National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Parallel CFD & Optimization Unit

Quantum-Inspired Evolutionary Algorithms Assisted by Metamodels

Diploma Thesis

Ioannis A. Gerogiannis

Advisors: Kyriakos C. Giannakoglou, Professor NTUA,
Dr. V. Asouti, Adjunct Lecturer NTUA

Athens, February 2026

Acknowledgments

First, I would like to thank my supervisor, Professor Kyriakos Giannakoglou, both for giving me the opportunity to work on an interesting topic and for his invaluable support, not only during the preparation of my diploma thesis but also throughout my undergraduate studies. He has been a decisive influence on the way I think as an engineer. Through his lectures and his remarkable ability to communicate complex concepts with clarity and intuition, I developed a strong foundation in topics related to fluid mechanics, turbomachinery, CFD, and optimization. Beyond knowledge communication, he also has the ability to impart mindset, critical thinking abilities, and problem-solving skills, all of which are essential qualities for an engineer.

Furthermore, I wish to express my gratitude to Dr. and Lecturer Varvara Asouti for her invaluable support throughout the preparation of this diploma thesis and for the many things I learned from her during my undergraduate studies.

Above all, I wish to express my heartfelt gratitude to my family, my friends, and everyone who has supported me and stood by my side throughout this journey.

I owe a special debt of gratitude to my mother, Despoina, who, through her immense effort, countless sacrifices, and unwavering strength in the face of adversity, ensured that my sister and I would always have what we needed and the opportunity to study at the school of our first choice, an opportunity I have never taken for granted. Her life and boundless inner strength have been a constant source of inspiration to me, as she has never wavered from her purpose, regardless of the difficulties she encountered.

Finally, I would like to express my deepest gratitude to my late father, Apostolis. He was the most influential figure in my life and played a decisive role in shaping my values and aspirations. Also, it was through him that I first came to know the profession of mechanical engineering, which I ultimately chose to pursue.

National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Parallel CFD & Optimization Unit

Quantum-Inspired Evolutionary Algorithms Assisted by Metamodels

Diploma Thesis

Ioannis A. Gerogiannis

Advisors: Kyriakos C. Giannakoglou, Professor NTUA,
Dr. V. Asouti, Adjunct Lecturer NTUA

Athens, February 2026

Abstract

The subject of this thesis is the Quantum-Inspired Evolutionary Algorithm (QIEA), a metaheuristic optimization algorithm that incorporates principles of quantum mechanics and quantum computing into the classical evolutionary algorithm. The aim is to investigate the performance of this alternative version of the evolutionary algorithm in numerical optimization problems and to assess the extent to which the use of this different approach can contribute to and benefit the advancement of the field of optimization.

It is worth noting that QIEA is not a quantum algorithm in the sense that it runs on a quantum computer. It is an algorithm designed to run on a conventional digital computer, which draws on ideas from quantum mechanics to formulate an algorithm in the same way that biological evolution inspired the conception of genetic optimization algorithms.

Initially, an introduction is given to evolutionary algorithms as well as to the fundamental principles of quantum computing, such as quantum bits (qubits), superposition, the Bloch sphere, and quantum gates. These notions constitute the foundation upon which the QIEA algorithm will be formulated.

Next, the algorithmic framework of QIEA is presented. In contrast to classical evolutionary algorithms, where the population of individuals represents specific candidate solutions, in QIEA each individual models a probability distribution of the optimal solution in the search space, through a sequence of quantum binary digits (qubits). The observation of this quantum state, or sampling from the probability distribution, produces specific actual solutions. The probability amplitudes of the qubits are adjusted over generations using evolutionary operators that simulate quantum gates, in order to lead to more promising solutions.

With regard to Multi-Objective Optimization with QIEA (Multi-Objective QIEA - MO-QIEA), a challenge arises: comparison via Pareto dominance, which is crucial for the operation of the QIEA, cannot be applied to non-dominated solutions produced by a multi-objective function, since these solutions are incomparable. To address this difficulty, a new variant of QIEA is introduced. In this variant, the uncertainty about whether an individual has improved or deteriorated leads to its duplication. Thus, one copy evolves as if an improvement had occurred and the other as if a deterioration had occurred. Subsequently, the NSGA-II algorithm is applied

to rank the overall population in each generation. In this way, the best solutions are preserved, the population size is controlled, and at the same time the diversity of solutions is ensured.

The work also investigates the extent to which the use of metamodels in QIEA (MAQIEA) can reduce the number of calls to the exact and computationally expensive evaluation function and consequently improve convergence speed. Two types of metamodels are examined: standard Radial Basis Function networks (simple RBF networks) and adaptive Radial Basis Function networks that use self-organizing maps (Growing SOM RBF networks).

The performance of the algorithm is evaluated using standard benchmark functions, both for single-objective and multi-objective optimization problems. These results are compared with those obtained from the classical evolutionary algorithm, which serves as a reference measure for assessing the performance of the QIEA.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης

Εξελικτικοί Αλγόριθμοι Εμπνευσμένοι από την Κβαντομηχανική, Υποστηριζόμενοι από Μεταπρότυπα

Διπλωματική Εργασία

Ιωάννης Α. Γερογιάννης

Επιβλέποντες: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ,
Δρ. Β. Ασούτη, Εντεταλμένη Διδάσκουσα ΕΜΠ

Αθήνα, Φεβρουάριος 2026

Abstract

Αντικείμενο της διπλωματικής εργασίας είναι ο "Εμπνευσμένος από την Κβαντομηχανική Εξελικτικός Αλγόριθμος" (Quantum Inspired Evolutionary Algorithm - QIEA), ένας μετευρετικός αλγόριθμος βελτιστοποίησης που ενσωματώνει αρχές της κβαντομηχανικής και κβαντικής υπολογιστικής στον κλασικό εξελικτικό αλγόριθμο. Σκοπός είναι η διερεύνηση της απόδοσης της συγκεκριμένης εκδοχής του αλγορίθμου σε αριθμητικά προβλήματα βελτιστοποίησης και η διασαφήνιση του κατα πόσο αυτή η εκδοχή μπορεί να συνεισφέρει και να ωφελήσει στην εξέλιξη του πεδίου της βελτιστοποίησης.

Αξίζει να σημειωθεί ότι ο QIEA δεν είναι ένας κβαντικός αλγόριθμος υπό την έννοια ότι εκτελείται σε κβαντικό υπολογιστή. Είναι αλγόριθμος σχεδιασμένος ώστε να τρέχει σε συμβατικό δυαδικό υπολογιστή, ο οποίος αξιοποιεί ιδέες από την κβαντομηχανική για τη σύσταση ενός αλγορίθμου με τον ίδιο τρόπο που η βιολογική εξέλιξη ενέπνευσε την σύλληψη των γενετικών αλγορίθμων βελτιστοποίησης.

Αρχικά γίνεται μία εισαγωγή στους εξελικτικούς αλγορίθμους καθώς και τις θεμελιώδεις αρχές κβαντικής υπολογιστικής, όπως το κβαντικό bit (qubit), η υπέρθεση, η σφαίρα Bloch και οι κβαντικές πύλες. Αυτές οι έννοιες αποτελούν το υπόβαθρο πάνω στο οποίο θα δομηθεί ο αλγόριθμος QIEA.

Στη συνέχεια, παρουσιάζεται το αλγοριθμικό πλαίσιο του Κβαντικού Εξελικτικού Αλγορίθμου (Quantum-inspired Evolutionary Algorithm - QIEA). Σε αντίθεση με τους κλασικούς εξελικτικούς αλγορίθμους, όπου ο πληθυσμός των ατόμων αναπαριστά συγκεκριμένες υποψήφιες λύσεις, στον QIEA κάθε άτομο μοντελοποιεί μια κατανομή πιθανοτήτων της βέλτιστης λύσης στον χώρο αναζήτησης, μέσω μιας ακολουθίας κβαντικών δυαδικών ψηφίων (qubits). Η παρατήρηση αυτής της κβαντικής κατάστασης, ή η δειγματοληψία από την πιθανοτική κατανομή, παράγει συγκεκριμένες πραγματικές λύσεις. Τα πλάτη πιθανότητας των qubits προσαρμόζονται με την πάροδο των γενεών, χρησιμοποιώντας εξελικτικούς τελεστές που προσομοιώνουν κβαντικές πύλες, ώστε να οδηγήσουν σε πιο υποσχόμενες λύσεις.

Όσον αφορά την Πολυκριτηριακή Βελτιστοποίηση με QIEA (Multi-Objective QIEA - MO-QIEA), ανακύπτει μια πρόκληση: η σύγκριση μέσω της κυριαρχίας κατά Pareto, η οποία είναι κρίσιμη

για τη λειτουργία του QIEA, δεν μπορεί να εφαρμοστεί σε μη-κυριαρχούμενες λύσεις που προκύπτουν από μια αντικειμενική συνάρτηση πολλαπλών στόχων, καθώς αυτές είναι μη συγκρίσιμες. Για την αντιμετώπιση αυτής της δυσκολίας, εισάγεται μια νέα παραλλαγή του QIEA. Σε αυτήν, η αβεβαιότητα για το αν ένα άτομο βελτιώθηκε ή επιδεινώθηκε οδηγεί στον διπλασιασμό του. Έτσι, το ένα αντίγραφο εξελίσσεται ως να υπήρξε βελτίωση και το άλλο σαν να υπήρξε επιδείνωση. Ακολούθως, εφαρμόζεται ο αλγόριθμος NSGA-II για την ταξινόμηση του συνολικού πληθυσμού σε κάθε γενιά. Με αυτόν τον τρόπο, διατηρούνται οι βέλτιστες λύσεις, περιορίζοντας το πλήθος των ατόμων και διασφαλίζοντας ταυτόχρονα την ποικιλομορφία των λύσεων.

Η εργασία εξετάζει επίσης το κατά πόσο η χρήση μεταμοντέλων στον QIEA (Metamodel Assisted QIEA - MAQIEA) μπορεί να μειώσει τον αριθμό των κλήσεων της ακριβούς και υπολογιστικά ακριβής συνάρτησης αξιολόγησης και συνεπώς να βελτιώσει την ταχύτητα σύγκλισης. Χρησιμοποιούνται δύο τύποι μεταμοντέλων: απλά δίκτυα ακτινικής βάσης συναρτήσεων (RBF networks) και αυξανόμενα δίκτυα ακτινικής βάσης συναρτήσεων με αυτο-οργανούμενους χάρτες (growing SOM RBF networks).

Η αξιολόγηση της απόδοσης του αλγορίθμου πραγματοποιείται με τη χρήση τυπικών δοκιμαστικών συναρτήσεων, τόσο για προβλήματα βελτιστοποίησης ενός στόχου όσο και πολλαπλών στόχων. Τα αποτελέσματα αυτά συγκρίνονται με εκείνα που προκύπτουν από τον κλασικό εξελικτικό αλγόριθμο, ο οποίος λειτουργεί ως μέτρο αναφοράς για την εκτίμηση της αποτελεσματικότητας του QIEA.

Contents

1 Introduction **1**
- 1.1 Motivation for Quantum Computing . . . 1
- 1.2 Obstacles and Strategies for Addressing Them . . . 2
  - 1.2.1 The NISQ Era . . . 2
  - 1.2.2 Error Correction and Fault Tolerance . . . 3
  - 1.2.3 Addressing the Nonlinearity Challenge . . . 3
  - 1.2.4 Hybrid Quantum-Classical Approaches . . . 3
- 1.3 Quantum-Inspired Evolutionary Algorithms . . . 4
- 1.4 Thesis Outline . . . 4

2 Quantum Mechanics &Quantum Computing Fundamentals **6**
- 2.1 Basic Unit of Information - Qubit . . . 6
  - 2.1.1 Physical Implementations of Qubits . . . 6
- 2.2 Mathematical Representation and Bra-Ket Notation . . . 7
  - 2.2.1 Kets . . . 7
  - 2.2.2 Bras . . . 7
  - 2.2.3 Inner Product (Bra-ket) . . . 7
  - 2.2.4 Outer Product (Ket-Bra) . . . 8
  - 2.2.5 Orthonormal Basis and Computational Basis . . . 8
- 2.3 Quantum State &Superposition . . . 9
- 2.4 Measurements and Probabilities . . . 9
  - 2.4.1 Born's Rule and Quantum Measurement . . . 9
  - 2.4.2 Measurement Process and Fundamental Limitations . . . 10
- 2.5 Geometric Representation: The Bloch Sphere . . . 10
  - 2.5.1 From Complex Amplitudes to Cartesian Coordinates . . . 10
  - 2.5.2 Converting to Spherical Coordinates . . . 11
  - 2.5.3 The Bloch Sphere Parametrization . . . 12
- 2.6 Multiple Qubits and Tensor Product . . . 13
  - 2.6.1 The Tensor Product Operation . . . 13
  - 2.6.2 Separable and Entangled States . . . 14
  - 2.6.3 Classical vs. Quantum Information Storage . . . 15
- 2.7 Gates &Operations . . . 15
  - 2.7.1 Basic Single-qubit Gates . . . 16
  - 2.7.2 Two-Qubit Gates . . . 17

3 Evolutionary Algorithms **19**
- 3.1 Description of an Evolutionary Algorithm . . . 19
  - 3.1.1 Evolution Operators . . . 20
    - Parent Selection Operator . . . 20

Crossover Operator . . . 20
Mutation Operator . . . 21
Elitism Operator . . . 21
3.1.2 Algorithmic Procedure . . . 21
3.2 Evolutionary Algorithms in Multi-Objective Optimization . . . 22
3.2.1 Pareto Dominance and Pareto Front . . . 22

4 QIEA **23**
4.1 Solution Encoding and Decoding . . . 23
4.1.1 Binary Encoding Scheme . . . 23
4.1.2 Multi-Variable Encoding . . . 24
4.1.3 Implementation Considerations . . . 24
4.2 Solution Representation . . . 25
4.2.1 Quantum Representation of Individual . . . 25
4.2.2 Angular Parameterization and Bloch Sphere . . . 25
4.2.3 Measurement in QIEA . . . 27
4.2.4 Visualization of Quantum Individual . . . 27
4.3 Implementation of Single-Objective QIEA . . . 28
4.3.1 Measurement &Evaluation . . . 28
4.3.2 Initialization . . . 30
4.3.3 Q-Gate Rotation . . . 31
4.3.4 Quantum Crossover . . . 32
4.3.5 Quantum Mutation . . . 33
4.3.6 Migration . . . 33
4.3.7 Elites Selection . . . 34
4.3.8 Elitism . . . 34
4.3.9 Parent Selection . . . 34
4.3.10 Stagnation Mechanism . . . 35
4.3.11 QIEA Workflow . . . 35
4.4 Multi-Objective QIEA . . . 36
4.4.1 Challenges in Quantum Gate Application . . . 36
4.4.2 NSGA-II Selection Algorithm . . . 37
4.4.3 Complete Algorithm Description . . . 38

5 Metamodel Assisted QIEA - MAQIEA **40**
5.1 Surrogate Models in Evolutionary Optimization . . . 40
5.1.1 Local versus Global Metamodeling . . . 40
5.1.2 Low-Cost Pre-Evaluation Technique . . . 41
5.1.3 Challenges in Multi-Objective Optimization . . . 41
5.2 Metamodel-Assisted QIEA . . . 42
5.3 Metamodel Implementation Details . . . 42
5.3.1 Local Pattern Selection Using Minimum Spanning Trees . . . 42
5.3.2 Outlier Detection and Reliability Assessment . . . 43
5.3.3 Radial-Basis Function Networks . . . 44
Network Architecture . . . 44
Training Procedure . . . 45
5.3.4 Growing RBF Networks with Self-Organizing Maps . . . 46
Algorithm Overview . . . 46
Self-Organizing Map for Center Placement . . . 46

RBF Width Determination via MST 48
Center Splitting Strategy 48

6 Benchmark Cases **49**
6.1 Benchmark for Single-Objective Optimization 49
6.1.1 Grid Search and Convergence Analysis of Optimal Configurations . 52
6.1.2 Parametric Analysis 56
6.1.3 QIEA - EA Comparison 62
6.2 Benchmark Cases for Multi-Objective Optimization 65

7 Conclusions **70**

Chapter 1

Introduction

Quantum computing has attracted considerable attention from the scientific community in recent years. The aim of this chapter is to provide a brief, high-level overview of quantum computing and its core principles. In addition, the chapter reviews the progress achieved to date, describes some of the domains in which it could be applied, and discusses the developments anticipated in the near future.

Quantum computing is a different approach to computation. It relies on the principles of quantum mechanics and leverages quantum effects such as superposition and entanglement to encode and process information in a way that differs from traditional binary computers. The distinct principles on which quantum computers are based may be able to solve specific problems that challenge classical computers.

The field, however, is still in its early stages. There are several limitations and obstacles that must be overcome before quantum computers can become truly useful for practical applications [1]. These limitations arise both from hardware restrictions and from constraints imposed by the quantum nature of these systems, which makes it challenging to use them for problems that are not inherently governed by quantum-mechanical principles.

1.1 Motivation for Quantum Computing

Classical computers have driven tremendous progress in science and engineering over the past half-century. High-performance computing enables simulation, optimization, and analysis across countless domains. However, they face difficulty handling problems where the information required to represent the system state or solution space grows exponentially with problem size, quickly exceeding available memory and computational capacity. Such cases include quantum mechanical simulation, where representing n particles requires 2^n complex amplitudes Another example that pushes the limits of traditional computers is direct numerical simulation (DNS), which involves solving the three-dimensional, time-dependent Navier–Stokes equations to determine the instantaneous fluid velocity at every point in space and time. For turbulent flows at high Reynolds numbers, DNS would require grids with spatial resolutions fine enough to resolve almost every individual eddy. The resulting demands on memory and computational power exceed the capabilities of a conventional binary computer.

Representing information in a quantum computer using qubits gives us the ability to store and process a volume of data that would be practically impossible on a classical computer. Indicatively, the largest quantum computer that exists today (IBM Condor)

has 1121 qubits. The information that this system can encode corresponds to more than $2 \cdot 10^{317} ZB$ if we attempted to represent it on a classical computer. For comparison, the total amount of digital information on the internet by the end of 2025 is estimated to be around $180 ZB$.

This different way of representing information could, under certain conditions, give us the ability to solve problems that we are not able to solve with today's computers.

1.2 Obstacles and Strategies for Addressing Them

While quantum computing holds substantial promise for addressing computationally demanding problems, several fundamental technical barriers must be overcome before these systems can deliver practical advantages for real-world applications. These obstacles arise from two principal sources: the physical limitations of current quantum hardware, and the inherent mismatch between quantum computational properties and problems not governed by quantum-mechanical principles.

Regarding the types of problems for which the quantum computer offers advantages, it is clear that it is particularly distinguished in solving and simulating quantum mechanical systems, or systems governed by quantum mechanical properties, such as, for example, the simulation of molecules. Quantum mechanical systems are inherently linear because they are governed by the (linear) Schrödinger equation. This intrinsic linearity of quantum computing poses challenges when attempting to use it to tackle nonlinear problems, like CFD applications [2] [3].

Regarding the hardware, contemporary quantum processors suffer from noise, limited qubit coherence times, and imperfect gate operations that introduce errors during computation. These physical imperfections constrain both the number of sequential operations (circuit depth) that can be reliably executed and the scale of problems that can be addressed without error correction.

1.2.1 The NISQ Era

Modern quantum computing is currently in what researchers describe as the NISQ era (Noisy Intermediate-Scale Quantum). This term reflects the present technological landscape, quantum processors equipped with a modest to a few thousand qubits (typically from about 50 to over 1,000), yet still subject to substantial noise and error rates that constrain circuit depth and overall reliability. Quantum systems are extremely sensitive to environmental noise, like surrounding particles, electromagnetic fields, and thermal fluctuations. This environmental interaction causes decoherence, leading to loss of quantum information and computational errors. Each quantum operation introduces errors that accumulate rapidly across many sequential gates. Current superconducting quantum systems achieve gate fidelities above 99.9% for single-qubit operations and above 99% for two-qubit gates, with near-term targets pushing toward four and three nines of fidelity respectively [4]. While these percentages appear high, errors accumulate rapidly across hundreds of operations, and the resulting noise limits current quantum computers to relatively shallow circuits before accumulated errors compromise computational results. Additionally, limited qubit connectivity, where each qubit can only directly interact with its nearest neighbors, requires extra operations to move information across the chip, introducing further errors.

1.2.2 Error Correction and Fault Tolerance

Quantum error correction provides a way to safeguard quantum systems from noise that arises due to interactions with the environment or from imperfections in the control of the system. Rather than trying to create flawless qubits, quantum error correction bundles multiple physical qubits, forming a so-called logical qubit [5], in such a way that the information is distributed redundantly. The approach resembles redundancy methods in classical systems, where the same data is stored in several locations so that errors can be identified and corrected. However, quantum error correction is more challenging because quantum information cannot be directly duplicated according to the no-cloning theorem, and observing qubits collapses their quantum states. Specialized techniques such as the surface code have been developed to work around these constraints, and recent experiments have demonstrated that these approaches can successfully protect quantum information.

The resource cost of error correction is substantial. Current estimates suggest that a single reliable logical qubit may require hundreds or even thousands of physical qubits working together, depending on how error-prone the physical qubits are. This means that quantum algorithms requiring thousands of logical qubits to solve practical problems, such as simulating complex molecules, might need millions of physical qubits in total. The long-term goal is fault-tolerant quantum computing is systems with enough error-corrected qubits to perform long, complex calculations reliably [6]. While the basic principles have been validated in laboratories, building practical fault-tolerant quantum computers capable of outperforming classical computers on real-world problems remains a major engineering challenge that likely requires years of additional development.

1.2.3 Addressing the Nonlinearity Challenge

Beyond hardware limitations, the linear nature of quantum computation itself poses challenges for nonlinear problems. As discussed earlier, the inherent quantum properties of quantum computers allow them to efficiently simulate quantum systems. For problems governed by nonlinear equations, unlike quantum systems, it is difficult to simulate them using quantum computers. One approach that has attracted research interest is Carleman linearization [7], which transforms nonlinear differential equations into infinite-dimensional linear systems that can be truncated and solved using quantum linear systems algorithms. The efficiency of this approach depends critically on the degree of linearity. For strongly nonlinear systems, such as turbulent flows at high Reynolds numbers, the case where DNS becomes computationally prohibitive, the method fails to produce reliable results.

1.2.4 Hybrid Quantum-Classical Approaches

Considering the constraints of current hardware and the inherent difficulties associated with the principles that govern quantum systems, another option is to employ a hybrid approach that combines quantum and classical computing. Rather than attempting to run entire algorithms on quantum hardware, hybrid methods partition computational tasks between classical and quantum processors. Classical computers manage problem setup, optimization loops, and post-processing, while quantum processors handle specific subtasks where quantum effects might provide advantage.

The Variational Quantum Eigensolver (VQE) exemplifies this hybrid approach. It

was originally developed to find the ground state energy of molecules. The quantum processor prepares a candidate quantum state and measures its energy. A classical optimizer then adjusts the parameters to minimize the energy through repeated iterations. This hybrid approach does not treat today's quantum computers as independent problem-solving devices, but instead as specialized co-processors that can speed up specific types of calculations when integrated with traditional hardware.

Hybrid quantum–classical approaches, including the Variational Quantum Eigensolver (VQE) and the Variational Quantum Linear Solver (VQLS), have been explored for use in CFD. They have been employed to solve the Poisson pressure equation arising from the incompressible Navier–Stokes equations, using Chorin's pressure-based projection method to decouple velocity and pressure calculations [2]. However, as with other quantum approaches for CFD, these methods require heavy mathematical simplifications and remain tailored to specific problem formulations.

1.3 Quantum-Inspired Evolutionary Algorithms

Another strategy for leveraging quantum computing in the NISQ era involves quantum-inspired algorithms (QIAs), which emulate key concepts of quantum computation while operating on classical hardware [8]. These approaches aim to exploit the core principles of quantum computing to develop new classical algorithms that can share and leverage some of the same beneficial features.

Quantum-Inspired Evolutionary Algorithms (QIEAs) provide a representative example of this approach [9] [10] [11] [12]. They are optimization methods that incorporate concepts from quantum computing into evolutionary algorithms that run on classical computers.

The key feature of QIEAs is their probabilistic representation. Traditional genetic algorithms encode candidate solutions as definite binary strings or real-valued vectors.In contrast, QIEAs use representations that imitate quantum bits (Q-bits), thereby capturing the probabilistic characteristics of quantum mechanics. This probabilistic encoding enables an individual to encode a linear superposition of all possible states, which may in turn facilitate more efficient exploration and exploitation of the search space.

QIEAs has been applied to various combinatorial optimization problems including the knapsack problem, the travelling salesman problem, the N-Queens problem, and job shop scheduling, demonstrating promising results in these domains.

Adapting QIEAs to continuous optimization problems has proven more challenging, as there is no obvious way to represent a superposition of real values. Early attempts either discretized the continuous domain through binary encoding or abandoned the superposition property by extracting real solutions almost deterministically from Q-bit amplitudes [13]. A recursive deepening strategy has been proposed in the literature, according to which superposition is maintained while progressively narrowing the search space, combined with local search methods to improve solution quality [14].

1.4 Thesis Outline

In the following chapters, the theoretical background and the algorithmic framework of QIEA is introduced. Chapter 2 establishes the theoretical foundation by presenting the core concepts of quantum mechanics relevant to quantum-inspired algorithms. Topics in-

clude quantum bits (qubits), superposition, the Bloch sphere representation, and quantum gates. These principles form the basis for the probabilistic representation and evolution mechanisms employed in QIEA.

Chapter 3 provides a concise overview of classical evolutionary algorithms, while Chapter 4 introduces the QIEA algorithm. This chapter explains how individuals are modeled as probability distributions using sequences of qubits, and it defines the quantum rotation operator, which is the primary operator in QIEA. It also presents a proposed method for solving multi-objective problems that integrates QIEA with NSGA-II.

Chapter 5 examines the incorporation of surrogate models into QIEA to reduce computational costs. It presents radial basis function networks (RBFNs) and presents an integration method for single-objective optimization problems.

Chapter 6 presents benchmark carried out on both single-objective and multi-objective optimization problems, while Chapter 7 summarizes the main findings, analyzes the performance characteristics and limitations of QIEA, and suggests possible directions for future research.

For the purposes of this work, code in the Python programming language was developed and used.

Chapter 2

Quantum Mechanics & Quantum Computing Fundamentals

Before proceeding to the description of QIEA, it is useful to explain some concepts from quantum mechanics and quantum computing that are used by the algorithm. While most of the theory described is used by QIEA, certain parts are included for the sake of completeness. The sections that are directly connected to QIEA will be indicated.

2.1 Basic Unit of Information - Qubit

Classical computers encode information using bits, the fundamental unit of data. A bit exists in one of two discrete states: 0 or 1, which correspond to distinct physical states such as different voltage levels in an electrical circuit.

Quantum computing uses a different approach to information representation through the **qubit** (quantum bit). Qubits leverage quantum mechanical properties to exist in a **quantum state** that can be expressed as a linear combination of two computational basis elements:

- Quantum state 0: $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$
- Quantum state 1: $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$

This quantum mechanical foundation enables qubits to exhibit properties such as superposition and entanglement. These properties provide quantum computers with computational advantages over their classical counterparts for certain problem classes.

2.1.1 Physical Implementations of Qubits

The abstract qubit model corresponds to any physical two-level quantum system. In practice, qubits are realized using various physical platforms:

- **Spin-1/2 particles**: Electron or nuclear spin states, where the two quantum states are represented by spin-up and spin-down along a chosen quantization axis.
- **Photon polarization**: Light polarization states, where the two quantum states are represented by horizontal and vertical polarization.

- **Atomic systems**: Two selected energy levels of an atom, where the two quantum states correspond to the ground state and an excited state.
- **Superconducting circuits**: Distinct charge or flux states in superconducting quantum devices, such as transmon qubits, where the two quantum states correspond to different macroscopic quantum configurations of the circuit.

Regardless of the physical implementation, all qubits share the same mathematical structure and can be manipulated using the same quantum mechanical principles.

2.2 Mathematical Representation and Bra-Ket Notation

To work systematically with quantum states, we represent them as vectors in a two-dimensional complex vector space $\mathbb{C}^2$. Quantum mechanical calculations frequently involve vector operations and inner products between states. The **Bra-ket notation**, introduced by Paul Dirac in 1939 [15], provides an elegant framework for these operations.

2.2.1 Kets

Kets $|\psi\rangle$ are column vectors representing quantum states. For a qubit with complex amplitudes α and β:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{2.1}$$

The ket notation provides a compact way to denote quantum states without explicitly writing out the column vector representation.

2.2.2 Bras

Bras $\langle\psi|$ are row vectors obtained by taking the complex conjugate transpose of kets. For the qubit state $|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$, the corresponding bra is:

$$\langle\psi| = \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix} \tag{2.2}$$

where * denotes complex conjugation.

2.2.3 Inner Product (Bra-ket)

The inner product $\langle\psi|\phi\rangle$ computes the overlap between two quantum states, yielding a complex number. For states $|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ and $|\phi\rangle = \begin{pmatrix} \gamma \\ \delta \end{pmatrix}$:

$$\langle\psi|\phi\rangle = \alpha^*\gamma + \beta^*\delta \tag{2.3}$$

The quantity $|\langle\psi|\phi\rangle|^2$ represents the transition probability between quantum states. Specifically, it gives the probability of measuring state $|\psi\rangle$ given that the qubit is prepared

in state $|\phi\rangle$. For example:

$$\langle 0|0\rangle = 1 \quad \text{(perfect overlap)} \tag{2.4}$$

$$\langle 0|1\rangle = 0 \quad \text{(orthogonal states)} \tag{2.5}$$

$$\langle 0| \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) = \frac{1}{\sqrt{2}} \quad \text{(partial overlap)} \tag{2.6}$$

2.2.4 Outer Product (Ket-Bra)

The outer product $|\psi\rangle\langle\phi|$ creates an operator that acts on quantum states. In matrix form:

$$|\psi\rangle\langle\phi| = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \begin{pmatrix} \gamma^* & \delta^* \end{pmatrix} = \begin{pmatrix} \alpha\gamma^* & \alpha\delta^* \\ \beta\gamma^* & \beta\delta^* \end{pmatrix} \tag{2.7}$$

When this operator acts on a state $|\chi\rangle$, it performs a two-step process:

$$(|\psi\rangle\langle\phi|)|\chi\rangle = |\psi\rangle\langle\phi|\chi\rangle = \langle\phi|\chi\rangle|\psi\rangle \tag{2.8}$$

First, the inner product $\langle\phi|\chi\rangle$ measures how much $|\chi\rangle$ overlaps with $|\phi\rangle$, yielding a complex number. This number then scales the state $|\psi\rangle$ to produce the final result.

A concrete example illustrates this process. The projection operator $|0\rangle\langle 0|$ acts as a filter that extracts only the $|0\rangle$ component of any state:

$$|0\rangle\langle 0| \left(\alpha|0\rangle + \beta|1\rangle\right) = \alpha\langle 0|0\rangle|0\rangle + \beta\langle 0|1\rangle|0\rangle = \alpha|0\rangle \tag{2.9}$$

The inner products evaluate to $\langle 0|0\rangle = 1$ and $\langle 0|1\rangle = 0$, so only the $|0\rangle$ component survives with its original amplitude α.

2.2.5 Orthonormal Basis and Computational Basis

The computational basis states form an orthonormal basis for the qubit space:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.10}$$

These satisfy the orthonormality conditions:

$$\langle 0|0\rangle = \langle 1|1\rangle = 1, \quad \langle 0|1\rangle = \langle 1|0\rangle = 0 \tag{2.11}$$

The completeness relation ensures any qubit state can be expressed in this basis:

$$|0\rangle\langle 0| + |1\rangle\langle 1| = \mathbf{I} \tag{2.12}$$

where $\mathbf{I}$ is the 2×2 identity matrix. This relation guarantees that the computational basis spans the entire two-dimensional Hilbert space $\mathbb{C}^2$, allowing any arbitrary qubit state $|\psi\rangle$ to be uniquely decomposed as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $\alpha, \beta \in \mathbb{C}$.

2.3 Quantum State & Superposition

Building on the concept of qubits introduced earlier, we now formalize what constitutes a **quantum state**. Unlike classical bits that exist in definite states (0 or 1), qubits can exist in a **superposition** of both basic states simultaneously.

Mathematically, any qubit state $|\psi\rangle$ can be written as a linear combination of the computational basis states [16]:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.13}$$

where α and β are complex numbers called **probability amplitudes**. These amplitudes must satisfy the **normalization condition**:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2.14}$$

The physical interpretation is that $|\alpha|^2$ gives the probability of measuring the qubit in state $|0\rangle$, while $|\beta|^2$ gives the probability of measuring it in state $|1\rangle$. This probabilistic framework enables **quantum parallelism**, allowing quantum operations to process exponentially many possibilities simultaneously within the quantum state space.

The mathematical representation of the qubit will be used for the formulation of the individual of the QIEA population. The use of qubits instead of simple bits for encoding endows the individuals with the property of superposition, turning them into probabilistic distributions of the candidate solutions in the search space.

2.4 Measurements and Probabilities

2.4.1 Born's Rule and Quantum Measurement

Born's rule, formulated by Max Born in 1926 [17] [18], provides the fundamental connection between quantum states and measurement outcomes. For a quantum system in state $|\psi\rangle$ and a measurement corresponding to eigenstate $|\phi\rangle$, the probability of observing the system in state $|\phi\rangle$ is given by:

$$P(\phi) = |\langle\phi|\psi\rangle|^2 \tag{2.15}$$

An **eigenstate** is a quantum state that yields a definite, predictable result when measured. For computational basis measurements, the eigenstates are $|0\rangle$ and $|1\rangle$, which when measured produce the classical bit values 0 and 1, respectively, with certainty.

For a qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the measurement probabilities are:

$$P(|0\rangle) = |\langle 0|\psi\rangle|^2 = |\alpha|^2 \tag{2.16}$$
$$P(|1\rangle) = |\langle 1|\psi\rangle|^2 = |\beta|^2 \tag{2.17}$$

where the normalization condition $|\alpha|^2 + |\beta|^2 = 1$ ensures that the total probability equals one.

In the QIEA algorithm, we will use the measurement operation in order to obtain candidate solutions from the quantum individuals, which represent probability distributions.

2.4.2 Measurement Process and Fundamental Limitations

When measuring quantum states, we face inherent constraints:

- Cannot directly observe superposition
- Only measure definite outcomes ($|0\rangle$ or $|1\rangle$)
- Measurement causes irreversible state collapse to the observed state
- Original superposition information is permanently lost [19]

To reconstruct quantum state information requires multiple measurements on identically prepared systems and statistical estimation of probability amplitudes.

2.5 Geometric Representation: The Bloch Sphere

Although the vector representation offers a rigorous mathematical framework for qubits, visualizing quantum operations and state transformations in this form can be challenging. The Bloch sphere provides an intuitive geometric representation that maps every possible qubit state to a point on the surface of a unit sphere.

Through the Bloch sphere and Bloch para, the qubits of the individuals in the QIEA population can, on the one hand, be visualized and, on the other hand, be parameterized in a more convenient way.

2.5.1 From Complex Amplitudes to Cartesian Coordinates

Starting with the general qubit state:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.18}$$

where $|\alpha|^2 + |\beta|^2 = 1$.

Since α and β are complex numbers, we can express them in polar form:

$$\alpha = |\alpha|e^{i\phi_\alpha} \tag{2.19}$$

$$\beta = |\beta|e^{i\phi_\beta} \tag{2.20}$$

Substituting into our quantum state:

$$|\psi\rangle = |\alpha|e^{i\phi_\alpha}|0\rangle + |\beta|e^{i\phi_\beta}|1\rangle \tag{2.21}$$

Using the distributive property, we can factor out the common phase $e^{i\phi_\alpha}$ from both terms:

$$|\psi\rangle = e^{i\phi_\alpha}\left(|\alpha||0\rangle + |\beta|e^{i(\phi_\beta-\phi_\alpha)}|1\rangle\right) \tag{2.22}$$

In this factored form, we can identify two types of phases. The factor $e^{i\phi_\alpha}$ that multiplies the entire state is called the **global phase**, while the term $e^{i(\phi_\beta-\phi_\alpha)}$ represents the **relative phase** between the two computational basis states. The global phase has no physical significance by the Born rule.

According to Born Rule, all measurement probabilities are given by $|\langle\phi|\psi\rangle|^2$.

Since $|\langle\phi|e^{i\alpha}\psi\rangle|^2 = |e^{i\alpha}|^2|\langle\phi|\psi\rangle|^2 = |\langle\phi|\psi\rangle|^2$, states $e^{i\alpha}|\psi\rangle$ and $|\psi\rangle$ are physically equivalent for any real α.

Therefore, we can ignore the global phase factor, and the equation above is equivalent to:

$$|\psi\rangle = |\alpha||0\rangle + |\beta|e^{i(\phi_\beta - \phi_\alpha)}|1\rangle \tag{2.23}$$

Let $\phi = \phi_\beta - \phi_\alpha$, giving us:

$$|\psi\rangle = |\alpha||0\rangle + |\beta|e^{i\phi}|1\rangle \tag{2.24}$$

Applying Euler's equation $e^{i\phi} = \cos\phi + i\sin\phi$:

$$|\psi\rangle = |\alpha||0\rangle + |\beta|(\cos\phi + i\sin\phi)|1\rangle \tag{2.25}$$

$$|\psi\rangle = |\alpha||0\rangle + |\beta|\cos\phi|1\rangle + i|\beta|\sin\phi|1\rangle \tag{2.26}$$

Now we make the following substitutions to connect to Cartesian coordinates:

$$z = |\alpha| \tag{2.27}$$
$$x = |\beta|\cos\phi \tag{2.28}$$
$$y = |\beta|\sin\phi \tag{2.29}$$

This gives us:

$$|\psi\rangle = z|0\rangle + (x + iy)|1\rangle \tag{2.30}$$

From the normalization condition $\langle\psi|\psi\rangle = 1$:

$$|z|^2 + |x + iy|^2 = 1 \tag{2.31}$$

Since $z = |\alpha|$ is real, $|z|^2 = z^2$. For the complex term:

$$|x + iy|^2 = (x + iy)(x - iy) = x^2 - ixy + ixy + y^2 = x^2 + y^2 \tag{2.32}$$

Therefore:

$$z^2 + x^2 + y^2 = 1 \tag{2.33}$$

This shows that the coordinates (x, y, z) lie on the surface of a unit sphere!

2.5.2 Converting to Spherical Coordinates

To parametrize points on a unit sphere, we start with the standard spherical coordinates:

$$x = \sin\theta\cos\phi \tag{2.34}$$
$$y = \sin\theta\sin\phi \tag{2.35}$$
$$z = \cos\theta \tag{2.36}$$

where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$.

