



Article

Thermonuclear Fusion Based Quantum-Inspired Algorithm for Solving Multiobjective Optimization Problems

Liliya Demidova and Vladimir Maslennikov

Special Issue

Numerical Optimization and Algorithms: 4th Edition

Edited by

Prof. Dr. Dunhui Xiao and Prof. Dr. Shuai (Steven) Li



Article

Thermonuclear Fusion Based Quantum-Inspired Algorithm for Solving Multiobjective Optimization Problems

Liliya Demidova *  and Vladimir Maslennikov * 

Institute of Information Technologies, Federal State Budget Educational Institution of Higher Education, MIREA—Russian Technological University, 78, Vernadsky Avenue, 119454 Moscow, Russia

* Correspondence: liliya.demidova@rambler.ru (L.D.); vldmsn@yahoo.com (V.M.)

Abstract

This paper introduces a novel quantum-inspired algorithm for numerical multiobjective optimization, uniquely integrating the multilevel structure of qudits with principles of controlled thermonuclear fusion. Moving beyond conventional qubit-based approaches, the algorithm leverages the qudit's higher-dimensional state space to enhance search capabilities. Fusion-inspired dynamics—modeling particle interaction, energy release, and plasma cooling—provide a powerful metaheuristic framework for navigating complex, high-dimensional Pareto fronts. A hybrid quantum-classical version of the algorithm is presented, designed to exploit the complementary strengths of both computational paradigms for improved efficiency in solving dynamic multiobjective problems. Experimental evaluation on standard dynamic multiobjective benchmarks demonstrates clear performance advantages. Both the quantum-inspired and hybrid variants consistently outperform leading classical algorithms such as NSGA-III, MOEA/D and GDE3, as well as the quantum-inspired NSGA-III, in key metrics: identifying a greater number of unique non-dominated solutions, ensuring superior uniformity along the Pareto front, maintaining stable convergence across generations, and achieving higher accuracy in approximating the ideal solution.

Keywords: multiobjective optimization; quantum-inspired algorithm; qudit; thermonuclear fusion; quantum-classical system; quantum logic circuit; dynamic optimization; Pareto front



Academic Editor: Frank Werner

Received: 28 August 2025

Revised: 1 December 2025

Accepted: 12 December 2025

Published: 15 December 2025

Citation: Demidova, L.; Maslennikov, V. Thermonuclear Fusion Based Quantum-Inspired Algorithm for Solving Multiobjective Optimization Problems. *Algorithms* **2025**, *18*, 793. <https://doi.org/10.3390/a18120793>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The third decade of the XXI century is characterized by rapid development and global proliferation of information technologies, driving the continuous creation of tools for automation and optimization of various processes across numerous domains of human activity. Depending on the domain and specific application, processes may exhibit complex structures, significantly complicating the implementation of tools for their automation or optimization with the required efficiency. For instance, in many intelligent predictive systems and decision support systems based on machine learning algorithms, optimization algorithms are employed to solve parameter search problems—aimed at finding values of certain parameters that achieve an extremum of the objective function. For complex models, such as deep neural networks, the task of hyperparameter tuning can be non-trivial due to the high computational cost of the model, the high dimensionality of the search space, nonlinearity and multimodality of the objective function, stochasticity of the training process, constraints on the feasible domain, and other associated factors [1]. Considering

the combination of these factors, achieving an extremum of the objective function within reasonable time and with the required accuracy poses a significant challenge for many classical optimization algorithms—such as evolutionary algorithms—particularly in the case of multiobjective optimization, where extrema of multiple objective functions are sought simultaneously. However, an alternative approach based on the simulation of quantum computations offers the potential to substantially accelerate the discovery of high-quality solutions. This approach is known as «quantum-inspired».

Quantum-inspired computing enables the solution of standard problems on classical computing devices and systems by leveraging concepts and methods from quantum mechanics and quantum computation. According to the study [2], which investigates the practical potential of quantum-inspired algorithms for recommendation systems and solving linear systems of equations, such algorithms exhibit a polylogarithmic dependence on data dimensionality up to a certain threshold. This provides an asymptotically exponential speedup compared to classical methods for problems involving low-rank matrices. However, their complexity is upper-bounded by a polynomial dependence, which remains significantly higher than that of true quantum computations.

The application of quantum-inspired computing to numerical and combinatorial single-objective optimization [3,4] has yielded algorithms that outperform classical counterparts in execution time, convergence speed, and solution accuracy. However, since quantum-inspired methods run on classical hardware, they cannot fully exploit genuine quantum advantages, such as superposition and entanglement, that enable exponential or polynomial speedups in true quantum algorithms like Grover's or Shor's. These speedups typically require structured problems or compact data representations allowing efficient use of quantum parallelism via components like quantum oracles. While quantum-inspired computing simulates selected quantum concepts on classical systems, thereby improving computational logic and performance, it remains fundamentally limited compared to actual quantum devices [5].

In existing quantum-inspired single-objective optimization algorithms [3,4,6–8], the qubit, characterized by two basis states, $|0\rangle$ and $|1\rangle$, serves as the core element. Solutions are encoded in superposition states, and their probability amplitudes are iteratively updated via unitary transformations, with the highest-amplitude state ultimately selected as the solution. While this approach yields strong performance in terms of accuracy, convergence speed, and execution time for low-dimensional problems (typically $n \leq 5$), its effectiveness diminishes as dimensionality grows, often matching or even falling behind classical methods.

Another approach employs the qudit [9]—a multilevel quantum system—as the core logical unit. In ref. [10], a qudit-based adaptation of the quantum-inspired genetic algorithm [11] was tested on benchmark functions with dimensions $n = 15$ and $n = 50$, as well as on real-world problems in manufacturing scheduling and machine learning. Results showed consistent superiority over both classical methods and the original qubit-based version in terms of accuracy, convergence speed, execution time, and solution distribution density.

Despite promising results in single-objective settings, quantum-inspired optimization algorithms struggle in high-dimensional multiobjective problems. As shown in refs. [12,13], methods like quantum-inspired NSGA-II/III or particle swarm optimization suffer rapid performance degradation with increasing decision variables due to the «curse of dimensionality»: the exponentially growing search space overwhelms qubit-based encodings, causing premature convergence and poor exploration-exploitation balance. Although QIEAs (Quantum-Inspired Evolutionary Algorithms) initially boost diversity via superposition, they often fail to maintain it or navigate complex, non-convex Pareto fronts. In ref. [13] notes that both classical and quantum-inspired approaches lack efficiency in such scenarios, prompting interest in spe-

cialized frameworks like VQAs (Variational Quantum Algorithms), while ref. [12] highlights that QIEAs' theoretical advantages are offset by high computational overhead and sensitivity to parameter tuning, limiting their real-world scalability.

Moreover, current hybrid quantum-classical architectures face significant practical limitations. As noted in ref. [14], integrating emerging quantum processors with existing classical IT infrastructure is complex, often requiring custom middleware and non-standard APIs that hinder interoperability. This fragmentation introduces latency—especially in iterative algorithms that repeatedly transfer data between classical and quantum subsystems—eroding potential quantum advantages. Additionally, these systems demand rare dual expertise in quantum information and classical distributed computing. Most frameworks are rigid, monolithic, or tightly bound to specific quantum backends, lacking the modularity and adaptability needed for evolving hardware or real-world dynamic optimization tasks.

These limitations reveal a critical research gap: existing quantum-inspired multiobjective algorithms lack a principled mechanism to sustain diversity, adaptively balance exploration and exploitation, and scale efficiently in high-dimensional dynamic environments, and they rarely leverage physically grounded, self-regulating dynamics to encode such adaptivity. To address this gap, the present study introduces a novel quantum-inspired algorithm for numerical multiobjective optimization that uniquely integrates multilevel qudit-based agents with physics-inspired dynamics from controlled thermonuclear fusion. By leveraging the qudit's higher-dimensional state space and explicitly modeling particle interaction, energy release, and plasma cooling, the algorithm establishes a self-regulating mechanism that effectively navigates complex, high-dimensional Pareto landscapes. A hybrid quantum-classical variant is also presented, along with explicit quantum circuit designs for key operations—qudit initialization, phase encoding of objective functions, and Grover-based amplitude amplification. Rigorous testing on dynamic multiobjective benchmarks demonstrates consistent, statistically significant performance advantages over state-of-the-art methods.

2. Related Studies

Quantum-inspired multiobjective optimization is a topical research area due to its potential to solve global optimization problems characterized by nonlinear constraints, exponential complexity of the search space, and correlated objectives with ill-defined priorities—features that are particularly relevant in modern applications such as machine learning, logistics, energy systems, and bioinformatics. Unlike classical optimization methods, quantum-inspired approaches offer fundamentally new ways of information processing based on quantum phenomena such as superposition and entanglement, enabling more efficient exploration of the solution space. However, adapting these quantum phenomena to classical computing devices and systems is a complex and multifaceted challenge that requires careful consideration of numerous specific aspects of quantum mechanics. This has led to a wide range of research efforts exploring diverse methods for simulating quantum particles and novel methodologies for solving optimization problems.

In the context of quantum-inspired modeling for multiobjective optimization algorithms, ref. [15] proposes the DMQSSA (Decomposition-Based Quantum-Inspired Salp Swarm Algorithm), which combines quantum-inspired principles with the salp swarm algorithm [16]. DMQSSA employs a delta-potential well model [17] to enhance convergence, a decomposition-based approach for the crossover operator, and an intelligent strategy for maintaining an archive of non-dominated solutions, thereby ensuring population diversity and generating a well-distributed set of Pareto-optimal solutions. The algorithm QCCEA (Quantum-Inspired Competitive Coevolution Algorithm) [18], a quantum-inspired coevo-

lutionary approach, employs a quantization technique based on Gaussian distribution, quantum rotation gates applied to qubits on the Bloch sphere for solution diversification, and a «Hall of Fame» strategy [19] for selecting the best solutions. The study presented in ref. [20] describes a quantum-inspired optimization algorithm, MOQSOA (Multiobjective Quantum-Inspired Seagull Optimization Algorithm), which combines quantum principles with the natural behavior of seagulls. The algorithm employs quantum state encoding as a linear superposition of positive and deceptive states to represent solutions, an opposition-based learning approach [21] to enhance the initial population, a grid-based density ranking method [22] for maintaining an archive of non-dominated solutions, a correction mechanism to prevent premature convergence, and incorporates behavioral patterns of seagulls—such as migration and attacking—for searching optimal solutions.

