EVOLUTIONARY COMPUTATION FOR SOLVING CONSTRAINED AND LARGE-SCALE OPTIMIZATION PROBLEMS

XU WEINAN

NATIONAL UNIVERSITY OF SINGAPORE

2019

EVOLUTIONARY COMPUTATION FOR SOLVING CONSTRAINED AND LARGE-SCALE OPTIMIZATION PROBLEMS

XU WEINAN

(*B.Eng., National University of Singapore*)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

NUS GRADUATE SCHOOL FOR INTEGRATIVE SCIENCES AND ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2019

Supervisor:

Professor Wong Lim Soon

Examiners:

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Xu Weinan

December 2, 2019

Acknowledgment

This work could not be done without the supports, encouragements and guidance of many people.

First, I would like to express my deep gratitude to my previous supervisor and co-supervisor, Professor Xu Jianxin and Professor Tan Kay Chen for guiding me through the hardest time of my Ph.D study. Prof. Xu guided me on how to conduct a solid research work. He instructed me from selecting a topic, discussing efficiently on solving tough issues, as well as writing a concise and high-quality research paper. He gave me the confidence, patience and great support, which helped me to build the confidence on my Ph.D study. Although Prof. Xu passed away, his kindness and integrity will encourage me throughout my life. Prof. Tan led me to start this research journey and constantly guided me to keep on the right research track. He was strict to both my research work and paper writing, and could correctly point out my mistakes and weaknesses. He provided countless sincere help to me in each of my research studies. It is his support and rigorous requirements that pushed me to complete two solid works. I will remember his insightful words and earnest guidance. I also express my deep gratitude to my current supervisor, Professor Wong Lim Soon. Prof. Wong was very kind to accept me as his Ph.D student. After Prof. Xu passed away, he took care of me so that I could feel the sense of security when I joined the new group. He gave me patient guidance and lots of constructive suggestions to my last research work as well as my thesis writing.

I am also grateful to some other senior scholars who helped me a lot on my Ph.D journey. I thank Professor Michael I. Jordan from UC Berkeley. Thanks for his openness and constructive supports which brought a different view of research. He also offered more collaborative opportunities to me. I thank Professor Weng

Kee Wong from UC Los Angeles. His patient and continuous support and assistance encouraged me to complete my second work. I also thank Professor David Hsu and Professor Dipti Srinivasan for their kind suggestions as my Thesis Advisory Committee.

My thanks also go to my seniors as well as other lab buddies who helped me and shared their invaluable experience. Qiu Xin provided many useful suggestions and helps in my early research study to demonstrate smart methods to conduct a research work. His cheerful attitude and kindness rekindled me to move on when facing difficulties. Zhang Chong, as my senior and good friend, shared with me a lot of her research experiences and provided many suggestions to me both in research and career. Jibin and Zihan raised many curious and exciting topics during lunch time. I thank Wenhao, Luyu, Weixin, Neamul and Stefano to make me feel very comfortable and happy in the new research group. I also thank Ruoxu, Chunjiang, Chen Lei, Cui'e, Sivam, Raj, Mr. Zhang Hengwei and Ms. Sara for their continuous assistance in various tasks. Thanks for the great helps in the project from Danhua, Zhongxiang and Richard. All these buddies have provided their assistance and brought me a pleasant research environment.

Last but not the least, I express my deep appreciation to my parents and husband. It's my father who sets the example for me to pursue and provides his selfless support and precious suggestion. My mother is the one who grants me the strength all along the way. It's my husband who always supports me at my back and builds the future together with me. The unconditional love from them is the source of my faith and courage.

Xu Weinan

December 2, 2019

Contents

Summary

Evolutionary computation is a family of nature-inspired metaheuristic algorithms for solving various optimization problems. The most significant advantage of evolutionary computation is that it does not make any assumptions about the characteristics and the underlying landscapes of the optimization problem being considered. Thus, evolutionary computation can tackle a variety of optimization problems even when decision variables and objective functions have complex characteristics. Decision variables having complex characteristics are very common in most real applications, but can be hardly handled by conventional optimization algorithms. In this thesis, some new evolutionary computation algorithms are proposed to handle two representative complex characteristics of decision variables, and are applied to solve real problems.

The first complex characteristic is decision variables that are with constraints. Constrained optimization is more difficult for the following reasons: rigorous pressure of searching towards the optimal region is exerted since only a subset of the decision space, termed as a feasible region, is considered for optimization. Furthermore, due to the conflicting effects of optimization and reducing constraint violations, it is hard for a general constraint-handling technique to balance both the task of optimization and the task of reducing constraint violations for different problems. My proposed constraint-handling technique ε-SEC can tackle these difficulties. ε-SEC partitions candidate solutions into different sections of the feasible region to enhance evolutionary search diversity. In addition, the upper bound of each section is reduced dynamically to drive the convergence of optimized solutions. The effective balance of both tasks is achieved by concurrently enhancing the diversity of solutions and driving solutions gradually

convergent to the optimal feasible region. The proposed constraint-handling technique ε-SEC is applied to solve a formulated parametric optimization problem with several constraints of a dynamic-system model. My experiment results show the effectiveness and robustness of ε-SEC in solving a constrained parametric optimization problem compared with other state-of-the-art constraint-handling techniques in evolutionary computation. Moreover, with the optimized parameters, the dynamic system model can be used for prognostication of outcome.

However, there exist small-data issues in this constrained parameter optimization. To tackle this issue, a data-augmentation pipeline is formulated. Empirical results provide evidence that this data-augmentation pipeline is effective in improving model resolution by various constraint-handling techniques. The results also provide evidence that ε-SEC sustains its superior model resolution compared to the other four constraint-handling techniques post data augmentation. Lastly, the results provide evidence that training the other constraint-handling techniques using the pseudo training data set generated by ε-SEC improves their resolution more than using their own respective pseudo training data set.

The second complex characteristic is decision variables that are high-dimensional. When a problem becomes high-dimensional, the issue of premature convergence is raised and solutions are easily trapped in local optima. Moreover, if decision variables are not only high-dimensional but also dependent on each other, the problem becomes non-separable and it is more difficult for solutions to escape local optima. Although there are several state-of-the-art evolutionary computation algorithms for solving high-dimensional problems, there is no specially-designed one to enhance the exploration of novelty solutions and to assist in escaping local optima. Differential evolution (DE) in the family of evolutionary computation is applied as the base algorithm to solve high-dimensional problems since DE can solve numerical problems well and has simple operations. My proposed algorithm NovDE is specially designed to explore novelty regions in the decision

space different from previous explored regions so that NovDE remedies the lack of exploration in other state-of-the-art DE variants when handling high-dimensional problems. The diversity of solutions is enhanced so that solutions are capable of escaping local optima with a higher chance. NovDE is applied to solve the domain applications of optimal designs which are widely used in industrial designs and bio-medical fields. A commonly used optimal design is D-optimal design, but previous studies can only tackle trivial D-optimal designs. High-dimensional D-optimal design problems are never touched in previous studies. My experiment results show that NovDE can find optimal design points for some high-dimensional D-optimal design problems with different settings. Moreover, NovDE can also find the optimal design of a real problem with more demanding requirements.

To sum up, this thesis proposes new evolutionary computation algorithms for solving the optimization problems with decision variables that are with constraints or are high-dimensional. When handling constrained optimization problems, the proposed technique ε-SEC has successfully addressed the issue of balancing the two conflicting tasks of optimizing objective function and reducing constraint violations. To solve the small-data issue in some constrained optimization problems, a data-augmentation pipeline is formulated. With more generated pseudo data fed into the model, constraint-handling techniques can improve their model resolution. However, even with more data, ε-SEC still presents superior effectiveness and robustness compared to other constraint-handling techniques. When handling high-dimensional problems, the proposed approach NovDE has successfully solved the issue of solutions easily getting trapped in local optima. The effectiveness and robustness of these proposed algorithms are supported by empirical studies.

List of Tables

List of Figures

List of Algorithms

List of Symbols

D	Input data dimension
$[\cdot]^T, [\cdot]^{-1}$	Transpose, inverse operation
$\mathcal{N}$	Normal distribution
$f, f(\cdot)$	Objective vector or functions
g	Generation
F	Scaling factor
CR	Crossover rate
$\phi(\cdot)$	Constraint violation

List of Abbreviations

Bin Binomial

CC Cooperative co-evolution

CpG Cyclophosphamide Toll-like receptor agonist

CY Cyclophosphamide

DE Differential Evolution

EA Evolutionary Algorithm

EC Evolutionary Computation

ES Evolution Strategy

FEs Function evaluation times

FIFO First-in-First-out

MER Multiple Exponential Recombination

PSO Particle Swarm Optimization

SaDE Self-adaptation Differential Evolution

Chapter 1

Introduction

Solving an optimization problem is to search the space of its decision variables x that optimizes the objective function $f(x)$ of the problem. If its decision variables do not have complex characteristics, the problem can be trivially solved by conventional optimization techniques or some heuristic methods [1]. However, it is common in real problems that their decision variables have complex characteristics. This thesis studies decision variables with various complex characteristics.

1.1 Background and Motivation

In real problems, two common complex characteristics of decision variables are that of constraints and high-dimensionality. Decision variables with constraints are discussed at first.

If there are constraints on decision variables, only a subset of the decision space termed as a feasible region (c.f. Fig. 1.1) is considered for optimization [2]. Compared with unconstrained optimization, constrained optimization is more difficult due to the need to balance both the task of optimization and the task of reducing constraint violations [3]. While more emphasis on optimization would make it difficult to drive solutions into the feasible region, more emphasis

on reducing constraint violations would obtain the solutions far away from the optimal one. Thus, balancing these two tasks makes the problem more complex. In various industrial fields, decision variables generally follow some principles as constraints. For instance, in the field of scheduling [4, 5], in addition to pursuing the optimal objective value, the maximum capacity for transportation is formulated as constraints on decision variables. In the area of clinical therapy [6], the optimization objective is to assign the dosage of each medicine to achieve optimal treatment outcomes. However, the dosage of each medicine should also follow specific physiological and pharmacological principles to avoid adverse effects after the treatment. Thus, the decision variables are required to be constrained as well. Conventionally, Lagrangian function is used to handle constrained optimization problems. However, Lagrangian function is effective only when both the objective function and the constraints are convex [7]. In real life, this precondition is not a general case. Thus, the issue of solving real optimization problems with constraints effectively is raised.

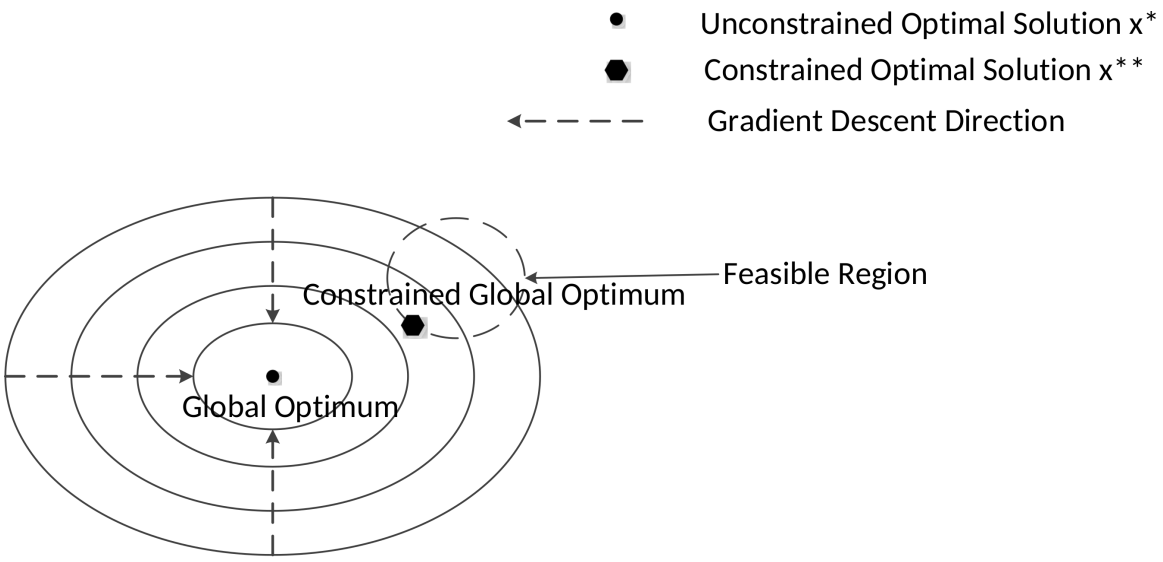

Figure 1.1: Decision space of a constrained optimization problem.

In addition to decision variables with constraints, high-dimensionality of decision variables is another complex characteristic. When a problem becomes

high-dimensional, the issue of premature convergence often arises and solutions often get trapped in local optima [8]. The complexity of finding an optimal solution becomes larger as the number of dimensions increases. Furthermore, if the high-dimensional decision variables are dependent on each other, the problem is non-separable [9], resulting in difficulty to escape local optima. It is far more difficult to solve the problem if the decision variables are both high-dimensional and dependent on each other. However, this is the more general case in real problems, and the scale of the decision variables can be in the order of hundreds or thousands in real problems. Such problems cannot be solved effectively by conventional optimization techniques. Thus, the issue of effectively solving high-dimensionality and non-separability real problems is raised.

Conventional optimization techniques fail in solving the issues above because they require an optimization problem to satisfy certain preconditions such as convexity and low-dimensionality in its decision space. Stochastic search techniques like meta-heuristics are more suitable and efficient for solving such real problems.

As one of the most famous meta-heuristics, the family of evolutionary computation (EC) has demonstrated a capability of solving optimization problems in many practical applications [10–12]. Evolutionary Algorithm (EA) is one of the representative algorithms of EC. EA is a population-based optimization algorithm and mimics the natural evolution process [13]. Each solution is represented as a single individual in the population. Each individual evolves itself by performing the operations of crossover, mutation and survival selection iteratively. The significant advantages of EA are that first, no assumptions are made in terms of the underlying landscape of the optimization problems as well as the constraints; second, population-based search encourages the exploration of various regions in the decision space at early evolutionary stage to avoid getting trapped in local optima.

This thesis aims at addressing all the aforementioned issues by proposing new algorithms of EC.

1.2 Basic Definition

Evolutionary Computation

Evolutionary computation (EC) is a family of algorithms for global optimization. Evolutionary algorithm (EA) is one of the representative algorithms of EC [14].

1.2.1 Evolutionary Algorithm

EA is a population-based optimization algorithm inspired by the natural evolution process. Each individual in the population represents a single solution in the decision space. These individuals evolve themselves generation by generation by performing the operations of crossover, mutation and survival selection sequentially [15].

Each individual is encoded as a chromosome and is initialized using a sampling method. All the individuals together constitute a population. As shown in Fig. 1.2, for crossover, two parent individuals are randomly selected from the population, and part of the decision variables of these two individuals are interchanged to form child individuals so that some information of the chromosome can be combined and passed to the offspring. For mutation shown in Fig. 1.3, a child individual obtained after crossover is mutated on one or several decision variable(s) to enhance the diversity of the distributions of the population. The operation of survival selection is that parent and child individuals are combined together to select elite individuals that would survive into the next generation based on their objective function values.

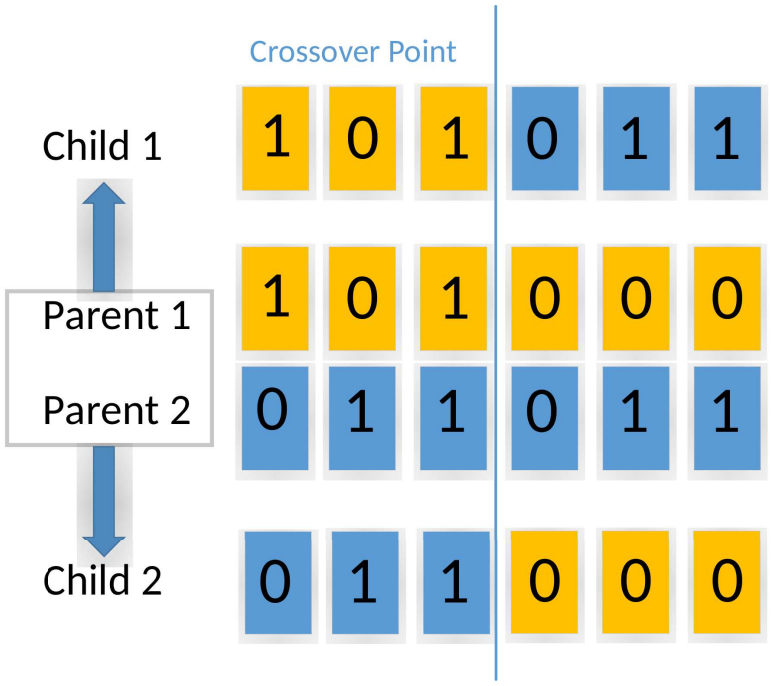

Figure 1.2: The crossover operation of the evolutionary algorithm.

Figure 1.3: The mutation operation of the evolutionary algorithm.

Differential Evolution

Differential Evolution (DE) [16] is one of the algorithms from the family of EA, and was proposed by Storn and Price in 1995. DE is a population-based optimization algorithm that searches for the optimum iteratively. DE is simple to implement and has good performance for solving various types of optimization problems. Compared with the standard evolutionary algorithm (EA), DE does not need to encode each solution into a binary code as the chromosome for conducting crossover and mutation operations and after that, to decode the

chromosome into the solution vector. DE can conduct crossover and mutation operations directly on each solution without encoding and decoding, so DE is suitable to solve numerical optimization problems. Furthermore, the space complexity of DE is low [17] and the number of control parameters in DE is small [18–20] . There are two control parameters in DE: a scaling factor F for mutation and a crossover rate CR for crossover. The parameter F controls convergence speed and the parameter CR affects both the convergence and the diversity of populations [21, 22].

To fix ideas, suppose $f(X)$ is a given objective function and we want to minimize it over a user-selected D-dimensional space comprising the decision variables. DE has three main operations: mutation, crossover and survival selection. Each solution of generation g is represented by $X_{i,g}$, where i is the index of the corresponding solution. Sometimes $X_{i,g}$ is referred to as the target vector, which needs to be updated for the next generation $g + 1$. A mutant vector $V_{i,g}$ is generated by mutation. After the mutation operation, a trial vector $U_{i,g}$ is generated by crossover using both $V_{i,g}$ and $X_{i,g}$. Survival selection is conducted after crossover, and a decision is made whether $X_{i,g+1} = U_{i,g}$ or $X_{i,g+1} = X_{i,g}$ based on the objective function values of $U_{i,g}$ and $X_{i,g}$. Some details for the three operations are as follows:

i) Mutation

Each target vector $X_{i,g}$ generates a new individual, called the mutant vector $V_{i,g}$. Some frequently used mutation strategies are listed below.

"DE/rand/1":

$$V_{i,g} = X_{r1,g} + F \cdot (X_{r2,g} - X_{r3,g}) \tag{1.1}$$

"DE/rand-to-best/2":

$$V_{i,g} = X_{i,g} + F \cdot (X_{best,g} - X_{i,g}) + F \cdot (X_{r1,g} - X_{r2,g}) + F \cdot (X_{r3,g} - X_{r4,g}) \tag{1.2}$$

"DE/rand/2":

$$V_{i,g} = X_{r1,g} + F \cdot (X_{r2,g} - X_{r3,g}) + F \cdot (X_{r4,g} - X_{r5,g}) \tag{1.3}$$

"DE/current-to-rand/1":

$$V_{i,g} = X_{i,g} + K \cdot (X_{r1,g} - X_{i,g}) + F \cdot (X_{r2,g} - X_{r3,g}) \tag{1.4}$$

In (1.4), K is randomly generated from $[0, 1]$. In (1.1)−(1.4), $X_{r1,g}$ to $X_{r5,g}$ represent the random individuals selected from the population pool at generation g.

ii) Crossover

Crossover operation is employed after mutation. In crossover, the mutant vector $V_{i,g}$ is recombined with the original individual $X_{i,g}$ to form the trial vector $U_{i,g}$. Two types of crossover schemes of DE are binomial crossover and exponential crossover, respectively. Binomial crossover is commonly used in DE to determine the trial vector as follows [23]:

$$U_{i,g}^{j} = \begin{cases} V_{i,g}^{j}, \text{ if } \mathrm{rand_{ij}}(0,1) \leq \mathrm{Cr} \text{ or j} = \mathrm{rand_i}(1,\mathrm{D}) \\ X_{i,g}^{j}, \text{ otherwise} \end{cases} \quad j = 1, 2, \cdots D \tag{1.5}$$

where $U_{i,g}^{j}$ is the $j-th$ element of $U_{i,g}^{j}$, and $rand_i(1, D)$ is an index randomly selected from $\{1, 2, 3, \cdots, D\}$ to ensure that the trial vector $U_{i,g}$ can get at least one variable from the mutant vector $V_{i,g}$. For each i and j, the notation

$rand_{ij}(0,1)$ is a fresh uniform random number from the interval [0,1] and Cr is the pre-specified crossover rate.

An exponential crossover is another way to implement the crossover operation [24]. An integer z is randomly generated from [1,D]. Another integer L, i.e. the length of decision variables to be mutated, is determined as follows:

L=0

WHILE(rand(0,1)$\leq Cr$ AND $L \leq$ D-z)

DO($L = L + 1$)

After determining the value of L, the exponential crossover operation is conducted as follows:

If $L \geq$1, the trial vector $U_{i,g}$ is generated as follows:

$$U_{i,g}^{j} = \begin{cases} V_{i,g}^{j} \ \text{, if j} \in \{\text{z}, \text{z}+1, \text{z}+2, \cdots, \text{z}+\text{L}-1\} \\ X_{i,g}^{j} \ \text{, otherwise} \end{cases} \quad j = 1, 2, \cdots D \quad (1.6)$$

If $L = 0$, then $U_{i,g} = X_{i,g}$.

iii) Survival Selection

Survival selection is the last operation to determine whether the trial vector $U_{i,g}$ survives to enter the next generation based on the objective function value $f(U_{i,g})$.

The survival selection operation in DE is described below:

$$X_{i,g+1} = \begin{cases} U_{i,g}, \text{if } f(U_{i,g}) \leq f(X_{i,g}) \\ X_{i,g}, \text{otherwise} \end{cases} \quad (1.7)$$

1.2.2 Formulation of Constrained Optimization Problem and High-dimensional Problem

Constrained Optimization Problem

The constrained optimization problem is formulated as follows:

$$\begin{aligned} &\min f(x) \\ &s.t.\ \mathrm{g}_j(x) \leq 0, \text{where } j = 1, 2, \cdots n \end{aligned} \tag{1.8}$$

In [25], constraint violation can be defined in terms of the maximum of all constraints or the sum of all constraints. In this thesis, constraint violation is defined as the maximum of all constraints:

$$\phi(\boldsymbol{x}) = \max_j \{\max\{0, g_j(\boldsymbol{x})\}\}, \tag{1.9}$$

where $g_j(\boldsymbol{x})$ is the constraint value of each constraint j. If any constraint j is violated, $\phi(\boldsymbol{x}) > 0$. Otherwise, $\phi(\boldsymbol{x}) = 0$.

High-Dimensional and Non-Separable Problem

When the dimension of its decision variables is at least 50, the problem is generally regarded as high-dimensional in the EC community. In this thesis, the dimension of the decision variables is assumed to be larger than 100.

If a problem is non-separable, it means that the decision variables are dependent on each other. For instance, the objective function is an additive model with variable interactions such as:

$$f(x) = x_i x_j + x_j x_k + x_k x_l, \tag{1.10}$$

where x_i, x_j, x_k, x_l are decision variables of solution x. These decision variables, e.g. x_j, cannot be optimized independently. Since x_j has interactions with x_i and

x_k, the optimization of x_j should be conducted concurrently with both x_i and x_k. In order to escape local optima, it is essential to explore regions in the decision space that are different from the current x_i, x_j and x_k together.

In real problems, it is common that the decision variables are both high-dimensional as well as non-separable.

1.3 List of Contributions

This thesis focuses on designing evolutionary computation algorithms to solve optimization problems having decision variables with complex characteristics. Both the scope and contributions are listed below:

1. Although recent studies have shown that evolutionary constraint-handling techniques are capable of solving optimization problems with constraints, these techniques are often evaluated based on benchmark test functions instead of challenging real problems. Furthermore, these state-of-the-art constraint-handling techniques concentrate more on the task of optimizing objective function values or reducing constraint violations. It is difficult for these techniques to balance both conflicting tasks. To balance the task of optimization and the task of reducing constraint violations, the constraint-handling technique ε-SEC is proposed in this thesis. ε-SEC preserves search diversity and concurrently imposes selective pressure towards the optimal feasible region in the evolutionary optimization process. Compared with other state-of-the-art constraint-handling techniques, my empirical studies provide evidence that ε-SEC has achieved more effective and robust performance in general. Furthermore, with the optimized parameters, the established model can be used for both replication of collected results and prediction of unseen outcomes.

2. Although the constraint-handling technique ε-SEC can solve the con-

strained parameter optimization problems effectively, "small-data" issues are encountered in some constrained parameter optimization problems, where the optimization is performed given very few data points. To tackle this issue and improve the resolution (in terms of error size) of the model, a data-augmentation pipeline is proposed and tested on several constraint-handling techniques, including ε-SEC. My empirical studies provide evidence that the data-augmentation pipeline can improve the resolution of the model derived by each of these constraint-handling techniques. Therefore, I believe this data-augmentation pipeline is a feasible solution that can be applied to other constrained optimization problems that have small-data issues.

3. Solving high-dimensional problems poses a big challenge for both conventional optimization and meta-heuristic optimization algorithms. Since premature convergence is severe when an optimization problem becomes high-dimensional, even some meta-heuristic optimization algorithms such as Particle Swarm Optimization (PSO) fails to solve the problem due to difficulties in escaping local optima. Good capability in search space exploration plays an important role in getting out of local optima. Although there are several state-of-the-art evolutionary algorithms for solving high-dimensional problems, the exploration capability of these algorithms does not present an overall effective performance. Furthermore, in the field of optimal designs, there is a lack of studies on tackling high-dimensional optimal design problems. It is crucial to fill obvious research gaps in optimal designs in order to handle more realistic cases. In this thesis, the problem of decision variables with high-dimensionality is tackled using a new differential evolution (DE) algorithm. The proposed algorithm NovDE remedies the ineffective exploration capability of other state-of-the-art DE algorithms when handling high-dimensional problems. NovDE

is specially-designed to explore novelty regions in the decision space different from previously explored regions so that the diversity of solutions can be preserved. My empirical studies show that NovDE can handle high-dimensional optimal design problems, which is previously a research gap. This is demonstrated on logistic models on various design spaces and a real application with more demanding requirements.

1.4 Organization

The organization of the remaining chapters is as follows:

Chapter 2 reviews existing state-of-the-art approaches of evolutionary computation related to techniques for constraint-handling and approaches to solving high-dimensional problems. Issues in state-of-the-art approaches are commented upon and research gaps are identified.

Chapter 3 proposes a new evolutionary constraint-handling technique, namely, ε-SEC which can enhance evolutionary search diversity and drive the convergence of solutions towards feasible region. ε-SEC is then deployed on a formulated parameter optimization problem with several constraints and is compared with four other state-of-the-art evolutionary constraint-handling techniques.

Chapter 4 formulates a data-augmentation pipeline to solve small-data issues that are sometimes encountered when doing constrained parameter optimization, like in Chapter 3. Some observations are made on the effectiveness of five constraint-handling techniques when pseudo data are included in optimizing a model, and on the effectiveness of the constraint-handling technique ε-SEC compared with other four techniques.