Substituting back into our quantum state $|\psi\rangle = z|0\rangle + (x + iy)|1\rangle$:

$$|\psi\rangle = \cos\theta|0\rangle + [\sin\theta\cos\phi + i\sin\theta\sin\phi]\,|1\rangle \tag{2.37}$$

Factoring out $\sin\theta$ and applying Euler's equation in reverse:

$$|\psi\rangle = \cos\theta|0\rangle + \sin\theta(\cos\phi + i\sin\phi)|1\rangle \tag{2.38}$$

$$|\psi\rangle = \cos\theta|0\rangle + e^{i\phi}\sin\theta|1\rangle \tag{2.39}$$

For this parameterization, we have:

$$\theta = 0: \quad |\psi\rangle = |0\rangle \tag{2.40}$$

$$\theta = \frac{\pi}{2}: \quad |\psi\rangle = e^{i\phi}|1\rangle \equiv |1\rangle \tag{2.41}$$

To extend the range of θ from $[0, \pi/2]$ to $[0, \pi]$, we set $\tilde{\theta} = 2\theta$. This gives us $0 \leq \theta \leq \pi/2 \Rightarrow 0 \leq \tilde{\theta} \leq \pi$. Replacing θ with $\tilde{\theta}/2$ in the original equation and renaming $\tilde{\theta} \to \theta$, we obtain:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{2.42}$$

This way we have:

$$\theta = 0: \quad |\psi\rangle = |0\rangle \tag{2.43}$$

$$\theta = \pi: \quad |\psi\rangle = |1\rangle \tag{2.44}$$

2.5.3 The Bloch Sphere Parametrization

From our previous derivation, we established that any qubit state can be written as:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{2.45}$$

This is the standard **Bloch sphere parametrization**, where any qubit state is completely determined by just two angles:

- $\theta \in [0, \pi]$: the **polar angle** that controls how much the state "leans" toward $|0\rangle$ or $|1\rangle$
- $\phi \in [0, 2\pi)$: the **azimuthal angle** that determines the relative phase between the amplitudes

To visualize this quantum state geometrically, we map it to a point (x, y, z) on the surface of a unit sphere using the standard spherical coordinate transformation:

$$x = \sin\theta\cos\phi \tag{2.46}$$

$$y = \sin\theta\sin\phi \tag{2.47}$$

$$z = \cos\theta \tag{2.48}$$

This creates the **Bloch sphere** (Figure 2.1), a unit sphere where every possible qubit state $|\psi\rangle$ corresponds to exactly one point on its surface, specified by the angles ϕ and θ [20]. The quantum state $|0\rangle$ is located at the north pole, $|1\rangle$ at the south pole, with superposition states occupying all other points on the sphere's surface.

Key geometric features:

- When $\theta = 0$: The state is $|0\rangle$, located at the north pole $(0, 0, 1)$
- When $\theta = \pi$: The state is $|1\rangle$, located at the south pole $(0, 0, -1)$
- When $\theta = \pi/2$: The state is an equal superposition, located on the equator

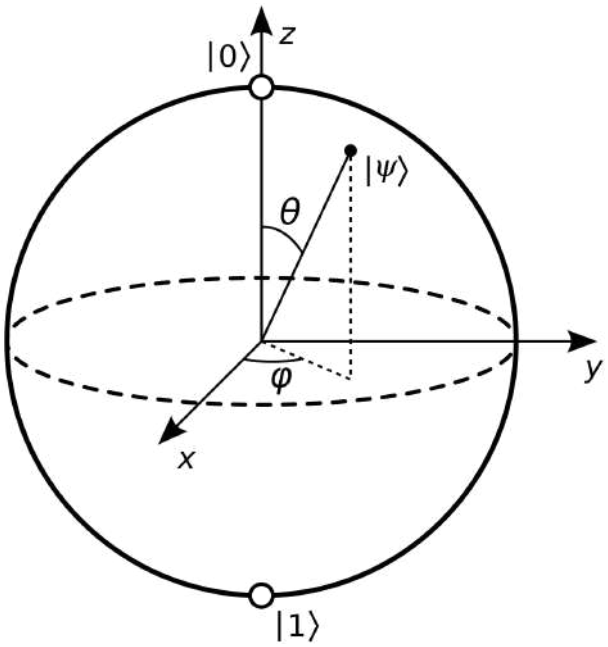


Figure 2.1: The Bloch Sphere

- The ϕ angle determines which point on the equator (or any latitude circle)

The measurement probabilities are directly related to the sphere coordinates:

$$P(|0\rangle) = \cos^2\left(\frac{\theta}{2}\right) \tag{2.49}$$

$$P(|1\rangle) = \sin^2\left(\frac{\theta}{2}\right) \tag{2.50}$$

This means the higher up on the sphere (larger z), the more likely we are to measure $|0\rangle$. The lower down (smaller z), the more likely we are to measure $|1\rangle$. Points on the equator ($z = 0$) give equal probabilities for both outcomes.

2.6 Multiple Qubits and Tensor Product

For systems containing multiple qubits, we need a mathematical framework to combine individual quantum systems into composite systems. This is achieved through the **tensor product** operation, which allows us to construct multi-qubit states from single-qubit components.

2.6.1 The Tensor Product Operation

The tensor product $|\psi\rangle \otimes |\phi\rangle$ (often written as $|\psi\rangle|\phi\rangle$ or $|\psi\phi\rangle$) combines separate quantum systems into a composite system. For single qubits $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and $|\phi\rangle = \gamma|0\rangle + \delta|1\rangle$:

$$|\psi\rangle \otimes |\phi\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) \tag{2.51}$$

$$= \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle \tag{2.52}$$

Matrix Representation: If $|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$ and $|\phi\rangle = \begin{pmatrix}\gamma\\ \delta\end{pmatrix}$, then:

$$|\psi\rangle \otimes |\phi\rangle = \begin{pmatrix}\alpha\gamma\\ \alpha\delta\\ \beta\gamma\\ \beta\delta\end{pmatrix} \tag{2.53}$$

Properties of Tensor Products:

- **Distributivity:** $(a|\psi_1\rangle + b|\psi_2\rangle) \otimes |\phi\rangle = a|\psi_1\rangle \otimes |\phi\rangle + b|\psi_2\rangle \otimes |\phi\rangle$
- **Associativity:** $(|\psi\rangle \otimes |\phi\rangle) \otimes |\chi\rangle = |\psi\rangle \otimes (|\phi\rangle \otimes |\chi\rangle)$
- **Mixed Products:** $(\langle\psi_1| \otimes \langle\phi_1|)(|\psi_2\rangle \otimes |\phi_2\rangle) = \langle\psi_1|\psi_2\rangle\langle\phi_1|\phi_2\rangle$

The state space of a system with n qubits has dimension 2^n. For two qubits, the basic computational basis states are:

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \tag{2.54}$$

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \tag{2.55}$$

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \tag{2.56}$$

$$|11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \tag{2.57}$$

where $\otimes$ denotes the **tensor product**.

The general state of two qubits is expressed as a linear combination of all computational basis states:

$$|\Psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle \tag{2.58}$$

with the normalization condition:

$$\sum_{i,j\in\{0,1\}} |\alpha_{ij}|^2 = 1 \tag{2.59}$$

2.6.2 Separable and Entangled States

A two-qubit state is called **separable** if it can be written as a tensor product of two single-qubit states:

$$|\Psi\rangle_{\text{separable}} = |\psi\rangle \otimes |\phi\rangle \tag{2.60}$$

If a state cannot be written in this form, it is called **entangled**. A famous example of an entangled state is the Bell state:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{2.61}$$

2.6.3 Classical vs. Quantum Information Storage

The exponential growth of quantum state spaces reveals a fundamental difference between classical and quantum information storage requirements.

Quantum State Description: An n-qubit quantum system requires 2^n complex probability amplitudes to fully specify its state:

$$|\Psi\rangle = \sum_{i_1,i_2,\ldots,i_n \in \{0,1\}} \alpha_{i_1 i_2 \ldots i_n} |i_1 i_2 \ldots i_n\rangle \tag{2.62}$$

where each $\alpha_{i_1 i_2 \ldots i_n} \in \mathbb{C}$ is a complex number with real and imaginary parts.

Classical Storage Requirements: To store the complete quantum state information on a classical computer, we need:

- 2^n complex amplitudes
- Each complex number = 2 real numbers (real and imaginary parts)
- Each real number = b bits (e.g., 64 bits for double precision)
- **Total bits required:** $2 \times 2^n \times b = 2^{n+1} \times b$

Comparison Table:

Qubits (n)	Quantum Amplitudes	Real Numbers	Bits (64-bit precision)
1	$2^1 = 2$	4	256
2	$2^2 = 4$	8	512
3	$2^3 = 8$	16	1,024
10	$2^{10} = 1,024$	2,048	$\approx$ 16 kB
20	$2^{20} \approx 10^6$	$\approx 2 \times 10^6$	$\approx$ 134 MB
50	$2^{50} \approx 10^{15}$	$\approx 2 \times 10^{15}$	$\approx$ 144 PB
300	$2^{300} \approx 10^{90}$	$\approx 2 \times 10^{90}$	$\approx 10^{92}$ bits

Implications:

- For small n (1-10 qubits), classical simulation is feasible
- Beyond $n \approx 50$ qubits, classical storage becomes prohibitive
- At $n = 300$ qubits, the required classical memory exceeds the estimated number of atoms in the observable universe ($\approx 10^{80}$)

2.7 Gates & Operations

Quantum gates are the building blocks of a quantum computer and allow us to manipulate the state of qubits [21]. Quantum gates implement unitary and, therefore, reversible transformations. Mathematically, each quantum gate is described by a **unitary matrix** U, which satisfies the condition:

$$U^\dagger U = U U^\dagger = I \tag{2.63}$$

where $U^\dagger$ is the conjugate transpose of U and I is the identity matrix. This property ensures the preservation of normalization and the reversibility of quantum operations.

The action of a quantum gate U on a state $|\psi\rangle$ is given by:

$$|\psi'\rangle = U|\psi\rangle \tag{2.64}$$

The use of quantum gates, and specifically the *Hadamard* and R_y gates presented below, will allow us to control the quantum state of the qubits both during their initialization and throughout the evolution phase of the QIEA algorithmic process. The rest of the gates presented are not used in the implementation of the QIEA in this work but are included for the sake of completeness.

2.7.1 Basic Single-qubit Gates

Pauli Gates The Pauli gates constitute the fundamental set for manipulating individual qubits:

Pauli-X Gate (Bit Flip):

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2.65}$$

The X gate, also referred to as the NOT gate, flips the qubit state: $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$. Geometrically, the X gate performs a rotation by π around the x-axis of the Bloch sphere (Figure 2.1).

Pauli-Z Gate (Phase Flip):

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.66}$$

The Z gate introduces a phase of -1 to the state $|1\rangle$: $Z|0\rangle = |0\rangle$ and $Z|1\rangle = -|1\rangle$. Geometrically, the Z gate performs a rotation by π radians around the z-axis of the Bloch sphere (Figure 2.1).

Pauli-Y Gate:

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{2.67}$$

The Y gate combines both bit and phase flip operations: $Y|0\rangle = i|1\rangle$ and $Y|1\rangle = -i|0\rangle$. Geometrically, the Y gate performs a rotation by π radians around the y-axis of the Bloch sphere (Figure 2.1). It can be expressed as the composition $Y = iXZ$.

Hadamard Gate The Hadamard gate is crucial for creating superposition:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.68}$$

with action:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{2.69}$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{2.70}$$

The Hadamard gate creates states of maximum superposition from the computational basis states.

Rotation Gates Rotation gates perform rotations around the principal axes of the Bloch sphere through a continuous angle parameter θ:

Rotation-X Gate:

$$R_x(\theta) = \begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \tag{2.71}$$

The R_x gate rotates the qubit state around the X-axis of the Bloch sphere (Figure 2.1).

Rotation-Y Gate:

$$R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \tag{2.72}$$

The R_y gate rotates the qubit state around the Y-axis of the Bloch sphere (Figure 2.1).

Rotation-Z Gate:

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \tag{2.73}$$

The R_z gate rotates the qubit state around the Z-axis of the Bloch sphere (Figure 2.1).

It can be shown that any single-qubit unitary operator can be decomposed as

$$U = R_Z(\alpha)\, R_Y(\beta)\, R_Z(\gamma). \tag{2.74}$$

Here is the refined and expanded section:

2.7.2 Two-Qubit Gates

While single-qubit gates provide the foundation for quantum operations, two-qubit gates enable interactions between qubits and are essential for creating entangled states.

Controlled-NOT (CNOT) Gate The Controlled-NOT (CNOT) gate is one of the most fundamental and essential two-qubit gates. It is a controlled operation that acts on a pair of qubits: a **control qubit** and a **target qubit**. The gate applies a Pauli-X operation to the target qubit if and only if the control qubit is in state $|1\rangle$.

Mathematical Representation: The CNOT gate can be expressed using tensor products of projectors:

$$\text{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X \tag{2.75}$$

In matrix form, acting on the computational basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.76}$$

Action on Basis States: The CNOT gate transforms the computational basis states as follows:

$$\text{CNOT}|00\rangle = |00\rangle \quad \text{(control = 0, target unchanged)} \tag{2.77}$$

$$\text{CNOT}|01\rangle = |01\rangle \quad \text{(control = 0, target unchanged)} \tag{2.78}$$

$$\text{CNOT}|10\rangle = |11\rangle \quad \text{(control = 1, target flipped)} \tag{2.79}$$

$$\text{CNOT}|11\rangle = |10\rangle \quad \text{(control = 1, target flipped)} \tag{2.80}$$

Properties and Applications:

- **Reversibility**: The CNOT gate is its own inverse, meaning $\text{CNOT}^2 = I$.
- **Entanglement Generation**: When applied to superposition states, CNOT can create entangled states. For example:

$$\text{CNOT}\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |0\rangle\right) = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{2.81}$$

  which is the maximally entangled Bell state $|\Phi^+\rangle$.

Controlled Phase Gate Another important two-qubit gate is the Controlled-Z (CZ) gate, which applies a phase flip to the target qubit conditional on the control qubit being in state $|1\rangle$. Its matrix representation is:

$$\text{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{2.82}$$

Unlike CNOT, the CZ gate is symmetric with respect to the two qubits, and it preserves the computational basis while introducing a relative phase.

Other Quantum Gates There exist several other single-qubit gates, such as the T gate (which introduces a $\pi/4$ phase) and the S gate (which applies a $\pi/2$ phase), as well as numerous multi-qubit gates, including the SWAP gate (which exchanges the states of two qubits), the Toffoli gate (a three-qubit controlled-controlled-NOT gate), and the Fredkin gate (a controlled-SWAP gate) and many more. We include them as a reference; however, a detailed analysis of them would fall outside the scope of the present work.

Chapter 3

Evolutionary Algorithms

Consider a continuous optimization problem defined as

$$\min_{\mathbf{x}\in\mathcal{D}} F(\mathbf{x}), \tag{3.1}$$

where $F : \mathbb{R}^m \to \mathbb{R}$ is the objective function to be minimized, $\mathbf{x} = (x_1, x_2, \ldots, x_m)^T$ is an m-dimensional decision vector, and $\mathcal{D} \subseteq \mathbb{R}^m$ is the feasible search space. The fitness of a candidate solution $\mathbf{x}_j$ is evaluated by computing $F(\mathbf{x}_j)$, with lower values indicating better solutions.

3.1 Description of an Evolutionary Algorithm

Evolutionary algorithms handle populations of solutions. During the evolution, which is the main procedure of optimal solution search, a population of μ evaluated solutions (parents as per genetic terminology) evolves to a population of λ offspring. These offspring are new solutions that arise from the μ parents, possibly having better characteristics. From the λ offspring, the μ parents of the next generation are selected based on their fitness. The process continues until some termination criterion is satisfied. Such termination criteria may be the absence of further improvement of the solution for a number of evaluations or generations, the homogenization of the population, and/or the exhaustion of the maximum available computational time.

Encoding of Design Variables Evolutionary algorithms can use either binary or real encoding of the design variables. In this subsection the binary encoding will be briefly presented. Given a vector of design variables $\mathbf{x}$ with k components $x_1, x_2, ..., x_k$, binary encoding requires defining a lower boundary $x_{\ell,l}$, an upper boundary $x_{\ell,u}$ and the count of binary digits m_ℓ for every variable $\ell, \ell \in \mathbb{N}$. The count m_ℓ defines the discretization precision of the ℓ-th variable, given by the equation

$$\Delta x_\ell = \frac{x_{\ell,u} - x_{\ell,l}}{2^{m_\ell} - 1} \tag{3.2}$$

The combination of the binary arrays of all the variables creates the unified binary array (chromosome) that completely describes every candidate solution.

3.1.1 Evolution Operators

Parent Selection Operator

The parent selection operator T_μ is responsible for generating the parent set $S^{g+1,\mu}$ of the next generation by selecting individuals from the offspring set $S^{g,\lambda}$ and the parent set $S^{g,\mu}$ of the current generation.

The parent selection process gives higher chances to the individuals of $S^{g,\lambda}$ with better objective function values to participate in the creation of offspring. Conversely, the worst of the current solutions have lower probabilities of participating in the set $S^{g+1,\mu}$. Therefore, selection is based on (mainly relative) values of the objective function. There are several parent selection methods, some of which are mentioned below.

Roulette wheel selection In roulette wheel selection for minimization problems, the selection probability is inversely proportional to the objective function value. In the simple case where $F(\mathbf{x}_j) > 0$, the j-th individual among the λ individuals is given a probability

$$p_j = \frac{1/F(\mathbf{x}_j)}{\sum_{i=1}^{\lambda} 1/F(\mathbf{x}_i)}.$$

Parents are selected with the aid of a random number generator, taking into account the above probabilities. Thus, individuals with lower objective function values have a higher chance of being selected. Nevertheless, there is a risk that the presence of a comparatively good solution in the initial generations may lead to its premature dominance and therefore to entrapment in a local optimum.

Rank selection In rank selection, an individual's chance of being chosen is based not on its absolute fitness value but on its position in the fitness ranking within the population. The precise fitness values are not required; a sorting of the individuals is sufficient.

Tournament selection In tournament selection, k individuals are randomly chosen at a time, they compete with each other based on the value of the objective function, and the best one is selected as another member of $S^{g+1,\mu}$.

Crossover Operator

For binary encoding, crossover combines two or more randomly selected parents to create an equal number of offspring. In the literature, we can also encounter crossover operations involving more than two parents.

In binary encoding, the so-called single-point crossover first randomly selects a position between two consecutive binary digits of the chromosome. Then, it randomly selects two parents from the pool of selected parents using the selection operators presented in the previous paragraph. The two parent sequences undergo a crossover, meaning their segments on each side of the crossover point are swapped, resulting in two new individuals.

In the literature we will encounter a variety of crossover schemes. Based on the above, it is easy to understand two-point crossover (two random crossover positions; the first offspring is formed from the first and third segment of the first parent and the second segment of the second parent, etc.), or one-point crossover per variable (a random crossover point is selected for each variable, and the rest of the procedure is identical).

Finally, it is worth mentioning that, in the context of a stochastic algorithm, such as an evolutionary algorithm, crossover is performed with probability P_r (usually close to one, e.g., $P_r = 0.90$). The role of a non-unit crossover probability is to allow, with a small probability (equal to $1 - P_r$), the transfer of unaltered characteristics from the parents to the offspring. With probability P_r the crossover proceeds as usual, and with probability $1 - P_r$ the offspring are produced as exact copies of their parents.

Mutation Operator

The mutation operator is applied to every new offspring that has resulted from the application of the crossover operator. The purpose of mutation is to introduce new genetic material into the offspring population, with probability P_m (usually very small, for example $P_m = 0.01$).

In binary encoding, mutation allows (with the above small probability) the selected binary digit to be flipped (if it was 0 it becomes 1 and vice versa). In practice, for every binary digit of each offspring produced by crossover, a random number is generated (in the interval $[0, 1]$), and mutation is performed only if this random number is smaller than P_m.

Elitism Operator

The selection operator replaces individuals in the parents' set $S^{g,\lambda}$ with individuals from the elites' set $S^{g,e}$. The elements to be replaced depends on the value of the objective function (typically the poorest-performing ones), although a subset may also be selected at random. The elitism operator can be represented mathematically as follows:

$$S^{g,\lambda} = T_{e2}(S^{g,\lambda} \cup S^{g+1,e}) \tag{3.3}$$

Elitism ensures that no generation will produce solutions worse than those of the previous generation. At the same time, it rewards and exploits the good solutions that have been found so far.

3.1.2 Algorithmic Procedure

The whole algorithmic procedure of genetic algorithm is presented.

1. **Initialization**: Configuration parameters of the algorithm are set (μ and λ). Members of initial population $S^{0,\lambda}$ are randomly selected
2. **Main Evolution Loop**: While no termination criterion is satisfied:
   (a) **Evaluation**: λ individuals are evaluated by computing the value of the objective function
   (b) **Elites Selection**: Elites' population $S^{g,e}$ is updated after the evaluation of the new individuals
   (c) **Elitism Operator Application**: Some members of the parents' population $S^{g,\mu}$ are replaced by elites
   (d) **Offspring Generation**: The following sequence of operators is applied until λ offspring have been obtained
      i. **Parent Selection**

ii. **Crossover**
iii. **Mutation**

3.2 Evolutionary Algorithms in Multi-Objective Optimization

A brief presentation is given of the principle by which a single-objective evolutionary algorithm can be transformed so that it can solve multi-objective problems [22]. The formulation is made for minimization problems, without this being a limitation.

$$min\mathbf{F}(\mathbf{x}), \mathbf{F} : \mathbb{R}^m \rightarrow \mathbb{R}^M \tag{3.4}$$

where M is the number of objective functions.

3.2.1 Pareto Dominance and Pareto Front

A large family of evolutionary algorithms for multi-objective optimization makes use of the principle of **Pareto dominance**. Considering two candidate solutions $\mathbf{x}$ and $\mathbf{y}$, according to the definition of Pareto dominance:

$$\begin{cases} F_i(\mathbf{x}) \leq F_i(\mathbf{y}) \ \forall \ i \in \{1, 2, \ldots, M\} \\ \exists \ j \in \{1, 2, \ldots, M\} : F_j(\mathbf{x}) < F_j(\mathbf{y}) \end{cases} \tag{3.5}$$

where the notation $\mathbf{x} \prec \mathbf{y}$ means that **x dominates on y**.

The **Pareto front** is defined as the subset of candidate solutions that are not dominated by any other solution.

Two solutions of the Pareto front cannot be compared to each other. Therefore, multi-objective optimization methods that rely on the Pareto concept do not return a single solution but a set of solutions that form a Pareto front.

When distinguishing between the true Pareto front of a problem and the Pareto front obtained by an optimization method, there are two primary objectives. First, the computed front should approximate the true front as closely as possible. The second objective is that the finite set of points that comprise the computed front should be distributed as uniformly as possible along the true front; in other words, this finite set of solutions should cover the widest possible range of the true front, with a sufficient presence of solutions in all segments.

Various Pareto-based multi-objective optimization methods have been developed, which attempt to satisfy the requirements formulated above. Some of the most well-known and broadly used ones are the Non-dominated Sorting Genetic Algorithm II (NSGA-II), the Strength Pareto Evolutionary Algorithm 2 (SPEA2), and the Pareto Archived Evolution Strategy (PAES). These methods have different ranking schemes, strategies for preserving diversity, and archiving, but all are based on the Pareto dominance principle for guiding the search to approximate the true Pareto-optimal front.

Chapter 4

QIEA

This chapter presents the Quantum-Inspired Evolutionary Algorithm (QIEA), a metaheuristic optimization method that incorporates quantum mechanical principles into evolutionary computation. Building upon the quantum computing fundamentals and evolutionary paradigms introduced in previous chapters, we present an algorithm where individuals represent probability distributions over the solution space through quantum superposition, rather than discrete candidate solutions. We define how individuals are represented based on quantum mechanical principles and specify the core components of the algorithm.

4.1 Solution Encoding and Decoding

While QIEA is predominantly applied to combinatorial optimization problems in the literature, this work adapts it for continuous numerical optimization. This adaptation requires an encoding scheme to represent real-valued decision variables as binary strings that can be processed by the quantum-inspired operators.

4.1.1 Binary Encoding Scheme

We employ a standard binary encoding approach for real-valued variables. Consider a decision variable x with domain $[x_l, x_u]$ that is encoded using m bits. The binary representation is given by the bit sequence:

$$x_{encoded} = \mathbf{b} = [b_0 b_1 \dots b_{m-1}], \quad b_i \in \{0, 1\} \tag{4.1}$$

To convert the binary sequence $\mathbf{b}$ to its decimal representation $(x_{encoded})_{10}$, we apply the standard positional notation formula:

$$(x_{encoded})_{10} = \sum_{i=0}^{m-1} 2^{m-1-i} b_i \tag{4.2}$$

The decoding process maps the binary chromosome to a real value by uniformly distributing all possible binary representations across the variable's domain. With m bits, there are 2^m distinct binary strings, yielding $2^m - 1$ intervals between the upper bound b_u and lower bound b_l. The resulting discretization step size is:

$$\Delta x = \frac{x_u - x_l}{2^m - 1} \tag{4.3}$$

Each binary value, then, corresponds to a real value in the domain according to:

$$x = x_l + \Delta x \cdot (x_{encoded})_{10} \tag{4.4}$$

Substituting the expressions for $(x_{encoded})_{10}$ and Δx, we obtain the complete decoding formula:

$$x = x_l + \frac{x_u - x_l}{2^m - 1} \sum_{i=0}^{m-1} 2^{m-i-1} b_i \tag{4.5}$$

This mapping ensures that the binary string $00\ldots0$ corresponds to the lower bound x_l, while $11\ldots1$ corresponds to the upper bound x_u, with uniform spacing between intermediate values.

4.1.2 Multi-Variable Encoding

For optimization problems involving k decision variables, the complete binary chromosome consists of k concatenated segments, where each segment represents a single design variable encoded according to the scheme described in the previous section.

Let m_ℓ denote the number of bits allocated to variable ℓ. The total chromosome length is then:

$$m = \sum_{\ell=1}^{k} m_\ell \tag{4.6}$$

The bit segment corresponding to variable ℓ is extracted from the complete chromosome as:

$$\mathbf{b}_\ell = \left[b_{\sum_{s=1}^{\ell-1} m_s}, b_{1+\sum_{s=1}^{\ell-1} m_s}, \ldots, b_{m_\ell - 1 + \sum_{s=1}^{\ell-1} m_s} \right], \quad \ell = 1, 2, \ldots, k \tag{4.7}$$

where the summation $\sum_{s=1}^{\ell-1} m_s$ represents the cumulative bit count of all preceding variables, determining the starting index of segment ℓ within the chromosome.

Each variable segment is decoded independently using the single-variable formula:

$$x_\ell = x_{l,\ell} + \frac{x_{u,\ell} - x_{l,\ell}}{2^{m_\ell} - 1} \sum_{i=0}^{m_\ell - 1} 2^{m_\ell - i - 1} b_{i + \sum_{s=1}^{\ell-1} m_s}, \quad \ell = 1, 2, \ldots, k \tag{4.8}$$

where $x_{l,\ell}$ and $x_{u,\ell}$ denote the lower and upper bounds for variable ℓ, respectively.

The complete solution vector is then constructed as:

$$\mathbf{x} = [x_1, x_2, \ldots, x_k]^T \tag{4.9}$$

4.1.3 Implementation Considerations

In our implementation, the evolutionary algorithm generates solutions directly in binary form through quantum-inspired operators. The decoding process is applied only when solutions require evaluation by the objective function, which operates on real-valued inputs.

4.2 Solution Representation

In the Quantum-Inspired Evolutionary Algorithm (QIEA), the representation of solutions is based on the fundamental principles of quantum mechanics described in Chapter 2. Instead of the classical binary sequences of conventional evolutionary algorithms, population individuals are represented as qubit sequences, exploiting the superposition capability for simultaneous representation of multiple possible solutions.

Section 2.6 discusses the computational limitations of qubit representations, where a fully entangled representation of m qubits would require 2^m bits. This becomes prohibitive for real-world problems where $m \geq 100$. As a foundational approach in this work, we therefore design the quantum representation by treating each bit position as an independent qubit. This choice avoids exponential memory scaling while preserving quantum mechanical principles through superposition and quantum gate operations. Future implementations could explore multi-qubit entangled subsystems per design variable to potentially capture additional quantum correlations.

4.2.1 Quantum Representation of Individual

An individual $\mathbf{q}$ in QIEA is represented as a sequence of m single qubits:

$$\mathbf{q} = \left[\begin{pmatrix} \alpha_0 \\ \beta_0 \end{pmatrix} \quad \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \quad \cdots \quad \begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} \quad \cdots \quad \begin{pmatrix} \alpha_{m-1} \\ \beta_{m-1} \end{pmatrix} \right]$$

Each qubit i is described by the quantum state:

$$|\psi_i\rangle = \alpha_i|0\rangle + \beta_i|1\rangle$$

According to the fundamental theory of qubits, the probability amplitudes α_i and β_i satisfy the normalization condition:

$$|\alpha_i|^2 + |\beta_i|^2 = 1$$

and their physical interpretation, according to Born's interpretation, is:

- $|\alpha_i|^2$: probability of measuring state $|0\rangle$ for the i-th qubit
- $|\beta_i|^2$: probability of measuring state $|1\rangle$ for the i-th qubit

4.2.2 Angular Parameterization and Bloch Sphere

Exploiting the Bloch sphere parameterization (Equation 2.45), we adopt a simplification suitable for practical implementation of QIEA. Specifically, we restrict the quantum state to real amplitudes, i.e., $\alpha_i, \beta_i \in \mathbb{R}$, which corresponds to setting the azimuthal angle $\phi = 0$. Under this constraint, the general state vector reduces to:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)|1\rangle, \tag{4.10}$$

where the polar angle $\theta \in [0, \pi]$ fully parameterizes the state within the real meridian of the Bloch sphere.

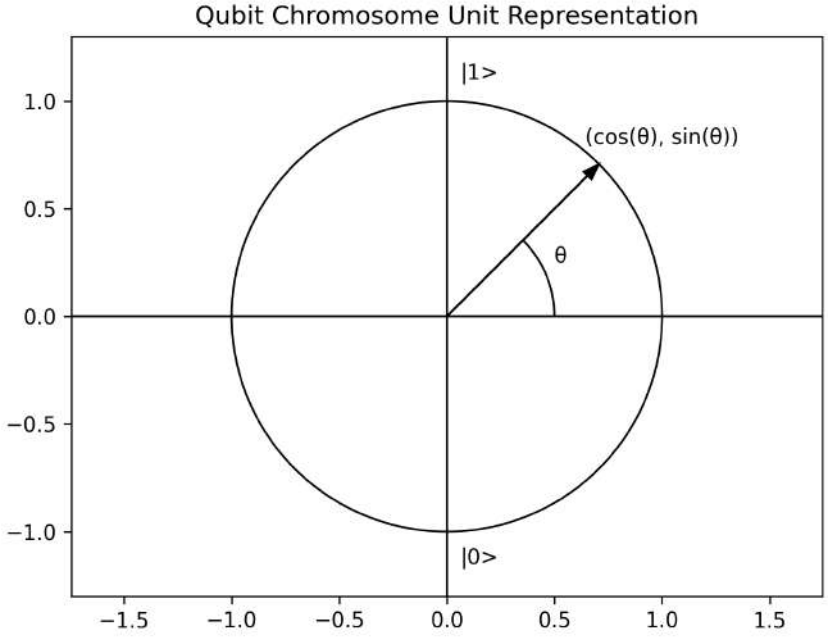


Figure 4.1: Qubit Chromosome Unit Representation

Next, we introduce the parameterization $\theta = \tilde{\theta} + \frac{\pi}{2}$. The state becomes:

$$|\psi\rangle = \cos\left(\frac{\tilde{\theta}}{2} + \frac{\pi}{4}\right)|0\rangle + \sin\left(\frac{\tilde{\theta}}{2} + \frac{\pi}{4}\right)|1\rangle$$

For simplicity, we rename $\tilde{\theta}$ to θ and continue with this notation. Using trigonometry, the measurement probabilities result:

$$P(|0\rangle)(\theta) = \cos^2\left(\frac{\theta}{2} + \frac{\pi}{4}\right) = \frac{1}{2} - \frac{1}{2}\sin(\theta), \quad \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \tag{4.11}$$

$$P(|1\rangle)(\theta) = \sin^2\left(\frac{\theta}{2} + \frac{\pi}{4}\right) = \frac{1}{2} + \frac{1}{2}\sin(\theta), \quad \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \tag{4.12}$$

This parameterization preserves the physically interpretable properties:

- $\theta_i = 0$: $P(|0\rangle)_i = P(|1\rangle)_i = \frac{1}{2}$ (maximum superposition - Hadamard state)
- $\theta_i = \frac{\pi}{2}$: $P(|0\rangle)_i = 0, P(|1\rangle)_i = 1$ (pure state $|1\rangle$)
- $\theta_i = -\frac{\pi}{2}$: $P(|0\rangle)_i = 1, P(|1\rangle)_i = 0$ (pure state $|0\rangle$)

The graphical representation of the qubit's quantum state based on the angular parameter θ, as defined above, is shown in Figure 4.1.

With angular parameterization, an individual q is expressed compactly as:

$$\mathbf{q} = \begin{bmatrix}\theta_0 & \theta_1 & \cdots & \theta_i & \cdots & \theta_{m-1}\end{bmatrix}$$

where each $\theta_i \in \mathbb{R}$ completely determines the quantum state of the corresponding qubit.

This formulation enables the representation of a probability distribution across the entire search domain using a single quantum individual. Rather than representing a single point in the search space, as is the case in the evolutionary algorithm, the quantum individual simultaneously represents all possible solutions, each with an associated measurement probability. Solutions with higher probabilities correspond to regions of the search space that are more likely to contain optimal solutions. In this manner, a quantum individual represents the complete set of solution candidates, with the probability distribution guiding the search toward more promising regions.

4.2.3 Measurement in QIEA

In order to obtain a solution out of a quantum individual, we need to perform a measurement on each qubit of the individual. As already mentioned the measurement probabilities of $|0\rangle$ and $|1\rangle$ for each qubit are given by relations (4.11) and (4.12). By applying these relations to a sequence of qubits, we can calculate the probability of measuring a binary sequence in a quantum individual whose qubits' quantum state is known.

For a quantum individual $\mathbf{q}$ with m qubits, the probability of measuring a specific binary string $\mathbf{b}$ is calculated as:

$$P(\mathbf{b}) = \prod_{i=0}^{m-1} P(|b_i\rangle)(\theta_i), \quad b_i \in \{0, 1\} \tag{4.13}$$

where $P(|0\rangle)(\theta_i)$ and $P(|1\rangle)(\theta_i)$ are calculated according to Equations (4.11) and (4.12).

Based on what we discussed in Chapter 2, when a measurement is performed on a real qubit of an actual quantum computer, the wave function collapses. This means that the qubit ceases to be in superposition. Its quantum state has now degenerated into one of the two basic quantum states, $|0\rangle$ and $|1\rangle$.

However, in QIEA the information about the quantum state of our individual is useful even after measurement. By repeatedly taking measurements of a quantum individual, we can evaluate whether the region of the search space on which the individual is focused yields worthwhile solutions or whether we need to readjust the angles in order to be guided toward more promising areas of the design space.

It is now evident that QIEA is not designed to run on quantum hardware. Instead, it is an algorithm intended for classical computers that incorporates quantum-inspired principles in its operation.

4.2.4 Visualization of Quantum Individual

An example is provided in order to make the nature of the representation of the quantum individual more comprehensible.

We consider a problem with 2 design variables, each of which is encoded with 5 qubits. The search space of the problem is a 2D plane with $2^5 = 32$ discrete values for each variable. Therefore, there are $32 \times 32 = 1024$ candidate solutions in total.

Let's assume the following 3 different individuals for this specific encoding with the following values for the qubit angles.

For each of the above individuals, we will calculate the probability of measuring each candidate solution and represent these probabilities in diagrams. In the diagrams presented below, the x and y axes each represent one of the design variables of the problem.

Qubit Angle	Individual 1	Individual 2	Individual 3
θ_0	0	0.0000	-0.3490
θ_1	0	$-\pi/2$	-0.0729
θ_2	0	$-\pi/2$	-1.5708
θ_3	0	$-\pi/2$	0.4554
θ_4	0	$-\pi/2$	-0.3666
θ_5	0	0.0000	-0.9258
θ_6	0	$\pi/2$	1.1986
θ_7	0	$\pi/2$	0.3257
θ_8	0	$\pi/2$	-0.1874
θ_9	0	$\pi/2$	0.6692

Table 4.1: Example Individuals

Therefore, the xy-plane constitutes the design space, and the points on the plane are the candidate solutions. Because we use binary encoding, the candidate solutions are discrete. The color scale indicates the probability of measuring each candidate solution by each quantum individual. The results are presented in the Figure 4.2.

Individual 1 has all angles set to $\theta = 0$. Based on the above, this corresponds to a 50–50 probability of measuring 0 or 1 for each qubit. Therefore, this is the Hadamard state. As can also be seen in the diagram, all points in the search space have the same measurement probability, $P = 1/1024 = 9.76562 \times 10^{-4}$.

For individual 2, it appears that all angles except 2 have converged to values of $\frac{\pi}{2}$ or $-\frac{\pi}{2}$. For these converged qubits, the probability of measuring 1 or 0 respectively is equal to 100%. The θ_0 and θ_5 are equal to 0, which means there is a 50–50 probability of measuring 0 or 1 for each of them. Therefore, for individual 2 we have 4 candidate solutions, which can be measured with equal probability. This is reflected in the diagram.

For individual 3, we observe a more disordered arrangement of angles, whose probability distribution is also shown in the diagram.

4.3 Implementation of Single-Objective QIEA

In this paragraph the basic components of QIEA algorithm will be described as well as the complete sequence of algorithmic steps.

The current implementation of the QIEA uses three populations of individuals, similarly to the classical evolutionary algorithm: one for the parents, one for the offspring, and one for the elite.

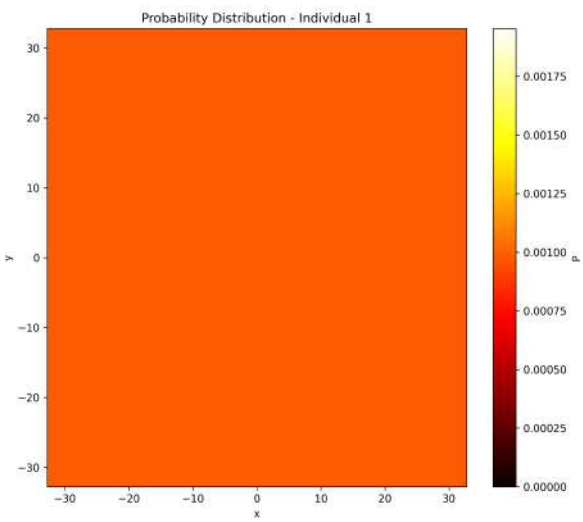
4.3.1 Measurement & Evaluation

Measurement allows us to acquire a binary representation of a candidate solution from a quantum individual.