Further enriching the landscape of quantum-inspired optimization, recent studies have introduced novel algorithms that leverage quantum principles in distinct and innovative ways. One study [23] presents a pioneering application of the QAOA (Quantum Approximate Optimization Algorithm) to solve the IDP (Independent Domination Problem), a combinatorial optimization challenge with practical implications in network design. This work is significant as it marks the first application of QAOA to the IDP, demonstrating efficacy in finding optimal solutions with computational complexity surpassing that of classical methods. The transformation of the IDP into the QUBO (Quadratic Unconstrained Binary Optimization) model, followed by its encoding into a Hamiltonian, enables solution via the hybrid quantum-classical QAOA framework. Robustness testing reveals strong dependence of the algorithm's performance on parameter tuning, such as the number of layers and penalty coefficients, offering valuable insights for future QAOA applications in similar discrete optimization problems.

Another recent study [24] proposes the QPPA (Quantum Predator-Prey Algorithm), a metaheuristic designed for real-parameter optimization. QPPA uniquely fuses fundamentals of quantum mechanics—specifically, the delta-potential well model [17]—with dynamics of a predator-prey ecological system. Unlike many quantum-inspired algorithms focusing on qubit-based representations, QPPA employs the quantum model to mathematically derive movement equations for «predator» agents as they pursue «prey» agents in the search space. This quantum formulation governs the exploration phase, enabling effective escape from local optima. QPPA demonstrates superior performance and rapid convergence on benchmark functions, outperforming established algorithms such as PSO, GA and GWO—particularly in high-dimensional settings. This underscores the versatility of quantum-inspired principles, illustrating successful adaptation not only to discrete problems via QAOA but also to continuous optimization through novel metaheuristic frameworks like QPPA.

The qudit has found application not only in quantum-inspired genetic algorithms for single-objective optimization [10], but also in solving the classical combinatorial optimization problem of graph coloring [25]. The authors of the study proposed using qudits as graph nodes, parameterized by multidimensional spherical coordinates. In ref. [25], two strategies are considered: one involving initialization of qudits in random states combined with qudit-based gradient descent to minimize the cost function, and an adapted strategy of local quantum annealing based on qudits, which implements an adiabatic transition from a simple initial function to a problem-specific target function.

The application of physical principles of nuclear reactions is reflected in the study presenting the classical optimization algorithm NRO (Nuclear Reaction Optimization) [26]. This algorithm simulates the behavior of atomic nuclei in a confined space, including the processes of nuclear fission and nuclear fusion. Specifically, the nuclear fission process decomposes complex solutions into simpler ones to explore new regions of the solution

space, while the nuclear fusion process combines simple solutions to improve current results. The NRO algorithm operates in three stages. In the first stage, a large number of «small nuclei» are generated, representing initial solutions, which ensures exploration of the new solution space. Then, the «nuclei» interact with each other through nuclear fission and fusion processes. In the final stage, «stable nuclei» are formed, corresponding to optimal solutions. During testing, the NRO algorithm demonstrated superior capability in locating global optima in problems with numerous local minima, adaptability to various problem conditions, and applicability to both single-objective and multiobjective optimization.

In parallel with quantum-inspired developments, significant progress has been made in classical DMOEAs (Dynamic Multiobjective Evolutionary Algorithms), which specifically address the challenges posed by time-varying objectives and constraints. A recent study [27] proposes a DMOEA based on the classification of environmental change intensity and a collaborative prediction strategy to enhance adaptation in dynamic environments. The algorithm first refines the static optimization phase to improve the spatial resolution of individuals in the objective space, thereby increasing the sensitivity and accuracy of change detection. Upon detecting an environmental shift, it employs a mutual information-based metric to classify the intensity of the change—low, medium, or high—and accordingly adapts the velocity update rules of the particle swarm to avoid misleading evolutionary directions. Furthermore, a collaborative prediction mechanism is introduced to generate a forecasted population that closely approximates the true Pareto set in the new environment. This is followed by a dual individual screening strategy that intelligently combines promising candidates from both the predicted population and the pre-change archive to initialize the new generation. Experimental validation on 20 benchmark DMOPs (Dynamic Multiobjective Optimization Problems) demonstrates that this approach consistently outperforms state-of-the-art DMOEAs in terms of convergence, diversity, and responsiveness to environmental dynamics, highlighting the critical importance of explicitly modeling change intensity and leveraging predictive mechanisms in dynamic multiobjective settings.

Addressing the challenges posed by highly constrained optimization environments, ref. [28] introduces the MOACO-DCE (Multiobjective Ant Colony Optimization Algorithm Based on a Dynamic Constraint Evaluation Strategy). This method is specifically designed for HCMOPs (Highly Constrained Multiobjective Problems), such as vehicle routing and shop scheduling, where feasible regions are typically small, fragmented, and difficult to locate. To overcome the limitations of conventional evolutionary approaches in such landscapes, MOACO-DCE employs a dynamic constraint violation metric that quantifies the degree to which solutions violate problem constraints. Based on this metric, the population is adaptively partitioned into two subpopulations: one with evolutionary advantage (relatively low constraint violation) and another with severe constraint violations. For the former, a dynamic transfer probability-based evolutionary strategy is applied to accelerate convergence toward high-quality feasible solutions. For the latter, a Gaussian variation operator is introduced to enhance diversity and facilitate escape from infeasible regions. Furthermore, the algorithm features a collaborative pheromone updating mechanism that enables information exchange between subpopulations, along with a constraint-aware pheromone update rule that explicitly incorporates constraint violation levels into the reinforcement process. Experimental comparisons demonstrate that MOACO-DCE achieves superior performance on HCMOP benchmarks, particularly in maintaining feasibility while preserving solution diversity, highlighting the effectiveness of dynamically balancing constraint handling and evolutionary search in severely restricted search spaces.

Further advancing the field of dynamic multiobjective optimization, a recent study [29] introduces MOEA/D-MDDM—a prediction-based evolutionary algorithm that leverages

a MDDM (Multi-Directional Difference Model) to anticipate shifts in the Pareto-optimal solution set. Recognizing that rapid adaptation to environmental changes hinges on accurate initialization near the new Pareto-optimal front, the authors propose a dedicated Pareto-optimal solution set estimation strategy that generates multiple candidate populations based on historical trajectory data. These estimates are then fused through the MDDM to produce a refined prediction of the next Pareto-optimal solution set location, which guides the re-initialization of the population. To enhance robustness across diverse problem landscapes, the algorithm incorporates an adaptive crossover-rate mechanism that dynamically adjusts genetic diversity in response to the geometric structure of the evolving Pareto-optimal solution set, particularly effective for problems with single-modality characteristics and continuous manifolds. Comprehensive experiments on 19 benchmark DMOPs show that MOEA/D-MDDM outperforms six state-of-the-art DMOPs in tracking accuracy and convergence stability, underscoring the value of combining directional prediction with adaptive variation operators in dynamic environments.

3. Materials and Methods

3.1. The Process of Controlled Thermonuclear Fusion

Controlled thermonuclear fusion is a process of producing heavy atomic nuclei through the fusion of lighter nuclei, aimed at energy generation. Unlike the synthesis of heavy atomic nuclei in thermonuclear explosive devices [30], this process is controlled and sustained under engineered conditions.

Atomic nuclei consist of two types of nucleons—protons and neutrons—held together by the strong nuclear force. The binding energy of each nucleon to the rest of the nucleus depends on the total number of nucleons in the nucleus. This dependence is quantitatively described by the semi-empirical mass formula, which expresses the total binding energy as a sum of volume, surface, Coulomb, asymmetry, and pairing terms:

$$E_B(A, Z) = a_V A - a_S A^{\frac{2}{3}} - a_C \frac{Z(Z-1)}{A^{\frac{1}{3}}} - a_A \frac{(A-2Z)^2}{A} + \delta(A, Z), \quad (1)$$

where E_B —nuclear binding energy; A —mass number (total number of nucleons); a_V , a_S , a_C , a_A —coefficients of volume, surface, Coulomb, and asymmetry energies (according to the liquid drop model [31]); $\delta(A, Z)$ —pairing correction term.

For light nuclei, the binding energy per nucleon increases with the number of nucleons, whereas for heavy nuclei, it decreases. Adding nucleons to light nuclei or removing them from heavy nuclei results in a difference in binding energy. This energy difference is released as net energy, manifesting as the difference between the energy input required to initiate the reaction and the kinetic energy of the emitted particles.

Since protons in a nucleus carry a positive electric charge, nuclei must possess significant kinetic energy—sufficient to overcome the Coulomb barrier—in order to approach each other closely enough for nuclear forces to become effective. The Coulomb potential between two nuclei with charges $Z_1 e$ and $Z_2 e$ is given by:

$$V_C(r) = \frac{1}{4\pi\epsilon_0} \cdot \frac{Z_1 Z_2 e^2}{r}, \quad (2)$$

where $V_C(r)$ —Coulomb repulsion potential at the distance r between nuclei; Z_1 , Z_2 —atomic numbers of the colliding nuclei; e —elementary charge; ϵ_0 —vacuum permittivity.

Nuclear fusion reactions can occur with low probability even under thermodynamically non-equilibrium conditions, for example, by accelerating nuclei of one or more reaction components to high velocities and directing them onto a target containing nuclei

of another component [32]. At low energies, fusion is made possible by the quantum tunneling effect. Under conditions of extremely high matter density, picoscale nuclear reactions may occur, driven by zero-point oscillations of nuclei at the lattice points of a crystalline structure. Nevertheless, substantial energy release is achieved primarily through thermonuclear reactions occurring in a medium heated to extremely high temperatures, where overcoming the Coulomb barrier becomes a collective, large-scale process (with matter in the plasma state). For successful thermonuclear fusion, the high-temperature plasma must be confined for a sufficient duration to ensure effective reaction rates. This requirement is quantitatively expressed by the Lawson criterion [33], which defines the minimum product of plasma density, confinement time, and temperature necessary to achieve net energy gain:

$$n\tau_E T > L, \quad (3)$$

where n —plasma density; τ_E —energy confinement time; T —plasma temperature; L —threshold value (for the deuterium-tritium reaction, $L \approx 3 \times 10^{21} \text{ keV}\cdot\text{s}\cdot\text{m}^{-3}$, corresponding to the minimum triple product required for net energy gain).

In general, the thermonuclear fusion process can be described as follows: at sufficiently high temperatures, light atomic nuclei approach each other to distances where a strong but short-range interaction becomes significant. This interaction overcomes the Coulomb barrier between the positively charged nuclei, leading to their fusion and the formation of a new, heavier nucleus. During this process, part of the nucleon mass is converted into binding energy, and a large amount of energy is released due to the strong nuclear force.