Chapter 5 describes details of the proposed differential evolution algorithm NovDE which enhances exploration of novelty regions in the decision space different from previously explored regions. NovDE is specially designed to

enhance the diversity of solutions and to improve the capability of escaping from local optima, which is a severe issue in high-dimensional and non-separable problems. NovDE has been implemented on a variety of high-dimensional D-optimal design problems and a real application. NovDE is compared with seven other state-of-the-art differential evolution algorithms to demonstrate its effectiveness in solving high-dimensional and non-separable problems.

Chapter 6 concludes this thesis and discusses possible future research directions.

Chapter 2

Literature Survey

This chapter summarizes and reviews existing research in the field of evolutionary algorithms (EA) that is related to tackling decision variables that are with constraints or are high-dimensional. Current research gaps are identified and insights are sought for proposing new algorithms for solving both constrained optimization problems and high-dimensional problems in real-life domain applications.

2.1 Constraint-handling Techniques in EA

Constraint-handling methods in the field of EA can be classified into four categories, namely, penalty function methods, methods based on preference of feasible solutions over infeasible solutions, methods based on multi-objective optimization techniques and repairing methods as listed in Table 2.1.

2.1.1 Penalty Function Methods

Penalty function methods construct a fitness function by adding a penalty term proportional to the amount of constraint violation into an objective function, and then use this revised objective function to compare the individuals [26]. The

general formulation of penalty function methods is given in Equation (2.1).

$$
\begin{aligned}
&f'(x) = f(x) + \sum_{j=1}^{m} r_j g_j(x) \\
&f(x) : \text{actual objective function;} \\
&r_j : \quad \text{penalty coefficient for constraint j;} \\
&g_j(x) : \text{amount of constraint violation of constraint j corresponding to the individual x;} \\
&f'(x) : \text{revised objective function value after adding penalty.}
\end{aligned}
\tag{2.1}
$$

Penalty function methods have two different categories, viz. static penalty and dynamic penalty. In static penalty, the penalty coefficient r_j is fixed throughout the entire evolutionary process. In dynamic penalty, r_j is changed based on the performance of individuals or the current evolutionary stage. Since penalty coefficients are problem-dependent, the success of the static penalty approach depends on a proper choice of penalty coefficients. These coefficients need prior knowledge about the amount of constraint violation present in a problem. Therefore, the tuning of penalty coefficients leads to a lot of computation for even very simple problems. In the dynamic penalty approach, there are several related works to be reviewed. In [27], the penalty assigned to each individual depends on the generation number G, a scaling constant C and some other hyperparameters in addition to the violation of constraint j, which is referred to as $g_j(x)$. Although the penalty degree for each individual is changed based on the evolution process, the difficulty involved in tuning some additional parameters has limited the applicability of this method. In [28, 29], the r_j for each constraint violation j is altered based on the objective function values of the current population. However, the average or the best of the objective function values of the current population may not be correlated to the degree of penalty, which results in the misleading of the selection pressure towards the feasible region. Since penalty coefficients are sensitive and problem-dependent, it is difficult to find appropriate values of these

penalty coefficients either by static or dynamic penalty approaches.

2.1.2 Methods based on Preference of Feasible Solutions over Infeasible Solutions

Methods based on preference of feasible solutions over infeasible solutions revise the feasibility rule to handle constraints. In [30], ranking is used as the fitness value for each individual. This ranking scheme ranks feasible solutions based on objective function values and ranks infeasible solutions based on constraint violations. The ranking of feasible solutions is superior to the ranking of infeasible solutions so that all feasible solutions dominate infeasible solutions. This method may work well if both feasible and infeasible solutions are present at the start of an evolutionary process. However, if only infeasible solutions are present at the beginning, selective pressure is concentrated entirely on driving all of the solutions into the feasible region at first. The main drawback is that some potential infeasible solutions which can assist in balancing the optimization of objective function and the reduction of constraint violations are overlooked.

To relax some infeasible solutions into feasible ones, the method of ε-differential evolution (ε-DE) is proposed. The feasibility rule of ε-DE [25] works as follows:

Let $\phi(x)$ be the constraint violation of the solution x.

$$\phi(\boldsymbol{x}) = \max_j\{\max\{0, g_j(\boldsymbol{x})\}\}. \tag{2.2}$$

Suppose there are two solutions x_1 and x_2 for comparisons. Based on ε-DE, x_1 is superior to x_2 if any one of these three conditions is satisfied:

i) If $\phi(x_1) < \varepsilon$ and $\phi(x_2) < \varepsilon$, $f(x_1) < f(x_2)$.

ii) If $\phi(x_1) = \phi(x_2)$, $f(x_1) < f(x_2)$.

iii) Otherwise, $\phi(x_1) < \phi(x_2)$.

The introduction of ε may drive near-feasible solutions into the feasible and optimal region. Compared with penalty function methods, there are no additional parameters to be tuned besides ε. In addition, there is no additional computational cost incurred for the operation. However, since only objective function values are compared for ε-feasible solutions (i.e. solutions whose constraint violations are less than ε), any feasible solutions obtained from previous generations are likely discarded due to their inferior objective function values so that the selection pressure towards feasible region may get undermined. Furthermore, for ε-feasible solutions, the comparison of objective function values alone tends to deteriorate the diversity of solutions and results in premature convergence of solutions in the ε-feasible region. Thus, the core issue of this method is, due to the lack of diversity, solutions may converge to a feasible region with inferior objective function values or converge to an infeasible region with superior objective function values.

2.1.3 Methods based on Multi-Objective Optimization Techniques

Methods based on multi-objective techniques convert a single-objective constrained optimization problem into a multi-objective optimization problem by treating the constraints as one or more objectives to be minimized concurrently. The general formulation of this method for bi-objective optimization is given in Equation (2.3).

$$\min_x(f(x), \phi(x)) \tag{2.3}$$

In [31–33], the technique of non-dominated sorting [33] commonly utilized in multi-objective optimization is employed to solve the problem of minimizing both the objective function and constraint violations. The definition of non-dominated solutions is: if there is no other solution x' than x in the decision space

such that $f(x') < f(x)$ and $\phi(x') < \phi(x)$, then x is a non-dominated solution. The non-dominated sorting technique aims to select a set of non-dominated solutions from the candidate solutions. Compared with ε-DE, non-dominated sorting can assist in preserving the feasible solutions obtained in the previous generation. Furthermore, since most non-dominated sorting techniques have the property of preserving the diversity of solutions, the lack of diversity in methods based on preference of feasible solutions over infeasible solutions can somewhat be alleviated. However, the disadvantage of this method is that, due to the utilization of non-dominated sorting, a high computational cost of $O(N^2)$ (N is the population size) according to [33] is incurred, and some useless solutions with better objective function values but worse constraint violations which are considered as non-dominated solutions would also be preserved.

2.1.4 Repairing Methods

Repairing methods use the gradient information from constraint functions to repair infeasible solutions. In [34], the gradients of all violated constraints of an infeasible solution x are calculated and put into a matrix $\nabla_x V$. Then the pseudo-inverse $\nabla_x V^+$ is taken. In each generation g, each infeasible solution x^g is updated as per Equation (2.4) where ΔV is the constraint violation of infeasible solution x^g.

$$x^{g+1} = x^g + (\nabla_x V)^+ \cdot \Delta V \tag{2.4}$$

Based on [34], the update procedure can result in large steps in the decision space. These large steps may lead to quick fixes of constraint violations, but these large steps may also induce possible new violations of constraint functions. In [35], the gradients of constraint surrogates instead of the real constraint functions assist in forming a region to sample potential feasible solutions. The calculation of

Table 2.1: Four Categories of Constraint-handling Techniques in EA

S/N	Category Name	Representative Methods	Issues
1	Penalty function methods	APFEC [36] and FROFI [37]	Penalty parameters are problem-dependent and are difficult to be precisely tuned.
2	Preference methods	ε-DE [25]	Obtained solutions are lack of diversity.
3	Methods based on multi-objective optimization	CMODE [38]	High computational cost is incurred. Some obtained solutions are useless.
4	Reparing methods	COBRA [35]	Require gradient information of constraint functions.

the surrogate gradients and the sampling of possible feasible solutions can help to explore new points in the feasible region so that the performance of [35] is better than [34]. Although repairing methods are efficient in handling constraints, the requirement of the gradients or surrogate gradients of constraint functions restricts its wide applicability.

2.2 High-dimensional Problems in EA

Approaches for tackling high-dimensional optimization in EA can be categorized into two groups: decomposition approaches and non-decomposition approaches. In decomposition approaches, a divide-and-conquer strategy is applied to divide decision variables into smaller subcomponents, and each of which is optimized using a separate EA. Non-decomposition approaches tackle the high-dimensionality problems without decomposition and focus on modifying new operators and introducing structured populations.

2.2.1 Decomposition Approaches

Cooperative co-evolution (CC) proposed by Potter and De Jong [39] is an effective method for solving high-dimensional optimization problems through a divide-and-conquer paradigm. The core idea of CC is to partition the decision variables of an optimization problem into smaller subcomponents, and each of the subcomponents is optimized using a separate EA. Thus, the performance of CC methods is highly dependent on how decision variables get partitioned into different subcomponents. Based on [40], if a problem is separable, which means decision variables are independent from each other, CC methods perform well since how to decompose decision variables becomes trivial. If a problem is non-separable, which means some decision variables have interactions with other decision variables, decomposition of decision variables may be a difficult task, and CC methods may not perform well on non-separable problems.

To tackle these issues in non-separable problems, a random grouping approach is proposed such that a solution x with dimension D is decomposed into k subcomponents whose dimensions are s. The elements of x are then randomly allocated to subcomponents in every co-evolutionary cycle [41]. The random grouping approach aims to have some chances of optimizing interacting variables within the same subcomponent. However, experiment results in [41] show that random grouping performs well only on some non-separable problems which have only several interacting decision variables. When the number of interacting variables grows, it is more difficult to assign these interacting variables into the same subcomponent for optimization. Furthermore, dynamically changing subcomponents without using some prior experiences may result in the divergence of performance. Thus, random grouping is ineffective for problems with large numbers of interacting decision variables.

To utilize prior knowledge of interactions between variables, a differential grouping approach [42] is proposed to achieve an automatic, near-optimal de-

composition of decision variables. Differential grouping tries to identify whether two decision variables are interacting first and then groups these interacting variables into the same subcomponent. The process of the pairwise examination of interactions is as follows:

Let $\overrightarrow{p_1}$ and $\overrightarrow{p_2}$ are two sets of solutions. In order to check for interaction between the ith and the jth dimensions, the vector $\overrightarrow{p_2}$ is set to be equal to $\overrightarrow{p_1}$ except the ith dimension. The ith variable of $\overrightarrow{p_2}$ is set to the upper bound of the search space for that dimension. Then Δ_1 is calculated as $\Delta_1 = f(\overrightarrow{p_1}) - f(\overrightarrow{p_2})$. Then, the jth element of $\overrightarrow{p_2}$ is set to the center of the search space for that dimension and Δ_2 is calculated in the same manner as the calculation of Δ_1. If $|\Delta_1 - \Delta_2|$ is greater than a pre-defined threshold ϵ, then the i-th and j-th dimensions interact with each other and they are grouped into the same subcomponent. After all subcomponents are formed, each subcomponent is optimized by a separate EA. Empirical studies in [42] show that the differential grouping approach outperforms the random grouping approach when tackling high-dimensional non-separable problems. However, the differential grouping approach fails to handle some non-separable problems due to premature convergence. Since the effectiveness of the differential grouping approach is problem-dependent, this approach may not be generally applicable to all non-separable problems. In addition, due to the pairwise examination of interacting variables, differential grouping approach has higher computational complexity compared to the random grouping approach, and the complexity is even larger as solution dimension grows.

To sum up, decomposition approaches are effective when tackling high-dimensional separable problems. To solve high-dimensional non-separable problems, decomposition approaches have two main issues: first, high computational complexity is incurred since the optimization of each solution x is equivalent to the optimization of several subcomponent groups of x; second, the grouping strategy is problem-dependent, and it is difficult to propose a general grouping

strategy with robust and effective performance.

2.2.2 Non-Decomposition Approaches

Non-decomposition approaches optimize a high-dimensional problem as a whole, and no divide-and-conquer methods are involved. These non-decomposition approaches usually focus on alterations such as modifying operators and introducing structured populations [8, 43].

In terms of modifying operators, there are several related works in literature. In [44], jDElsgo is proposed to alter the DE control parameters F and CR adaptively based on the uniform distribution within [0,1] so that widely distributed values of control parameters can help to obtain diverse solutions in the decision space. In [45], JADE uses a DE mutation strategy 'DE/current-to-pbest' and tunes F and CR adaptively according to the Cauchy distribution based on previous experiences. In [46], ANDE applies a proposed triangular DE mutation strategy so that the mutated vector of a solution is a combination of the best, the medium, and the worst directions derived from any three solutions. For DE, modifying operators involves proposing new mutation or crossover strategies and control parameters adaptation schemes. However, new strategies and new adaptation schemes are effective for early or medium stages of an evolutionary process and ineffective for late stages. At late stages, solutions tend to be similar and close to global or local optimal. New solutions generated based on new strategies or new control parameters adaptation schemes have fewer chances to replace current solutions.

In terms of EA with structured populations, populations are divided into several subpopulations and each subpopulation evolves independently. According to some specific probability, solutions in different subpopulations are exchanged to preserve population diversity, which is referred to as migration [47]. To prevent premature convergence, some exchange methods are proposed accordingly. In

[48], a new migration scheme is proposed to replace the oldest solution of each subpopulation instead of the worst one. In [49], a novel migration scheme is proposed to control the substitution of solutions to occur only when the new solution is better than the one chosen to be replaced. In [50], inspired by a phenomenon known as biological invasion, solutions with better fitness values than the average fitness value of the same subpopulation can enter neighboring subpopulations as invasion. Although these migration schemes introduce some new solutions from other subpopulations to the current one, an important issue is that new solutions are not ensured to be relatively different from the current subpopulations and may not assist in preserving or enhancing the diversity of the current subpopulations. The number of new solutions and the diversity of these new solutions are crucial to preserving and enhancing subpopulation diversity.

To sum up, non-decomposition approaches are effective when tackling high-dimensional non-separable problems but these approaches are still problem-dependent and are not robust to tackle various problems. Compared with decomposition approaches, non-decomposition approaches have advantages of simple operations and lower computational costs, which are essential for solving high-dimensional problems.

Chapter 3

An Evolutionary Constraint-Handling Technique for Parameter Optimization of a Cancer Immunotherapy Model

3.1 Introduction

Many real-world modeling problems can be solved with the help of parameter optimization based on data collected from physical experiments, real events, or complex numerical simulations [51]. Depending on the collected data and some criteria, some objective functions with constraints can be formulated. Solving an optimization problem based on the collected data and the formulated objective function(s) is known as data-driven optimization. There are two types of such data-driven optimization approaches; on-line data-driven optimization for which new data is acquired during the optimization process, such as aerodynamic shape optimization [52, 53]; and off-line data-driven optimization for which no data is acquired during the optimization process, such as optimizing a trauma

system design which can only use incidents in previous years and cannot use new incidents collected during the optimization process [54, 55]. In this chapter, data from cancer clinical trials are collected for the further study. In most cancer clinical trials, due to cost and time constraints, only off-line limited clinical results can be collected. This work presents a case study of off-line data-driven parameter optimization for a cancer immunotherapy model. The cancer immunotherapy uses the immune system to defend human bodies against attacks by malignant tumor cells [56]. To replicate and predict therapeutic effects, it is essential to establish a bio-dynamic model based on biological principles and to formulate an objective function with constraints for optimizing its model parameters based on limited clinical results. Subsequently, the established model can provide significant inferences for further clinical trials.

A mathematical model has been established, based on biological principles, to express disease progression either with or without drug treatments where two drugs cyclophosphamide (CY) and a Toll-like receptor agonist (CpG) are used [57]. Although the dynamic system model is established, it is difficult to find a set of appropriate parameter values since a dynamic system model is sensitive to parameter settings. In the literature [58–62], two approaches are commonly used to determine the values of immunotherapy model parameters. In [58], some parameters are qualitatively determined and estimated based on biological principles. This approach is effective for models based on similar experiment settings and with fewer estimated parameters. In the proposed model, there are 15 parameters to be optimized, and this approach may not work effectively. In [58], some parameters are empirically determined via a trial-and-error approach with clinical data. Since this approach requires extensive efforts in manual fine-tuning, it is neither efficient nor robust for optimizing such a model.

Evolutionary algorithm (EA) is a meta-heuristic global optimization approach which can be applied to solve optimization problems with single or multiple ob-

jectives [63–65]. EA has shown much success in solving a variety of optimization problems in real-world applications [66–71]. Considering these advantages, EA is applied in this chapter to solve the formulated off-line data-driven parameter optimization problem with constraints.

Constraint-handling methods in EA can be categorized into four types, namely, penalty function methods, methods based on preference of feasible solutions over infeasible solutions, methods based on multi-objective optimization techniques and repairing methods [3]. They have been explicitly introduced and reviewed in Chapter 2.

The contributions of this chapter are summarized as follows,

1) A cancer immunotherapy model is proposed based on biological principles and the clinical results in [72]. To the best of our knowledge, this work is among the first work to conduct parameter optimization in cancer immunotherapy.

2) A new constraint-handling technique ε-SEC is proposed to preserve search diversity and to impose selective pressure towards the optimal feasible region in an evolutionary optimization process. Thus, decision variables with constraints are studied and explored to tackle real problems.

3) With the optimized parameter values, the formulated cancer immunotherapy model can be used for prognostic outcomes in clinical trials. The predictability is considered significant for such a parameter optimization approach.

3.2 Technical details of the Mathematical Model

Breast Cancer Immunotherapy Model

Based on biological principles and the clinical results in [72], a mathematical model for the breast cancer intratumoral immune system is established. The model expresses the change of the entities with and without treatments. CY and CpG are two different kinds of drugs for treatments mentioned in [72]. They

can eradicate advanced mouse tumors. The model takes four major entities: breast cancer cells (C), effector cells (E), regulatory cells (R), and tumoricidal myeloid CD11b+Gr1dim cells (M). The model assumes a homogeneous tumor cell population.

Some specific biology assumptions are made below according to biological principles [58, 59, 72, 73]:

(1) C cells grow logistically in the absence of treatments.

(2) E cells are capable of killing cancer cells. E cells are activated by cancer cells, but eventually become inactivated after some number of interactions with tumor cells.

(3) R cells suppress E cells.

(4) M cells can be induced by CY and CpG to kill cancer cells.

The model inputs are the initial cell counts of these four cell types, the current time t, and the dosage of CpG and CY at time t, which are represented by $V_{CpG}(t)$ and $V_{CY}(t)$ in the units of 'mg/kg'. The model outputs are the cell counts of these four cell types at time t, respectively.

In the model, multiplication rules are applied for the interaction between two kinds of cells since multiplication rules have been widely adopted in system biology to model interactions between cells. Similarly, the multiplication rules have been used to model interaction between cells and drugs as well.

Tumor Cells C

The evolution of the population of tumor cells depends on the cancer growth terms and the removal of cancer cells by immune cells. The change of tumor cells is expressed as:

$$\frac{dC}{dt} = a_c C(1 - \frac{C}{C_{\max}}) - b_c EC - c_c MC - d_c V_{CY}(t)C. \tag{3.1}$$

The growth term is represented by $a_c C(1 - C/C_{max})$. C_{max} is the maximum ca-

pacity of tumor cells. The removal terms are $-b_cEC$, $-c_cMC$, and $-d_cV_{CY}(t)C$. They represent the removal of tumor cells by the interactions with E cells and M cells. Since CY is an anti-cancer chemotherapy drug, it can weaken or destroy cancer cells. Hence, $-d_cV_{CY}(t)C$ is the last removal term.

Effector Cells E

The change of effector cells (E) is represented as:

$$\frac{dE}{dt} = a_eC - b_eE - c_eEC - d_eRE. \tag{3.2}$$

The term a_eC characterizes the rate at which E cells accumulate in the tumor micro-environment due to the presence of tumor cells. The term $-b_eE$ characterizes the E cells dying naturally at a rate of b_e. The term $-c_eEC$ is the inactivation rate due to the interaction between tumor cells and E cells. The term $-d_eRE$ represents the E cells suppressed by the regulatory cells.

Regulatory Cells R

The equation for regulatory cells is expressed as follows:

$$\frac{dR}{dt} = a_rE - b_rR - k_1RV_{CY}(t) - k_2RV_{CpG}(t). \tag{3.3}$$

R cells originate from E cells [74], which are described by the term a_rE. The term $-b_rR$ represents the natural death of R cells. Since CY has a property to selectively kill R cells [75–78], the term $-k_1RV_{CY}(t)$ is applied to indicate the interaction between R cells and CY. Since the number of R cells decreases after injection of CpG, the term $-k_2RV_{CpG}(t)$ is applied.

Tumoricidal Myeloid CD11b+Gr1dim Cells M

The tumoricidal myeloid CD11b+Gr1dim cells (M) are represented as:

$$\frac{dM}{dt} = -a_mM + k_3V_{CY}(t) + k_4V_{CpG}(t). \tag{3.4}$$

The death of M cells is represented by $-a_m M$. The terms $k_3 V_{CY}(t) + k_4 V_{CpG}(t)$ describe the increase of M cells by CpG+CY treatment.

Remark: Comparing with a static function, the impact from a parameter to the output response is much more complicated for a dynamic system. First, the output response of a dynamic system is not only dependent on the parameters, but also on the initial conditions as well as exogenous inputs, such as CY and CpG in our model. Thus, extra dimensions are added to the original optimization problem. Second, the relations between dynamic output response and system parameters are far more complicated than that of a stationary mapping. For instance, the gradient of the output of a stationary function with respect to a parameter is finite. In a dynamic system, however, the variation of a parameter can result in many complex behaviors in corresponding solution trajectories, for instance, stable or unstable response, bifurcation, subharmonics, finite escape time, and even chaos. The gradient of a parameter can be as steep as infinite. In other words, the dynamic response can be extremely sensitive to parameter variations. Hence, parameter optimization becomes much more challenging for dynamic systems.

3.3 Technical details of the Formulation of the Parameter Optimization Problem

3.3.1 Decision Variables

There are 16 parameters in the model. C_{max} is fixed as 4×10^9. According to the experiments in [72], the tumor weight of the mouse is approximately $4g$. The volume of $1g$ tumor is about $1cm^3$. There are around 1×10^9 tumor cells per cm^3. Hence, the maximum tumor carrying capacity C_{max} is 4×10^9. All the other 15 parameters constitute the solution $\boldsymbol{x}$. Hence, $\boldsymbol{x}$ =

$(a_c, b_c, c_c, d_c, a_e, b_e, c_e, d_e, a_r, b_r, k_1, k_2, a_m, k_3, k_4)$.

Each parameter has certain biological meaning. The lower bound (X_{lb}) and upper bound (X_{ub}) of each parameter are manually set as one order of magnitude lower and one order of magnitude higher than the magnitude of the estimated values from the literature [58, 60–62, 72] except for a_c since a_c cannot be too large due to its biological meaning in the model. The upper and lower bound of each parameter are presented in Table 3.1.

Table 3.1: Upper and Lower Bound of 15 Parameters

Parameters	**Units**	$\boldsymbol{X_{lb}}$	$\boldsymbol{X_{ub}}$
a_c	day^{-1}	1×10^{-1}	5×10^{0}
b_c	$(cell\ \ day)^{-1}$	1×10^{-7}	1×10^{-5}
c_c	$(cell\ \ day)^{-1}$	1×10^{-7}	1×10^{-5}
d_c	$(cell\ \ day)^{-1}$	1×10^{-7}	1×10^{-5}
a_e	day^{-1}	1×10^{-5}	1×10^{-3}
b_e	day^{-1}	1×10^{-3}	1×10^{-1}
c_e	$(cell\ \ day)^{-1}$	1×10^{-11}	1×10^{-9}
d_e	$(cell\ \ day)^{-1}$	1×10^{-7}	1×10^{-5}
a_r	day^{-1}	1×10^{-6}	1×10^{-4}
b_r	day^{-1}	1×10^{-4}	1×10^{-2}
k_1	$mg^{-1}\ kg\ day^{-1}$	1×10^{-4}	1×10^{-2}
k_2	$mg^{-1}\ kg\ day^{-1}$	1×10^{-1}	1×10^{1}
a_m	day^{-1}	1×10^{-3}	1×10^{-1}
k_3	$mg^{-1}\ kg\ cell\ day^{-1}$	1×10^{1}	1×10^{3}
k_4	$mg^{-1}\ kg\ cell\ day^{-1}$	1×10^{3}	1×10^{5}

3.3.2 Objective Function

The experiment in [72] was conducted when CY was given to cancer-bearing mice on day 0 and CpG on day 3 of each seven day cycle. Tumors were allowed to reach 10-12 mm diameter prior to commencement of therapy on day 17. In

this chapter, the tumor diameter is assumed to reach the medium value $11mm$ on day 17. Then the volume of tumor is 696.91mm^3, calculated by the formula $V=\frac{4\pi r^3}{3}$ where V is the tumor volume and r is the tumor radius. Since 1 mm^3 tumor has 1×10^6 tumor cells according to [72], the tumor cells on day 17, i.e. the initial tumor cell counts C_0 is 6.9691×10^8. In addition, based on the experiments from [72] as well, the yield of E cells per mm^3 tumor is 500 and the yield of R cells per mm^3 tumor is 150, so the initial E cell counts and R cell counts are $E_0 = 500 \times 696.91 = 3.4845 \times 10^5$ and $R_0 = 150 \times 696.91 = 1.0454 \times 10^5$. The initial cell counts are shown in Table 3.2.

Table 3.2: Initial Conditions of Number of Tumor Cells C, Effector Cells E, and Regulatory Cells R

C_0	E_0	R_0
6.9691×10^8	3.4845×10^5	1.0454×10^5

Also based on experiments from [72], CpG, CY and CpG+CY treatments start to be conducted on day 17, respectively. After 10 days of treatments (i.e. on day 27), tumor volume reduces to approximately four fifths (4/5) of day 17, two thirds (2/3) of day 17 and a fourth (1/4) of day 17, respectively. That is, the corresponding tumor volumes are 557.53 mm^3, 464.61 mm^3 and 174.23 mm^3, respectively. On day 27, for CpG therapy, the yield of E cells per mm^3 is 90 and the yield of R cells per mm^3 is 4; for CY therapy, the yield of E cells per mm^3 is 150 and the yield of R cells per mm^3 is 20; for CpG+CY therapy, the yield of E cells per mm^3 is 70 and the yield of R cells per mm^3 is 0.13. In addition, in [72], at least 90% eradication of the tumor cells is estimated after 7 cycles with CpG+CY therapy. Thus, the the corresponding tumor volumes are smaller than 17.42 mm^3 on day 66 (=17+(7$\times$7)). Thus, there are 4 different drug schedules contributing to 10 collected clinical results as shown in Table 3.3.