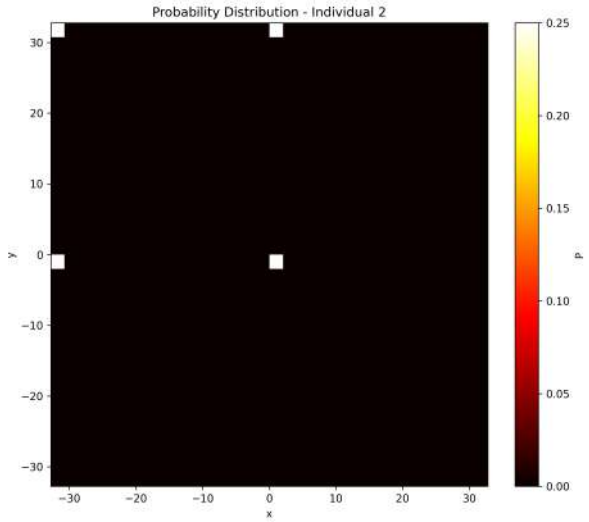
For each individual q_j^g at each generation g, the measurement process is performed independently for each qubit i:

1. **Probability Calculation**: The measurement probability for state $|0\rangle$ is computed according to Eq. (4.11):

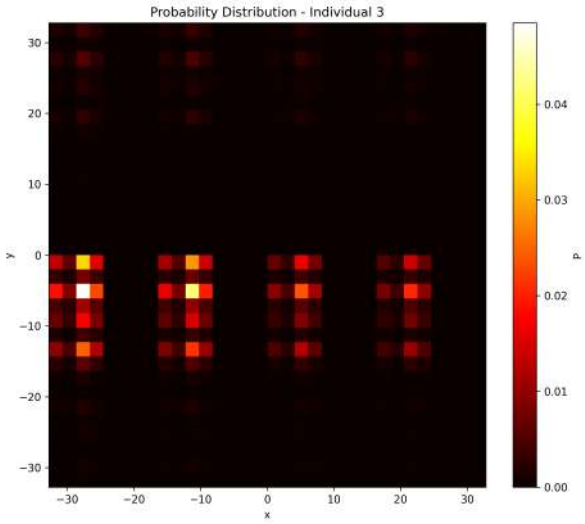
$$p_{j,i}^g = P(|0\rangle)_{j,i}^g = \frac{1}{2} - \frac{1}{2}\sin(\theta_{j,i}^g)$$



(a) Individual 1



(b) Individual 2



(c) Individual 3

Figure 4.2: Probability Distributions

2. **Stochastic Collapse**: A random number $r_{j,i} \sim \mathcal{U}(0,1)$ determines the measurement outcome:

$$b_{j,i}^g = \begin{cases} 0 & \text{if } r_{j,i} \leq p_{j,i} \\ 1 & \text{if } r_{j,i} > p_{j,i} \end{cases} \tag{4.14}$$

After performing the measurement, we can decode the obtained binary candidate solution according to Eq. (4.8) and evaluate it. The result of the procedure outlined above is depicted in the following diagram:

$$\mathbf{q}_j^g \xrightarrow{\text{measurement}} \mathbf{b}_j^g \xrightarrow{\text{decoding}} \mathbf{x}_j^g \xrightarrow{\text{evaluation}} F(\mathbf{x}_j^g)$$

Given that the individual $\mathbf{q}_j^g$ represents a probability distribution of the candidate binary solutions, it is clear that if we repeat the operation of measurement on the same individual, we will obtain different binary representations each time. Also as explained earlier, although a quantum state collapses after a measurement in a real quantum system,

in QIEA we retain the quantum state after measurement. This allows us to perform multiple measurements on a single individual during each measurement phase.

For each individual $\mathbf{q}_j^g$ at generation g, the process of measurement and evaluation is repeated *mpi* (measurements per individual) times, producing *mpi* independent solutions:

$$X_j^g = \{\mathbf{x}_j^{g,1}, \mathbf{x}_j^{g,2}, \ldots, \mathbf{x}_j^{g,mpi}\} \tag{4.15}$$

Among these *mpi* measurements, the one with the best fitness is selected as the individual's latest measurement, which is then used for the comparison and rotation operations which we will describe in the following sections.

$$\mathbf{x}_{\text{latest},j}^g = \arg\min_{\mathbf{x} \in X_j^g} F(\mathbf{x}) \tag{4.16}$$

The number *mpi* of measurements per individual is parameterized by the user.

For each individual q, in addition to the qubit angles, we also store the most recent solution $\mathbf{x}_{latest,j}^g$ and the best solution found so far $\mathbf{x}_{best,j}^g$. Clearly, for solutions that have been evaluated, the binary encodings $\mathbf{b}_{best,j}^g$ and $\mathbf{b}_{latest,j}^g$ are likewise available. The encodings are just as important to us as the quantum angles and the solutions themselves, for reasons that will be clarified later. Therefore, for simplicity, throughout the remainder of the description, we will assume that whenever a solution x is stored, its associated binary encoding b is stored as well, and any mention of this will be omitted.

4.3.2 Initialization

The algorithm begins by initializing the population of n quantum individuals. Each individual $\mathbf{q}_j$ consists of m qubits, where m is the total number of bits required to encode all design variables.

All qubits are initialized with angular parameter $\theta_{j,i}^0 = 0$:

$$\theta_{j,i}^0 = 0 \quad \forall j \in \{1, 2, \ldots, n\}, \quad \forall i \in \{0, 1, \ldots, m-1\}$$

This initialization establishes the Hadamard state for each qubit, where the measurement probabilities are equal:

$$P(|0\rangle)_{j,i}^0 = P(|1\rangle)_{j,i}^0 = \frac{1}{2}$$

corresponding to the maximum superposition quantum state:

$$|\psi\rangle_{\text{initial}} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

The role of the Hadamard gate in the initialization phase closely resembles how we set up the initial population in a classical evolutionary algorithm. In each case, the starting candidate solutions are selected randomly.

During initialization, each individual performs its first *mpi* measurements and evaluations. The fittest solutions of each individual are stored as the latest and best-so-far solutions $\mathbf{x}_{latest}^0$ and $\mathbf{x}_{best}^0$.

The global best solution $\mathbf{x}_{global\_best}^0$ is identified as the best among all solutions of all individuals and also gets stored and updated during the algorithmic procedure.

4.3.3 Q-Gate Rotation

After measurement and evaluation, the current solution $\mathbf{x}^g_{latest,j}$ is compared with the individual's best-so-far solution $\mathbf{x}^{g-1}_{best,j}$. This comparison determines whether improvement occurred, which in turn guides the rotation strategy for updating the quantum chromosome.

For each qubit i of individual j, the rotation angle increment $\Delta\theta_{j,i}$ is determined from a configurable rotation strategy table. The table maps three inputs to a rotation direction: the measured bit value $b^g_{latest,j,i}$, the best so far solution's bit value $b^{g-1}_{best,j,i}$, and whether improvement occurred ($F(\mathbf{x}^g_{latest,j}) \leq F(\mathbf{x}^{g-1}_{best,j})$). In literature the rotation strategy shown in Table 4.2 is commonly used.

$b^g_{latest,j,i}$	$b^{g-1}_{best,j,i}$	$F(\mathbf{x}^g_{latest,j}) \leq F(\mathbf{x}^{g-1}_{best,j})$	$\Delta\theta_{j,i}$
0	0	True	0
0	1	True	0
1	0	True	0
1	1	True	0
0	0	False	0
0	1	False	$+\delta$
1	0	False	$-\delta$
1	1	False	0

Table 4.2: Standard rotation strategy table for QIEA

The table's values are interpreted as follows. When improvement occurs, no rotation is applied ($\Delta\theta_{j,i} = 0$). When deterioration occurs, rotation increases the probability of the best solution's bit value: positive rotation ($+\delta$) if $b^{t-1}_{j,i} = 1$, negative rotation ($-\delta$) if $b^{t-1}_{j,i} = 0$, or no rotation if $x^t_{j,i} = b^{t-1}_{j,i}$. The magnitude δ is the base rotation angle and is parameterized during the algorithm's configuration.

$b^g_{latest,j,i}$	$b^{g-1}_{best,j,i}$	$F(\mathbf{x}^g_{latest,j}) \leq F(\mathbf{x}^{g-1}_{best,j})$	$\Delta\theta_{j,i}$
0	0	True	a_1
0	1	True	a_2
1	0	True	a_3
1	1	True	a_4
0	0	False	a_5
0	1	False	a_6
1	0	False	a_7
1	1	False	a_8

Table 4.3: Configurable rotation strategy table for QIEA

In this work, we examined to what extent different choices for the rotation angles can help improve convergence. For this purpose, the table above was parameterized as shown in Table 4.3. In the parameterization we use, each of the 8 entries that result from the above table is allowed to take one of the following values:

- a fixed number (typically 1, but in principle any constant),
- a random integer in the range $[-1, 1]$, (for every gate operation we call a random number generator)

- a uniformly distributed random real number in the range $[-1, 1]$, (for every gate operation we call a random number generator)

In addition to the 8 parameters of the rotation table, we also introduce an angular parameter $\Delta\theta$, which is multiplied by each of the table entries described above.

Once the angle $\Delta\theta_{j,i} = a_c \cdot \Delta\theta$ is determined according to which row of the table describes each qubit's condition during each gate operation, the qubit's angular parameter is updated as follows:

$$\theta_{j,i}^{t} \leftarrow \theta_{j,i}^{t-1} + \Delta\theta_{j,i}, \theta_{j,i} \in [-\theta_{\max}, +\theta_{\max}] \tag{4.17}$$

The bounds $[-\theta_{\max}, +\theta_{\max}]$ of each qubit's angle $\theta_{j,i}$ are also configurable. A qubit angle equal to $\pi/2$ or $-\pi/2$ can cause the algorithm to converge prematurely to local optima, reducing its ability to explore other regions of the search space. To mitigate this effect, we introduce bounds on the angle θ.

Q-Gate rotation is the fundamental evolutionary operator of QIEA. Its purpose is to guide individuals toward regions of the search space that appear promising and to move them away from regions that appear barren. Furthermore, by parameterizing the values of the rotation table and the angular parameter, we can adjust the balance between exploration and exploitation.

A direct analogy to some operator or function of the classical evolutionary algorithm is difficult to identify.

4.3.4 Quantum Crossover

The crossover operator adapts the evolutionary algorithm recombination to the quantum chromosome representation, exchanging angular parameters between parent individuals to create offspring that inherit probability distribution characteristics from both parents. The implementation uses single-point crossover.

Two parents are first selected via tournament selection, where each parent is the best individual from a randomly selected subset of t individuals. Crossover is performed with probability p_x. A random integer $c \in \{1, 2, \ldots, k-1\}$ is chosen, where c denotes the crossover point and k represents the total number of design variables. The first c variables are assigned to the first offspring, while the remaining $k - c$ variables are assigned to the second offspring. The chromosome-level crossover point c' is defined as

$$c' = \sum_{\ell=1}^{c} m_\ell, \tag{4.18}$$

where m_ℓ denotes the number of bits allocated to the ℓ-th design variable.

$$\boldsymbol{q}_{\text{off}}^{(1)} = [\theta_0^{(1)}, \ldots, \theta_{c'-1}^{(1)}, \theta_{c'}^{(2)}, \ldots, \theta_{m-1}^{(2)}] \tag{4.19}$$

$$\boldsymbol{q}_{\text{off}}^{(2)} = [\theta_0^{(2)}, \ldots, \theta_{c'-1}^{(2)}, \theta_{c'}^{(1)}, \ldots, \theta_{m-1}^{(1)}] \tag{4.20}$$

If crossover is not performed (with probability $1 - p_x$), the parents are simply copied to the offspring population.

It is noted that, in order to be able to apply the Q-Gate rotation operator, we need an existing solution with which we can compare the solutions that will result from the new measurements. Therefore, after applying the crossover operator, we can either take

a new measurement from the new chromosome that was produced and evaluate it, or we can make use of the best solutions of the parents obtained so far. After trial-and-error testing, the following implementation was chosen: for each of the offspring, we randomly select with a 50–50 probability the best solution so far of one of the parents.

The crossover operation is performed at regular intervals of T_x generations, providing sufficient time for Q-gate rotation to guide the solutions toward more promising regions.

Compared to the crossover operator of the classical evolutionary algorithm, the quantum crossover does not differ significantly. Both of them attempt, in a very similar way, to exploit the positive traits from two or more individuals. The difference between the two is that the former operates at the bit level, combining two or more specific solutions, whereas the quantum version operates at the qubit level and combines features of probability distributions of candidate solutions.

4.3.5 Quantum Mutation

Two mutation mechanisms introduce stochastic perturbations at different stages of the evolution process, enhancing exploration and preventing premature convergence. The operation and purpose of the operator do not differ significantly from the mutation in the classical evolutionary algorithm. In QIEA, we can apply mutation either at the bit level or at the qubit level, each of which takes place at different stages of the algorithm.

Qubit Angle Mutation During the application of the Q-Gate operator the set of $\Delta\theta_{j,i}$ is determined for each qubit of an individual. Before adjusting the qubit angle as previously described, a random number $r_{j,i} \sim \mathcal{U}(0,1)$, is used to decide whether the angle will be updated in the standard way or flipped instead, according to the mutation probability set as configuration parameter.

$$\theta_{j,i}^{t} \leftarrow \begin{cases} -(\theta_{j,i}^{t-1} + \Delta\theta_{j,i}) & \text{with probability } p_m^{\text{rot}} \\ \theta_{j,i}^{t-1} + \Delta\theta_{j,i} & \text{with probability } 1 - p_m^{\text{rot}} \end{cases} \tag{4.21}$$

This operates at the quantum chromosome level, affecting the probability distribution for all future measurements from that qubit.

Measurement Mutation After each bit is measured but before fitness evaluation, measurement mutation may occur with a user defined probability p_m^{meas}. When mutation occurs, the bit value is flipped:

$$b_{j,i}^{t,\text{final}} = \begin{cases} 1 - b_{j,i}^{t} & \text{with probability } p_m^{\text{meas}} \\ b_{j,i}^{t} & \text{with probability } 1 - p_m^{\text{meas}} \end{cases} \tag{4.22}$$

This operates at the classical solution level, modifying only the current measurement without affecting the quantum chromosome.

4.3.6 Migration

Migration mechanisms facilitate information transfer within and across the population by replacing individuals' best known solutions with better solutions discovered elsewhere. Two distinct strategies are implemented with different spatial scopes.

Local Migration Local migration operates within neighborhoods. The population is partitioned into non-overlapping neighborhoods of user-defined size. Within each neighborhood, the locally best solution is identified and replaces each individual's best-so-far solution $\mathbf{x}^g_{best,j}$. This operation occurs every T_{local} generations promoting local convergence while allowing different neighborhoods to explore different regions semi-independently.

Global Migration Global migration broadcasts the globally best solution $\mathbf{x}^0_{global\_best}$ to all individuals in the population. This occurs every T_{global} generations and has a strong convergence effect, directing the entire population toward the best known solution.

Overwriting the individual's best-so-far solution affects the outcome of the Q-Gate operation on the specific individual for the next generations.

Migration operation resembles the distributed EAs (DEA). In both QIEA and DEA, information is exchanged between groups of the population. However, this exchange of information occurs in a different way. While in DEA individuals move from one neighborhood to another, in QIEA individuals do not move. Instead, communication takes place via the individual's best-so-far solution, which guides the search toward the region where the fittest solution is observed, either within the neighborhood, in case of local migration, or within the entire population, in case of global migration.

4.3.7 Elites Selection

As in the standard evolutionary algorithm, so also in QIEA, at the end of each generation the individuals of the offspring and the current elite populations are ranked, and the e fittest among them form the updated elite population. The individuals of the elite population on the one hand constitute the best solutions that have been found at any given time during the algorithmic process, and on the other hand, can be selected as parents for the crossover operations.

4.3.8 Elitism

Elitism preserves high-quality solutions by injecting elite individuals into the parent population. With probability p_{elite}, a subset of the elite population is included in parent selection. The number of elite individuals included in the parent population is given by multiplying the size of the parent population by the elitism rate r_{elite}. These n_{elite} individuals are randomly sampled from the elite population.

4.3.9 Parent Selection

Parent selection determines which individuals participate in crossover to produce the next generation. Parents are selected from the combined pool of the previous parent population and the current offspring population. This is the same standard procedure that is followed in the EA as well.

In the current implementation two selection methods are available, specified as a configuration parameter:

Tournament Selection For each required parent, a tournament of size k is conducted. The method has already been explained in the Quantum Crossover paragraph.

Linear Weighted Selection Individuals are selected with probability proportional to linearly decreasing weights based on their fitness rank. The combined pool is sorted by fitness as $\{\mathbf{q}_1^g, \mathbf{q}_2^g, \ldots, \mathbf{q}_N^g\}$ where $F(\mathbf{x}_{best,1}^{g-1}) \leq F(\mathbf{x}_{best,2}^{g-1}) \leq \cdots$. Each individual q_i is assigned a weight:

$$w_i = N - i + 1 \tag{4.23}$$

The selection probability for an individual q_i is then:

$$p_i = \frac{N - i + 1}{\sum_{j=1}^{N}(N - j + 1)} \tag{4.24}$$

Once an individual is selected, it is removed from the pool and the weights are recalculated for subsequent selections. This is done to promote greater diversity among the resulting crossover offspring.

4.3.10 Stagnation Mechanism

To prevent premature convergence and enhance exploration when the algorithm becomes trapped in local optima, a stagnation detection and recovery mechanism is implemented. Stagnation is monitored by tracking the number of fitness evaluations elapsed since the last improvement in the best fitness value. Specifically, at each generation, the current generation's best fitness is compared against the previously recorded best. If an improvement is detected, the stagnation counter is reset. Otherwise, it accumulates the number of evaluations performed without progress. When the stagnation counter exceeds a predefined threshold N_{stag}, stagnation is declared and recovery procedures are initiated.

Upon detecting stagnation, the algorithm enters a recovery phase consisting of two sequential actions applied over a limited number of generations. First, immediately after stagnation detection, a partial renewal of the parent population occurs. Elitism is temporarily disabled, and approximately half of the parent population is replaced by newly initialized individuals, while the remaining half is selected from the existing parent and offspring populations via the standard selection mechanism. This ensures a substantial injection of diversity. Second, during the subsequent generations, crossover operations are enforced, further promoting genetic mixing and exploration. After the recovery phase concludes, the algorithm resumes standard operation with the stagnation counter reset, allowing the process to continue from a diversified state.

4.3.11 QIEA Workflow

The complete single-objective QIEA executes the following sequence of operations:

1. **Initialization**:
   (a) Initialize offspring population with all qubits at $\theta_{j,i}^0 = 0$
   (b) Perform initial measurement and evaluation for each individual
   (c) Set each individual's best solution to its initial measurement
   (d) Identify global best solution $\mathbf{x}_{global\_best}^0$
2. **Main Loop**:

(a) Detect stagnation and set recovery flags if threshold N_{stag} exceeded

(b) Perform parent selection:

i. Include $\mu \cdot r_{elite}$ elite individuals with probability p_{elite} (unless in recovery mode)

ii. If stagnation recovery is active, $\mu/2$ new individuals are initialized

iii. Select remaining parents via tournament or linear weighted selection

(c) Apply quantum crossover (if $g \bmod T_{\text{crossover}} = 0$ or after stagnation detection):

i. Select parent pairs via tournament selection

ii. Perform single-point crossover with probability p_x or clone parent

iii. Assign offspring best solution randomly from one parent

(d) Evolve offspring population:

i. Perform *mpi* measurements per individual (with measurement mutation)

ii. Select fittest among the new *mpi* measurement as latest x_{latest}

iii. Apply Q-gate rotation based on comparison with best solution x_{best}

iv. Apply rotation mutation with probability p_x^{rot}

v. Update individual's best solution x_{best} if improvement occurred

(e) Update global best solution $\mathbf{x}^g_{global\_best}$

(f) Apply migration (if scheduled):

i. Global migration if $g \bmod T_{\text{global}} = 0$

ii. Local migration if $g \bmod T_{\text{local}} = 0$

(g) Perform elite selection from combined offspring and elite populations

(h) Update stagnation counter based on cycle best fitness

3. **Termination**: Output final elite population and global best solution

4.4 Multi-Objective QIEA

Multi-objective minimization problem and Pareto dominance were formulated in Chapter 3 and were described by Equations (3.4) and (3.5)

In this section, we describe how the QIEA algorithm is adapted to address multi-objective optimization problems.

4.4.1 Challenges in Quantum Gate Application

In single-objective QIEA, the application of the Q-Gate is determined by the condition $F(x) \leq F(b)$. In multi-objective problems, however, three possible states arise when comparing a new solution $\mathbf{x}^g_{\text{latest},j}$ with the best-known solution $\mathbf{x}^g_{\text{best},j}$:

1. $\mathbf{x}^g_{\text{latest},j} \prec \mathbf{x}^g_{\text{best},j}$: The new solution dominates (improvement)
2. $\mathbf{x}^g_{\text{best},j} \prec \mathbf{x}^g_{\text{latest},j}$: The best-so-far solution dominates (deterioration)
3. $\mathbf{x}^g_{\text{latest},j} \not\sim \mathbf{x}^g_{\text{best},j}$: The solutions are non-comparable (uncertainty)

The third case constitutes the main challenge, as it is not clear whether the quantum gate should be applied for individual evolution. To address the uncertainty in non-comparable solutions, Multi-Objective QIEA uses the following approach. When two solutions $\mathbf{x}^g_{\text{best},j}$ and $\mathbf{x}^g_{\text{latest},j}$ are non-comparable, the corresponding individual $\mathbf{q}^g_j$ is duplicated:

$$\mathbf{q}^g_j \to \{\mathbf{q}^{g,1}_j, \mathbf{q}^{g,2}_j\}$$

where $\mathbf{q}^{g,1}_j = \mathbf{q}^{g,2}_j = \mathbf{q}^g_j$ (identical copies). Then the two copies evolve with different assumptions:

- **Individual $\mathbf{q}^{g,1}_j$**: Evolves assuming that $\mathbf{x}^g_{\text{latest},j} \prec \mathbf{x}^g_{\text{best},j}$ (improvement)
  - Quantum gate is applied using the improvement entries of the rotation table
  - The new solution $\mathbf{x}^g_{\text{latest},j}$ replaces $\mathbf{x}^g_{\text{best},j}$
- **Individual $\mathbf{q}^{g,2}_j$**: Evolves assuming that $\mathbf{x}^g_{\text{best},j} \prec \mathbf{x}^g_{\text{latest},j}$ (deterioration)
  - Quantum gate is applied using the deterioration entries of the rotation table
  - The old solution $\mathbf{x}^g_{\text{best},j}$ is maintained

4.4.2 NSGA-II Selection Algorithm

After the evolution phase, the population size may exceed the original limit n due to individual duplication. To maintain the size, the **NSGA-II** (Non-dominated Sorting Genetic Algorithm II) algorithm is applied [23]:

Phase 1: Non-dominated Sorting All solutions are sorted into levels $(L_1, L_2, \ldots)$ where:

- L_1: First front (non-dominated solutions)
- L_2: Second front (solutions dominated only by L_1)
- $\vdots$

Phase 2: Crowding Distance Calculation For each solution, the **crowding distance** is calculated:

$$d_i = \sum_{m=1}^{p} \frac{f_m^{i+1} - f_m^{i-1}}{f_m^{\max} - f_m^{\min}}$$

where f_m^{i+1} and f_m^{i-1} are the neighboring solutions of the i-th solution in the m-th objective.

Phase 3: Survival Selection Selection of the n best individuals is done with the criteria:

1. **Primary criterion**: Lower front level (better L_i)
2. **Secondary criterion**: Larger crowding distance (better diversity)

4.4.3 Complete Algorithm Description

The multi-objective QIEA shares several building blocks with its single-objective counterpart, such as initialization, measurement, Q-gate operations, crossover, and mutation. However, certain procedures differ because of the specific characteristics of multi-objective optimization.

To maintain conceptual and implementation simplicity, the multi-objective variant uses only a single population. This choice is justified because, through the NSGA-II–based sorting procedure, elite individuals are always available without the need for a dedicated elite population. The use of NSGA-II also promotes diversity by preferring solutions that are well distributed in the objective space. Consequently, maintaining separate parent and elite populations is redundant. We also evaluated a variant that explicitly stored parent and elite populations for the multi-objective case and observed no improvement in performance.

Likewise, mechanisms such as global best solution storage and migrations are not meaningful in the multi-objective context, since there is no unique globally optimal solution. A stagnation mechanism is also unnecessary. Under the Pareto dominance criterion used to compare candidate solutions, it is highly improbable that the best individuals will remain unchanged over time.

Algorithm Pseudocode

1. **Initialization**:
   (a) Initialize offspring population with all qubits at $\theta^0_{j,i} = 0$
   (b) Perform initial measurement and evaluation for each individual
   (c) Set each individual's best solution to its initial measurement
2. **Main Loop**: While termination criterion is not satisfied:
   (a) Apply quantum crossover (if $g \bmod T_x = 0$):
      i. Select parent pairs via tournament selection
      ii. Perform single-point crossover with probability p_x or clone parent
      iii. Perform an initial measurement for the new offspring individuals
   (b) **Dominance-Based Evolution**: For each individual $\mathbf{q}^g_j$ and each measurement $\mathbf{x}^g_{\text{latest},j}$, compare with $\mathbf{x}^g_{\text{best},j}$ and evolve according to the dominance relation:
      - If $\mathbf{x}^g_{\text{latest},j} \prec \mathbf{x}^g_{\text{best},j}$ (improvement): Apply quantum gate using improvement entries of the rotation table; update $\mathbf{x}^g_{\text{best},j} \leftarrow \mathbf{x}^g_{\text{latest},j}$; return single individual
      - If $\mathbf{x}^g_{\text{best},j} \prec \mathbf{x}^g_{\text{latest},j}$ (deterioration): Apply quantum gate using deterioration entries of the rotation table; maintain $\mathbf{x}^g_{\text{best},j}$; return single individual
      - If $\mathbf{x}^g_{\text{latest},j} \not\prec \mathbf{x}^g_{\text{best},j}$ (non-comparable): Duplicate individual into $\{\mathbf{q}^{g,1}_j, \mathbf{q}^{g,2}_j\}$; for $\mathbf{q}^{g,1}_j$ apply quantum gate using improvement entries and set $\mathbf{x}^g_{\text{best},j} \leftarrow \mathbf{x}^g_{\text{latest},j}$; for $\mathbf{q}^{g,2}_j$ apply quantum gate using deterioration entries and maintain $\mathbf{x}^g_{\text{best},j}$; return both individuals
   (c) **Population Assembly**: Collect all evolved individuals into the new population

(d) **Duplicate Elimination**: Remove individuals with identical chromosomes

(e) **NSGA-II Selection**: Perform non-dominated sorting

(f) **Eliminate overpopulation** Removing the less fit individuals that exceed the population size

3. **Return**: Pareto front extracted from the final population

Chapter 5

Metamodel Assisted QIEA - MAQIEA

5.1 Surrogate Models in Evolutionary Optimization

Metamodels, also referred to as surrogate models, constitute approximate representations of complex optimization functions. Their primary purpose is to reduce computational cost during the evolutionary process by replacing expensive exact evaluations with significantly more economical approximations. This approach is particularly beneficial in engineering applications where the evaluation of the objective function requires extensive computational resources, such as computational fluid dynamics (CFD) simulations in aerospace design, finite element method (FEM) analysis in structural mechanics, and performance evaluation in turbomachinery systems.

In optimization contexts where a single function evaluation may require minutes to hours of computation, the cumulative cost of thousands of evaluations becomes prohibitive. Surrogate models address this challenge by learning the underlying input-output mapping from a limited number of exact evaluations and subsequently providing rapid approximations for new candidate solutions [24].

5.1.1 Local versus Global Metamodeling

Metamodel construction strategies can be classified into global and local approaches. Global metamodels are trained using all available data and aim to approximate the objective function over the entire design space. While conceptually simpler, global models may exhibit poor accuracy in problems where the objective function has significantly different characteristics in different regions, such as varying smoothness, multimodality, or the presence of discontinuities.

Local metamodeling addresses these limitations by constructing a specialized model for each new candidate solution using only data points in its vicinity. This approach offers several advantages:

- **Adaptive accuracy**: The model adapts to local function behavior, providing higher fidelity in regions where the function exhibits specific patterns or anomalies
- **Computational efficiency**: Training complexity scales with the number of local points rather than the entire database, which is particularly important for high-dimensional problems
- **Robustness**: Local models are less affected by distant outliers or regions with fundamentally different function characteristics

The local metamodeling strategy employed in this work constructs a new radial-basis function network for each candidate solution, using only the nearest points from the database. The selection of these training patterns is performed using a Minimum Spanning Tree (MST) algorithm, which ensures topological compactness and excludes extreme observations that could degrade model quality.

5.1.2 Low-Cost Pre-Evaluation Technique

The Low-Cost Pre-Evaluation (LCPE) [25] technique is a selective sampling strategy that combines surrogate-based screening with targeted exact evaluation. The fundamental principle of LCPE is to evaluate all candidate solutions using a computationally inexpensive metamodel and then re-evaluate only the most promising individuals using the exact objective function.

The LCPE technique is typically activated after an initial phase of pure exact evaluations. This preliminary phase, usually spanning several generations, serves to construct a database of sufficient size and quality to train the metamodel. The number of initial exact evaluations must be adequate to capture the essential characteristics of the objective function landscape, particularly in regions relevant to the optimization search.

The implementation of LCPE involves the following procedural steps:

1. Construction of a metamodel using all available exact evaluations stored in the database
2. Approximate evaluation of new candidate solutions using the trained metamodel
3. Selection of a subset of candidates for exact evaluation based on predicted fitness or other criteria
4. Exact evaluation of selected candidates and updating of the database
5. Periodic retraining of the metamodel as additional exact evaluations accumulate

The effectiveness of LCPE depends critically on the quality of the surrogate model and the selection strategy for exact evaluations. An overly conservative strategy (too many exact evaluations) diminishes computational savings, while an overly aggressive strategy (too few exact evaluations) risks convergence to suboptimal solutions due to metamodel inaccuracies.

5.1.3 Challenges in Multi-Objective Optimization

The efficiency of LCPE-based strategies decreases significantly when extended to multi-objective optimization. This degradation stems from fundamental differences in how single-objective and multi-objective algorithms distribute computational effort:

1. **Solution distribution**: In single-objective problems, the population progressively converges toward a single optimal solution, creating a dense concentration of evaluations in a localized region. This concentration facilitates the training of accurate local metamodels. Conversely, in multi-objective problems, the population must maintain diversity along the entire Pareto front, distributing evaluations across a potentially extensive manifold in objective space.

2. **Dominance sensitivity**: Multi-objective selection relies on dominance comparisons, which are sensitive to prediction errors. A small error in a single objective value may alter the dominance relationship between two solutions, potentially leading to incorrect selection decisions. In contrast, single-objective optimization is more tolerant to small errors, as ranking is based on scalar fitness differences.

3. **Front quality degradation**: Inaccurate metamodel predictions can introduce spurious non-dominated solutions or exclude genuinely non-dominated solutions. This directly impacts both the convergence and diversity characteristics of the approximated Pareto front.

These challenges necessitate careful adaptation of the LCPE technique for multi-objective optimization, including modified triggering criteria for exact evaluation and specialized handling of dominance-based comparisons, as described in subsequent sections.

5.2 Metamodel-Assisted QIEA

The metamodel-based assistance was developed for the single-objective QIEA, since extending it to multi-objective problems proved challenging.

The metamodel-assisted single-objective QIEA follows the same procedure as the non-assisted variant, with a modification in the measurement phase to reduce computational cost. For each individual in the population, instead of performing m exact evaluations, the algorithm generates m candidate solutions whose fitness is estimated using the surrogate model. Among these m surrogate-evaluated candidates, the one with the best predicted fitness score is selected. This selected candidate is then evaluated using the exact objective function, and the standard evolution operations (quantum gate rotation) are applied based on the exact fitness value. The surrogate model is activated only after a minimum number of exact evaluations have been collected in the database. Before this threshold, the algorithm operates without the surrogate model.

5.3 Metamodel Implementation Details

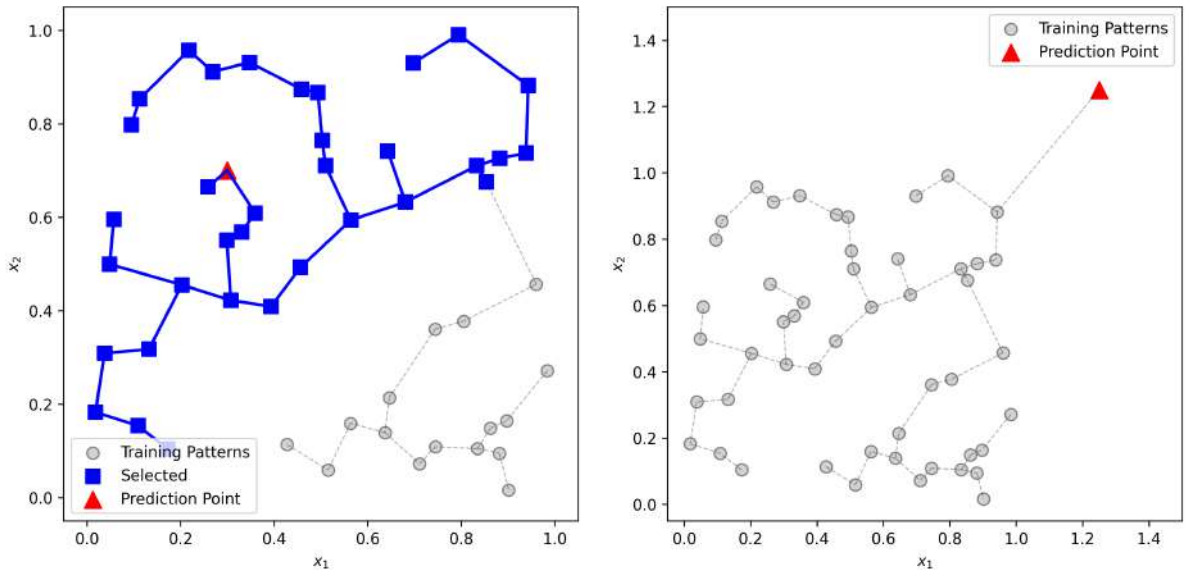
This section describes the technical components underlying the metamodel-assisted QIEA, including pattern selection strategies, radial-basis function network architecture, and adaptive training procedures.

5.3.1 Local Pattern Selection Using Minimum Spanning Trees

For each new candidate solution requiring metamodel evaluation, a local training set is constructed from the database. The selection process employs a Minimum Spanning Tree (MST) algorithm to identify patterns that form a compact topological structure around the candidate.

The MST-based pattern selection procedure consists of the following steps:

1. **Nearest neighbor retrieval**: Identify the 100–150 nearest points to the candidate solution based on Euclidean distance in the design variable space. This initial pool provides a coarse neighborhood from which the final training set will be refined.



(a) MST structure and selected training patterns

(b) MST-based identification of outlier patterns

Figure 5.1: Minimum spanning tree (MST) for training pattern selection and outlier detection

2. **MST construction**: Construct the Minimum Spanning Tree connecting all retrieved points using standard algorithms such as Kruskal's or Prim's algorithm. The MST is a graph-theoretic structure that connects all points with the minimum total edge length, ensuring efficient connectivity without redundant connections.

3. **Tree traversal and selection**: Starting from the candidate solution, traverse the MST and incrementally add points to the training set (see Figure 5.1a). A point connected by branch i is included if:

$$L_i < \bar{L} + k \cdot \sigma_L \tag{5.1}$$

where L_i is the length of branch i, $\bar{L}$ is the mean branch length computed over branches already included in the training set, σ_L is the corresponding standard deviation, and k is a tolerance parameter (typically $k \approx 2$).

4. **Iterative statistics update**: As each new point is added to the training set, the statistics $\bar{L}$ and σ_L are recomputed to reflect the current topological structure. This adaptive criterion prevents the inclusion of distant outliers while allowing the training set to expand naturally in directions where data density is higher.

The final training set consists of points that form a topologically compact cluster around the candidate solution. It is ensured that way that the metamodel is trained primarily on data relevant to the local function behavior, excluding observations from distant regions that may not be representative.

5.3.2 Outlier Detection and Reliability Assessment

Not all regions of the design space are equally well-represented in the evaluation database. Candidates located in sparsely sampled regions are at higher risk of inaccurate metamodel

prediction. To identify such outlier candidates, a distance-based reliability criterion is employed (Figure 5.1b).

After the MST-based pattern selection process, the candidate's distance to its k nearest training patterns (typically $k = 2$) is computed. The candidate is considered reliable for metamodel evaluation if:

$$d_k < \bar{L} + m \cdot \sigma_L \tag{5.2}$$

where d_k is the distance to the k-th nearest training pattern, and $m \approx 3$ is a conservativeness parameter. Both $\bar{L}$ and σ_L are statistics derived from the MST pattern selection process described above.

The choice of $k = 2$ (second nearest neighbor) provides robustness against the presence of a single accidentally close point, which might give a misleading impression of data density. For candidates identified as outliers, exact evaluation is preferred over metamodel prediction, as approximations in regions with sparse data can introduce significant errors that negatively impact algorithm convergence.

5.3.3 Radial-Basis Function Networks

Radial-basis function networks (RBFNs) were selected as the metamodel class due to their ability to provide local approximations and the simplicity of their training procedure.

Network Architecture

An RBFN consists of three distinct layers (Figure 5.2):

- **Input layer**: Receives the design variable vector $\mathbf{x} \in \mathbb{R}^N$ without performing any transformation.
- **Hidden layer**: Contains M neurons, each implementing a radial basis function centered at $\mathbf{c}_m \in \mathbb{R}^N$ with radius $r_m > 0$.
- **Output layer**: Computes the network output as a weighted linear combination of hidden layer activations.