In practice, for achieving controlled thermonuclear fusion, fusion reactions that occur at moderate temperatures and exhibit the highest cross-section values—characterizing the probability of particle reactions upon collision—are considered. Particular interest is focused on the reaction between the nuclei of heavy hydrogen isotopes—deuterium and tritium. Studies [34,35] show that initiating the fusion reaction of a deuterium-tritium mixture (Figure 1) requires less energy than the amount released during the reaction. This net energy gain is explicitly quantified by the deuterium-tritium fusion reaction:

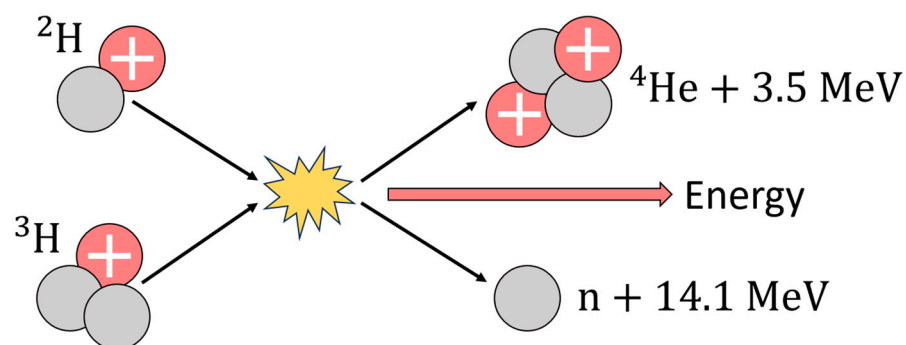


Figure 1. Deuterium-tritium fusion reaction scheme.

The most promising systems for achieving controlled thermonuclear fusion are those operating in steady-state or quasi-steady-state regimes. Among such systems are magnetic traps, which confine high-temperature plasma using a magnetic field. This field restricts the motion of charged particles, providing a magnetic insulation effect. The most widely used type of magnetic trap is the tokamak—a toroidal system in which the magnetic configuration is formed by external coils and an electric current flowing through the plasma [36]. Although tokamaks theoretically allow for indefinite confinement of individual

charged particles, particle collisions and the development of plasma turbulence lead to plasma losses.

In the context of a quantum-inspired multiobjective optimization algorithm, the concept of controlled thermonuclear fusion is proposed to be utilized in terms of the following aspects:

1. Each algorithm agent represents a particle in a state corresponding to a solution of the optimization problem. The ensemble of all agents forms a unified system—plasma.
2. Each agent possesses a certain energy—lower energy corresponds to a better solution.
3. During algorithm execution, energy is released through interactions between agents. This interaction is modeled as a process of information fusion or information exchange, which may improve solution quality.
4. The agent system has a temperature that controls the intensity of interactions. The initial temperature sets the initial energy level of the system.
5. System cooling is modeled as a gradual reduction in temperature, allowing the system to stabilize and converge toward optimal solutions. At high temperature, the system actively explores the solution space. At low temperature, solutions «condense» around local or global optima.

3.2. Qudit as a Multilevel Quantum Agent

A qudit [9] is a multilevel unit of quantum information that, unlike a conventional qubit, can exist in a discrete set of states of dimensionality $d \geq 3$. In general, a qudit can be expressed as:

$$\begin{aligned}
 |\psi\rangle &= \alpha_0|0\rangle + \alpha_1|1\rangle + \alpha_2|2\rangle + \alpha_3|3\rangle + \dots + \alpha_{d-1}|d-1\rangle \\
 &= \sum_{j=0}^{d-1} \alpha_j|j\rangle \\
 &= (\alpha_0, \alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{d-1})^T \in \mathbb{C}^d,
 \end{aligned}
 \tag{5}$$

where $|\psi\rangle$ is a quantum system existing simultaneously in more than two states; $|j\rangle$ is an orthonormal basis vector; $\alpha_0 \dots \alpha_{d-1}$ are complex probability amplitudes such that $\alpha_0 \dots \alpha_{d-1} \in \mathbb{C}^d$; $|\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 + \dots + |\alpha_{d-1}|^2 = 1$.

To simplify the treatment of the qudit as a physical system, its description in terms of the density matrix ρ is proposed [37]. In general, the density matrix is an operator that characterizes the quantum state of a physical system, since any physical observable can be represented as the expectation value of a corresponding Hermitian operator. If the quantum system is in a pure state described by a single wave function, the density matrix takes the following form:

$$\rho = |\Psi\rangle\langle\Psi|,
 \tag{6}$$

where Ψ is the wave function.

Substituting (5) into (6) yields:

$$\begin{aligned}
 \rho &= \left(\sum_{j=0}^{d-1} \alpha_j|j\rangle\right) \left(\sum_{k=0}^{d-1} \alpha_k^*\langle k|\right) \\
 &= \sum_{j=0}^{d-1} \sum_{k=0}^{d-1} \alpha_j \alpha_k^* |j\rangle\langle k| \Rightarrow \rho_{jk} = \alpha_j \alpha_k^*,
 \end{aligned}
 \tag{7}$$

where α_k^* denotes the complex conjugate of the amplitudes α_k , $k = \overline{0, d-1}$; ρ_{jk} is the element of the density matrix corresponding to the transition between the states $|j\rangle$ and $\langle k|$.

Since the interpretation of states with dimensionality $d \geq 3$ using the Bloch sphere is impossible, and Formulation (7) does not allow for an explicit analysis of the system’s coherence and dynamics, an alternative approach based on Heisenberg-Weyl operators [38] is

proposed. These operators are defined as unitary transformations acting on a d -dimensional Hilbert space \mathcal{H}_d :

$$\mathbf{W}_{p,q} = \sum_{n=0}^{d-1} e^{2\pi i n q/d} |n+p\rangle \langle n|, \tag{8}$$

where $p, q \in \{0, 1, 2, 3, \dots, d-1\}$; $|n+p\rangle$ is a cyclically shifted basis vector (modulo d); $e^{2\pi i n q/d}$ is a phase vector; for $\mathbf{W}_{p,q}$ the property $\mathbf{W}_{p,q}^\dagger \mathbf{W}_{p,q} = \mathbf{I}$ holds, with \mathbf{I} being the identity matrix and $\mathbf{W}_{p,q}^\dagger$ is the Hermitian conjugate of $\mathbf{W}_{p,q}$. Moreover, $\text{Tr}(\mathbf{W}_{p,q}^\dagger \mathbf{W}_{p',q'}) = d \times \delta_{p,p'} \delta_{q,q'}$, where $\delta_{p,p'}$ and $\delta_{q,q'}$ are Kronecker delta symbols.

The expansion of a bounded density matrix operator ρ in \mathbb{C}^d in the Heisenberg-Weyl operator basis takes the following form:

$$\rho = \frac{1}{d} \sum_{p=q=0}^{d-1} \mathbf{c}_{p,q} \mathbf{W}_{p,q}, \tag{9}$$

$$\mathbf{c}_{p,q} = \text{Tr}(\rho \mathbf{W}_{p,q}^\dagger), \tag{10}$$

where $\mathbf{c}_{p,q}$ are the expansion coefficients, determined based on the fact that the Heisenberg-Weyl operators form a complete orthonormal basis in the $d \times d$ -dimensional space.

The Hermitian conjugate of the Heisenberg-Weyl operators $\mathbf{W}_{p,q}^\dagger$ can be expressed, taking into account (8), as:

$$\mathbf{W}_{p,q}^\dagger = \sum_{m=0}^{d-1} e^{-2\pi i m q/d} |m\rangle \langle m+p|. \tag{11}$$

Then, using the linearity property of the trace and substituting (11) into (10), we obtain:

$$\begin{aligned} \mathbf{c}_{p,q} &= \text{Tr}(\rho \sum_{m=0}^{d-1} e^{-2\pi i m q/d} |m\rangle \langle m+p|) \\ &= \sum_{m=0}^{d-1} e^{-2\pi i m q/d} \text{Tr}(\rho |m\rangle \langle m+p|). \end{aligned} \tag{12}$$

The trace of the product of operators can be written in terms of matrix elements as $\text{Tr}(\rho |m\rangle \langle m+p|) = \langle m+p | \rho |m\rangle$. Re-defining the index m to align with the state $|m\rangle$ by substituting $m' = m + p$ (modulo d) leads to the transformation $\langle m+p | \rho |m\rangle \rightarrow \langle m | \rho |m+p\rangle$. As a result, the expression for the expansion coefficients $\mathbf{c}_{p,q}$ reduces to the following equation:

$$\mathbf{c}_{p,q} = \sum_{m=0}^{d-1} \langle m | \rho |m+p\rangle e^{-2\pi i m q/d}. \tag{13}$$

Substituting the expansion coefficients (13) and the operators (8) into (9) results in the final density matrix ρ taking the form:

$$\begin{aligned} \rho &= \frac{1}{d} \sum_{p=q=0}^{d-1} \left(\sum_{m=0}^{d-1} \langle m | \rho |m+p\rangle e^{-2\pi i m q/d} \right) \left(\sum_{n=0}^{d-1} e^{2\pi i n q/d} |n+p\rangle \langle n| \right) \\ &= \frac{1}{d} \sum_{p=q=0}^{d-1} \sum_{m=n=0}^{d-1} \langle m | \rho |m+p\rangle e^{-2\pi i m q/d} e^{2\pi i n q/d} |n+p\rangle \langle n| \\ &= \frac{1}{d} \sum_{p=q=0}^{d-1} \sum_{m=n=0}^{d-1} \langle m | \rho |m+p\rangle e^{2\pi i q(n-m)/d} |n+p\rangle \langle n| \\ &= \frac{1}{d} \sum_{p=m=0}^{d-1} d \langle m | \rho |m+p\rangle |m+p\rangle \langle m| \\ &= \sum_{p=m=0}^{d-1} \langle m | \rho |m+p\rangle |m+p\rangle \langle m| \\ &= \sum_{p=m=0}^{d-1} \rho_{m,m+p} |m+p\rangle \langle m|, \end{aligned} \tag{14}$$

where $\sum_{q=0}^{d-1} e^{2\pi i q(n-m)/d} = d\delta_{n,m}$, since the exponential term completes a full cycle when $n \neq m$; $\rho_{m,m+p} = \langle m | \rho |m+p\rangle$ denotes the elements of the density matrix.

Since, in the context of a quantum-inspired multiobjective optimization algorithm, a qudit represents an agent, and a collection of such agents forms a unified system simulating a plasma, there arises a need to describe the interaction characteristics of agents in terms of the concept of controlled thermonuclear fusion.

Let ρ_i be the density matrix of the i -th qudit, satisfying the conditions $\rho_i^\dagger = \rho_i$, $\text{Tr}(\rho_i) = 1$, $\rho_i \geq 0$. The number of qudits in the system is N . Then, the probability of interaction between adjacent qudits ρ_i and ρ_{i+1} is calculated as:

$$P_\delta = \exp(-|E_i - E_{i+1}|), \tag{15}$$

where $E_i = \text{Tr}(\rho_i \hat{H})$ and $E_{i+1} = \text{Tr}(\rho_{i+1} \hat{H})$ denote the energies of the i -th and $(i + 1)$ -th qudits, respectively, with $\hat{H} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d - 1 \end{pmatrix}$ being the Hamiltonian.