Limited clinical results were collected before the optimization process, and an off-line data-driven optimization problem is formulated based on the clinical

Table 3.3: C Cells, E Cells and R Cells Counts with CpG, CY and CpG+CY Treatments from the Clinical Results in [72]

	C	**E**	**R**
CpG (Day 27)	5.5753×10^8	5.0178×10^4	2.2301×10^3
CY (Day 27)	4.6461×10^8	6.9692×10^4	9.2922×10^3
CpG+CY (Day 27)	1.7423×10^8	1.2196×10^4	22.65
CpG+CY (Day 66)	$\leq1.7423\times10^7$	N.A.	N.A.

results and biological principles. The aim is to minimize errors between the computed results from the model and the clinical results in [72]. The sum of absolute error is selected as the objective at first. Normalization is then applied because the number of C cells, E cells and R cells are in different order of magnitudes. In [72], the order of magnitudes of C cells, E cells and R cells are 10^8, 10^4 and 10^3, respectively. Hence, the normalization used for the differences of C cells, E cells, R cells are 10^{-8}, 10^{-4} and 10^{-3}, respectively.

It should be noted that if the sum of absolute error is the objective function, the computed results of C cells, E cells and R cells for CpG+CY on day 27 from our model are frequently one order of magnitude lower than the corresponding clinical results in [72]. Since these clinical results are close to zero after the normalization, the objective function can be insensitive to the change of errors between the clinical results and computed results from the model. To make the objective function sensitive to the change of errors, absolute log ratio function is applied to these terms.

In this work, I set the threshold to be 2.5 so that if the clinical results after normalization are smaller than 2.5, absolute log ratio will be applied to that term. Otherwise, absolute error will be applied.

The inputs of the objective function are the clinical results of the cell counts under different drug schedules at time t, and the computed cell counts from the established cancer immunotherapy model under different drug schedules at time

t. The outputs of the objective function are the sum of the errors between the clinical results and computed results under different drug schedules.

The objective function is thus formulated as follows:

$$\begin{aligned} min\ f(\boldsymbol{x}) = & f_{CpG}(\boldsymbol{x}, t = 27) + f_{CY}(\boldsymbol{x}, t = 27) \\ & + f_{CpG+CY}(\boldsymbol{x}, t = 27) \\ & + f_{CpG+CY}(\boldsymbol{x}, t = 66) \end{aligned} \tag{3.5}$$

where

$$\begin{aligned} f_{CpG}(\boldsymbol{x}, t = 27) = & \\ & |(C_{CpG}(\boldsymbol{x}, t = 27) - \widehat{C}_{CpG}(\boldsymbol{x}, t = 27))|/10^8 \\ & + |(E_{CpG}(\boldsymbol{x}, t = 27) - \widehat{E}_{CpG}(\boldsymbol{x}, t = 27))|/10^4 \\ & + |\log(R_{CY}(\boldsymbol{x}, t = 27)/\widehat{R}_{CY}(\boldsymbol{x}, t = 27))|, \end{aligned}$$

$$\begin{aligned} f_{CY}(\boldsymbol{x}, t = 27) = & \\ & |(C_{CY}(\boldsymbol{x}, t = 27) - \widehat{C}_{CY}(\boldsymbol{x}, t = 27))|/10^8 \\ & + |(E_{CY}(\boldsymbol{x}, t = 27) - \widehat{E}_{CY}(\boldsymbol{x}, t = 27))|/10^4 \\ & + |(R_{CpG}(\boldsymbol{x}, t = 27) - \widehat{R}_{CpG}(\boldsymbol{x}, t = 27))|/10^3, \end{aligned}$$

$$\begin{aligned} f_{CpG+CY}(\boldsymbol{x}, t = 27) = & \\ & |\log(C_{CpG+CY}(\boldsymbol{x}, t = 27)/\widehat{C}_{CpG+CY}(\boldsymbol{x}, t = 27))| \\ & + |\log(E_{CpG+CY}(\boldsymbol{x}, t = 27)/\widehat{E}_{CpG+CY}(\boldsymbol{x}, t = 27))| \\ & + |\log(R_{CpG+CY}(\boldsymbol{x}, t = 27)/\widehat{R}_{CpG+CY}(\boldsymbol{x}, t = 27))|, \end{aligned}$$

$$\begin{aligned} f_{CpG+CY}(\boldsymbol{x}, t = 66) = & \\ & |\log(C_{CpG+CY}(\boldsymbol{x}, t = 66)/\widehat{C}_{CpG+CY}(\boldsymbol{x}, t = 66))|, \end{aligned}$$

where $\boldsymbol{x}$ denotes the set of parameters in the model. The terms with hat represent the clinical results in [72], and the terms without hat represent the computed results from the model.

3.3.3 Constraints

For the proposed nonlinear biological system as in Equation (3.1) to (3.4), the stability criteria need to be satisfied for CpG on day 27, CY on day 27, the combination of CpG+CY on day 27 and the combination of CpG+CY on day 66. The stability of the equilibrium point without treatment should also be satisfied. A total of 23 constraints are formulated to satisfy the system equilibrium and disease progression characteristics. 20 of them are based on bio-stability, and they are handled by the death penalty method. In control theory, a continuous-time linear time-invariant system is stable if and only if all of the eigenvalues λ_i of the system matrix have negative real parts. The constraints of bio-stability are formulated such that under various drug schedules as well as equilibrium condition, eigenvalues of the system matrix should have negative real parts. The death penalty method is the simplest way to handle such constraints, but it requires the feasible region to be easily identified [79]. Based on the results derived from 30 independent runs by the death penalty method, more than 70% generated solutions of each generation satisfy the constraints of bio-stability for the first 100 out of 500 generations. After half of an evolutionary process, all solutions of each generation satisfy the constraints of bio-stability. Thus the death penalty method is effective in handling the constraints of bio-stability.

The other 3 constraints are based on biological principles. Based on our empirical studies from 30 independent runs, feasible solutions cannot be generated by the trivial death penalty method so these 3 constraints are more suitable to be handled by non-trivial constraint-handling methods under the four categories described in Chapter 2 [79].

To satisfy the disease progression without treatments, 3 constraints should be formulated according to [58, 59, 72, 73]. Without any treatments, the tumor cells (C) should not decrease at the starting point. The first constraint is thus $dC/dt \geq 0$ at t = 17. Without any treatments, E cells should to not increase at

the starting point. The second constraint is thus $dE/dt \leq 0$ at t = 17. However, E cells should not decrease so rapidly at the starting point. I approximate the rate of change at the starting point using the difference of E cells between two close but not consecutive days (day 17 and day 20 used in this work) divided by the difference in days. I manually restrict the minimum E cells counts on day 20 as 5×10^4, and the lower bound of the rate of the change of E cells on day 17 is -9.9485$\times10^4$ (=(5$\times10^4$-E_0)/(20-17)) where E_0 is from Table 3.2. The third constraint is thus $dE/dt \geq -9.9485 \times 10^4$ at t = 17.

In [25], the constraint violation can be defined as the maximum of all constraints or the sum of all constraints, and the maximum of all constraints was selected to be the constraint violation. In this work, since the constraint-handling technique is proposed based on [25], the constraint violation is also defined as the maximum of all constraints:

$$\phi(\boldsymbol{x}) = \max_j\{\max\{0, g_j(\boldsymbol{x})\}\}, \tag{3.6}$$

where $g_j(\boldsymbol{x})$ is the constraint value of each constraint j for solution x. If some constraint j is violated, $\phi(\boldsymbol{x}) > 0$. Otherwise, $\phi(\boldsymbol{x}) = 0$.

Since the magnitudes of these three constraints are different, normalization should be applied to them. In Table 3.3, the order of magnitudes of C cells and E cells are 10^8 and 10^4. Hence, the normalization used for the constraints related to C cells and E cells is 10^{-8} and 10^{-4}, respectively. All the 23 constraints are

formulated as follows:

$$
\begin{aligned}
&g_{i1}(\boldsymbol{x}) = \lambda_j(\boldsymbol{x}, t = 27, CpG) < 0, where\ i1 = \{1, 2, 3, 4\},\\
&and\ j = \{1, 2, 3, 4\}.\\
&g_{i2}(\boldsymbol{x}) = \lambda_j(\boldsymbol{x}, t = 27, CY) < 0, where\ i2 = \{5, 6, 7, 8\},\\
&and\ j = \{1, 2, 3, 4\}.\\
&g_{i3}(\boldsymbol{x}) = \lambda_j(\boldsymbol{x}, t = 27, CpG + CY) < 0,\\
&where\ i3 = \{9, 10, 11, 12\}, and\ j = \{1, 2, 3, 4\}.\\
&g_{i4}(\boldsymbol{x}) = \lambda_j(\boldsymbol{x}, t = 66, CpG + CY) < 0,\\
&where\ i4 = \{13, 14, 15, 16\}, and\ j = \{1, 2, 3, 4\}.\\
&g_{i5}(\boldsymbol{x}) = \lambda_j(\boldsymbol{x}, equilibrium) < 0,\\
&where\ i5 = \{17, 18, 19, 20\}, and\ j = \{1, 2, 3, 4\}.\\
&g_{21}(\boldsymbol{x}) = -(\frac{dC(w/o\ therapy)(\boldsymbol{x})}{dt}\Big|_{t=17})/10^8 \leq 0,\\
&g_{22}(\boldsymbol{x}) = (\frac{dE(w/o\ therapy)(\boldsymbol{x})}{dt}\Big|_{t=17})/10^4 \leq 0,\\
&g_{23}(\boldsymbol{x}) = -9.9485 \times 10^4/10^4 -\\
&(\frac{dE(w/o\ therapy)(\boldsymbol{x})}{dt}\Big|_{t=17})/10^4 \leq 0,
\end{aligned}
\tag{3.7}
$$

where λ_1 to λ_4 are the four eigenvalues of the system matrix under various drug schedules and equilibrium condition.

3.4 Technical Details of the Proposed Constraint-Handling Technique ε-SEC

A constraint-handling technique ε-Sectional (ε-SEC) is proposed to solve the data-driven constrained optimization problem formulated in Section 3.3. The proposed ε-SEC remedies drawbacks in methods based on preference of feasible solutions over infeasible solutions, especially ε-DE.

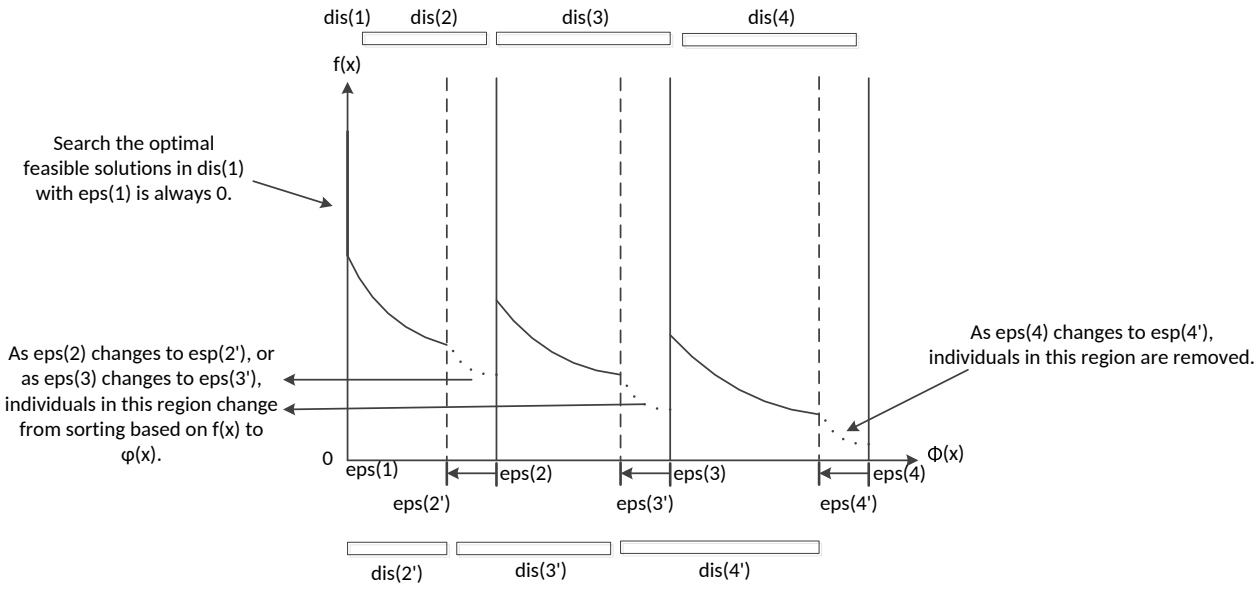

Figure 3.1: Working diagram of ε-SEC. In this example, the number of sections is M=4. dis(1),dis(2), dis(3), and dis(4) represent the original region of these four sections. dis(1) always represents the feasible region. eps(1), eps(2), eps(3), and eps(4) represent the upper bounds of the corresponding sections. Individuals are partitioned into different sections based on their constraint violations $\phi(\boldsymbol{x})$. In each section, half of the individuals are selected based on $f(\boldsymbol{x})$ and the other half based on $\phi(\boldsymbol{x})$ so that part of the Pareto front in each section can be approximated. Each section dynamically shifts close to the feasible region. For instance, as dis(4) changes to dis(4'), the solid curve remains. The dotted curve in the lower right of the remained solid curve diminishes since the upper bound changes from eps(4) to eps(4').

3.4.1 Proposed Constraint-Handling Technique-ε-SEC

According to ε-DE in [25], a soft constraint-tolerance ε is introduced to assist the search for feasible solutions. The soft constraint-tolerance ε is set to converge to zero as the number of generation is increased. If constraint violation is smaller than ε, only objective function values are compared. Otherwise, only constraint violations are compared. If we compare an infeasible solution whose constraint violation is no larger than ε with a feasible solution, the infeasible solution may be superior due to the more optimal objective function value.

To circumvent the issue of ε-DE discussed in Chapter 2, a new constraint-handling technique of ε-SEC is proposed in this Chapter. In ε-SEC as shown

in Fig. 3.1, several consecutive sections are formed based on the value of ε with respect to the current generation g, and one of these sections is fixed as the feasible region. The value of ε is also set to converge to zero as the number of generation is increased. To enhance diversity of population, both parent solutions and offspring are distributed into different sections based on their constraint violations. The selection operation is conducted within each section to collectively form the population in the next generation. To enhance convergence of population, the search in the current section can provide information for the search process in neighboring sections. Furthermore, since ε is reduced towards 0, the upper bounds of each section are reduced towards 0 accordingly. Thus, each section is dynamically shifted towards the feasible region. Thereby ε-SEC imposes selective pressure on individuals in each section towards the optimal feasible region to enhance convergence. To balance objective function optimization and constraint violation minimization, in each section, half of the individuals are selected based on their objective function values, and the other half are selected based on their constraint violations since the objective function and constraint violations are typically in conflict with each other.

The constraint violation $\phi(\boldsymbol{x})$ of each individual $\boldsymbol{x}$ is defined in Equation (3.6). The soft constraint tolerance $\varepsilon(g)$ with respect to the current generation g is defined the same as in ε-DE [25]:

$$\begin{aligned} \varepsilon(0) &= \phi(\boldsymbol{x_\theta}) \\ \varepsilon(g) &= \begin{cases} \varepsilon(0)(1-\frac{g}{G_c})^{cp}, 0 < g < G_c \\ 0, g \geq G_c \end{cases} . \end{aligned} \tag{3.8}$$

$\phi(\boldsymbol{x_\theta})$ is θ percentage of violation in descending order of the entire initialized individuals. g represents the current generation. G_c is the threshold of the generation. cp controls the convergence speed of $\varepsilon(g)$, and $cp = (-5 - \log(\varepsilon(0)))/\log 0.05$.

Algorithm 3.1 presents the proposed constraint-handling technique ε-SEC.

The constraint violation is divided into M sections which are consecutive to each other. It is noted that for the setting of ε-SEC, the first district always represents the feasible region. Hence, for $M = 1$, there is only one district, and the proposed technique is equivalent to the death penalty method. For $M = 2$, there are two districts such that the first is the feasible region, and the second is the region with $\phi(x) \leq \varepsilon(g)$, where g is the current generation. Hence, for $M = 2$, ε-SEC is equivalent to ε-DE. In order to enhance diversity of the population distribution, the setting of M should be $M \geq 3$. From steps 3 to 5, the upper bound of each section s is determined. The upper bound of each section is the corresponding ε values in Equation (3.8) with respect to $h(1)$, $h(2)$, $\cdots$, $h(M)$, which are set in step 3. The districts of these M sections are represented as $dis(1), \cdots, dis(M)$ in Fig. 3.1. The upper bound of each section s is $eps(s)$ as shown in Fig. 3.1.

The most conventional differential evolution (DE) strategy-'DE/rand/1/bin' is employed to generate N offspring based on the population in the current generation. Hence, the size of P_{com} is $2N$, and it is the same size for f_{com} and ϕ_{com}. P_{com} are distributed based on ϕ_{com} into M sections and the region with $\phi(x) > eps(M)$. The original desired population size for each section s is *popsize* $= N/M$. After the distribution of P_{com}, the population size of each section s is counted and represented as $cursz(s)$. If there are more than N individuals distributed to M sections, only N individuals will be selected from these M sections for the next generation. Otherwise, all of the individuals distributed in these M sections will be selected for the next generation. From steps 12 to 31, the procedure of selecting the elitist N individuals for the next generation is presented. From steps 13 to 25, the $selectsz(s)$, which is the number of individuals selected for the next generation, is determined for each section s. It is desired to select equal number of individuals denoted as *popsize* from each section s so that the individuals can be evenly distributed. However, at the early stages of evolution, there may be fewer than *popsize* individuals distributed to sections

close to the feasible region. Thus, $selectsz(s)$ is initialized to be the minimum of $cursz(s)$ and *popsize*. The number of potential individuals for selection into the next generation is calculated as $leftsz(s)$ for each section s. The number of individuals to be selected from subsequent sections are calculated and updated as $remain$. From steps 17 to 25, $leftsz(s)$ is compared with $remain$ starting from the feasible region to the M-th section to check whether the number of potential individuals $leftsz(s)$ of section s should be selected into P_{g+1} so that $selectsz(s)$ can be updated and determined. After $selectsz(s)$ is determined for all the M sections, the selection of individuals into the next generation is presented in step 26. For each section s, candidates are ranked in ascending order based on the objective function values $f(x)$ at first. Candidates are then ranked in ascending order based on constraint violations $\phi(x)$. Candidates ranking before $selectsz(s)/2$ based on $f(x)$ are selected for P_{g+1}. Subsequently, candidates ranking before $selectsz(s)/2$ based on $\phi(x)$ are selected for P_{g+1} as well.

3.4.2 Analysis of ε-SEC

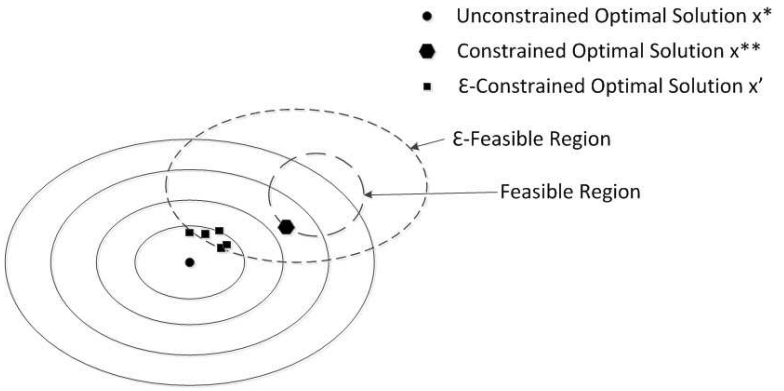

Figure 3.2: Schematic of ε-DE. Ellipses represent the contours of objective function. The outer dotted ellipse represents the contour of ε-feasible region in the objective space. The inner dotted ellipse represents the contour of the feasible region in the objective space.

One issue of ε-DE is that solutions tend to be crowded in the infeasible region at the early evolutionary stages. If $\phi(\boldsymbol{x})$ is smaller than or equal to $\varepsilon(g)$, only

Algorithm 3.1 ε-SEC

Input: population size N, maximum generation G_{max}, threshold generation G_c, $\varepsilon(g)$, section size M, current population P_g, combined population P_{com}, combined objective function values f_{com}, combined constraint violations ϕ_{com}.

Output: population P_{g+1}

for $g = 1, 2, \cdots, G_{max}$ **do**
  if $g <= G_c$ **then**
    $h(s) = floor(G_c - (G_c - g)/(M-1) \times (s-1))$, where s is the corresponding section index, and $s = 1, 2, \cdots, M$.
    $eps(s) = \varepsilon(0)(1 - h(s)/G_c)^{cp}$, for $s = 1, 2, \cdots, M$.
    $dis(s)$ represents each section s. The lower and upper bound of $dis(s)$ is $eps(s-1)$ and $eps(s)$. $eps(0) = eps(1) = 0$.
  else
    $eps(s) = 0$ for $s = 1, 2, \cdots, M$.
  end if
  P_{com} are distributed into M sections and the region with $\phi(\boldsymbol{x}) > \varepsilon(g)$ based on the $\phi(\boldsymbol{x})$ in ϕ_{com}.
  The current population size of each section s is denoted as $cur\_sz(s)$.
  The required population size of each section s is denoted as $popsize = N/M$.
  if $cursz(1) + \cdots + cursz(M) > N$ **then**
    For each s in $s = 1, 2, \cdots, M$:
    $selectsz(s) = min(cursz(s), popsize)$.
    $leftsz(s) = max(0, cursz(s) - popsize)$.
    $remain = N - (selectsz(1) + \cdots + selectsz(M))$.
    for $s = 1, \cdots, M$ **do**
      if $remain <= leftsz(s)$ **then**
        $selectsz(s) = selectsz(s) + remain$.
        break.
      else
        $selectsz(s) = selectsz(s) + leftsz(s)$.
        $remain = remain - leftsz(s)$.
      end if
    end for
    For each s, rank the individuals in $dis(s)$ in ascending order based on f and ϕ, respectively. Individuals ranking before $selectsz(s)/2$ in terms of f are selected into P_{g+1}. Similarly, individuals ranking before $selectsz(s)/2$ in terms of ϕ are selected into P_{g+1}.
  else
    All of the individuals in P_{com} with $\phi(\boldsymbol{x}) <= \varepsilon(g)$ are selected into P_{g+1}.
    $remain = N - (cursz(1) + \cdots + cursz(M))$.
    Rank the individuals with $\phi(\boldsymbol{x}) > \varepsilon(g)$ in ascending order based on $\phi(\boldsymbol{x})$. Select the individuals ranking before $remain$ into P_{g+1}.
  end if
end for

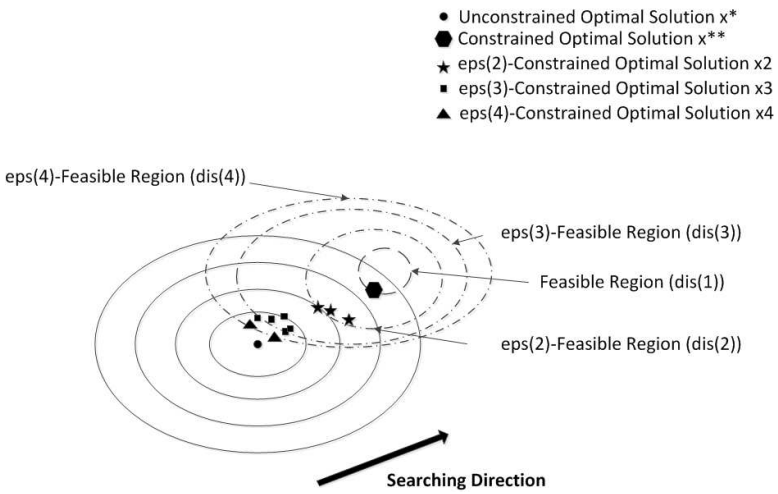

Figure 3.3: Schematic of ε-SEC. In this example, the number of sections is M=4. The ellipses represent the contours of objective function. The dotted ellipses represent the contours for dis(1), dis(2), dis(3), and dis(4) in the objective space. The searching direction arrow indicates that the searching in the outer section can promote the searching in the neighboring inner section. The searching in these four sections collaboratively drives the solutions into the feasible region dis(1).

the objective function values are compared. Hence, as shown in Fig. 3.2, in ε-feasible region, those ε-constrained optimal solutions x' tend to be crowded in the objective space, and the diversity of the solutions are deteriorated. As the value of $\varepsilon(g)$ becomes small enough, due to the lack of diversity, the selective pressure will change to focus on searching for the feasible solutions. The distribution of solutions in the objective space needs to be abruptly changed. These convergent solutions $\boldsymbol{x}'s$ have little effects on promoting the search towards the optimal feasible region.

For the proposed ε-SEC, each section will dynamically shift towards the feasible region at each generation. The selection of solutions is conducted within the population in each section instead of the entire population. Hence, solutions are less likely to be crowded or converged to only one part of the infeasible region. In Fig. 3.3, solutions obtained in different sections can enhance the search in neighboring sections, and collaboratively promote the search towards the feasible region. For example, based on information collected from the solutions in $dis(4)$,

the search of optimal solutions in $dis(3)$ can be enhanced. Similarly, it works for the search of optimal solutions in $dis(2)$ and the feasible region (i.e. $dis(1)$). This makes the search collaboratively towards the optimal feasible region and avoids cases where solutions abruptly switch their distributions in the objective space as caused by the change of ε. Hence, the proposed ε-SEC is less sensitive to the value of $\varepsilon(g)$ as compared to ε-DE.

3.5 Results, Analysis and Discussion

3.5.1 Experiment Setup

To support the effectiveness of ε-SEC in solving the breast cancer immunotherapy model, ε-SEC is compared with four state-of-the-art constraint-handling techniques viz. CMODE [38], APFEC [36], ε-DE [25] and FROFI [37].

The parameters used in the implementation are listed as follows:

1)Population size N is 100. Maximum generation G_{max} is 500. F is 0.5, and CR is 0.9. The maximum run is 30. Function evaluation times (FEs) for each run is 1.5×10^5.

2) For the proposed technique ε-SEC, θ is 90%, and section number M is 5. G_c is 400.

3) For the compared technique ε-DE, based on the settings in [25], θ is 90%, and G_c is 400.

4) For the compared technique CMODE, based on the settings in [38], λ is 8, and k is 22.

5) For the compared technique APFEC, based on the settings in [36], α is 0.5.

6) For the compared technique FROFI, based on the settings in [37], MRN is $D/2$, where D is the dimension of solutions.

3.5.2 The Comparison of Constraint-handling Techniques

Since the optimal solution cannot be known in advance, the average and standard deviation of the objective function values obtained in 30 runs, feasible rate, and p-value are considered here. Table 3.4 presents the mean and standard deviation of the objective function values obtained in 30 runs. Among the five techniques, ε-SEC achieves the best performance in terms of the mean and standard deviation. To examine the robustness of algorithms, the feasible rate is used, which is the percentage of runs where at least one feasible solution is found. Herein, '$*$' denotes that feasible solutions cannot be consistently found by the corresponding technique in all runs. In Table 3.4, the feasible rates of ε-SEC, ε-DE and FROFI are all 100%. Hence, these three techniques are relatively robust. The Wilcoxon rank-sum test [80] is also conducted at the 5% significance level. The entries which are significantly better than the compared technique are marked with $**$. In Table 3.4, all of the p-values are smaller than the threshold level of 0.05. Hence, the effectiveness and robustness of ε-SEC are statistically supported for solving this constrained parameter optimization problem.

Table 3.4: Experimental Results of ε-SEC, FROFI, ε-DE, CMODE, and APFEC over 30 runs on the formulated problem. Mean, standard deviation, feasible rate, and p-value are presented, respectively. $*$ represents the technique with less than 100% feasible rate. $**$ represents $p < 0.05$ so that ε-SEC significantly performs better than the corresponding technique. The best results are highlighted in bold.