The output of the network for input vector $\mathbf{x}$ is given by:

$$y(\mathbf{x}) = \sum_{m=1}^{M} \psi_m \, G_m(\|\mathbf{x} - \mathbf{c}_m\|) \tag{5.3}$$

where $\psi_m \in \mathbb{R}$ are the synaptic weights connecting the hidden layer to the output layer, $\mathbf{c}_m$ are the RBF centers, and $G_m : \mathbb{R}_{\geq 0} \to \mathbb{R}$ are the radial basis functions. The Gaussian function is commonly employed:

$$G(u, r) = \exp\left(-\frac{u^2}{r^2}\right) \tag{5.4}$$

where $u = \|\mathbf{x} - \mathbf{c}_m\|$ is the Euclidean distance from the input to the center, and $r = r_m$ is the radius parameter controlling the width of the basis function.

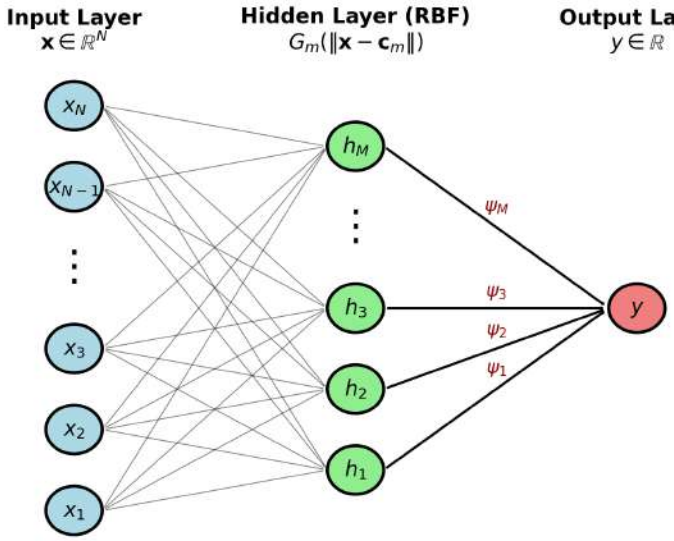


Figure 5.2: RBF Network Architecture

Training Procedure

Given a set of T training patterns $\{(\hat{\mathbf{x}}_t, \hat{y}_t)\}_{t=1}^{T}$, the training objective is to determine the optimal synaptic weights $\boldsymbol{\psi} = [\psi_1, \psi_2, \ldots, \psi_M]^T$ that minimize the squared error:

$$E(\boldsymbol{\psi}) = \frac{1}{2} \sum_{t=1}^{T} \left(\hat{y}_t - \sum_{m=1}^{M} \psi_m \, G_m(\|\hat{\mathbf{x}}_t - \mathbf{c}_m\|) \right)^2 \tag{5.5}$$

This objective can be expressed in compact matrix form as:

$$E(\boldsymbol{\psi}) = \frac{1}{2} \left\| \hat{\mathbf{y}} - \mathbf{G}\boldsymbol{\psi} \right\|^2 \tag{5.6}$$

where $\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T]^T$ is the vector of desired outputs, and $\mathbf{G} \in \mathbb{R}^{T \times M}$ is the design matrix with elements:

$$G_{tm} = G_m(\|\hat{\mathbf{x}}_t - \mathbf{c}_m\|), \quad t = 1, \ldots, T, \quad m = 1, \ldots, M \tag{5.7}$$

Explicitly, the design matrix is:

$$\mathbf{G} = \begin{bmatrix} G_1(\|\hat{\mathbf{x}}_1 - \mathbf{c}_1\|) & G_2(\|\hat{\mathbf{x}}_1 - \mathbf{c}_2\|) & \cdots & G_M(\|\hat{\mathbf{x}}_1 - \mathbf{c}_M\|) \\ G_1(\|\hat{\mathbf{x}}_2 - \mathbf{c}_1\|) & G_2(\|\hat{\mathbf{x}}_2 - \mathbf{c}_2\|) & \cdots & G_M(\|\hat{\mathbf{x}}_2 - \mathbf{c}_M\|) \\ \vdots & \vdots & \ddots & \vdots \\ G_1(\|\hat{\mathbf{x}}_T - \mathbf{c}_1\|) & G_2(\|\hat{\mathbf{x}}_T - \mathbf{c}_2\|) & \cdots & G_M(\|\hat{\mathbf{x}}_T - \mathbf{c}_M\|) \end{bmatrix} \tag{5.8}$$

The solution $\boldsymbol{\psi}^*$ to this linear least-squares problem is obtained by solving the normal equations:

$$\mathbf{G}^T\mathbf{G}\,\boldsymbol{\psi}^* = \mathbf{G}^T\hat{\mathbf{y}} \tag{5.9}$$

In the special case where the number of RBF centers equals the number of training patterns ($M = T$) and each center coincides with a training pattern ($\mathbf{c}_m = \hat{\mathbf{x}}_m$), the network interpolates the training data exactly. The synaptic weights are then given by:

$$\mathbf{G}\,\boldsymbol{\psi}^* = \hat{\mathbf{y}} \tag{5.10}$$

provided that $\mathbf{G}$ is invertible.

5.3.4 Growing RBF Networks with Self-Organizing Maps

When the number of training patterns is large, using all patterns as RBF centers can lead to overfitting, where the network memorizes training data but generalizes poorly to new inputs. To address this, a growing network strategy is employed that incrementally increases model complexity while monitoring generalization performance.

Algorithm Overview

The growing SOM-RBFN algorithm adaptively determines the number of RBF centers and their locations through an iterative procedure:

1. **Initialization**: Set the initial number of centers to $M = M_{\min}$ (e.g., $M_{\min} = 5$). Partition the available data into training and validation sets to monitor overfitting.

2. **Center placement via SOM**: Apply a Self-Organizing Map (SOM) algorithm to determine the positions of the M centers $\{\mathbf{c}_m\}_{m=1}^{M}$ based on the distribution of training patterns. This ensures that centers are placed in regions with higher data density.

3. **Radius calculation**: Compute the radius r_m for each center using an MST-based procedure (detailed below).

4. **Weight training and error evaluation**: Solve the least-squares problem to obtain synaptic weights $\boldsymbol{\psi}^*$. Evaluate training error E_{train} and validation error E_{test}.

5. **Termination check**: If the validation error E_{test} increases relative to the previous iteration or if $M \geq M_{\max}$, terminate the procedure. An increasing validation error indicates overfitting.

6. **Center splitting**: Identify the center responsible for the largest approximation errors in the training set. Split this center into two nearby centers to increase model flexibility in regions with high error.

7. **Iteration**: Increment $M \leftarrow M + 1$ and return to step 2.

Self-Organizing Map for Center Placement

The SOM algorithm positions centers adaptively based on the density and distribution of training patterns. Figure 5.3a shows the initial random placement of centers, while Figure 5.3b illustrates the adapted configuration after convergence. The procedure is as follows:

1. **Center initialization**: Initialize centers $\mathbf{c}_m^{(0)}$, $m = 1, \ldots, M$, either randomly within the design space or using results from a previous iteration.

2. **Competitive learning**: For each iteration n, randomly select a training pattern $\hat{\mathbf{x}}_t$ and identify the nearest center (winner):

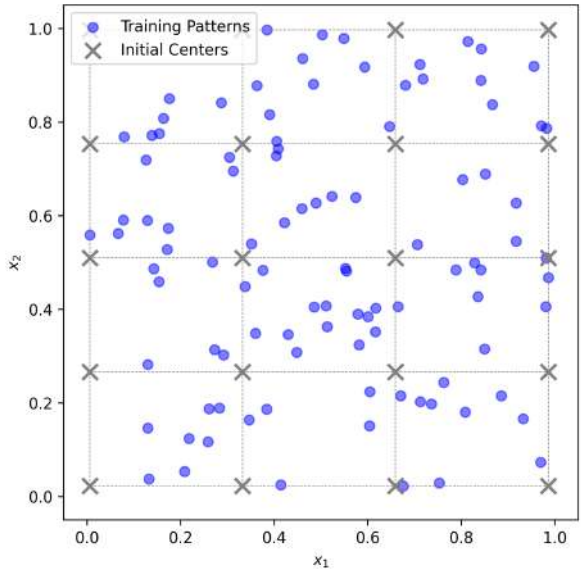
$$i(\hat{\mathbf{x}}_t) = \arg \min_{m=1,\ldots,M} \|\hat{\mathbf{x}}_t - \mathbf{c}_m^{(n)}\| \tag{5.11}$$

3. **Cooperative update**: Update all centers according to:

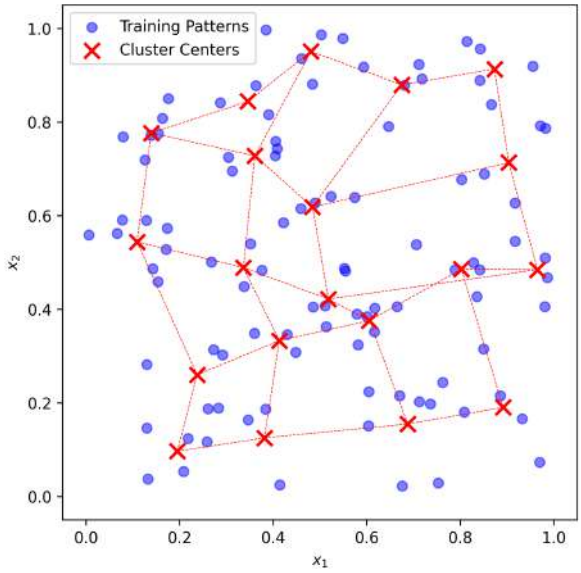
$$\mathbf{c}_m^{(n+1)} = \mathbf{c}_m^{(n)} + \eta(n)\, h_{m,i}(n)\, \left(\hat{\mathbf{x}}_t - \mathbf{c}_m^{(n)}\right) \tag{5.12}$$

where $\eta(n) \in (0, 1]$ is the learning rate and $h_{m,i}(n) \in [0, 1]$ is a neighborhood function that determines the influence of the winner i on center m. Both parameters decrease monotonically with n to ensure convergence.

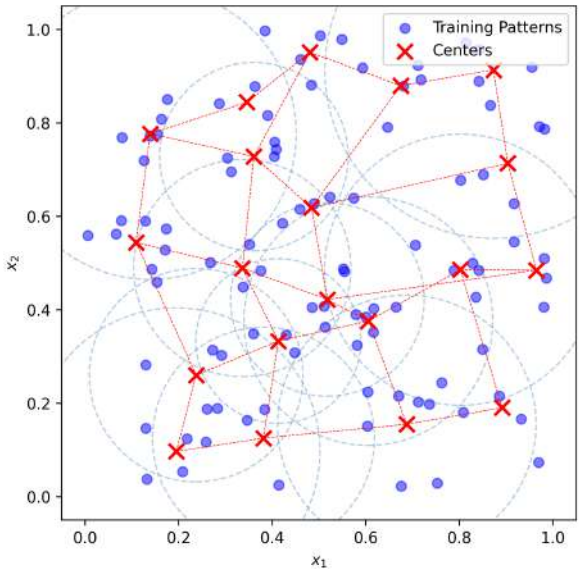
4. **Convergence check**: If the center positions change negligibly between successive iterations, terminate. Otherwise, increment $n \leftarrow n + 1$ and return to step 2.



(a) Initial random placement of SOM centers before training.



(b) SOM centers after convergence, adapted to the distribution of training patterns.



(c) Trained SOM centers with associated influence radii for RBF network construction.

Figure 5.3: Self-organizing map center placement: (a) initial configuration, (b) after training, (c) with influence radii r_m.

The neighborhood function $h_{m,i}(n)$ is typically defined as a Gaussian function of the topological distance between centers m and i.

RBF Width Determination via MST

The radius r_m of each RBF controls the locality of its influence (Figure 5.3c). These radii are determined using an MST-based procedure:

1. **MST construction**: Construct the Minimum Spanning Tree connecting all centers $\{\mathbf{c}_m\}_{m=1}^{M}$ based on their Euclidean distances.
2. **Local neighborhood identification**: For each center $\mathbf{c}_m$, identify all centers that can be reached by traversing at most n_b branches of the MST starting from $\mathbf{c}_m$. Denote this set of neighbors as C_m.
3. **Radius calculation**: Compute the radius as the mean distance to neighbors:

$$r_m = \frac{1}{|C_m|} \sum_{\mathbf{c}_i \in C_m} \|\mathbf{c}_m - \mathbf{c}_i\| \tag{5.13}$$

4. **Application**: Repeat for all centers $m = 1, \ldots, M$.

A value of $n_b \approx 4$ typically provides satisfactory coverage of the training patterns.

Center Splitting Strategy

When validation error has not yet increased and $M < M_{\max}$, the algorithm identifies the center requiring refinement. This is done by:

1. Computing the approximation error for each training pattern.
2. Selecting a percentage (e.g., 20%) of patterns with the largest errors.
3. Identifying the center closest to the majority of these high-error patterns.
4. Splitting the center, responsible for the largest number of the most erroneous estimates, into two centers.

Chapter 6

Benchmark Cases

6.1 Benchmark for Single-Objective Optimization

In order to evaluate the performance of QIEA, we use it on a set of test functions, each of which exhibits different characteristics.

Modality Modality refers to the number and distribution of optimal and suboptimal regions in the fitness landscape.

- **Unimodal:** A single global optimum with no local optima, enabling exploitation-focused strategies to converge reliably.
- **Multimodal:** Multiple local optima, necessitating a balance between exploration and exploitation.
- **Highly multimodal:** An exceptionally large number of local optima, often scaling exponentially with dimension, requiring sustained population diversity.

Separability Separability characterizes the degree of variable interaction (epistasis).

- **Separable:** Each variable contributes independently, allowing decomposition into n independent one-dimensional subproblems.
- **Non-separable:** Variable interactions couple the decision variables, requiring coordinated multidimensional search.

Conditioning Conditioning quantifies landscape anisotropy, measured by the Hessian condition number (ratio of largest to smallest eigenvalue).

- **Well-conditioned:** Comparable eigenvalues (condition number near unity), resulting in approximately spherical level sets.
- **Ill-conditioned:** Eigenvalues differing by orders of magnitude, creating elongated valleys with highly directional convergence.
- **Moderately conditioned:** Intermediate anisotropy introducing directional bias without severely impeding progress.

Additional Characteristics

- **Deceptive:** Fitness gradients direct search away from the global optimum.
- **Narrow valley:** The optimum resides in a long, narrow region requiring precise variable coordination.
- **High-frequency noise:** Rapid local oscillations obscure the global structure.

The chosen benchmark functions—Sphere, Ackley, Rastrigin, Griewank, Schwefel, and Rosenbrock, whose mathematical expressions are shown in Table 6.1—provide comprehensive coverage across these characteristics, as detailed in Table 6.2.

Function	**Mathematical Expression**
Sphere	$F(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$
Ackley	$F(\mathbf{x}) = -20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$
Rastrigin	$F(\mathbf{x}) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)]$
Griewank	$F(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
Schwefel	$F(\mathbf{x}) = 418.9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$
Rosenbrock	$F(\mathbf{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$

Table 6.1: Mathematical Expressions of Benchmark Functions

Function	**Modality**	**Separability**	**Conditioning**	**Deceptiveness**	**Primary Test Objective**
Sphere	Unimodal	Separable	Well-conditioned	Low	Baseline convergence speed and exploitation
Ackley	Multimodal	Non-separable	Moderate	High	Exploration-exploitation balance
Rastrigin	Highly multimodal	Separable	Well-conditioned	Low	Resistance to numerous local optima
Griewank	Multimodal	Non-separable	Moderate	Low	Variable interaction and scalability
Schwefel	Multimodal	Separable	Moderate	Very high	Robustness against deceptive landscapes
Rosenbrock	Unimodal	Non-separable	Ill-conditioned	Low	Convergence in narrow, curved valleys

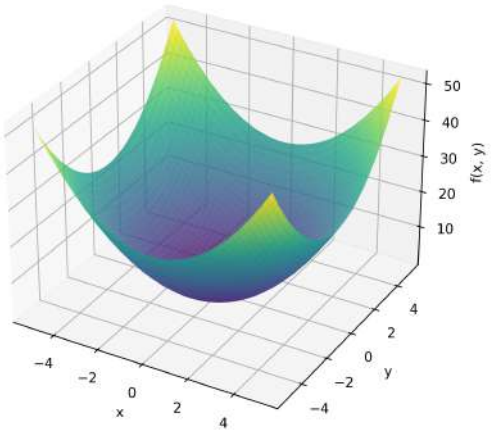
Table 6.2: Characteristics of Benchmark Functions and Properties Tested

All functions were optimized with 10 design variables, each encoded using 10 bits for binary representation, resulting in a total chromosome length of 100 bits. This yields a discrete search space of $2^{100} \approx 1.27 \times 10^{30}$ possible solutions. The search bounds for each variable are function-specific and are detailed in Table 6.3.

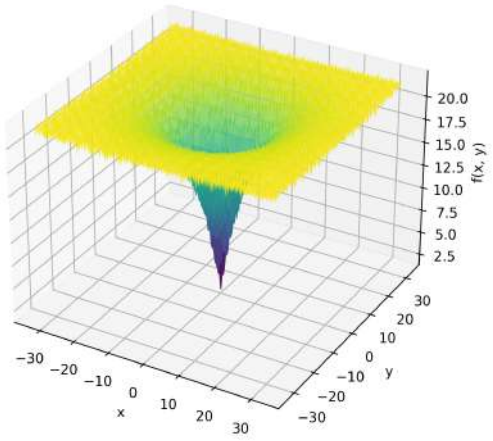
Test Function	**Bounds**	**Dimension**	**Bits per Variable**
Sphere	$[-5.12, 5.12]$	10	10
Ackley	$[-32.768, 32.768]$	10	10
Rastrigin	$[-5.12, 5.12]$	10	10
Griewank	$[-600, 600]$	10	10
Schwefel	$[-500, 500]$	10	10
Rosenbrock	$[-5, 10]$	10	10

Table 6.3: Benchmark test functions and their optimization parameters

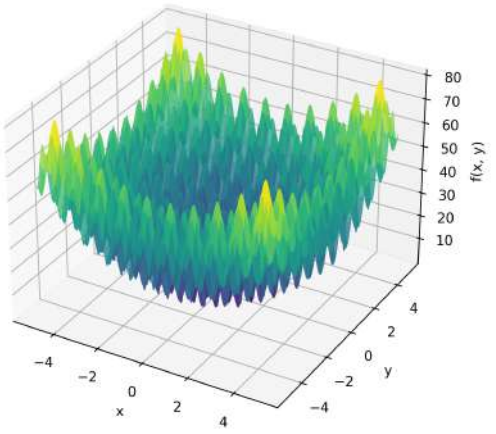
The landscape characteristics of the benchmark functions are shown in Figure 6.1.



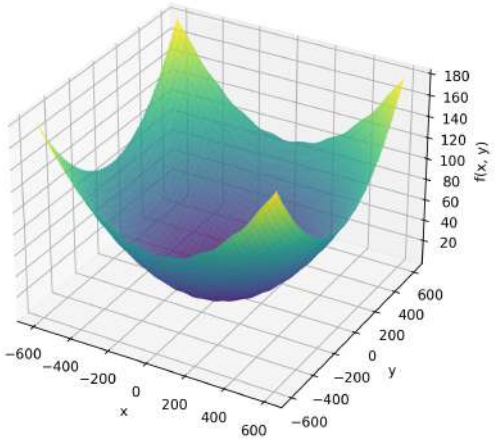
(a) Sphere



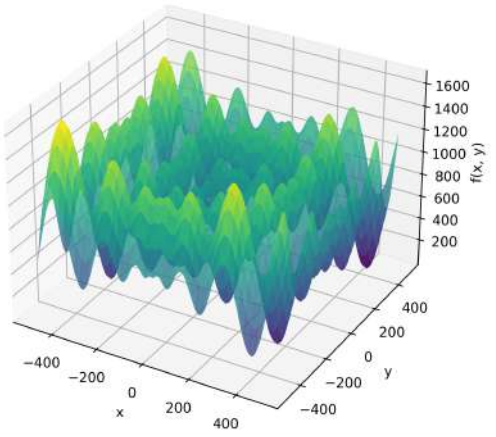
(b) Ackley



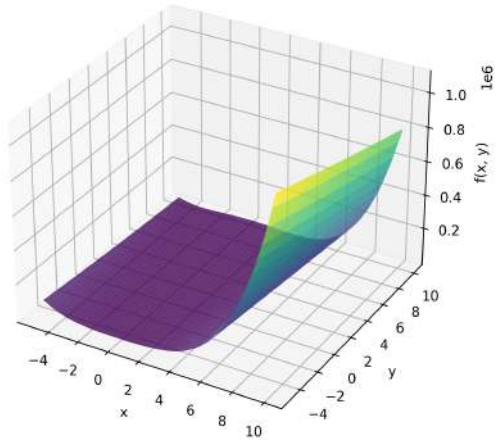
(c) Rastrigin



(d) Griewank



(e) Schwefel



(f) Rosenbrock

Figure 6.1: Visualization of benchmark test functions for two-dimensional input space.

6.1.1 Grid Search and Convergence Analysis of Optimal Configurations

The QIEA was executed across a grid of parameter configurations to identify the most effective settings for each benchmark test function. The parameter space explored is summarized in Table 6.4, where all listed parameter values were combined to form the grid of configurations. Parameters not included in the tested values column were held constant at the specified value throughout all experiments. All configurations were evaluated over 10 independent runs with different random seeds, and the averaged convergence behavior was recorded. For each test function, the five configurations yielding the fastest convergence were identified and documented. Figure 6.2 presents a comparative visualization of the averaged convergence curves for these top-performing configurations across all benchmark functions.

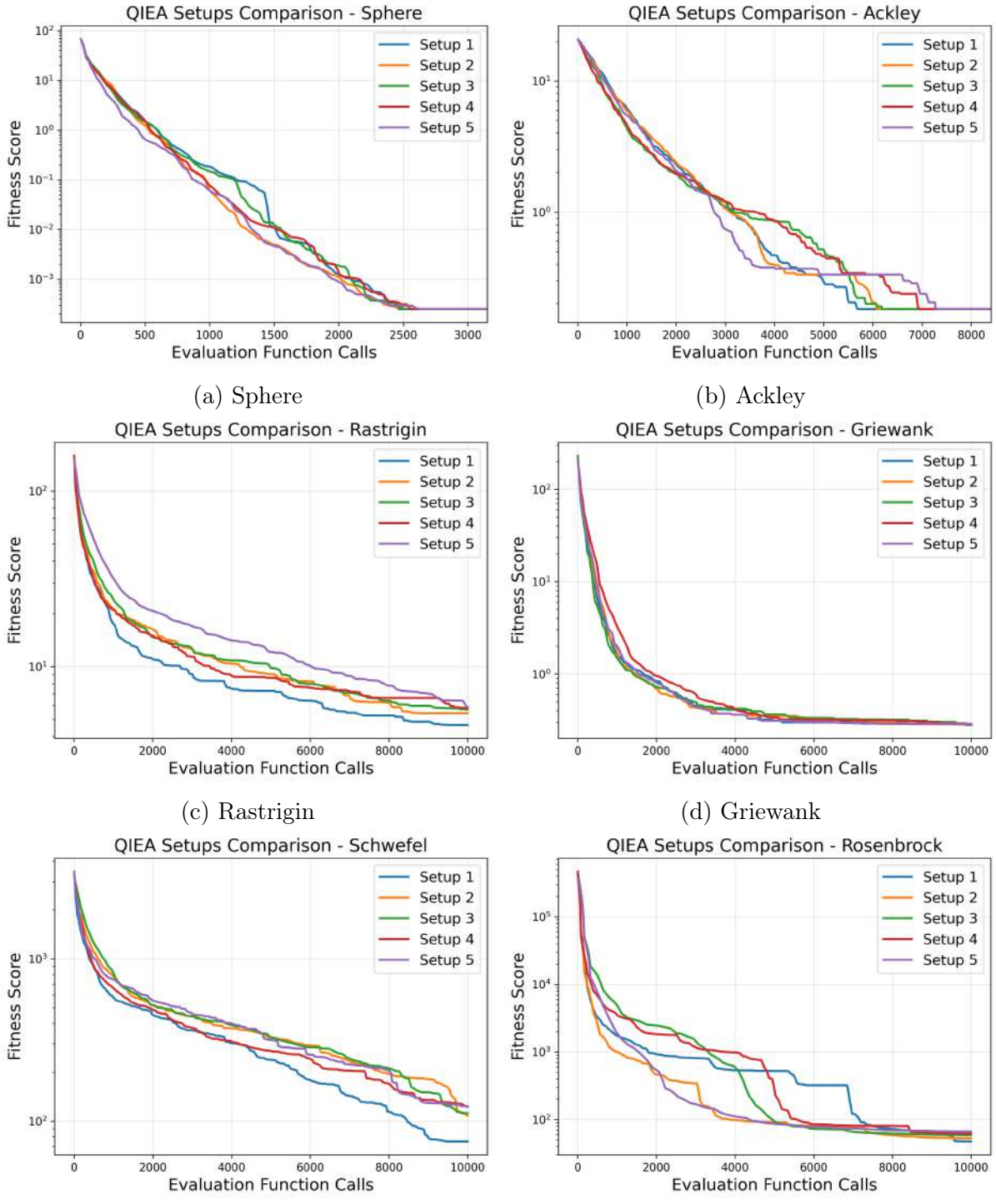
To provide insight into the behavior of the algorithm under different initializations, the individual convergence trajectories of all 10 seeds for the single best-performing configuration on each test function are displayed in Figure 6.3.

Parameter	**Notation**	**Tested Values**
Parent population size	μ	$\{6, 12\}$
Offspring population size	λ	$\{4, 6, 8\}$
Elites' population size	e	$\{6, 12\}$
Measurements per individual	m/i	$\{1, 2\}$
Q-Gate rotation table		Custom configuration[1]
Q-Gate angle of rotation	$\Delta\theta$	1.0
Qubit bound	$\theta_{\max}$	$\{\pi/2, 0.96875 \cdot \pi/2\}$
Crossover probability	p_x	0.2
Crossover period	T_{x}	$\{10, 15\}$
Mutation probability (rotation)	p_m^{rot}	0.0
Mutation probability (measurement)	p_m^{meas}	$\{0.01, 0.02\}$
Neighbourhood size	N_{nbr}	2
Local migration period	T_{loc}	30
Global migration period	T_{glob}	–
Selection method		Linear weighted
Elitism rate	r_{elite}	0.2
Elite inclusion probability	p_{elite}	0.2

Table 6.4: QIEA configuration parameters and their tested values. Parameters with a single value were held constant across all configurations.

Observations In the plots of the different seeds, it is observed that, for the same parameterization on the same test functions, the number of evaluations required to achieve convergence of a similar order of magnitude for the different seeds differs significantly. This observation indicates that the algorithm is susceptible to becoming stuck in premature convergence, which is obviously related to the small number of populations for which the algorithm exhibits good convergence.

[1] The rotation table uses the configuration: [`randfloat`, −1, 1, `randfloat`, 0, 1, −1, 0].



(a) Sphere

(b) Ackley

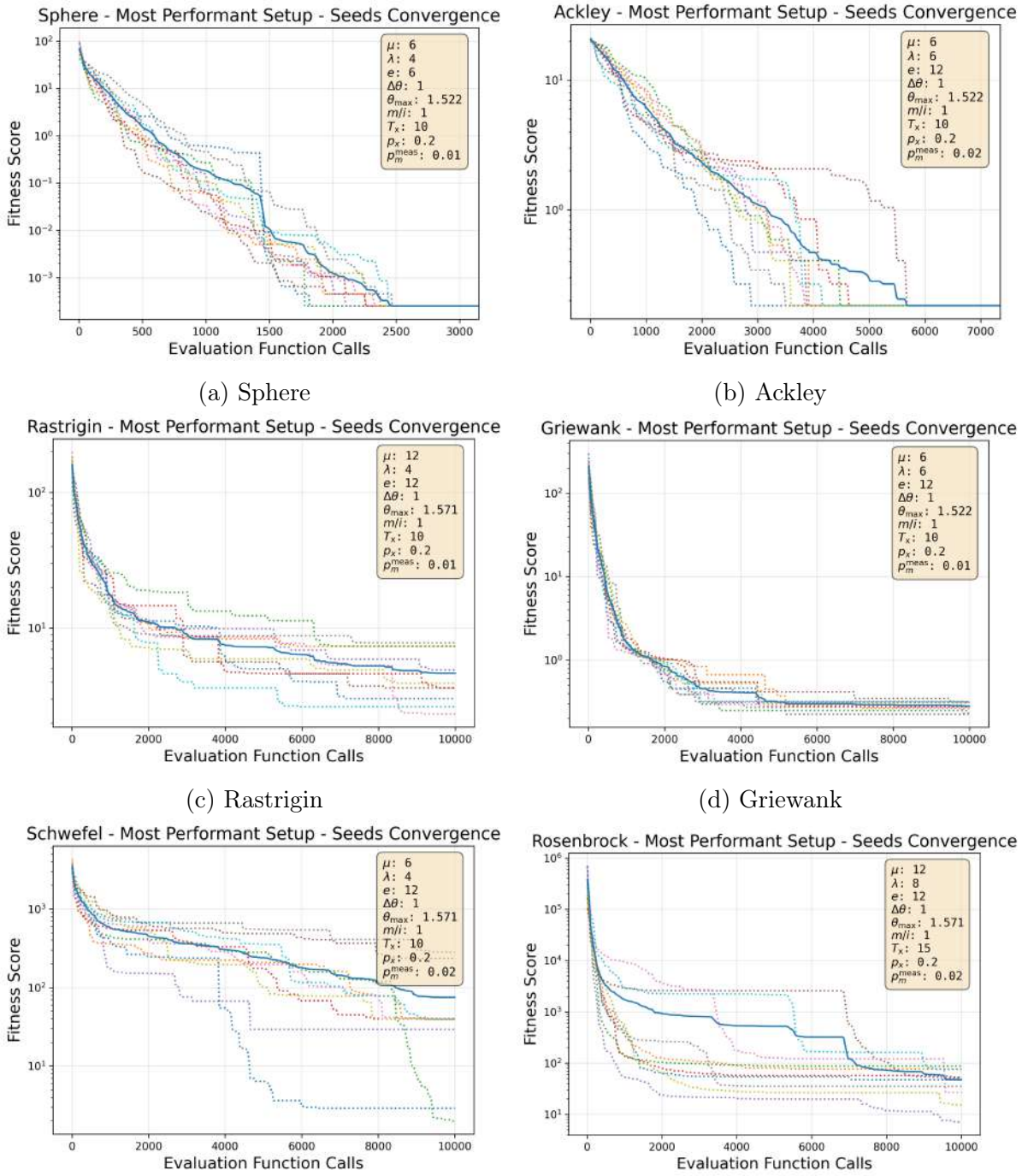
(c) Rastrigin

(d) Griewank

(e) Schwefel

(f) Rosenbrock

Figure 6.2: QIEA most performant setups comparison



(a) Sphere

(b) Ackley

(c) Rastrigin

(d) Griewank

(e) Schwefel

(f) Rosenbrock

Figure 6.3: Individual seed convergence trajectories for the best-performing QIEA configuration on each test function

Setup	μ	λ	e	m/i	$\Delta\theta$	$\theta_{\max}$	$T_{\mathbf{x}}$	$p_m^{\mathbf{meas}}$
Setup 1	6	4	6	1	1.0	$0.96875\pi/2$	10	0.01
Setup 2	6	4	12	1	1.0	$\pi/2$	15	0.01
Setup 3	6	4	12	1	1.0	$0.96875\pi/2$	10	0.01
Setup 4	6	4	6	1	1.0	$\pi/2$	15	0.01
Setup 5	6	4	12	1	1.0	$\pi/2$	10	0.01

Table 6.5: 5 Most Performant QIEA Parameter Configurations for Sphere Function

Setup	μ	λ	e	m/i	$\Delta\theta$	$\theta_{\max}$	$T_{\mathbf{x}}$	$p_m^{\mathbf{meas}}$
Setup 1	6	6	12	1	1.0	$0.96875\pi/2$	10	0.02
Setup 2	6	8	6	1	1.0	$0.96875\pi/2$	10	0.01
Setup 3	12	4	12	1	1.0	$\pi/2$	15	0.02
Setup 4	6	4	6	1	1.0	$\pi/2$	10	0.02
Setup 5	12	6	12	1	1.0	$0.96875\pi/2$	15	0.01

Table 6.6: 5 Most Performant QIEA Parameter Configurations for Ackley Function

Setup	μ	λ	e	m/i	$\Delta\theta$	$\theta_{\max}$	$T_{\mathbf{x}}$	$p_m^{\mathbf{meas}}$
Setup 1	12	4	12	1	1.0	$\pi/2$	10	0.01
Setup 2	6	6	12	1	1.0	$0.96875\pi/2$	15	0.01
Setup 3	6	4	12	2	1.0	$0.96875\pi/2$	10	0.01
Setup 4	6	4	6	1	1.0	$\pi/2$	15	0.01
Setup 5	6	6	12	2	1.0	$\pi/2$	15	0.02

Table 6.7: 5 Most Performant QIEA Parameter Configurations for Rastrigin Function

Setup	μ	λ	e	m/i	$\Delta\theta$	$\theta_{\max}$	$T_{\mathbf{x}}$	$p_m^{\mathbf{meas}}$
Setup 1	6	6	12	1	1.0	$0.96875\pi/2$	10	0.01
Setup 2	6	6	6	1	1.0	$\pi/2$	10	0.01
Setup 3	12	4	6	1	1.0	$\pi/2$	15	0.02
Setup 4	12	8	12	1	1.0	$\pi/2$	15	0.01
Setup 5	12	6	12	1	1.0	$0.96875\pi/2$	15	0.01

Table 6.8: 5 Most Performant QIEA Parameter Configurations for Griewank Function

Setup	μ	λ	e	m/i	$\Delta\theta$	$\theta_{\max}$	$T_{\mathbf{x}}$	$p_m^{\mathbf{meas}}$
Setup 1	6	4	12	1	1.0	$\pi/2$	10	0.02
Setup 2	12	6	6	1	1.0	$\pi/2$	15	0.02
Setup 3	12	8	6	1	1.0	$0.96875\pi/2$	15	0.02
Setup 4	6	6	12	1	1.0	$0.96875\pi/2$	10	0.02
Setup 5	6	4	12	1	1.0	$\pi/2$	10	0.01

Table 6.9: 5 Most Performant QIEA Parameter Configurations for Schwefel Function

Setup	μ	λ	e	m/i	$\Delta\theta$	$\theta_{\max}$	$T_{\mathbf{x}}$	$p_m^{\mathbf{meas}}$
Setup 1	12	8	12	1	1.0	$\pi/2$	15	0.02
Setup 2	12	8	12	1	1.0	$0.96875\pi/2$	10	0.02
Setup 3	12	8	6	2	1.0	$0.96875\pi/2$	15	0.01
Setup 4	6	4	12	2	1.0	$\pi/2$	10	0.01
Setup 5	12	8	12	2	1.0	$0.96875\pi/2$	10	0.02

Table 6.10: 5 Most Performant QIEA Parameter Configurations for Rosenbrock Function

6.1.2 Parametric Analysis

A parametric analysis was conducted to assess the influence of individual parameters on algorithm performance. For each test function, the best-performing configuration identified in the grid search was used as a baseline. Parameters were then varied individually while holding all other parameters constant.

The parameter values examined in this analysis are summarized in Table 6.11. The rotation table strategies, which govern the quantum gate rotation behavior, are detailed separately in Table 6.12.

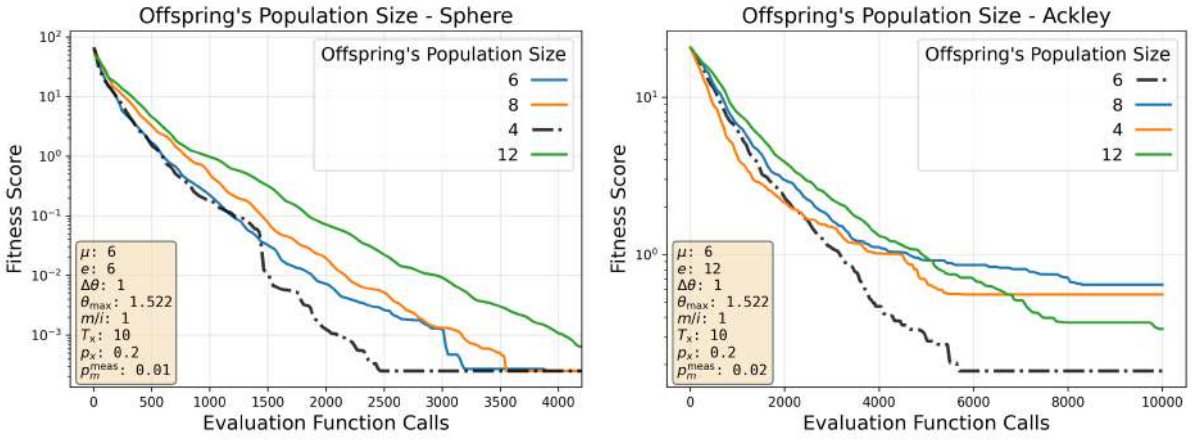
Parameter	**Notation**	**Tested Values**
Parent population size	μ	$\{4, 6, 8, 12\}$
Offspring population size	λ	$\{4, 6, 8, 12\}$
Elites' population size	e	$\{4, 6, 8, 12\}$
Measurements per individual	m/i	$\{1, 2\}$
Q-Gate Angle of rotation	$\Delta\theta$	$\{0.8, 1.0, 1.2\}$
Mutation probability (rotation)	p_m^{rot}	0.0
Mutation probability (measurement)	p_m^{meas}	$\{0.01, 0.02, 0.04\}$
Qubit bound	$\theta_{\max}$	$\{\pi/2, 0.975 \cdot \pi/2, 0.96875 \cdot \pi/2, 0.95 \cdot \pi/2\}$
Crossover period	T_{x}	$\{8, 10, 12, 15\}$

Table 6.11: Parameter values tested in the parametric analysis

ID	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$	$\mathbf{a_4}$	$\mathbf{a_5}$	$\mathbf{a_6}$	$\mathbf{a_7}$	$\mathbf{a_8}$
RT1	0	0	0	0	0	1	-1	0
RT2	0	-1	1	0	0	1	-1	0
RT3	`randfloat`	0	0	`randfloat`	0	1	-1	0
RT4	`randint`	0	0	`randint`	0	1	-1	0
RT5	`randfloat`	-1	1	`randfloat`	0	1	-1	0
RT6	`randint`	-1	1	`randint`	0	1	-1	0
RT7	0	0	0	0	`randfloat`	1	-1	`randfloat`
RT8	0	-1	1	0	`randfloat`	1	-1	`randfloat`
RT9	`randfloat`	0	0	`randfloat`	`randfloat`	1	-1	`randfloat`
RT10	`randint`	0	0	`randint`	`randfloat`	1	-1	`randfloat`
RT11	`randfloat`	-1	1	`randfloat`	`randfloat`	1	-1	`randfloat`
RT12	`randint`	-1	1	`randint`	`randfloat`	1	-1	`randfloat`
RT13	0	0	0	0	`randint`	1	-1	`randint`
RT14	0	-1	1	0	`randint`	1	-1	`randint`
RT15	`randfloat`	0	0	`randfloat`	`randint`	1	-1	`randint`
RT16	`randint`	0	0	`randint`	`randint`	1	-1	`randint`
RT17	`randfloat`	-1	1	`randfloat`	`randint`	1	-1	`randint`
RT18	`randint`	-1	1	`randint`	`randint`	1	-1	`randint`

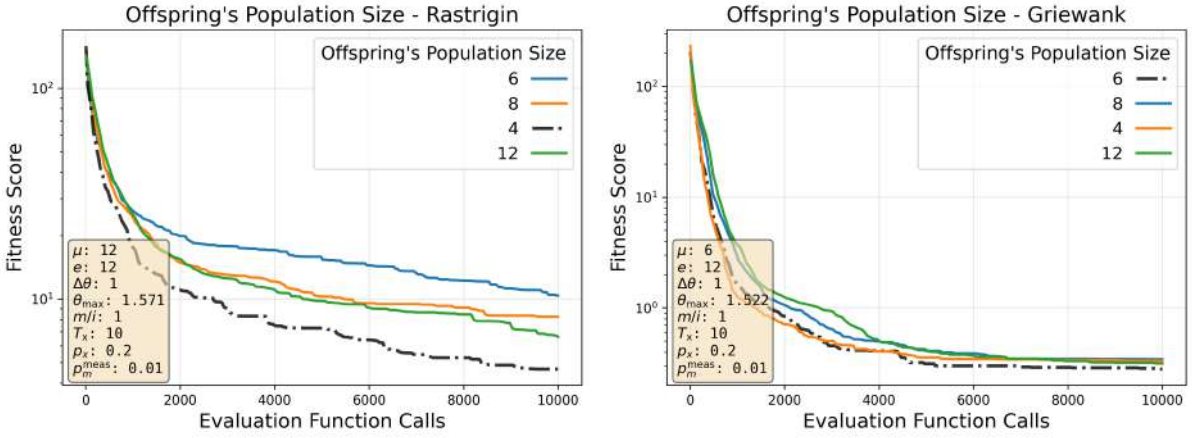
Table 6.12: Rotation table strategies tested in the parametric analysis. Each configuration defines an eight-element lookup table used to determine rotation direction and magnitude based on fitness comparisons. Elements denoted as `randfloat` represent uniform random sampling from $[-1, 1]$, while `randint` denotes uniform random sampling from $\{-1, 0, 1\}$.