If the condition $r < P_\delta$, where $r \in [0, 1)$ is a random number modeling the stochasticity of the quantum process, is satisfied, then the new density matrices are calculated as a linear combination of the current ones:

$$\begin{aligned} \rho'_i &= \alpha \rho_i + (1 - \alpha) \rho_{i+1}, \\ \rho'_{i+1} &= \alpha \rho_{i+1} + (1 - \alpha) \rho_i, \end{aligned} \tag{16}$$

where α is a coefficient with a random value drawn uniformly from the interval $[0, 1)$, determining the contribution of each density matrix to the new state. This allows the creation of new states described by the density matrices ρ'_i and ρ'_{i+1} , which represent a «mixture» of the original states. Specifically, the state described by ρ'_i is formed as a weighted sum of the states represented by ρ_i and ρ_{i+1} , where the weight coefficient for ρ_i is α and that for ρ_{i+1} is $1 - \alpha$. Similarly, the state described by ρ'_{i+1} is formed as the same weighted combination of ρ_{i+1} and ρ_i with identical weights.

The new density matrices ρ'_i and ρ'_{i+1} preserve the Hermitian property, since a linear combination of Hermitian matrices with real coefficients is also Hermitian:

$$\begin{aligned} (\rho'_i)^\dagger &= (\alpha \rho_i + (1 - \alpha) \rho_{i+1})^\dagger = \alpha \rho_i^\dagger + (1 - \alpha) \rho_{i+1}^\dagger = \alpha \rho_i + (1 - \alpha) \rho_{i+1} = \rho'_i, \\ (\rho'_{i+1})^\dagger &= (\alpha \rho_{i+1} + (1 - \alpha) \rho_i)^\dagger = \alpha \rho_{i+1}^\dagger + (1 - \alpha) \rho_i^\dagger = \alpha \rho_{i+1} + (1 - \alpha) \rho_i = \rho'_{i+1}. \end{aligned} \tag{17}$$

After updating, the density matrices ρ'_i and ρ'_{i+1} are normalized to preserve the property $\text{Tr}(\rho) = 1$:

$$\begin{aligned} \rho'_i &\leftarrow \frac{\rho'_i}{\text{Tr}(\rho'_i)}, \\ \rho'_{i+1} &\leftarrow \frac{\rho'_{i+1}}{\text{Tr}(\rho'_{i+1})}. \end{aligned} \tag{18}$$

When agents in the system interact, energy is released. This process is simulated through local optimization of solutions from the Pareto front obtained by the quantum-inspired algorithm. Here, the Pareto front represents a set of non-dominated solution vectors $F = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \dots, \mathbf{s}_v\}$, where a non-dominated solution vector \mathbf{s}_b with $b = \overline{1, v}$ corresponds to a solution vector $\mathbf{x}_2 = (x_{2,1}, x_{2,2}, x_{2,3}, \dots, x_{2,n})$ (n —total number of variables) from the set of all solutions $K = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_Q\}$ with $z = \overline{1, Q}$ (Q —total number of solutions). Each non-dominated solution vector \mathbf{s}_b is characterized by a vector of objective function values $\mathbf{f}_{\mathbf{x}_2} = [f_1(\mathbf{x}_2), f_2(\mathbf{x}_2), f_3(\mathbf{x}_2), \dots, f_h(\mathbf{x}_2)]$ (h —number of objectives) and belongs to the feasible region defined by the constraints $g_\tau(\mathbf{x}_2) \in G$ (G —set of constraints, $\tau = \overline{1, Q}$, Q —total number of constraints). For each solution vector \mathbf{x}_2 , the non-domination condition is checked. For example, in a minimization problem, for two solution vectors $\mathbf{x}_{21}, \mathbf{x}_{22} \in K$, we say that \mathbf{x}_{21} dominates \mathbf{x}_{22} if the following conditions are satisfied:

$$\forall l \in \{1, 2, 3, \dots, h\}, f_l(\mathbf{x}_{21}) \leq f_l(\mathbf{x}_{22}) \wedge \exists l : f_l(\mathbf{x}_{21}) < f_l(\mathbf{x}_{22}). \tag{19}$$

Similarly, in a maximization problem, for two solution vectors $\mathbf{x}_{21}, \mathbf{x}_{22} \in K$, we say that \mathbf{x}_{21} dominates \mathbf{x}_{22} if the following conditions are satisfied:

$$\forall l \in \{1, 2, 3, \dots, h\}, f_l(\mathbf{x}_{21}) \geq f_l(\mathbf{x}_{22}) \wedge \exists l : f_l(\mathbf{x}_{21}) > f_l(\mathbf{x}_{22}). \tag{20}$$

During the physical process of thermonuclear fusion, plasma inevitably cools down due to energy losses through radiation, absorption by the reactor walls, plasma instabilities, and other factors [39]. By analogy with physical plasma, cooling in the agent system is modeled as a gradual reduction in the parameter T , which controls the system temperature. This leads to stabilization of the probabilities of quantum states, and the system evolves toward a state of minimum energy.

During the cooling process, for each qudit the density matrix ρ_i ($i = \overline{1, N}$, N being the number of qudits in the system) is scaled using the current temperature T :

$$\rho_i^{exp} = \exp\left(\frac{\rho_i}{T}\right), \tag{21}$$

where $T > 0$.

The resulting matrix ρ_i^{exp} is normalized such that its trace equals 1:

$$\rho'_i = \frac{\rho_i^{exp}}{\text{Tr}(\rho_i^{exp})}. \tag{22}$$

After updating all density matrices, the system parameter T is decreased according to the formula:

$$T' = T \cdot \gamma^{0.5}, \tag{23}$$

where T is the current temperature; T' is the new temperature; $\gamma \in (0, 1)$ is the cooling coefficient.

3.3. Description of the Algorithm

In general, the structure of the quantum-inspired multiobjective optimization algorithm based on thermonuclear fusion (hereinafter referred to as TF-QIMOA) consists of a set of interconnected components: initialization of density matrices of quantum states, evaluation of solution fitness, simulation of quantum interactions between qudits as particles, local optimization of solutions from the Pareto front, and simulation of the cooling process of the qudit system.

The execution of the algorithm for initializing density matrices of quantum states (Algorithm 1) begins with the construction of the generalized Hadamard gate matrix H , which is required to create a uniform superposition of all basis states of the qudit in d -dimensional space [10]. Next, for each qudit i in the system of N qudits, the initial state is set to $|\psi\rangle_{init} = |0\rangle$, $|0\rangle = (1, 0, 0, \dots, 0)^T \in \mathbb{C}^d$. The generalized Hadamard gate is applied to the initial state $|\psi\rangle_{init}$, transforming it into the superposition state $|\psi\rangle_{sup}$. Then, a random noise term $\eta \sim \mathcal{N}(0, \sigma^2)$ is added to the superposition state $|\psi\rangle_{sup}$ to simulate quantum fluctuations in the system, where η is defined as a noise vector generated from a normal distribution with zero mean and variance σ^2 . The resulting state $|\psi\rangle_{noisy}$ is normalized to obtain the state $|\psi\rangle_{norm} = \frac{|\psi\rangle_{noisy}}{||\psi\rangle_{noisy}|}$, where $||\psi\rangle_{noisy}| = \sqrt{\sum_{j=0}^{d-1} |\psi_{noisy,j}|^2}$. Afterwards, the density matrix ρ_i is constructed for the current qudit i , computed according to (14) as $\rho_i = \sum_{p=m=0}^{d-1} c_m c_{m+p}^* |m+p\rangle\langle m|$, taking into account that $\rho_{m,m+p} = c_m c_{m+p}^* \rightarrow |\rho_{m,m+p}| = |c_m| \times |c_{m+p}^*|$, where c_m is the probability amplitude of the state $|m\rangle$, and c_{m+p}^* is the complex conjugate of the probability amplitude

of the state $|m + p\rangle$. The resulting density matrix ρ_i is added to the list ρ_{lst} . After the list ρ_{lst} is populated with density matrices for all N qudits, it is returned as the output of Algorithm 1.

Algorithm 1. Initialization of Density Matrices

Input: $d \in \mathbb{N}$ \triangleright dimension of the qudit (number of basic states).
 $N \in \mathbb{N}$ \triangleright number of qudits in the system.
 $\rho_{lst} = \emptyset$ \triangleright list for storing density matrices.

1. $H(\kappa, \nu) = \frac{1}{\sqrt{d}} e^{2\pi i \kappa \nu / d}$ \triangleright define the generalized Hadamard gate matrix
2. with $\kappa, \nu = \overline{0, d-1}$.
3. **For each** qudit $i = \overline{1, N}$ **do:**
4. $|\psi\rangle_{init} = (1, 0, 0, \dots, 0)^T \in \mathbb{C}^d$ \triangleright set initial state to $|0\rangle$.
5. $|\psi\rangle_{sup} = H|\psi\rangle_{init}$ \triangleright apply generalized Hadamard gate to initial state.
6. $|\psi\rangle_{noisy} = |\psi\rangle_{sup} + \eta$ \triangleright add random noise η to the superposition state
7. $|\psi\rangle_{sup}$.
8. $|\psi\rangle_{norm} = \frac{|\psi\rangle_{noisy}}{\| |\psi\rangle_{noisy} \|}$ \triangleright normalize the state.
9. $\rho_i = \sum_{p=m=0}^{d-1} c_m c_m^* |m+p\rangle\langle m|$ \triangleright construct the density matrix for the current qudit.
10. $\rho_{lst} \leftarrow \rho_{lst} \cup \{\rho_i\}$ \triangleright add density matrix ρ_i to list ρ_{lst} .
11. **End loop.**
12. **Return** ρ_{lst} \triangleright list of density matrices.

Fitness evaluation (Algorithm 2) begins with performing the Cartesian product of the set of qudit indices $i_{ind} \in [0, N)$ and the set of state sample indices $o_{spl} \in [0, Q)$, where N is the number of qudits in the system, Q is the number of state samples per qudit. From the resulting set M after computing the Cartesian product, the qudit index i_{ind} is iteratively selected and passed to the function $\Theta(i_{ind})$ for mapping the qudit to the feasible region. This function can be written as:

$$\Theta(i_{ind}) = \begin{cases} (lower_1, upper_1)_{i_{ind}}, & n = 1 \\ ([lower_1, upper_1], [lower_2, upper_2], \dots, [lower_n, upper_n])_{i_{ind}}, & n > 1 \end{cases} \quad (24)$$

where $lower_\varepsilon$ is the lower bound of the variable's value range, $upper_\varepsilon$ is the upper bound of the variable's value range; $\varepsilon = \overline{1, n}$, n is the number of variables.