	ε-SEC	**FROFI**	**ε-DE**	**CMODE**	**APFEC**
Mean	**18.5347**	50.4050	704.6794	260.5527	225.8266
Standard Deviation	**0.7296**	68.5855	1525.4077	817.4933	546.5923
Feasible Rate	100%	100%	100%	$40\%^{*}$	$0\%^{*}$
p-value	N.A.	5.76E-04**	9.38E-07**	8.35E-06**	7.37E-05**

To analyze the performance of ε-DE, FROFI and ε-SEC, objective function

values and constraint violations of solutions obtained from these methods at various generations are plotted in Fig. 3.4, Fig. 3.5, and Fig. 3.6, respectively. In Fig. 3.4, it can be observed that solutions are crowded in the circle-marked region of the objective space at the 100th generation. Since the soft constraint-tolerance level $\varepsilon(g)$ is relatively large at the 100th generation compared with constraint violations of these solutions, most solutions are compared based on objective function values and are convergent in a region. As $\varepsilon(g)$ becomes smaller, due to the lack of diversity, many ε-feasible solutions becomes infeasible. The selective pressure thus concentrates on the search of feasible solutions instead of optimizing objective function values. In Fig. 3.4 at the 500th generation, it can be observed that variation of the feasible objective values are large. For the evolution of FROFI in Fig. 3.5, at 100th generation, solutions are distributed between the objective function values of 25 and 100. Compared with ε-DE, at the 100th generation, the diversity of solutions are better. At the 500th generation as shown in Fig. 3.5, obtained solutions are mostly feasible solutions. The objective function values of the obtained feasible solutions are between 25 to 120 so FROFI achieves better convergence than ε-DE. In Fig. 3.6, at the 100th generation, the individuals are distributed between objective values of 10 to 20, and 20 to 40. Compared with ε-DE as shown in Fig. 3.4, the solutions generated by ε-SEC are not crowded in certain regions at the 100th generation. The individuals are distributed in different sections to enhance both objective optimization and convergence to the feasible region. Compared with FROFI at 100th generation as shown in Fig. 3.5, ε-SEC does not push individuals into the region close to the feasible region at early generations. Since ε-SEC enhances both the convergence to feasible regions and objective optimization simultaneously, at 500th generation, it achieves the best convergence among the three techniques with the objective values around 17 to 20. Thus, ε-SEC can balance the task of optimizing objective function and the task of minimizing

constraint violations well for solving this problem.

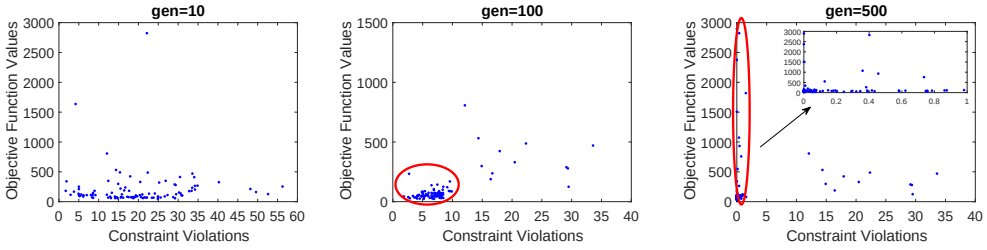

Figure 3.4: Evolution of ε-DE over the median performance run at gen=10, 100, and terminated generation 500 on the data-driven parameter optimization problem. At gen=500, a sub-figure which enlarges the constraint violation between 1$\sim$2 is included inside the figure. Red circles mark the crowded regions of solutions at gen=100 and 500.

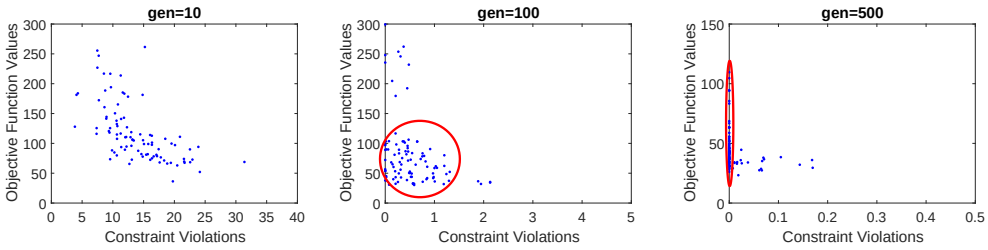

Figure 3.5: Evolution of FROFI over the median performance run at gen=10, 100, and terminated generation 500 on the data-driven parameter optimization problem. Red circles mark the crowded regions of solutions at gen=100 and 500.

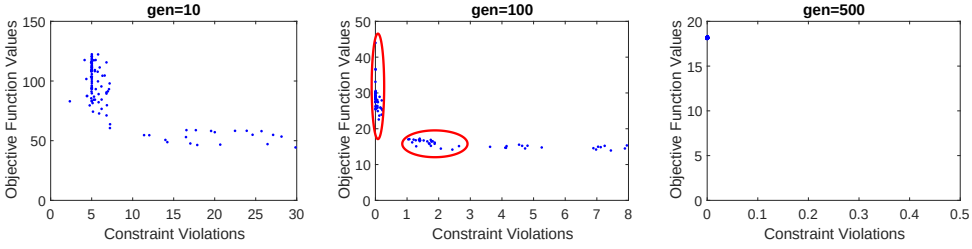

Figure 3.6: Evolution of the proposed technique ε-SEC over the median performance run at gen=10, 100, and terminated generation 500 on the data-driven parameter optimization problem. Red circles mark the crowded regions of solutions at gen=100. At gen=500, the solutions are crowded within a very tiny region, so a blank space with a point is shown.

To explore the robustness of ε-SEC, the best solution obtained versus the number of generations from the best run, median run, and worst run over 30 runs

is presented in Fig. 3.7. As shown in the figure, the convergence is achieved after the 100th generation for the best run, median run and worst run. The best feasible solution generated from ε-SEC is presented in Table 3.5.

Table 3.5: Obtained Best Feasible Solution Generated by ε-SEC

Parameters	**Units**	**Value**
a_c	day^{-1}	1.11×10^{-1}
b_c	$(cell\ \ day)^{-1}$	2.63×10^{-7}
c_c	$(cell\ \ day)^{-1}$	8.19×10^{-7}
d_c	$(cell\ \ day)^{-1}$	8.18×10^{-6}
a_e	day^{-1}	1.57×10^{-5}
b_e	day^{-1}	9.98×10^{-2}
c_e	$(cell\ \ day)^{-1}$	2.94×10^{-10}
d_e	$(cell\ \ day)^{-1}$	1.07×10^{-7}
a_r	day^{-1}	1.18×10^{-6}
b_r	day^{-1}	2.49×10^{-4}
k_1	$mg^{-1}\ kg\ day^{-1}$	6.04×10^{-3}
k_2	$mg^{-1}\ kg\ day^{-1}$	1.21×10^{0}
a_m	day^{-1}	9.99×10^{-2}
k_3	$mg^{-1}\ kg\ cell\ day^{-1}$	8.04×10^{2}
k_4	$mg^{-1}\ kg\ cell\ day^{-1}$	4.14×10^{4}

The section number M is an important parameter of ε-SEC. I have conducted a sensitivity analysis of the section number M, for which $M = \{3, 4, 5, 6\}$ over 30 runs in solving the data-driven parameter optimization problem.

In Table 3.6, the mean of the objective values when $M = 3$ is larger compared with the results when $M = 4$, $M = 5$ and $M = 6$. This is because if the number of sections is small, solutions will not be widely distributed to different regions and diversity of solutions may not be preserved. As a result, the issues encountered in ε-DE would likely happen when $M = 3$. After $\varepsilon(g)$ reduces to an extent, the selective pressure will change abruptly from optimizing the objective

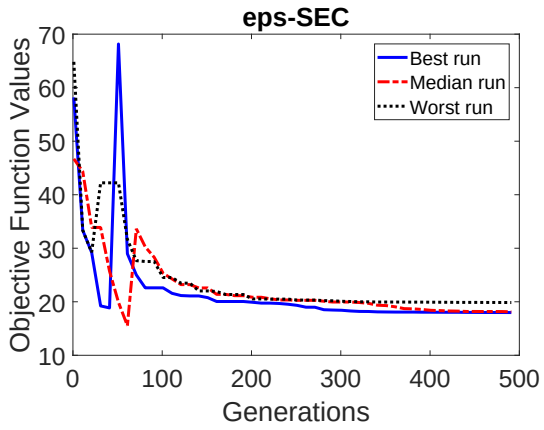

Figure 3.7: Convergence plot of the best solution obtained from the best run, median run and worst run of ε-SEC.

function into minimizing the constraint violation. The selective pressure no longer focuses on optimizing the objective function, the mean of the objective value can be large when $M = 3$. Although the best performance is achieved when $M = 5$, it is comparable when $M = 4$, $M = 5$ and $M = 6$.

In Fig. 3.8, the box plot of the parameter M is plotted when $M = 4$, $M = 5$ and $M = 6$. The mediums are relatively close to each other and the variance when $M = 5$ is smaller. Overall, M is not very sensitive when $M > 3$. In our paper, M is set to be 5.

Table 3.6: Experimental Results of ε-SEC for M=3, 4, 5, 6 over 30 Runs.

	M=3	**M=4**	**M=5**	**M=6**
Mean	411.5892	18.6498	**18.5347**	18.6166
Standard Deviation	1413.8743	0.7877	**0.7296**	0.8149

3.5.3 Tumor Response without Treatments

The simulation results of C cells, E cells and R cells without treatments are shown in Fig. 3.9.

Based on the parameters generated from ε-SEC, tumor cells (C) increase logistically from the starting point and are finally stable. These are consistent

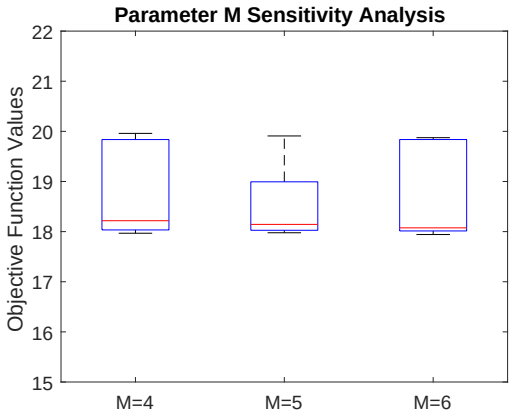

Figure 3.8: Box plot of the parameter sensitivity analysis for $M = 4$, $M = 5$, and $M = 6$.

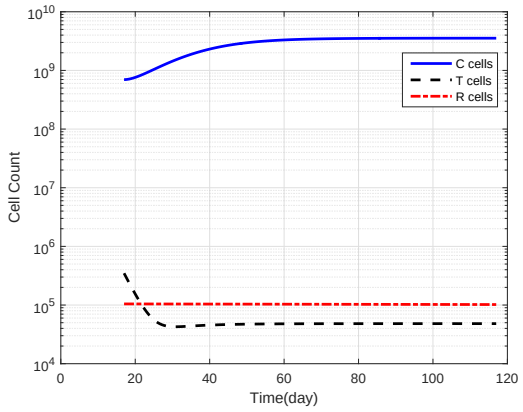

Figure 3.9: Simulation results of C cells, E cells, and R cells without therapy.

with the biological principles that tumor cells increase logistically as well as the stability of system biology [58]. E cells decrease moderately followed by a slight increase and reach a stable level when C cells become stable. The parameters generated from ε-SEC rationalizes the immunotherapy model and satisfies corresponding constraints.

3.5.4 Stability of Equilibrium

A system will move towards an equilibrium point if that point is stable. To analyze the long-term dynamics of the system, the next step is to investigate the stability of the equilibrium point without any treatments. Based on the established

model in Section 3.2 and the parameters obtained by ε-SEC, two equilibrium points E_1 and E_2 are found in Table 3.7.

Table 3.7: Location of Equilibrium

Equilibrium	**C**	**E**	**R**	**M**
E_1	0	0	0	0
E_2	3.5403×10^9	4.8660×10^4	2.3008×10^2	0

Jacobian Matrix in Equation (3.9) is the system matrix. The Jacobian Matrix is applied to check the stability of these two equilibrium points, and the eigenvalues are displayed in Table 3.8.

$$J = \begin{bmatrix} a_c - \frac{2a_cC}{C_{\max}} - b_cE - c_cM & -b_cC & 0 & -c_cC \\ a_e - c_eE & -b_e - c_eC - d_eR & -d_eE & 0 \\ 0 & a_r & -b_r & 0 \\ 0 & 0 & 0 & -a_m \end{bmatrix}. \quad (3.9)$$

Table 3.8: Eigenvalues of Equilibrium and Stability

Equilibrium	**Stability**	$\boldsymbol{\lambda_1}$	$\boldsymbol{\lambda_2}$	$\boldsymbol{\lambda_3}$	$\boldsymbol{\lambda_4}$
E_1	Unstable	-0.0002	-0.0998	0.1113	-0.0999
E_2	Stable	-0.0002	-0.0997	-1.1397	-0.0999

In Table 3.8, E_1 is found to be unstable since λ_3 is positive. The high tumor equilibrium E_2 is stable. In addition, according to [72], euthanasia is conducted to mice if the cross-sectional area of primary tumor reaches 250 mm^2. Since each tumor cell is approximately a sphere, the volume of each tumor cell is around 3000 mm^3. There are 10^6 tumor cells per mm^3 indicated in [72], so the number of tumor cells should be larger than or equal to 3×10^9 at the equilibrium point. The treatment should not be conducted if the mice are dead. Hence, at the equilibrium point, it is required to check $C \geq 3 \times 10^9$. In Table 3.7, the number

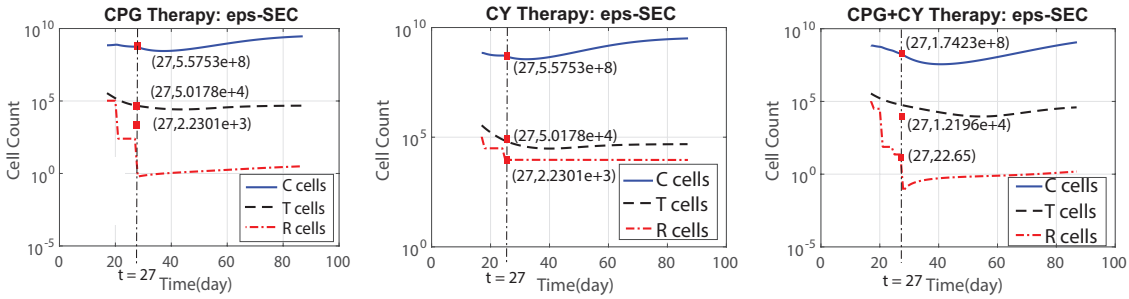

Figure 3.10: Simulation results based on ε-SEC with three different treatments which start on day 17. Squares represent cell counts from [72], and the dotted vertical line intersects with the cell counts obtained from the model on day 27.

of tumor cells C is larger than 3×10^9, which satisfies the requirement of tumor cell counts at the equilibrium point.

3.5.5 Tumor Response with Treatments

According to [72], for each cycle (i.e. 7 days per cycle) starting on day 17, mice were injected with CY (200mg/kg) on day 0 and with CpG (5mg/kg) on day 3.

In Fig. 3.10, the tumor responses are displayed with CpG, CY, and CpG+CY treatments on day 27, respectively. Squares represent cell counts from [72], and the dotted vertical line intersects with the cell counts obtained from the model on day 27. In Table 3.9, the computed results from the model and the clinical results in [72] are shown. The numbers in bold are the computed results from the model. In Table 3.9 and Fig. 3.10, all of the computed results of C, E and R cells for CpG and CY on day 27 are acceptable as compared to the clinical results since the differences between the clinical and computed results are mostly within 10%. The number of tumor cells are reduced with the help of CpG and CY therapy.

In Fig. 3.10 and Table 3.9, after 10 days of CpG+CY therapy, the number of tumor cells are reduced to nearly 20% of C_0, and nearly 80% of the tumors are eradicated. Based on the parameters generated from ε-SEC, the computed number of tumor cells are within 10% difference with the clinical results as

shown in Table 3.9. Combining CpG and CY together can eradicate more tumors as compared with the CpG therapy or CY therapy alone. According to Equation (3.1) to (3.4), as the injection of CpG and CY, more M cells are generated and R cells are significantly reduced according to [75, 76]. R cells suppress E cells. As R cells become smaller and others do not change, the reduction of E cells is slower as compared with only one therapy conducted. Hence, the number of E cells become larger after this treatment. Since E cells can kill tumor cells, according to [72], more E cells and more M cells inhibit the growth of tumor cells, which lead to the eradication of more tumor cells.

To support the performance of ε-SEC, the simulation results with CpG and CpG+CY therapy based on the best set of parameters generated by FROFI are shown in Fig. 3.11 since FROFI produced the second-best performance among the five techniques studied. For the CpG therapy, the computed result of R cells on day 27 is around 50, but the obtained clinical result of R cells in [72] is 2.2301×10^3. The difference between the clinical and computed results is in two orders of magnitude, which is not acceptable in the field of immunotherapy. Furthermore, the behavior marked in the circle is abnormal. According to [72], within the first two therapy cycles, CpG is administered on day 20 and day 27. One of the effects of the injection of CpG is to reduce the number of R cells [72]. Although for the first treatment cycle after day 20, no CpG is administered, the number of R cells should increase or decrease gradually instead of increasing rapidly between day 20 and day 27 as marked in the circle in Fig. 3.11. Similarly for the CpG+CY therapy, abnormal behavior is also observed as marked in the circle. According to [72], within the two therapy cycles of CpG+CY, CpG is administered on day 20 and day 27, and CY is administered on day 17 and day 24. One of the effects of the administration of CY is to reduce the number of R cells based on [72]. In addition to the abnormality of the rapid increase of the number of R cells after the injection of CpG, the fast increase and decrease of the number

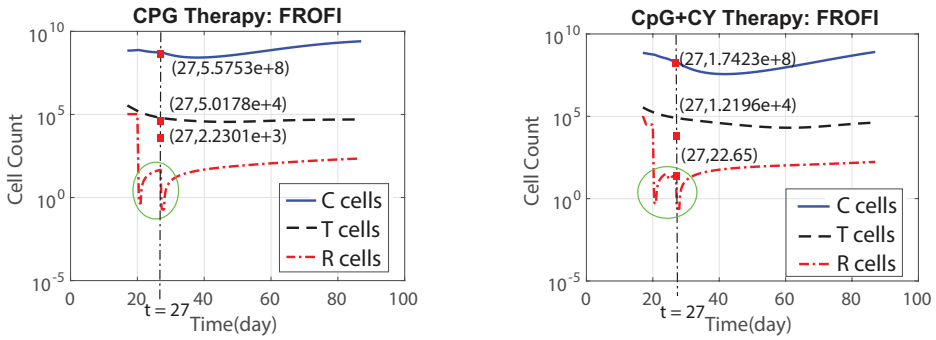

Figure 3.11: Simulation results with CpG and CpG+CY treatments based on the parameters generated from FROFI. The circle marks the abnormal behavior of R cells.

of R cells after the injection of CY within only three days (between day 24 to day 27) is not acceptable. Hence, based on the model with parameters obtained from FRORI, the therapeutic effects of CpG and CpG+CY do not provide lasting and stable therapeutic effects, which make the model not applicable for further clinical studies.

3.5.6 Model-based Prognostics with Different Drug Schedules

In Table 3.9, I also present the tumor cells counts in the long term after 7 cycles of CpG+CY therapy. The computed result shows more than 99% eradication of the tumor cells, which is consistent with the estimation in [72] that at least 90% eradication of the tumor cells was achieved. In the long term, the tumors can be largely eradicated with the combination of CpG+CY treatment, provided the assumption of the model on tumour homogeneity holds.

From clinical practices, it is impossible to exhaustively try all possible combinations of drug schedules due to cost and time constraints. With the established model and the obtained parameters from ε-SEC, we can predict the therapeutic effects with different ways of drug delivery. To explore treatment strategies, the CpG+CY drug delivery schedules in [72] can be changed, for example, to the following two cases:

Table 3.9: Number of Tumor Cells C, Effector Cells E, Regulatory Cells R with CY, CpG and CY+CpG Treatments. The numbers in the upper line are the clinical results in [72], and the numbers in the bottom line in bold are computed from our model

	C	E	R
CpG (Day 27)	5.5753×10^8	5.0178×10^4	2.2301×10^3
	$\mathbf{5.5759}\times10^8$	$\mathbf{5.0201}\times10^4$	$\mathbf{2.5107}\times10^2$
CY (Day 27)	4.6461×10^8	6.9692×10^4	9.2922×10^3
	$\mathbf{4.2079}\times10^8$	$\mathbf{5.4192}\times10^4$	$\mathbf{9.2933}\times10^3$
CpG+CY (Day 27)	1.7423×10^8	1.2196×10^4	22.65
	$\mathbf{1.9164}\times10^8$	$\mathbf{5.9027}\times10^4$	**22.6534**
CpG+CY (Day 66)	$\leq1.7423\times10^7$	–	–
	$\mathbf{2.2969}\times10^3$	**–**	**–**

Case 1: Fix CY delivery on day 0 per cycle, and modify the delivery time of CpG on day 0, 1, 2, 3, 4, 5, 6 per cycle so that there are 7 different drug schedules in total.

Case 2: Fix CpG delivery on day 0 per cycle, and modify the delivery time of CY on day 0, 1, 2, 3, 4, 5, 6 per cycle so that there are 7 different drug schedules in total.

As indicated in [72], there are approximately 1×10^6 number of tumor cells per mm^3. Tumor volumes are used to represent the disease progression in Table 3.10. The number in bold represents the tumor cell volumes based on the drug protocol in [72].

From Table 3.10, it can be observed that

1) If both CpG and CY are administered on day 0 of each cycle, the best effects of therapy is obtained with the total tumor volume 105.06 mm^3. This value is lower than the standard protocol used in clinical trials.

2) The performance of case 2 is generally better than that of case 1. We can give some suggestions to the clinical practices that CpG is better to be delivered

Table 3.10: Comparisons of tumor volumes (mm^3) on day 27 of different treatment schemes of CpG+CY. The number in bold is the tumor volumes based on the drug protocol in [72]. For the first row, it represents on which day CpG or CY is administered of each cycle.

	day0	day1	day2	day3	day4	day5	day6
Case 1	105.06	129.62	162.83	**174.23**	209.42	231.00	257.59
Case 2	105.06	123.57	147.49	167.68	179.69	193.98	211.22

on day 0, and modify the delivery time of CY of each cycle.

From the biological perspective, observation (1) implies that the treatment should start as early as possible.

3.6 Chapter Conclusion

In this work, I have formulated an off-line data-driven parameter optimization problem for a breast cancer immunotherapy model, and proposed a new constraint-handling technique to solve the problem. The mechanisms of the proposed technique can preserve the diversity of solutions and enhance the solutions for convergence to the near-optimal feasible region. The simulation results have been compared with four state-of-the-art evolutionary constraint-handling techniques, which show the effectiveness and robustness of the proposed technique in solving the data-driven parameter optimization problem. The optimized parameters obtained from the proposed constraint-handling technique not only rationalizes the established model, but also generalizes the model for both the replication and prognostication of clinical therapeutic outcomes.

In future work, the model should assume tumor heterogeneity since in real life, a tumor is often heterogeneous. Such intra-tumour heterogeneity often gives rise to treatment resistance and cancer relapse. It is therefore important to know when

the predictions of the model are diverging too much from actual observations on a patient (e.g. to generate confidence intervals for the model's predictions), and thus to switch the treatment regime for the patient. Moreover, more clinical results can be collected to refine the established cancer immunotherapy model. In addition, if real-time clinical results can be collected, on-line data-driven optimization task can also be explored for the cancer immunotherapy model. Although the proposed constraint-handling technique was developed explicitly for the parameter optimization of a cancer immunotherapy model, the approach can be applied to other problems such as experimental designs, industrial prognostics and so on.

Chapter 4

A Data Augmentation Pipeline for Solving the Small-Data Issue of our Cancer Immunotherapy Model

4.1 Introduction

Many real-world modeling problems can be solved with the help of parameter optimization based on data collected from physical experiments, real events, or complex numerical simulations [51]. However, due to time, cost and resource constraints, in some cases, only very small amounts of data can be collected and used for training a model. For instance, the clinical trials from cancer immunotherapy in Chapter 3 collected only 10 real data points. These small amounts of collected data can lead to poor-fit issues and undermine resolution and generalizability of a model [81]. Thus, to tackle small-data issues and to improve the resolution of the cancer immunotherapy model in Chapter 3, it is crucial to figure out how to generate more training data.

There are several methods of tackling small-data issues in the field of computational intelligence. A representative method is data augmentation, i.e.

adding more artificially generated data by different methods. The trivial data-augmentation method is to replicate the same set of data. However, more replicated data does not insert enough additional information to improve the resolution of a model. In addition to the trivial method, a common one is data transformation such as adding some noise to the collected data in industrial domain applications or transforming the training images with shifting and rotation operations in the field of computer vision [82]. Another representative method for solving small-data issues is fine-tuning [83], which uses well-trained model parameters obtained from training a large dataset to initialize new model parameters. New model parameters are updated based on training using additionally collected small datasets. The fine-tuning method is generally utilized in deep neural networks, and is more useful when using gradient-based approaches to update model parameters. The fine-tuning method is not applicable to updating model parameters using population-based and gradient-free approaches such as evolutionary algorithms.

To improve the cancer immunotherapy model in Chapter 3 by solving small-data issues, the following data augmentation pipeline is proposed.

For any constraint-handling technique `X`, apply `X` to an initial small set of data points to obtain a draft model `X_draft`. Then use `X_draft` to generate some additional pseudo data points. These additional pseudo data points are combined with the initial small set of real data points, and divided into a new training and a new testing data set. The constraint-handling technique `X` is then applied to the new training data set (after adding some Gaussian noise to the new training data set to mitigate overfitting) to obtain a new model `X_new`, which is then tested on the new testing data set.

This data-augmentation procedure is demonstrated on five constraint-handling techniques, including ε-SEC, using the cancer immunotherapy model from Chapter 3. The results provide evidence that this data-augmentation procedure

is universally effective in improving the resolution (viz. error size) of these constraint-handling techniques. The results also provide evidence that ε-SEC sustains its superior resolution compared to the other four constraint-handling techniques post data augmentation. Lastly, the results provide evidence that training the other constraint-handling techniques using the pseudo training data set generated by ε-SEC improves their resolution more than using their own respective pseudo training data set.

4.2 Details of the Data Augmentation Pipeline

In order to solve small-data issues in constrained parameter optimization problems, I introduce a data augmentation pipeline for constraint-handling techniques as shown in Fig. 4.1. For any constraint-handling technique `X`, apply `X` to the initial small set of data points $S0$ to obtain a draft model `X_draft`. Then use `X_draft` to generate some additional pseudo data points $S1$. These additional pseudo data points $S1$ are combined with the initial small set of real data points $S0$, and divided into a new training and a new testing data set. The constraint-handling technique `X` is then applied on the new training data set (after adding some Gaussian noise to the new training data set to mitigate overfitting) to obtained a new model `X_new`, which is then tested on the new testing data set.