Indicative diagrams are presented for some of these parametric analyses.



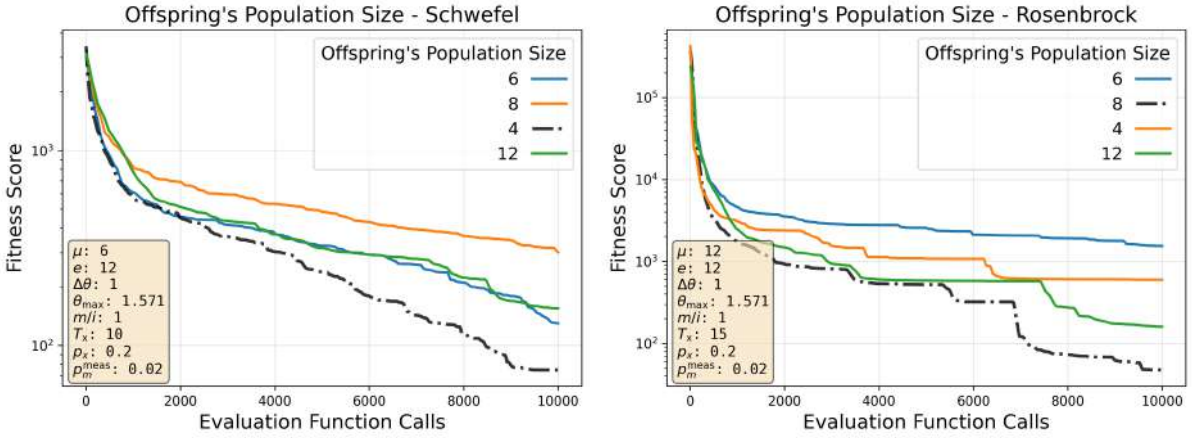
(a) Sphere

(b) Ackley



(c) Rastrigin

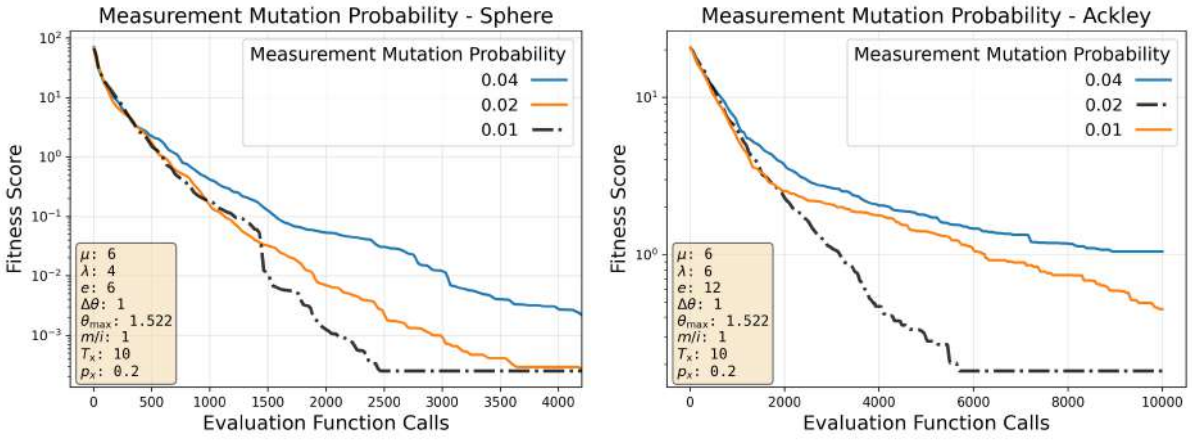
(d) Griewank



(e) Schwefel

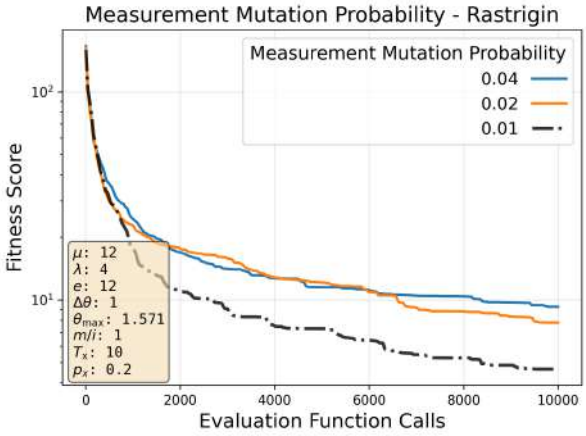
(f) Rosenbrock

Figure 6.4: Influence of offspring population size parameter

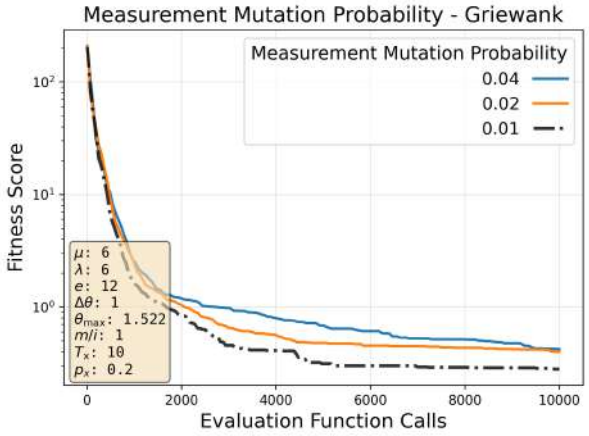


(a) Sphere

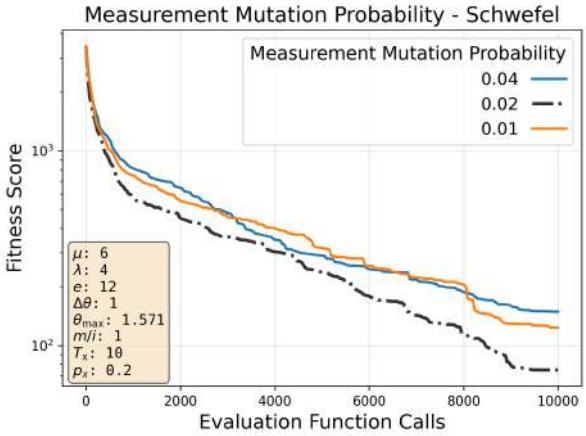
(b) Ackley



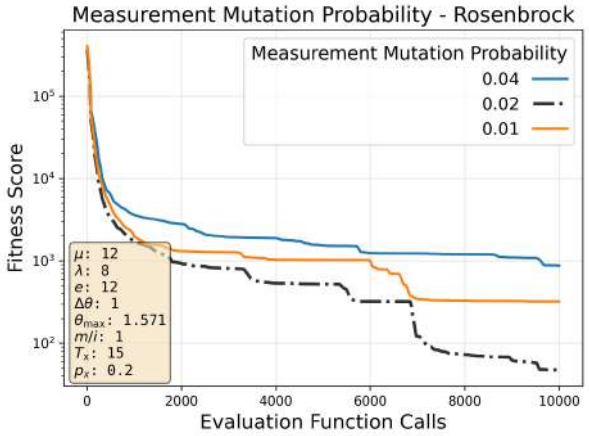
(c) Rastrigin



(d) Griewank

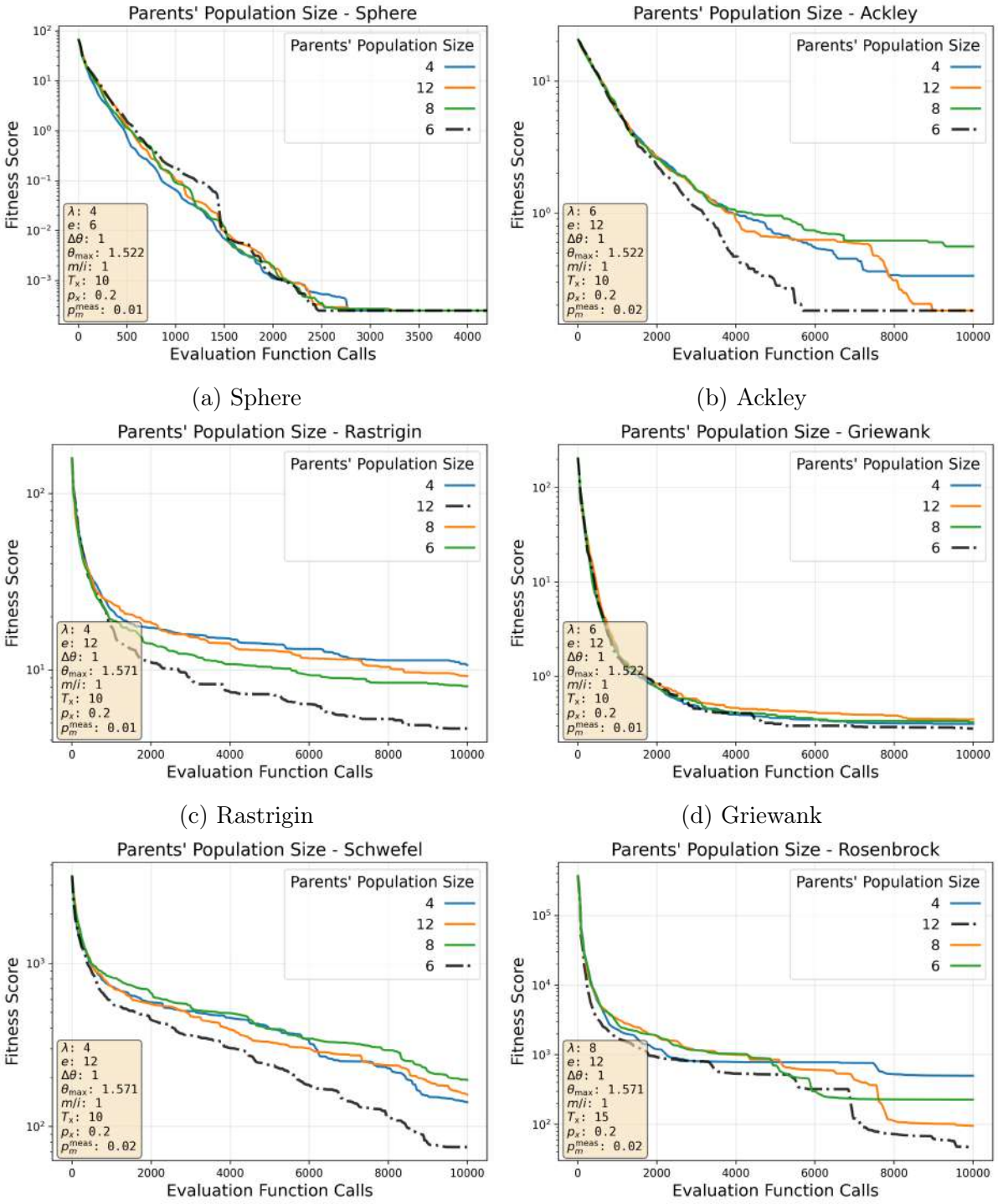


(e) Schwefel



(f) Rosenbrock

Figure 6.5: Influence of mutation probability on measurement outcomes



(a) Sphere

(b) Ackley

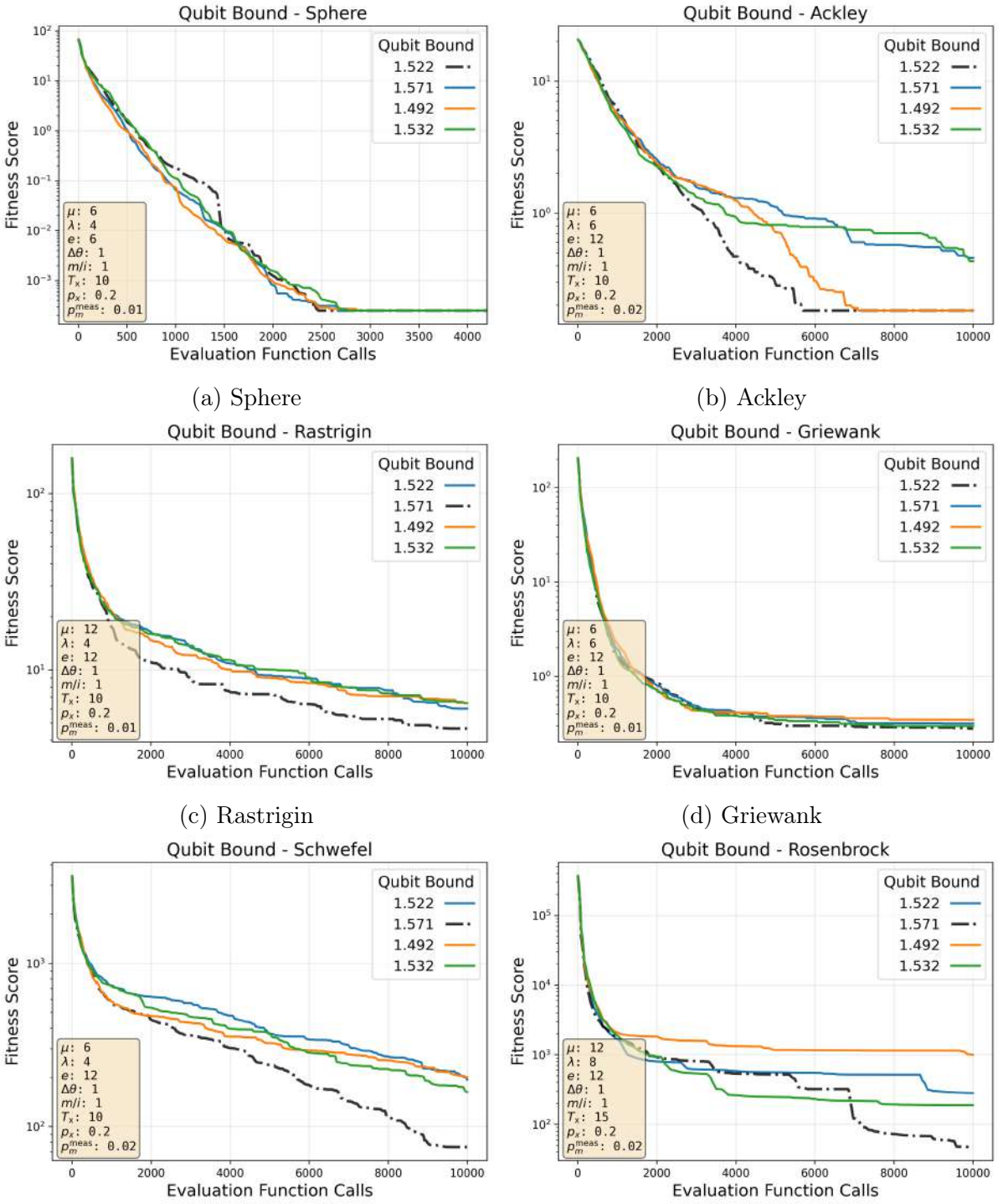
(c) Rastrigin

(d) Griewank

(e) Schwefel

(f) Rosenbrock

Figure 6.6: Influence of parents population size parameter



(a) Sphere

(b) Ackley

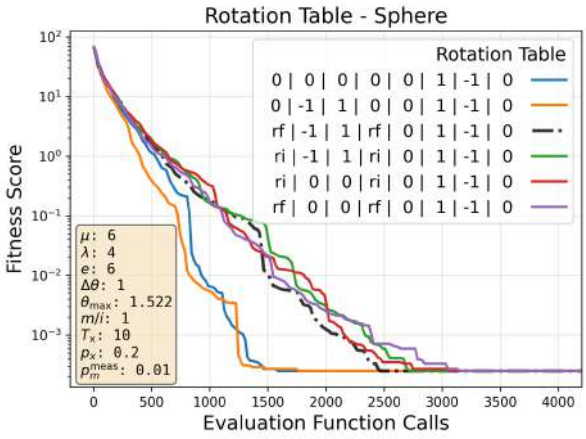
(c) Rastrigin

(d) Griewank

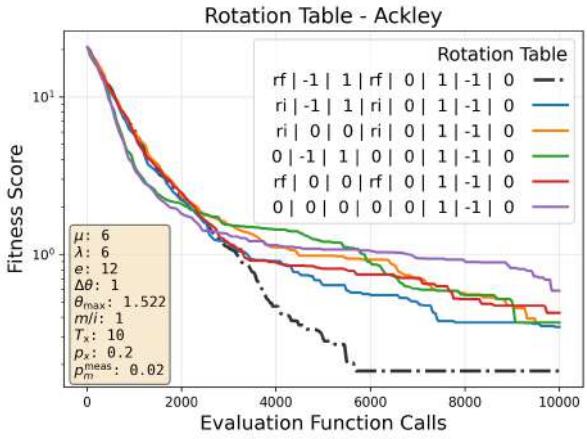
(e) Schwefel

(f) Rosenbrock

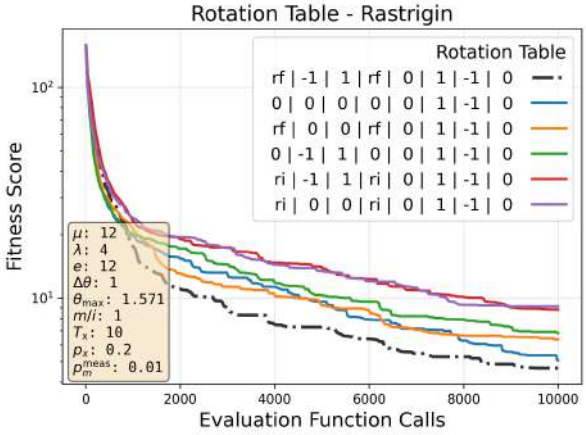
Figure 6.7: Influence of qubit bound parameter



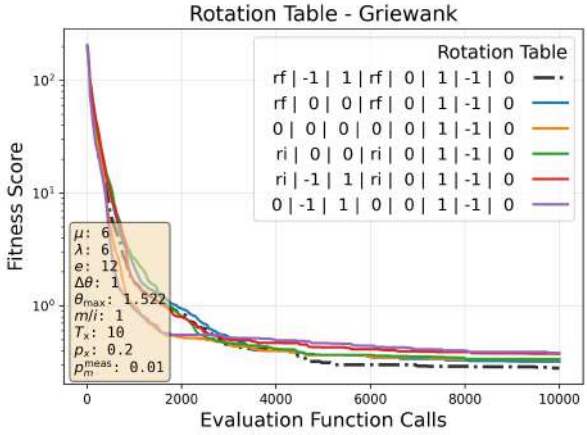
(a) Sphere



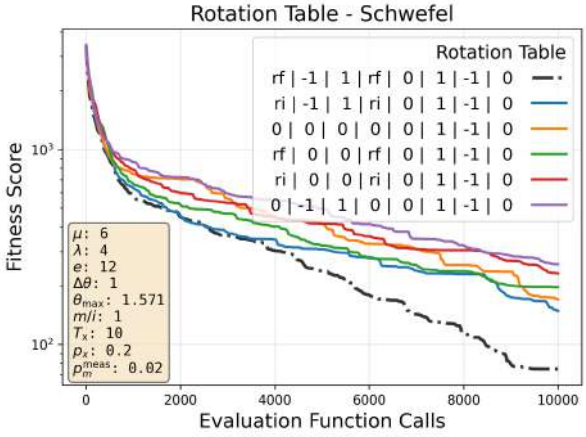
(b) Ackley



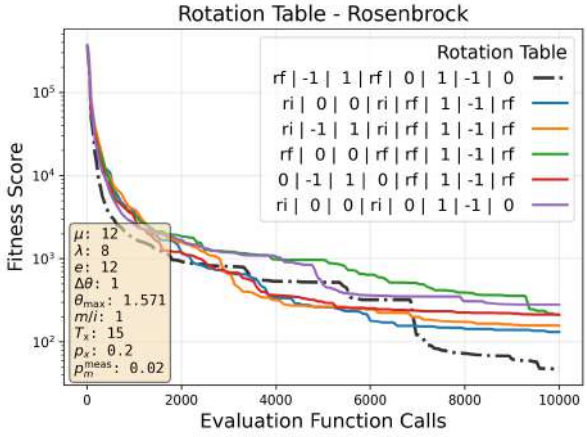
(c) Rastrigin



(d) Griewank



(e) Schwefel



(f) Rosenbrock

Figure 6.8: Influence of rotation table parameter

Observations It is observed that small changes in the value of a parameter (like for example the parent population size) lead to significant changes in the convergence speed, which indicates a strong sensitivity of the algorithm's convergence speed to the parameter values. Furthermore, it is observed that for a value of a parameter for which the convergence speed may be satisfactory, when the same parameter value is used in a different parametrization of the QIEA or in a different optimization problem, the effect of the parameter changes. This shows that there is a correlation both among the parameters themselves and between the parametrization and the problem being optimized.

Another interesting observation concerns the rotation table. A total of eighteen different parameterizations were evaluated, and the six that performed best were shown in each figure of the corresponding parametric analysis diagrams. For all the test cases, the parameterization RT1[2] seems to give the best convergence, except in the case of the sphere. For the sphere, the simplest parameterization RT2[3] does. Each parameterization exhibits different characteristics. The first case introduces more exploration, which helps prevent the algorithm from getting trapped too easily in local optima, or to search for different solutions even when it has already found a relatively good one. The second and simpler parameterization is more exploitation-oriented, which is beneficial for a smooth landscape such as that of the sphere function.

6.1.3 QIEA - EA Comparison

Following the parametric analysis, the rotation table strategies $[0, 0, 0, 0, 0, 1, -1, 0]$ and $[0, -1, 1, 0, 0, 1, -1, 0]$ demonstrated enhanced convergence speed on the sphere function. Based on these findings, the most performant configurations identified in the preliminary experiments were re-evaluated with metamodel assistance enabled for the Sphere, Ackley, and Rosenbrock functions.

The metamodel configuration parameters employed for each benchmark function are summarized in Table 6.13. A simple radial basis function (RBF) metamodel was used across all test cases, with variations in the minimum number of exact evaluations required before metamodel training.

Parameter	**Sphere**	**Ackley**	**Rosenbrock**
Metamodel type	Simple RBF	Simple RBF	Simple RBF
Minimum number of exact evaluations	100	100	200
Number of training patterns	8	8	8

Table 6.13: Metamodel configuration parameters for benchmark functions.

For the Sphere and Ackley functions, 100 exact function evaluations were used before metamodel training, while 200 exact evaluations were used for the Rosenbrock function. These values were selected based on preliminary trials as they provided the fastest convergence for the respective test cases.

In addition to the quantum-inspired evolutionary algorithm (QIEA), a conventional evolutionary algorithm (EA) was evaluated using the EASY software. A grid search was conducted over the following parameter space:

The most performant EA configuration, determined by minimum objective function value over independent runs, was subsequently re-evaluated with metamodel assistance

[2]RT1: [`randfloat`, -1, 1, `randfloat`, 0, 1, -1, 0]
[3]RT2: [0, 0, 0, 0, 0, 1, -1, 0]

Table 6.14: EA configuration parameters and their tested values.

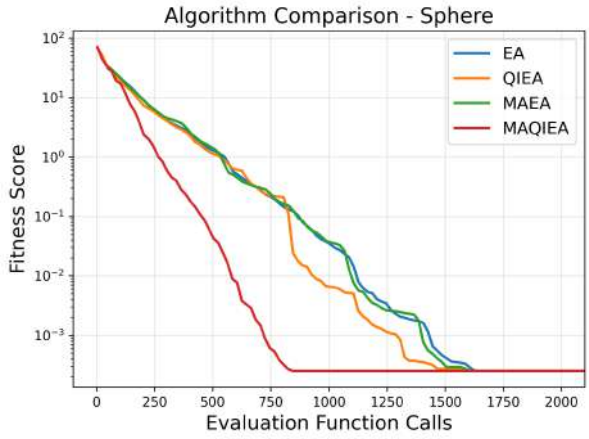
Parameter	**Notation**	**Tested Values**
Parents' population size	μ	$\{10, 15, 20, 30, 50\}$
Offspring's population size	λ	$\{50, 60, 100, 150\}$
Elitism rate	r_{elite}	$\{0.1, 0.2, 0.3\}$
Crossover probability	p_x	$\{0.7, 0.9\}$
Mutation probability	p_m	$\{0.01, 0.02, 0.05\}$

for the Sphere, Ackley, and Rosenbrock functions. This allowed for a direct comparison between the QIEA and EA under both standard and metamodel-assisted conditions. The comparative results are presented in Figure 6.9.

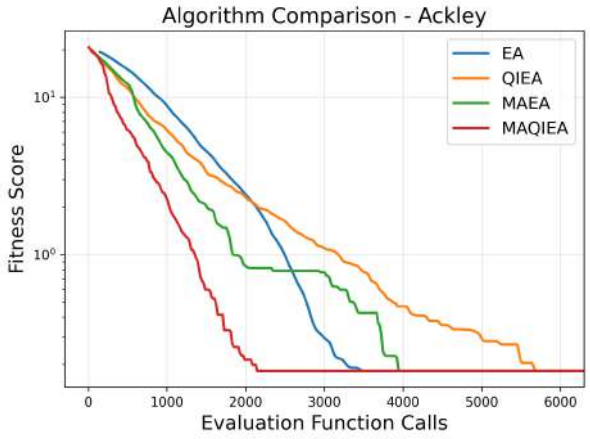
Observatios It should be noted that the results presented here are meant to assess the behavior of the QIEA algorithm by evaluating it in parallel with a classical EA. Even if some plots appear to indicate that QIEA converges faster than EA, this should not be interpreted as evidence of its superiority. In developing QIEA, we tested many different combinations of methods and parameters, and we report only those settings that performed best. We did not follow the same exhaustive tuning process for the results obtained with the EASY software, because our goal was not to rigorously compare the two algorithms. Instead, we used the performance of the EA, an algorithm with which we were more familiar, as a baseline for assessing the behavior and convergence speed of QIEA. Consequently, for the EA we generated a set of results without focusing on achieving the fastest possible convergence through extensive hyperparameter optimization.

In cases such as the Sphere function or Ackley, we see that for certain configurations the convergence speed is satisfactory. For cases such as Rastrigin and Griewank, which are highly multimodal, we observe good convergence speed at the beginning, which then deteriorates after a certain number of evaluations, something that suggests that the algorithm is prone to becoming trapped in local optima, which was also observed from the plots of the different seeds and is consistent with the small population sizes for which QIEA works best. In deceptive landscapes such as Schwefel, the algorithm appears to struggle even more, which further indicates its tendency to get trapped in local optima. For Rosenbrock we again observe good convergence at the beginning, which however worsens after some evaluations, indicating that the correlation among the design variables in this particular problem also made things more difficult for the algorithm.

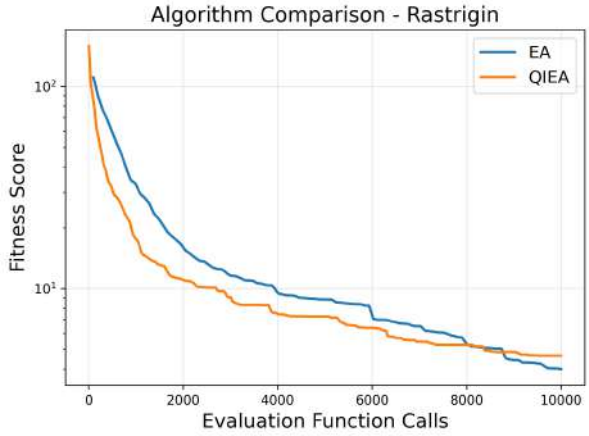
The effect of the metamodels appears to be quite strong, as it reduces the number of evaluations to about one third, compared to the non-assisted QIEA, in order to achieve convergence of a similar order of magnitude.



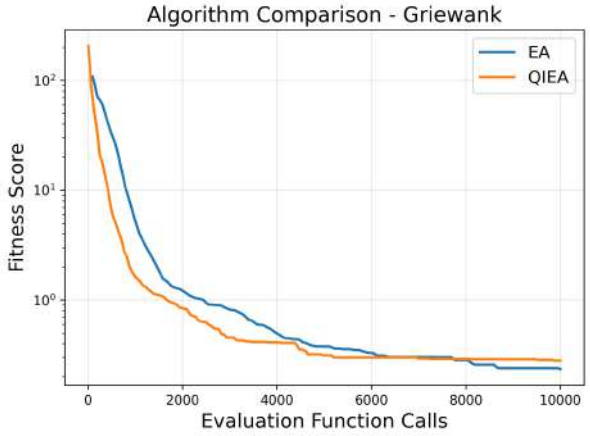
(a) Sphere



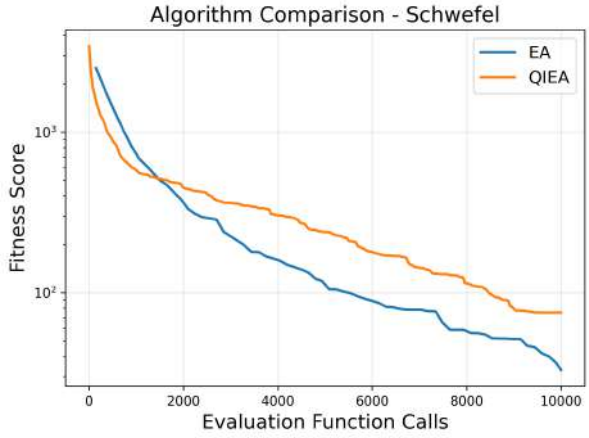
(b) Ackley



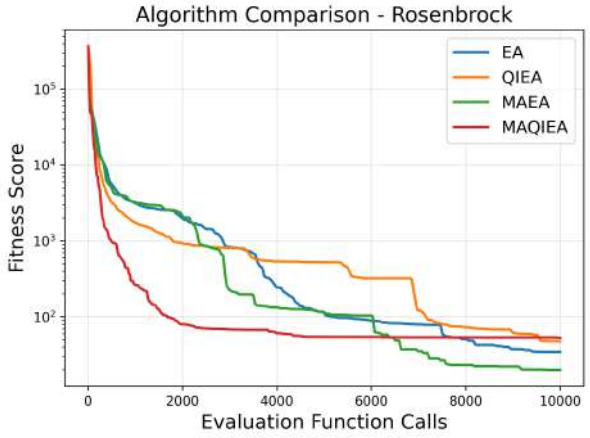
(c) Rastrigin



(d) Griewank



(e) Schwefel



(f) Rosenbrock

Figure 6.9: QIEA - EA comparison plots

6.2 Benchmark Cases for Multi-Objective Optimization

For multi-objective optimization, three benchmark functions were selected: ZDT1, ZDT2, and ZDT3. Table 6.16 summarizes the key characteristics of these functions and the primary algorithmic capabilities each is designed to test. The mathematical expressions of the chosen functions are shown in Table 6.15.

Function	Mathematical Expression
ZDT1	$g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$
	$F_1(\mathbf{x}) = x_1$
	$F_2(\mathbf{x}) = g(\mathbf{x})\left[1 - \sqrt{\frac{F_1(\mathbf{x})}{g(\mathbf{x})}}\right]$
ZDT2	$g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$
	$F_1(\mathbf{x}) = x_1$
	$F_2(\mathbf{x}) = g(\mathbf{x})\left[1 - \left(\frac{F_1(\mathbf{x})}{g(\mathbf{x})}\right)^2\right]$
ZDT3	$g(\mathbf{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$
	$F_1(\mathbf{x}) = x_1$
	$F_2(\mathbf{x}) = g(\mathbf{x})\left[1 - \sqrt{\frac{F_1(\mathbf{x})}{g(\mathbf{x})}} - \frac{F_1(\mathbf{x})}{g(\mathbf{x})}\sin(10\pi F_1(\mathbf{x}))\right]$

Table 6.15: Mathematical Expressions of Multi-Objective Benchmark Functions

Function	Pareto Front	Separability	Convexity	Continuity	Primary Test Objective
ZDT1	Convex	Separable	Convex	Continuous	Baseline convergence to Pareto front
ZDT2	Non-convex	Separable	Non-convex	Continuous	Handling non-convex Pareto fronts
ZDT3	Disconnected	Separable	Convex (per segment)	Discontinuous	Maintaining diversity across disconnected regions

Table 6.16: Characteristics of Multi-Objective Benchmark Functions and Properties Tested

All benchmark problems were solved for 10 design variables bounded in $[0, 1]$. Each decision variable was encoded using a 10-bit binary representation. Experiments were conducted with a computational budget of 15,000 function evaluations over 3 independent runs with different random seeds.

For the QIEA, a parameter grid search was conducted across multiple configurations. Table 6.17 presents the parameter values explored. Parent selection employed a linear-weighted method, and mutation probabilities for both measurement and rotation operations were set to 0.0.

Similarly, for the EA, a parameter grid search was performed. Table 6.18 presents the parameter values explored.

The best-performing parameter configurations identified through the grid search are presented in Tables 6.19 and 6.20. Figures 6.10, 6.11, and 6.12 compare the best-

performing QIEA configurations against the most competitive EA configurations identified through grid search.

Parameter	Notation	Values
Parent population size	μ	$\{20, 30, 40\}$
Offspring population size	λ	$\{20, 30, 40\}$
Elites' population size	e	$\{20, 30, 40\}$
Measurements per individual	m/i	$\{1, 2\}$
Q-Gate Angle of rotation	$\Delta\theta$	$\{0.1, 0.25, 0.5\}$
Crossover period	T_{x}	$\{5, 6, 10\}$
Crossover probability	p_{x}	$\{0.9, 1.0\}$

Table 6.17: Parameter Grid for QIEA Multi-Objective Experiments

Parameter	Notation	Values
Parent population size	μ	$\{10, 20, 30\}$
Offspring population size	λ	$\{20, 40, 60\}$
Elitism rate	r_{elite}	$\{0.1, 0.2\}$
Crossover probability	p_x	$\{0.9, 0.95\}$
Mutation probability	p_m	$\{0.02\}$

Table 6.18: Parameter Grid for EA Multi-Objective Experiments

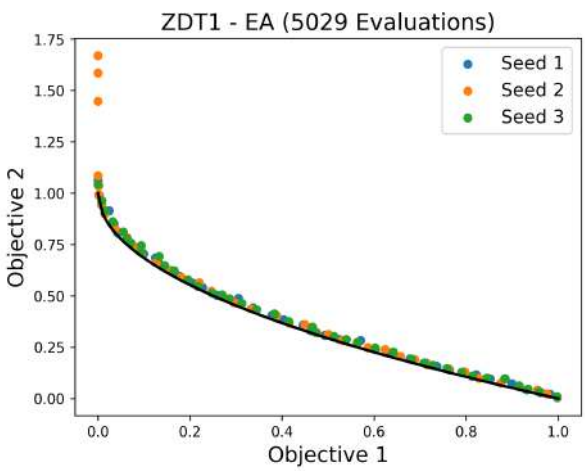
Function	μ	λ	r_e	$p_{\mathbf{c}}$	$p_{\mathbf{m}}$
ZDT1	20	40	0.2	0.90	0.02
ZDT2	20	40	0.1	0.90	0.02
ZDT3	10	20	0.2	0.95	0.02

Table 6.19: Best-Performing EA Configurations for Multi-Objective Benchmark Problems

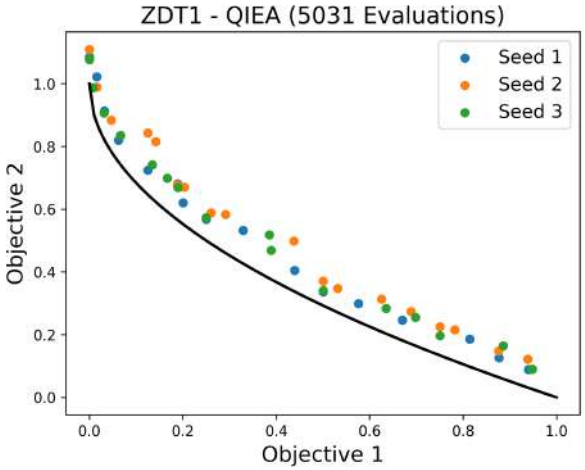
Function	μ	λ	e	m/i	$\Delta\theta$	$T_{\mathbf{x}}$	$p_{\mathbf{x}}$
ZDT1	20	20	20	2	0.25	6	1.0
ZDT2	20	20	20	2	0.25	6	1.0
ZDT3	40	40	40	2	0.25	6	1.0

Table 6.20: Best-Performing QIEA Configurations for Multi-Objective Benchmark Problems

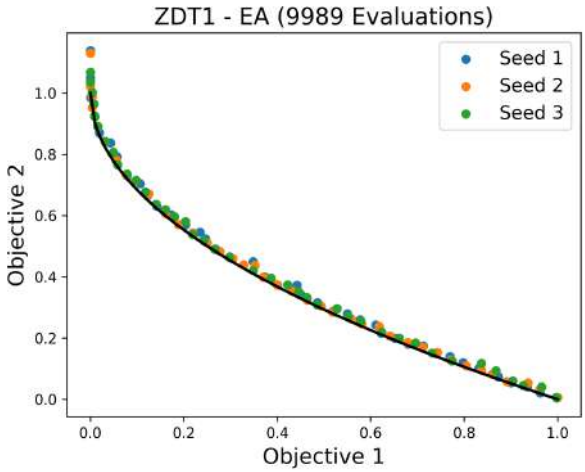
Observations The results for the multi-objective optimization indicate that its convergence is slower than that of the EA. The EA achieves a comparable degree of proximity to the true Pareto front and a denser Pareto front than QIEA using only 30% or 50% of the evaluations required by QIEA. It appears that there is significant room for improvement in the multi-objective QIEA.



