all (Septem Riza et al. 2019). The package originally came with 11 algorithms and was later extended to have 10 more. Only certain algorithms were functional for solving optimization problem (4.24), mostly from the original 11 algorithms in the package. In Table 4.1, we present computational speeds for one simulation of each algorithm solving optimization problem (4.24) for a different number of iterations: 250, 500 (default), 1000, and 2000. We see the majority of the algorithms solve the problem quickly with 2000 iterations in under a minute and a half. The Shuffled Frog Leaping algorithm (SFL), Ant Lion Optimizer (ALO), and PSO algorithms all took significantly longer than the others and were not worth further investigation.

Table 4.1. A time comparison of one run of metaheuristics in the `metaheuristicOpt` R package on minimizing total cost ($ millions).

	Number of iterations			
Algorithm	250	500	1000	2000
Whale Optimization Algorithm	4	8	10	19
Moth Flame Optimizer	3	7	11	22
Sine Cosine Algorithm	5	10	18	35
Harmony Search	6	12	23	45
Clonal Selection	7	15	28	54
Grey Wolf Optimizer	8	15	29	56
Bat Algorithm	10	21	37	73
Dragonfly Algorithm	12	23	45	92
Shuffled-Frog Leaping	76	157	292	601
Ant Lion Optimizer	16	46	167	629
Particle Swarm Optimizer	88	182	351	680

Figure 4.2 then presents a boxplot of 20 simulations of each algorithm at default settings for the problem and the resulting cost when minimizing (4.24). The MFO, GWO, and HS algorithms were the best three performers, and hence were then selected for further experimentation and implementation. The Dragonfly algorithm (DA) and Bat algorithm (BA) performed quite poorly in comparison.

Figure 4.2. Total cost ($ millions) on 20 simulations for each algorithm in metaheuristicOpt.



The cumulative simulations, tables and figures helped us decide which algorithms to use going forward and which ones to drop. In particular, we work with the best algorithms based on solution quality and their computing time. The GWO, HS, and MFO algorithms performed well in 2000 iterations which took under 45 seconds for each. The package not only performs well but is open-source and editable too. We were able to manipulate the package through its central function `metaOpt` and each algortihm's "engine". We manually edited items in the package so that the user can designate a starting population, log previous search positions, and provide a convergence criterion (Septem Riza et al. 2019). The modified package is now found in our publicly available updated version in our GitHub repository at https://github.com/maschepps/metaheuristicOpt. In the next subsection, we explore

further the performance properties of these algorithms.

4.3.3 Convergence criteria

How many iterations is worthwhile? In the preliminary simulations we just showed, the algorithms searched for up to a set number of iterations, and then stopped the search process at, at most, 2000 iterations. However, there was no quantification nor guarantee that this number of iterations was complete and not a premature stoppage point. While the algorithms iteratively update and search for better and better solutions, the population can potentially hover around the current best value for some time, unable to find anything better. This period of stagnation can be calculated by counting the number of iterations each algorithm takes to improve from one solution to the next. To do this, we used the manually edited metaheuristic package on GitHub to track the trajectory of an empirically long search, 50,000 iterations for each algorithm. This search took 24 minutes for the MFO algorithm, 47 minutes for the GWO algorithm, and 72 minutes for the HS algorithm. These 20 trajectories for each algorithm are shown in Figure 4.3. The y-axis is the total cost ($ millions), to be minimized, and the highlighted line shows the convergence trajectory of the particular simulation which found the optimal recruitment plan per algorithm. Once the trajectory was captured, we used the `rle()` function in the `data.table` package to count the length of stagnation in each algorithm's search history. The percentiles of each stagnation length are presented in Figure 4.4 to enable an informed decision making process on a stagnation length stoppage rule.

Figure 4.4 is a quantile plot comparing the length of stagnation seen in the 50,0000 iterations across the 20 simulations each of the three different metaheuristic algorithmsGrey Wolf Optimizer (GWO), Harmony Search (HS), and Moth Flame Optimization (MFO). For instance, each of the MFO simulations found their local optimal solution within 1000 iterations, rendering the remaining 49,000+ iterations useless and time consuming. Once a proper convergence criteria is implemented, computation time lost from unnecessary itera-

tions can be saved. As we saw in the initial speed comparisons of each algorithm, Table 4.1, 2000 iterations took at most 45 seconds, and in Figure 4.4, the horizontal line represents a stagnation length of 2000 iterations which covered 90% of possible searches. This level is chosen as a balance between allowing the algorithms enough time to potentially find a better solution and not letting the search process run excessively long. We observe that a stagnation length criteria of 5,000 will cover 95% of the stagnations, but we decided the extra 5% garnered was not worth the additional computation time and does not significantly increase the percentage of finding the best optimum. In the current decision to implement a 2,000 run stagnation stopping rule, we will overcompute for MFO as it did not stagnate for more than 1,000 iterations, but only at the expense of at most 30 seconds (Table 4.1). Further, we decided to implement an upper bound of 50,000 iterations as each simulation found their optimal value before iteration 48,000.

Figure 4.3. Convergence analysis of 20 simulations at 50,000 iterations for each algorithm when minimizing for total cost ($ millions) with a 90% PoS constraint highlighting the optimal trajectory.

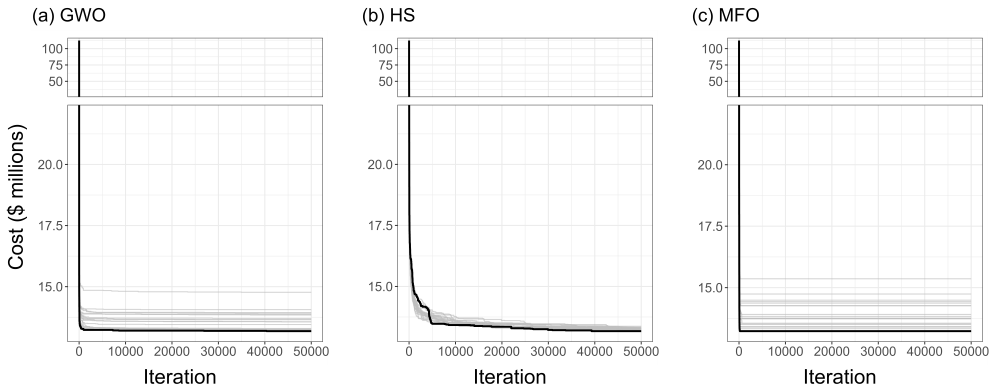
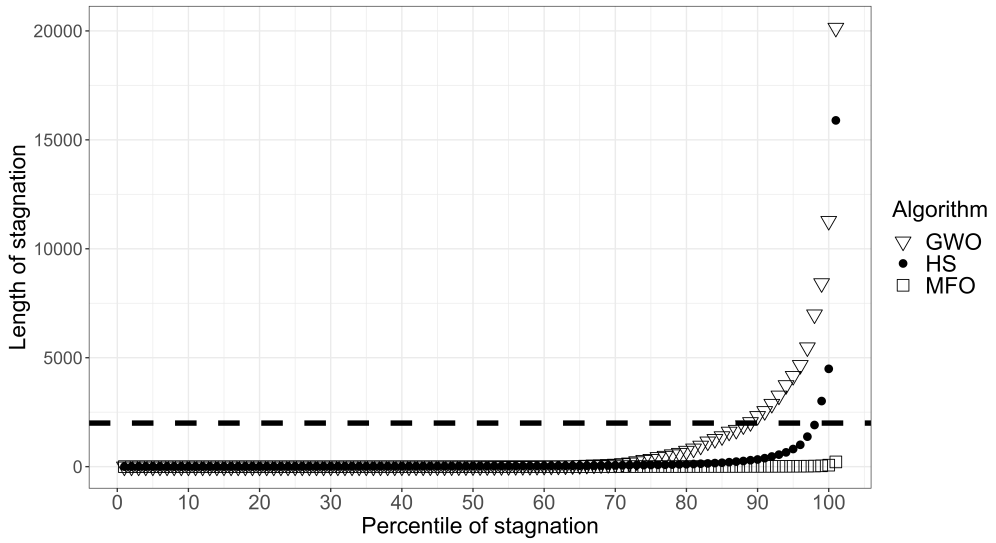


Figure 4.4. Quantile plot of repeated value stagnation in the metaheuristic search process for GWO in minimizing total cost in the large clinical trial.



4.4 The global clinical trial case study and the results

Now that we have introduced the case study and chosen metaheuristics to solve our optimization problems based on preliminary analyses, we will use the selected metaheuristics to solve the above complicated and multifaceted optimization problems. The resulting outcome will present the best possible plan or plans to the clinical trial recruitment organizers and decision makers. We compare performances of the various algorithms and solutions obtained from all simulations by using summary statistics, graphical analyses and non-parametric tests, such as the Kruskal-Wallis test and posthoc Wilcoxon rank-sum test (Kruskal and Wallis 1952, Wilcoxon 1945).

4.4.1 The trade-off between probability and cost

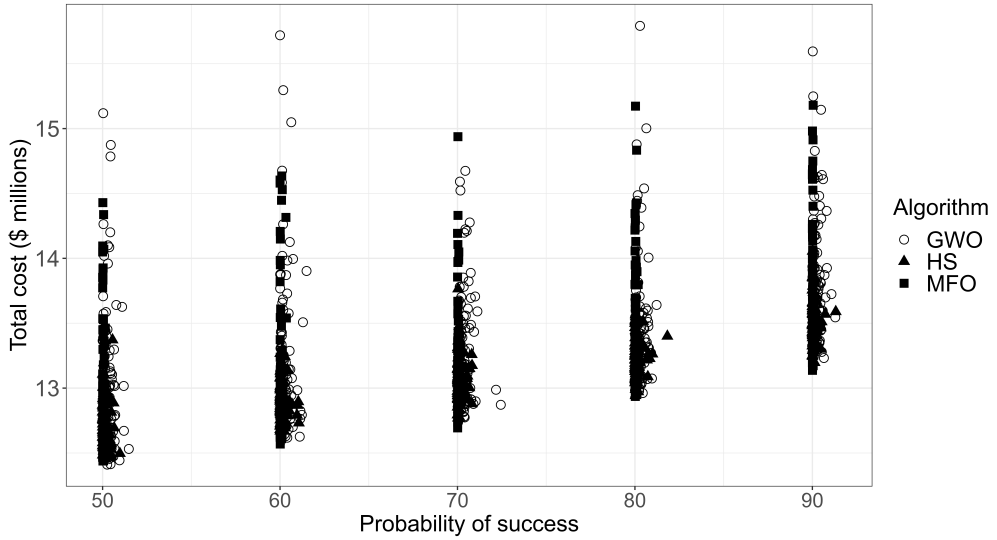
The PoS is an important variable used to quantify the value of a recruitment plan. A prespecified PoS constraint can be set at different values such as 50%, 60%, 70%, 80%,

and 90% PoS to see a cost-benefit trade-off. This will require a unique simulation for each PoS. Because the algorithms are stochastic, and we had ample time to compute, we ran the optimization problem to minimize the total cost ($ millions), (4.24), 100 times per each PoS on each of the three separate algorithms across 20 parallelized CPUs. Here, we present results of these optimizations in Figure 4.5 and Tables 4.2 - 4.5.

Figure 4.5 presents a scatter plot that shows results of all 300 simulations. We compare the PoS to the associated total cost ($ millions) of the $300 \times 5 = 1500$ local optima created using a unique simulation for the five different probabilities. Across all algorithms, there is an observable trend where the solutions with a higher PoS are generally associated with a higher cost. The costs range approximately from $18.5 million to over $23 million as the PoS grows from 50% to 90%. The majority of solutions are close to, sometimes within three decimal places, to the prespecified percentage border with only a few venturing past, e.g. 50% vs 50.1% vs 55%. We, in fact, see many inferior solutions. For instance, in the 90% PoS cluster, locally optimal solutions can vary from just above $13 millions to just under $17 millions. The less expensive recruitment plan is objectively more desirable as it provides the same PoS benefit as the $4 millions reduction in spending. To look at these plans more exactly, we present the lowest cost solution found at the 90% PoS constraint for each algorithm for comparison in Table 4.2.

Table 4.2 displays the lowest cost recruitment plan found from the 100 simulations per each algorithm for the 90% PoS constraint. This enables the trial planners to analyze the impact of using different algorithms. Table 4.2 presents a comparative analysis of the performance of metaheuristic optimization techniques in the context of clinical trial site allocation. The data is organized to show the number of clinical trial sites allocated in each country under the 90% PoS threshold for each algorithm. Each row in Tables 4.2 - 4.5 represents country-by-country information. The first six columns are inputs and constants for each country, such as the maximal number of sites or patients allowed, estimated average recruitment rate, and cost per patient. The columns in Table 4.2 under the "GWO", "HS", and

Figure 4.5. Scatterplot results of metaheuristic simulations analyzing the relationship of probability of success to total cost ($ millions).



"MFO" headings represent the respective optimal recruitment plans. Thus, the table offers a concrete plan for decision makers. Each algorithm allocates a different recommended number of countries and sites within those countries. All three algorithms (GWO, HS, and MFO) demonstrate similar total cost ($ million), with GWO and HS both at $13.2 million and MFO marginally lower at $13.1 million. All three algorithms achieve very similar realized PoS, hovering around the 90% PoS target. The GWO plan used the least amount of sites and countries compared to the other two algorithms signifying less global spread for a similar total cost ($ millions) (372 sites in 13 countries vs 466 sites in 16 countries vs 466 sites in 14 countries). This indicates that despite different strategies in site and patient allocation, the overall effectiveness in achieving the trial's success criteria is comparable across algorithms. The patient cap per country $L(s)$ is approached and recruitment is maximized by all algorithms for countries Hungary, Sweden, Romania, Argentina, Colombia, and Iceland. The GWO recruitment plan nearly maximizes recruitment in Brazil, reaching near the cap of 900 patients and well short of the maximum sites, using 36 of the 60, signifying recruitment

was stopped here near the patient cap. The HS and MFO plan use sites in Norway when the GWO did not. Other differences include the HS plan using the maximum number of sites in Malaysia when no other algorithm recommends recruiting there, and the MFO plan recruits more patients from Latvia. Table 4.2 demonstrates the potential for each algorithm to optimize Equation 4.24 at the 90% PoS. Each algorithm found a nearby solution with a different set of sites and countries.

The 90% PoS constraint is focused on because of the practical near-guarantee it applies to the important and expensive problem of clinical trial recruitment optimization. We do present the cost-benefit PoS trade-off in Tables 4.3 - 4.5, each table separate for each algorithm because, as you have seen, each algorithm with their diverse search strategy found similar costs and different recruitment plans. However, there are some consistencies between Tables 4.3 - 4.5. As the required PoS increases from 50% to 90%, there's a noticeable trend in an increased cost and the total number of patients recruited. The total cost associated with the recruitment plans also increased with the PoS constraint, ranging from $12.4 million at 50% PoS to $13.2 million at 90% PoS. This increase reflects the additional resources required to achieve higher success rates. The same goes for country-by-country differences. In Table 4.3, we see the country of Brazil is increasingly used at each increasing PoS, indicating a more aggressive recruitment strategy in this country would ensure higher success probabilities. Similarly, in Table 4.5, Latvia is used in the 70, 80, and 90% PoS plans, but not the lesser two. However, the relationship between sites, countries, cost, and PoS is not monotonic. The 70% GWO plan uses less sites and less countries than the 60% plan for a slightly greater cost (12.8 vs 12.7 $ millions). This presumably means the 70% plan uses the more expensive sites with good recruitment rates.