An alternative data augmentation pipeline is to use the draft model produced by a constraint handling technique `X` to produce new training and testing data sets, and then applied another constraint-handling technique `Y` to this new training data set (after adding some Gaussian noise to mitigate overfitting) to obtain a new model `X/Y_new`, which is then tested on the new testing data.

Here, I postulate and investigate the following three hypotheses related to the effectiveness of various constraint-handling techniques when data augmentation is involved.

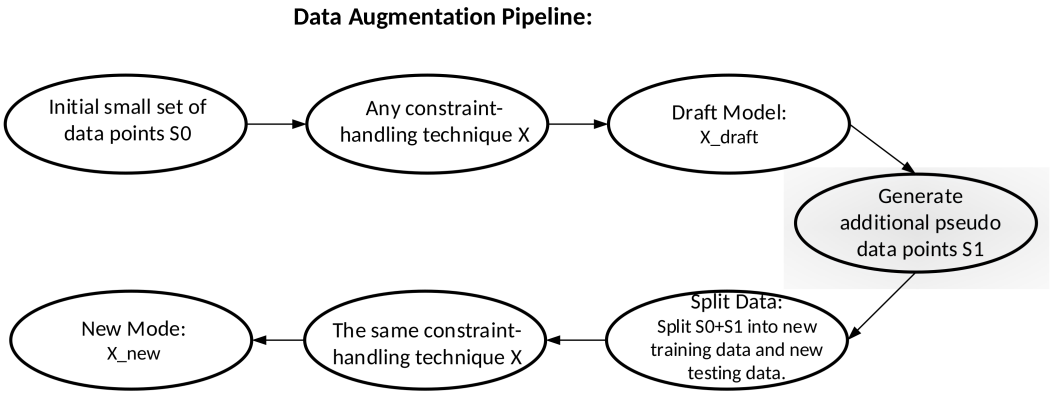

Figure 4.1: A data augmentation pipeline used for cancer immunotherapy model.

Hypothesis 1: `X_new` has better resolution (i.e. smaller error) than `X_draft`, for each of the five constraint-handling technique `X` considered; thus the data augmentation pipeline is a universal technique for improving the resolution of constraint-handling techniques.

Let $S0$ be the set of 10 real data points from [72], and let `X_draft` be the cancer immunotherapy model with the best parameter set obtained by applying a constraint-handling technique `X` (where `X` is any of the five techniques considered in Chapter Chapter 3) on $S0$. Inputs into `X_draft` are initial conditions of cell counts of C cells, E cells and R cells, drug schedule per cycle, and time point t. Outputs of `X_draft` are the predicted (by `X_draft`) cell counts of C cells, E cells and R cells at time t in accordance to the input drug schedules. Let $S1$ be a set of 60 pseudo data points predicted by `X_draft` (by feeding `X_draft` 20 artificially created drug schedules). Since the 10 real data points are from 4 real drug schedules, there are 70 data points in total from 24 real and artificial drug schedules. These 70 data points form $S0$+$S1$. I split $S0$+$S1$ into 40 data points from 14 drug schedules as the new training data and 30 data from 10 drug schedules as the new testing data. Gaussian multiplicative noise, with multiplicative factors distributed according to $N(1, 0.01)$, is inserted into each training data point in order to train a more robust model. `X_new` is the

cancer immunotherapy model with the best parameter set obtained by the same constraint-handling technique X using these 40 modified training data points. The 30 testing data points are then evaluated based on X_new.

If the errors from X_new based on $S0$+$S1$ are smaller than the errors from X_draft and $S0$, it is an evidence that supports hypothesis 1. Otherwise, hypothesis 1 fails.

Hypothesis 2: ε-SEC/Y_new has better resolution than Y_draft, for each of the other four constraint-handling technique Y considered; thus the pseudo data points generated by ε-SEC can assist in improving the resolution of other constraint-handling techniques.

With reference to the pipeline in Fig. 4.1, let $S0$ be the set of 10 real data points from [72], and let ε-SEC_draft be the cancer immunotherapy model with the best parameter set obtained by applying ε-SEC on $S0$. Inputs into ε-SEC_draft are initial conditions of cell counts of C cells, E cells and R cells, drug schedule per cycle, and time point t. Outputs of ε-SEC_draft are cell counts of C cells, E cells and R cells at time t predicted by ε-SEC_draft in accordance to the input drug schedules. Let $S1$ be the 60 additional data points predicted by ε-SEC_draft on the same 20 artificial drug schedules created earlier. Together with the 10 real data points from the 4 drug schedules in $S0$, there are 70 data points in total from 24 drug schedules. These 70 data points form $S0$+$S1$. As before, I split $S0$+$S1$ into a new training dataset with 40 data points and a new testing dataset with 30 data points. Gaussian multiplicative noise, with multiplicative factors distributed according to $N(1, 0.01)$, is inserted into each data point in this new training dataset in order to train a more robust model. Let ε-SEC/Y_new be the cancer immunotherapy model with the best parameter set obtained by applying a constraint-handling technique Y (where Y is any of the other four techniques considered in Chapter 3) on these 40 modified training data points. The 30 testing data points are then evaluated based on

ε-SEC/Y_new.

If the errors from ε-SEC/Y_new and $S0$+$S1$ are smaller than the errors from Y_draft and $S0$, it is an evidence that supports hypothesis 2. Otherwise, hypothesis 2 fails.

Hypothesis 3: ε-SEC_new has better resolution than Y_new, for each of the other four constraint-handling technique Y considered; thus ε-SEC retains its superior resolution over other constraint-handling techniques post data augmentation.

With reference to the pipeline in Fig. 4.1, let $S0$ be the set of 10 real data points from [72], and let ε-SEC_draft and Y_draft be the cancer immunotherapy model with the best parameter set obtained by applying ε-SEC and each of the other four constraint-handling techniques Y on $S0$. Inputs into ε-SEC_draft and Y_draft are initial conditions of cell counts of C cells, E cells and R cells, drug schedule per cycle, and time point t. Outputs of ε-SEC_draft and Y_draft are cell counts of C cells, E cells and R cells at time t predicted by ε-SEC_draft and Y_draft in accordance to the input drug schedules. Let ε-SEC_S1 and Y_S1 be the respective 60 additional data points predicted by ε-SEC_draft and Y_draft on the same 20 artificial drug schedules created earlier. The 10 real data points from the 4 drug schedules in $S0$ are combined with ε-SEC_S1 (respectively, Y_S1) to obtain a combined $S0$+ε-SEC_S1 (respectively, $S0$+Y_S1) dataset having 70 data points in total from 24 drug schedules. As before, I split each of these $S0$+$S1$ combined datasets into a new training dataset with 40 data points and a new testing dataset with 30 data points. Gaussian multiplicative noise, with multiplicative factors in $N(1, 0.01)$, is inserted into each data point in each new training dataset in order to train a more robust model. Let ε-SEC_new and Y_new respectively be the cancer immunotherapy model with the best parameter set obtained by applying ε-SEC and each of the other four constraint-handling techniques Y on their respective

40 modified training data points. The respective 30 new testing data points are then evaluated respectively based on ε-SEC_new and Y_new.

If the errors from ε-SEC_new and $S0$+ε-SEC_S1 are smaller than the errors from Y_new and the corresponding $S0$+Y_S1, it is an evidence that supports hypothesis 3. Otherwise, hypothesis 3 fails.

4.3 Results, Analysis and Discussion

In this section, empirical studies are conducted to assess the three hypotheses raised in the previous section. To evaluate the resolution of a model, both training and testing errors are used based on the mean of the absolute errors between computed results from the model and ground truth. The constraints mentioned in this section are exactly those defined in Chapter 3.

4.3.1 Support for Hypothesis 1

According to the pipeline in Fig. 4.1, pseudo data points $S1$ are generated based on X_draft by the constraint-handling technique X. The combined data points $S0$+$S1$ are used to optimize model parameters by the same constraint-handling technique X, and X_new is obtained.

The respective 40 training data in $S0$+$S1$ with Gaussian multiplication noise are used for training by the five constraint-handling techniques used in Chapter 3 with 20 independent runs.

Since the optimal solution cannot be known in advance, the average of the training and testing errors obtained in 20 runs and feasible rate are considered here. The feasible rate indicates the percentage of runs where at least one feasible solution is found. Table 4.1 presents the mean and the standard deviation of the training errors for the 10 original data points from clinical trials based on model X_draft, the mean and the standard deviation of the testing errors based on

model `X_new` for the 10 original data points from clinical trials, the mean and the standard deviation of the training errors for the 40 pseudo training data points, and the mean and the standard deviation of the testing errors for the 30 pseudo testing data points. For all the five constraint-handling techniques, the testing errors from `X_new` and $S0$, and the training and testing errors from `X_new` and $S0$+$S1$ are smaller than the training errors from `X_draft` and $S0$. Thus, there is evidence that the data augmentation pipeline is a universal technique for improving the resolution of constraint-handling techniques when there are small-data issues. Hypothesis 1 is thus supported.

Table 4.1: Based on the pipeline and model X_new, the experiment results of five constraint-handling techniques over 20 runs for training and testing errors.

	ε-SEC	ε-DE	FROFI	CMODE	APFEC
X_draft Training Error (Original 10 data)					
Mean (Std)	1.85 (0.07)	70.5 (15.25)	5.04 (0.68)	26.55 (8.17)	22.58 (5.47)
Feasible Rate	100%	100%	100%	40%	0%
X_new Testing Error (Original 10 data)					
Mean (Std)	1.43 (0.04)	2.71 (1.07)	2.24 (0.69)	23.35 (7.14)	20.67 (5.13)
Feasible Rate	100%	100%	100%	45%	0%
X_new Training Error (40 data)					
Mean (Std)	0.25 (0.01)	0.91 (0.73)	0.59 (0.05)	14.15 (1.37)	16.07 (2.39)
Feasible Rate	100%	100%	100%	35%	0%
X_new Testing (30 data)					
Mean (Std)	0.28 (0.01)	0.89 (0.78)	0.64 (0.07)	14.24 (1.43)	16.79 (2.51)
Feasible Rate	100%	100%	100%	35%	0%

4.3.2 Support for Hypothesis 2

I list the $S0$+$S1$, generated using ε-SEC_draft, for assessing hypothesis 2 in Table 4.2 (viz. the 40 pseudo training data points) and Table 4.3 (viz. the 30 pseudo testing data points). $S0$+$S1$ are the same for all the five constraint-handling techniques. $S0$+$S1$ consist of 40 pseudo training data points from 14 drug schedules and 30 pseudo testing data points from 10 drug schedules. The injection of CpG and CY are scheduled based on the exact day of each cycle, and each cycle consists of 7 days. In Table 4.2 and Table 4.3, if only one drug is injected, only one schedule is shown for each row under the column Drug Schedule. If two drugs are injected, two schedules are shown for each row under the column Drug Schedule. The exact day on each row indicates the cell counts of C cells, E cells and R cells on that day. For example, CpG+CY ((day 1/cycle, day 0/cycle), Day 27) means CpG injected on day 1 of each cycle, CY injected on day 0 of each cycle, and the C, E, R columns are the corresponding cell counts on day 27.

The 40 training data points, generated by ε-SEC_draft, with Gaussian multiplicative noise added, are used for training by four other constraint-handling techniques Y used in Chapter 3, excluding ε-SEC, with 20 independent runs.

Table 4.4 presents the mean and the standard deviation of the training errors for the 10 original data points from clinical trials, the mean and the standard deviation of the testing errors based on ε-SEC/Y_new for the 10 original data points from clinical trials based on model Y_draft, the mean and the standard deviation of the training errors for the 40 training data points, and the mean and the standard deviation of the testing errors for the 30 testing data points. For all these four constraint-handling techniques, the testing errors from ε-SEC/Y_new and $S0$, and the training and testing errors from ε-SEC/Y_new and $S0$+$S1$ are smaller than the training errors from Y_draft and $S0$. Thus the pseudo data points generated by ε-SEC_draft can assist in improving

Table 4.2: Number of Tumor Cells C, Effector Cells E, Regulatory Cells R with CY, CpG and CY+CpG Treatments in training group from various drug schedules.

	Drug Schedule	**C**	**E**	**R**
1	CpG ((day 0/cycle), Day 27)	3.0937×10^{8}	5.7838×10^{4}	7.5060×10^{-1}
2	CpG ((day 2/cycle), Day 27)	4.7478×10^{8}	5.2031×10^{4}	5.9200×10^{-1}
3	CpG ((day 3/cycle), Day 27) (Real data)	5.5753×10^{8}	5.0178×10^{4}	2.2301×10^{3}
4	CpG ((day 4/cycle), Day 27)	6.0446×10^{8}	4.9022×10^{4}	2.4636×10^{2}
5	CY ((day 1/cycle), Day 27)	4.9115×10^{8}	5.1958×10^{4}	9.3100×10^{3}
6	CY ((day 3/cycle), Day 27)	6.5952×10^{8}	4.9047×10^{4}	3.1158×10^{4}
7	CpG+CY ((day 0/cycle,day 0/cycle), Day 27)	1.0482×10^{8}	6.7928×10^{4}	2.4600×10^{-1}
8	CpG+CY ((day 1/cycle,day 0/cycle), Day 27)	1.2932×10^{8}	6.4046×10^{4}	1.4650×10^{-1}
9	CpG+CY ((day 5/cycle,day 0/cycle), Day 27)	2.3045×10^{8}	5.6234×10^{4}	2.2194×10^{1}
10	CpG+CY ((day 0/cycle,day 3/cycle), Day 27)	1.6726×10^{8}	6.1370×10^{4}	3.5230×10^{-1}
11	CpG+CY ((day 0/cycle, day 4/cycle), Day 27)	1.7028×10^{8}	5.7142×10^{4}	3.3100×10^{-1}
12	CpG+CY ((day 0/cycle,day 5/cycle), Day 27)	1.9349×10^{8}	5.9234×10^{4}	3.4560×10^{-1}
13	CpG+CY ((day 0/cycle, day 6/cycle), Day 27)	2.0015×10^{8}	5.5654×10^{4}	3.2640×10^{-1}
14	CpG+CY ((day 3/cycle, day 0/cycle), Day 66) (Real data)	$\leq1.7423\times10^{7}$	–	–

the resolution of constraint-handling techniques, and hypothesis 2 is supported. Furthermore, compared to Table 4.1, the testing errors based on ε-SEC/`Y_new` from the original 10 real data points are smaller than the testing errors based on `Y_draft`/`Y_new`. The difference in hypothesis 1 and 2 is in the dataset $S1$, where $S1$ is generated by the respective `X_draft` in hypothesis 1 but it is generated solely by ε-SEC`_draft` in hypothesis 2. If the pseudo data is generated based on a model with better resolution, the replication of real data points can be improved. Thus, training the other constraint-handling techniques using the pseudo training data set generated by ε-SEC improves their resolution more than using using their own respective pseudo training data set. Also, when Table 4.4 is compared to the ε-SEC column in Table 4.1, ε-SEC`_new` presents the smallest training and testing errors, so it is still the most effective

Table 4.3: Number of Tumor Cells C, Effector Cells E, Regulatory Cells R with CY, CpG and CY+CpG Treatments in testing group from various drug schedules.

	Drug Schedule	**C**	**E**	**R**
1	CpG ((day 1/cycle), Day 27)	3.6081×10^{8}	5.1718×10^{4}	6.2880×10^{-1}
2	CpG ((day 5/cycle), Day 27)	6.2978×10^{8}	4.5755×10^{4}	2.3394×10^{2}
3	CY ((day 0/cycle), Day 27) (Real data)	4.6461×10^{8}	6.9692×10^{4}	9.2922×10^{3}
4	CY ((day 2/cycle), Day 27)	5.5394×10^{8}	4.7763×10^{4}	8.8444×10^{3}
5	CpG+CY ((day 2/cycle,day 0/cycle), Day 27)	1.5432×10^{8}	5.8167×10^{4}	6.1500×10^{-2}
6	CpG+CY ((day 3/cycle,day 0/cycle), Day 27) (Real data)	1.7423×10^{8}	1.2196×10^{4}	2.2650×10^{1}
7	CpG+CY ((day 4/cycle,day 0/cycle), Day 27)	1.9848×10^{8}	5.4592×10^{4}	2.1124×10^{1}
8	CpG+CY ((day 6/cycle,day 0/cycle), Day 27)	2.4413×10^{8}	5.2580×10^{4}	2.1051×10^{1}
9	CpG+CY ((day 0/cycle,day 1/cycle), Day 27)	1.1711×10^{8}	6.1826×10^{4}	1.8100×10^{-1}
10	CpG+CY ((day 0/cycle,day 2/cycle), Day 27)	1.3977×10^{8}	5.9817×10^{4}	1.2200×10^{-1}

constraint-handling technique when solving this problem.

4.3.3 Support for Hypothesis 3

According to the pipeline in Fig. 4.1, pseudo data points ε-SEC_S1 (respectively, Y_S1) are generated based on ε-SEC_draft (respectively, Y_draft) by the constraint-handling technique X (respectively, Y). The respective combined data points $S0$+$S1$ are used to optimize model parameters by the same constraint-handling technique ε-SEC or Y, and ε-SEC_new and Y_new are obtained, respectively, with 20 independent runs each.

According to Table 4.1 and Table 4.4, the testing errors from ε-SEC_new and $S0$ are smaller than the testing errors from ε-SEC/Y_new and $S0$, and the testing errors from Y_new and $S0$. The training and testing errors from ε-SEC_new and $S0$+$S1$ are smaller than Y_new and the corresponding $S0$+$S1$. Thus, the superiority of ε-SEC over other four constraint-handling techniques, in solving the constrained parameter optimization problem post data augmentation,

Table 4.4: Based on the pipeline and model Y_new, the experiment results of the other four constraint-handling techniques Y over 20 runs for training and testing errors.

	ε-DE	**FROFI**	**CMODE**	**APFEC**
Y_draft Training Error (Original 10 data)				
Mean (Std)	70.5 (15.25)	5.04 (0.68)	26.55 (8.17)	22.58 (5.47)
Feasible Rate	100%	100%	40%	0%
ε-SEC/Y_new Testing Error (Original 10 data)				
Mean (Std)	2.55 (0.94)	2.08 (0.53)	21.44 (7.05)	19.58 (4.87)
Feasible Rate	100%	100%	35%	0%
ε-SEC/Y_new Training Error (40 data)				
Mean (Std)	1.92 (0.81)	0.88 (0.06)	14.27(1.52)	16.14 (2.47)
Feasible Rate	100%	100%	35%	0%
ε-SEC/Y_new Testing Error (30 data)				
Mean (Std)	1.07 (0.73)	1.03 (0.07)	14.34 (1.54)	17.21 (2.59)
Feasible Rate	100%	100%	30%	0%

is supported. In other words, hypothesis 3 is supported. In addition, its 100% feasible rate indicates the robustness of ε-SEC in handling this problem.

4.3.4 Remark on ε-SEC

Based on Table 4.1 and Table 4.4, ε-SEC presents the best model resolution compared with the other four constraint-handling techniques. Table 4.5 shows that, for ε-SEC, the differences between predicted and actual tumor cell counts on day 27 for different drug schedules are improved from within 10% (draft

Table 4.5: For Real Data Points on Day 27, the Number of Tumor Cells C, Effector Cells E, Regulatory Cells R with CY, CpG and CY+CpG Treatments based on Draft Model and New model.

	C	E	R
CpG ((day 3/cycle), Day 27) (Draft Model)	5.5759×10^8 (0.01 %)	5.0201×10^4	2.5107×10^2
CpG ((day 3/cycle), Day 27) (New Model)	5.5124×10^8 (1.13%)	4.9694×10^4	2.4374×10^2
CpG ((day 3/cycle), Day 27) (Clinical results)	5.5753×10^8	5.0178×10^4	2.2301×10^3
CY ((day 0/cycle), Day 27) (Draft Model)	4.2079×10^8 (9.43%)	5.4192×10^4	9.2933×10^3
CY ((day 0/cycle), Day 27) (New Model)	4.6998×10^8 (1.16%)	5.3766×10^4	9.3585×10^3
CY ((day 0/cycle), Day 27) (Clinical results)	4.6461×10^8	6.9692×10^4	9.2922×10^3
CpG+CY ((day 3/cycle,day 0/cycle), Day 27) (Draft Model)	1.9164×10^8 (9.99%)	5.9027×10^4	22.6534
CpG+CY ((day 3/cycle,day 0/cycle), Day 27) (New Model)	1.7161×10^8 (1.50%)	5.8881×10^4	22.3570
CpG+CY ((day 3/cycle,day 0/cycle), Day 27) (Clinical results)	1.7423×10^8	1.2196×10^4	22.65

model) to within 1.5% (new model). This supports my data augmentation idea that, when more data are collected or generated artificially, the model gets better resolution in a small-data scenario.

4.4 Chapter Conclusion

In this work, to tackle small-data issues in the cancer immunotherapy model, I have formulated a data augmentation pipeline and have raised three hypotheses related to the effectiveness of the pipeline and the constraint-handling techniques. The results provide evidence that this data-augmentation procedure is universally effective in improving the resolution (viz. error size) of these constraint-handling techniques. The results also provide evidence that ε-SEC sustains its superior resolution compared to the other four constraint-handling techniques post data augmentation. Lastly, the results provide evidence that training the other constraint-handling techniques using the pseudo training data set generated by ε-SEC improves their resolution more than using their own respective pseudo training data set.

In future work, the formulated pipeline can be conducted in multiple rounds to study the effects on model resolution. This pipeline can also be applied to

other constrained optimization problems in real applications that have small-data issues.

Chapter 5

Finding High-Dimensional D-Optimal Designs for Logistic Models via Differential Evolution

5.1 Introduction

Optimal design problems frequently arise in scientific investigations when we want to obtain the most accurate statistical inference at minimal cost. For example, D-optimal designs are commonly used to estimate parameters in a statistical model by finding a series of design points to minimize the volume of the confidence ellipsoid of the parameters. Before optimization, nominal values for the parameters are required to replace unknown parameters of a model. Thus, the resulting optimal design is termed locally optimal [84, 85] because it depends on the nominal values for the parameters. Nominal values for the parameters may come from an expert's opinion or from a pilot study. The locally D-optimal design is then implemented to generate data to estimate the model parameters, and the estimated parameters become the nominal values in the next step. After a couple of iterations, the estimates are expected to become stable.

In the literature, the optimal design is usually found from theory. If the optimal design of a nonlinear model is found from theory, there is usually only one or two factors [86, 87] in the model. Such a theoretical approach encounters mathematical difficulties when the nonlinear model has more factors or the design criterion becomes complicated. Under such situations, classical numerical optimization techniques fail to find the locally optimal design or they become very inefficient. This is because as the number of factors in the model increases, the number of parameters in the model also increases. Consequently, the number of design points for the optimal design increases, resulting in having substantially many more variables to optimize. Thus, the design problem becomes quickly high-dimensional and also non-separable when factors interact with one another. Premature convergence can become a severe issue for high-dimensional and non-separable problems.

Nature-inspired metaheuristic algorithms are now increasingly applied to solve a large variety of complicated optimization problems [88, 89]. Particle Swarm Optimization (PSO) [90] is one such algorithm [90–92], which has been recently used to solve various optimal design problems in the literature [87,93,94]. In [87], Qiu et al. is the first to use the conventional PSO to find a variety of optimal designs for biomedical problems. In [93], PSO is applied to find optimal designs for a variety of mixture models, and in [94], PSO is modified to find minimax optimal designs, which is notoriously difficult to find because it has a non-differentiable design criterion. However, the D-optimal design problems in these papers have only 3 or fewer factors in their respective models, and so premature convergence may not be an issue. Since PSO exerts selective pressure onto some current best solutions termed as $gbest$ and $pbest$, models with 4 or more factors can cause PSO to experience premature convergence and make PSO less effective [95, 96].

Differential Evolution (DE) is a representative evolutionary algorithm. Muta-

tion, crossover and selection are three fundamental operations in DE [16,17]. One advantage that DE has over other evolutionary algorithms is that DE has fewer control parameters [18–20], and works well in handling numerical optimization problems [69, 70, 97, 98]. Compared with PSO, DE can alleviate the premature convergence issue moderately [16] since most of the mutation strategies of DE do not exert selective pressure on the current best solution [23, 45, 99–101]. However, based on some studies of DE variants for solving high-dimensional problems, there is a lack of specially designed mechanism to explore various novelty regions in the search space and to improve the diversity of the population.

To circumvent the above issues and also motivated by novelty search methods [102, 103] which are capable of escaping from local optima by trying some novelty solutions for efficient exploration, I propose a new novelty-based mutation strategy. At the start of the evolution, a portion of individuals is randomly selected as the novelty-based individuals, and their aim is to explore various individuals which are potentially novelty individuals. For each novelty-based individual, some difference vectors to be added to the current individual are sampled. Among these sampled difference vectors, the one which has the largest angle difference from the difference vectors used in the previous generation is selected. Each novelty-based individual explores a region of the search space different from the regions explored in the previous generation so that novelty solutions can be obtained. As evolution proceeds, various regions of the search space are explored and the diversity of the population is enhanced. The novelty-based mutation strategy is combined with two common mutation strategies, 'DE/rand/2' and 'DE/current-to-rand/1'. These two mutation strategies can balance the exploration and exploitation well at the early and medium stage of evolution as compared with other mutation strategies [104]. When the individuals obtained from these two mutation strategies converge, the novelty-based individuals can provide some novelty searching regions in the decision space to these convergent

individuals so that the diversity of these convergent individuals can be improved.

I apply the proposed algorithm to generate locally D-optimal designs for logistic models with several factors with and without interactions on various design spaces. Logistic regression models have a binary response with one or more factors and are among the most frequently used in scientific investigations across many disciplines. Using a broad simulation study, I show that my proposed algorithm consistently outperforms several of its top competitors. As an application, I also implement my DE-based algorithm to re-design a 10-factor car refueling experiment with both discrete and continuous factors, with and without factor interactions.

5.2 Background Knowledge of Generalized Linear Model and Optimal Design

A generalized linear model is commonly used to study the mean of a real response variable Y (i.e. Y is the actual output) based on a real input vector X with dimension n (i.e. n factors) [85]. In the field of optimal designs, Y is an observed response variable, which is a random variable as well [85]. Thus, it is reasonable to study the mean of Y. I focus on models with a binary response variable Y as 0 or 1 even though the methodology proposed herein applies more generally.

Let l be the index of an input vector, and $l = 1, \cdots, L$. For the interpretation in the field of optimal designs, l represents the index of a design point referred to as a support point, and L is the total number of support points in a design ξ. Let $\mu_l = E(Y_l)$, where $E(Y_l)$ represents the mean of Y_l based on a support point X^l. The relationship between inputs and outputs is postulated here as a linear model. Let η_l=$r(X^l)^T\beta$ be the linear predictor, where $r(X^l)^T$ is the linear model input generated based on support point X^l,

β is the linear model parameter set, and η_l is the corresponding model output. $r(X^l)^T$ is represented as $r(X^l)^T = \{1, x_1^l, \cdots, x_n^l\}$ (additive model) or $r(X^l)^T = \{1, x_1^l, \cdots, x_n^l, x_1^l x_2^l, \cdots, x_{n-1}^l x_n^l\}$ (model with all pairwise interaction terms).

To link the model outputs to actual outputs, let $g(.)$ be the link function such that $g(\mu_l) = \eta_l$ [105].