Based on the mapping $R_{i_{ind}} = \Theta(i_{ind})$, a random solution vector from the feasible region $R_{i_{ind}}$ is generated as $\mathbf{x}_2 \sim U(R_{i_{ind}})$, where U denotes the uniform distribution. If the condition of satisfying all constraints is met, i.e., $\forall \tau, \mathbb{Y}(g_\tau(\mathbf{x}_2)) = 1$ (\mathbb{Y} is the indicator function, $g_\tau(\mathbf{x}_2) \in G$), then the values of all objective functions $\mathbf{f}_{\mathbf{x}_2}$ are calculated, and the set of non-dominated solution vectors F is updated as:

$$F \leftarrow \{ \mathbf{x}_2 \in F : \neg \exists \mathbf{y}_b \in F (\mathbf{y}_b \prec \mathbf{x}_2) \} \cup \{ \mathbf{x}_2 : \neg \exists \mathbf{y}_b \in F (\mathbf{x}_2 \prec \mathbf{y}_b) \}, \quad (25)$$

where \mathbf{x}_2 is the vector of the current candidate solution; \mathbf{y}_b is a solution vector from the set of already found non-dominated solutions; z is the index of the current solution; b is the index of a solution in the non-dominated set; the operator \prec denotes the dominance relation.

Then, the new solution vector \mathbf{x}_2 is added to the list of all solution vectors S_{all} . At the end of Algorithm 2, the set of non-dominated solution vectors F and the list of all solution vectors S_{all} are returned.

Algorithm 2. Fitness Evaluation

Input: $S_{all} = \emptyset$ ▷ list of all solution vectors.
 $F = \emptyset$ ▷ set of non-dominated solution vectors.
 $N \in \mathbb{N}$ ▷ number of qudits in the system.
 $Q \in \mathbb{N}$ ▷ number of samples per qudit.
 $M = \{(i_{ind}, o_{spl}) : i_{ind} \in [0, N), o_{spl} \in [0, Q)\}$ ▷ Cartesian product of the set of qudit indices and the set of sample indices.

1. $M = \{(i_{ind}, o_{spl}) : i_{ind} \in [0, N), o_{spl} \in [0, Q)\}$ ▷ Cartesian product of the set of qudit indices and the set of sample indices.
- 2.
3. **For each** qudit index i_{ind} **in** $(i_{ind}, o_{spl}) \in M$ **do:**
4. $R_{i_{ind}} = \mathcal{O}(i_{ind})$ ▷ map qudit to feasible region.
5. $\mathbf{x}_2 \sim U(R_{i_{ind}})$ ▷ generate a random solution vector \mathbf{x}_2 .
6. **If** $\forall \tau, \forall (g_\tau(\mathbf{x}_2)) = 1$ **then:**
7. $\mathbf{f}_{\mathbf{x}_2} = [f_1(\mathbf{x}_2), f_2(\mathbf{x}_2), f_3(\mathbf{x}_2), \dots, f_h(\mathbf{x}_2)]$ ▷ evaluate all objective functions.
8. $F \leftarrow \{ \mathbf{x}_2 \in F : \neg \exists \mathbf{y}_h \in F (\mathbf{y}_h \prec \mathbf{x}_2) \} \cup \{ \mathbf{x}_2 : \neg \exists \mathbf{y}_h \in F (\mathbf{x}_2 \prec \mathbf{y}_h) \}$ ▷ update the set of non-dominated solutions.
9. set of non-dominated solutions.
10. $S_{all} \leftarrow S_{all} \cup \{ (\mathbf{x}_2, \mathbf{f}_{\mathbf{x}_2}) \}$ ▷ add new solution vector to the list of all solution vectors.
- 11.
12. **End loop.**
13. **Return** F, S_{all} ▷ updated set of non-dominated solution vectors and list of all solution vectors.
- 14.

Algorithm 3 simulates quantum interaction between qudits in the system. The algorithm takes as input the list of density matrices ρ_{lst} . In a loop, the current qudit i is iteratively selected under the condition $i \in \{2 \cdot \epsilon + 1 : \epsilon \in \mathbb{Z}, 0 \leq 2 \cdot \epsilon + 1 < N\}$, and the interaction probability P_δ between ρ_i and ρ_{i+1} is computed using Formula (15). If the condition $r < P_\delta$, where $r \in [0, 1)$, is satisfied, the following steps are performed sequentially: calculation of the new density matrices ρ'_i and ρ'_{i+1} according to Equation (16), normalization of the density matrices ρ'_i and ρ'_{i+1} using Equation (18), update of the original density matrices $\rho_i \leftarrow \rho'_i$ and $\rho_{i+1} \leftarrow \rho'_{i+1}$. Upon completion of the algorithm, the updated list of density matrices ρ_{lst} is returned.

Algorithm 4 implements local optimization of solutions from the Pareto front. In the first step, the weight vector \mathbf{w} is initialized. If $\mathbf{w} = \emptyset$, then the elements w_l ($l = \overline{1, h}$) of the vector \mathbf{w} are set as $w_l = 1/h$. Then, an empty set D is populated with discrete grids generated as:

$$D \leftarrow D \cup \left\{ a_\epsilon + \vartheta \cdot \frac{n_\epsilon - a_\epsilon}{d - 1} : \vartheta = \overline{0, d - 1} \right\}, \tag{26}$$

where $[a_\epsilon, n_\epsilon]$ is the range of variable values, $\epsilon = \overline{1, n}$ and n is the number of variables.

After that, the Cartesian product of all discrete grids is calculated as:

$$V = D_1 \times D_2 \times \dots \times D_n, \tag{27}$$

where $D_\epsilon \in D$.

The result is converted into a list of grid point combination vectors:

$$V \leftarrow \{ \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{d^n} \}, \tag{28}$$

where $\mathbf{v}_k \in V \wedge \mathbf{v}_k \in \mathbb{Q}^n, k = \overline{1, d^n}$.

The second step of the algorithm begins with the creation of a list P_{all} , into which the vectors of diagonal elements of the density matrices $\rho_i \in \rho_{lst}$ are added. Each diagonal element represents the probability of the quantum system being in the corresponding basis state. Next, a unified vector of averaged probability values is created:

$$\mathbf{P}_{mean} = \{ p_1, p_2, p_3, \dots, p_j \}, \tag{29}$$

where $p_j = \frac{\sum_{i=1}^N p_{ij}}{N}, p_{ij} \in P_{all}, j = \overline{1, d}$.

Algorithm 3. Quantum Interaction

Input: $\rho_{lst} = \{\rho_1, \rho_2, \rho_3, \dots, \rho_i\} \triangleright$ list of density matrices $\rho_i \in \mathbb{C}^{d \times d}$, $i = \overline{1, N}$.
 $N \in \mathbb{N} \triangleright$ number of qudits in the system.

1. **For each** qudit $i \in \{2 \cdot \epsilon + 1 : \epsilon \in \mathbb{Z}, 0 \leq 2 \cdot \epsilon + 1 < N\}$ **do:**
2. $P_\delta = \exp(-|E_i - E_{i+1}|) \triangleright$ calculate the probability of interaction between
3. neighboring qudits ρ_i and ρ_{i+1} .
4. **If** $r < P_\delta$ **then:**
5. $\rho'_i = \alpha \rho_i + (1 - \alpha) \rho_{i+1} \triangleright$ calculate new density matrix ρ'_i .
6. $\rho'_{i+1} = \alpha \rho_{i+1} + (1 - \alpha) \rho_i \triangleright$ calculate new density matrix ρ'_{i+1} .
7. $\rho'_i \leftarrow \frac{\rho'_i}{\text{Tr}(\rho'_i)} \triangleright$ normalize the density matrix ρ'_i .
8. $\rho'_{i+1} \leftarrow \frac{\rho'_{i+1}}{\text{Tr}(\rho'_{i+1})} \triangleright$ normalize the density matrix ρ'_{i+1} .
9. $\rho_i \leftarrow \rho'_i \triangleright$ update the density matrix ρ_i .
10. $\rho_{i+1} \leftarrow \rho'_{i+1} \triangleright$ update the density matrix ρ_{i+1} .
11. **End loop.**
12. **Return** $\rho_{lst} \triangleright$ updated list of density matrices.

Then, a list L is generated, containing tuples of solutions and the corresponding products of the scalar function value $\varepsilon(\mathbf{v}_k, \mathbf{w}, B)$ and a randomly selected probability $p_r \in \mathbf{p}_{mean}$ for each feasible point \mathbf{v}_k . Here, the value of the function $\varepsilon(\mathbf{v}_k, \mathbf{w}, B)$ is calculated as:

$$\varepsilon(\mathbf{v}_k, \mathbf{w}, B) = \sum_{l=1}^h w_l \cdot f_l(\mathbf{v}_k), \tag{30}$$

where $w_l \in \mathbf{w}$; $f_l \in B$; B is the set of objective functions; h is the number of objective functions.

If the list L remains empty, the values of all objective functions \mathbf{f}_{s_b} are computed, and a tuple containing the solution vector \mathbf{s}_b and the corresponding objective function value vector \mathbf{f}_{s_b} is returned.

At the end of the second step, the list L is sorted according to an order determined by the optimization type σ : if $\sigma = \text{“min”}$, the sorting is performed in ascending order as $f(\mathbf{v}_{\chi(1)}) \leq f(\mathbf{v}_{\chi(2)}) \leq \dots \leq f(\mathbf{v}_{\chi(dn)})$, if $\sigma = \text{“max”}$, in descending order as $f(\mathbf{v}_{\chi(1)}) \geq f(\mathbf{v}_{\chi(2)}) \geq \dots \geq f(\mathbf{v}_{\chi(dn)})$, where χ denotes the sorting permutation. The first ζ solutions from the resulting sorted list L_{sorted} are then selected and placed into the list L_{top} .

In the third step of the algorithm, a set C is created, containing tuples of candidate solutions $c \in L_{top}$ and perturbed point vectors [40–42], calculated as:

$$\mathbf{t} = \Upsilon(c + \delta, \phi, u), \tag{31}$$

where Υ denotes the operation of clamping the values of $c + \delta$ within the bounds $[\phi, u]$, δ is a random perturbation vector.

Then, for each perturbed point vector \mathbf{t} from $(c, \mathbf{t}) \in C$, if the feasibility condition is satisfied—i.e., $\forall \tau, \Upsilon(g_\tau(\mathbf{t})) = 1$ (Υ is the indicator function, $g_\tau(\mathbf{t}) \in G$)—the value $\gamma_{\mathbf{t}}$ is calculated as the product of the scalar function $\varepsilon(\mathbf{t}, \mathbf{w}, B)$ and a randomly selected probability $p_r \in \mathbf{p}_{mean}$. The scalar function $\varepsilon(\mathbf{t}, \mathbf{w}, B)$ is calculated analogously to (30). Subsequently, if the condition $\sigma = \text{“min”} \wedge \gamma_{\mathbf{t}} < \wp$ or $\sigma = \text{“max”} \wedge \gamma_{\mathbf{t}} > \wp$ holds, the best solution vector \mathbf{s}_{best} is updated to \mathbf{t} , and the best value \wp is replaced with $\gamma_{\mathbf{t}}$.