We presented here, the optimal solutions found for given scenarios. Next, we will look into the long-term performance of each algorithm. This is important because in most real-life scenarios, there may not be 100 CPU's available for parallel computation. If only a select number of simulations are allowed, it will be crucial to know how robust each algorithm is.

Table 4.2. Country and site specific recruitment plans for each algorithm's lowest cost simulation for a 90% PoS constraint.

	Constants					Algorithms					
	Min Sites	Max Sites	Max Patients	Recruitment Rate	Cost per Patient	GWO		HS		MFO	
Country	$\bar{\mathbf{H}}_\mathbf{s}$	$\bar{\mathbf{U}}_\mathbf{s}$	$L(s)$	$m(s)$	$\bar{\mathbf{c}}$	Sites	Patients	Sites	Patients	Sites	Patients
Hungary	-	60	1100	1.4	121	60	1073	59	1068	60	1073
Sweden	-	50	600	0.8	162	50	584	49	580	49	580
Romania	-	50	300	0.5	135	50	282	46	270	50	282
Argentina	-	55	400	0.7	43	44	384	44	384	48	393
Colombia	-	50	500	1.4	182	37	438	44	478	50	493
Malaysia	-	40	200	0.3	112	-	-	40	158	-	-
Norway	-	40	200	0.4	106	-	-	33	179	35	185
Turkey	-	45	300	0.8	178	2	21	30	275	31	279
Lithuania	-	35	300	0.7	180	28	236	26	279	30	292
United States	25	230	3100	1.5	700	25	626	25	626	25	626
Latvia	-	40	300	0.5	190	-	-	19	163	40	290
China	15	85	1100	2.5	313	15	397	16	424	15	397
Iceland	-	35	200	1.5	106	14	198	12	193	13	196
Japan	10	105	600	0.4	900	10	69	10	69	10	69
United Kingdom	10	115	1600	0.9	633	10	155	10	155	10	155
India	-	55	400	0.8	236	-	-	3	32	-	-
Brazil	-	60	900	2.5	227	36	885	-	-	-	-
...	-	-	-	-	-	-	-	-	-	-	-
Total	60	1015	-	-	-	372	5340	466	5333	466	5310
Total cost ($ millions)	-	-	-	-	-	13.2		13.2		13.1	
Number of countries	4	49	-	-	-	13		16		14	
Realized PoS	-	-	-	-	-	90.6		90.6		90.0	

Table 4.3. Low cost country and site specific recruitment plans found by the GWO algorithm under five different scenarios.

Country	Constants					Probability of success constraints									
	Min Sites $\bar{\mathbf{H}}_s$	Max Sites $\bar{\mathbf{U}}_s$	Max Patients $L(s)$	Recruitment Rate $m(s)$	Cost per Patient $\bar{\mathbf{c}}$	50%		60%		70%		80%		90%	
						Sites	Patients	Sites	Patients	Sites	Patients	Sites	Patients	Sites	Patients
Hungary	-	60	1100	1.4	121	60	1073	60	1073	60	1073	60	1073	60	1073
Sweden	-	50	600	0.8	162	39	513	46	567	42	540	50	584	50	584
Romania	-	50	300	0.5	135	36	222	50	282	32	199	47	273	50	282
Argentina	-	55	400	0.7	43	45	387	44	384	46	389	40	369	44	384
Colombia	-	50	500	1.4	182	34	412	30	371	35	422	43	474	37	438
Brazil	-	60	900	2.5	227	17	546	27	798	30	841	32	861	36	885
United States	25	230	3100	1.5	700	25	626	25	626	25	626	25	626	25	626
Lithuania	-	35	300	0.7	180	24	268	-	-	21	246	-	-	19	228
China	15	85	1100	2.5	313	15	397	15	397	15	397	15	397	15	397
Iceland	-	35	200	1.5	106	11	190	13	196	12	193	13	196	14	198
Japan	10	105	600	0.4	900	10	69	10	69	10	69	10	69	10	69
United Kingdom	10	115	1600	0.9	633	10	155	10	155	10	155	10	155	10	155
Turkey	-	45	300	0.8	178	-	-	-	-	-	-	-	-	2	21
Norway	-	40	200	0.4	106	25	144	27	155	-	-	25	144	-	0
...	-	-	-	-	-	-	-	-	-	-	-				
Total	60	1015	-	-	-	351	5002	357	5073	338	5171	370	5242	372	5340
Total cost ($ millions)	-	-	-	-	-	12.4		12.6		12.8		13		13.2	
Number of countries	-	49	-	-	-	13		13		12		12		12	
Realized PoS	-	-	-	-	-	50.3		60.0		70.4		80.4		90.6	

Table 4.4. Low cost country and site specific recruitment plans found by the HS algorithm under five different scenarios.

Country	Constants					Probability of success constraints									
	Min Sites $\bar{\mathbf{H}}_s$	Max Sites $\bar{\mathbf{U}}_s$	Max Patients $L(s)$	Recruitment Rate $m(s)$	Cost per Patient $\bar{\mathbf{c}}$	50%		60%		70%		80%		90%	
						Sites	Patients	Sites	Patients	Sites	Patients	Sites	Patients	Sites	Patients
Hungary	-	60	1100	1.4	121	59	1068	57	1056	60	1073	60	1073	59	1068
Sweden	-	50	600	0.8	162	32	432	43	547	47	572	43	547	49	580
Romania	-	50	300	0.5	135	42	253	44	262	46	270	42	253	46	270
Argentina	-	55	400	0.7	43	43	381	44	384	45	387	44	384	44	384
Colombia	-	50	500	1.4	182	34	412	33	403	40	459	39	453	44	478
Malaysia	-	40	200	0.3	112	-	-	17	68	31	124	22	88	40	158
Norway	-	40	200	0.4	106	26	150	34	182	34	182	32	176	33	179
Turkey	-	45	300	0.8	178	24	238	26	253	33	286	23	230	30	275
Lithuania	-	35	300	0.7	180	24	268	25	274	26	279	30	292	26	279
United States	25	230	3100	1.5	700	25	626	25	626	25	626	25	626	25	626
Latvia	-	40	300	0.5	190	23	197	8	69	-	-	28	236	19	163
China	15	85	1100	2.5	313	17	450	16	424	15	397	15	397	16	424
Iceland	-	35	200	1.5	106	12	193	12	193	12	193	12	193	12	193
Japan	10	105	600	0.4	900	10	69	10	69	10	69	10	69	10	69
United Kingdom	10	115	1600	0.9	633	10	155	10	155	10	155	10	155	10	155
India	-	55	400	0.8	236	-	-	-	-	2	22	2	22	3	32
Brazil	-	60	900	2.5	227	3	97	-	-	-	-	1	32	-	0
France	-	65	500	0.5	254	-	-	-	-	5	40	-	-	-	0
Poland	-	90	1100	1.7	269	-	-	3	82	-	-	-	-	-	0
South Africa	-	65	1100	1.1	324	1	12	1	12	-	-	-	-	-	0
Bulgaria	-	45	500	0.9	395	-	-	1	13	-	-	-	-	-	0
...	-	-	-	-	-	-	-	-	-	-	-				
Total	60	1015	-	-	-	385	5001	409	5072	441	5134	438	5226	466	5333
Total cost ($ millions)	-	-	-	-	-	12.5		12.7		12.8		12.9		13.2	
Number of countries	-	49	-	-	-	16		18		16		17		16	
Realized PoS	-	-	-	-	-	50.3		60.0		70.4		80.4		90.6	

Table 4.5. Low cost country and site specific recruitment plans found by the MFO algorithm under five different scenarios.

Country	Constants					Probability of success constraints									
	Min Sites $\bar{\mathbf{H}}_\mathbf{s}$	Max Sites $\bar{\mathbf{U}}_\mathbf{s}$	Max Patients $L(s)$	Recruitment Rate $m(s)$	Cost per Patient $\bar{\mathbf{c}}$	50%		60%		70%		80%		90%	
						Sites	Patients	Sites	Patients	Sites	Patients	Sites	Patients	Sites	Patients
Hungary	-	60	1100	1.4	121	60	1073	60	1073	60	1073	60	1073	60	1073
Romania	-	50	300	0.5	135	50	282	43	258	46	270	50	282	50	282
Colombia	-	50	500	1.4	182	50	493	40	459	41	465	42	470	50	493
Sweden	-	50	600	0.8	162	47	572	44	554	46	567	50	584	49	580
Argentina	-	55	400	0.7	43	46	389	55	399	55	399	55	399	48	393
Latvia	-	40	300	0.5	190	-	-	-	-	29	243	32	261	40	290
Norway	-	40	200	0.4	106	34	182	29	164	28	159	40	194	35	185
Turkey	-	45	300	0.8	178	29	270	23	230	27	259	18	185	31	279
Lithuania	-	35	300	0.7	180	26	279	26	279	23	262	26	279	30	292
United States	25	230	3100	1.5	700	25	626	25	626	25	626	25	626	25	626
China	15	85	1100	2.5	313	15	397	15	397	15	397	15	397	15	397
Iceland	-	35	200	1.5	106	13	196	11	190	12	193	11	190	13	196
Japan	10	105	600	0.4	900	10	69	10	69	10	69	10	69	10	69
United Kingdom	10	115	1600	0.9	633	10	155	10	155	10	155	10	155	10	155
Poland	-	90	1100	1.7	269	-	-	8	217	-	-	2	54	-	0
Taiwan	-	40	300	0.6	241	2	16	-	-	-	-	-	-	-	0
...	-	-	-	-	-	-	-	-	-	-	-				
Total	60	1015	-	-	-	417	4999	399	5070	427	5137	446	5218	466	5310
Total cost ($ millions)	-	-	-	-	-	12.4		12.6		12.7		12.9		13.1	
Number of countries	-	49	-	-	-	14		14		14		15		14	
Realized PoS	-	-	-	-	-	50.0		60.0		70.0		80.0		90.0	

4.4.2 The performance evaluation of other metrics

In this subsection, we detail the comprehensive simulation analysis of the selected meta-heuristic algorithms, focusing on their search capabilities and robustness. The analysis aims to provide insights into various critical metrics such as trial cost, geographical spread, site utilization, and computational efficiency derived from the 100 simulations performed by each algorithm in optimization problem (4.24) at the 90% PoS constraint.

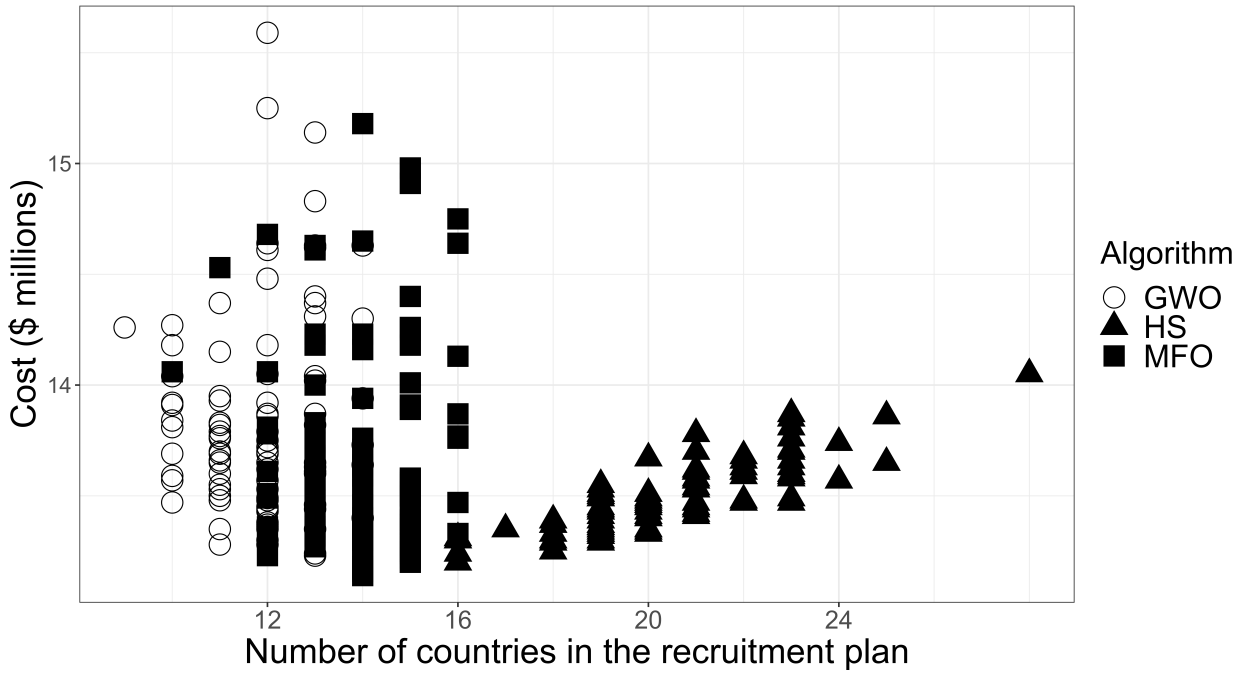
Table 4.6 displays the summary statistics for the 100 simulations of each algorithm, emphasizing key metrics like trial cost, number of countries and sites used, and computation speed. Notably, the MFO algorithm found the lowest cost solution at $13.10 million and the GWO algorithm had the highest cost at $15.60 million. Statistical analysis using the Kruskal-Wallis test revealed no significant differences in the cost medians between the algorithms (p-value = 0.20).

Table 4.6. Summary statistics of one hundred simulations of each algorithm when minimizing for total cost ($ millions).

	GWO		HS		MFO	
Metrics	Min - Max	Mean (SD)	Min - Max	Mean (SD)	Min - Max	Mean (SD)
Total cost ($ millions)	13.23 - 15.60	13.80 (0.462)	13.20 - 14.10	13.50 (0.160)	13.10 - 15.20	13.70 (0.470)
Number of countries	9 - 14	11.9 (1.14)	16 - 28	20.5 (2.14)	10 - 16	14.0 (1.26)
Number of sites	282 - 470	386 (41.8)	408 - 473	442 (12.0)	340 - 532	458 (40.3)
Computation speed	206 - 904s	415s (156)	242 - 1328s	657s (235)	27 - 89s	60s (11.2)

Interestingly, despite the minor cost differences among the algorithms, HS exhibited the most stable worst-case scenario, with costs ranging from $13.20 to $14.10 million. In terms of geographical spread, as shown in both Table 4.6 and Figure 4.6, the HS algorithm recommended the broadest global reach, using on average 20.5 countries. This contrasts with the more localized approaches of GWO and MFO, which averaged 11.9 and 14.0 countries, respectively. HS's plans consistently involved at least 16 countries, while GWO and MFO's

Figure 4.6. Global spread of 100 simulations for each algorithm when minimizing for total cost ($ millions).