In this chapter, since the response variable is binary, according to [105], $g(.)$ uses a logistic link function:

$$g(\mu_l) = \log(\frac{\mu_l}{1-\mu_l}) = \eta_l. \tag{5.1}$$

I present a real case of a generalized linear model. In [106], an experiment related to finding factors that influence the failure of semiconductors when exposed to electrostatic discharge (ESD) was conducted. The response variable Y was whether or not a certain part of the semiconductor failed, and the model was a generalized linear model with five factors that are Wafer Type A, Wafer Type B, ESD, Pulse, and Voltage, respectively. Taking the random variable Y to be 1 if the semiconductor fails and 0 otherwise, we have $Y \sim Bernoulli(\mu)$ and μ is the mean of Y, and the model of interest is $\eta = \beta_0 + \beta_1 Wafer\ Type\ A + \beta_2 Wafer\ Type\ B + \beta_3 ESD + \beta_4 Pulse + \beta_5 Voltage$. To link model output to the mean of the response variable, we have $\eta = log(\mu/(1-\mu))$, where μ is the mean of the response variable Y.

For optimal designs, the design space is a user-selected compact set and contains all allowable combination levels of the factors to observe the response variable.

The goal of optimal designs of a generalized linear model is to find L distinct support points $X^1, \cdots, X^L$ to estimate the parameter set β in the linear model with n factors [84, 107] with respect to the optimality based on some statistical

criteria (e.g. the parameter set β is estimated with minimum variance with respect to β) when resources are given to take N input vectors, referred to as taking N observations in the field of optimal designs. Each observation of a support point X^l means taking X^l as the input vector for once. In optimal designs, since the observed response variable Y_l is a random variable based on the same support point X^l, it is common that the same support point X^l can be taken as an observation (i.e. input vector) for multiple times to derive the mean of Y_l. Thus, N is commonly larger than L.

It is required to determine the optimal number of support points, i.e. the value of L, the best choices of the support points $X^1, \cdots, X^L$ from a given design space, and the optimal number of replicates n_l at $X^l,\ l = 1, \cdots, L$ and $n_1 + \cdots + n_L = N$. This is a constrained optimization problem where some of the variables to be optimized are positive integers and constrained to sum to N.

Following [107], the worth of a L-point design ξ with n_l replicates at X^l is determined by its Fisher information matrix defined by

$$I_\xi = \sum_{l=1}^{L} n_l \Upsilon(\eta_l) r(X^l) r(X^l)^T, \tag{5.2}$$

where $\Upsilon(\eta_l) = \frac{(du_l/d_{\eta_l})^2}{u_l(1-u_l)}$. For the logistic regression model, the link function is the logit function in (5.1) and

$$\Upsilon(\eta_l) = \frac{1}{2 + e^{\eta_l} + e^{-\eta_l}} = \frac{e^{\eta_l}}{(1 + e^{\eta_l})^2}. \tag{5.3}$$

A locally D-optimal design maximizes the log-determinant of the Fisher information matrix I_ξ in (5.2), or equivalently minimizes the generalized variance of the estimates of the parameters [85]. Thus, D-optimal designs provide the most accurate estimates of all the model parameters in β. Clearly, I_ξ depends on β and so nominal values for β are required before optimization. Frequently, the

nominal values for β come from prior experiences or a pilot study [108].

I focus on approximate designs obtained by replacing each n_l by $p_l = n_l/N$, which is the proportion of the total observations to be taken at X^l. More generally, p_l is allowed to take on any value between 0 and 1 and doing so the problem is turned into a convex optimization problem where convex optimization tools can be used to find and verify the optimality of a design. Designs with weights $p_l's$ that sum to 1 are called approximate designs.

For D-optimality, the design criterion is $-log|I(\xi, \theta)|$ and this is a convex function over the space of all approximate designs on the given and compact design space of interest [85]. Following [109], the approximate design ξ is locally D-optimal among all designs if and only if for all X in the design space, the following checking condition is satisfied:

$$\frac{e^{r(X)^T\beta}}{\left(1+e^{r(X)^T\beta}\right)^2} r(X)^T I_\xi^{-1} r(X) - k \leq 0 \tag{5.4}$$

with equality at each support point of ξ. Here k is the dimension of β and the left-hand side of (5.4) is sometimes called the sensitivity function.

Often, it is difficult to obtain a theoretically optimal design. The optimal efficiency is frequently used to evaluate the worth of a design ξ by its efficiency relative to the optimal design ξ^* [85]. For D-optimality, the D-efficiency of a design ξ is

$$D - efficiency = \left(\frac{\det(I_\xi)}{\det(I_{\xi^*})}\right)^{1/k}. \tag{5.5}$$

If its D-efficiency is near 1, ξ is close to ξ^*. If the theoretical optimal design ξ^* is unknown, the proximity of a design ξ to ξ^* can be determined from convex analysis theory. Specifically, its D-efficiency is at least

$$D - efficiency \geq e^{-\theta/k} \tag{5.6}$$

where θ is the maximum positive value of the sensitivity function across the entire design space [110]. If the D-efficiency lower bound $e^{-\theta/k}$ is close to 1, the design ξ is close to the D-optimal design ξ^*.

5.3 Technical Details of the Proposed Algorithm NovDE

5.3.1 Overview

Since the D-optimal design problems in this work are high-dimensional and non-separable, premature convergence can be a severe issue with solutions easily getting trapped at local optima. Compounding the problem is that most state-of-the-art DE methods do not have a special mechanism to preserve diversity of their solutions and so the issue of premature convergence is not completely solved. For mutation strategies such as 'DE/rand-to-best/2', from the start of evolution, solutions tend to be close to the current best region thus limiting its exploration capability at the early stage. For mutation strategies such as 'DE/rand/2' or 'DE/current-to-rand/1', solutions tend to be close to each other at the early or medium stage of the evolution. Thus, to circumvent the issue of premature convergence of DE-based algorithms for solving high-dimensional and non-separable optimization problems, a mechanism for exploring diverse novelty regions of the search space should be specially designed and combined with other DE mutation strategies.

Assume that there are n factors in the model and a design with L support points is denoted by $\xi = ([x_1^1 x_2^1 \cdots x_n^1 p_1], \cdots, [x_1^l x_2^l \cdots x_n^l p_l], \cdots [x_1^L x_2^L \cdots x_n^L p_L])$, where p_l is the proportion of the total observations to be taken at the l-th support point $[x_1^l x_2^l \cdots x_n^l p_l]$. If each individual $X_{i,g}$ in the current generation g with population index i represents a combination of L support points, $X_{i,g}$ is

constructed as $X_{i,g} = (x_1^1 x_2^1 \cdots x_n^1 p_1 \cdots x_1^l x_2^l \cdots x_n^l p_l \cdots x_1^L x_2^L \cdots x_n^L p_L)$. For an additive model with no interactions among the factors, the dimension D of $X_{i,g}$ is $(n+1)L$.

I propose a new novelty-based DE-based algorithm, which is denoted as NovDE, to solve these complex optimization problems using a novelty-based mutation strategy. At the start of the evolution process, a group of individuals are randomly selected to be novelty-based individuals. To preserve the diversity of solutions, various regions of the search space are explored by these novelty-based individuals. Fig. 5.1 shows the difference vector $d_{i,g-1}$ which is the difference between the trial vector $u_{i,g-1}$ and the target vector $x_{i,g-1}$ in the previous generation $g-1$. For the current generation g and a user-selected value of m, m difference vectors are obtained. Each difference vector is calculated by $x_{r1,g} - x_{r2,g}$ where $x_{r1,g}$ and $x_{r2,g}$ are two randomly selected individuals from population in the current generation g. Thus, m difference vectors are represented as $d_{i,g}^1 \cdots d_{i,g}^m$. Fig. 5.1 displays the computed angle θ^s between a difference vector $d_{i,g}^s$ in the current generation g and the difference vector $d_{i,g-1}$ in the previous generation $g-1$ where $s = 1, 2, \cdots, m$. The difference vector $d_{i,g}^*$, which has the largest angle difference between $d_{i,g}^s$ and $d_{i,g-1}$ among the m samples, is added to the target vector $x_{i,g}$ to generate the mutant vector $v_{i,g}$. This is because the largest angle differences between $d_{i,g}^*$ and $d_{i,g-1}$ would enhance each novelty-based individual to explore a region in the search space entirely different from what was explored in the previous generation $g-1$. As the evolution proceeds, novelty-based individuals can gradually explore diverse novelty regions in the search space and the diversity of solutions can be preserved. The proposed novelty-based mutation strategy is combined with 'DE/rand/2' and 'DE/current-to-rand/1' since they can balance exploration and exploitation at the early or medium stage of evolution [104]. If the individuals obtained based on 'DE/rand/2' and 'DE/current-to-rand/1' are close to each other, the novelty-based

individuals can provide some novelty searching directions to those convergent individuals. The convergent individuals can either exploit in their current region of the search space or explore towards some novelty regions of the search space.

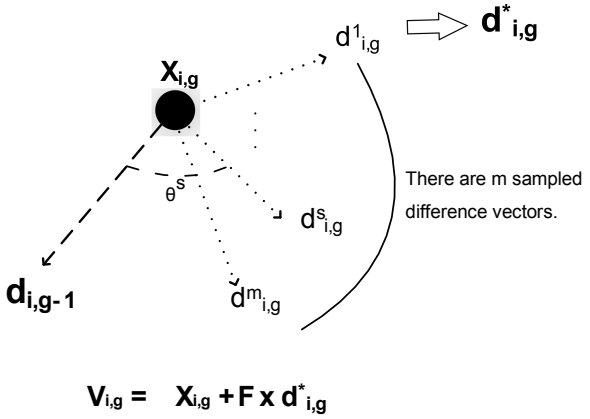

Figure 5.1: The operation of novelty-based mutation strategy. The target vector is $X_{i,g}$, and the difference vector from the previous generation is $d_{i,g-1}$. In the current generation, the m difference vectors are $d^1_{i,g} \cdots d^m_{i,g}$ and θ^s is the computed angle between $d^s_{i,g}$ and $d_{i,g-1}$, where $s = 1, \cdots, m$. The $d^s_{i,g}$ with the largest angle differences θ^s is selected to be $d^*_{i,g}$ and the mutant vector $V_{i,g}$ is generated based on $X_{i,g}$ and $d^*_{i,g}$.

The D-optimality criterion is a function of the information matrix in Equation (5.2), where x_l is part of the consecutive component in the decision variables. The term $x_l x_l^T$ in Equation (5.2) is the consecutive variables multiplied by each other, so physically proximate variables have stronger correlation and the problem is non-separable. According to [111], the crossover method Multiple Exponential Recombination (MER) can solve non-separable problems more efficiently than the binomial or exponential crossover method for the same CR rate. Further, MER updates the consecutive variables altogether which is more suitable for the structure of the decision variables in my problem. Thus, MER is selected to be the crossover method for my problem.

I adapt the control parameters F and CR to find locally D-optimal designs. The adaptation of F is the same as the state-of-the-art adaptive DE algorithm Self-adaptation Differential Evolution (SaDE) [104] where the F value for each

individual is generated from $F = N(0.5, 0.3)$. In this way, the value of F falls in the range [-0.4,1.4] with probability of 0.997, which covers exploration capability when F is large and exploitation capability when F is small [104]. Because the novelty-based individuals function as exploration, F is not required to be adaptive for both exploration and exploitation so it is fixed at 0.5. My adaptation method of CR in NovDE is new. A First-in-First-out (FIFO) memory $CRpool_k$ with a fixed size is applied, and the memory size for each mutation strategy k is proportional to the number of individuals involved in mutation strategy k. The $CRmean_k$ is the mean value of the successful CR values stored in $CRpool_k$ memory. The mean value of $CRmean_k$ for each strategy k is adapted based on the success values of CR stored in $CRpool_k$ for strategy k. This adaptation method updates the distribution of CR more frequently based on the solutions in the current evolution stage. In NovDE, CR value for each individual for mutation strategy k is generated from a Gaussian distribution as $CR = N(CRmean_k, 0.1)$. The initial value of $CRmean_k$ is selected to be 0.7 since if the value of CR is larger, the exploration would be encouraged. At the start of the evolution, exploration should be encouraged.

5.3.2 Algorithm Structure

The proposed algorithm NovDE is displayed in Algorithm 1. In NovDE, three mutation strategies 'DE/rand/2', 'DE/current-to-rand/1' and the proposed 'novelty-based DE' are employed to generate the mutant vector $V_{i,g}$. Populations are assigned to these three groups based on the pre-defined ratios $p1$ and $p2$. From step 9 to step 16, the proposed novelty-based DE mutation is presented. For each novelty-based individual $X_{i,g}$, F is fixed to be 0.5. The number of m difference vectors are represented as $d^1_{i,g} \cdots d^m_{i,g}$, and the value of m is user-selected. For each $d^s_{i,g}$ in the samples for $s = 1, \cdots, m$, the angle between $d^s_{i,g}$ and the difference vector from last generation $d_{i,g-1}$ is computed and denoted as θ^s. The $d^s_{i,g}$

with the largest θ_s is denoted as $d^*_{i,g}$. Then in step 15, the mutant vector $V_{i,g}$ can be generated based on target vector $X_{i,g}$ and difference vector $d^*_{i,g}$.

For the adaptation of CR, a first-in-first-out queue for each mutation strategy k is established as $CRpool_k$ with size LP_k. $CRpool_k$ is to store the values of CR that make the trial vector $U_{i,g}$ successfully replace the target vector $X_{i,g}$ for strategy k. The $CRmean_k$ is computed as the mean value of elements in $CRpool_k$, and CR for each individual is generated from $N(CRmean_k, 0.1)$. The crossover method is MER. After the crossover operation, the novelty-based individuals should update $d_{i,g}$ to be used in the next generation as $d_{i,g+1}$.

5.4 Results, Analysis and Discussion on Locally D-Optimal Design Problems with Various Settings

In this section, the locally D-optimal design problems of a generalized linear model with n factors are studied. The inputs of the problem are L support points $X^1, \cdots, X^L$ and their corresponding proportion $p_1, \cdots, p_L$. The outputs are the corresponding linear model output $\eta_1, \cdots, \eta_L$, which are used to estimate the mean of the corresponding actual output Y_l (a random binary response variable as 0 or 1) when input is X^l where $l = 1, \cdots, L$. The aim of locally D-optimal designs is that based on the nominal values of parameter set β in $\eta_l = r(X^l)^T\beta$, generate L support points and their corresponding proportions so that parameter set β can be estimated with minimum variance. According to [85], this aim can be formulated that the objective function is to maximize the log determinant of the Fisher information matrix defined in Equation (5.2) replacing n_l with p_l, and the decision variables are the concatenation of L support points and their proportions. The observation of actual output $Y^1, \cdots, Y^L$ and estimating the confidence intervals of parameter set β are conducted after the process of locally D-optimal design finishes and L support points and their proportions are

Algorithm 5.1 NovDE

Input: Target Vector $X_{i,g} = (x_{i,g}^1, x_{i,g}^2, \cdots, x_{i,g}^D)$, population size N, p1=0.45, p2=0.9, sample size m, $CRpool_k$ with size LP_k, where k represents the $k-$th mutation strategy.

Output: Trial Vector $U_{i,g} = (u_{i,g}^1, u_{i,g}^2, \cdots, u_{i,g}^D)$.

if $i \leq p1 * N$ **then**
  F is generated from $N(0.5, 0.3)$.
  $X_{i,g}$ performs 'DE/rand/2' to generate mutant vector $V_{i,g}$.
end if
if $p1 * N < i \leq p2 * N$ **then**
  F is generated from $N(0.5, 0.3)$.
  $X_{i,g}$ performs 'DE/current-to-rand/1' to generate mutant vector $V_{i,g}$.
end if
if $i > p2 * N$ **then**
  F is fixed to be 0.5.
  $d_{i,g-1}$ is the differences between trial vector $U_{i,g-1}$ and target vector $X_{i,g-1}$ in the previous generation $g-1$.
  Obtain number of m difference vectors as $d_{i,g}^1 \cdots d_{i,g}^m$.
  Compute the angle θ^s between $d_{i,g}^s$ and $d_{i,g-1}$ where $s = 1, 2, \cdots, m$.
  $d_{i,g}^*$ is the one with the largest θ^s among $d_{i,g}^1 \cdots d_{i,g}^m$.
  The mutant vector $V_{i,g} = X_{i,g} + F * d_{i,g}^*$.
end if
CR is generated from $(CRmean_k, 0.1)$ for different mutation strategy k.
Trial vector $U_{i,g}$ is generated based on mutant vector V_{ig} and crossover method MER and crossover rate CR.
if $i > p2 * N$ **then**
  $d_{i,g} = U_{i,g} - X_{i,g}$.
end if
if $f(U_{i,g}) < f(X_{i,g})$ **then**
  Record CR value into the corresponding $CRpool_k$.
  Perform first-in-first-out operation once the size of $CRpool_k$ exceeds LP_k.
  Update $CRmean_k$ as the mean value of elements in $CRpool_k$.
end if

generated, and they are not covered and discussed in this chapter.

I evaluate the performance of NovDE for finding locally D-optimal designs for logistic models on various design spaces with several factors. Specifically, I compare NovDE with six state-of-the-art variants of the DE algorithms. 'DE/rand/2/bin' [16] and SaDE [104] are effective in handling general numerical optimization problems; SaDE+MER [111] is effective in solving non-separable optimization problems; JADE [45] is an effective DE variant for its control parameter adaptation scheme; ANDE [46] and DDE-AMS [47] are effective in solving high-dimensional optimization problems. In order to validate the effectiveness of novelty-based mutation, I also compare the novelty-based mutation combined with the conventional crossover (i.e. binomial crossover), which is termed as NovDE-Bin. I compare using logistic models on various design spaces with seven continuous factors and five sets of nominal values. The design space of each factor is first selected to be on the prototype interval [-1, 1] before I vary the design space to [-3, 3], followed by the interval [0, 3]. I next describe the details of my experiment setup for comparing the eight algorithms.

5.4.1 Experiment Setup

I compare the performance of the proposed NovDE with NovDE-Bin and six competitive DE-based algorithms viz. 'DE/rand/2/bin' [16], ANDE [46], SaDE [104], SaDE+MER [111], JADE [45] and DDE-AMS [47] using 3 different design spaces to illustrate that NovDE is an effective DE variant in solving the high-dimensional D-optimal design of a logistic model.

1) Population size is 100.

2) The preset upper bound on the number of support points L is 100.

3) The dimension D of the problem to be optimized for seven factors without interactions is 800 (=$(7 + 1) \times 100$). The dimension for each support point is 8, which includes the number of factors (i.e. 7) and the dimension of the

corresponding portion of observations taken at each support point (i.e. 1).

4) For all my experiments, I set the maximum number of generation to be 20000.

5) The maximum number of run is 30.

6) For 'DE/rand/2/bin', according to [16], I set $F = 0.5$ and $CR = 0.9$.

7) For ANDE, I follow recommendations in [46] and generate $F1$, $F2$, and $F3$ from the uniform distribution on [0, 1]. I select CR accordingly to [46].

8) For SaDE and SaDE+MER, according to [104] and [111], I set $LP = 20$, initial value of $p_k = 0.25$ for each strategy, the initial CRm for each strategy as 0.5 and F sampled from the normal distribution [0.5,0.3].

9) For JADE, according to [45], I set $p = 0.05$, $c = 0.1$, $\mu_{CR} = 0.5$ and $\mu_F = 0.5$.

10) For DDE-AMS, according to [47], I use 4 sub-populations, and set $U_p = 25, T = 80, D_r = 0.3$, $\phi = 0.05, F = 0.5$ and $CR = 0.9$.

11) For NovDE and NovDE-Bin, I set $p1 = 0.45, p2 = 0.9$ and $m = 10$. Initial $CRmean_k$ for each strategy is 0.7 to encourage the exploration at the start of evolution. The upper bound of $CRmean_k$ is 0.9, and the lower bound of $CRmean_k$ is 0.1. For both 'DE/rand/2' and 'DE/current-to-rand/1', I set $LP = 50$, and for the novelty-based mutation strategy, I set $LP = 10$. I generate values of F for both 'DE/rand/2' and 'DE/current-to-rand/1' from the normal distribution [0.5, 0.3], and set $F = 0.5$ for the novelty-based mutation strategy.

12) I generate each of the nominal values of parameter set β defined in the linear predictor $\eta_l = r(X^l)^T\beta$ as $\beta^T = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7)$ in an additive 7-factor generalized linear model randomly from the interval [-1,1] without loss of generality. In this experiment, I generate five parameter sets and they are as follows:

$\beta 1 = (0.6294, 0.8116, -0.7460, 0.8268, 0.2647, -0.8049, -0.4430, 0.0938)$,

$\beta 2 = (-0.6710, 0.8256, -0.9221, 0.8348, 0.0538, 0.8664, 0.9186, 0.7741)$,

$\beta 3 = (-0.4926, -0.6280, -0.3283, 0.4378, 0.5283, -0.6120, -0.6837, -0.2061)$,

$\beta 4 = (-0.4336, 0.3501, -0.8301, 0.3295, 0.0853, 0.5650, 0.0870, 0.1688)$,

$\beta 5 = (0.8379, -0.5372, 0.1537, -0.1094, -0.2925, 0.2599, -0.8201, -0.8402)$.

13) The program is implemented in MATLAB R2017b.

14) In this chapter, the D-efficiency lower bound criterion as in Equation (5.6) is applied to evaluate the optimality of the generated design ξ and "DE/rand/2/bin" with F=0.5 and CR=0.9 is used to find the maximum positive value of the sensitivity function θ. Recall that this value is used to compute the D-efficiency lower bound of the design ξ, which is exp $(-\theta/k)$ where k is the dimension of β. In what is to follow, if a design has at least 95% D-efficiency, the design is considered close enough to the optimum.

5.4.2 Results and Discussions

Since the optimal designs of the logistic model under various sets of nominal values and design spaces are unknown, the average of the objective function values obtained in 30 runs is considered as one performance indicator. Since the aim is to maximize the log-determinant, the larger the objective function value is, the better is the performance of the algorithm. Another performance indicator is the success rate, which is the percentage of runs where the generated design has at least 95% D-efficiency. To judge whether the proposed NovDE algorithm outperforms each of the other seven DE-based algorithms in a statistically significant way, I employ a nonparametric statistical test called Wilcoxon rank-sum test [80] at the 5% significance level. For each algorithm, the numbers in the upper line in each entry represent the mean and standard deviation of the objective values. The numbers in the bottom line represent the success rate of the algorithm. The best values of the mean and success rates are in bold, and entries with * represent NovDE significantly outperforms the other algorithm based on Wilcoxon rank-sum test at the 0.05 significance level.

For each design space, there are 5 different settings with nominal values $\beta 1$ to $\beta 5$. Hence, for the three different design spaces, there are 15 different settings in total. For the 15 different settings, when NovDE is compared with the other seven DE algorithms, NovDE ranks first 9 times out of 15 in terms of the mean of the objective function values. Furthermore, in these 9 cases, excluding NovDE-Bin, NovDE significantly outperforms the other six DE algorithms in 6 out of 9 times. NovDE also ranks first 10 times out of 15 in terms of the success rate. These empirical results suggest that since the novelty-based mutation strategy combined with the MER crossover has advantages of superior capability of exploration [102] and maintaining the dependent variables structure [111], NovDE can work well in handling non-separable problems and can avoid getting trapped in local optima with higher chances. Thus, NovDE is more effective in generating locally D-optimal designs based on these 15 different settings compared with the other seven algorithms.

To show the effectiveness of novelty-based mutation alone, NovDE-Bin which is the novelty-based mutation with the conventional crossover-binomial crossover, is compared with the other 6 DE algorithms. For the 15 different settings, NovDE-Bin ranks first 3 times out of 15, and second 8 times out of 15 in terms of the mean of the objective function values. NovDE-Bin also ranks first 8 times out of 15 in terms of success rate. These empirical results suggest that the novelty-based mutation strategy presents better exploration capability and can prevent solutions from getting trapped at local optima, which is consistent with the advantages of novelty search methods as illustrated in [102].

To give a clearer picture of the performance difference between NovDE and the other seven DE algorithms, Fig. 5.2 plots the change of best-of-run objective function values over generations for each DE algorithm. The plots in Fig. 5.2 are based on nominal parameter $\beta 3$ and plots based on the other 4 sets of nominal values showed a similar pattern. As can be observed from Tables 5.1 to Table 5.3

and Fig. 5.2, both NovDE and NovDE-Bin clearly outperform 'DE/rand/2/bin' for all of the settings. Although 'DE/rand/2/bin' converges faster than NovDE, 'DE/rand/2/bin' has the issue of premature convergence so that the solutions tend to become close to each other and its exploration capability is deteriorated. The better performance of both NovDE-Bin and NovDE demonstrates that exploration is important for solving high-dimensional non-separable problems which have local optima. Furthermore, novel information collected from exploration can be provided to individuals generated from 'DE/rand/2' and 'DE/current-to-rand/1' to enhance both exploration and exploitation. As shown in Fig. 5.2, NovDE has the best converged objective function values close to the global optimum on various design spaces.

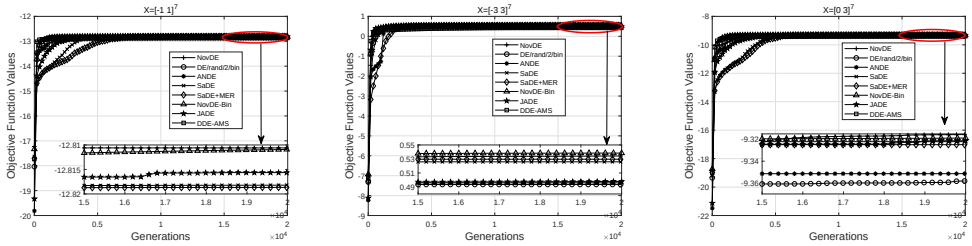

Figure 5.2: Average best-of-run objective function values of 30 independent runs over generations for $X = [-1\ 1]^7$, $X = [-3\ 3]^7$ and $X = [0\ 3]^7$, respectively. The nominal parameter is $\beta 3$.

Since $CRmean$ can represent the overall CR values of the individuals under different strategies, it is instructive to plot the $CRmean$ values versus generations

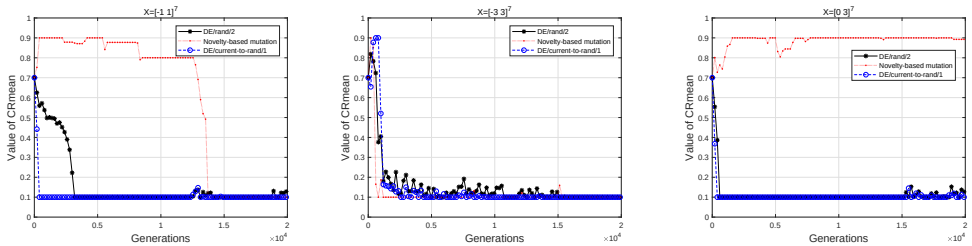

Figure 5.3: Adaptation behaviors of the median run among the 30 multiple runs of the CRmean values in NovDE for $X = [-1\ 1]^7$, $X = [-3\ 3]^7$ and $X = [0\ 3]^7$, respectively. The nominal parameter is $\beta 3$.

Table 5.1: Performances of NovDE, NovDE-Bin and six competitors for finding locally D-optimal designs on $[-1, 1]^7$ using 5 sets of nominal values. In each cell, the numbers in the upper line are the mean and standard deviation of the values of the objective function over 30 runs, and the number at the bottom line is its success rate. For each set of nominal values, the best values of the mean and success rates are in bold. The entries with an $*$ means that NovDE significantly outperforms the other algorithm based on Wilcoxon rank-sum test.