In the fourth step of the algorithm, the condition $\mathbf{s}_{best} = \emptyset$ is checked. If this condition holds, the values of all objective functions \mathbf{f}_{s_b} are calculated, and a tuple containing the solution vector \mathbf{s}_b and the objective function value vector \mathbf{f}_{s_b} is returned. Otherwise, the values of all objective functions $\mathbf{f}_{s_{best}}$ are calculated and the set of non-dominated solution vectors F is updated as:

$$F \leftarrow F' \cup \{\mathbf{s}_{best} : \neg \exists \mathbf{s}_{crt} \in F' (\mathbf{s}_{crt} \prec \mathbf{s}_{best})\}, \tag{32}$$

where \mathbf{s}_{crt} is the current solution vector.

The updated set of non-dominated solutions F is then returned at the end of Algorithm 4.

Algorithm 4. Local Optimization of Solutions

Input: $F = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \dots, \mathbf{s}_v\} \triangleright$ set of non-dominated solution vectors $\mathbf{s}_b \in \mathbb{Q}^{\mathbf{N}}$, where $b = \overline{1, v}$.
 $G = \{g_1, g_2, g_3, \dots, g_Q\} \triangleright$ set of constraints, Q —number of constraints.
 $B = \{f_1, f_2, f_3, \dots, f_h\} \triangleright$ set of objective functions.
 $\rho_{lst} = \{\rho_1, \rho_2, \rho_3, \dots, \rho_i\} \triangleright$ list of density matrices $\rho_i \in \mathbb{C}^{d \times d}$, where $i = \overline{1, N}$.
 $D = \emptyset \triangleright$ set of discrete grids for variables.
 $L = \emptyset \triangleright$ list of tuples of solutions and function values.
 $\mathbf{w} \in \mathbb{Q}^h \triangleright$ weight coefficient vector of dimension h .
 $\mathbf{s}_{best} = \emptyset \triangleright$ vector storing the best solution values.
 $N \in \mathbb{N} \triangleright$ number of qudits in the system.
 $h \in \mathbb{N} \triangleright$ number of objective functions.
 $d \in \mathbb{N} \triangleright$ qudit dimension (number of basic states).
 $\zeta \in \mathbb{N} \triangleright$ number of best solutions.
 $\overline{\omega} \in \mathbb{N} \triangleright$ maximum number of iterations.
 $\mathcal{O} \in \{\text{“min”}, \text{“max”}\} \triangleright$ optimization type.
 $\wp \in \mathbb{R} \triangleright$ best value.

1. $\wp \leftarrow +\infty$ **if** $\mathcal{O} = \text{“min”}$ **else** $-\infty$.
2. **For each** vector $\mathbf{s}_b \in F$ **do**:
3. **if** $\neg \exists \mathbf{s}'_b \in F : \mathbf{s}'_b \prec \mathbf{s}_b$ **then**:
4. **if** $\mathbf{w} = \emptyset$ **then**:
5. $w_l = 1/h \triangleright$ set the weight coefficient $w_l \in \mathbf{w}$, $\forall l \in \{1, 2, 3, \dots, h\}$.
6. $D \leftarrow D \cup \{a_\varepsilon + \vartheta \cdot \frac{n_\varepsilon - a_\varepsilon}{d-1} : \vartheta = \overline{0, d-1}\} \triangleright$ generate a discrete grid for
7. variable ε over the range $[a_\varepsilon, n_\varepsilon]$ ($\varepsilon = \overline{1, N}$) and add it to the set of all
8. discrete grids.
9. $V = D_1 \times D_2 \times \dots \times D_N \triangleright$ Cartesian product of all discrete grids.
10. $V \leftarrow \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{d^{\mathbf{N}}}\} \triangleright$ convert V into a list of grid point vectors,
11. where $\mathbf{v}_k \in V \wedge \mathbf{v}_k \in \mathbb{Q}^{\mathbf{N}}$, $k = \overline{1, d^{\mathbf{N}}}$.
12. $P_{all} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_i\} \triangleright$ create a list of vectors containing the diagonal
13. elements of the density matrices from ρ_{lst} , where $i = \overline{1, N}$.
14. $\mathbf{p}_{mean} = \{p_1, p_2, p_3, \dots, p_j\} \triangleright$ calculate the vector of mean probability
15. values, where $p_j = (\sum_{i=1}^N p_{i,j})/N$, with $p_{i,j} \in P_{all}$, $j = \overline{1, d}$.
16. $L = \{(\mathbf{v}_k, \varepsilon(\mathbf{v}_k, \mathbf{w}, B) \cdot p_r) : p_r \in \mathbf{p}_{mean}, \mathbf{v}_k \in V \wedge \forall g_\tau \in G, \forall (g_\tau(\mathbf{v}_k)) = 1\} \triangleright$
17. create a list of tuples containing feasible solutions \mathbf{v}_k and the
18. corresponding values of the scalar function $\varepsilon(\mathbf{v}_k, \mathbf{w}, B)$ scaled by
19. a randomly selected probability p_r from \mathbf{p}_{mean} .
20. **if** $L = \emptyset$ **then**:
21. $\mathbf{f}_{s_b} = [f_1(\mathbf{s}_b), f_2(\mathbf{s}_b), f_3(\mathbf{s}_b), \dots, f_h(\mathbf{s}_b)] \triangleright$ create the objective function
22. value vector for solution \mathbf{s}_b .
23. **Return** $(\mathbf{s}_b, \mathbf{f}_{s_b}) \triangleright$ a tuple containing the original solution and
24. its corresponding objective function value vector.
25. $L_{sorted} = \left\{ \left(\mathbf{v}_{\chi(1)}, f(\mathbf{v}_{\chi(1)}) \right), \left(\mathbf{v}_{\chi(2)}, f(\mathbf{v}_{\chi(2)}) \right), \dots, \left(\mathbf{v}_{\chi(d^{\mathbf{N}})}, f(\mathbf{v}_{\chi(d^{\mathbf{N}})}) \right) \right\} \triangleright$
26. create a sorted list of tuples $(\mathbf{v}_{\chi(k)}, f(\mathbf{v}_{\chi(k)})) \in L$, ordered according to
27. the sorting permutation χ .
28. $L_{top} = \{(\mathbf{v}_\zeta, f(\mathbf{v}_\zeta)) : (\mathbf{v}_\zeta, f(\mathbf{v}_\zeta)) \in L_{sorted}, \zeta = \overline{1, \zeta}\} \triangleright$ select the first
29. ζ tuples from the list L_{sorted} .
30. **For each** iteration $\overline{\omega}_i = \overline{1, \overline{\omega}}$ **do**:
31. **For each** candidate $c \in L_{top}$ **do**:
32. $C = \{(c, \mathbf{t}) : c \in L_{top}, \mathbf{t} = \Upsilon(c + \delta, \phi, u)\} \triangleright$ create a set of tuples
33. containing candidate solutions and perturbed point vectors,
34. where perturbation is clamped to the feasible range $[\phi, u]$.
35. **End loop.**
36. **End loop.**
37. **For each** pair $(c, \mathbf{t}) \in C$ **do**:
38. **if** $\forall \tau, \forall (g_\tau(\mathbf{t})) = 1$ **then**:
39. $\gamma_{\mathbf{t}} = \varepsilon(\mathbf{t}, \mathbf{w}, B) \cdot p_r \triangleright$ calculate the value $\gamma_{\mathbf{t}}$ as the product of the
40. scalar function $\varepsilon(\mathbf{t}, \mathbf{w}, B)$ and a randomly selected probability
41. p_r from the vector \mathbf{p}_{mean} , for the perturbed point \mathbf{t} .
42. **if** $(\mathcal{O} = \text{“min”} \wedge \gamma_{\mathbf{t}} < \wp) \vee (\mathcal{O} = \text{“max”} \wedge \gamma_{\mathbf{t}} > \wp)$ **then**:
43. $\mathbf{s}_{best} \leftarrow \mathbf{t} \triangleright$ update the value of \mathbf{s}_{best} .
44. $\wp \leftarrow \gamma_{\mathbf{t}} \triangleright$ update the value of \wp .

Algorithm 4. Cont.

```

45.      End loop.
46.      If  $s_{best} = \emptyset$  then:
47.           $f_{s_b} = [f_1(s_b), f_2(s_b), f_3(s_b), \dots, f_h(s_b)]$   $\triangleright$  create the objective function
48.          value vector for solution  $s_b$ .
49.          Return  $(s_b, f_{s_b})$   $\triangleright$  a tuple containing the original solution and
50.          its corresponding objective function value vector.
51.           $f_{s_{best}} = [f_1(s_{best}), f_2(s_{best}), f_3(s_{best}), \dots, f_h(s_{best})]$   $\triangleright$  create the objective
52.          function value vector for the best solution  $s_{best}$ .
53.           $F \leftarrow F' \cup \{s_{best} : \neg \exists s_{crt} \in F' (s_{crt} \prec s_{best})\}$   $\triangleright$  update the set of
54.          non-dominated solution vectors, where  $F' = \{s_{crt} \in F : \neg (s_{best} \prec s_{crt})\}$ .
55.      End loop.
56.      Return  $F$   $\triangleright$  updated set of non-dominated solution vectors.
    
```

Algorithm 5 describes the cooling process of the qudit system. The algorithm takes as input the list of density matrices ρ_{lst} and the current system temperature T . In a loop, each density matrix $\rho_i \in \rho_{lst}$ is scaled using the current temperature T according to Equation (21). The resulting matrix ρ_i^{exp} is then normalized using Equation (22), and the result is written back to ρ_i . After all matrices in ρ_{lst} have been updated, the temperature T is decreased according to Equation (23). At the end of the algorithm, the updated list of density matrices ρ_{lst} and the updated temperature T are returned.

Algorithm 5. Qudit System Cooling

```

Input:    $\rho_{lst} = \{\rho_1, \rho_2, \rho_3, \dots, \rho_i\}$   $\triangleright$  list of density matrices  $\rho_i \in \mathbb{C}^{d \times d}$ , where  $i = \overline{1, N}$ .
            $N \in \mathbb{N}$   $\triangleright$  number of qudits in the system.
            $T \in \mathbb{R}$   $\triangleright$  initial temperature of the system.
1.      For each density matrix  $\rho_i \in \rho_{lst}$  do:
2.           $\rho_i^{exp} = \exp(\frac{\rho_i}{T})$   $\triangleright$  scale the density matrix  $\rho_i$ .
3.           $\rho_i \leftarrow \frac{\rho_i^{exp}}{\text{Tr}(\rho_i^{exp})}$   $\triangleright$  normalize the density matrix  $\rho_i^{exp}$ .
4.      End loop.
5.       $T \leftarrow T \cdot \gamma^{0.5}$   $\triangleright$  reduce the system temperature.
6.      Return  $\rho_{lst}, T$   $\triangleright$  updated list of density matrices and system temperature.
    
```

The TF-QMOA algorithm runs until at least one of two termination conditions is met. The first condition is satisfied when the current system temperature falls below the minimum allowable temperature, i.e., $T < T_{min}$. The second condition is met when convergence is achieved, based on the average change in objective function values for neighboring solutions.