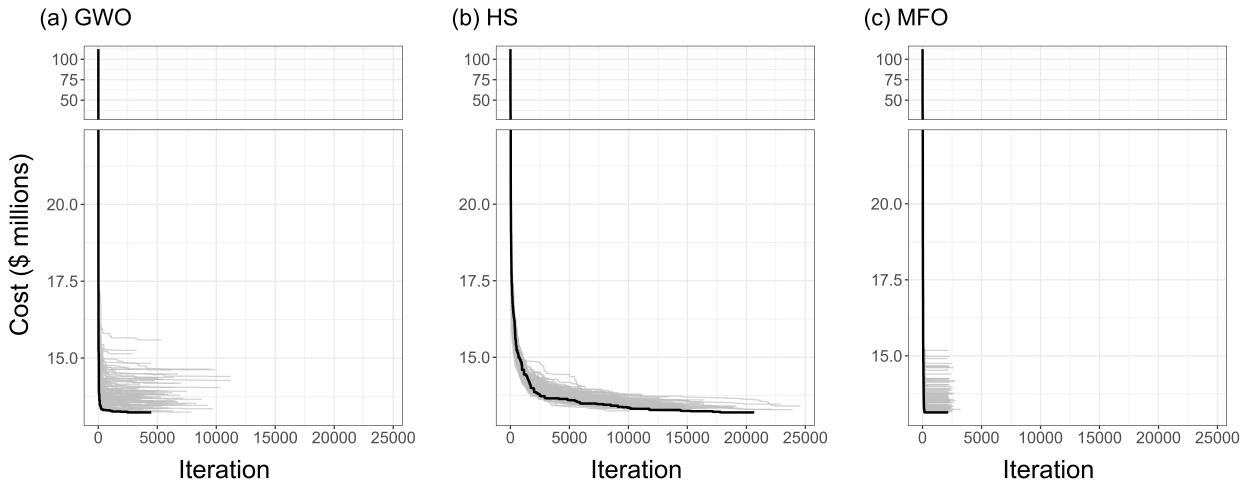


plans ranged from 9 to 14 and 10 to 16 countries, respectively, suggesting a narrower geographic focus. We observe the GWO is the sole algorithm to offer quality plans using 10 or less countries.

The computational speed of the algorithms also varied, with MFO demonstrating the fastest convergence, often completing simulations in under a minute (27 to 89 seconds). Figure 4.7 illustrates the convergence behavior of each algorithm, highlighting that the best solutions did not always correlate with the number of iterations. Only one out of 100 searches reached the global optima. We do not have access to the time between each iteration, only the total amount of time the search required, and provide Figure 4.8 to show the linear relationship between convergence iteration and computation time for each algorithm.

In summary, while GWO generally found the most cost-effective solutions, HS provided

Figure 4.7. Convergence analysis of each of the 100 simulations for each algorithm when minimizing for total cost ($ millions) with a 90% PoS constraint highlighting the optimal trajectory.

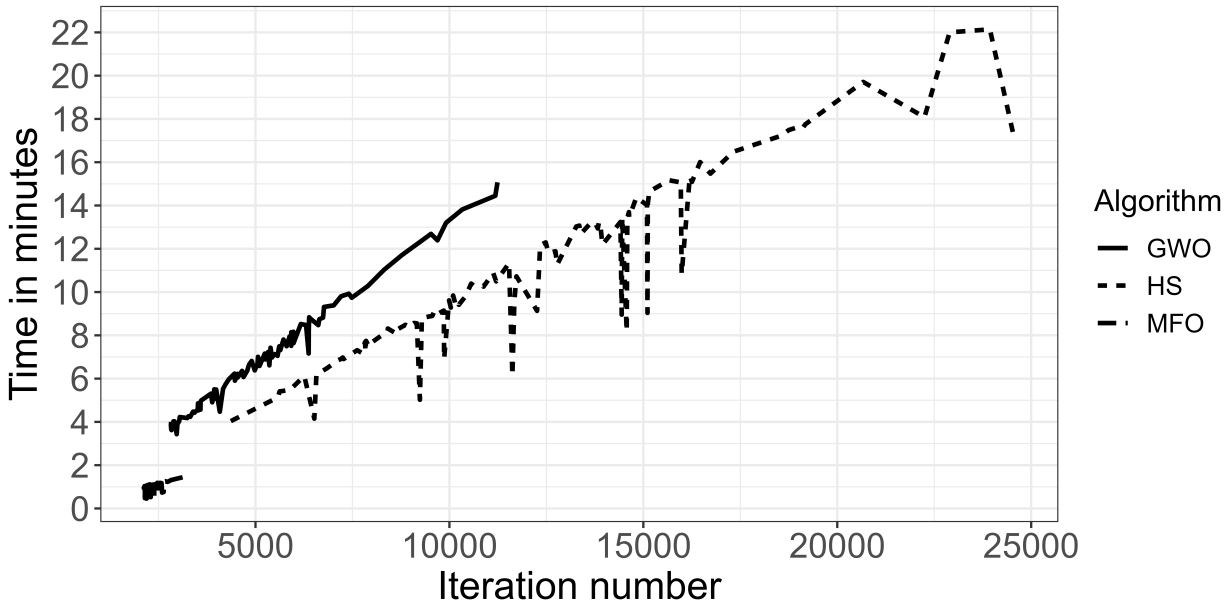


more diverse recruitment plans with a wider geographic reach, and MFO excelled in rapid computation. Given their respective strengths and the practical runtime of at most 23 minutes for each algorithm, we recommend using multiple simulations of all three algorithms for comprehensive decision-making. This approach offers decision-makers a diverse range of options to choose from, tailored to specific trial requirements.

4.4.3 Minimizing the global spread of a recruitment plan

As mentioned, the lowest cost recruitment plan at the 90% PoS cost $13.10 millions found by MFO, and may be the one to enact. This plan has the lowest total cost ($ millions) out of the 300 simulations and used 466 clinical trial sites. However, a similar cost plan with 90% PoS plan was found by GWO at $13.23 millions with 385 sites, and 81 fewer. This begs the question, which recruitment plan is the one to implement: which trade-off is worth it, 80 sites versus an extra of $0.13 millions. In this subsection, we analyze this trade-off further

Figure 4.8. The relationship between overall computed time in minutes versus the stop iteration.



and compare the number of sites, number of countries, and total cost ($ millions) using the three different optimization problems, Equations (4.24) - (4.26).

Figure 4.9 quantifies the total cost in millions of dollars on the vertical axis and enumerates the number of involved sites on the horizontal axis. The asterisk-marked solutions optimize (4.24), the square solutions optimize (4.25), and the open-circle solutions indicate an emphasis on minimizing the number of sites (4.26). When we seek to minimize the number of sites, plans were found with under 250 sites used, but they cost more compared to the asterisk shaped solutions ranging from $16 to $25 millions (Figure 4.9). If minimizing sites escalates costs, a decision must be made whether to choose the plan with less global roll out or favor a slightly increased number of sites for an overall total cost ($ millions) reduction. Similarly, Figure 4.10 quantifies the cost on the vertical axis and the number of countries on the horizontal axis. We observe that plans that minimize the total number of countries required cost significantly more, but there exists plans that could also use a minimal spread of only 4 countries. Consequently, metaheuristics provide several design options for the user.

Figure 4.9. Difference between optimizing for number of sites or total cost ($ millions).

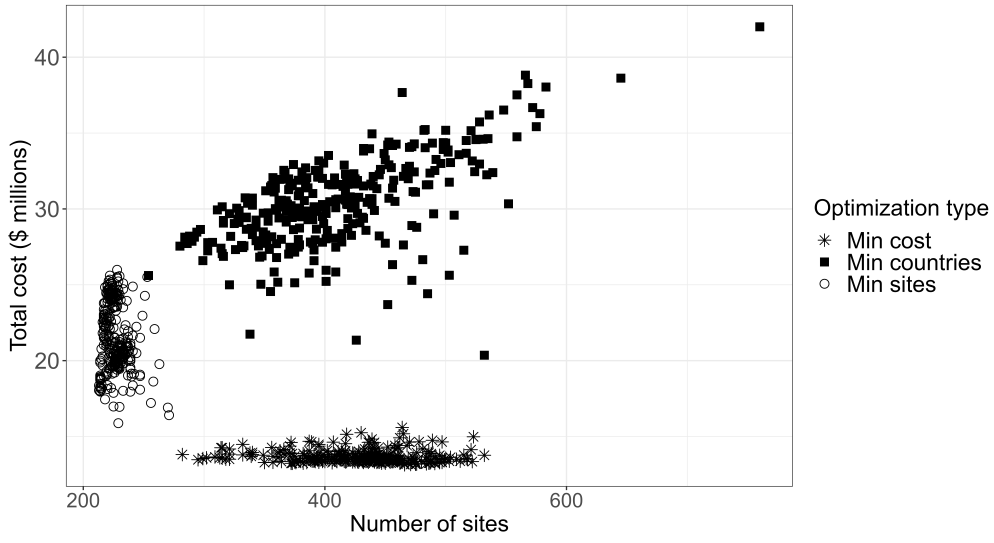
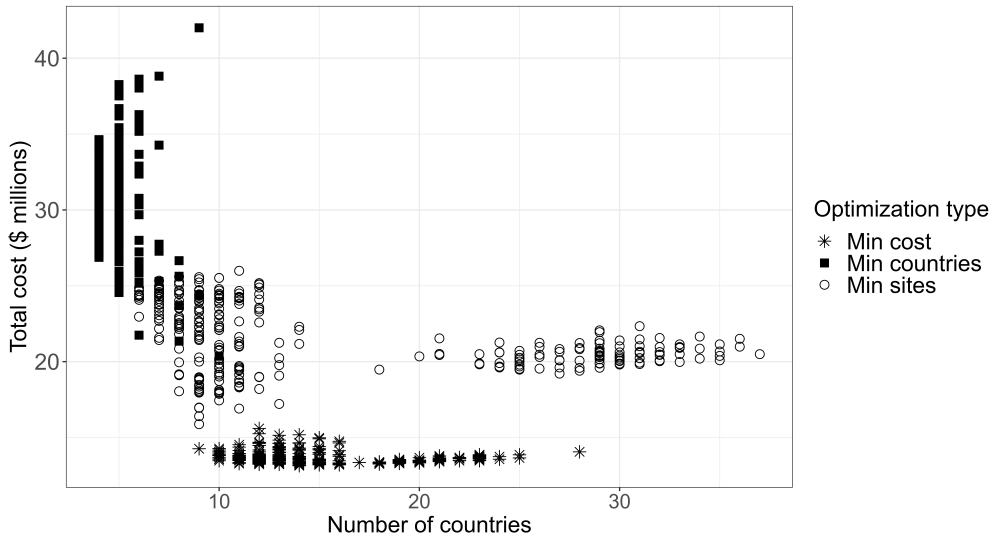


Figure 4.10. Difference between optimizing for number of countries ($ millions).



4.4.4 Clusters of solutions

Our results demonstrated that metaheuristics can tackle complex optimization problems and find optimal recruitment plans for the global clinical trial recruitment plan that meet the strict constraints, well under the budget; however, the optimal plans or designs can have noticeably different numbers of countries and varying numbers of sites within each country. Such a finding can be advantageous because it provides not one but several options to the pharmaceutical company or companies conducting the trials. This included presenting a cost-benefit analysis at a different PoS and also an analysis of the global spread of the solutions. Some of these recruitment plans were very similar, differing in only a few countries. In this subsection, we analyze clusters of solutions. This first involves determining the proper clustering method and then the optimal number of clusters. Once we have identified the optimal number of clusters using these methods, we can proceed to select the recruitment plan that has the lowest cost. This plan represents the most cost-effective strategy within each cluster. By presenting these selected plans from each cluster, the decision-making team

will have a diverse set of cost-effective options that cover different scenarios or strategies in the global recruitment context. This approach ensures a comprehensive and efficient exploration of the solution space, providing valuable insights for strategic planning.

The popular clustering method, k-means, is not suitable for our problem (Swarndeep Saket and Pandya 2016). This is because the mean of various recruitment plans does not work well. For instance, if one plan used 20 sites in China and 0 sites in Lithuania, and we compare it to a plan that used 0 sites in China and 20 sites in Lithuania, it is not guaranteed that a plan with 10 sites in China and 10 sites in Lithuania will still be an acceptable recruitment plan. Instead, we decide to use PAM which stands for partition around medoids and searches for the plans located in the center of different clusters (Reynolds et al. 2006, Schubert and Rousseeuw 2019). We use the version of PAM implemented in the `cluster` R package (Maechler et al. 2022).

Now that we know our clustering method, we have to make a decision on the optimal number of clusters. One effective approach is to utilize the silhouette method which assesses how similar each recruitment plan is to its own cluster compared to other clusters (Rousseeuw 1987). By calculating the silhouette score for a range of cluster numbers, we can identify the number of clusters that maximizes the average silhouette score, indicating a good balance between and within clusters (Shahapure and Nicholas 2020). We used the R package `factoextra` which calculates the silhouette scores using the PAM method for a range of total number of clusters (Kassambara and Mundt 2017). When applied to our data set, the silhouette method recommended using two clusters, which is good because it will hopefully give a clear distinction between clusters of recruitment plans and also should not overwhelm the clinical trial decision making team.

We present the lowest cost solution found from the two clusters in Table 4.7. The two lowest cost solutions from each cluster were from the GWO and MFO algorithms respectively. We observe many similarities between the plans, with the main difference being only 5 countries differed significantly. This may signify the overlapping countries between the

two clusters' best plans are vital to the recruitment plan. The non-overlapping countries of Latvia, Norway, Turkey, Lithiuana, and Poland come at a tradeoff. It appears that Poland can account for recruitment in all four of these countries, and may be worthwhile to use Cluster 1's plan despite the $200,000 increase in total cost. Table 4.8 showcases the separation of algorithms' solutions into different clusters. When we analyze two clusters, the HS algorithm only has solutions in one, and MFO mostly in one. The GWO algorithm provides solutions in both clusters, and may be another reason to use multiple algorithms during the search process. Similarly, the HS algorithm does not give solutions to many clusters like Cluster 4 in the second scenario and Clusters 5-8 in the third clustering scenario using 8 clusters (Figure 4.8).

4.5 Multiple objective optimal global designs

In practice, clinical trials likely have several objectives, and there may not be equally important. In this section, we discuss how to design a global clinical trial a trial when there are multiple objectives in the study. We consider, as examples, three primary objectives: (1) minimize the total cost ($ millions) of the clinical trial, (2) minimize the number of countries involved, and (3) minimize the number of clinical trial sites. This multiple objective minimization problem can be formulated as in Equation 4.28 below. If objective 2, the number of countries is to be maximized rather than minimized, it can be represented as $f_2 \times -1$. In this section, we will compare different computational methods to construct Pareto fronts and use metrics, like hypervolume to select and implement a multiple objective metaheuristic algorithm to find an optimal clinical trial recruitment plan.

$$
\begin{aligned}
&\text{minimize:} && (f_1, f_2, f_3) && && (4.28)\\
&\text{subject to:} && P(n, T, \bar{N}) \geq 90\% && \text{PoS`} &&\\
& && H_s \leq N_s \leq U_s && s = 1, 2, .., S &&\\
& && n(I_s, t) \leq L(s) && s = 1, 2, .., S &&\\
&\text{where:} && f_1(N_s) = C(T, \bar{N}, \bar{C}, \bar{c}, \bar{Q}) && \text{Total cost} &&\\
& && f_2(N_s) = \Sigma_{s=1}^{S}(N_s > 0) && \text{Number of countries} &&\\
& && f_3(N_s) = \Sigma_{s=1}^{S}(N_s) && \text{Number of sites} &&\\
& && && && (4.29)
\end{aligned}
$$

4.5.1 Weighted sum

Our initial exploration to create the Pareto front is with the weighted sum method. We transform the multiple objective problem into a single objective by assigning a weight to each objective function and optimizing for the sum of these weighted objectives. We then use the same single objective metaheuristics we used in Section 4.2.5. The weights assigned to the objectives are $w_1, w_2,$ and w_3, respectively, and they represent the relative importance of each of the objectives in the optimization problem. For example, $w_1 = 50\%$ for preference in total cost ($ millions), $w_2 = 40\%$ for preference in minimizing the number of countries, and $w_3 = 10\%$ for preference in minimizing the number of sites. By systematically varying the weights for the different objectives, we were able to explore a wide range of solutions corresponding to a Pareto front for the multiple objective optimization problem effectively using the hypervolume values.