Algorithm	$\boldsymbol{\beta 1}$	$\boldsymbol{\beta 2}$	$\boldsymbol{\beta 3}$	$\boldsymbol{\beta 4}$	$\boldsymbol{\beta 5}$
NovDE	**-13.2018 (0.0079)**	-13.5845 (0.0153)	**-12.8106 (0.0054)**	-12.6012 (0.0041)	**-13.0221 (0.0093)**
	86.67%	90%	**73.33**%	83.33%	**90**%
NovDE-Bin	-13.2028 (0.0077)*	-13.5958 (0.0526)*	-12.8116 (0.0049)	**-12.5946 (0.0044)**	-13.0224 (0.0242)
	86.67%	90%	70%	**96.67**%	86.67%
DE/rand/2/bin	-13.2274 (0.0153)*	-13.5973 (0.0092)*	-12.8459 (0.0128)*	-12.6412 (0.0253)*	-13.0650 (0.0205)*
	3.33%	0%	3.33%	3.33%	0%
ANDE	-13.2340 (0.0146)*	-13.6009 (0.0197)*	-12.8460 (0.0095)*	-12.6330 (0.0144)*	-13.0543 (0.0163)*
	3.33%	6.67%	0%	3.33%	3.33%
SaDE	-13.2030 (0.0060)*	-13.5778 (0.0052)	-12.8195 (0.0106)*	-12.6006 (0.0078)	-13.0505 (0.0651)*
	73.33%	93.33%	23.33%	**96.67**%	73.33%
SaDE+MER	-13.2032 (0.0071)*	**-13.5761 (0.0012)**	-12.8186 (0.0057)*	-12.6022 (0.0052)	-13.0233 (0.0059)*
	80%	**96.67**%	53.33%	90%	76.67%
JADE	-13.2035 (0.0021)*	-13.5799 (0.0021)	-12.8156 (0.0033)*	-12.5958 (0.0072)	-13.0248 (0.0039)*
	30%	73.33%	53.33%	40%	33.33%
DDE-AMS	-13.2390 (0.0096)*	-13.5961 (0.0186)*	-12.8690 (0.0154)*	-12.6258 (0.0169)*	-13.0549 (0.0204)*
	3.33%	3.33%	0%	6.67%	0%

for each design space. Fig. 5.3 plots the $CRmean$ values based on the median run using β_3 as nominal values for the same reason explained earlier. In Fig. 5.3, I observe that the $CRmean$ values for 'DE/rand/2' and 'DE/current-to-rand/1' would converge to 0.1, which is the lower bound of the $CRmean$ in NovDE. The variation of the $CRmean$ values for novelty-based strategy presents distinct patterns under different design spaces. When $X = [-1\ 1]^7$ and $X = [-3\ 3]^7$, the $CRmean$ converge to 0.1, which is the lower bound of the $CRmean$ in NovDE. Under these two design spaces, the decrease of $CRmean$ values indicates their

Table 5.2: Performances of NovDE, NovDE-Bin and six competitors for finding locally D-optimal designs on $[-3, 3]^7$ using 5 sets of nominal values. In each cell, the numbers in the upper line are the mean and standard deviation of the values of the objective function over 30 multiple runs, and the number at the bottom line is its success rate. For each set of nominal values, the best values of the mean and success rates are in bold. The entries with $*$ represent NovDE significantly outperforms the other algorithm based on Wilcoxon rank-sum test.

Algorithm	β**1**	β**2**	β**3**	β**4**	β**5**
NovDE	-0.1052 (0.0048)	-0.4441 (0.0038)	0.5343 (0.0128)	**0.7487 (0.0076)**	**0.3678 (0.0028)**
	100%	80%	**46.67**%	**93.33**%	**90**%
NovDE-Bin	-0.1054 (0.0049)	-0.4438 (0.0036)	**0.5384 (0.0177)**	0.7476 (0.0051)	0.3645 (0.0055)
	93.33%	**90**%	40%	**93.33**%	**90**%
DE/rand/2/bin	-0.1209 (0.0135)*	-0.4581 (0.0221)*	0.4879 (0.0283)*	0.7140 (0.0127)*	0.3396 (0.0061)*
	43.33%	23.33%	0%	6.67%	16.67%
ANDE	-0.1131 (0.0080) *	-0.4561 (0.0113)*	0.4940 (0.0207)*	0.7097 (0.0295)*	0.3412 (0.0138)*
	53.33%	40%	3.33%	3.33%	6.67%
SaDE	-0.1033 (0.0027)	**-0.4435 (0.0040)**	0.5234 (0.0155)*	0.7457 (0.0109)	0.3641 (0.0070)
	80%	**90**%	40%	86.67%	73.33%
SaDE+MER	**-0.1022 (0.0032)**	-0.4440 (0.0030)	0.5240 (0.0143) *	0.7474 (0.0067)	0.3661 (0.0061)
	100%	86.67%	36.67%	90%	83.33%
JADE	-0.1031 (0.0025)	-0.4436 (0.0038)	0.4979 (0.0274)	0.7467 (0.0058)	0.3667 (0.0031)
	50%	46.67%	10%	83.33%	83.33%
DDE-AMS	-0.1241 (0.0320)*	-0.4590 (0.0304)*	0.4278 (0.0350)*	0.7130 (0.0197)*	0.3547 (0.0098)*
	30%	16.67%	0%	3.33%	33.33%

exploration capability tends to be restricted as evolution proceeds; for $X = [-3\ 3]^7$, the $CRmean$ values would converge faster. When $X = [0\ 3]$, the $CRmean$ converges to around 0.9, which is the upper bound of the $CRmean$ in NovDE. The increase of $CRmean$ values indicates their exploration capability tends to be enhanced as evolution proceeds. For different design spaces, the $CRmean$ for the novelty-based strategy presents its adaptation to the exploration capability.

Table 5.4 to Table 5.6 present the support points of the locally D-optimal designs when $\beta 3$ is the set of nominal values. Interestingly, each support point

Table 5.3: Performances of NovDE, NovDE-Bin and six competitors for finding locally D-optimal designs on $[0, 3]^7$ using 5 sets of nominal values. In each cell, the numbers in the upper line are the mean and standard deviation of the values of the objective function over 30 multiple runs, and the number at the bottom line is its success rate. For each set of nominal values, the best values of the mean and success rates are in bold. The entries with $*$ represent NovDE significantly outperforms the other algorithm based on Wilcoxon rank-sum test.

Algorithm	$\boldsymbol{\beta 1}$	$\boldsymbol{\beta 2}$	$\boldsymbol{\beta 3}$	$\boldsymbol{\beta 4}$	$\boldsymbol{\beta 5}$
NovDE	**-8.2056 (0.0091)**	**-11.0117 (0.0023)**	**-9.3156 (0.0191)**	-7.6625 (0.0183)	**-9.1025 (0.0124)**
	83.33%	**100**%	**26.67**%	83.33%	76.67%
NovDE-Bin	-8.2064 (0.0047)	-11.0134 (0.0027)	-9.3188 (0.0314)	**-7.6508 (0.0038)**	-9.1076 (0.0171)
	80%	**100**%	23.33%	**96.67**%	**80**%
DE/rand/2/bin	-8.2344 (0.0196)*	-11.0293 (0.0121)*	-9.3562 (0.0193)*	-7.6977 (0.0184)*	-9.1549 (0.0194)*
	3.33%	16.67%	3.33%	6.67%	3.33%
ANDE	-8.2430 (0.0194)*	-11.0249 (0.0055)*	-9.3500 (0.0256)*	-7.6861 (0.0163)*	-9.1510 (0.0135)*
	3.33%	46.67%	0%	0%	3.33%
SaDE	-8.2147 (0.0145)*	-11.0141 (0.0019)*	-9.3282 (0.0213)*	-7.6571 (0.0063)	-9.1066 (0.0128)*
	66.67%	90%	10%	86.67%	46.67%
SaDE+MER	-8.2067 (0.0052)	-11.0139 (0.0030)*	-9.3348 (0.0270)*	-7.6594 (0.0121)	-9.1098 (0.0147)*
	70%	86.67%	16.67%	76.67%	56.67%
JADE	-8.2085 (0.0019)*	-11.0330 (0.0093)*	-9.3213 (0.0207)*	-7.6517 (0.0040)	-9.1095 (0.0032)*
	20%	20%	6.67%	33.33%	36.67%
DDE-AMS	-8.2212 (0.0145)*	-11.0239 (0.0114)*	-9.3939 (0.0279)*	-7.7216 (0.0250)*	-9.1529 (0.0377)*
	3.33%	26.67%	0%	0%	3.33%

of these locally D-optimal designs has at most one factor level supported at its non-extreme values. This observation may provide an impetus for further study using analytical tools.

Table 5.4: NovDE-generated locally D-optimal design for the logistic model with seven variables when the vector of nominal values for the parameters is $\beta 3$, and $X = [-1, 1]^7$.

Support point	$X1$	$X2$	$X3$	$X4$	$X5$	$X6$	$X7$	P_i
1	1	-1	1	-1	-1	-1	-1	0.0230
2	1	-1	1	-1	-1	-1	1	0.0160
3	1	1	1	1	1	-1	1	0.0255
4	1	1	-1	1	1	-1	-1	0.0223
5	1	-1	1	-1	-1	1	1	0.0152
6	-1	-1	1	1	1	1	1	0.0212
7	-1	1	1	-1	1	-1	1	0.0269
8	1	-1	1	-1	1	-1	-1	0.0101
9	1	-1	1	-1	-1	1	-1	0.0269
10	-1	1	1	-1	-1	1	-1	0.0117
11	1	-1	-1	1	1	-1	1	0.0269
12	-1	-1	1	1	1	1	1	0.0142
13	-1	1	1	1	1	-1	1	0.0219
14	1	1	1	-1	-1	-1	1	0.0182
15	1	1	-1	1	-1	-1	1	0.0183
16	-1	-1	1	1	1	1	-1	0.0199
17	-1	-1	-1	-1	-1	-1	1	0.0269
18	-1	-1	-1	-1	1	-1	-1	0.0101
19	-1	-1	1	-1	1	1	-1	0.0269
20	-1	1	1	-1	-1	-1	-1	0.0163
21	1	-1	1	1	-1	-1	1	0.0102
22	-1	1	-1	-1	1	-1	-1	0.0269
23	-1	1	-1	1	-1	-1	1	0.0241
24	-1	-1	-1	-1	-1	-1	1	0.0213
25	-1	-1	-1	-1	-1	1	1	0.0269
26	-1	-1	1	-1	1	-1	-1	0.0269
27	1	-1	-1	1	-1	-1	-1	0.0269
28	-1	1	-1	-1	-1	1	-1	0.0184
29	-1	-1	-1	1	-1	1	-1	0.0269
30	1	-1	-1	1	-1	1	-1	0.0161
31	1	-1	1	1	-1	1	-1	0.0165
32	1	-1	-1	-1	-1	-1	1	0.0143
33	-1	1	-1	-1	1	-1	-1	0.0269
34	-1	-1	1	1	1	1	-1	0.0124
35	-1	1	1	1	1	-1	-1	0.0269
36	-1	1	1	-1	-1	1	1	0.0204
37	1	1	1	1	-1	1	-1	0.0269
38	-1	1	-1	1	1	-1	-1	0.0103
39	1	-1	1	1	1	-1	1	0.0269
40	-1	1	-1	1	-1	1	1	0.0152
41	1	1	-1	1	-1	-1	-1	0.0150
42	1	1	1	-1	-1	-1	-1	0.0260
43	-1	-1	-1	-1	-1	-1	1	0.0144
44	-1	1	-1	1	-1	1	-1	0.0260
45	-1	-1	-1	1	1	1	1	0.0269
46	1	1	1	1	-1	1	1	0.0203
47	-1	1	1	1	-1	1	1	0.0245
48	1	-1	-1	1	1	-1	-1	0.0269

Table 5.5: NovDE-generated locally D-optimal design for the logistic model with seven variables when the vector of nominal values for the parameters is $\beta 3$, and $X = [-3, 3]^7$.

Support point	$X1$	$X2$	$X3$	$X4$	$X5$	$X6$	$X7$	P_i
1	-3	-3	-3	-3	3	-3	-3	0.0311
2	3	3	3	3	-3	3	3	0.0100
3	-3	-3	-3	-3	3	-3	3	0.0347
4	3	-3	3	3	3	-2.9971	3	0.0480
5	-3	-3	3	-3	3	2.9217	-3	0.0292
6	3	-3	-3	-3	-3	-3	-3	0.0100
7	-3	3	-3	-3	2.0891	-3	-3	0.0107
8	-3	-3	3	-3	-2.7545	3	3	0.0468
9	3	3	3	3	3	-3	-3	0.0187
10	3	3	3	3	-3	3	-3	0.0298
11	-3	-3	3	3	3	3	3	0.0409
12	3	-3	-3	3	-3	3	-3	0.0403
13	3	3	3	-3	-3	-3	3	0.0100
14	3	3	-3	3	-3	-3	3	0.0100
15	-3	3	3	3	3	3	-3	0.0385
16	3	-3	-3	-3	-3	-3	-3	0.0260
17	3	3	3	-3	-3	-3	-3	0.0517
18	-3	-3	-3	3	3	3	-3	0.0338
19	3	-3	3	-3	3	-3	-3	0.0375
20	-3	3	-3	-3	-3	-3	3	0.0303
21	3	3	-3	3	-3	-3	3	0.0258
22	-2.9190	3	-3	3	-3	3	-3	0.0451
23	-3	3	3	-3	-3	3	3	0.0431
24	3	-3	-3	3	3	-3	-3	0.0100
25	3	-3	-3	-3	-3	-3	3	0.0316
26	-3	-3	-3	-3	-3	3	-3	0.0136
27	-3	3	3	-3	3	-3	-3	0.0356
28	-3	3	-3	3	-3	3	3	0.0162
29	3	-3	3	3	-3	3	3	0.0438
30	-3	3	3	-3	-3	3	3	0.0305
31	3	-3	-3	3	-3	3	-3	0.0140
32	-3	3	-3	3	3	-3	3	0.0517
33	3	3	3	3	3	-3	3	0.0512

Table 5.6: NovDE-generated locally D-optimal design for the logistic model with seven variables when the vector of nominal values for the parameters is $\beta 3$, and $X = [0, 3]^7$.

Support point	$X1$	$X2$	$X3$	$X4$	$X5$	$X6$	$X7$	P_i
1	3	0	3	3	3	0	0	0.0232
2	0	3	3	3	3	0	0	0.0262
3	0	3	3	0	0	0	3	0.0151
4	0	3	3	3	0	3	0	0.0270
5	3	3	3	3	0	0	0	0.0191
6	3	0	3	0	0	0	0	0.0113
7	0	0	3	3	0	3	0	0.0103
8	3	0	3	0	0	0	0	0.0215
9	0	3	3	0	0	0	3	0.0220
10	3	0	0	3	0	0	0	0.0106
11	3	0	3	3	0	0	3	0.0366
12	0	0	0	3	3	0	3	0.0389
13	0	3	3	3	0	0	0	0.0318
14	0	3	0	3	0	0	3	0.0309
15	0	0	3	3	0	3	3	0.0171
16	0	0	3	3	3	0	3	0.0315
17	0	3	3	0	0	0	3	0.0385
18	0	0	0	0	0	0	0	0.0367
19	0	0	3	0	3	0	0	0.0319
20	0	0	3	3	3	0	0	0.0323
21	0	0	0	3	3	0	0	0.0128
22	0	0	0	3	0	0	3	0.0217
23	3	3	3	3	0	0	0	0.0333
24	0	0	3	3	3	0	3	0.0114
25	0	0	0	3	0	3	0	0.0389
26	0	0	0	0	0	0	0	0.0107
27	0	0	3	0	0	3	0	0.0271
28	0	0	3	0	0	0	0	0.0380
29	0	3	3	3	3	0	0	0.0263
30	3	0	3	3	0	0	3	0.0298
31	0	0	3	3	0	3	0	0.0346
32	3	0	0	3	0	0	0	0.0316
33	0	3	0	3	0	0	0	0.0312
34	0	0	3	0	0	0	0	0.0131
35	0	0	0	3	0	0	3	0.0225
36	0	3	0	3	0	0	0	0.0138
37	0	3	3	3	0	3	3	0.0300
38	3	0	3	3	0	0	0	0.0100
39	0	0	3	3	3	3	0	0.0118
40	0	0	3	0	0	0	3	0.0146
41	0	0	3	3	0	3	3	0.0242

5.5 Results, Analysis and Discussion on a Real Application

I now apply the proposed NovDE algorithm to a real application that is to redesign a ten-factor experiment to test the functionality of a vision-based car refueling system [112]. The investigators were interested in finding whether a computer-controlled nozzle was able to insert into gas pipe correctly or not, implying that the response variable in the study is binary as 0 (fail) or 1 (success). Table 5.7 lists the ten factors, which are the factors of each input vector. Four factors are discrete, each with two levels -1 or +1, and six factors are continuous. Table 5.7 shows the range of values for each continuous factor and they do vary considerably. The proposed NovDE algorithm is applied to find a locally D-optimal design for this high-dimensional nonlinear model with mixed factors using the vector of nominal values $\beta = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9, \beta_{10})^T = (3, 0.5, 0.75, 1.25, 0.8, 0.5, 0.8, -0.4, -1, 2.65, 0.65)$ from literature [112].

Design issues for this ten-factor experiment were also considered in [113] but without interaction terms. In practice, the binary response is likely dependent on the joint changes in two or more of the factors, suggesting that interaction terms should be in the model. To fix ideas, I include five pairwise interactions into the model and believe that this is the first design work for such a high-dimensional logistic model. Previous attempts using conventional optimization methods like multiplicative and modified Fedorov-Wynn algorithms did not converge for this problem [113]. The vector of nominal values for the model with the five selected pairwise interactions is $\beta = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9, \beta_{10}, \beta_{1,9}, \beta_{2,5}, \beta_{3,4}, \beta_{6,7}, \beta_{8,10})^T = (3, 0.5, 0.75, 1.25, 0.8, 0.5, 0.8, -0.4, -1, 2.65, 0.65, 0.01, -0.02, 0.03, -0.04, 0.05)^T$ based on literature [112].

Some of the tuning parameters used to find the locally D-optimal designs are the population size, maximum number of generations and maximum number of

support points. For the model without factor interactions, the population size is 100, and the maximum number of generations is 10000. The maximum number of support points L is set to 100 so the dimension D of the problem is 1100 (=$(10 + 1) \times 100$). The dimension for each support point is 11, which includes the number of factors (i.e. 10) and the dimension of the corresponding portion of observations taken at each support point (i.e. 1). For the model with factor interactions, the population size is 100, and the maximum number of generations is 20000. The maximum number of support points L is 100 so the dimension D of the problem is 1600 (=$(10 + 1 + 5) \times 100$).

Due to the number of factors in this study, it is hard to construct and visually appreciate the high-dimensional sensitivity function of the generated design to confirm its optimality. I apply "DE/rand/2/bin" with F=0.5 and CR=0.9 to find the maximum positive value of the function and compute its D-efficiency lower bound. The lower bound D-efficiency is defined as exp $(-\theta/k)$ where k is the dimension of the model parameter β. Since the variables of this problems are mixed, the variation of lower bound D-efficiency is very large. In what is to follow, if a design has at least 90% D-efficiency, I accept the design as close enough to the optimum.

5.5.1 Without Factor Interactions

Table 5.8 compares the mean of locally D-optimal objective value and success rate of NovDE with NovDE-Bin and the other six differential evolution algorithms. Wilcoxon rank-sum test [80] is also conducted at the 5% significance level. In Table 5.8, both the mean of the objective value and the success rate of NovDE-Bin are the highest. NovDE ranks the second. Both NovDE-Bin and NovDE significantly outperform all the other six algorithms. Thus, my empirical results support the effectiveness of novelty-based mutation strategy in solving the car refueling experiment. By extension, my work suggests that the

Table 5.7: Factor types and levels for the car refueling experiment.

Type	Factor	Level	
		Low	High
Discrete	Ring Type	White paper	Reflective
	Lighting	Room lighting	2 flood lights and room lights
	Sharpen	No	Yes
	Smooth	No	Yes
Continuous	Lighting angle	from 50 degrees to 90 degrees	
	Gas-cap angle (Z axis)	from 30 degrees to 55 degrees	
	Gas-cap angle (Y axis skew)	from 0 degrees to 10 degrees	
	Car distance	from 18 in. to 48 in.	
	Reflective ring thickness	from 0.125 in. to 0.425 in.	
	Threshold step value	from 5 to 15	

NovDE and NovDE-Bin are effective for searching locally D-optimal designs for high-dimensional non-separable problems with mixed variables on various design spaces. In problems with mixed factors, the solutions obtained from D-optimal design with mixed factors are more likely to be convergent at the early evolution stage. Thus, it is more crucial to handle the premature convergence issue especially for the problems with mixed factors. NovDE and NovDE-Bin have advantages in handling the premature convergence issue and preserve the diversity of solutions. As a result, NovDE performs even better than it performs on handling D-optimal design with continuous factors. Table 5.9 lists the locally D-optimal design for the car refueling experiment, and 12 support points are generated. The design criterion value is -35.9178. A direct calculation shows that the D-efficiency lower bound for the generated design is 94.60%. This is not surprising even though I set the lower bound to be 90% for this problem. The reason is because the algorithm is not monotonic in the sense that it does

not necessarily produce increasingly more efficient designs with each iteration. Another reason is that some higher D-efficiency optimal design may exist in the continuous design spaces instead of the mixed design spaces. Based on Table 5.9, the common rule is observed: for each support point, there is at most one factor value not at the boundary of the design space. This is consistent with the observation in Section 5.4.

Table 5.8: Comparisons of the performance of NovDE, NovDE-Bin and six competitors for the car refueling experiments without factor interactions. The best values of the mean and success rates are in bold. The entries with $*$ represent NovDE significantly outperforms the other algorithm based on Wilcoxon rank-sum test.

Algorithm	**Success Rate**	**Mean (std)**
NovDE	**90**%	-35.9390 (0.1060)
NovDE-Bin	**90**%	-35.9180 (0.0035)
DE/rand/2/bin	13.33%	-37.4963 (0.8482)*
ANDE	26.67%	-36.7708 (0.8923)*
SaDE	70%	-35.9480 (0.0485)*
SaDE+MER	46.67%	-36.2242 (0.7181)*
JADE	70%	-35.9645 (0.0911)*
DDE-AMS	10%	-39.5935 (0.5128)*

Table 5.9: NovDE-generated locally D-optimal design for car refueling experiment without factor interactions.

Support point	Ring Type	Lighting	Sharpen	Smooth	Lighting Angle	Z axis	Y axis skew	Car Dist.	Ring Thick.	Threshold Step-size	P_i
1	-1	-1	-1	1	50	30	10	48	0.1250	5	0.0909
2	-1	-1	-1	-1	50	30	4.1991	48	0.1250	5	0.0909
3	-1	-1	-1	-1	50	30	10	48	0.1250	8.5698	0.0909
4	-1	-1	-1	-1	50	30	10	48	0.1250	5	0.0807
5	-1	-1	-1	-1	54.6407	30	10	48	0.1250	5	0.0909
6	-1	1	-1	-1	50	30	10	48	0.1250	5	0.0909
7	1	-1	-1	-1	50	30	10	48	0.1250	5	0.0752
8	1	-1	-1	-1	50	30	10	48	0.4250	5	0.0397
9	-1	-1	-1	-1	50	32.9005	10	48	0.1250	5	0.0909
10	-1	-1	-1	-1	50	30	10	45.6796	0.1250	5	0.0909
11	-1	-1	-1	-1	50	30	10	48	0.4250	5	0.0772
12	-1	-1	1	-1	50	30	10	48	0.1250	5	0.0909

5.5.2 With Factor Interactions

It seems realistic that there are factor interactions between Ring type and Reflective ring thickness, Lighting and Lighting angle, Sharpen and Smooth, Gas-cap angle (Z axis) and Gas-cap angle (Y axis skew) and Car distance and Threshold step value, respectively. The former two interactions are between a discrete factor and a continuous factor; the third interaction is between a discrete factor and a discrete factor and the latter two interactions are between a continuous factor and a continuous factor. In practice, the researcher uses content information to specify interaction terms in the model and implements a parsimonious model. My conjecture that interaction terms were ignored in earlier design work for such a model is to simplify the design construction.

Table 5.10: Comparisons of the performance of NovDE, NovDE-Bin and six competitors for the car refueling experiment with factor interactions. The best values of the mean and success rates are in bold. The entries with $*$ represent NovDE significantly outperforms the other algorithm based on Wilcoxon rank-sum tests

Algorithm	**Success Rate**	**Mean (std)**
NovDE	**80**%	**-71.5401 (0.3365)**
NovDE-Bin	70%	-71.5640 (0.3453)
DE/rand/2/bin	0%	-74.4495 (1.4082)*
ANDE	23.33%	-72.0390 (0.6751)*
SaDE	3.33%	-71.7135 (0.3182)*
SaDE+MER	20%	-71.6072 (0.1860)*
JADE	30%	-71.5843 (0.3562)*
DDE-AMS	6.67%	-71.7058 (0.3136)*

Table 5.10 compares the mean of locally D-optimal objective value and success rate of NovDE with NovDE-Bin and the other six differential evolution algorithms. Wilcoxon rank-sum test [80] is also conducted at the 5% significance level. In Table 5.10, both the mean of the objective value and the success rate of NovDE is the highest, and NovDE-Bin is the second highest. NovDE

Table 5.11: NovDE-generated locally D-optimal design for the car refueling experiment with five pairwise factor interactions.

Support point	Ring Type	Lighting	Sharpen	Smooth	Lighting Angle	Z axis	Y axis skew	Car Dist.	Ring Thick.	Threshold Step-size	P_i
1	-1	1	-1	-1	50	35.5152	10	48	0.1250	5	0.0625
2	-1	1	1	1	50	30	10	48	0.1250	5	0.0625
3	-1	-1	-1	-1	50	30	10	48	0.1250	5	0.0466
4	-1	1	-1	-1	50	30	10	48	0.1250	5	0.0511
5	-1	1	-1	-1	50	34.5716	8.8571	48	0.1250	5	0.0625
6	-1	1	-1	-1	50	30	8.6212	48	0.1250	5	0.0625
7	-1	1	-1	-1	50	30	10	48	0.4250	5	0.0486
8	-1	1	-1	-1	50	30	10	45.1640	0.1250	5.6974	0.0625
9	1	1	-1	-1	50	30	10	48	0.4250	5	0.0625
10	-1	1	-1	-1	50	30	10	48	0.1250	5.7233	0.0625
11	-1	1	1	-1	50	30	10	48	0.1250	5	0.0625
12	-1	1	-1	-1	54.5960	30	10	48	0.1250	5	0.0625
13	1	-1	-1	-1	50	30	10	48	0.1250	5	0.0242
14	1	1	-1	-1	50	30	10	48	0.1250	5	0.0499
15	-1	-1	-1	-1	54.1003	30	10	48	0.1250	5	0.0625
16	-1	-1	-1	-1	50	30	10	48	0.4250	5	0.0296
17	-1	1	-1	1	50	30	10	48	0.1250	5	0.0625
18	-1	1	-1	-1	50	30	10	45.0586	0.1250	5	0.0625

significantly outperforms all the other six algorithms. I observe that the overall outperformance of the NovDE and NovDE-Bin algorithms relative to the other six algorithms are less dramatic than when the model has no interaction terms, my results still show it is effective in solving non-separable high-dimensional locally D-optimal design problems with mixed factors on various design spaces. In particular, it shows the NovDE is able to handle premature convergence and non-separable issues well in complex optimization problems. The proposed NovDE algorithm can produce optimal designs for a more realistic situation and so represents an advancement. Table 5.11 shows the optimal design for the car refueling experiment with five pairwise factor interactions. There are 18 support points, and the design criteria value is -71.4284. The design has a D-efficiency of 95% or higher. An interesting note is that Table 5.11 shows each support point can have one or more factors supported other than at its extreme values. This violates the common rule mentioned earlier in a model without factor interactions and serves to show that as the model gets more complicated, the structure of the optimal design also becomes harder to characterize and understand.