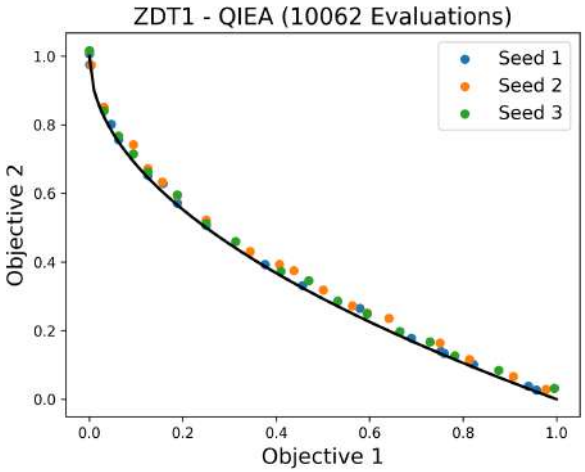
(a) EA at 33% of total evaluations



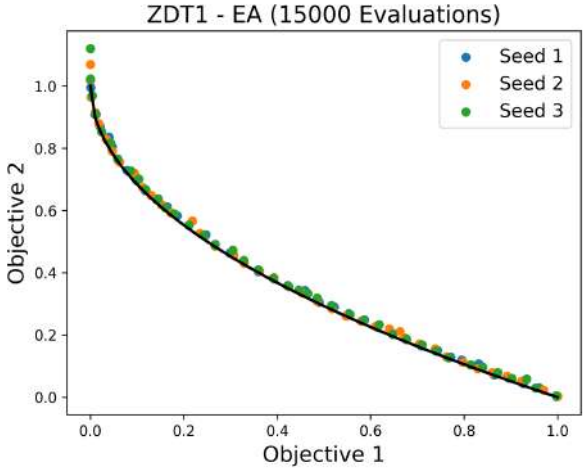
(b) QIEA at 33% of total evaluations



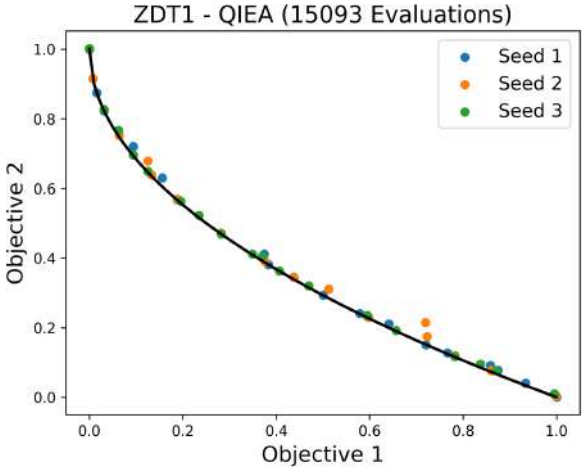
(c) EA at 67% of total evaluations



(d) QIEA at 67% of total evaluations

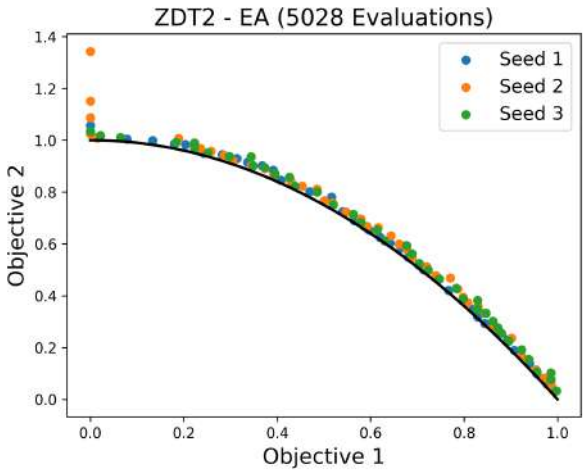


(e) EA at 100% of total evaluations

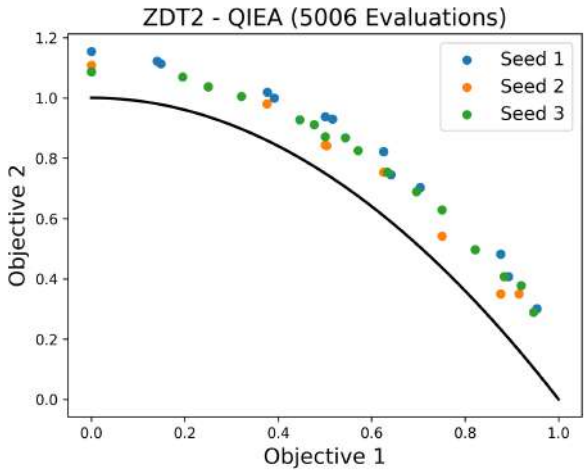


(f) QIEA at 100% of total evaluations

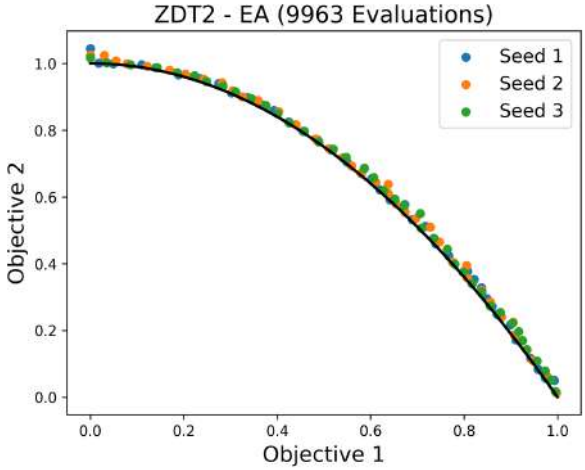
Figure 6.10: Pareto front approximation for ZDT1 at 33%, 67%, and 100% of total function evaluations, comparing EA and QIEA



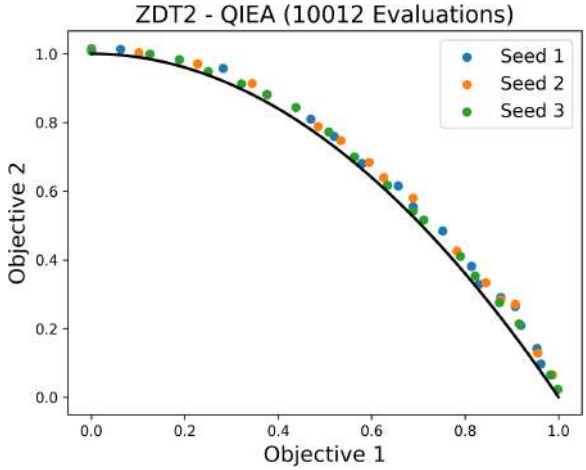
(a) EA at 33% of total evaluations



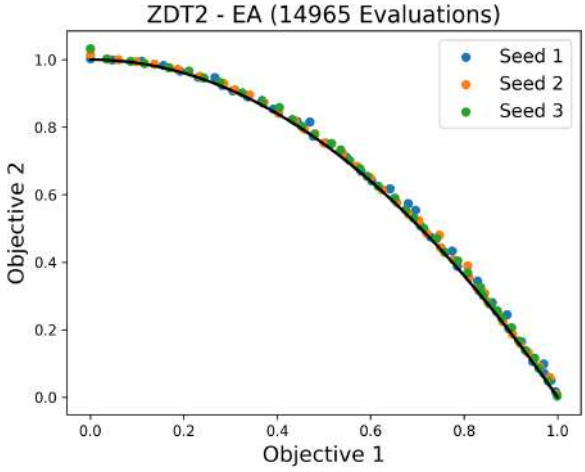
(b) QIEA at 33% of total evaluations



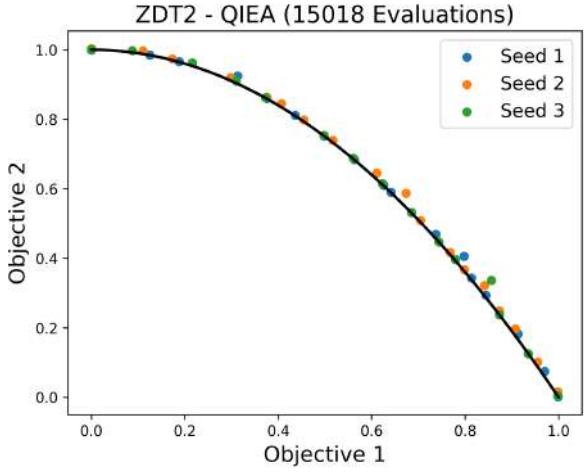
(c) EA at 67% of total evaluations



(d) QIEA at 67% of total evaluations

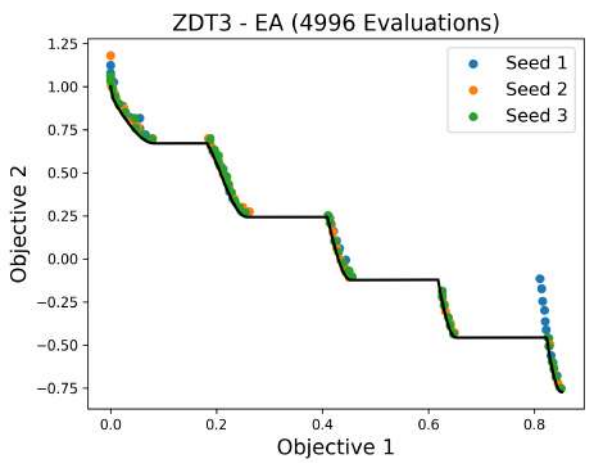


(e) EA at 100% of total evaluations

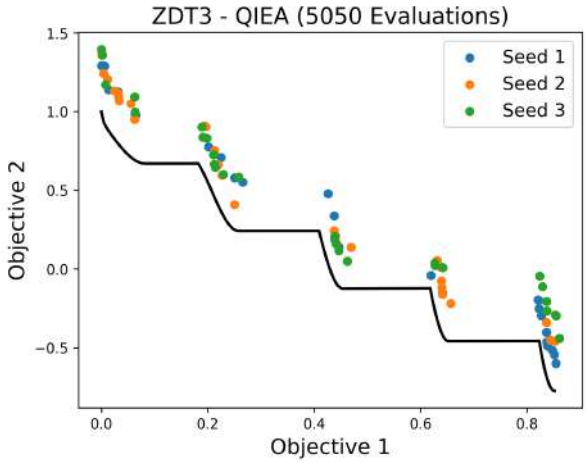


(f) QIEA at 100% of total evaluations

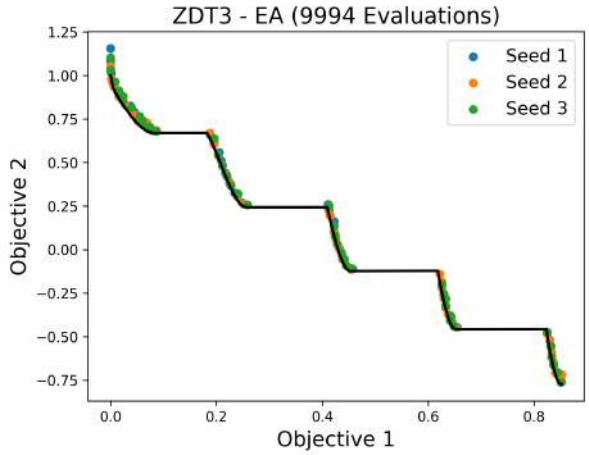
Figure 6.11: Pareto front approximation for ZDT2 at 33%, 67%, and 100% of total function evaluations, comparing EA and QIEA



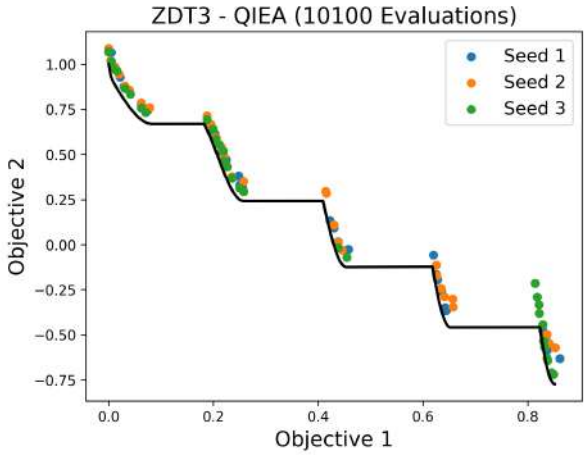
(a) EA at 33% of total evaluations



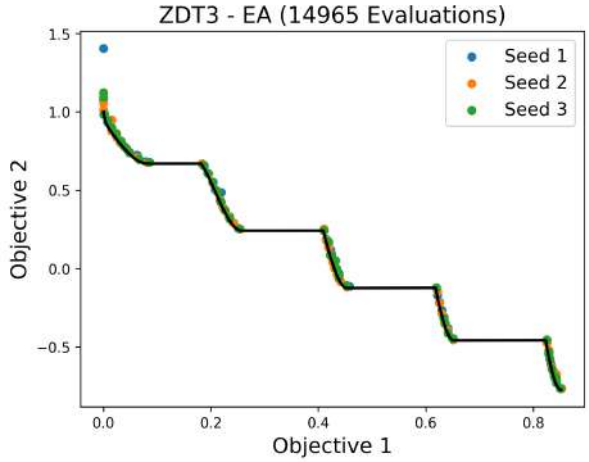
(b) QIEA at 33% of total evaluations



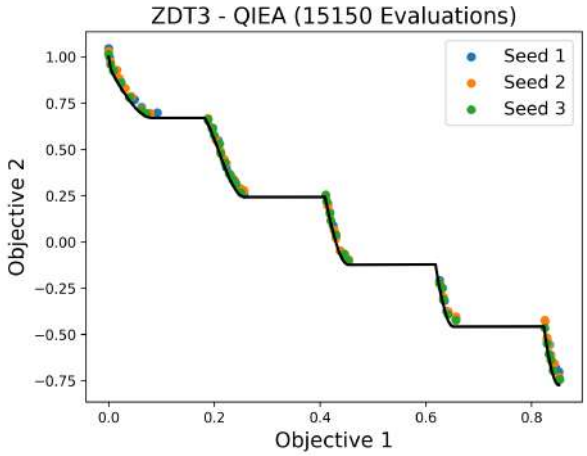
(c) EA at 67% of total evaluations



(d) QIEA at 67% of total evaluations



(e) EA at 100% of total evaluations



(f) QIEA at 100% of total evaluations

Figure 6.12: Pareto front approximation for ZDT3 at 33%, 67%, and 100% of total function evaluations, comparing EA and QIEA

Chapter 7

Conclusions

We observe that QIEA has a large number of parameters, and the convergence of the algorithm appears to be sensitive to them. We also observe strong coupling both between the parameters themselves and between the parameters and the optimization problem.The small number of populations for which it works well makes it vulnerable to premature convergence and entrapment in local optima. Nevertheless, in some of the test cases, such as the sphere and the Ackley functions, its convergence speed was satisfactory and comparable to that of the EA. In conclusion, it shows some promising results. However, improvements are needed for it to become more robust and suitable for use in real-world applications.

In this direction, the following proposals are put forward, which could form the subject of future research on the QIEA. As we have explained, the different rotation tables are a good parameterization of how exploratory or exploitative we want the algorithm to be. Different rotation tables could be used for the various individuals during the same optimization so that each individual "chases" the solution with a different strategy (in the same way that the ghosts chase PAC-MAN with different strategies). In this way, on the one hand we could alleviate the elitist character of the QIEA that was observed, and on the other hand we could make the QIEA more adaptable to different optimization problems.

Another possible approach is to apply QIEA to a small population, in combination with EA rather than as a replacement for it. Evidence suggests that QIEA works particularly well with small populations. Its strong exploitative behavior and fast convergence could be effectively utilized if QIEA were employed in a supporting role alongside the classical EA.

Classical evolutionary algorithms have many more years of research and application behind them, so we did not anticipate that QIEA would outperform them. Nonetheless, the results we obtained are encouraging and may potentially contribute to the advancement of the field of optimization.

Bibliography

[1] Sukhpal Singh Gill, Oktay Çetinkaya, Stefano Marrone, Daniel Claudino, David Haunschild, Leon Schlote, Huaming Wu, Carlo Ottaviani, Xiaoyuan Liu, Sree Machupalli, Kamalpreet Kaur, Priyansh Arora, Ji Liu, Ahmed Farouk, Houbing Song, Steve Uhlig, and Kotagiri Ramamohanarao. *Quantum computing: vision and challenges*, pages 19–42. 01 2025.

[2] Giulio Malinverno and Javier Alberto. A review of the current state-of-the-art of quantum computing for cfd: Approaches, advantages, and limitations. *Aerotecnica Missili & Spazio*, 06 2025.

[3] Zhen Lu and Yue Yang. Quantum computing of reacting flows via hamiltonian simulation. *Proceedings of the Combustion Institute*, 40:105440, 07 2024.

[4] Paul Steinacker, Nard Stuyck, Wee Lim, Tuomo Tanttu, Mengke Feng, Santiago Serrano, Andreas Nickl, Marco Candido, Jesus Cifuentes, Ensar Vahapoglu, Samuel Bartee, Fay Hudson, Kok Chan, Stefan Kubicek, Julien Jussot, Yann Canvel, Sofie Beyne, Yosuke Shimura, R. Loo, and Andrew Dzurak. Industry-compatible silicon spin-qubit unit cells exceeding 99 *Nature*, 646:81–87, 09 2025.

[5] Paul Klimov, Andreas Bengtsson, Chris Quintana, Alexandre Bourassa, Sabrina Hong, Andrew Dunsworth, Kevin Satzinger, William Livingston, Volodymyr Sivak, Yuezhen Niu, Trond Andersen, Yaxing Zhang, Desmond Chik, Zijun Chen, Charles Neill, Catherine Erickson, Alejandro Dau, Anthony Megrant, Pedram Roushan, and Hartmut Neven. Optimizing quantum gates towards the scale of logical qubits. *Nature Communications*, 15, 03 2024.

[6] R Barends, J Kelly, A Megrant, Andrzej Veitia, D Sank, E Jeffrey, T White, J Mutus, Austin Fowler, B Campbell, Yu Chen, Zhaoshi Chen, B Chiaro, A Dunsworth, C Neill, P O'Malley, P Roushan, A Vainsencher, J Wenner, and John Martinis. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, 508:500–3, 04 2014.

[7] Javier Gonzalez-Conde, Dylan Lewis, Sachin Bharadwaj, and Mikel Sanz. Quantum carleman linearization efficiency in nonlinear fluid dynamics. *Physical Review Research*, 7, 06 2025.

[8] Shahin Hakemi, Mahboobeh Houshmand, Esmaeil KheirKhah, and Seyyed Abed Hosseini. A review of recent advances in quantum-inspired metaheuristics. *Evolutionary Intelligence*, 17:627–642, 10 2022.

[9] Kuk-Hyun Han and Jong-Hwan Kim. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *Evolutionary Computation, IEEE Transactions on*, 6:580 – 593, 01 2003.

[10] Guanghui Zhang, Wenjing Ma, Keyi Xing, Lining Xing, and Kesheng Wang. Quantum-inspired distributed memetic algorithm. *Complex System Modeling and Simulation*, 2(4):334–353, 2022.

[11] Kuk-Hyun Han and Jong-Hwan Kim. Quantum-inspired evolutionary algorithms with a new termination criterion, $h_epsilon$ gate, and two-phase scheme. *Evolutionary Computation, IEEE Transactions on*, 8:156 – 169, 05 2004.

[12] Zhou Liang, Shao Ming, and Ma Chengqian. Adaptive collaborative quantum-inspired evolutionary algorithm for global numerical functions. *ITM Web of Conferences*, 16:02010, 01 2018.

[13] Md. Amjad Hossain, Md. Kowsar Hossain, and M.M.A Hashem. A generalized hybrid real-coded quantum evolutionary algorithm based on particle swarm theory with arithmetic crossover. *International Journal of Computer Science and Information Technology*, 2(4):172–187, August 2010.

[14] Hichem Talbi and Amer Draa. A new real-coded quantum-inspired evolutionary algorithm for continuous optimization. *Applied Soft Computing*, 61:765–791, 2017.

[15] P. Dirac. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35:416 – 418, 07 1939.

[16] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition, 2010.

[17] Max Born. On the quantum mechanics of collisions. In John A. Wheeler and Wojciech H. Zurek, editors, *Quantum Theory and Measurement*, pages 52–55. Princeton University Press, 1983.

[18] Mark M. Wilde. *Quantum Information Theory*. Cambridge University Press, 2nd edition, 2017.

[19] Eleanor G. Rieffel and Wolfgang H. Polak. *Quantum Computing: A Gentle Introduction*. MIT Press, 2011.

[20] N. David Mermin. *Quantum Computer Science: An Introduction*. Cambridge University Press, 2007.

[21] John Wetering. Constructing quantum circuits with global gates. *New Journal of Physics*, 23, 04 2021.

[22] K. Giannakoglou, D. Tsahalis, Jacques Periaux, K. Papailiou, T. (eds, and Eckart Zitzler. Evolutionary algorithms for multiobjective optimization. 11 2001.

[23] Shanu Verma, Millie Pant, and Vaclav Snasel. A comprehensive review on nsga-ii for multi-objective combinatorial optimization problems. *IEEE Access*, 9, 04 2021.

[24] Marios Karakasis and Kyriakos Giannakoglou. On the use of surrogate evaluation models in multi-objective evolutionary algorithms. *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS*, pages 24–28, 08 2004.

[25] Dimitrios H. Kapsoulis. *Low-Cost Metamodel-Assisted Evolutionary Algorithms with Application in Shape Optimization in Fluid Dynamics*. PhD thesis, National Technical University of Athens, 2019.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης

Εξελικτικοί Αλγόριθμοι Εμπνευσμένοι από την Κβαντομηχανική, Υποστηριζόμενοι από Μεταπρότυπα

Διπλωματική Εργασία
Εκτενής Περίληψη στην Ελληνική

Ιωάννης Α. Γερογιάννης

Επιβλέποντες: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ,
Δρ. Β. Ασούτη, Εντεταλμένη Διδάσκουσα ΕΜΠ

Αθήνα, Φεβρουάριος 2026

Περιεχόμενα

1 Εισαγωγή **1**

2 Θεμελιώδης Αρχές Κβαντομηχανικής &Κβαντικής Υπολογιστικής **3**
2.1 Θεμελιώδης Μονάδα Πληροφορίας - Qubit 3
2.2 Μαθηματική Αναπαράσταση και Σημειολογία Bra-Ket 3
2.3 Υπέρθεση, Κβαντική Κατάσταση και Μέτρηση 4
2.4 Γεωμετρική Αναπαράσταση: Η Σφαίρα Bloch 5
2.5 Κβαντικές Πύλες 5

3 Εξελικτικοί αλγόριθμοι **7**
3.1 Περιγραφή ενός Εξελικτικού Αλγορίθμου 7
3.2 Αλγοριθμική Διαδικασία 8
3.3 Εξελικτικοί Αλγόριθμοι στην πολυκριτηριακή βελτιστοποίηση 8

4 QIEA **9**
4.1 Αναπαράσταση του Ατόμου στον QIEA 9
4.2 Υλοποίηση μονοκριτιριακού QIEA 10
4.2.1 Μέτρηση &Αξιολόγηση 10
4.2.2 Αρχικοποίηση 11
4.2.3 Πύλη Κβαντικής Περιστροφής (Q-Gate Rotation) 11
4.2.4 Τελεστής Κβαντικού Crossover 12
4.2.5 Μετάλλαξη 12
4.2.6 Μετανάστευση 12
4.2.7 Επιλογή και Ελιτισμός 12
4.2.8 Μηχανισμός Στασιμότητας 13
4.3 Αλγόριθμος QIEA 13
4.4 Multi-Objective QIEA 14
4.4.1 Προσαρμογή Τελεστή Q-Gate 14
4.4.2 Ταξινόμηση και Διαχείριση Πληθυσμού με NSGA-II 14
4.4.3 Δομικές Διαφορές από το Μονο-αντικειμενικό QIEA 14
4.4.4 Αλγοριθμική Διαδικασία 15

5 QIEA Υποβοηθούμενος από Μεταμοντέλα **16**
5.1 Δίκτυα Ακτινικής Βάσης (Radial-Basis Function Networks) 16

6 Αξιολόγηση Απόδοσης και Σχολιασμός Αποτελεσμάτων **18**

7 Συμπεράσματα **23**

Κεφάλαιο 1

Εισαγωγή

Ο τομέας της κβαντικής υπολογιστικής έχει προσελκύσει αρκετό ενδιαφέρον από την επιστημονική κοινότητα τον τελευταίο καιρό. Αποτελεί μία διαφορετική προσέγγιση στην ανάπτυξη υπολογιστικών συστημάτων που βασίζονται στις αρχές της κβαντομηχανικής. Οι διαφορετικές ιδιότητες των κβαντικών υπολογιστών διαφαίνονται υποσχόμενες στην αντιμετώπιση και επίλυση πολύπλοκων και υπολογιστικά απαιτητικών προβλημάτων, τα οποία τα συμβατικά δυαδικά υπολογιστικά συστήματα που διαθέτουμε σήμερα αδυνατούν να επιλύσουν.

Χαρακτηριστικά παραδείγματα αποτελούν η προσομοίωση κβαντιμηχανικών συστημάτων ή οι προσομοιώσεις μοριακής δυναμικής. Ένα άλλο απαιτητικό πρόβλημα για τους συμβατικούς υπολογιστές είναι η ευθεία αριθμητική προσομοίωση (DNS) τυρβώδους ροής. Το κοινό χαρακτηριστικό αυτών των προβλημάτων που τα καθιστά δυσεπίλυτα είναι το μέγεθος τους και οι υψηλές απαιτήσεις σε μνήμη και υπολογιστική ισχύ. Συγκεκριμένα προκειμένου να αναπαρασταθεί ένα κβαντομηχανικό σύστημα n μορίων χρειάζεται να αποθηκεύσουμε 2^n μιγαδικά αριθμούς. Ομοίως για την ευθεία προσομοίωση μίας τυρβώδους ροής με υψηλό αριθμό Reynolds χρειάζεται να επιλύσουμε της εξισώσεις Navier-Stokes για ένα πλέγμα τόσο πύκνο ώστε να μπορούν να μοντελοποιηθούν και οι πιο μικρές δίνες της ροής. Είναι πρακτικά αδύνατο να αναπαρασταθούν, πόσο μάλλον να επιλυθούν, με χρήση ενός δυαδικού υπολογιστή, και τα δύο παραπάνω προβλήματα για διαστάσεις και συνθήκες σε εφαρμογές του πραγματικού κόσμου.

Το χαρακτηριστικό των κβαντικών υπολογιστών που μας κάνει να πιστεύουμε ότι θα μπορούσε δυνητικά να επιλύσει τέτοιου είδους απαιτητικά προβλήματα, είναι ο τρόπος αναπαράστασης της ψηφιακής πληροφορίας. Εκμεταλλευόμενος κβαντομηχανικές ιδιότητες όπως η υπέρθεση, ένας κβαντικός υπολογιστικής έχει τη δυνατότητα να αναπαραστήσει μεγάλο όγκο πληροφορίας με χρήση πολύ λίγων qubit, το κβαντικό ανάλογο των bit. Συγκεκριμένα ο μεγαλύτερος κβαντικός υπολογιστής σήμερα (IBM Condor) διαθέτι 1121 qubits. Προκειμένου να αναπαραστήσουμε τον ίδιο όγκο πληροφορίας σε έναν δυαδικό υπολογιστή θα χρειαζόμασταν κατά προσέγγιση $2 \cdot 10^{317} ZB$ αποθηκευτικού χώρου. Ενδεικτικά, ο συνολικός όγκος ψηφιακής πληροφορίας ολόκληρου του διαδικτύου στα τέλη του 2025 υπολογίζεται στα $180 ZB$.

Αν και η κβαντική υπολογιστική δείχνει πως θα μπορούσε να μας φανεί χρήσιμη, απαιτείται η επίλυση δυσκολιών και εμποδίων πριν να είναι σε θέση να εφαρμοστεί σε πρακτικά και πραγματικά προβλήματα [1]. Η περίοδος που διανύουμε χαρακτηρίζεται ως περίοδος NISQ (Noisy Intermediate-Scale Quantum). Προς το παρόν, η εφαρμογή τους υπόκειται σε περιορισμούς που επιβάλλονται από το υλικό (hardware). Τα κβαντομηχανικά υπολογιστικά συστήματα που υπάρχουν σήμερα διαθέτουν μερικές εκατοντάδες qubits, τα οποία είναι ευαίσθητα σε περιβαλλοντικούς θορύβους. Λόγω αυτών των θορύβων υπεισέρχονται σφάλματα που συσσωρεύονται και οδηγούν σε υπολογισμούς με σημαντικές αποκλίσεις. Η διόρθωση

κβαντικών σφαλμάτων είναι μία προσέγγιση αντιμετώπισης αυτών των αδυναμιών του υλικού με χρήση πολλών φυσικών qubit τα οποία συνθέτουν ένα λογικό qubit, κατ' αναλογία με τον πλεονασμό που χρησιμοποιείται στην ανίχνευση και διόρθωση σφαλμάτων στους δυαδικούς υπολογιστές. Ωστόσο, η διόρθωση κβαντικών σφαλμάτων είναι πιο δύσκολη καθώς η κβαντομηχανική πληροφορία δε μπορεί να αντιγραφεί βάσει του θεωρήματος μη κλωνοποίησης.

Ένα άλλο εμπόδιο που πρέπει να ξεπεραστεί ώστε οι κβαντικοί υπολογιστές να έχουν ευρεία χρήση, είναι η προσαρμογή των προβλημάτων που καλούνται να επιλύσουν σε μία μορφή που μπορούν να διαχειριστούν. Οι κβαντικοί υπολογιστούν υπόκεινται και υπακούουν στους νόμους της κβαντομηχανικής. Αυτό τους καθιστά κατάλληλους και ιδιαίτερα ικανούς στην επίλυση και προσομοίωση άλλων κβαντομηχανκών συστημάτων ή μοριακών φαινομένων που διέπονται από τους ίδιους νόμους. Η εξίσωση Schrodinger η οποία κρύβεται πίσω από την αρχή λειτουργίας των κβαντομηχανικών συστημάτων είναι γραμμική, συνεπώς και ο κβαντικός υπολογιστής μπορεί να αντιμετωπίζει με ευχέρεια γραμμικά συστήματα. Αντίθετα οι Navier-Stokes λόγου χάριν είναι μη-γραμμικές, και ο κβαντικός υπολογιστής αδυνατεί να τις διαχειριστεί. Για αυτό το σκοπό αναπτύσσονται μέθοδοι και αλγόριθμοι, μέσω των οποίων η κβαντική υπολογιστική μπορεί να βρει εφαρμογή σε μεγαλύτερο εύρος προβλημάτων. Ένα παράδειγμα τέτοιου αλγορίθμου είναι η μέθοδος κβαντικής γραμμικοποίησης Carleman (quantum Carleman linearization - QCL).

Δεδομένων των παραπάνω δυσκολιών, αναπτύσσονται ταυτόχρονα υβριδικές τεχνικές οι οποίες αξιοποιούν τον κβαντικό υπολογιστή για το τμήμα της επίλυσης ενός προβλήματος στο οποίο είναι πιο αποδοτικός, ενώ τις υπόλοιπες διεργασίες τις διαχειρίζεται ένας δυαδικός υπολογιστής. Παράδειγμα αυτής της υβριδικής προσέγγισης είναι ο Variational Quantum Eigensolver (VQE). Η μέθοδος αρχικά αναπτύχθηκε για τον υπολογισμό της ενέργειας θεμελιώδους κατάστασης μορίων. Ο κβαντικός επεξεργαστής προετοιμάζει μια υποψήφια κβαντική κατάσταση και μετρά την ενέργειά της. Έπειτα, ένας κλασικός αλγόριθμος βελτιστοποίησης ρυθμίζει τις παραμέτρους ώστε να ελαχιστοποιήσει την ενέργεια μέσω επαναλαμβανόμενων επαναλήψεων. Ο αλγόριθμος μπορεί να χρησιμοποιηθεί για την προσομοίωση μορίων, ενώ στη βιβλιογραφία έχει προταθεί ως τρόπος επίλυσης διακριτοποιημένων γραμμικών συστημάτων που έχουν προκύψει από την εξίσωση Poisson σε προβλήματα ρευστοδυναμικής.

Μία άλλη κατηγορία αλγορίθμων, που επιχειρούν να αξιοποιήσουν τις κβαντομηχανικές ιδιότητες κατά την περίοδο NISQ είναι οι quantum-inspired αλγόριθμοι (QIA). Αυτοί οι αλγόριθμοι στοχεύουν στην αξιοποίηση των βασικών ιδιοτήτων των κβαντομηχανικών υπολογιστικών συστημάτων προς την ανάπτυξη νέων αλγορίθμων οι οποίοι εκτελούνται σε δυαδικούς υπολογιστές και ενδεχομένως μπορούν να επωφεληθούν από αυτές τις ιδιότητες. Σε αυτή την κατηγορία αλγορίθμων ανήκει και ο Quantum Inspired Evolutionary Algorithm (QIEA) [2] [3]. Κάθε άτομο του πληθυσμού δεν αποτελεί πλέον μία μοναδική υποψήφια λύση στο χώρο αναζήτησης αλλά κατανομή πιθανότητας ανίχνευσης αυτής σε μία περιοχή του χώρου αναζήτησης. Αυτή η εγγενής πιθανοκρατικής κωδικοποίηση θα μπορούσε δυνητικά να οδηγήσει σε πιο αποδοτική εξερεύνηση και εκμετάλλευση του χώρου αναζήτησης.

Για τις ανάγκες της εργασίας εξελίχθηκε και χρησιμοποιήθηκε κώδικας σε γλώσσα Python.

Κεφάλαιο 2

Θεμελιώδης Αρχές Κβαντομηχανικής & Κβαντικής Υπολογιστικής

2.1 Θεμελιώδης Μονάδα Πληροφορίας - Qubit

Σε έναν συμβατικό διαδικό υπολογιστή, η πλέον στοιχειώδης μονάδα πληροφορίας, το bit μπορεί να βρίσκεται σε μία από τις δύο καταστάσεις: 0 ή 1. Αυτές οι δύο καταστάσεις έχουν φυσική σημασία και μπορούν να αναπαρασταθούν από δύο διακριτές καταστάσεις τάσης σε έναν ηλεκτρικό κύκλωμα.

Σε έναν κβαντικό υπολογιστή, την πλέον στοιχειώδη μονάδα πληροφορίας αποτελεί το **qubit**, το κβαντικό ανάλογο του bit. Το qubit χαρακτηρίζεται από την **κβαντική κατάσταση** αυτού, η οποία είναι ένας γραμμικός συνδυασμός των δύο ακολούθων βασικών καταστάσεων:

$$\text{κβαντική κατάσταση } 0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{κβαντική κατάσταση } 1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

2.2 Μαθηματική Αναπαράσταση και Σημειολογία Bra-Ket

Η μαθηματική περιγραφή των κβαντικών καταστάσεων βασίζεται στη θεωρία των διανυσματικών χώρων Hilbert. Κάθε κβαντική κατάσταση αναπαρίστανται ως ένα **διάνυσμα κατάστασης** (state vector) σε έναν μιγαδικό διανυσματικό χώρο. Για ένα qubit, αυτός ο χώρος είναι διδιάστατος και συνήθως συμβολίζεται ως $\mathbb{C}^2$.

Σημειολογία Dirac Προς διευκόλυνση της αναπαράστασης, στην κβαντική υπολογιστική και γενικότερα στην κβαντομηχανική, γίνεται χρήση της **σημειολογίας Bra-ket** (Dirac notation), η οποία εισήχθη από τον φυσικό Paul Dirac [4]. Αυτή η σημειολογία βασίζεται στη διχοτόμηση του εσωτερικού γινομένου σε δύο μέρη:

- Τα **ket vectors** $|\psi\rangle \in \mathcal{H}$ αναπαριστούν τις κβαντικές καταστάσεις ως στήλες διανύσματα
- Τα **bra vectors** $\langle\psi| \in \mathcal{H}^*$ αναπαριστούν τους γραμμικούς συναρτησιακούς (dual space) ως γραμμές διανύσματα

- Το **bracket** $\langle\psi|\phi\rangle$ αναπαριστά το εσωτερικό γινόμενο δύο καταστάσεων

Μαθηματικά, αν $|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}$, τότε το αντίστοιχο bra είναι:

$$\langle\psi| = \begin{pmatrix}\alpha^* & \beta^*\end{pmatrix} \tag{2.1}$$

όπου το σύμβολο * δηλώνει τον μιγαδικό συζυγή. Το εσωτερικό γινόμενο δύο καταστάσεων $|\psi\rangle$ και $|\phi\rangle$ υπολογίζεται ως:

$$\langle\psi|\phi\rangle = \begin{pmatrix}\alpha^* & \beta^*\end{pmatrix}\begin{pmatrix}\gamma\\ \delta\end{pmatrix} = \alpha^*\gamma + \beta^*\delta \tag{2.2}$$

Ορθοκανονική Βάση και Υπολογιστική Βάση Οι δύο βασικές καταστάσεις του qubit, αναπαρίστανται ως:

$$|0\rangle = \begin{pmatrix}1\\ 0\end{pmatrix}, \quad |1\rangle = \begin{pmatrix}0\\ 1\end{pmatrix} \tag{2.3}$$

Αυτές οι δύο καταστάσεις σχηματίζουν μια **ορθοκανονική βάση** (orthonormal basis) για τον χώρο των καταστάσεων του qubit, γνωστή ως **υπολογιστική βάση** ή **Z-βάση**. Οι ιδιότητες ορθοκανονικότητας ικανοποιούνται από την υπολογιστική βάση και εκφράζονται ως:

$$\langle 0|0\rangle = \langle 1|1\rangle = 1, \quad \langle 0|1\rangle = \langle 1|0\rangle = 0 \tag{2.4}$$

Επιπλέον, η βάση ικανοποιεί την **σχέση πληρότητας** (completeness relation):

$$|0\rangle\langle 0| + |1\rangle\langle 1| = \mathbf{I} \tag{2.5}$$

όπου $\mathbf{I}$ είναι ο ταυτοτικός πίνακας διάστασης 2×2. Η σχέση πληρότητας εξασφαλίζει ότι οποιοδήποτε qubit μπορεί να εκφραστεί συναρτήσει της υπολογιστικής βάσης.