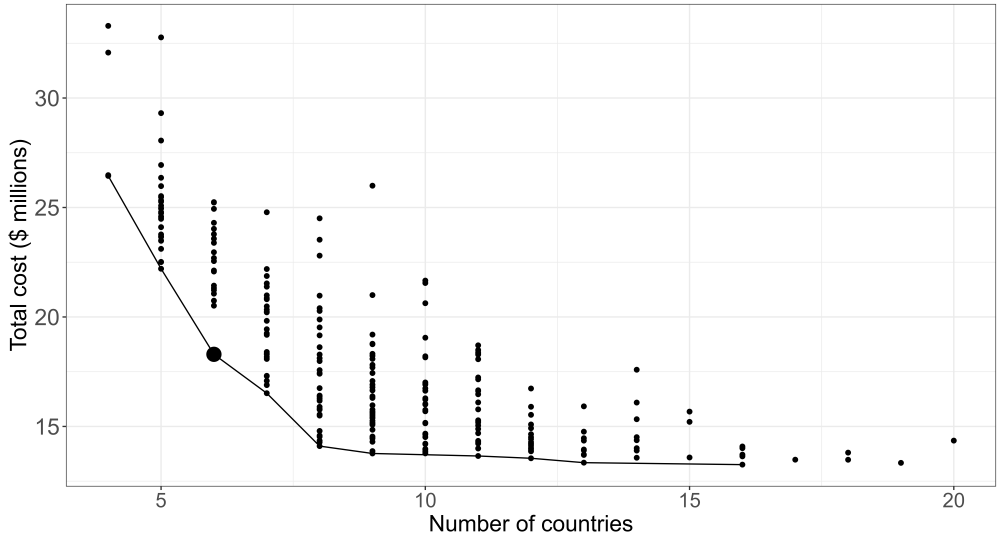
First, we considered the two dimensional weighted sum optimization problem, (4.30) optimizing for f_1, total cost ($ millions), and f_2, number of countries, only. We operationalized this method across a spectrum of fifty different weight combinations, ranging from 0 to 1

in increments of 0.02. This narrow sampling allowed us to obtain a birds eye view of the weighted sum approach and its consequent impact on the Pareto front. Subsequent single-objective optimizations were performed to procure the optimal solutions for each weight set. Ultimately this created a set of solutions from which we can derive the Pareto front represented by the connected line in Figure 4.12. The solution set demonstrated a notable level of effectiveness, as indicated by its normalized hypervolume value of 0.925. This high score suggests comprehensive coverage of the solution space and is evidenced by our previous analyses. The value with the largest individual hypervolume had a value of 0.845 and represented a design costing $18.3 millions using 6 countries and 281 sites, signified by the larger sized point in Figure 4.30.

$$
\begin{aligned}
&\text{minimize:} && w_1 * f_1 + w_2 * f_2 && && (4.30)\\
&\text{subject to:} && P(n, T, \bar{N}) \geq 90\% && \text{PoS`}\\
& && H_s \leq N_s \leq U_s && s = 1, 2, .., S\\
& && n(I_s, t) \leq L(s) && s = 1, 2, .., S\\
&\text{where:} && f_1(N_s) = C(T, \bar{N}, \bar{C}, \bar{c}, \bar{Q}) && \text{Total cost}\\
& && f_2(N_s) = \Sigma_{s=1}^{S}(N_s > 0) && \text{Number of countries}
\end{aligned}
$$

We then ran the procedure for three dimensions and solved the weighted sum problem in (4.31, which) now additionally include the number of sites, f_3. This time, we used a step size of 0.1 between weight combinations for a total of 121 combinations. The solution set demonstrated a notable level of effectiveness, as indicated by its normalized hypervolume value of 0.871. The decrease in the hypervolume value when moving from two to three dimensions can be attributed to the increased complexity of the solution space. With an additional objective (number of sites), the optimization problem becomes more challenging, and it's harder to find solutions that perform exceptionally well across all objectives. This large normalized hypervolume still suggests comprehensive coverage of the solution space.

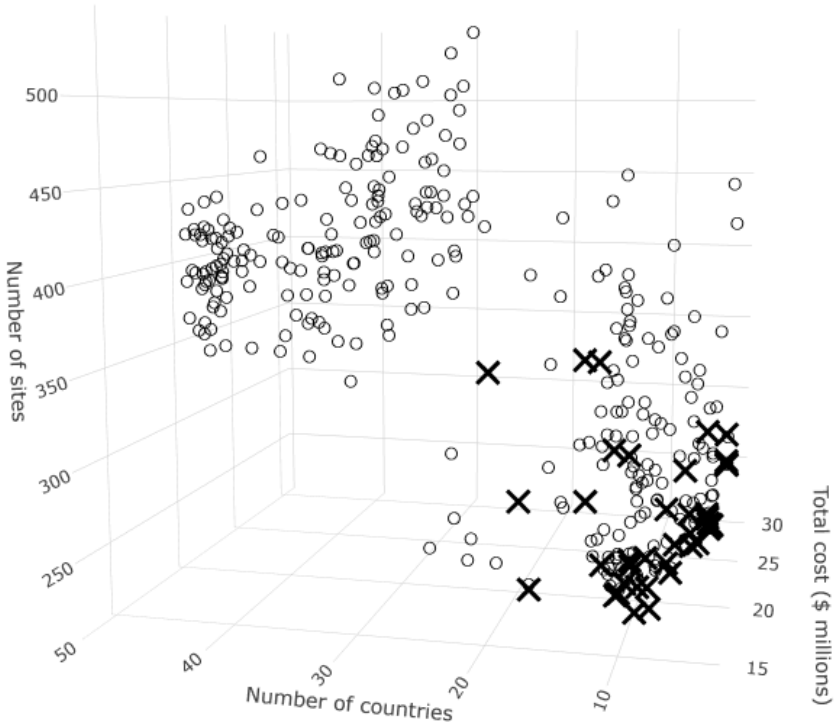
Figure 4.11. Weighted sum results to minimize the total cost ($ millions) and number of countries.



The value with the largest individual hypervolume had a value of 0.781 and represented a design that would cost $16.8 millions, used 7 countries, and used 250 sites.

$$
\begin{aligned}
&\text{minimize:} && w_1 * f_1 + w_2 * f_2 + w_3 * f_3 && (4.31)\\
&\text{subject to:} && P(n, T, \bar{N}) \geq 90\% && \text{PoS‘}\\
& && H_s \leq N_s \leq U_s && s = 1, 2, .., S\\
& && n(I_s, t) \leq L(s) && s = 1, 2, .., S\\
&\text{where:} && f_1(N_s) = C(T, \bar{N}, \bar{C}, \bar{c}, \bar{Q}) && \text{Total cost}\\
& && f_2(N_s) = \Sigma_{s=1}^{S}(N_s > 0) && \text{Number of countries}\\
& && f_3(N_s) = \Sigma_{s=1}^{S}(N_s) && \text{Number of sites}
\end{aligned}
$$

Figure 4.12. Weighted sum results to minimize three objectives: total cost ($ millions), number of countries, and number of sites.



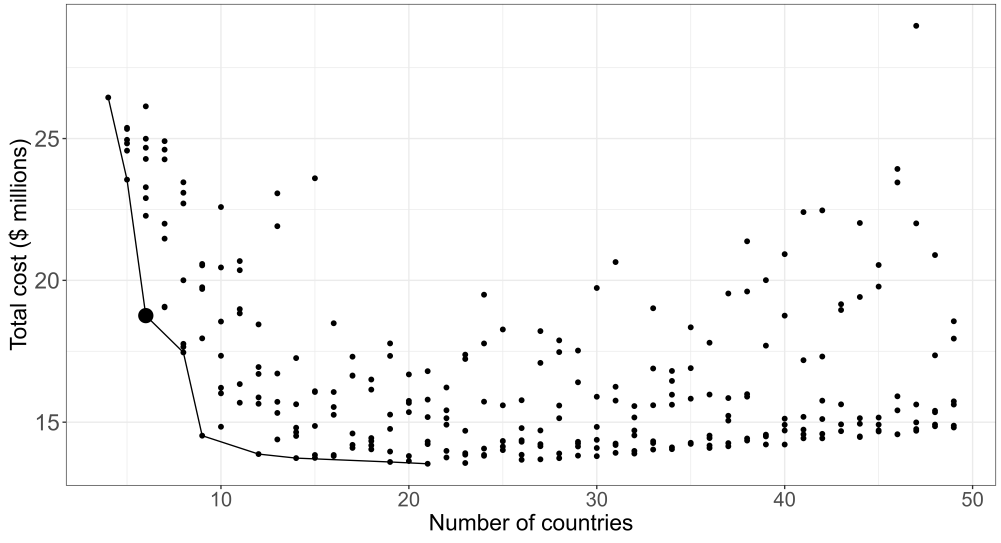
4.5.2 Epsilon constraint

We next used the epsilon constraint method to create a Pareto front. This method minimizes one objective while constraining the rest. Again, we first considered two objectives, minimize f_1, total cost ($ millions) and constrain only f_2, the number of countries. The metaheuristic search process begins at the upper bound using all 49 countries. When minimizing for f_1, we first implemented a strict constraint on the number of countries. We attempted to search for the optimal cost plan for each individual number of countries i.e. $f_2 = s$, s = 4,5,.., 49, totaling 45 separate optimization functions. We first implemented the death penalty for a

constraint violation at a value of $1e24$, much higher than either the total cost ($ millions) and number of countries, so the penalty has no affect on the search other than rendering a solution useless and not worthwhile. With this implementation, the metaheuristics had difficulty finding a feasible solution. For this reason, we constructed an adaptive penalty as discussed in Section 2.3.1 to guide the recruitment plan towards the desired number of countries. The infeasible recruitment plans are penalized according to the absolute value of the difference between the countries currently in use and the specific constraint. We used a penalty of magnitudes smaller than the death penalty, but still larger than the objective values, $1e18$, multiplied by the absolute difference. For instance, say we are constraining all designs to be for 10 countries. The infeasible recruitment plan which uses 12 countries will be penalized $2 \times 1e15$ and the infeasible recruitment plan which uses 18 countries will be penalized $8 \times 1e15$. Doing so guide to a solution with 10 countries. The adaptive penalty is smaller than the death penalty, and the objective function is still able to be guided towards feasible solutions. Using this method to optimize (4.32), the overall hypervolume from the Pareto front seen in Figure 4.13 is 0.925, signifying again good coverage. The maximum individual normalized hypervolume had a value of 0.745 and was very similar to the maximum individual in the optimization for the two objective problem constructed by the weighted sum and used $18.3 millions and 6 countries and 289 sites.

$$
\begin{array}{lll}
\text{minimize:} & f_1 & \\
\text{subject to:} & P(n,T,\bar{N}) \geq 90\% & \text{PoS‘} \\
 & f_2 = s & s = 4,5,...,49 \\
 & f_3 \leq \epsilon_2 & \\
 & H_s \leq N_s \leq U_s & s = 1,2,..,S \\
 & n(I_s,t) \leq L(s) & s = 1,2,..,S \\
\text{where:} & f_1(N_s) = C(T,\bar{N},\bar{C},\bar{c},\bar{Q}) & \text{Total cost} \\
 & f_2(N_s) = \Sigma_{s=1}^{S}(N_s > 0) & \text{Number of countries} \\
 & f_3(N_s) = \Sigma_{s=1}^{S}(N_s) & \text{Number of sites}
\end{array}
\tag{4.32}
$$

Figure 4.13. Epsilon constrained 2d with an adaptive penalty to search for the optimal recruitment plan for the exact number of countries. ($ millions).



We set up a similar epsilon constrained method for three dimensions seen in (4.33). Here, we relaxed the restriction on f_2, the number of countries, to be less than 9, and looked for

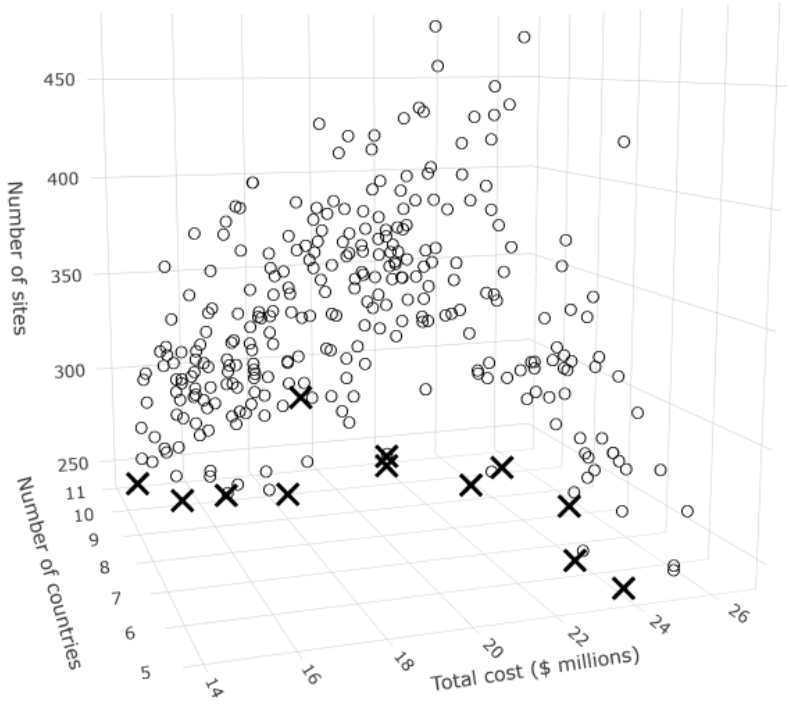
solutions with under 275 sites. We applied the same penalty to f_2, $1e18$, and then a penalty of smaller magnitude for any violation of f_3, $1e15$, again multiplied by the absolute difference between the value currently used and the specified constraint. Figure 4.14 shows the values found from this search and the three dimensional Pareto front marked by the X's. Many solutions violated the constraint. The normalized hypervolume found by the Pareto front here was 0.84 with a maximum individual value of 0.774 which represented a plan which used \$14.8 millions, 8 countries, and 264 sites.

$$
\begin{array}{rll}
\text{minimize:} & f_1 & \\
\text{subject to:} & P(n, T, \bar{N}) \geq 90\% & \text{PoS`} \\
& f_2 \leq 9 & \\
& f_3 \leq 275 & \\
& H_s \leq N_s \leq U_s & s = 1, 2, .., S \\
& n(I_s, t) \leq L(s) & s = 1, 2, .., S \\
\text{where:} & f_1(N_s) = C(T, \bar{N}, \bar{C}, \bar{c}, \bar{Q}) & \text{Total cost} \\
& f_2(N_s) = \Sigma_{s=1}^{S}(N_s > 0) & \text{Number of countries} \\
& f_3(N_s) = \Sigma_{s=1}^{S}(N_s) & \text{Number of sites}
\end{array}
\tag{4.33}
$$

4.5.3 Multiple objective metaheuristics

We then used the popular NSGA-II algorithm implemented in the `mco` R package to optimize the original multiple objective optimization problem, (4.28) and also its two-dimensional equivalent only minimizing total cost (\$ millions) and the number of countries. We acknowledge, just like the single objective optimization problem, there are many other multiple objective R packages which can have varying results and require further testing including `caRamel`, `ecr`, `mopsocd`, and `rmoo`.