Convergence assessment involves measuring the difference between the current Pareto front F_{new} and the previous Pareto front F_{old} . If $F_{new} = \emptyset$ or $F_{old} = \emptyset$, convergence assessment is not performed. If $F_{new} \neq \emptyset$ and $F_{old} \neq \emptyset$, then for each pair of solution vectors $(x_2^{new}, f_{x_2^{new}}) \in F_{new}$ and $(x_x^{old}, f_{x_x^{old}}) \in F_{old}$, the following is calculated:

$$\Delta f_{x_2^{new}, x_x^{old}} = |f_{x_2^{new}} - f_{x_x^{old}}|, \tag{33}$$

where $\Delta f_{x_2^{new}, x_x^{old}}$ is the difference in objective function values between the two solution vectors; x_2^{new} is a solution vector from the current Pareto front F_{new} ; x_x^{old} is a solution vector from the previous Pareto front F_{old} ; $f_{x_2^{new}}$ is the objective function value vector for x_2^{new} ; $f_{x_x^{old}}$ is the objective function value vector for x_x^{old} .

Next, the total change is computed as $\Delta f_{total} = \sum \Delta f_{x_2^{new}, x_x^{old}}$ and the number of matches is calculated as $\tilde{h} = |\{(x_2^{new}, x_x^{old}) : |x_2^{new} - x_x^{old}| \leq \aleph\}|$, where $\aleph \in \mathbb{R}^+$ is a small positive number defining the acceptable deviation threshold (by default set to 1×10^{-6}). Convergence is considered achieved if:

$$\frac{\Delta f_{total}}{\tilde{h}} < \mathfrak{U}, \tag{34}$$

where $\Delta f_{total} / \hbar$ is the average change; $\mathfrak{U} \in \mathbb{R}^+$ is a small positive number defining the threshold value (by default set to 1×10^{-3}).

During the execution of the TF-QIMOA algorithm, the number of non-dominated solutions may exceed the population size. To prevent unrestricted growth of the archive of non-dominated solutions and maintain computational efficiency, a truncation procedure based on the density distribution of solutions in the objective function space is applied. The essence of this approach is as follows: first, for each solution, a so-called «crowding distance» is calculated—a measure indicating how close the nearest neighbors are to the given solution [43]. Solutions located in densely populated areas of the Pareto front receive smaller values of this distance, whereas solutions at the edges or in sparse regions receive larger values. After evaluating all distances, the most «crowded» solutions are selected and sequentially removed until the total number of non-dominated solutions equals the original population size. This mechanism not only limits the size of the archive but also preserves solution diversity, including extreme (boundary) points and uniformly distributed internal solutions, which is critically important for comprehensive coverage of the entire Pareto front.

3.4. Adaptation of the Algorithm to a Hybrid Quantum-Classical System

The value of existing quantum-inspired algorithms lies in their practical applicability and accessibility to any industry [44], due to their ability to operate without quantum computing devices and systems. In the future, with the emergence of quantum hardware on the market possessing sufficient power and robustness of qubits against environmental noise [10], it will become possible to seamlessly transfer quantum-inspired algorithms to such hardware [44]. However, research in the implementation of quantum-inspired algorithms based on quantum logic circuits [45–48] indicates the need to adapt these algorithms for quantum-classical computing systems. Therefore, the task of interpreting the TF-QIMOA algorithm as a set of quantum logic circuits integrated with classical algorithms that process solutions and prepare data for quantum encoding becomes increasingly relevant, forming a unified quantum-classical algorithmic system.

Since the qudit is used as the primary logical unit in the TF-QIMOA algorithm, and states of dimensionality $d \geq 3$ cannot be represented using the Bloch sphere, the implementation of a qudit in a quantum logic circuit is proposed to be realized using qubits. The number of qubits required per qudit can be calculated using the formula:

$$n = \lceil \log_2 d \rceil, \tag{35}$$

where n is the number of qubits; d is the number of basis states of the qudit.

Then the qudit state $|j\rangle$ (where $j \in \{0, 1, \dots, d - 1\}$) is encoded into the basis states of n qubits:

$$|j\rangle \mapsto |b_{n-1}b_{n-2} \dots b_0\rangle, \tag{36}$$

where b_k is a classical bit (with $b_k = \lfloor j/2^k \rfloor \bmod 2$); k is the bit position (with $k = \overline{0, n - 1}$).

For example, if a qudit of dimensionality $d = 4$ is required, 2 qubits are needed, encoding the qudit's basis states as follows:

$$|0\rangle_d \equiv |00\rangle, |1\rangle_d \equiv |01\rangle, |2\rangle_d \equiv |10\rangle, |3\rangle_d \equiv |11\rangle, \tag{37}$$

where the subscript d indicates that the state belongs to the qudit.

It is important to note that when d is not a power of two, the number of required qubits is still $n = \lceil \log_2 d \rceil$. However, in this case, some states available in the qubit space

remain unused. For example, for $d = 3$, two qubits are still required, but the state $|11\rangle$ is ignored.

To create a superposition of all possible states in a system of n qubits $|\psi\rangle$, the tensor product of n Hadamard gates is applied:

$$H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle, \tag{38}$$

where H is the Hadamard gate (with $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$); $|j\rangle$ is the qudit state encoded in the basis states of n qubits.

After creating the superposition of states in the system $|\psi\rangle$, it is necessary to retain only those states that are valid according to the number d of qudit basis states. To this end, an ancillary qubit in the state $|0\rangle$ is added to the system $|\psi\rangle$. The system now becomes:

$$|\psi'\rangle = |\psi\rangle \otimes |0\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle \otimes |0\rangle. \tag{39}$$

A *CNOT* gate is applied between the first main qubit and the ancillary qubit in the system $|\psi'\rangle$. This operation transforms the first n states in the system $|\psi'\rangle$ into values corresponding to the basis states of n qubits. In general, this can be expressed as:

$$CNOT|c\rangle|t\rangle = (|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X)|c\rangle|t\rangle = \begin{cases} |c\rangle|t\rangle, & c = 0 \\ |c\rangle X|t\rangle, & c = 1 \end{cases}, \tag{40}$$

where $|c\rangle$ is the control qubit, $|t\rangle$ is the target qubit; $|0\rangle\langle 0|$ and $|1\rangle\langle 1|$ are projectors onto the control qubit's states; I is the identity operator; X is the Pauli- X operator.

The Pauli- X operator is applied to the ancillary qubit, which inverts its state to the opposite one.

The *CNOT* gate is now applied again to the first main qubit and the ancillary qubit, but with the roles reversed: the main qubit acts as the target and the ancillary qubit as the control. This operation sets the state of the ancillary qubit to $|1\rangle$ for all states of the system $|\psi'\rangle$.

Then, the set of valid qudit states is defined as $\mathcal{B} = \{|j\rangle : j = \overline{0, d-1}\}$ and a quantum oracle $U_{\mathcal{B}}$ is applied, which transforms the ancillary qubit into the state $|1\rangle$ if the main qubits are in one of the valid states:

$$U_{\mathcal{B}} : |j\rangle \otimes |0\rangle \rightarrow \begin{cases} |j\rangle \otimes |1\rangle, & |j\rangle \in \mathcal{B} \\ |j\rangle \otimes |0\rangle, & |j\rangle \notin \mathcal{B} \end{cases}. \tag{41}$$

After applying the quantum oracle $U_{\mathcal{B}}$, the resulting state of the system is:

$$|\psi''\rangle = \frac{1}{\sqrt{2^n}} (\sum_{|j\rangle \in \mathcal{B}} |j\rangle \otimes |1\rangle + \sum_{|j\rangle \notin \mathcal{B}} |j\rangle \otimes |0\rangle). \tag{42}$$

Next, the ancillary qubit is measured. If its state is $|1\rangle$, the main qubits collapse into the subspace of valid states [49]:

$$|\psi_s\rangle = \frac{1}{\sqrt{|\mathcal{B}|}} \sum_{|j\rangle \in \mathcal{B}} |j\rangle, \tag{43}$$

where $|\mathcal{B}|$ is the number of valid states.

The complete process of qudit initialization via qubits on a quantum logic circuit is illustrated in Figure 2.

Taking (43) into account, the overall state of the qudit system consisting of N qudits is given by the tensor product of the states of all individual qudits:

$$\begin{aligned} |\psi_s^{(N)}\rangle &= \frac{1}{\sqrt{|\mathcal{B}|^N}} \sum_{(|j_1\rangle, |j_2\rangle, \dots, |j_N\rangle) \in \mathcal{B}} |j_1\rangle \otimes |j_2\rangle \otimes \dots \otimes |j_N\rangle \\ &= \frac{1}{\sqrt{|\mathcal{B}|^N}} \sum_{(|j_1\rangle, |j_2\rangle, \dots, |j_N\rangle) \in \mathcal{B}} |j_1, j_2, \dots, j_N\rangle. \end{aligned} \tag{44}$$

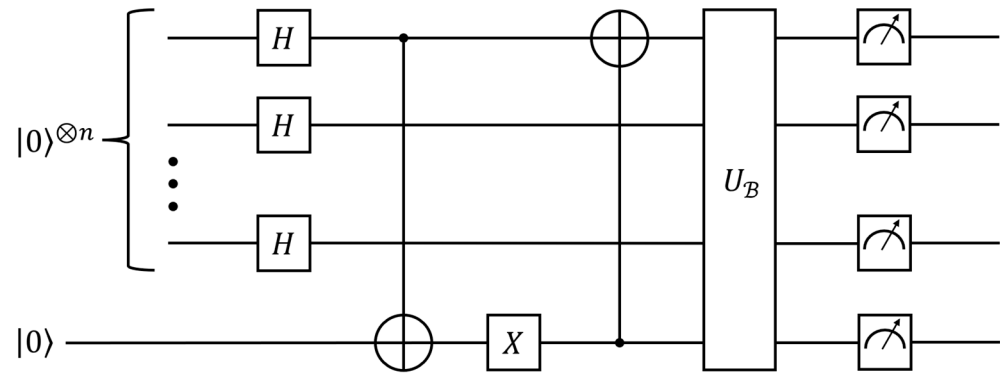


Figure 2. Quantum logic circuit for the qudit initialization process.