2.3 Υπέρθεση, Κβαντική Κατάσταση και Μέτρηση

Τα qubits έχουν την ιδιότητα να υπάρχουν σε **υπέρθεση** (superposition). Οι καταστάσεις $|0\rangle$ και $|1\rangle$ δεν αποτελούν τις μοναδικές καταστάσεις που μπορεί να βρίσκεται ένα qubit. Η **κβαντική κατάσταση** του qubit, η οποία συμβολίζεται ως $|\Psi\rangle$, εκφράζεται ως ο γραμμικός συνδυασμός των δύο βασικών καταστάσεων [5]:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} \tag{2.6}$$

όπου $\alpha, \beta \in \mathbb{C}$ είναι τα **πλάτη πιθανότητας** (probability amplitudes) και ικανοποιούν τη **συνθήκη κανονικοποίησης** (normalization condition):

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2.7}$$

Η φυσική σημασία των πλατών είναι η εξής. Ο όρος $|\alpha|^2$ εκφράζει την πιθανότητα μέτρησης της κατάστασης $|0\rangle$ και ο όρος $|\beta|^2$ εκφράζει την πιθανότητα μέτρησης της κατάστασης $|1\rangle$.

Με τον όρο **μέτρηση** αναφερόμαστε στην μη αναστρέψιμη πράξη ανάγνωσης των καταστάσεων των qubit, η οποία εξαναγκάζει τις υπερτιθέμενες, πιθανοκρατικές κυματοσυναρτήσεις τους να καταρρεύσουν σε συγκεκριμένα κλασικά αποτελέσματα $|0\rangle$ ή $|1\rangle$.

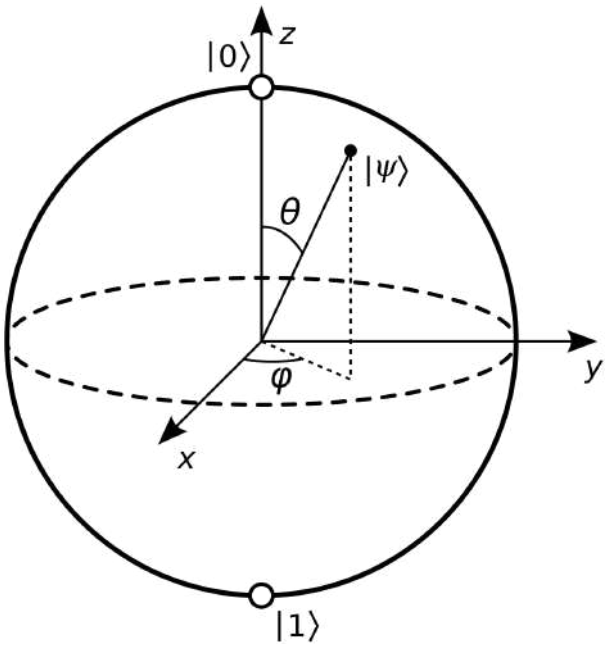


Figure 2.1: Η σφαίρα Bloch

2.4 Γεωμετρική Αναπαράσταση: Η Σφαίρα Bloch

Αποδεικνύεται ότι η κβαντική κατάσταση ενός qubit μπορεί να εκφραστεί σε σφαιρικές συντεταγμένες στη μορφή:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{2.8}$$

Αυτή είναι η τυπική **παραμετροποίηση της σφαίρας Bloch**, όπου οποιαδήποτε κατάσταση qubit καθορίζεται πλήρως μόνο από δύο γωνίες, την πολική γωνία $\theta \in [0, \pi]$ που ορίζει κατά πόσο η κατάσταση «κλίνει» προς $|0\rangle$ ή $|1\rangle$ και τη γωνία αζιμουθίου $\phi \in [0, 2\pi)$ που καθορίζει τη σχετική φάση μεταξύ των πλατών.

Σχηματίζεται έτσι η **σφαίρα Bloch** (Σχήμα 2.1), μια μοναδιαία σφαίρα όπου κάθε δυνατή κατάσταση qubit $|\psi\rangle$ αντιστοιχεί σε ακριβώς ένα σημείο στην επιφάνειά της, το οποίο καθορίζεται από τις γωνίες ϕ και θ. Η κβαντική κατάσταση $|0\rangle$ βρίσκεται στον βόρειο πόλο, η $|1\rangle$ στον νότιο πόλο, ενώ οι καταστάσεις υπέρθεσης καταλαμβάνουν όλα τα άλλα σημεία στην επιφάνεια της σφαίρας.

Οι πιθανότητες μέτρησης συναρτήσει των σφαιρικών συντεταγμένων υπολογίζονται ως εξής:

$$P(|0\rangle) = \cos^2\left(\frac{\theta}{2}\right), \quad P(|1\rangle) = \sin^2\left(\frac{\theta}{2}\right) \tag{2.9}$$

Συνεπώς όσο πιο κοντά στο βόρειο πόλο της σφαίρας βρισκόμαστε τόσο πιο πιθανό είναι να μετρήσουμε την κατάσταση $|0\rangle$, ενώ αντίστοιχα οι καταστάσεις που βρίσκονται πιο κοντά στο νότιο πόλο δίνουν μεγαλύτερη πιθανότητα μέτρησης της κατάστασης $|1\rangle$. Τα σημεία στον ισημερινό δίνουν ίσες πιθανότητες και για τα δύο αποτελέσματα.

2.5 Κβαντικές Πύλες

Οι κβαντικές πύλες αποτελούν δομικά στοιχεία ενός κβαντικού υπολογιστή και μας επιτρέπουν να χειριζόμαστε την κατάσταση των qubit [6]. Οι κβαντικές πύλες υλοποιούν μοναδι-

ακούς (unitary) και, συνεπώς, αναστρέψιμους μετασχηματισμούς. Μαθηματικά, κάθε κβαντική πύλη περιγράφεται από μια **μοναδιακή μήτρα** U, η οποία ικανοποιεί την εξής συνθήκη:

$$U^\dagger U = UU^\dagger = I \tag{2.10}$$

όπου $U^\dagger$ είναι ο ανάστροφος συζυγής του U και I είναι ο μοναδιαίος πίνακας. Αυτή η ιδιότητα διασφαλίζει τη διατήρηση της κανονικοποίησης και την αντιστρεψιμότητα των κβαντικών πράξεων.

Η δράση μιας κβαντικής πύλης U σε μια κατάσταση $|\psi\rangle$ εκφράζεται ως:

$$|\psi'\rangle = U|\psi\rangle \tag{2.11}$$

Υπάρχουν πύλες οι οποίες δρουν σε ένα μόνο qubit η σε πολλαπλά qubit. Στις πύλες ενός qubit ανήκουν οι πύλες Pauli, οι πύλες περιστροφής και η πύλη Hadamard και άλλες όπως η πύλες S και T. Παρουσιάζονται ενδεικτικά κάποιες εξ αυτών.

Πύλη Hadamard Η πύλη Hadamard είναι χρήσιμη για τη δημιουργία υπέρθεσης. Δίνεται η μαθηματική της έκφραση και το αποτέλεσμα της δράσης αυτής στις καταστάσεις της υπολογιστικής βάσης:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \quad H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{2.12}$$

Πύλη Pauli-X (Bit Flip) Η πύλη Pauli-X, γνωστή και ως πύλη NOT, αντιστρέφει την κβαντική κατάσταση: $X|0\rangle = |1\rangle$ και $X|1\rangle = |0\rangle$. Γεωμετρικά, η πύλη Pauli-X πραγματοποιεί μια περιστροφή κατά π γύρω από τον άξονα x της σφαίρας Bloch (Σχήμα 2.1).

$$X = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} \tag{2.13}$$

Η πύλη Pauli-Z εισάγει φάση ίση με -1 στην κατάσταση $|1\rangle$: $Z|0\rangle = |0\rangle$ και $Z|1\rangle = -|1\rangle$. Η πύλη Pauli-Y συνδυάζει τόσο τις πράξεις αντιστροφής bit όσο και αντιστροφής φάσης: $Y|0\rangle = i|1\rangle$ και $Y|1\rangle = -i|0\rangle$. Οι πύλες περιστροφής πραγματοποιούν περιστροφές κατά $\Delta\theta$ γύρω από τους κύριους άξονες της σφαίρας Bloch.

Υπάρχουν και πύλες οι οποίες δρουν σε περισσότερα από ένα qubit. Η πύλη Controlled-NOT (CNOT) είναι μία από τις πιο θεμελιώδεις και βασικές πύλες πολλαπλών qubit. Δρα σε ένα ζεύγος qubits: ένα **qubit ελέγχου** και ένα **qubit στόχου**. Η πύλη εφαρμόζει μια πράξη Pauli-X στο qubit στόχου αν και μόνο αν το qubit ελέγχου βρίσκεται στην κατάσταση $|1\rangle$.

Άλλες πύλες πολλαπλών qubit είναι η πύλη Controlled-Z (CZ), η οποία εφαρμόζει μια αναστροφή φάσης στο qubit-στόχο υπό την προϋπόθεση ότι το qubit ελέγχου βρίσκεται στην κατάσταση $|1\rangle$, η πύλη SWAP, η οποία ανταλλάσσει τις καταστάσεις δύο qubit, η πύλη Toffoli, μια τριπλή πύλη Controlled-Controlled-NOT, η πύλη Fredkin (μια Controlled-SWAP gate) και άλλες.

Κεφάλαιο 3

Εξελικτικοί αλγόριθμοι

Για ένα συνεχές πρόβλημα βελτιστοποίησης που ορίζεται ως

$$\min_{\mathbf{x}\in\mathcal{D}} F(\mathbf{x}), \tag{3.1}$$

όπου $F : \mathbb{R}^m \to \mathbb{R}$ η αντικειμενική συνάρτηση προς ελαχιστοποίηση, $\mathbf{x} = (x_1, x_2, \ldots, x_m)^T$ το m-διάστατο διάνυσμα μεταβλητών σχεδιασμού και $\mathcal{D} \subseteq \mathbb{R}^m$ ο εφικτός χώρος αναζήτησης λύσεων. Η καταλληλότητα μιας υποψήφιας λύσης $\mathbf{x}_j$ αξιολογείται υπολογίζοντας τη συνάρτηση $F(\mathbf{x}_j)$, όπου για πρόβλημα ελαχιστοποίησης οι μικρότερες τιμές υποδηλώνουν καλύτερες λύσεις.

3.1 Περιγραφή ενός Εξελικτικού Αλγορίθμου

Οι εξελικτικοί αλγόριθμοι οδηγούνται προς καλύτερες λύσεις του χώρου αναζήτησης μέσω μίας επαναλαμβανόμενης εξελικτικής διαδικασίας που εφαρμόζεται σε έναν πληθυσμό υποψηφίων λύσεων. Ένα σύνολο λύσεων-γονέων αξιοποιούνται για την παραγωγή απογόνων μέσω τελεστών εμπνευσμένων από τη βιολογική εξέλιξη. Τα καταλληλότερα άτομα από το σύνολο των απογόνων επιλέγονται ως γονείς για την επόμενη γενιά. Αυτός ο επαναληπτικός βρόχος συνεχίζεται μέχρι να ικανοποιηθεί ένα κριτήριο τερματισμού, όπως η απουσία περαιτέρω προόδου, η σύγκλιση του πληθυσμού σε παρόμοια άτομα, ή η εξάντληση ενός προκαθορισμένου υπολογιστικού προϋπολογισμού. Οι μεταβλητές σχεδιασμού μπορούν να κωδικοποιηθούν σε δυαδική ή πραγματική μορφή. Στη δυαδική αναπαράσταση, κάθε μεταβλητή αντιστοιχεί σε μια αλληλουχία bit που λειτουργεί ως χρωμόσωμα για τις εξελικτικές διεργασίες του αλγορίθμου.

Η εξελικτική διαδικασία απαρτίζεται από τέσσερις κύριους τελεστές. Ο τελεστής επιλογής χρησιμοποιείται για την επιλογή γονέων βάσει της καταλληλότητάς τους, χρησιμοποιώντας τεχνικές όπως η roulette wheel selection, η rank selection ή tournament selection. Ο τελεστής crossover στη συνέχεια παράγει νέους απογόνους, ανταλλάσσοντας τμήματα των χρωμοσωμάτων από ζεύγη γονέων. Η μετάλλαξη πραγματοποιεί τυχαίες τροποποιήσεις αναστρέφοντας μεμονωμένα bits με μικρή πιθανότητα, διατηρώντας έτσι την ποικιλομορφία και μειώνοντας τον κίνδυνο πρόωρης σύγκλισης. Τέλος, ο ελιτισμός διατηρεί τις κορυφαίες λύσεις στο πέρας των γενεών, διασφαλίζοντας ότι η συνολική ποιότητα της λύσης δεν υποβαθμίζεται με την πάροδο του χρόνου.

3.2 Αλγοριθμική Διαδικασία

Παρουσιάζεται η αλγοριθμική διαδικασία ενός εξελικτικού αλγορίθμου.

1. **Αρχικοποίηση**: Επιλέγονται τιμές για την παραμετροποίηση του αλγορίθμου (μ και λ). Τα μέλη του αρχικού πληθυσμού $S^{0,\lambda}$ επιλέγονται τυχαία
2. **Κύριος Βρόχος Εξέλιξης**: Όσο δεν ικανοποιείται κανένα κριτήριο τερματισμού:
   (a) **Αξιολόγηση**: Αξιολογούνται λ άτομα με χρήση της αντικειμενικής συνάρτησης
   (b) **Επιλογή Ελίτ**: Ο πληθυσμός των ελίτ $S^{g,e}$ ενημερώνεται μετά την αξιολόγηση των νέων ατόμων
   (c) **Εφαρμογή Τελεστή Ελιτισμού**: Ορισμένα μέλη του γονικού πληθυσμού $S^{g,\mu}$ αντικαθίστανται από επίλεκτα άτομα
   (d) **Δημιουργία Απογόνων**: Η ακόλουθη ακολουθία τελεστών εφαρμόζεται μέχρι να παραχθούν λ απόγονοι
      i. **Επιλογή Γονέων**
      ii. **Crossover**
      iii. **Μετάλλαξη**

3.3 Εξελικτικοί Αλγόριθμοι στην πολυκριτηριακή βελτιστοποίηση

Δίνεται η μαθηματική διατύπωση για προβλήματα ελαχιστοποίησης πολλαπλών στόχων.

$$min\mathbf{F}(\mathbf{x}), \mathbf{F} : \mathbb{R}^m \rightarrow \mathbb{R}^M \tag{3.2}$$

όπου M ο αριθμός των στόχων ή αντικειμενικών συναρτήσεων.

Κυριαρχία κατά Pareto και Μέτωπο Pareto Μια μεγάλη οικογένεια εξελικτικών αλγορίθμων για πολυκριτηριακή βελτιστοποίηση κάνει χρήση της αρχής της **κυριαρχίας κατά Pareto** [7]. Εξετάζοντας δύο υποψήφιες λύσεις $\mathbf{x}$ και $\mathbf{y}$, σύμφωνα με τον ορισμό της κυριαρχίας κατά Pareto:

$$\begin{cases} F_i(\mathbf{x}) \leq F_i(\mathbf{y}) \ \forall \ i \in \{1, 2, \ldots, M\} \\ \exists \ j \in \{1, 2, \ldots, M\} : F_j(\mathbf{x}) < F_j(\mathbf{y}) \end{cases} \tag{3.3}$$

όπου η σημειογραφία $\mathbf{x} \prec \mathbf{y}$ σημαίνει ότι το $\mathbf{x}$ κυριαρχεί επί του $\mathbf{y}$.

Το **μέτωπο Pareto** ορίζεται ως το υποσύνολο των υποψήφιων λύσεων που δεν κυριαρχούνται από καμία άλλη λύση. Δύο λύσεις του μετώπου Pareto δεν μπορούν να συγκριθούν μεταξύ τους. Ως εκ τούτου, οι πολυκριτηριακές μέθοδοι βελτιστοποίησης που βασίζονται στην έννοια του Pareto δεν επιστρέφουν μία μόνο λύση, αλλά ένα σύνολο λύσεων που σχηματίζουν ένα μέτωπο Pareto.

Έχουν αναπτυχθεί διάφορες πολυκριτηριακές μέθοδοι βελτιστοποίησης βασισμένες στο Pareto, οι οποίες επιχειρούν να ικανοποιήσουν τις παραπάνω απαιτήσεις. Μερικές από τις πιο γνωστές και ευρέως χρησιμοποιούμενες είναι η Non-dominated Sorting Genetic Algorithm II (NSGA-II), η Strength Pareto Evolutionary Algorithm 2 (SPEA2) και η Pareto Archived Evolution Strategy (PAES).

Κεφάλαιο 4

QIEA

4.1 Αναπαράσταση του Ατόμου στον QIEA

Στην παρούσα εργασία χρησιμοποιείται δυαδική κωδικοποίηση. Κάθε άτομο του πληθυσμού q αναπαρίσταται από μία ακολουθία από qubit, όπου κάθε qubit περιγράφεται από την σχέση (2.6).

$$\mathbf{q} = \left[\begin{pmatrix} \alpha_0 \\ \beta_0 \end{pmatrix} \quad \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \quad \cdots \quad \begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix} \quad \cdots \quad \begin{pmatrix} \alpha_{m-1} \\ \beta_{m-1} \end{pmatrix} \right] \tag{4.1}$$

Με χρήση της παραμετροποίησης Bloch (2.8 και με την παραδοχή ότι τα πλάτη α και β έχουν μηδενικό μιγαδικό μέρος ($\varphi = 0$) λαμβάνουμε τη εξής απλοποιημένη σχέση:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \left(\frac{\theta}{2}\right)|1\rangle \tag{4.2}$$

Εφαρμόζοντας την παραμετροποίηση $\theta = \tilde{\theta} + \frac{\pi}{2}$ και μετονομάζοντας τη νέα παράμετρο $\tilde{\theta}$ σε θ η σχέση μετασχηματίζεται ως εξής:

$$|\psi\rangle = \cos\left(\frac{\tilde{\theta}}{2} + \frac{\pi}{4}\right)|0\rangle + \sin\left(\frac{\tilde{\theta}}{2} + \frac{\pi}{4}\right)|1\rangle \tag{4.3}$$

Η παραπάνω παραμετροποίηση διατηρεί τις εξής ιδιότητες:

- $\theta_i = 0$: $P(|0\rangle)_i = P(|1\rangle)_i = \frac{1}{2}$ (μέγιστη υπέρθεση - κατάσταση Hadamard)
- $\theta_i = \frac{\pi}{2}$: $P(|0\rangle)_i = 0, P(|1\rangle)_i = 1$ (καθαρή κατάσταση $|1\rangle$)
- $\theta_i = -\frac{\pi}{2}$: $P(|0\rangle)_i = 1, P(|1\rangle)_i = 0$ (καθαρή κατάσταση $|0\rangle$)

Η γραφική αναπαράσταση της κβαντικής κατάστασης του qubit με βάση την γωνιακή παράμετρο θ, όπως ορίστηκε παραπάνω, παρουσιάζεται στο Σχήμα 4.1.

Με βάση την παραπάνω γωνιακή παραμετροποίηση, ένα άτομο q μπορεί να εκφραστεί ως:

$$\mathbf{q} = \begin{bmatrix} \theta_0 & \theta_1 & \cdots & \theta_i & \cdots & \theta_{m-1} \end{bmatrix} \tag{4.4}$$

όπου κάθε $\theta_i \in \mathbb{R}$ καθορίζει πλήρως την κβαντική κατάσταση του αντίστοιχου qubit.

Οι πιθανότητες μέτρησης των καταστάσεων βάσης, μέσω της σχέσης (2.9) και με χρήση της παραπάνω παραμετροποίησης υπολογίζονται ως εξής:

$$P(|0\rangle)(\theta) = \frac{1}{2} - \frac{1}{2}\sin(\theta), \quad P(|1\rangle)(\theta) = \frac{1}{2} + \frac{1}{2}\sin(\theta), \quad \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \tag{4.5}$$

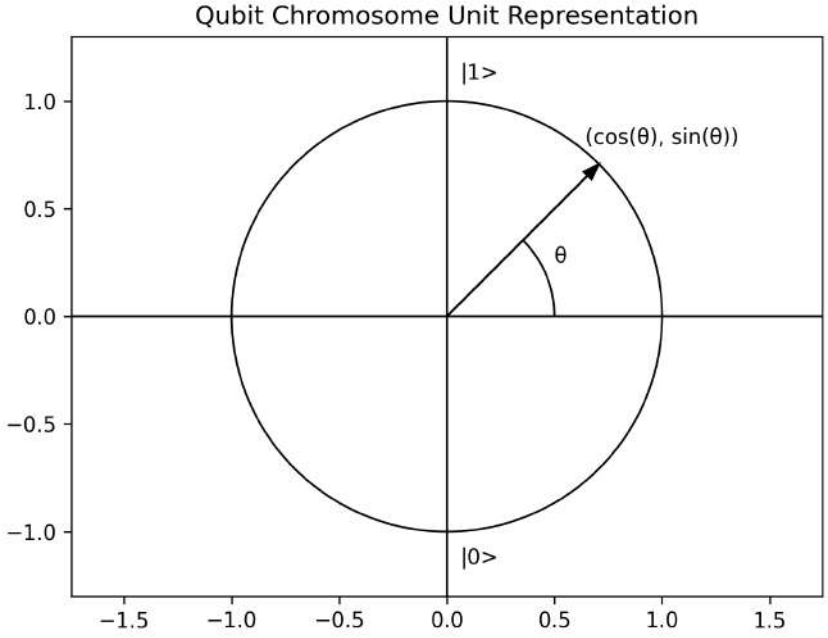


Figure 4.1: Γραφική Αναπαράσταση Κατάστασης Qubit Χρωμοσώματος

4.2 Υλοποίηση μονοκριτιριακού QIEA

Ο QIEA διατηρεί τρεις πληθυσμούς (γονέων, απογόνων και ελίτ) κατ' αναλογία με τον κλασικό εξελικτικό αλγόριθμο. Ακολουθεί η περιγραφή των κύριων συνιστωσών του αλγορίθμου.

4.2.1 Μέτρηση & Αξιολόγηση

Μέσω της **μέτρησης** εξάγεται μία κλασική δυαδική λύση από το κβαντικό χρωμόσωμα του ατόμου.

Measurement produces a classical binary candidate solution from a quantum individual. Για κάθε qubit i ενός ατόμου q_j^g της γενιάς g, η πιθανότητα μέτρησης της κατάστασης $|0\rangle$ υπολογίζεται βάση της σχέσης (4.5), και το αποτέλεσμα της μέτρησης καθορίζεται στοχαστικά μέσω της σύγκρισης της εν λόγω πιθανότητας με έναν αριθμό που λαμβάνουμε από μία γεννήτρια τυχαίων αριθμών.

$$b_{j,i}^g = \begin{cases} 0 & \text{if } r_{j,i} \leq p_{j,i}^g \\ 1 & \text{if } r_{j,i} > p_{j,i}^g \end{cases}, \quad r_{j,i} \sim \mathcal{U}(0,1) \tag{4.6}$$

Η ακολουθία bits που προκύπτει, αποκωδικοποιείται και αξιολογείται ως εξής:

$$\mathbf{q}_j^g \xrightarrow{\text{μέτρηση}} \mathbf{b}_j^g \xrightarrow{\text{αποκωδικοποίηση}} \mathbf{x}_j^g \xrightarrow{\text{αξιολόγηση}} F(\mathbf{x}_j^g)$$

Σε αντίθεση με ένα φυσικό κβαντικό σύστημα, η κβαντική κατάσταση διατηρείται μετά τη μέτρηση, επιτρέποντας mpi (measurements per individual) ανεξάρτητες μετρήσεις ανά άτομο ανά γενιά. Η καταλληλότερη από αυτές διατηρείται ως η πιο πρόσφατη λύση του ατόμου:

$$X_j^g = \{\mathbf{x}_j^{g,1}, \ldots, \mathbf{x}_j^{g,mpi}\}, \qquad \mathbf{x}_{\text{latest},j}^g = \arg\min_{\mathbf{x}\in X_j^g} F(\mathbf{x}) \tag{4.7}$$

Για κάθε άτομο, ο αλγόριθμος αποθηκεύει τις γωνίες qubit $\boldsymbol{\theta}_j^g$, την πιο πρόσφατη λύση $\mathbf{x}_{\text{latest},j}^g$ και την καλύτερη λύση που έχει βρεθεί έως τώρα $\mathbf{x}_{\text{best},j}^g$, μαζί με τις αντίστοιχες δυαδικές τους κωδικοποιήσεις $\mathbf{b}_{\text{latest},j}^g$ και $\mathbf{b}_{\text{best},j}^g$.

4.2.2 Αρχικοποίηση

Όλα τα n κβαντικά άτομα αρχικοποιούνται με $\theta_{j,i}^0 = 0$ για όλα τα qubits, δημιουργώντας για κάθε qubit μία κατάσταση Hadamard:

$$P(|0\rangle)_{j,i}^0 = P(|1\rangle)_{j,i}^0 = \tfrac{1}{2}, \qquad |\psi\rangle_{\text{initial}} = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Αυτή η αρχικοποίηση εξασφαλίζει ομοιόμορφη εξερεύνηση του χώρου αναζήτησης στην αρχή του αλγορίθμου. Στη συνέχεια, κάθε άτομο εκτελεί τις πρώτες του r μετρήσεις· το καταλληλότερο αποτέλεσμα αποθηκεύεται τόσο ως $\mathbf{x}_{\text{latest}}^0$ όσο και ως $\mathbf{x}_{\text{best}}^0$. Η συνολικά καλύτερη λύση $\mathbf{x}_{\text{global\_best}}^0$ προσδιορίζεται ως η καλύτερη μεταξύ όλων των ατόμων.

4.2.3 Πύλη Κβαντικής Περιστροφής (Q-Gate Rotation)

Μετά τη μέτρηση, το $\mathbf{x}_{\text{latest},j}^g$ συγκρίνεται με το $\mathbf{x}_{\text{best},j}^{g-1}$ ώστε να διαπιστωθεί αν έχει προκύψει βελτίωση. Για κάθε qubit i, η γωνίας περιστροφής $\Delta\theta_{j,i}$ που θα εφαρμοστεί μέσω του συγκεκριμένου τελεστή, αναζητείται σε έναν παραμετροποιήσιμο πίνακα (Πίνακας 4.1), ο οποίος αντιστοιχίζει την τριάδα $(b_{\text{latest},j,i}^g,\ b_{\text{best},j,i}^{g-1},\ F(\mathbf{x}_{\text{latest},j}^g) \le F(\mathbf{x}_{\text{best},j}^{g-1}))$ σε μια τιμή $\Delta\theta_{j,i}$.

$b_{\text{latest},j,i}^g$	$b_{\text{best},j,i}^{g-1}$	Improvement	$\Delta\theta_{j,i}$
0	0	True	a_1
0	1	True	a_2
1	0	True	a_3
1	1	True	a_4
0	0	False	a_5
0	1	False	a_6
1	0	False	a_7
1	1	False	a_8

Table 4.1: Configurable rotation strategy table for QIEA.

Κάθε στοιχείο a_k του Πίνακα 4.1 μπορεί να λάβει μία από τις ακόλουθες τιμές: μία σταθερή τιμή (τυπικά ± 1), έναν τυχαίο ακέραιο από το σύνολο $\{-1, 0, 1\}$ ή έναν ομοιόμορφα κατανεμημένο τυχαίο πραγματικό αριθμό στο διάστημα $[-1, 1]$. Ένα καθολικό βήμα γωνίας $\Delta\theta$ πολλαπλασιάζεται με κάθε στοιχείο του πίνακα, ώστε η πραγματική μεταβολή να είναι $\Delta\theta_{j,i} = a_c \cdot \Delta\theta$. Η γωνία του qubit ενημερώνεται στη συνέχεια προσθέτοντας τη γωνία $\Delta\theta_{j,i}$ και φράζοντας το αποτέλεσμα στο διάστημα $[-\theta_{\max}, +\theta_{\max}]$:

$$\theta_{j,i}^t \leftarrow \text{clip}\big(\theta_{j,i}^{t-1} + \Delta\theta_{j,i},\ -\theta_{\max},\ +\theta_{\max}\big) \tag{4.8}$$

4.2.4 Τελεστής Κβαντικού Crossover

Η διασταύρωση ακολουθεί το τυπικό σχήμα crossover ενός σημείου, εφαρμοζόμενο στις ακολουθίες γωνιών των qubit. Επιλέγονται δύο γονείς μέσω επιλογής τουρνουά και ανασυνδυάζονται με πιθανότητα p_x. Το σημείο διασταύρωσης ορίζεται σε επίπεδο μεταβλητής: επιλέγεται ένας τυχαίος δείκτης μεταβλητής $c \in \{1, \ldots, k-1\}$, που οδηγεί σε ένα σημείο διάσπασης σε επίπεδο χρωμοσώματος στη θέση $c' = \sum_{\ell=1}^{c} m_\ell$, όπου m_ℓ είναι το μήκος σε bits της μεταβλητής ℓ:

$$\boldsymbol{q}_{\text{off}}^{(1)} = [\theta_0^{(1)}, \ldots, \theta_{c'-1}^{(1)},\ \theta_{c'}^{(2)}, \ldots, \theta_{m-1}^{(2)}] \tag{4.9}$$

$$\boldsymbol{q}_{\text{off}}^{(2)} = [\theta_0^{(2)}, \ldots, \theta_{c'-1}^{(2)},\ \theta_{c'}^{(1)}, \ldots, \theta_{m-1}^{(1)}] \tag{4.10}$$

Κάθε απόγονος κληρονομεί τη μέχρι στιγμής καλύτερη λύση του από έναν από τους δύο γονείς, επιλεγμένο με ίση πιθανότητα, καθώς δεν πραγματοποιείται νέα μέτρηση αμέσως μετά τη διασταύρωση. Η διασταύρωση εφαρμόζεται κάθε T_x γενιές.

4.2.5 Μετάλλαξη

Χρησιμοποιούνται δύο συμπληρωματικοί τελεστές μετάλλαξης.

Μετάλλαξη Γωνίας Qubit Εκτελείται κατά την εφαρμογή του τελεστή Q-Gate. Με πιθανότητα p_{rot} η ενημέρωση της γωνίας αντιστρέφεται (πολλαπλασιάζεται με -1):

$$\theta_{j,i}^{t} \leftarrow \begin{cases} -(\theta_{j,i}^{t-1} + \Delta\theta_{j,i}) & \text{με πιθανότητα } p_m^{\text{rot}} \\ \theta_{j,i}^{t-1} + \Delta\theta_{j,i} & \text{με πιθανότητα } 1 - p_m^{\text{rot}} \end{cases} \tag{4.11}$$

Μετάλλαξη Μέτρησης Μετά τη διαδικασία της μέτρησης και πριν από την αξιολόγηση, κάθε bit αντιστρέφεται ανεξάρτητα με μία πιθανότητα p_m^{meas}.

4.2.6 Μετανάστευση

Χρησιμοποιούνται δύο στρατηγικές μετανάστευσης, για τη μεταφορά πληροφορίας στα διάφορα άτομα του πληθυσμού, αντικαθιστώντας τις μέχρι στιγμής καλύτερες λύσεις των ατόμων και, επηρεάζοντας έτσι το αποτέλεσμα της δράσης του τελεστή Q-Gate.

Τοπική Μετανάστευση Ο πληθυσμός χωρίζεται σε γειτονιές σταθερού μεγέθους. Κάθε T_{local} γενιές, η καλύτερη λύση της γειτονιάς αντικαθιστά τη $\mathbf{x}_{\text{best},j}^{g}$ κάθε μέλους αυτής.

Καθολική Μετανάστευση Κάθε T_{global} γενιές, η συνολικά καλύτερη λύση $\mathbf{x}_{\text{global\_best}}^{g}$ μεταδίδεται σε όλα τα άτομα.

4.2.7 Επιλογή και Ελιτισμός

Η επιλογή γονέων πραγματοποιείται από το συνδυασμένο σύνολο γονέων και απογόνων, χρησιμοποιώντας είτε επιλογή τουρνουά είτε γραμμικά σταθμισμένη επιλογή. Η ελίτ επιλογή (elite selection) διατηρεί τα e καταλληλότερα άτομα από τον συνδυασμένο πληθυσμό απογόνων και ελίτ. Τα άτομα της ελίτ εισάγονται στο σύνολο γονέων με πιθανότητα p_{elite} σε ποσοστό ανάλογο προς το μέγεθος του πληθυσμού γονέων.

4.2.8 Μηχανισμός Στασιμότητας

Η στασιμότητα εντοπίζεται όταν ο αριθμός των αξιολογήσεων χωρίς βελτίωση στη βέλτιστη τιμή καταλληλότητας υπερβεί ένα κατώφλι N_{stag}. Αυτό υποδηλώνει ότι ο αλγόριθμος έχει παγιδευτεί σε τοπικό ακρότατο, όπου σε αυτή την περίπτωση ενεργοποιείται μία σειρά από λειτουργίας που θα βοηθήσουν τον αλγόριθμο να ξεφύγει από αυτό. Για κάποιο αριθμό γενεών η επιλογή elite απενεργοποιείται προσωρινά, περίπου ο μισός γονεϊκός πληθυσμός αντικαθίσταται από άτομα που αρχικοποιούνται εκ νέου, και επιβάλλεται ο τελεστής crossover ανεξαρτήτως του δείκτης γεννιάς.

4.3 Αλγόριθμος QIEA

Παρουσιάζεται η πλήρης αλγοριθμική διαδικασία του μονοκριτηριακού QIEA:

1. **Αρχικοποίηση**:
   (a) Αρχικοποίηση του πληθυσμού απογόνων με γωνία κατάστασης όλων των qubit $\theta^0_{j,i} = 0$
   (b) Αρχική μέτρηση και αξιολόγηση κάθε ατόμου
   (c) Εκχώρηση της αρχικής μέτρησης και αξιολόγησης στη μεταβλητή της καλύτερης μέχρι στιγμής λύσης κάθε ατόμου
   (d) Αποθήκευση της καθολικής μέχρι στιγμής βέλτιστης λύσης $\mathbf{x}^0_{global\_best}$
2. **Κύριος βρόχος**:
   (a) Επιλογή γονέων
   (b) Εφαρμογή χβαντικού crossover (εάν $g \bmod T_{\text{x}} = 0$)
      i. Επιλογή ζευγών γονέων
      ii. Εφαρμογή διασταύρωσης μονού σημείου με πιθανότητα p_x
      iii. Ανάθεση τυχαία στο απόγονο την καλύτερη λύση από έναν γονέα
   (c) Εξέλιξη πληθυσμού απογόνων:
      i. Λήψη mpi μετρήσεων ανά άτομο (και εφαρμογή μετάλλαξης στην μέτρηση)
      ii. Επιλογή της καταλληλότερης ανάμεσα στις νέες μετρήσεις mpi ως η πιο πρόσφατη x_{latest}
      iii. Εφαρμογή τελεστή Q-Gate βάσει της σύγκρισης με την καλύτερη λύση x_{best}
      iv. Εφαρμογή μετάλλαξης της γωνίας qubit με πιθανότητα p_m^{rot}
      v. Ενημέρωση της καλύτερης μέχρι στιγμής λύσης του ατόμου, x_{best}, αν έχει προκύψει βελτίωση
   (d) Ανανέωση της καθολικής βέλτιστης μέχρι στιγμής λύσης $\mathbf{x}^g_{global\_best}$
   (e) Εκτέλεση μετανάστευσης (ανάλογα με τον δείκτη γενιάς):
      i. Καθολική μετανάστευση εάν $g \bmod T_{\text{global}} = 0$
      ii. Τοπική μετανάστευση εάν $g \bmod T_{\text{local}} = 0$
   (f) Επιλογή ελίτ από την ένωση των πληθυσμών απογόνων και ελίτ

4.4 Multi-Objective QIEA

Το πρόβλημα πολυκριτηριακής ελαχιστοποίησης και η κυριαρχία κατά Pareto διατυπώνονται στο Κεφάλαιο 3, στις Εξισώσεις (3.2) και (3.3). Η ενότητα περιγράφει τις αλλαγές που απαιτούνται για την προσαρμογή του αλγορίθμου για αντιμετώπιση προβλημάτων πολλαπλών στόχων.

4.4.1 Προσαρμογή Τελεστή Q-Gate

Στην περίπτωση ενός μοναδικού αντικειμενικού κριτηρίου, η απόφαση της πύλης Q ανάγεται σε μια σύγκριση βαθμωτής μορφής $F(\mathbf{x}_{\text{latest}}) \leq F(\mathbf{x}_{\text{best}})$. Υπό κυριαρχία κατά Pareto, προκύπτουν τρεις περιπτώσεις όταν συγκρίνεται η $\mathbf{x}^g_{\text{latest},j}$ με την $\mathbf{x}^g_{\text{best},j}$:

1. $\mathbf{x}^g_{\text{latest},j} \prec \mathbf{x}^g_{\text{best},j}$: βελτίωση — εφαρμόζονται οι καταχωρήσεις βελτίωσης του πίνακα περιστροφής, ενημερώνεται η $\mathbf{x}^g_{\text{best},j}$
2. $\mathbf{x}^g_{\text{best},j} \prec \mathbf{x}^g_{\text{latest},j}$: επιδείνωση — εφαρμόζονται οι καταχωρήσεις επιδείνωσης, η $\mathbf{x}^g_{\text{best},j}$ διατηρείται
3. $\mathbf{x}^g_{\text{latest},j} \not\sim \mathbf{x}^g_{\text{best},j}$: μη συγκρίσιμες — η κατεύθυνση ενημέρωσης είναι ασαφής

Η περίπτωση μη συγκρισιμότητας αντιμετωπίζεται με τη δημιουργία δύο ταυτόσημων αντιγράφων του ίδιου ατόμου, τα οποία εξελίσσονται υπό αντίθετες υποθέσεις:

$$\mathbf{q}^g_j \;\rightarrow\; \{\mathbf{q}^{g,1}_j,\, \mathbf{q}^{g,2}_j\}, \qquad \mathbf{q}^{g,1}_j = \mathbf{q}^{g,2}_j = \mathbf{q}^g_j$$

Η $\mathbf{q}^{g,1}_j$ εξελίσσεται υπό την υπόθεση βελτίωσης (καταχωρήσεις βελτίωσης του πίνακα περιστροφής, ενημέρωση της $\mathbf{x}_{\text{best}}$), ενώ η $\mathbf{q}^{g,2}_j$ εξελίσσεται υπό την υπόθεση επιδείνωσης (καταχωρήσεις επιδείνωσης, η $\mathbf{x}_{\text{best}}$ διατηρείται). Και τα δύο άτομα προωθούνται στο επόμενο βήμα επιλογής.