Figure 4.14. Epsilon constrained 3d with an adaptive penalty to search for the optimal recruitment plan for the exact number of countries. ($ millions).

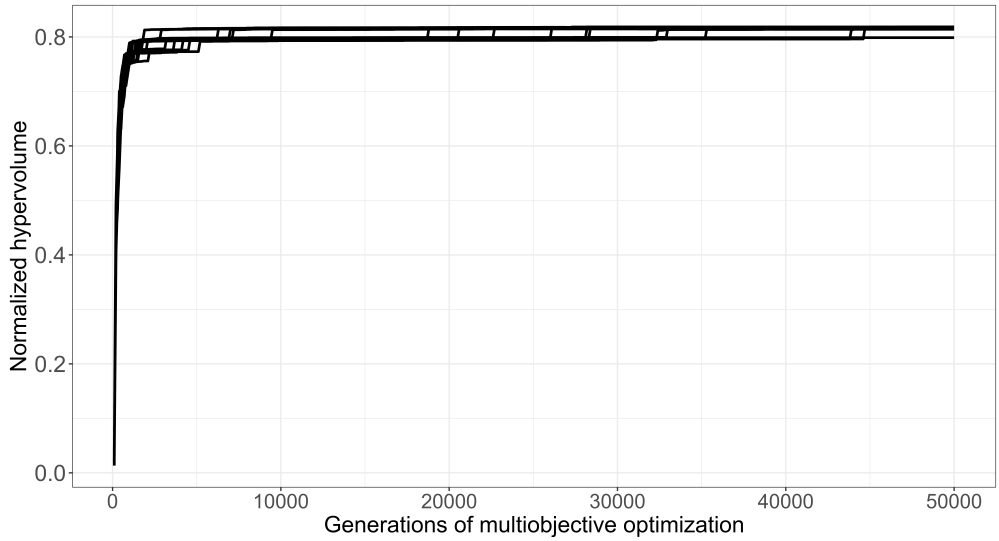


4.5.3.1 Convergence properties

Similar to section 4.3, we preliminarily analyze the convergence stagnation properties in order to make sure we run the metaheuristics in an optimal way without using worthless iterations and extra computation time. The convergence stagnation analysis enables us to decide when an algorithm is finished or near finished. In single objective optimization, we analyzed the iterative improvement in the single outcome and quantified it by percentiles of the length of stagnation. Rather in multiple objective optimization, we can measure the generative

improvement in hypervolume between each generation's Pareto front. The `mco` R package has an input for the `mco::nsga2()` function that allows storage of previous populations, e.g. setting `generations` to `1:50` will save each generation's population. In Figure 4.15, we analyzed the generational improvement of every 100 generations over 50,000 computed from 20 parallel simulations optimizing the three-dimensional problem at otherwise default settings in the `mco` R package. Each simulation took around 15 minutes and quickly reached a near peak hypervolume before the $10{,}000^{th}$ generation. We do still see improvement up to the $45{,}000^{th}$ generation, and the computation time is not infeasible, so we elect to perform 100 simulations at 50,000 generations.

Figure 4.15. Multiple objective results for the Pareto front and history of 50,000 generations for the three-dimensional problem of the `nsga2` in the `mco` R package.

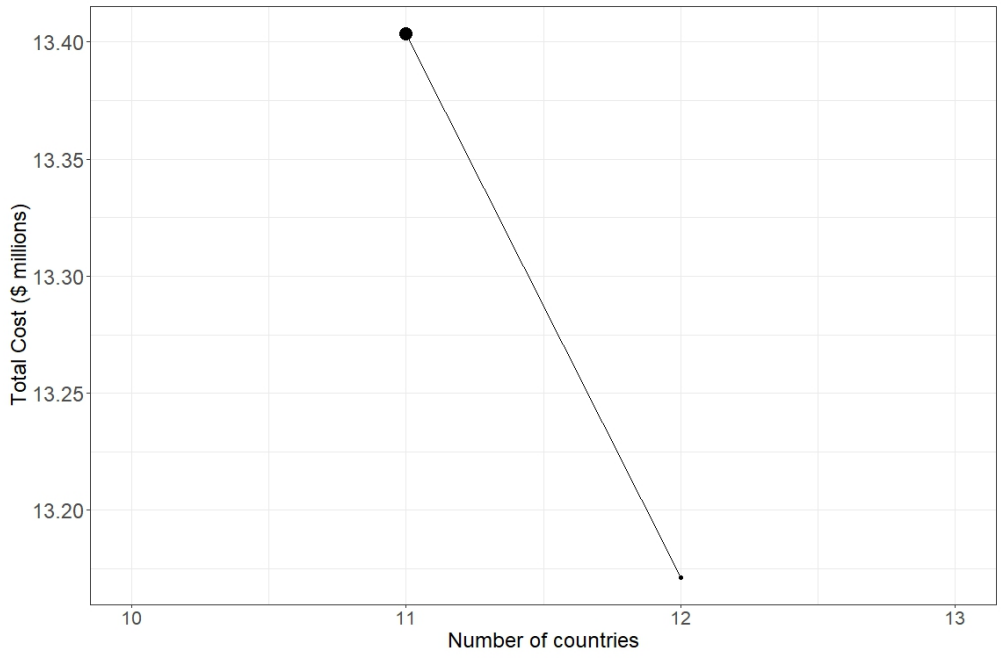


4.5.3.2 Pareto front results and history

We present in Figure 4.16 the results from the two-dimensional optimization. Interestingly, the population converged to only two solutions, namely 11 countries costing $13.4 millions

using 377 sites and a solution using 12 countries costing $13.2 million and 365 sites. This has a hypervolume of 0.788, less than the previous two methods. Further, we present in Figure 4.17, the historical trajectory of the NSGA-II search from the `mco` package. We plot the solutions from generation 1 to 50,000, and the Pareto front is represented by the orange dots. The normalized hypervolume of this Pareto front was 0.816. The individual with the largest normalized hypervolume had a value of 0.788 and was the plan which spent $16.0 millions, used 7 countries and 246 sites.

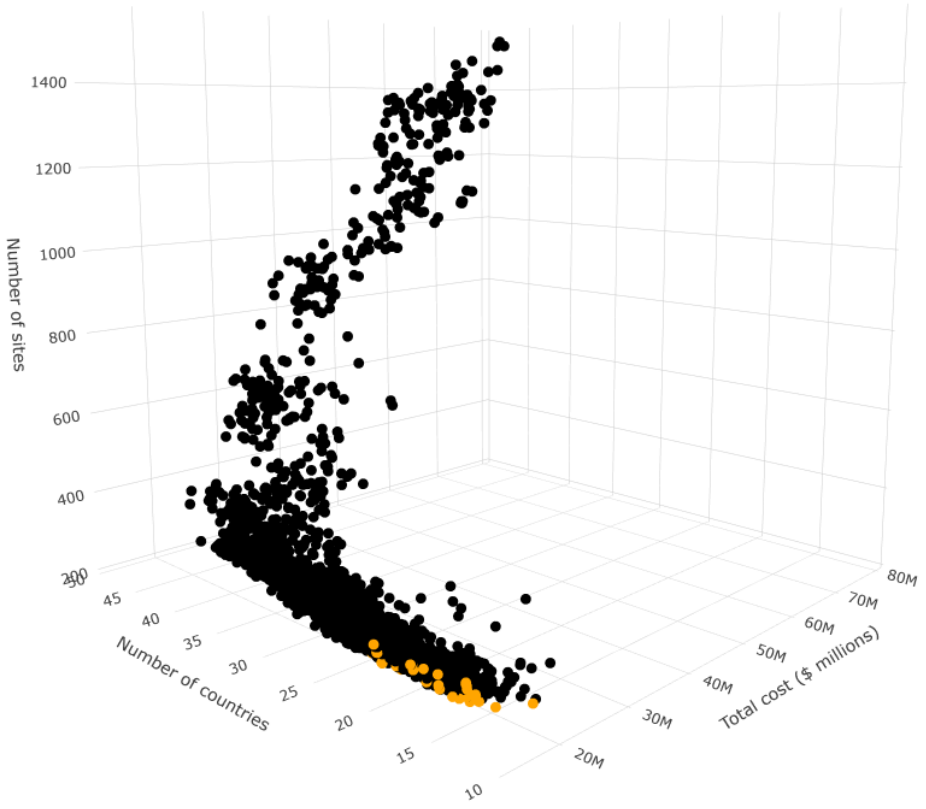
Figure 4.16. Two-dimensional multiple objective results for the Pareto front after 50,000 generations for the two-dimensional problem of the `nsga2` in the `mco` R package.



4.5.4 Final Pareto front

So far, in this section, we have used three methods to construct two- and three-dimensional Pareto fronts. Each gave normalized hypervolumes close to 1, signifying good coverage of

Figure 4.17. Multiple objective results for the Pareto front and history of 50,000 generations for the three-dimensional problem of the `nsga2` in the `mco` R package.



the solution space, and each gave different results when looking at the point with the largest hypervolume. We close by mentioning how we have presented many different methods to create a Pareto front. The tools we have used to find optimal results are robust to more than three objectives, like the stagnation calculations, Pareto front, and the hypervolume. To select the "optimal point" one ultimately has to make a decision. We combine each result created from the multiple objective optimization to perform a complete Pareto front analysis and include all of our single objective optimization results too. This resulted in 2,222 unique solutions out of 2,379. The overall hypervolume of the combined Pareto front is 0.873, larger

than any of the individual methods. The individual with the largest normalized hypervolume was the best from the `mco` multiple objective optimization. However, we have shown that this does not necessarily make it the best solution to implement. Other factors must be taken into account which real life decision makers can benefit from using the robust amount of recruitment plans created by the population-based metaheuristics. The usage of these algorithms are powerful for interpretability and require a skilled user behind them to extract the best data available. Multiple objective optimization is increasingly growing with many areas for future research (Coello Coello et al. 2020). We recognize the solutions created in this chapter are currently used to design worldwide clinical trial designs with multiple objectives and metaheuristics using the PG model for patient recruitment.

Table 4.7. Recruitment plans for separate clusters found by partitioning around medoids.

Country	Constants					Clusters			
	Min Sites $\bar{\mathbf{H}}$	Max Sites $\bar{\mathbf{U}}$	Max Patients $L(s)$	Recruitment Rate $m(s)$	Cost per Patient $\bar{\mathbf{c}}$	Cluster 1		Cluster 2	
						Sites	Patients	Sites	Patients
Hungary	0	60	1100	1.4	121	60	1073	60	1073
Romania	0	50	300	0.5	135	50	282	50	282
Colombia	0	50	500	1.4	182	39	453	50	493
Sweden	0	50	600	0.8	162	50	584	49	580
Argentina	0	55	400	0.7	43	52	397	48	393
United States	25	230	3100	1.5	700	25	626	25	626
China	15	85	1100	2.5	313	15	397	15	397
Iceland	0	35	200	1.5	106	12	193	13	196
Japan	10	105	600	0.4	900	10	69	10	69
United Kingdom	10	115	1600	0.9	633	10	155	10	155
Latvia	0	40	300	0.5	190	-	-	40	290
Norway	0	40	200	0.4	106	-	-	35	185
Turkey	0	45	300	0.8	178	-	-	31	279
Lithuania	0	35	300	0.7	180	-	-	30	292
Poland	0	90	1100	1.7	269	53	1093	-	-
...	-	-	-	-	-	-	-	-	-
-	-								
Total	60	1015	-	-	-	376	5322	466	5310
Total cost ($ millions)	-	-	-	-	-	13.3		13.1	
Number of countries	4	49	-	-	-	11		14	
Realized PoS	-	-	-	-	-	90.6		90.0	
Algorithm	-	-	-	-	-	GWO		MFO	

Table 4.8. Separation of metaheuristic algorithms into 2, 4, and 8 recruitment plan clusters.

	GWO	HS	MFO
Cluster 1	62	100	92
Cluster 2	38	0	8

	GWO	HS	MFO
Cluster 1	33	12	26
Cluster 2	28	54	22
Cluster 3	6	34	40
Cluster 4	33	0	12

	GWO	HS	MFO
Cluster 1	10	10	17
Cluster 2	10	50	16
Cluster 3	6	28	37
Cluster 4	16	12	20
Cluster 5	24	0	4
Cluster 6	17	0	2
Cluster 7	9	0	8
Cluster 8	8	0	6

CHAPTER 5

Conclusions

In this dissertation, we proposed metaheuristics to solve large scale complex optimization problems in clinical trials. We demonstrated the flexible techniques using two real trials. The first was to find different types of optimal sampling times for a longitudinal PKPD study under various user-specified constraints to ascertain effectiveness of lithium to treat bipolar patients. The second application was to develop efficient recruitment multiple-objective plans for a worldwide clinical trial subject to a set of different types of user-specified constraints. The algorithms were able to solve such complex problems with two-, three-, and four objectives using a robust metric. We also showed a method to determine the time and amount of stagnation in the algorithms when solving a single and multiple objective problem. We provided justifications for the convergence properties of the algorithms, so time can be implemented as a user-defined limit when computation is performed.

The next steps are to integrate this knowledge into software and explore other statistical models and optimization methods. PFIM has recently included GA and PSO into the R software package, but it still lacks the ability for finding D_s-optimal and multi-objective optimal designs (Mentré et al. 2023). The current recruitment planning software package uses certain performance metrics from this work and functions from `metaheuristicOpt`. In practice, how metaheuristics are implemented depends on the computing power allotted. In turn, the computing time determines the length of the search time and the number of simulations, before the decision on which algorithm(s) to use is made.

There are a few closing remarks and cautionary notes. First, there are many metaheuristic

algorithms and clearly we did not even attempt to use and compare performances with many of them. This implies that there is no guarantee the algorithms and the implementations we used offered the best solutions or computation time. However, they did provide solutions that meet or nearly meet all the constraints. The best algorithm should be the one that has been well tested for solving many complex optimization problems and came up consistently top or near the top.

Second, this work is developed for large clinical trials, and it is unclear if and whether the results will hold up for other types of trials or trials with small sample sizes. However, because metaheuristic algorithms are fast, such questions can be answered via simulations. Running a single metaheuristic algorithm repeatedly to ascertain the optimum can be risky business since different metaheuristic algorithms can produce different optimum. Hence several different types of algorithms should be run to ensure the optimum found by each algorithm are about the same. We also remind the reader that the tuning parameters in various metaheuristic algorithms can usually be tuned for better performance if the algorithm is not performing well. The `irace` R package has a focus on automatic tuning of parameters in certain metaheuristics. irace can be used to automatically create metaheuristics by using particular elements from various ones (Bezerra et al. 2015). The potential of these advanced approaches is substantial and warrants further study.