5.6 Chapter Conclusion

I propose a DE-based algorithm NovDE to search for locally D-optimal designs for logistic models with multiple factors that may or may not interact with one another. I employ a new novelty-based mutation strategy to explore various regions of the search space so that the diversity of the population is preserved. The new novelty-based mutation strategy is collaborated with 'DE/rand/2' and 'DE/current-to-rand/1' which can balance exploration and exploitation at early or medium stage of the evolution. Both convergence and diversity of the population are enhanced, and premature convergence issues are alleviated. I have demonstrated that NovDE provides the best objective function values and success rates compared with seven other DE-related evolutionary algorithms. NovDE also outperforms the others in terms of finding a highly efficient D-optimal design for the ten-factor car refueling study where there are discrete and continuous factors in the logistic model and some of them interact with one another. My empirical results also show that the distribution of the support points for optimal designs for models with interaction terms are more complex than those for models without interaction terms.

I focus on logistic models which are the most commonly used in practice to model binary responses. I expect the proposed algorithm works for other link functions as well, including cases when the response is continuous and there are many mixed factors. Future study includes testing the capability of my proposed algorithm for tackling these problems and multiple-objective optimal design problems.

Chapter 6

Conclusion & Future Work

6.1 Conclusion

The primary focus of this thesis is to propose new evolutionary computation algorithms for solving optimization problems with decision variables that have complex characteristics of being with constraints or are high dimensional.

In Chapter 3, a new evolutionary constraint-handling technique ε-SEC has been proposed. ε-SEC has successfully addressed difficulties of current evolutionary algorithms in balancing the two conflicting tasks of optimizing objective function and reducing constraint violations. Empirical results have supported the robustness and effectiveness of the proposed technique in solving the constrained parameter optimization problem of a breast cancer immunotherapy model. Furthermore, the optimized parameters obtained from the proposed technique generalizes the model for both the replication and prognostication of clinical therapeutic outcomes.

In Chapter 4, a data-augmentation pipeline has been formulated to solve the small-data issue in the constrained parameter optimization problem in Chapter 3. To study the effectiveness of the data-augmentation procedure, the pseudo data points generated by data-augmentation pipeline, and constraint-handling tech-

niques, three hypotheses are raised. These three hypotheses are supported by my empirical studies. First, the empirical results show that this data-augmentation procedure is effective in improving model resolution of various constraint-handling techniques in small-data scenarios. Second, the empirical results show that ε-SEC is the most effective constraint-handling technique for improving model resolution compared to the other four constraint-handling techniques post data augmentation. Third, the empirical results reveal that training the other four constraint-handling techniques using the pseudo training data set generated by ε-SEC improves their model resolution more than using their own respective pseudo training data set.

In Chapter 5, a DE-based algorithm NovDE has been proposed to solve locally D-optimal design problems with high-dimensional decision variables. The main issue of optimal design problems with high-dimensional decision variables is premature convergence. NovDE employs a new novelty-based mutation strategy to explore novelty regions of the search space so that the diversity of populations is preserved. Furthermore, the new novelty-based mutation strategy is used together with two other common mutation strategies which can balance exploration and exploitation at the early and medium stage of the evolution process so that both convergence and diversity of the population are enhanced, and premature convergence issues are alleviated. NovDE has successfully addressed the issue of solutions getting trapped into local optima. My empirical results show that the specially-designed mechanism, for exploring novelty regions in the decision space, in NovDE can effectively assist in escaping local optima with higher chance. NovDE achieves effective and robust performance when solving problems having more than 500 decision variables in D-optimal designs with demanding requirements.

6.2 Future Work

Generally, this thesis has provided an overview of solving optimization problems having decision variables with complex characteristics by EC techniques. Below are some additional future works can be explored.

The breast cancer immunotherapy model is formulated based on the assumption of intra-tumor homogeneity. However, actual observations may diverge from model predictions, thereby intra-tumor heterogeneity may be concluded and treatment strategy may need to be switched. Since intra-tumor heterogeneity is an increasingly accepted concept in oncology [114], a new model based on the assumption of intra-tumor heterogeneity can be established and the proposed constraint-handling technique can be used to optimize parameters in this new model. Furthermore, all of the data points are collected off-line, and the model parameters of a dynamic system are optimized based on off-line data. However, while the optimized parameter values for a dynamic system are optimal compared to all other fixed parameter values for this dynamic system, they may not be optimal compared to dynamic adaptively tuning of this dynamic system. For example, a current challenge in all forms of drug administration is that drug synergy is time-dependent, dose-dependent and patient-specific at any given point of treatment [115]; i.e. the dose changes dynamically and adaptively. In the future, if on-line data can be collected, both on-line data and off-line data can be used to incrementally and adaptively optimize model parameters to improve the effectiveness and robustness of the model. Currently, the proposed constraint-handling technique ε-SEC is applied to solve the constrained parameter optimization problem of a cancer immunotherapy model. ε-SEC can also be extended to solve other constrained optimization problems in real applications such as knapsack problems and traveling salesman problems.

In the proposed data-augmentation pipeline, there is one round of generating

pseudo data points in order to optimize model parameters based on both real and pseudo data points. Possibly, this pipeline can be conducted in multiple rounds such that a new set of pseudo data points are generated based on the new model obtained in the last round, and this new set of pseudo data points and real data are used to optimize model parameters. It seems worthwhile to study whether conducting the pipeline in multiple rounds can increase model resolution while successfully mitigating against over-tuning. Furthermore, this proposed data-augmentation pipeline can also be applied to other constrained optimization problems in real applications that have small-data issues.

In the field of optimal design, only locally D-optimal design problems are studied in this thesis. However, there are other optimal design criteria such as A-optimal design and G-optimal design. In real cases, it is more general to generate a design satisfying multiple design criteria instead of D-optimal design alone, so the problem is converted to multi-objective optimization. Stringently requiring decision variables to satisfy multiple criteria can be a big challenge to be solved in the future. Some state-of-the-art multi-objective evolutionary algorithms can be studied and modified to solve the multi-criteria optimal design problems more effectively. If the optimal design problems are both multi-objective and high-dimensional, it is far more challenging to be handled. In addition, the proposed algorithm NovDE can also be applied to solve other high-dimensional and non-separable problems in real applications such as traveling salesman problem or feature selection problem.

List of Publications

My papers that were published, accepted, and submitted during the course of my PhD study are listed as follows.

Journals

1. **W. Xu**, J. Xu, D. He, and K. C. Tan, "An Evolutionary Constraint-Handling Technique for Parametric Optimization of a Cancer Immunotherapy Model", *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 2, pages 151-162, April 2019.
2. **W. Xu**, W. K. Wong, K. C. Tan, and J. Xu, "Finding High-Dimensional D-Optimal Designs for Logistic Models via Differential Evolution", *IEEE Access*, vol. 7, no. 1, pages 7133-7146, December 2019 (Early Access).

Conferences

1. **W. Xu**, J. Xu, D. He, and K. C. Tan, "A combined differential evolution and NSGA-II approach for parametric optimization of a cancer immunotherapy model", *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Honolulu, USA, Nov. 27-Dec. 1, 2017, pages 1-8, 2017.
2. P. Vadakkepat, T. C. Chong, W. A. Arokiasami, and **W. Xu**, , "Fuzzy Logic Controllers for navigation and control of AR. Drone using

Microsoft Kinect", *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Vancouver, Canada, July 24-29, 2016, pages 856-863, 2016.

3. X. Qiu, **W. Xu**, J. Xu and K. C. Tan, "A new framework for self-adapting control parameters in multi-objective optimization", *Genetic and Evolutionary Computation Conference 2015 (GECCO2015)*, Madrid, Spain, July 11-15, 2015, pages. 743-750, 2015.

The first publication under Journals and the first publication under Conferences contribute to the work in Chapter 3.

The second publication under Journals contribute to the work in Chapter 5.

Bibliography

[1] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[2] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Academic Press, 2014.

[3] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.

[4] S. Khuri, T. Back, and J. Heitkotter, "The zero/one multiple knapsack problem and genetic algorithms," in *SAC*, vol. 94, pp. 188–193, 1994.

[5] M. Mahi, O. K. Baykan, and H. Kodaz, "A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem," *Applied Soft Computing*, vol. 30, pp. 484–490, 2015.

[6] W. Xu, J. Xu, D. He, and K. C. Tan, "A combined differential evolution and nsga-ii approach for parametric optimization of a cancer immunotherapy model," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, IEEE, 2017.

[7] D. P. Bertsekas and A. Scientific, *Convex optimization algorithms*. Athena Scientific Belmont, 2015.

[8] S. Mahdavi, M. E. Shiri, and S. Rahnamayan, "Metaheuristics in large-scale global continues optimization: A survey," *Information Sciences*, vol. 295, pp. 407–428, 2015.

[9] N. Awad, M. Ali, J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization," *Tech. Rep.*, 2016.

[10] A. A. Freitas, "A review of evolutionary algorithms for data mining," in *Data Mining and Knowledge Discovery Handbook*, pp. 371–400, Springer, 2009.

[11] S. K. Pal, S. Bandyopadhyay, and S. S. Ray, "Evolutionary computation in bioinformatics: A review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 36, no. 5, pp. 601–615, 2006.

[12] L. G. de la Fraga and C. A. C. Coello, "A review of applications of evolutionary algorithms in pattern recognition," in *Pattern Recognition, Machine Intelligence and Biometrics*, pp. 3–28, Springer, 2011.

[13] A. E. Eiben and J. Smith, "From evolutionary computation to the evolution of things," *Nature*, vol. 521, no. 7553, p. 476, 2015.

[14] T. Back, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, 1996.

[15] C. Grosan and A. Abraham, "Hybrid evolutionary algorithms: methodologies, architectures, and reviews," in *Hybrid Evolutionary Algorithms*, pp. 1–17, Springer, 2007.

[16] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[17] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[18] M. Keshk, H. Singh, and H. Abbass, "Automatic estimation of differential evolution parameters using hidden markov models," *Evolutionary Intelligence*, pp. 1–17, 2018.

[19] O. Kramer, "Evolutionary self-adaptation: a survey of operators and strategy parameters," *Evolutionary Intelligence*, vol. 3, no. 2, pp. 51–65, 2010.

[20] N. Boukhari, F. Debbat, N. Monmarche, and M. Slimane, "A study on self-adaptation in the evolutionary strategy algorithm," in *Computational Intelligence and Its Applications: 6th IFIP TC 5 International Conference, CIIA 2018, Oran, Algeria, May 8-10, 2018, Proceedings 6*, pp. 150–160, Springer, 2018.

[21] D. Zaharie, "Control of population diversity and adaptation in differential evolution algorithms," in *Proc. of MENDEL*, vol. 9, pp. 41–46, 2003.

[22] H. A. Abbass, "The self-adaptive pareto differential evolution algorithm," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 1, pp. 831–836, IEEE, 2002.

[23] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.

[24] S.-Z. Zhao and P. N. Suganthan, "Empirical investigations into the exponential crossover of differential evolutions," *Swarm and Evolutionary Computation*, vol. 9, pp. 27–36, 2013.

[25] T. Takahama and S. Sakai, "Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pp. 1–9, IEEE, 2010.

[26] T. Bck and F. Hoffmeister, "Extended selection mechanisms in genetic algorithms," pp. 92–99, Morgan Kaufmann, 1991.

[27] J. A. Joines and C. R. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with ga's," in *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pp. 579–584, IEEE, 1994.

[28] J. C. Bean and A. ben Hadj-Alouane, *A dual genetic algorithm for bounded integer programs*. 1993.

[29] H. J. Barbosa and A. C. Lemonge, "An adaptive penalty scheme in genetic algorithms for constrained optimization problems," in *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, pp. 287–294, Morgan Kaufmann Publishers Inc., 2002.

[30] J. T. Richardson, M. R. Palmer, G. E. Liepins, and M. R. Hilliard, "Some guidelines for genetic algorithms with penalty functions," in *Proceedings*

of the 3rd international conference on genetic algorithms, pp. 191–197, Morgan Kaufmann Publishers Inc., 1989.

[31] H. Lu and G. G. Yen, "Rank-density-based multiobjective genetic algorithm and benchmark test function study," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 4, pp. 325–343, 2003.

[32] G. G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 253–274, 2003.

[33] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[34] P. Chootinan and A. Chen, "Constraint handling in genetic algorithms using a gradient-based repair method," *Computers & Operations Research*, vol. 33, no. 8, pp. 2263–2281, 2006.

[35] P. Koch, S. Bagheri, W. Konen, C. Foussette, P. Krause, and T. Back, "A new repair method for constrained optimization," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 273–280, ACM, 2015.

[36] B. Tessema and G. G. Yen, "An adaptive penalty formulation for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 39, no. 3, pp. 565–578, 2009.

[37] Y. Wang, B.-C. Wang, H.-X. Li, and G. G. Yen, "Incorporating objective function information into the feasibility rule for constrained evolutionary optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2938–2952, 2016.

[38] Y. Wang and Z. Cai, "Combining multiobjective optimization with differential evolution to solve constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 117–134, 2012.

[39] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *International Conference on Parallel Problem Solving from Nature*, pp. 249–257, Springer, 1994.

[40] Y.-F. Ge, W.-J. Yu, Y. Lin, Y.-J. Gong, Z.-H. Zhan, W.-N. Chen, and J. Zhang, "Distributed differential evolution based on adaptive mergence and split for large-scale optimization," *IEEE transactions on cybernetics*, vol. 48, no. 7, pp. 2166–2180, 2017.

[41] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008.

[42] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 378–393, 2013.

[43] A. LaTorre, S. Muelas, and J.-M. Pena, "A comprehensive comparison of large scale global optimizers," *Information Sciences*, vol. 316, pp. 517–549, 2015.

[44] J. Brest, A. Zamuda, I. Fister, and M. S. Maučec, "Large scale global optimization using self-adaptive differential evolution algorithm," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pp. 1–8, IEEE, 2010.

[45] J. Zhang and A. C. Sanderson, "Jade: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[46] A. W. Mohamed and A. S. Almazyad, "Differential evolution with novel mutation and adaptive crossover strategies for solving large scale global optimization problems," *Applied Computational Intelligence and Soft Computing*, vol. 2017, 2017.

[47] Y.-F. Ge, W.-J. Yu, Y. Lin, Y.-J. Gong, Z.-H. Zhan, W.-N. Chen, and J. Zhang, "Distributed differential evolution based on adaptive mergence and split for large-scale optimization," *IEEE Transactions on Cybernetics*, vol. 48, no. 7, pp. 2166–2180, 2018.

[48] K. Kozlov and A. Samsonov, "New migration scheme for parallel differential evolution," in *Proceedings of the international conference on bioinformatics of genome regulation and structure*, pp. 141–144, 2006.

[49] J. Apolloni, G. Leguizamon, J. Garc▮a-Nieto, and E. Alba, "Island based distributed differential evolution: an experimental study on hybrid testbeds," in *2008 Eighth International Conference on Hybrid Intelligent Systems*, pp. 696–701, IEEE, 2008.

[50] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, and E. Tarantino, "Biological invasion–inspired migration in distributed evolutionary algorithms," *Information Sciences*, vol. 207, pp. 50–65, 2012.

[51] H. Wang, Y. Jin, and J. O. Jansen, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 939–952, 2016.

[52] K. Giannakoglou, "Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence," *Progress in Aerospace Sciences*, vol. 38, no. 1, pp. 43–76, 2002.

[53] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE Computational Intelligence Magazine*, vol. 4, no. 3, 2009.

[54] J. O. Jansen, J. J. Morrison, H. Wang, R. Lawrenson, G. Egan, S. He, and M. K. Campbell, "Optimizing trauma system design: the geos (geospatial evaluation of systems of trauma care) approach," *Journal of Trauma and Acute Care Surgery*, vol. 76, no. 4, pp. 1035–1040, 2014.

[55] J. O. Jansen, J. J. Morrison, H. Wang, S. He, R. Lawrenson, J. D. Hutchison, and M. K. Campbell, "Access to specialist care: optimizing the geographic configuration of trauma systems," *The Journal of Trauma and Acute Care Surgery*, vol. 79, no. 5, p. 756, 2015.

[56] B. Stephen and J. Hajjar, "Overview of basic immunology for clinical investigators," in *Immunotherapy*, pp. 1–31, Springer, 2017.

[57] A. Ghochikyan, A. Davtyan, A. Hovakimyan, H. Davtyan, A. Poghosyan, A. Bagaev, R. I. Ataullakhanov, E. L. Nelson, and M. G. Agadjanyan, "Primary 4t1 tumor resection provides critical window of opportunity for immunotherapy," *Clinical & Experimental Metastasis*, vol. 31, no. 2, pp. 185–198, 2014.

[58] X. Li and J.-X. Xu, "A mathematical prognosis model for pancreatic cancer patients receiving immunotherapy," *Journal of Theoretical Biology*, vol. 406, pp. 42–51, 2016.

[59] L. G. de Pillis, K. R. Fister, W. Gu, T. Head, K. Maples, T. Neal, A. Murugan, and K. Kozai, "Optimal control of mixed immunotherapy and

chemotherapy of tumors," *Journal of Biological Systems*, vol. 16, no. 01, pp. 51–80, 2008.

[60] V. A. Kuznetsov, I. A. Makalkin, M. A. Taylor, and A. S. Perelson, "Nonlinear dynamics of immunogenic tumors: parameter estimation and global bifurcation analysis," *Bulletin of Mathematical Biology*, vol. 56, no. 2, pp. 295–321, 1994.

[61] K.-L. Liao, X.-F. Bai, and A. Friedman, "Mathematical modeling of interleukin-35 promoting tumor growth and angiogenesis," *PLoS One*, vol. 9, no. 10, p. e110126, 2014.

[62] S. Wilson and D. Levy, "Functional switching and stability of regulatory t cells," *Bulletin of Mathematical Biology*, vol. 75, no. 10, pp. 1891–1911, 2013.

[63] F. Gu and Y.-M. Cheung, "Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 211–225, 2018.

[64] H.-L. Liu, F.-q. Gu, and Y.-m. Cheung, "T-moea/d: Moea/d with objective transform in multi-objective problems," in *Information Science and Management Engineering (ISME), 2010 International Conference of*, vol. 2, pp. 282–285, IEEE, 2010.

[65] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[66] P. J. Fleming and R. C. Purshouse, "Evolutionary algorithms in control systems engineering: a survey," *Control Engineering Practice*, vol. 10, no. 11, pp. 1223–1241, 2002.

[67] D. Dasgupta and Z. Michalewicz, *Evolutionary algorithms in engineering applications*. Springer Science & Business Media, 2013.

[68] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin, "Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 2, pp. 97–111, 2017.

[69] C. Zhang, P. Lim, A. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Transactions on Neural Networks and Learning Systems*, 2017.

[70] C. Zhang, K. C. Tan, and R. Ren, "Training cost-sensitive deep belief networks on imbalance data problems," in *Neural Networks (IJCNN), 2016 International Joint Conference on*, pp. 4362–4367, IEEE, 2016.

[71] C. Zhang, K. C. Tan, H. Li, and G. S. Hong, "A cost-sensitive deep belief network for imbalanced classification," *Neural Networks and Learning Systems, IEEE Transactions on*, 2017.

[72] S. Z. Manrique, A. L. Dominguez, N. Mirza, C. D. Spencer, J. M. Bradley, J. H. Finke, J. J. Lee, L. R. Pease, S. J. Gendler, and P. A. Cohen, "Definitive activation of endogenous antitumor immunity by repetitive cycles of cyclophosphamide with interspersed toll-like receptor agonists," *Oncotarget*, vol. 7, no. 28, p. 42919, 2016.

[73] B. Kyewski and E. Suri-Payer, *CD4+ CD25+ regulatory T cells: Origin, function and therapeutic potential*, vol. 293. Springer Science & Business Media, 2005.

[74] S. Sakaguchi, T. Yamaguchi, T. Nomura, and M. Ono, "Regulatory t cells and immune tolerance," *Cell*, vol. 133, no. 5, pp. 775–787, 2008.

[75] M. Beyer, M. Kochanek, K. Darabi, A. Popov, M. Jensen, E. Endl, P. A. Knolle, R. K. Thomas, M. von Bergwelt-Baildon, S. Debey, *et al.*, "Reduced frequencies and suppressive function of cd4+ cd25 hi regulatory t cells in patients with chronic lymphocytic leukemia after therapy with fludarabine," *Blood*, vol. 106, no. 6, pp. 2018–2025, 2005.

[76] F. Ghiringhelli, N. Larmonier, E. Schmitt, A. Parcellier, D. Cathelin, C. Garrido, B. Chauffert, E. Solary, B. Bonnotte, and F. Martin, "Cd4+ cd25+ regulatory t cells suppress tumor immunity but are sensitive to cyclophosphamide which allows immunotherapy of established tumors to be curative," *European Journal of Immunology*, vol. 34, no. 2, pp. 336–344, 2004.

[77] Y. Ikezawa, M. Nakazawa, C. Tamura, K. Takahashi, M. Minami, and Z. Ikezawa, "Cyclophosphamide decreases the number, percentage and the function of cd25+ cd4+ regulatory t cells, which suppress induction of contact hypersensitivity," *Journal of Dermatological Science*, vol. 39, no. 2, pp. 105–112, 2005.

[78] M. C. Lutsiak, R. T. Semnani, R. De Pascalis, S. V. Kashmiri, J. Schlom, and H. Sabzevari, "Inhibition of cd4+ 25+ t regulatory cell function implicated in enhanced immune response by low-dose cyclophosphamide," *Blood*, vol. 105, no. 7, pp. 2862–2868, 2005.

[79] S. Venkatraman and G. G. Yen, "A generic framework for constrained optimization using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 4, pp. 424–435, 2005.

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

[81] R. Kitchin and T. P. Lauriault, "Small data in the era of big data," *GeoJournal*, vol. 80, no. 4, pp. 463–475, 2015.

[82] J. Wang and L. Perez, "The effectiveness of data augmentation in image classification using deep learning," *Convolutional Neural Networks Vis. Recognit*, 2017.

[83] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[84] H. Chernoff, "Locally optimal designs for estimating parameters," *The Annals of Mathematical Statistics*, pp. 586–602, 1953.

[85] S. Silvey, *Optimal design: an introduction to the theory for parameter estimation*, vol. 1. Springer Science & Business Media, 2013.

[86] J. King and W.-K. Wong, "Minimax d-optimal designs for the logistic model," *Biometrics*, vol. 56, no. 4, pp. 1263–1267, 2000.

[87] J. Qiu, R.-B. Chen, W. Wang, and W. K. Wong, "Using animal instincts to design efficient biomedical studies via particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 18, pp. 1–10, 2014.

[88] J. M. Whitacre, "Recent trends indicate rapid growth of nature-inspired optimization in academia and industry," *Computing*, vol. 93, no. 2-4, pp. 121–133, 2011.

[89] J. M. Whitacre, "Survival of the flexible: explaining the recent popularity of nature-inspired optimization within a rapidly evolving world," *Computing*, vol. 93, no. 2, pp. 135–146, 2011.

[90] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, pp. 760–766, Springer, 2011.

[91] M. Clerc, *Particle swarm optimization*, vol. 93. John Wiley & Sons, 2010.

[92] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[93] W. K. Wong, R.-B. Chen, C.-C. Huang, and W. Wang, "A modified particle swarm optimization technique for finding optimal designs for mixture models," *PloS One*, vol. 10, no. 6, p. e0124720, 2015.

[94] R.-B. Chen, S.-P. Chang, W. Wang, H.-C. Tung, and W. K. Wong, "Minimax optimal designs via particle swarm optimization methods," *Statistics and Computing*, vol. 25, no. 5, pp. 975–988, 2015.

[95] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *ICML*, vol. 97, pp. 412–420, 1997.

[96] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H. S.-H. Chung, Y. Li, and Y.-H. Shi, "Particle swarm optimization with an aging leader and challengers," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 241–258, 2013.

[97] A. Qing, "Dynamic differential evolution strategy and applications in electromagnetic inverse scattering problems," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 1, pp. 116–125, 2006.

[98] A. Talukder, M. Kirley, and R. Buyya, "Multiobjective differential evolution for scheduling workflow applications on global grids," *Concurrency and Computation: Practice and Experience*, vol. 21, no. 13, pp. 1742–1756, 2009.

[99] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[100] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 99–119, 2011.

[101] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 482–500, 2012.

[102] S. Risi, S. D. Vanderbleek, C. E. Hughes, and K. O. Stanley, "How novelty search escapes the deceptive trap of learning to learn," in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, pp. 153–160, ACM, 2009.

[103] J. Lehman and K. O. Stanley, "Evolving a diversity of virtual creatures through novelty search and local competition," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pp. 211–218, ACM, 2011.

[104] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 2, pp. 1785–1791, IEEE, 2005.

[105] C. E. McCulloch, "Generalized linear models," *Journal of the American Statistical Association*, vol. 95, no. 452, pp. 1320–1324, 2000.

[106] C. Whitman, T. M. Gilbert, A. M. Rahn, and J. A. Antonell, "Determining factors affecting esd failure voltage using doe," *Microelectronics Reliability*, vol. 46, no. 8, pp. 1228–1237, 2006.

[107] I. Ford, B. Torsney, and C. J. Wu, "The use of a canonical form in the construction of locally optimal designs for non-linear problems," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 569–583, 1992.

[108] J. Stufken and M. Yang, "Optimal designs for generalized linear models," *Design and Analysis of Experiments, Special Designs and Applications*, vol. 3, p. 137, 2012.

[109] J. Kiefer and J. Wolfowitz, "Optimum designs in regression problems," *The Annals of Mathematical Statistics*, pp. 271–294, 1959.

[110] A. Pazman, *Foundations of optimum experimental design*, vol. 14. Springer, 1986.

[111] X. Qiu, K. C. Tan, and J.-X. Xu, "Multiple exponential recombination for differential evolution," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 995–1006, 2017.

[112] S. D. Grimshaw, B. J. Collings, W. A. Larsen, and C. R. Hurt, "Eliciting factor importance in a designed experiment," *Technometrics*, vol. 43, no. 2, pp. 133–146, 2001.

[113] J. Lukemire, A. Mandal, and W. K. Wong, "d-qpso: A quantum-behaved particle swarm technique for finding d-optimal designs with discrete and continuous factors and a binary response," *Technometrics*, no. just-accepted, pp. 1–27, 2018.

[114] A. Marusyk, V. Almendro, and K. Polyak, “Intra-tumour heterogeneity: a looking glass for cancer?,” *Nature Reviews Cancer*, vol. 12, no. 5, p. 323, 2012.

[115] D. Ho, P. Wang, and T. Kee, “Artificial intelligence in nanomedicine,” *Nanoscale Horizons*, vol. 4, no. 2, pp. 365–377, 2019.