Η αύξηση του πληθυσμού λόγω διπλασιασμού ελέγχεται μέσω της επιλογής NSGA-II, όπως περιγράφεται παρακάτω.

4.4.2 Ταξινόμηση και Διαχείριση Πληθυσμού με NSGA-II

Μετά τη φάση εξέλιξης, εφαρμόζεται ο αλγόριθμος NSGA-II για να μειωθεί ο πληθυσμός ξανά στο μέγεθος n. Τα άτομα κατατάσσονται σύμφωνα με το επίπεδο μη κυριαρχούμενου μετώπου (πρωτεύον κριτήριο) και την απόσταση συνωστισμού (δευτερεύον κριτήριο):

$$d_i = \sum_{m=1}^{p} \frac{f_m^{i+1} - f_m^{i-1}}{f_m^{\max} - f_m^{\min}}$$

4.4.3 Δομικές Διαφορές από το Μονο-αντικειμενικό QIEA

Η πολυ-αντικειμενική παραλλαγή χρησιμοποιεί έναν ενιαίο πληθυσμό. Εφόσον η ταξινόμηση NSGA-II διασφαλίζει ότι οι μη κυριαρχούμενες λύσεις διατηρούνται πάντα, ένας ξεχωριστός ελίτ πληθυσμός είναι περιττός. Παρομοίως, η καθολική μετανάστευση, η τοπική μετανάστευση και ο μηχανισμός στασιμότητας παραλείπονται: δεν υπάρχει μοναδική παγκόσμια βέλτιστη λύση προς διάδοση, και υπό την κυριαρχία κατά Pareto η πιθανότητα παρατεταμένης στασιμότητας στο μη κυριαρχούμενο μέτωπο είναι αμελητέα.

4.4.4 Αλγοριθμική Διαδικασία

1. **Αρχικοποίηση**:
   (a) Αρχικοποίηση του πληθυσμού με $\theta_{j,i}^0 = 0$ για όλα τα qubits
   (b) Αρχική μέτρηση και αξιολόγηση, η αρχική μέτρηση εκχωρείται και στη μεταβλητή $\mathbf{x}_{\text{best},j}^0$
2. **Κύριος βρόχος**:
   (a) Εφαρμογή crossover αν $g \bmod T_{\text{crossover}} = 0$ και αρχική μέτρηση για κάθε νέο απόγονο
   (b) Εφαρμογή τελεστή Q-Gate για κάθε άτομο $\mathbf{q}_j^g$
   (c) Αφαίρεση από τον πληθυσμό ατόμων με πανομοιότυπα χρωμοσώματα
   (d) Κατάταξη και επιλογή με χρήση της NSGA-II ώστε να διατηρηθεί το μέγεθος του πληθυσμού n

Κεφάλαιο 5

QIEA Υποβοηθούμενος από Μεταμοντέλα

Τα μεταμοντέλα είναι απλοποιημένες προσεγγίσεις πιο σύνθετων, υπολογιστικά κοστοβόρων και ακριβών μοντέλων. Χρησιμοποιούνται για τη μείωση του υπολογιστικού κόστους κατά τη διαδικασία της βελτιστοποίησης, αντικαθιστώντας ακριβείς αλλά κοστοβόρες αξιολογήσεις με λιγότερο ακριβείς αλλά σημαντικά πιο οικονομικές προσεγγίσεις [8] [9].

Ο Metamodel-Assisted QIEA (MAQIEA), που αναπτύχθηκε ακολουθεί την ίδια διαδικασία με τον μη υποβοηθούμενο QIEA με την εξής τροποποίηση. Για κάθε άτομο του πληθυσμού κατά τη φάση της μέτρησης, πριν εφαρμοστεί ο τελεστής κβαντικής στροφής, λαμβάνεται ένας σημαντικά μεγαλύτερος αριθμός μετρήσεων. Αρχικά αυτές οι μετρήσεις αξιολογούνται με τη χρήση του μεταμοντέλου. Η καλύτερη εξ'αυτών για κάθε άτομο επαναξιολογείται και χρησιμοποιείται κατά την εφαρμογή του τελεστή στροφής. Η υπόλοιπη διαδικασία παραμένει ίδια, με βάση τα όσα περιγράψαμε παραπάνω. Η χρήση του μεταμοντέλου ξεκινάει όταν έχει συγκεντρωθεί επαρκής αριθμός ακριβών αξιολογήσεων στη βάση δεδομένων.

Για κάθε προσεγγιστική αξιολόγηση κατασκευάζεται ένα νέο δίκτυο ακτινικής βάσης το οποίο εκπαιδεύεται με πρότυπα που λαμβάνονται στη γειτονιά της εκάστοτε υποψήφιας λύσης. Η αρχιτεκτονική και η διαδικασία εκπαίδευσης του δικτύου περιγράφονται παρακάτω.

5.1 Δίκτυα Ακτινικής Βάσης (Radial-Basis Function Networks)

Τα RBF δίκτυα διακρίνονται για την ικανότητά τους να παρέχουν τοπικές προσεγγίσεις, καθώς και για την απλότητα της εκπαίδευσής τους. Ένα RBFN αποτελείται από τρία διακριτά στρώματα:

- **Στρώμα Εισόδου**: Δέχεται το διάνυσμα των μεταβλητών σχεδιασμού $x \in \mathbb{R}^N$ χωρίς να εκτελεί κάποια επεξεργασία
- **Κρυφό Στρώμα**: Περιλαμβάνει M νευρώνες, καθένας από τους οποίους υλοποιεί μια συνάρτηση ακτινικής βάσης με κέντρο c_m και ακτίνα r_m
- **Στρώμα Εξόδου**: Υπολογίζει την τελική έξοδο ως γραμμικό συνδυασμό των εξόδων του κρυφού στρώματος

Η έξοδος του δικτύου για ένα δεδομένο διάνυσμα εισόδου x δίνεται από τη σχέση:

$$y(x) = \sum_{m=1}^{M} \psi_m G_m(\|x - c_m\|) \tag{5.1}$$

όπου ψ_m είναι τα συναπτικά βάρη που συνδέουν το κρυφό με το στρώμα εξόδου, c_m τα κέντρα των RBF και G_m οι συναρτήσεις ακτινικής βάσης. Η συνήθως χρησιμοποιούμενη Γκαουσιανή συνάρτηση ορίζεται ως:

$$G(u, r) = \exp(-u^2/r^2) \tag{5.2}$$

Η εκπαίδευση ενός RBF δικτύου, δεδομένων των T προτύπων εκπαίδευσης $\{(\hat{x}_t, \hat{y}_t)\}_{t=1}^{T}$, συνίσταται στην εύρεση των βέλτιστων συναπτικών βαρών $\boldsymbol{\psi} = [\psi_1, \psi_2, \ldots, \psi_M]^T$ που ελαχιστοποιούν το σφάλμα:

$$E(\boldsymbol{\psi}) = \frac{1}{2} \sum_{t=1}^{T} \left(\hat{y}_t - \sum_{m=1}^{M} \psi_m G_m(\|\hat{x}_t - c_m\|) \right)^2 \tag{5.3}$$

Η παραπάνω εκφραση μπορεί να διατυπωθεί και ως:

$$E(\boldsymbol{\psi}) = \frac{1}{2} \|\hat{\mathbf{y}} - \mathbf{G}\boldsymbol{\psi}\|^2 \tag{5.4}$$

όπου $\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T]^T$ αντιπροσωπεύει το διάνυσμα των επιθυμητών εξόδων και $\mathbf{G}$ είναι ο πίνακας σχεδιασμού (design matrix) με στοιχεία:

$$\mathbf{G} = \begin{bmatrix} G_1(\|\hat{x}_1 - c_1\|) & G_2(\|\hat{x}_1 - c_2\|) & \cdots & G_M(\|\hat{x}_1 - c_M\|) \\ G_1(\|\hat{x}_2 - c_1\|) & G_2(\|\hat{x}_2 - c_2\|) & \cdots & G_M(\|\hat{x}_2 - c_M\|) \\ \vdots & \vdots & \ddots & \vdots \\ G_1(\|\hat{x}_T - c_1\|) & G_2(\|\hat{x}_T - c_2\|) & \cdots & G_M(\|\hat{x}_T - c_M\|) \end{bmatrix} \tag{5.5}$$

Η εύρεση του ελαχίστου της συνάρτησης κόστους επιτυγχάνεται μέσω της επίλυσης των κανονικών εξισώσεων. Παίρνοντας την παράγωγο της $E(\boldsymbol{\psi})$ ως προς $\boldsymbol{\psi}$ και θέτοντάς την ίση με μηδέν:

$$\frac{\partial E(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}} = -\mathbf{G}^T(\hat{\mathbf{y}} - \mathbf{G}\boldsymbol{\psi}) = 0 \tag{5.6}$$

προκύπτουν οι κανονικές εξισώσεις:

$$\mathbf{G}^T\mathbf{G}\boldsymbol{\psi}^* = \mathbf{G}^T\hat{\mathbf{y}} \tag{5.7}$$

με λύση:

$$\boldsymbol{\psi}^* = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\hat{\mathbf{y}} \tag{5.8}$$

υπό την προϋπόθεση ότι ο πίνακας $\mathbf{G}^T\mathbf{G}$ είναι αντιστρέψιμος.

Στην ειδική περίπτωση όπου τα κέντρα των RBF συμπίπτουν με τα εκπαιδευτικά πρότυπα ($c_m = \hat{x}_m$ για $m = 1, 2, \ldots, T$ και $M = T$), ο πίνακας $\mathbf{G}$ καθίσταται τετραγωνικός και συμμετρικός με στοιχεία:

$$G_{ij} = G(\|\hat{x}_i - \hat{x}_j\|) \tag{5.9}$$

Σε αυτήν την περίπτωση, η επιφάνεια απόκρισης του δικτύου διέρχεται ακριβώς από όλα τα εκπαιδευτικά πρότυπα (παρεμβολή), και τα συναπτικά βάρη υπολογίζονται άμεσα από:

$$\boldsymbol{\psi}^* = \mathbf{G}^{-1}\hat{\mathbf{y}} \tag{5.10}$$

Κεφάλαιο 6

Αξιολόγηση Απόδοσης και Σχολιασμός Αποτελεσμάτων

Προκειμένου να αξιολογηθεί η απόδοση του αλγορίθμου, χρησιμοποιείται μία σειρά από δοκιμαστικές μαθηματικές συναρτήσεις. Για μονοκριτηριακή βελτιστοποίηση χρησιμοποιούνται οι συναρτήσεις Sphere, Ackley, Rastrigin, Griewank, Schwefel και Rosenbrock, ενώ για πολυκριτηριακή βελτιστοποίηση χρησιμοποιούνται οι ZDT1, ZDT2 και ZDT3. Οι δοκιμές εφαρμόστηκαν για 10 μεταβλητές σχεδιασμού με 10 *bits* ανά μεταβλητή.

Αρχικά, πραγματοποιείται συστηματική αναζήτηση στο χώρο παραμέτρων για τον QIEA, ώστε να εντοπιστούν οι παραμετροποιήσεις που οδηγούν στην ταχύτερη σύγκλιση για κάθε δοκιμαστική συνάρτηση. Εν συνεχεία, διεξάγεται παραμετρική ανάλυση γύρω από τις βέλτιστες διαμορφώσεις, εξετάζοντας μεμονωμένα κάθε παράμετρο με τις υπόλοιπες να διατηρούνται σταθερές.

Οι περιπτώσεις των συναρτήσεων Sphere, Ackley και Rosenbrock αξιολογούνται επίσης με χρήση υποβοήθησης μεταμοντέλων (*MAQIEA*). Για τη συγκριτική αξιολόγηση, ο QIEA αντιπαρατίθεται με τον κλασικό εξελικτικό αλγόριθμο (EA), ο οποίος εκτελείται μέσω της πλατφόρμας EASY. Σημειώνεται ότι τα αποτελέσματα που παρουσιάζονται δεν αποσκοπούν στην εξαγωγή συμπεράσματος για τον αν ο QIEA υπερτερεί του κλασσικού EA. Στόχος ήταν η αξιολόγηση της απόδοσης του QIEA σε ένα σύνολο προβλημάτων βελτιστοποίησης μέσω της αντιπαραβολής με αποτελέσματα που λήφθηκαν με χρήση του λογισμικού EASY. Αν σε ορισμένα διαγράμματα φαίνεται ότι ο QIEA συγκλίνει ταχύτερα από τον EA, δεν θα πρέπει να ερμηνευθεί ως απόδειξη υποροχής. Κατά την ανάπτυξη και αξιολόγηση του QIEA δοκιμάστηκε πληθώρα συνδιασμών μεθόδων και παραμέτρων. Δεν ακολουθήθηκε η ίδια εξονυχιστική διαδικασία ανεύρεσης της βέλτιστης παραμετροποίησης για κάθε πρόβλημα βελτιστοποίησης κατά την εξαγωγή αποτελεσμάτων με τον EA. Αντ' αυτού, τα αποτελέσματα του EA χρησιμοποιήθηκαν ως ένα σημείο αναφοράς για την αξιολόγηση της συμπεριφοράς και της ταχύτητας σύγκλισης του QIEA. Κατά συνέπεια, για τον EA παράγαμε ένα σύνολο αποτελεσμάτων χωρίς να εστιάζουμε στην επίτευξη της ταχύτερης δυνατής σύγκλισης μέσω εκτεταμένης βελτιστοποίησης παραμέτρων.

Παρουσιάζονται τα συγκριτικά αποτελέσματα και ενδεικτικά μία παραμετρική ανάλυση του πίνακα στροφής της Q-Gate. Τα πλήρη αποτελέσματα καθώς και οι συνθήκες και παραμετροποιήσεις περιγράφονται με λεπτομέρεια στο πλήρες κείμενο.

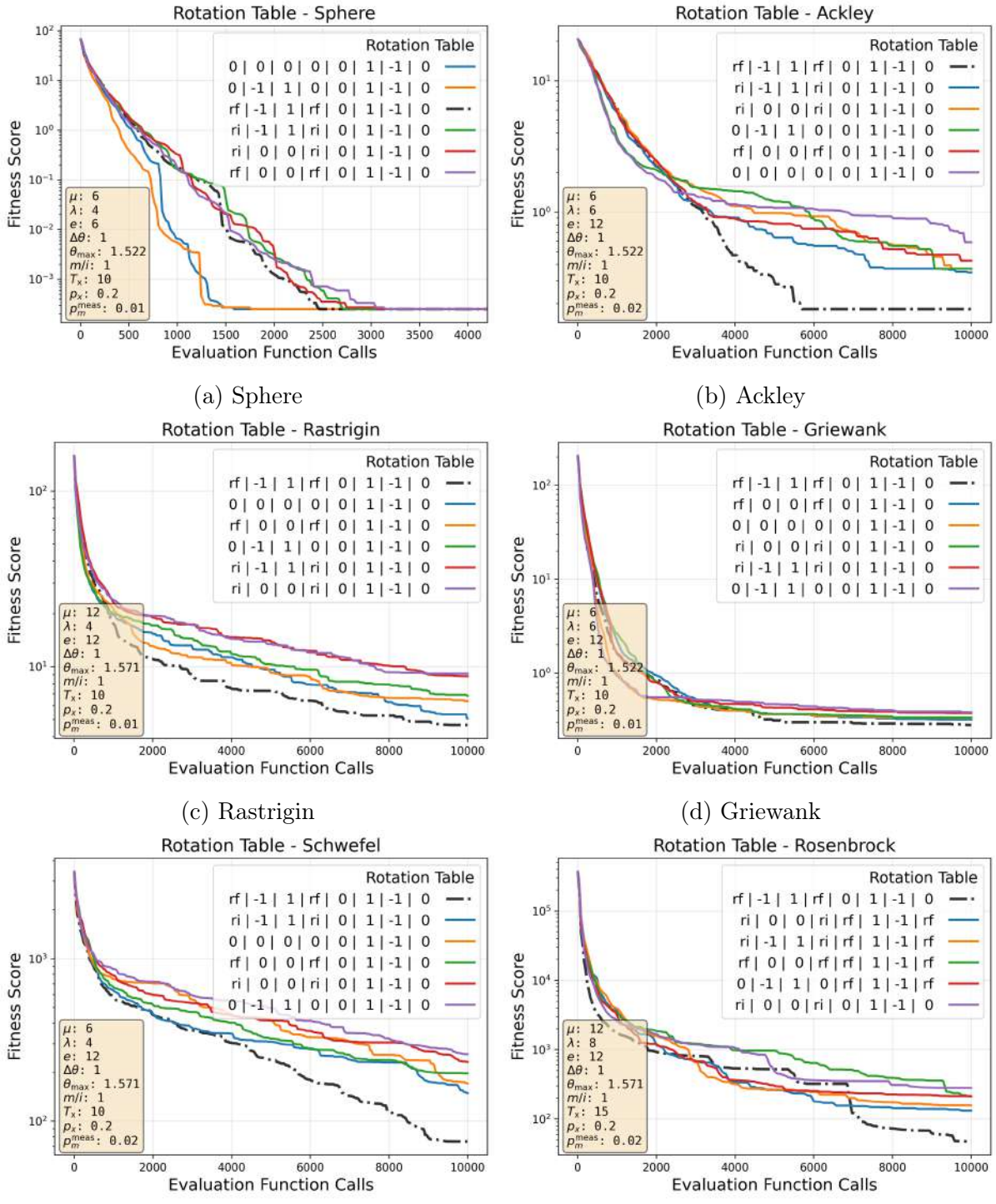
Παρατηρήσεις Κάποια αποτελέσματα του QIEA διαφαίνονται υποσχόμενα. Σε περιπτώσεις όπως η συνάρτηση σφαίρας ή η Ackley, παρατηρούμε ότι για ορισμένες παραμετροποιήσεις η ταχύτητα σύγκλισης είναι ικανοποιητική. Για περιπτώσεις όπως η Rastrigin και η Griewank, οι οποίες εμφανίζουν πολλά τοπικά ακρότατα, παρατηρούμε καλή ταχύτητα σύγκ-

λισης στην αρχή, η οποία στη συνέχεια επιδεινώνεται μετά από έναν ορισμένο αριθμό αξιολογήσεων, κάτι που υποδηλώνει ότι ο αλγόριθμος είναι επιρρεπής στο να εγκλωβίζεται σε τοπικά βέλτιστα. Σε παραπλανητικά τοπία, όπως αυτό της Schwefel, ο αλγόριθμος φαίνεται να δυσκολεύεται ακόμη περισσότερο, γεγονός που επιβεβαιώνει περαιτέρω την τάση του να εγκλωβίζεται σε τοπικά βέλτιστα. Για τη Rosenbrock παρατηρούμε και πάλι καλή σύγκλιση στην αρχή, η οποία όμως επιδεινώνεται μετά από κάποιες αξιολογήσεις, υποδεικνύοντας ότι η συσχέτιση μεταξύ των μεταβλητών σχεδιασμού σε αυτό το συγκεκριμένο πρόβλημα δυσκόλεψε επίσης τον αλγόριθμο.

Η επίδραση των μεταμοντέλων φαίνεται να είναι αρκετά ισχυρή, καθώς μειώνει τον αριθμό των αξιολογήσεων περίπου στο ένα τρίτο, σε σύγκριση με τον QIEA χωρίς βοήθεια μεταμοντέλων. Τα αποτελέσματα για τη βελτιστοποίηση πολλών στόχων υποδεικνύουν ότι η σύγκλισή του QIEA, στην παρούσα μορφή, είναι βραδύτερη από αυτή του EA.

Μια άλλη ενδιαφέρουσα παρατήρηση αφορά τον πίνακα στροφής. Δοκιμάστηκαν δεκαοκτώ διαφορετικές παραμετροποιήσεις. Για όλες τις περιπτώσεις, η παραμετροποίηση RT1[1] φαίνεται να δίνει την καλύτερη σύγκλιση, εκτός από την περίπτωση της σφαίρας. Για τη συνάρτηση σφαίρας, η απλούστερη παραμετροποίηση RT2[2] αποδίδει καλύτερα. Κάθε παραμετροποίηση παρουσιάζει διαφορετικά χαρακτηριστικά. Η πρώτη περίπτωση καθιστά τον αλγόριθμο πιο εξερευνητικό, κάτι που βοηθά στο να αποφευχθεί ο εγκλωβισμός του αλγορίθμου σε τοπικά βέλτιστα, ή στην αναζήτηση διαφορετικών λύσεων ακόμη και όταν έχει ήδη βρει μια σχετικά καλή. Η δεύτερη και απλούστερη παραμετροποίηση προσδίδει έναν λιγότερο εξερευνητικό και περισσότερο εκμεταλλευτικό χαρακτήρα, κάτι που είναι επωφελές για ένα ομαλό τοπίο όπως αυτό της σφαίρας.

[1]RT1: [`randfloat`, -1, 1, `randfloat`, 0, 1, -1, 0]
[2]RT2: [0, 0, 0, 0, 0, 1, -1, 0]



(a) Sphere

(b) Ackley

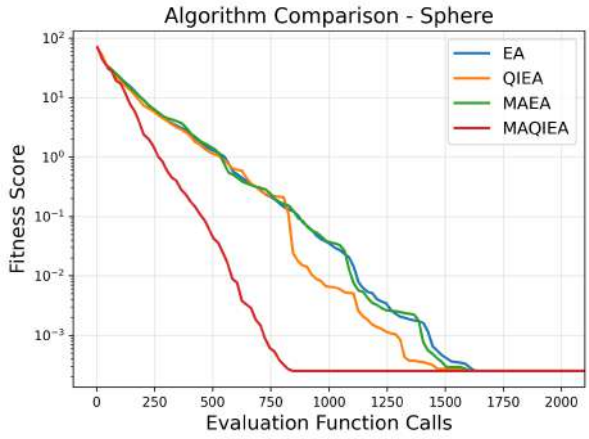
(c) Rastrigin

(d) Griewank

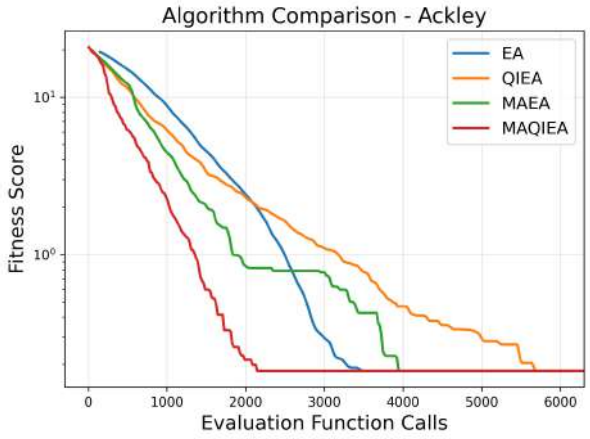
(e) Schwefel

(f) Rosenbrock

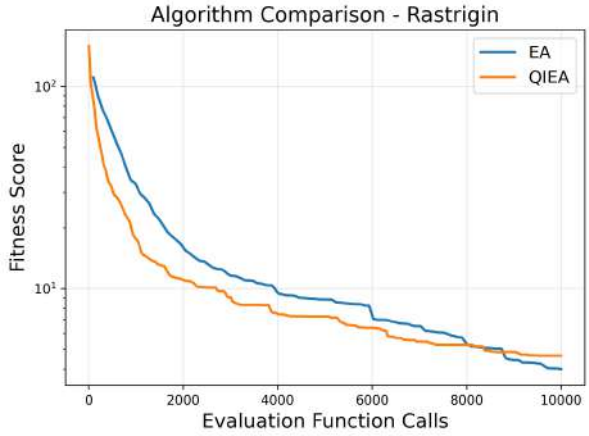
Figure 6.1: Επιρροή της παραμέτρου του πίνακα στροφής Q-Gate στην ταχύτητα σύγκλισης



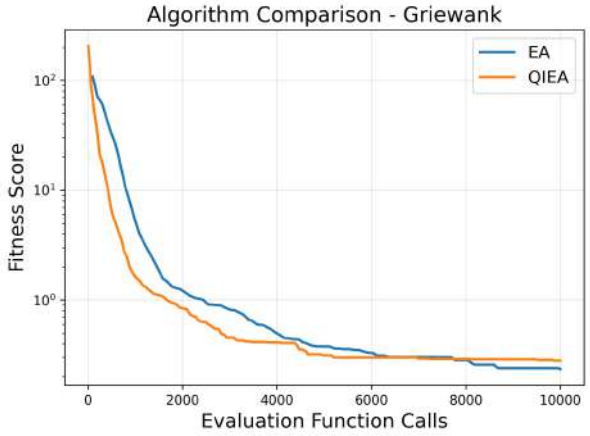
(a) Sphere



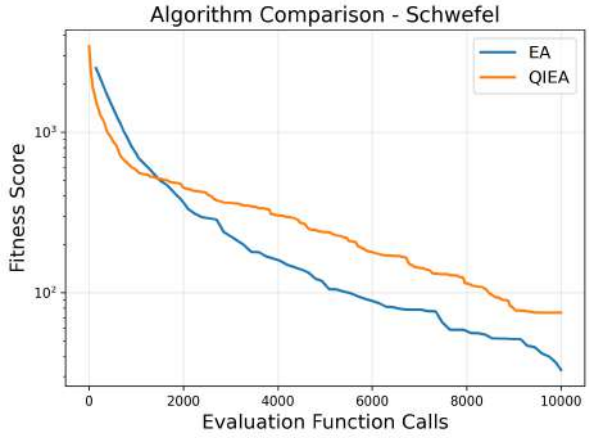
(b) Ackley



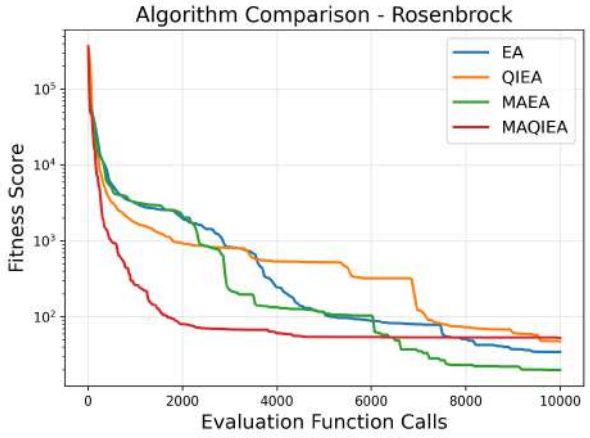
(c) Rastrigin



(d) Griewank

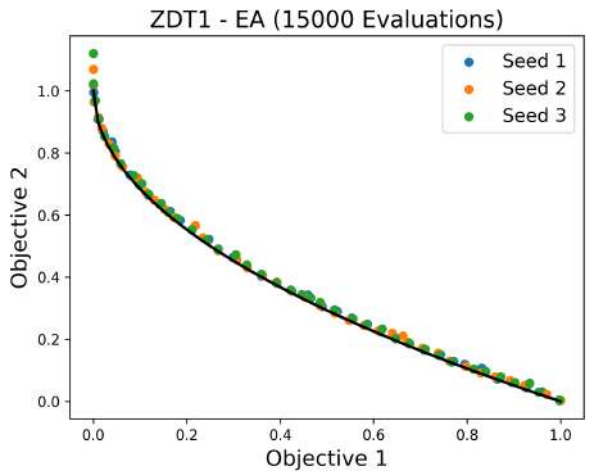


(e) Schwefel

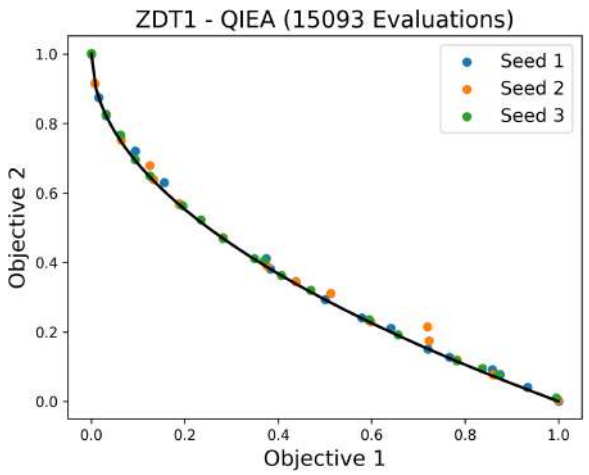


(f) Rosenbrock

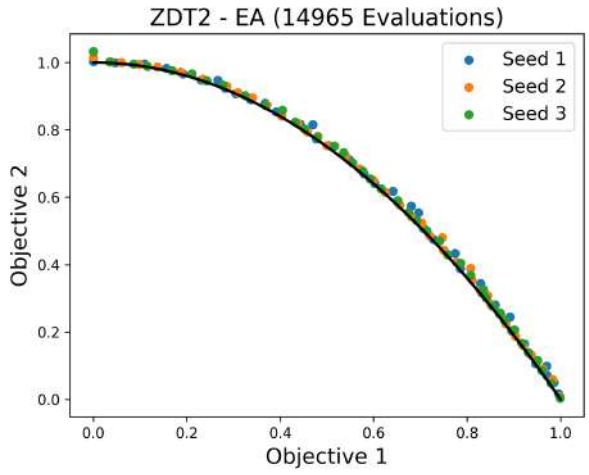
Figure 6.2: Συγκριτικά αποτελέσματα QIEA - EA



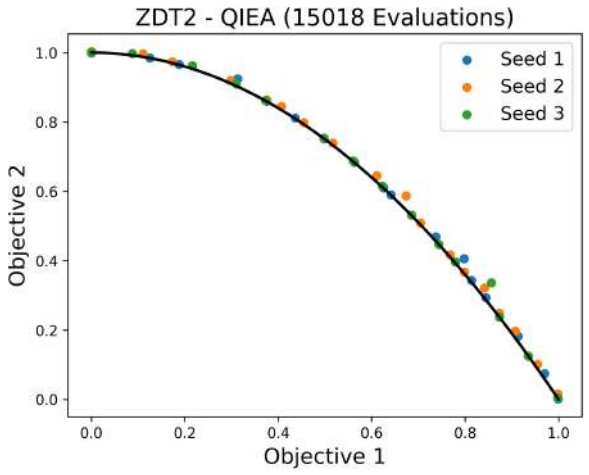
(a) ZDT1 - EA



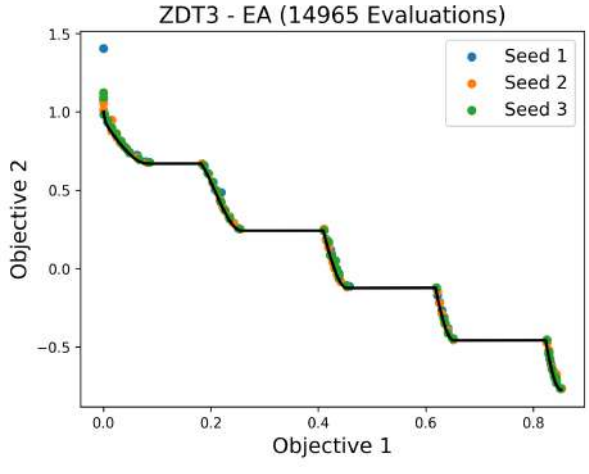
(b) ZDT1 - QIEA



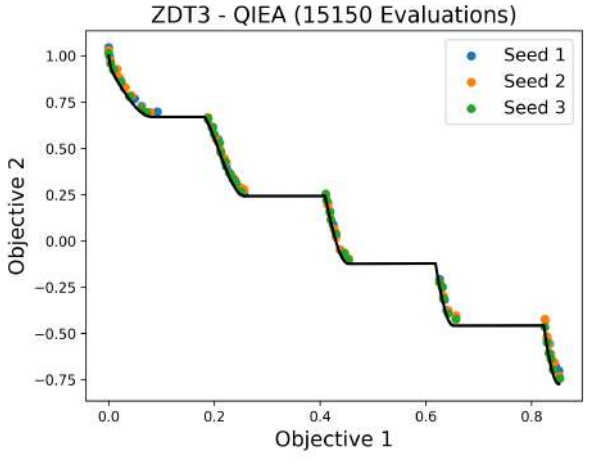
(c) ZDT2 - EA



(d) ZDT2 - QIEA



(e) ZDT3 - EA



(f) ZDT3 - QIEA

Figure 6.3: Προσέγγιση μετώπου Pareto για τα ZDT1, ZDT2 και ZDT3, με σύγκριση μεταξύ EA και QIEA

Κεφάλαιο 7

Συμπεράσματα

Παρατηρούμε ότι ο QIEA διαθέτει μεγάλο αριθμό παραμέτρων και ότι η σύγκλιση του αλγορίθμου φαίνεται να είναι ευαίσθητη σε αυτές. Παρατηρούμε επίσης ισχυρή συσχέτιση τόσο μεταξύ των ίδιων των παραμέτρων όσο και μεταξύ των παραμέτρων και του προβλήματος βελτιστοποίησης. Ο μικρός αριθμός πληθυσμών για τους οποίους λειτουργεί καλά τον καθιστά ευάλωτο σε πρόωρη σύγκλιση και παγίδευση σε τοπικά βέλτιστα. Παρ' όλα αυτά, σε ορισμένες από τις δοκιμαστικές περιπτώσεις, όπως οι συναρτήσεις σφαίρας και Ackley, η ταχύτητα σύγκλισής του ήταν ικανοποιητική και συγκρίσιμη με εκείνη του EA. Συνοψίζοντας, παρουσιάζει ορισμένα ενθαρρυντικά αποτελέσματα. Ωστόσο, απαιτούνται βελτιώσεις ώστε να γίνει πιο εύρωστος και κατάλληλος για χρήση σε εφαρμογές πραγματικού κόσμου.

Προς αυτή την κατεύθυνση, προτείνονται οι ακόλουθες προσεγγίσεις, που θα μπορούσαν να αποτελέσουν αντικείμενο μελλοντικής έρευνας για τον QIEA. Όπως εξηγήσαμε, οι διαφορετικοί πίνακες περιστροφής αποτελούν καλή παραμετροποίηση του βαθμού εξερεύνησης ή εκμετάλλευσης που επιθυμούμε από τον αλγόριθμο. Διαφορετικοί πίνακες περιστροφής θα μπορούσαν να χρησιμοποιούνται για τα διάφορα άτομα του πληθυσμού κατά τη διάρκεια της ίδιας βελτιστοποίησης, έτσι ώστε κάθε άτομο να αναζητά τη λύση με διαφορετική στρατηγική. Με αυτόν τον τρόπο, αφενός θα μπορούσαμε να μετριάσουμε τον ελιτίστικο χαρακτήρα του QIEA που παρουσιάστηκε, ο οποίος είναι επιρρεπής στον εγκλωβισμό σε τοπικά βέλτιστα, και αφετέρου θα μπορούσαμε να κάνουμε τον QIEA πιο ευπροσάρμοστο σε διαφορετικά προβλήματα βελτιστοποίησης.

Μια άλλη πιθανή προσέγγιση είναι η εφαρμογή του QIEA σε μικρούς υποπληθυσμούς, υποβοηθώντας τον EA και όχι ως υποκατάστατό αυτού. Τα αποτελέσματα ενδεικνύουν ότι ο QIEA λειτουργεί ιδιαίτερα καλά με μικρούς πληθυσμούς. Η ισχυρή εκμεταλλευτική του συμπεριφορά και η ταχεία σύγκλισή του στην αρχή της εξελικτικής διαδικασίας, θα μπορούσαν ενδεχομένως να αξιοποιηθούν αποτελεσματικά εάν ο QIEA χρησιμοποιούνταν σε υποστηρικτικό ρόλο παράλληλα με τον κλασικό EA.

Οι κλασικοί εξελικτικοί αλγόριθμοι έχουν πολλά περισσότερα χρόνια έρευνας και εφαρμογής πίσω τους, επομένως το αναμενόμενο ήταν ότι ο QIEA δεν θα εμφάνιζε κάποια ισχυρή υπεροχή έναντι των κλασσικών EA. Παρ' όλα αυτά, τα αποτελέσματα που εξάγαμε είναι ενθαρρυντικά και ενδέχεται να συμβάλουν στην πρόοδο του πεδίου της βελτιστοποίησης.

Βιβλιογραφία

[1] Sukhpal Singh Gill, Oktay Çetinkaya, Stefano Marrone, Daniel Claudino, David Haunschild, Leon Schlote, Huaming Wu, Carlo Ottaviani, Xiaoyuan Liu, Sree Machupalli, Kamalpreet Kaur, Priyansh Arora, Ji Liu, Ahmed Farouk, Houbing Song, Steve Uhlig, and Kotagiri Ramamohanarao. *Quantum computing: vision and challenges*, pages 19–42. 01 2025.

[2] Kuk-Hyun Han and Jong-Hwan Kim. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *Evolutionary Computation, IEEE Transactions on*, 6:580 – 593, 01 2003.

[3] Zhou Liang, Shao Ming, and Ma Chengqian. Adaptive collaborative quantum-inspired evolutionary algorithm for global numerical functions. *ITM Web of Conferences*, 16:02010, 01 2018.

[4] P. Dirac. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35:416 – 418, 07 1939.

[5] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition, 2010.

[6] N. David Mermin. *Quantum Computer Science: An Introduction*. Cambridge University Press, 2007.

[7] K. Giannakoglou, D. Tsahalis, Jacques Periaux, K. Papailiou, T. (eds, and Eckart Zitzler. Evolutionary algorithms for multiobjective optimization. 11 2001.

[8] Marios Karakasis and Kyriakos Giannakoglou. On the use of surrogate evaluation models in multi-objective evolutionary algorithms. *European Congress on Computational Methods in Applied Sciences and Engineering ECCOMAS*, pages 24–28, 08 2004.

[9] Javier Gonzalez-Conde, Dylan Lewis, Sachin Bharadwaj, and Mikel Sanz. Quantum carleman linearization efficiency in nonlinear fluid dynamics. *Physical Review Research*, 7, 06 2025.