In conclusion, our study has provided substantial evidence supporting the use of metaheuristic algorithms for tackling complex real world optimization problems. Even though we used metaheuristics for designing global trials and pkpd studies, they are clearly not limited to such applications. They also have been used to solve other statistical problems, like estimating model parameters or best subset variables selection. Metaheuristics are now widely used in medical diagnosis, predicting disease progression, and in the manufacture industry, with no end in sight. We hope that our work will stimulate greater interest in applying nature-inspired metaheuristics to solving challenging problems in public health and the world herein is a useful starting point for further research in metaheuristic and how they

can be usefully applied to solve challenging optimization problems in biostatistics.

Bibliography

Abdel-Basset, M., L. Abdel-Fatah, and A. K. Sangaiah (2018). Metaheuristic algorithms: A comprehensive review. *Computational intelligence for multimedia big data on the cloud with engineering applications*, 185–231.

Abualigah, L. and A. Diabat (2021). Advances in sine cosine algorithm: a comprehensive survey. *Artificial Intelligence Review 54*(4), 2567–2608.

Abualigah, L., E. S. Hanandeh, R. A. Zitar, C. L. Thanh, S. Khatir, and A. H. Gandomi (2023). Revolutionizing sustainable supply chain management: A review of metaheuristics. *Engineering Applications of Artificial Intelligence 126*, 106839.

Agrawal, P., H. F. Abutarboush, T. Ganesh, and A. W. Mohamed (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *IEEE Access 9*, 26766–26791.

Akay, B., D. Karaboga, and R. Akay (2022). A comprehensive survey on optimizing deep learning models by metaheuristics. *Artificial Intelligence Review*, 1–66.

Alam, M. N. (2016). Particle swarm optimization: Algorithm and its codes in matlab. *ResearchGate 8*(1), 10.

Ali, A. F. and A. E. Hassanien (2015). A survey of metaheuristics methods for bioinformatics applications. In *Applications of Intelligent Optimization in Biology and Medicine: Current Trends and Open Problems*, pp. 23–46. Springer.

Almufti, S. M. (2019). Historical survey on metaheuristics algorithms. *International Journal of Scientific World 7*(1), 1.

American Psychiatric Association, D., A. P. Association, et al. (2013). *Diagnostic and statistical manual of mental disorders: DSM-5*, Volume 5. American psychiatric association Washington, DC.

Anisimov, V. (2011). Statistical modeling of clinical trials (recruitment and randomization). *Communications in Statistics - Theory and Methods 40*(19-20), 3684–3699.

Anisimov, V. (2016). Discussion on the paper Real-time prediction of clinical trial enrollment

and event counts: a review, by D.F. Heitjan, Z. Ge, and G.S. Ying. *Contemporary Clinical Trials 46*, 7–10.

Anisimov, V. (2020). Modern analytic techniques for predictive modeling of clinical trial operations. In O. V. Marchenko and N. V. Katenka (Eds.), *Quantitative Methods in Pharmaceutical Research and Development*, pp. 361–408. Springer.

Anisimov, V. and M. Austin (2020). Centralized statistical monitoring of clinical trial enrollment performance. *Communications in Statistics: Case Studies, Data Analysis and Applications 6*(4), 392–410.

Anisimov, V. and M. Austin (2022). Modeling restricted enrollment and optimal cost-efficient design in multicenter clinical trials. *ArXiv*.

Anisimov, V. and M. Austin (2023). Forecasting and optimizing patient enrolment in clinical trials under various restrictions. In S. Silvestrov, A. Malyarenko, and M. Rancic (Eds.), *Stochastic Processes, Statistical Methods, and Engineering Mathematics*, Springer Proceedings in Mathematics & Statistics 408, pp. 511–540. Springer.

Anisimov, V., D. Downing, and V. Fedorov (2007). Recruitment in multicentre trials: prediction and adjustment. In *mODa 8–Advances in Model-Oriented Design and Analysis*, pp. 1–8. Physica-Verlag.

Anisimov, V. and V. Fedorov (2007a). Modeling, prediction and adaptive adjustment of recruitment in multicentre trials. *Statistics in Medicine 26*(27), 4958–4975.

Anisimov, V. and V. Fedorov (2007b). Design of multicentre clinical trials with random enrolment. In J. L. Auget, N. Balakrishnan, M. Mesbah, and G. Molenberghs (Eds.), *Advances in Statistical Methods for the Health Sciences, Series: Statistics for Industry and Technology*, pp. 387–400. Birkhäuser.

Anisimov, V., S. Gormley, R. Baverstock, and C. Kineza (2021). Advanced models for predicting event occurrence in event-driven clinical trials accounting for patient dropout, cure and ongoing recruitment. *ArXiv*.

Anisimov, V., S. Gormley, R. Baverstock, and C. Kineza (2022). Predicting event counts in event-driven clinical trials accounting for cure and ongoing recruitment. In K. N. Zafeiris, C. H.

Skiadas, Y. Dimotikalis, A. Karagrigoriou, and C. Karagrigroriou-Vonta (Eds.), *Data Analysis and Related Applications 2. Multivariate, Health and Demographic Data Analysis*, Volume 10 - Big Data, Artificial Intelligence and Data Analysis, pp. 121–142. Wiley & ISTE.

Atkinson, A., A. Donev, and R. Tobias (2007). *Optimum experimental designs, with SAS*, Volume 34. Oxford University Press.

Auger, A., J. Bader, D. Brockhoff, and E. Zitzler (2012). Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science 425*, 75–103.

Baden, L. R., H. M. El Sahly, B. Essink, K. Kotloff, S. Frey, R. Novak, D. Diemert, S. A. Spector, N. Rouphael, C. B. Creech, J. McGettigan, S. Khetan, N. Segall, J. Solis, A. Brosz, C. Fierro, H. Schwartz, K. Neuzil, L. Corey, P. Gilbert, H. Janes, D. Follmann, M. Marovich, J. Mascola, L. Polakowski, J. Ledgerwood, B. S. Graham, H. Bennett, R. Pajon, C. Knightly, B. Leav, W. Deng, H. Zhou, S. Han, M. Ivarsson, J. Miller, and T. Zaks (2021). Efficacy and safety of the mRNA-1273 SARS-CoV-2 vaccine. *New England Journal of Medicine 384*(5), 403–416.

Barnard, K. D., L. Dent, and A. Cook (2010). A systematic review of models to predict recruitment to multicentre clinical trials. *BMC Medical Research Methodology 10*(1), 63.

Barrow, J. C. and C. W. Lindsley (2023). The importance of PK-PD. *Journal of Medicinal Chemistry 66*(7), 4273–4274.

Bauer, R. J., A. C. Hooker, and F. Mentré (2021). Tutorial for design in NONMEM: clinical trial evaluation and optimization. *CPT: Pharmacometrics & Systems Pharmacology 10*(12), 1452–1465.

Bazzoli, C., S. Retout, and F. Mentré (2009). Fisher information matrix for nonlinear mixed effects multiple response models: evaluation of the appropriateness of the first order linearization using a pharmacokinetic/pharmacodynamic model. *Statistics in Medicine 28*(14), 1940–1956.

Becherer, N., J. Pecarina, S. Nykl, and K. Hopkinson (2019). Improving optimization of convolutional neural networks through parameter fine-tuning. *Neural Computing and Applications 31*(8), 3469–3479.

Behnamian, J. and S. F. Ghomi (2010). Development of a PSO-SA hybrid metaheuristic for a

new comprehensive regression model to time-series forecasting. *Expert Systems with applications 37*(2), 974–984.

Bernardo, J. M. and A. F. Smith (2004). *Bayesian Theory*. Wiley.

Best, N., I. Perevozskaya, D. Lunn, G. Archer, J. Euesden, C. Fillmore, V. Sherina, D. Thompson, and M. Zwierzyna (2022). Predicting the COVID-19 pandemic impact on clinical trial recruitment. *Statistics in Biopharmaceutical Research 14*(1), 67–79.

Bezerra, L. C., M. López-Ibánez, and T. Stützle (2015). Automatic component-wise design of multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation 20*(3), 403–417.

Bezerra, L. C., M. López-Ibánez, and T. Stützle (2017). An empirical assessment of the properties of inverted generational distance on multi-and many-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 31–45. Springer.

Bezerra, L. C., L. Manuel, et al. (2020). Automatically designing state-of-the-art multi-and many-objective evolutionary algorithms. *Evolutionary computation 28*(2), 195–226.

Bianchi, L., M. Dorigo, L. M. Gambardella, and W. J. Gutjahr (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing 8*, 239–287.

Birattari, M. and J. Kacprzyk (2009a). *Tuning Metaheuristics: A Machine Learning Perspective*, Volume 197. Springer.

Birattari, M. and J. Kacprzyk (2009b). *Tuning metaheuristics: a machine learning perspective*, Volume 197. Springer.

Birattari, M., T. Stützle, L. Paquete, K. Varrentrapp, et al. (2002). A racing algorithm for configuring metaheuristics. In *GECCO*, Volume 2.

Blasco, X., J. Herrero, J. Sanchis, and M. Martínez (2008). A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization. *Information Sciences 178*(20), 3908–3924. Special Issue on Industrial Applications of Neural Networks.

Blum, C., R. Chiong, M. Clerc, K. De Jong, Z. Michalewicz, F. Neri, and T. Weise (2012). Evolutionary optimization. *Variants of evolutionary algorithms for real-world applications*, 1–29.

Blum, C., A. Roli, and M. Sampels (2008a). *Hybrid metaheuristics: an emerging approach to optimization*, Volume 114. Springer.

Blum, C., A. Roli, and M. Sampels (2008b). *Hybrid Metaheuristics: An Emerging Approach to Optimization*. Springer.

Bogin, V. (2022). Lasagna's law: a dish best served early. *Contemporary Clinical Trials Communications 26*, 100900.

Bonnans, J.-F., J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal (2006). *Numerical Optimization: Theoretical and Practical Aspects*. Springer Science & Business Media.

Bossek, J. (2018). Performance assessment of multi-objective evolutionary algorithms with the R package ecr. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '18, pp. 1350–1356. ACM.

Boussaïd, I., J. Lepagnot, and P. Siarry (2013). A survey on optimization metaheuristics. *Information Sciences 237*, 82–117.

Braun, M. A., P. K. Shukla, and H. Schmeck (2015). Obtaining optimal pareto front approximations using scalarized preference information. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 631–638.

Brown, W. A. (2019). *Lithium: A Doctor, a Drug, and a Breakthrough*. Liveright Publishing.

Calvez, B. and G. Hutzler (2005). Automatic tuning of agent-based models using genetic algorithms. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pp. 41–57. Springer.

Carrasco, J., S. García, M. Rueda, S. Das, and F. Herrera (2020). Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. *Swarm and Evolutionary Computation 54*, 100665.

Carter, R. E. (2004). Application of stochastic processes to participant recruitment in clinical trials. *Controlled Clinical Trials 25*(5), 429–436.

Carter, R. E., S. C. Sonne, and K. T. Brady (2005). Practical considerations for estimating clinical trial accrual periods: application to a multi-center effectiveness study. *BMC Medical Research Methodology 5*, 11.

Chen, P. Y., R. B. Chen, H. C. Tung, and W. K. Wong (2017). Standardized maximin D-optimal designs for enzyme kinetic models. *Chemometrics and Intelligent Laboratory Systems 169*, 79–86.

Chen, R. B., S. P. Chang, W. Wang, H. C. Tung, and W. K. Wong (2015). Minimax optimal designs via particle swarm optimization methods. *Statistics and Computing 25*(5), 975–988.

Chernoff, H. (1972). *Sequential Analysis and Optimal Design*. SIAM.

Coello, C. A. C. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer.

Coello, C. A. C. (2022). Constraint-handling techniques used with evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1310–1333.

Coello, C. C. (2006). Evolutionary multi-objective optimization: a historical view of the field. *IEEE computational intelligence magazine 1*(1), 28–36.

Coello Coello, C. A., S. González Brambila, J. Figueroa Gamboa, M. G. Castillo Tapia, and R. Hernández Gómez (2020). Evolutionary multiobjective optimization: open research areas and some challenges lying ahead. *Complex & Intelligent Systems 6*, 221–236.

Comets, E. and F. Mentré (2021). Developing tools to evaluate non-linear mixed effect models: 20 years on the npde adventure. *The AAPS Journal 23*(4), 75.

Comfort, S. (2013). Improving clinical trial enrollment forecasts using SORM. *Applied Clinical Trials 22*(5).

Cook, R. D. and W. K. Wong (1994). On the equivalence of constrained and compound optimal designs. *Journal of the American Statistician Association 89*, 687–692.

Couffignal, C., J. Bertrand, S. Sportiche, M. Jarroir, S. El Balkhi, N. Djebrani-Oussedik, J. Poupon, X. Declèves, F. Mentré, and F. Bellivier (2019). Population pharmacokinetic modeling of sustained release lithium in the serum, erythrocytes and urine of patients with bipolar disorder. *European Journal of Clinical Pharmacology 75*(4), 519–528.

Das, S. and P. N. Suganthan (2010). Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation 15*(1), 4–31.

De Almeida-Filho, A. T., D. F. de Lima Silva, and L. Ferreira (2021). Financial modelling with

multiple criteria decision making: A systematic literature review. *Journal of the Operational Research Society 72*(10), 2161–2179.

De Castro, L. N. and F. J. Von Zuben (2000). The clonal selection algorithm with engineering applications. In *Proceedings of GECCO*, Volume 2000, pp. 36–39.

Deb, K. and H. Jain (2013). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation 18*(4), 577–601.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation 6*(2), 182–197.

Derrac, J., S. García, D. Molina, and F. Herrera (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation 1*(1), 3–18.

Dobslaw, F. (2010). A parameter tuning framework for metaheuristics based on design of experiments and artificial neural networks. In *International conference on computer mathematics and natural computing*. WASET.

Dokeroglu, T., A. Deniz, and H. E. Kiziloz (2022). A comprehensive survey on recent metaheuristics for feature selection. *Neurocomputing 494*, 269–296.

Dokeroglu, T., E. Sevinc, T. Kucukyilmaz, and A. Cosar (2019). A survey on new generation metaheuristic algorithms. *Computers & Industrial Engineering 137*, 106040.

Dorigo, M., M. Birattari, and T. Stutzle (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine 1*(4), 28–39.

Duffull, S. B., S. Retout, and F. Mentré (2002). The use of simulated annealing for finding optimal population designs. *Computer Methods and Programs in Biomedicine 69*(1), 25–35.

Dumont, C., G. Lestini, H. Le Nagard, F. Mentré, E. Comets, T. T. Nguyen, and the PFIM Group (2018). PFIM 4.0, an extended R program for design evaluation and optimization in nonlinear mixed-effect models. *Computer Methods and Programs in Biomedicine 156*, 217–229.

Dumont, C., G. Lestini, H. L. Nagard, F. Mentré, E. Comets, and T. Nguyen (2018). Pfim 4.0, an

extended r program for design evaluation and optimization in nonlinear mixed-effect models. *Computer methods and programs in biomedicine 156*, 217–229.

Edgeworth, F. Y. (1881). *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences*. Number 10. CK Paul.

Egea, J. A., D. Henriques, T. Cokelaer, A. F. Villaverde, A. MacNamara, D.-P. Danciu, J. R. Banga, and J. Saez-Rodriguez (2014). Meigo: an open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics. *BMC bioinformatics 15*(1), 1–9.

El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University-Science 22*(3), 123–131.

Erdem, E., T. Aydın, and B. Erkayman (2021). Flight scheduling incorporating bad weather conditions through big data analytics: A comparison of metaheuristics. *Expert Systems 38*(8), e12752.

Eusuff, M., K. Lansey, and F. Pasha (2006). Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering optimization 38*(2), 129–154.

Eusuff, M. M. and K. E. Lansey (2003). Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources planning and management 129*(3), 210–225.

Ezugwu, A. E., A. K. Shukla, R. Nath, A. A. Akinyelu, J. O. Agushaka, H. Chiroma, and P. K. Muhuri (2021). Metaheuristics: a comprehensive overview and classification along with bibliometric analysis. *Artificial Intelligence Review 54*, 4237–4316.

Falcón-Cardona, J. G. and C. A. C. Coello (2020). Indicator-based multi-objective evolutionary algorithms: A comprehensive survey. *ACM Computing Surveys (CSUR) 53*(2), 1–35.

Fedorov, V. (1972). *Theory of Optimal Experiments*. Academic Press.

Fedorov, V. V. and S. L. Leonov (2014). *Optimal Design for Nonlinear Response Models*. Chapman and Hall/ CRC Press.

Fischetti, M., M. Fischetti, et al. (2018). Matheuristics. In *Handbook of heuristics*, Volume 1, pp. 121–153. Springer International Publishing.

Fister Jr, I., X.-S. Yang, I. Fister, J. Brest, and D. Fister (2013). A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*.

Foo, L. K. and S. Duffull (2010, 07). Methods of robust design of nonlinear models with an application to pharmacokinetics. *Journal of Biopharmaceutical Statistics 20*, 886–902.

Francke, T. (2020). *ppso: Particle Swarm Optimization and Dynamically Dimensioned Search, optionally using parallel computing based on Rmpi*. R package version 0.9-99991.

Gadkar, K., D. Kirouac, D. Mager, P. H. van der Graaf, and S. Ramanujan (2016). A six-stage workflow for robust application of systems pharmacology. *CPT: pharmacometrics & systems pharmacology 5*(5), 235–249.

Gajewski, B. J., S. D. Simon, and S. E. Carlson (2008). Predicting accrual in clinical trials with Bayesian posterior predictive distributions. *Statistics in Medicine 27*(13), 2328–2340.

Geem, Z. W., J. H. Kim, and G. Loganathan (2001a). A new heuristic optimization algorithm: harmony search. *Simulation 76*(2), 60–68.

Geem, Z. W., J. H. Kim, and G. V. Loganathan (2001b). A new heuristic optimization algorithm: harmony search. *SIMULATION 76*(2), 60–68.

Ghaheri, A., S. Shoar, M. Naderan, and S. S. Hoseini (2015). The applications of genetic algorithms in medicine. *Oman medical journal 30*(6), 406.

Gilman, A. and J. Ross (1995). Genetic-algorithm selection of a regulatory structure that directs flux in a simple metabolic model. *Biophysical journal 69*(4), 1321–1333.

Gitlin, M. (2016). Lithium side effects and toxicity: prevalence and management strategies. *International Journal of Bipolar Disorders 4*(1), 1–10.

Gkioni, E., R. Rius, S. Dodd, and C. Gamble (2019). A systematic review describes models for recruitment prediction at the design stage of a clinical trial. *Journal of Clinical Epidemiology 115*, 141–149.

Gkiotsalitis, K. (2023). Multi-objective optimization. In *Public Transport Optimization*, pp. 355–373. Springer.

Gmys, J., T. Carneiro, N. Melab, E.-G. Talbi, and D. Tuyttens (2020). A comparative study of high-

productivity high-performance programming languages for parallel metaheuristics. *Swarm and Evolutionary Computation 57*, 100720.

Gogna, A. and A. Tayal (2013). Metaheuristics: review and application. *Journal of Experimental & Theoretical Artificial Intelligence 25*(4), 503–526.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning* (1st ed.). USA: Addison-Wesley Longman Publishing Co., Inc.

Grefenstette, J. J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics 16*(1), 122–128.

Grof, P. (2010). Sixty years of lithium responders. *Neuropsychobiology 62*(1), 8–16.

Hansen, N., A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff (2021). Coco: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software 36*(1), 114–144.

Harrison, R. K. (2016). Phase II and phase III failures: 2013–2015. *Nature Review Drug Discovery 15*, 817–818.

Higashi, N. and H. Iba (2003). Particle swarm optimization with gaussian mutation. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706)*, pp. 72–79. IEEE.

Holland, J. H. (1992a). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT press.

Holland, J. H. (1992b). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT press.

Holland, J. H. (1992c). Genetic algorithms. *Scientific american 267*(1), 66–73.

Huang, C., Y. Li, and X. Yao (2019). A survey of automatic parameter tuning methods for metaheuristics. *IEEE Transactions on Evolutionary Computation 24*(2), 201–216.

Hussain, K., M. N. Mohd Salleh, S. Cheng, and Y. Shi (2019). Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review 52*, 2191–2233.

Imran, M., R. Hashim, and N. E. Abd Khalid (2013). An overview of particle swarm optimization variants. *Procedia Engineering 53*, 491–496.

Jain, H. and K. Deb (2013). An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation 18*(4), 602–622.

Jain, M., V. Saihjpal, N. Singh, and S. B. Singh (2022). An overview of variants and advancements of pso algorithm. *Applied Sciences 12*(17), 8392.

Jiang, S., J. Zhang, Y.-S. Ong, A. N. Zhang, and P. S. Tan (2014). A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm. *IEEE Transactions on Cybernetics 45*(10), 2202–2213.

Jiang, Y., P. Guarino, S. Ma, S. Simon, M. S. Mayo, R. Raghavan, and B. J. Gajewski (2016). Bayesian accrual prediction for interim review of clinical studies: open source R package and smartphone application. *Trials 17*(1), 1–8.

Johnson, N. L., A. W. Kemp, and S. Kotz (2005). *Univariate Discrete Distributions*. John Wiley & Sons.

José-García, A. and W. Gómez-Flores (2016). Automatic clustering using nature-inspired metaheuristics: A survey. *Applied Soft Computing 41*, 192–213.

Karaboga, D. (2010). Artificial bee colony algorithm. *scholarpedia 5*(3), 6915.

Kassambara, A. and F. Mundt (2017). Package factoextra. *Extract and visualize the results of multivariate data analyses 76*(2).

Katoch, S., S. S. Chauhan, and V. Kumar (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications 80*, 8091–8126.

Kavita, S. and S. Shinde (2023). Metaheuristic evolutionary algorithms: Types, applications, future directions, and challenges. In *2023 3rd International Conference on Intelligent Technologies (CONIT)*, pp. 1–6. IEEE.

Kennedy, J. and R. Eberhart (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, Volume 4, pp. 1942–1948.

Khan, S., M. Asjad, and A. Ahmad (2015). Review of modern optimization techniques. *International Journal of Engineering Research and Technology 4*(4), 2278–0181.

Kim, S., A. C. Hooker, Y. Shi, G. H. J. Kim, and W. K. Wong (2021). Metaheuristics for pharmacometrics. *CPT: Pharmacometrics & Systems Pharmacology 10*(11), 1297–1309.

Konak, A., D. W. Coit, and A. E. Smith (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety 91*(9), 992–1007.

Kononova, A. V., D. B. Ingham, and M. Pourkashanian (2008). Simple scheduled memetic algorithm for inverse problems in higher dimensions: Application to chemical kinetics. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 3905–3912. IEEE.

Kruskal, W. H. and W. A. Wallis (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association 47*(260), 583–621.

Kulkarni, A. J., E. Mezura-Montes, Y. Wang, A. H. Gandomi, and G. Krishnasamy (2021). *Constraint Handling in Metaheuristics and Applications*. Springer.

Kumar, A., B. K. Singh, and B. Patro (2016). Particle swarm optimization: a study of variants and their applications. *International Journal of Computer Applications 135*(5), 24–30.

Lagaros, N. D., M. Kournoutos, N. A. Kallioras, and A. N. Nordas (2023). Constraint handling techniques for metaheuristics: a state-of-the-art review and new variants. *Optimization and Engineering*, 1–48.

Lambora, A., K. Gupta, and K. Chopra (2019). Genetic algorithm-a literature review. In *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pp. 380–384. IEEE.

Lange, K. (2013). *Optimization*, Volume 95. Springer Science & Business Media.

Lange, K., E. C. Chi, and H. Zhou (2014). A brief survey of modern optimization for statisticians. *International Statistical Review 82*(1), 46–70.

Laumanns, M., L. Thiele, and E. Zitzler (2005). An adaptive scheme to generate the pareto front based on the epsilon-constraint method. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Liu, J., J. Wick, Y. Jiang, M. Mayo, and B. Gajewski (2020). Bayesian accrual modeling and

prediction in multicenter clinical trials with varying center activation times. *Pharmaceutical Statistics 19*(5), 692–709.

Liu, W. and J. Wang (2019). A brief survey on nature-inspired metaheuristics for feature selection in classification in this decade. In *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 424–429. IEEE.

López-Ibáñez, M., L. P. Cáceres, J. Dubois-Lacoste, T. G. Stützle, and M. Birattari (2016). *The irace package: User guide*. IRIDIA.

López-Ibáñez, M., J. Dubois-Lacoste, L. Pérez Cáceres, T. Stützle, and M. Birattari (2016). The irace package: iterated racing for automatic algorithm configuration. *Operations Research Perspectives 3*, 43–58.

Ma, Z., G. Wu, P. N. Suganthan, A. Song, and Q. Luo (2023). Performance assessment and exhaustive listing of 500+ nature-inspired metaheuristic algorithms. *Swarm and Evolutionary Computation 77*, 101248.

Maechler, M., P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik (2022). *cluster: Cluster Analysis Basics and Extensions*. R package version 2.1.4 — For new features, see the 'Changelog' file (in the package source).

Maier, H. R., S. Razavi, Z. Kapelan, L. S. Matott, J. Kasprzyk, and B. A. Tolson (2019). Introductory overview: Optimization using evolutionary algorithms and other metaheuristics. *Environmental Modelling & Software 114*, 195–213.

Mantovani, R. G., A. L. Rossi, J. Vanschoren, B. Bischl, and A. C. De Carvalho (2015). Effectiveness of random search in svm hyper-parameter tuning. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. Ieee.

Marler, R. T. and J. S. Arora (2010). The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization 41*, 853–862.

Marmol, F. (2008). Lithium: bipolar disorder and neurodegenerative diseases possible cellular mechanisms of the therapeutic effects of lithium. *Progress in Neuro-Psychopharmacology and Biological Psychiatry 32*(8), 1761–1771.

Masoudi, E., B. P. M. Holling, H.and Duarte, and W. K. Wong (2019). Metaheuristic adaptive

cubature based algorithm to find bayesian optimal designs for nonlinear models. *Journal of Computational and Graphical Statistics 28*(4), 861–87.

McKnight, R. F., M. Adida, K. Budge, S. Stockton, G. M. Goodwin, and J. R. Geddes (2012). Lithium toxicity profile: a systematic review and meta-analysis. *The Lancet 379*(9817), 721–728.

Mebane Jr., W. R. and J. S. Sekhon (2011). Genetic optimization using derivatives: The rgenoud package for R. *Journal of Statistical Software 42*(11), 126.

Memon, A., I. Rogers, S. M. D. D. Fitzsimmons, B. Carter, R. Strawbridge, D. Hidalgo-Mazzei, and A. H. Young (2020). Association between naturally occurring lithium in drinking water and suicide rates: systematic review and meta-analysis of ecological studies. *The British Journal of Psychiatry 217*(6), 667678.

Mentré, F., R. Leroux, J. Seurat, and L. Fayette (2023). *PFIM: Population Fisher Information Matrix*. R package version 6.0.

Mercer, R. E. and J. Sampson (1978). Adaptive search using a reproductive meta-plan. *Kybernetes*.

Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*, Volume 12. Springer Science & Business Media.

Miller, B. L., D. E. Goldberg, et al. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems 9*(3), 193–212.

Miller, J. N. and D. W. Black (2020). Bipolar disorder and suicide: a review. *Current Psychiatry Reports 22*(2), 1–10.

Mirjalili, S. (2015a). The ant lion optimizer. *Advances in Engineering Software 83*, 80–98.

Mirjalili, S. (2015b). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems 89*, 228–249.

Mirjalili, S. (2015c). Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowledge-Based Systems 89*, 228–249.

Mirjalili, S. (2016a). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications 27*, 1053–1073.

Mirjalili, S. (2016b). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems 96*, 120–133.

Mirjalili, S. and A. Lewis (2016). The whale optimization algorithm. *Advances in Engineering Software 95*, 51–67.

Mirjalili, S., S. M. Mirjalili, and A. Lewis (2014). Grey wolf optimizer. *Advances in Engineering Software 69*, 46–61.

Mohakud, R. and R. Dash (2021). Designing a grey wolf optimization based hyper-parameter optimized convolutional neural network classifier for skin cancer detection. *Journal of King Saud University - Computer and Information Sciences*.

Mohakud, R. and R. Dash (2022). Designing a grey wolf optimization based hyper-parameter optimized convolutional neural network classifier for skin cancer detection. *Journal of King Saud University-Computer and Information Sciences 34* (8), 6280–6291.

Molina, D., J. Poyatos, J. D. Ser, S. García, A. Hussain, and F. Herrera (2020). Comprehensive taxonomies of nature-and bio-inspired optimization: Inspiration versus algorithmic behavior, critical analysis recommendations. *Cognitive Computation 12*, 897–939.

Moore, J. and R. Chapman (2003, 07). Application of particle swarm to multiobjective optimization.

Muñoz-Ordóñez, J., C. Cobos, M. Mendoza, E. Herrera-Viedma, F. Herrera, and S. Tabik (2018). Framework for the training of deep neural networks in tensorflow using metaheuristics. In *Intelligent Data Engineering and Automated Learning–IDEAL 2018: 19th International Conference, Madrid, Spain, November 21–23, 2018, Proceedings, Part I 19*, pp. 801–811. Springer.

Nadimi-Shahraki, M. H., H. Zamani, Z. Asghari Varzaneh, and S. Mirjalili (2023). A systematic review of the whale optimization algorithm: Theoretical foundation, improvements, and hybridizations. *Archives of Computational Methods in Engineering*, 1–47.

Nebro, A. J., J. J. Durillo, J. Garcia-Nieto, C. C. Coello, F. Luna, and E. Alba (2009). Smpso: A new pso-based metaheuristic for multi-objective optimization. In *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*, pp. 66–73. IEEE.

Nelder, J. and R. Mead (1965). A simplex method for function minimization. *Computer Journal 1*, 308–313.

Nematzadeh, S., F. Kiani, M. Torkamanian-Afshar, and N. Aydin (2022). Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases. *Computational biology and chemistry 97*, 107619.

Nyberg, J., C. Bazzoli, K. Ogungbenro, A. Aliev, S. Leonov, S. Duffull, A. C. Hooker, and F. Mentré (2015). Methods and software tools for design evaluation in population pharmacokinetics–pharmacodynamics studies. *British Journal of Clinical Pharmacology 79*(1), 6–17.

Nyberg, J., S. Ueckert, E. A. Strömberg, S. Hennig, M. O. Karlsson, and A. C. Hooker (2012). Poped: an extended, parallelized, nonlinear mixed effects models optimal design tool. *Computer methods and programs in biomedicine 108*(2), 789–805.

Osaba, E., E. Villar-Rodriguez, J. Del Ser, A. J. Nebro, D. Molina, A. LaTorre, P. N. Suganthan, C. A. C. Coello, and F. Herrera (2021). A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems. *Swarm and Evolutionary Computation 64*, 100888.

Osaba, E., X.-S. Yang, and J. Del Ser (2020). Traveling salesman problem: a perspective review of recent research and new results with bio-inspired metaheuristics. *Nature-Inspired Computation and Swarm Intelligence*, 135–164.

Pareto, V. (1896). Cours d'economie politique, Volume I and II. Lausanne. F.Rouge.

Perperoglou, A., Y. Zhang, and D. K. Kipourou (2022). Modeling time-varying recruitment rates in multicenter clinical trials. *Biometrical Journal 65*(6).

Polack, F. P., S. J. Thomas, N. Kitchin, J. Absalon, A. Gurtman, S. Lockhart, J. L. Perez, G. Pérez Marc, E. D. Moreira, C. Zerbini, R. Bailey, K. A. Swanson, S. Roychoudhury, K. Koury, P. Li, W. V. Kalina, D. Cooper, R. W. Frenck, L. L. Hammitt, O. Türeci, H. Nell, A. Schaefer, S. Ünal, D. B. Tresnan, S. Mather, P. R. Dormitzer, U. ahin, K. U. Jansen, and W. C. Gruber (2020). Safety and efficacy of the BNT162b2 mRNA COVID-19 vaccine. *New England Journal of Medicine 383*(27), 2603–2615.

Probst, P., M. N. Wright, and A.-L. Boulesteix (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9*(3),

e1301.

Puchinger, J. and G. R. Raidl (2005). Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In *International work-conference on the interplay between natural and artificial computation*, pp. 41–53. Springer.

Pukelsheim, F. (2006). *Optimal Design of Experiments (Classics in Applied Mathematics) (Classics in Applied Mathematics, 50)*. USA: Society for Industrial and Applied Mathematics.

Qian, H. and Y. Yu (2017). Solving high-dimensional multi-objective optimization problems with low effective dimensions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 31.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Rachmawati, L. and D. Srinivasan (2009). Multiobjective evolutionary algorithm with controllable focus on the knees of the pareto front. *IEEE Transactions on Evolutionary Computation 13*(4), 810–824.

Rahimi, I., A. H. Gandomi, F. Chen, and E. Mezura-Montes (2023). A review on constraint handling techniques for population-based algorithms: from single-objective to multi-objective optimization. *Archives of Computational Methods in Engineering 30*(3), 2181–2209.

Rahman, M. A. (2013). Quantile regression using metaheuristic algorithms. *International Journal of Computational Economics and Econometrics 3*(3-4), 205–233.

Rajwar, K., K. Deep, and S. Das (2023). An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. *Artificial Intelligence Review*, 1–71.

Rao, C. R. (1992). Information and the accuracy attainable in the estimation of statistical parameters. In *Breakthroughs in Statistics: Foundations and basic theory*, pp. 235–247. Springer.

Raquel, C. R. and P. C. Naval Jr (2005). An effective use of crowding distance in multiobjective particle swarm optimization. In *Proceedings of the 7th Annual conference on Genetic and Evolutionary Computation*, pp. 257–264.

Retout, S., E. Comets, A. Samson, and F. Mentré (2007). Design in nonlinear mixed effects

models: optimization using the Fedorov–Wynn algorithm and power of the Wald test for binary covariates. *Statistics in Medicine 26*(28), 5162–5179.

Reynolds, A. P., G. Richards, B. de la Iglesia, and V. J. Rayward-Smith (2006). Clustering rules: a comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms 5*, 475–504.

Romanycia, M. H. and F. J. Pelletier (1985). What is a heuristic? *Computational intelligence 1*(1), 47–58.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics 20*, 53–65.

Sadoff, J., G. Gray, A. Vandebosch, V. Cárdenas, G. Shukarev, B. Grinsztejn, P. A. Goepfert, C. Truyers, H. Fennema, B. Spiessens, K. Offergeld, G. Scheper, K. L. Taylor, M. L. Robb, J. Treanor, D. H. Barouch, J. Stoddard, M. F. Ryser, M. A. Marovich, K. M. Neuzil, L. Corey, N. Cauwenberghs, T. Tanner, K. Hardt, J. Ruiz-Guiñazú, M. Le Gars, H. Schuitemaker, J. Van Hoof, F. Struyf, and M. Douoguih (2021). Safety and efficacy of single-dose Ad26.COV2.S vaccine against COVID-19. *New England Journal of Medicine 384*(23), 2187–2201.

Sallan, J. M., O. Lordan, and V. Fernandez (2015). *Modeling and solving linear programming with R*. OmniaScience.

Santoso, F. Y., E. Sediyono, and H. D. Purnomo (2023). Genetic algorithm for convolutional neural network hyperparameter tuning. In *2023 3rd International Conference on Computing and Information Technology (ICCIT)*, pp. 232–236. IEEE.

Schlander, M., K. Hernandez-Villafuerte, C. Y. Cheng, J. Mestre-Ferrandiz, and M. Baumann (2021). How much does it cost to research and develop a new drug? A systematic review and assessment. *PharmacoEconomics 39*, 1–27.

Schubert, E. and P. J. Rousseeuw (2019). Faster k-medoids clustering: improving the pam, clara, and clarans algorithms. In *Similarity Search and Applications: 12th International Conference, SISAP 2019, Newark, NJ, USA, October 2–4, 2019, Proceedings 12*, pp. 171–187. Springer.

Schulze, T. G., M. Alda, M. Adli, N. Akula, R. Ardau, E. T. Bui, C. Chillotti, S. Cichon, P. Czerski,

M. Del Zompo, et al. (2010). The international consortium on lithium genetics (conligen): an initiative by the nimh and igsli to study the genetic basis of response to lithium treatment. *Neuropsychobiology 62*(1), 72–78.

Schutze, O., X. Esquivel, A. Lara, and C. A. C. Coello (2012). Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation 16*(4), 504–522.

Scrucca, L. (2013). GA: A package for genetic algorithms in R. *Journal of Statistical Software 53*(4), 1–37.

Senn, S. (1997). *Statistical Issues in Drug Development*. Wiley.

Senn, S. (1998). Some controversies in planning and analysing multi-centre trials. *Statistics in Medicine 17*(15-16), 1753–65.

Septem, R. L., E. P. Nugroho, M. B. Adi Prabowo, E. Junaeti, and A. G. Abdullah (2019). Metaheuristicopt: Metaheuristic for optimization. *R package version 2.0.0. https://CRAN.R-project.org/package=metaheuristicOptl*.

Septem Riza, L., Iip, E. Prasetyo Nugroho, M. B. Adi Prabowo, E. Junaeti, and A. G. Abdullah (2019). metaheuristicOpt: metaheuristic for optimization. R package version 2.0.0.

Shahapure, K. R. and C. Nicholas (2020). Cluster quality analysis using silhouette score. In *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*, pp. 747–748. IEEE.

Sheikh, M., M. Qassem, I. F. Triantis, and P. A. Kyriacou (2022). Advances in therapeutic monitoring of lithium in the management of bipolar disorder. *Sensors 22*(3), 736.

Sheiner, L. B., B. Rosenberg, and V. V. Marathe (1977). Estimation of population characteristics of pharmacokinetic parameters from routine clinical data. *Journal of Pharmacokinetics and Biopharmaceutics 5*, 445–479.

Shorter, E. (2009). The history of lithium therapy. *Bipolar Disorders 11*, 4–9.

Silberholz, J. and B. Golden (2010). Comparison of metaheuristics. *Handbook of Metaheuristics*, 625–640.

Silvey, S. D. (1980). *Optimum Design*. Chapman and Hall.

Singh, H., S. Tyagi, P. Kumar, S. S. Gill, and R. Buyya (2021). Metaheuristics for scheduling of heterogeneous tasks in cloud computing environments: Analysis, performance evaluation, and future directions. *Simulation Modelling Practice and Theory 111*, 102353.

Singh, N., S. Singh, et al. (2017). Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *Journal of Applied Mathematics 2017*.

Smith, A. E., D. W. Coit, T. Baeck, D. Fogel, and Z. Michalewicz (1997a). Penalty functions. *Handbook of evolutionary computation 97*(1), C5.

Smith, A. E., D. W. Coit, T. Baeck, D. Fogel, and Z. Michalewicz (1997b). Penalty functions. *Handbook of Evolutionary Computation 97*(1), C5.

Soler-Dominguez, A., A. A. Juan, and R. Kizys (2017). A survey on financial applications of metaheuristics. *ACM Computing Surveys (CSUR) 50*(1), 1–23.

Sörensen, K. and F. Glover (2013). Metaheuristics. *Encyclopedia of operations research and management science 62*, 960–970.

Srinivas, N. and K. Deb (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation 2*(3), 221–248.

Sully, B. G. O., S. A. Julious, and J. Nicholl (2013). A reinvestigation of recruitment to randomised, controlled, multicenter trials: a review of trials funded by two UK funding agencies. *Trials 14*(166), 1–9.

Sun, D., W. Gao, H. Hu, and S. Zhou (2022). Why 90% of clinical drug development fails and how to improve it? *Acta Pharmaceutica Sinica B 12*(7), 3049–3062.

Sun, J., J. M. Garibaldi, and C. Hodgman (2011). Parameter estimation using metaheuristics in systems biology: a comprehensive review. *IEEE/ACM transactions on computational biology and bioinformatics 9*(1), 185–202.

Swarndeep Saket, J. and S. Pandya (2016). An overview of partitioning algorithms in clustering techniques. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 5*(6), 1943–1946.

Syswerda, G. et al. (1989). Uniform crossover in genetic algorithms. In *ICGA*, Volume 3, pp. 2–9.

Talbi, E.-G. (2002). A taxonomy of hybrid metaheuristics. *Journal of heuristics 8*, 541–564.

Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation.* John Wiley & Sons.

Talbi, E. G. (2021). Machine learning into metaheuristics: A survey and taxonomy. *ACM Computing Surveys (CSUR) 54*(6), 1–32.

Talbi, E.-G. et al. (2013). *Hybrid metaheuristics*, Volume 166. Springer.

Talukder, A. K. A. and K. Deb (2020). Paletteviz: A visualization method for functional understanding of high-dimensional pareto-optimal data-sets to aid multi-criteria decision making. *IEEE Computational Intelligence Magazine 15*(2), 36–48.

Tendys, T. (2016). *gafit: Genetic Algorithm for Curve Fitting.* R package version 0.5.1.

Tessema, B. and G. G. Yen (2009). An adaptive penalty formulation for constrained evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 39*(3), 565–578.

Tian, Y., R. Cheng, X. Zhang, and Y. Jin (2017). PlatEMO: a MATLAB platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine 12*(4), 73–87.

Tominaga, D., N. Koga, and M. Okamoto (2000). Efficient numerical optimization algorithm based on genetic algorithm for inverse problem. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, pp. 251–258.

Umbarkar, A. J. and P. D. Sheth (2015). Crossover operators in genetic algorithms: a review. *ICTACT Journal on Soft Computing 6*(1).

Valencia-Rodríguez, D. C. and C. A. C. Coello (2023). Influence of the number of connections between particles in the performance of a multi-objective particle swarm optimizer. *Swarm and Evolutionary Computation 77*, 101231.

Van Thieu, N. and S. Mirjalili (2023). MEALPY: an open-source library for latest meta-heuristic algorithms in Python. *Journal of Systems Architecture 139*, 102871.

Verma, S., M. Pant, and V. Snasel (2021). A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems. *IEEE Access 9*, 57757–57791.

Vivekanandan, T. and N. C. S. N. Iyengar (2017). Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease. *Computers in biology and medicine 90*, 125–136.

Walters, S. J., I. Bonacho dos Anjos Henriques-Cadby, O. Bortolami, L. Flight, D. Hind, R. M. Jacques, C. Knox, B. Nadin, J. Rothwell, M. Surtees, et al. (2017). Recruitment and retention of participants in randomised controlled trials: a review of trials funded and published by the United Kingdom Health Technology Assessment Programme. *BMJ Open 7*, e015276.

Wang, Q. (1997). Using genetic algorithms to optimise model parameters. *Environmental Modelling & Software 12*(1), 27–34.

Whitacre, J. M. (2011a). Recent trends indicate rapid growth of nature-inspired optimization in academia and industry. *Computing 93*, 121–133.

Whitacre, J. M. (2011b). Survival of the flexible: explaining the recent dominance of nature-inspired optimization within a rapidly evolving world. *Computing 93*, 135–146.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin 1*(6), 80–83.

Williford, W. O., S. F. Bingham, D. G. Weiss, J. F. Collins, K. T. Rains, and W. F. Krol (1987). The "constant intake rate" assumption in interim recruitment goal methodology for multicenter clinical trials. *Journal of Chronic Diseases 40*(4), 297–307.

Wolpert, D. H. and W. G. Macready (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation 1*(1), 67–82.

Xiong, N., D. Molina, M. L. Ortiz, and F. Herrera (2015). A walk into metaheuristics for engineering optimization: principles, methods and recent trends. *International Journal of Computational Intelligence Systems 8*(4), 606–636.

Xue, B., M. Zhang, W. N. Browne, and X. Yao (2015). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation 20*(4), 606–626.

Yacobi, S. and A. Ornoy (2008). Is lithium a real teratogen? What can we conclude from the prospective versus retrospective studies? A review. *Israel Journal of Psychiatry and Related Sciences 45*(2), 95.

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In J. R. Gonzalez, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor (Eds.), *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Volume 284. Springer.

Yang, X.-S. (2012). *Swarm-based metaheuristic algorithms and no-free-lunch theorems.* Theory and New Applications of Swarm Intelligence.

Yang, X. S. (2020). *Nature-Inspired Optimization Algorithms.* Academic Press.

Yang, Z., X. Cai, and Z. Fan (2014). Epsilon constrained method for constrained multiobjective optimization problems: some preliminary results. In *Proceedings of the companion publication of the 2014 annual conference on genetic and evolutionary computation*, pp. 1181–1186.

Zavala, G. R., A. J. Nebro, F. Luna, and C. A. Coello Coello (2014). A survey of multi-objective metaheuristics applied to structural optimization. *Structural and Multidisciplinary Optimization 49*, 537–558.

Zhang, B., Q.-k. Pan, L.-l. Meng, C. Lu, J.-h. Mou, and J.-q. Li (2022). An automatic multi-objective evolutionary algorithm for the hybrid flowshop scheduling problem with consistent sublots. *Knowledge-Based Systems 238*, 107819.

Zhong, J., X. Hu, J. Zhang, and M. Gu (2005). Comparison of performance between different selection strategies on simple genetic algorithms. In *International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06)*, Volume 2, pp. 1115–1121. IEEE.

Zhu, W. and W. K. Wong (2001). Bayesian optimal designs for estimating a set of symmetric quantiles. *Statistics in Medicine 20*, 123–137.

Zitzler, E., D. Brockhoff, and L. Thiele (2007). The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings 4*, pp. 862–876. Springer.

Zitzler, E. and L. Thiele (1998). Multiobjective optimization using evolutionary algorithmsa comparative case study. In *International Conference on Parallel Problem Solving from Nature*, pp. 292–301. Springer.