The encoding of objective function values $\mathbf{f}_{\mathbf{x}_2}$ can be achieved through phase shift. Let $\mathbf{f}_{\mathbf{x}_2} = [f_1(\mathbf{x}_2), f_2(\mathbf{x}_2), \dots, f_h(\mathbf{x}_2)]$, $\mathbf{x}_2 = (x_{2,1}, x_{2,2}, \dots, x_{2,n})$, n is the number of variables, h is the number of objectives. For each state $|j_1, j_2, \dots, j_N\rangle_{\bar{\delta}}$ ($\bar{\delta} = 1, \dots, |\mathcal{B}|^N$) in the superposition $|\psi_s^{(N)}\rangle$ the objective function values $\mathbf{f}_{\mathbf{x}_2}$ are calculated based on the given constraints and the variable values assigned in \mathbf{x}_2 , with the values in \mathbf{x}_2 varying for each state $|j_1, j_2, \dots, j_N\rangle_{\bar{\delta}}$. Then, the phase encoding of the values for all objective functions in $\mathbf{f}_{\mathbf{x}_2}$ for the state $|j_1, j_2, \dots, j_N\rangle_{\bar{\delta}}$ can be written as:

$$\phi_{\bar{\delta}} = \sum_{l=1}^h o_l f_l(x_{2,1}, x_{2,2}, \dots, x_{2,n}) \tag{45}$$

where o_l is the weight coefficient for the l -th objective function, $o_l \in \mathbb{R}$.

The encoding is performed using the RZ gate, which is applied to each qubit of the state $|j_1, j_2, \dots, j_N\rangle_{\bar{\delta}}$ as follows:

$$\begin{aligned} RZ(\phi_{\bar{\delta}}) |j_1, j_2, \dots, j_N\rangle_{\bar{\delta}} &= e^{-i\phi_{\bar{\delta}}/2} |j_1, j_2, \dots, j_N\rangle_{\bar{\delta}} \\ &= e^{-i\phi_{\bar{\delta}}/2} (|j_1\rangle_{\bar{\delta}} \otimes |j_2\rangle_{\bar{\delta}} \otimes \dots \otimes |j_N\rangle_{\bar{\delta}}) \\ &= e^{-i\phi_{\bar{\delta}}/2} \left((|q\rangle^{\otimes n})_1 \otimes (|q\rangle^{\otimes n})_2 \otimes \dots \otimes (|q\rangle^{\otimes n})_N \right) \\ &= \left(e^{-i\phi_{\bar{\delta}}/2} |q_1\rangle \otimes e^{-i\phi_{\bar{\delta}}/2} |q_2\rangle \otimes \dots \otimes e^{-i\phi_{\bar{\delta}}/2} |q_n\rangle \right)_1 \\ &\quad \otimes \left(e^{-i\phi_{\bar{\delta}}/2} |q_1\rangle \otimes e^{-i\phi_{\bar{\delta}}/2} |q_2\rangle \otimes \dots \otimes e^{-i\phi_{\bar{\delta}}/2} |q_n\rangle \right)_2 \\ &\quad \otimes \left(e^{-i\phi_{\bar{\delta}}/2} |q_1\rangle \otimes e^{-i\phi_{\bar{\delta}}/2} |q_2\rangle \otimes \dots \otimes e^{-i\phi_{\bar{\delta}}/2} |q_n\rangle \right)_N, \end{aligned} \tag{46}$$

where $|q\rangle$ is the state of the qubit.

The state of the qudit system after phase encoding can be written as follows:

$$|\psi_f^{(N)}\rangle = \frac{1}{\sqrt{|\mathcal{B}|^N}} \sum_{(|j_1\rangle, |j_2\rangle, \dots, |j_N\rangle) \in \mathcal{B}} \left(e^{-i\phi/2} |j_1, j_2, \dots, j_N\rangle \right). \tag{47}$$

Next, the quantum oracle U_P acts on the system $|\psi_f^{(N)}\rangle$, flipping the phase of the state $|j_1, j_2, \dots, j_N\rangle_{\delta}$ if it corresponds to a non-dominated solution, while leaving all other states unchanged. The quantum oracle U_P operates according to the rule:

$$U_P : |j_1, j_2, \dots, j_N\rangle_{\delta} \rightarrow \begin{cases} -|j_1, j_2, \dots, j_N\rangle_{\delta}, & |j_1, j_2, \dots, j_N\rangle_{\delta} \Rightarrow \mathcal{L} \\ |j_1, j_2, \dots, j_N\rangle_{\delta}, & |j_1, j_2, \dots, j_N\rangle_{\delta} \not\Rightarrow \mathcal{L}' \end{cases} \quad (48)$$

where \mathcal{L} is a non-dominated solution; the operator \Rightarrow denotes the correspondence relation.

The Grover diffusion operator or «inversion about the mean» [50,51], amplifies the amplitudes of «marked» states, thereby enhancing the identified solutions from (48), which are encoded in these «marked» states, among all possible states of the system.

According to the «inversion about the mean» method, the average amplitude $\bar{\alpha}$ of the probability amplitudes of the quantum states in the system $|\psi_f^{(N)}\rangle$ is calculated using the formula:

$$\bar{\alpha} = \frac{1}{|\mathcal{B}|^N} \sum_{\delta=1}^{|\mathcal{B}|^N} \alpha_{\delta}, \quad (49)$$

where α_{δ} is the probability amplitude of the state $|j_1, j_2, \dots, j_N\rangle_{\delta}$.

Each amplitude α_{δ} is transformed under the action of the Grover diffusion operator as follows:

$$\alpha'_{\delta} = 2\bar{\alpha} - \alpha_{\delta}. \quad (50)$$

The transformation of the state amplitudes of the system $|\psi_f^{(N)}\rangle$ by the Grover diffusion operator is implemented on a quantum logic circuit, according to [51], by applying a Hadamard gate, a Pauli-X gate, and a multi-qubit C...CNOT gate to each qubit forming the multi-qudit state $|j_1, j_2, \dots, j_N\rangle_{\delta}$.

The implementation of the process of phase encoding of objective function values and amplitude amplification of states on a quantum logic circuit is shown in Figure 3.

Local optimization of solutions from the Pareto front $F = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \dots, \mathbf{s}_v\}$ is performed by generating a neighborhood of candidate solutions for each solution $\mathbf{s}_b \in F$, $b = \overline{1, v}$, and selecting among the new solutions those that are non-dominated. To this end, the phase of qudits in the system $|\psi_f^{(N)}\rangle$ is updated by sequentially applying the RZ gate to each qubit of the multi-qudit state $|j_1, j_2, \dots, j_N\rangle_{\delta}$. In this case, the new phase ϕ'_{δ} is formed by adding a small perturbation to the original phase ϕ_{δ} from (45) [52,53]:

$$\phi'_{\delta} = \phi_{\delta} + \delta\phi, \quad (51)$$

where $\delta\phi \sim \mathcal{U}(-\epsilon, \epsilon)$ is a random perturbation drawn from a uniform distribution over the interval $[-\epsilon, \epsilon]$, $\epsilon \in \mathbb{R}$ is a parameter controlling the scale of the perturbation.

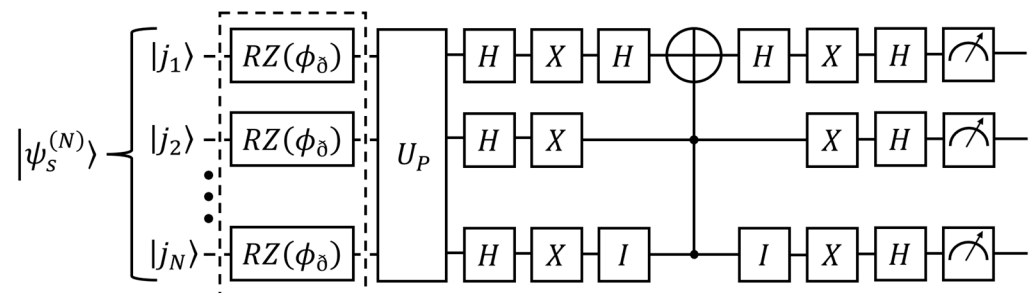


Figure 3. Quantum logic circuit for phase encoding of objective function values and amplitude amplification of states.

Then, the application of the RZ gate to each qubit of the state $|j_1, j_2, \dots, j_N\rangle_{\delta}$ is expressed analogously to (46), with ϕ_{δ} replaced by ϕ'_{δ} .

Then, the quantum oracle U_P is applied again to the system $|\psi_f^{(N)}\rangle$, flipping the phase of the state $|j_1, j_2, \dots, j_N\rangle_{\delta}$ if it corresponds to a non-dominated solution, while all other states remain unchanged. Subsequently, the Grover diffusion operator is applied to amplify the amplitudes of the «marked» states.

The implementation of the process of local optimization of solutions from the Pareto front on a quantum logic circuit is presented in Figure 4.

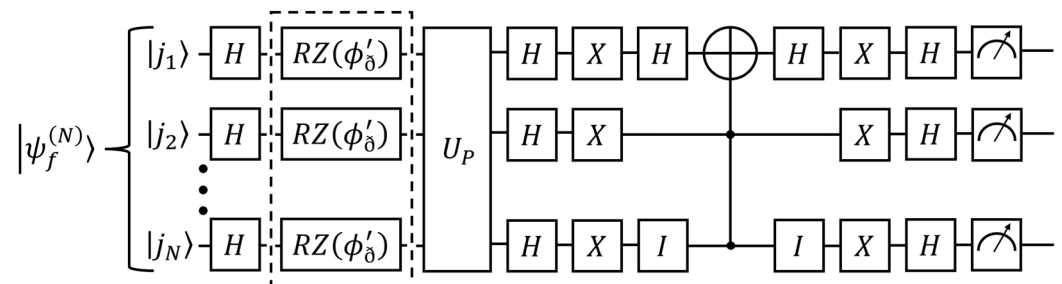


Figure 4. Quantum logic circuit for the local optimization of solutions from the Pareto front.

The overall structure of the quantum-classical algorithmic system implementing the TF-QIMO algorithm is shown in Figure 5.

The quantum-classical algorithmic system is executed in the following order.

Initially, a system of N quantum logic circuits is constructed, each implementing a qudit using n qubits. The set of valid basis states in the qubit space is then defined. Next, the resulting qudit system $|\psi_s^{(N)}\rangle$ is acted upon by RZ gates, which encode the values of the objective functions f_{x_2} , computed on the classical computing device, into the phases ϕ_{δ} (where $\delta = 1, |\mathcal{B}|^N$). Subsequently, on the classical computing device, solutions are checked for non-domination. If a solution is non-dominated, the quantum oracle U_P flips the phase of the corresponding multi-qudit quantum state $|j_1, j_2, \dots, j_N\rangle_{\delta}$, and the Grover diffusion operator amplifies the amplitude of this state. Non-dominated solutions are then stored in the Pareto front F on the classical computing device.

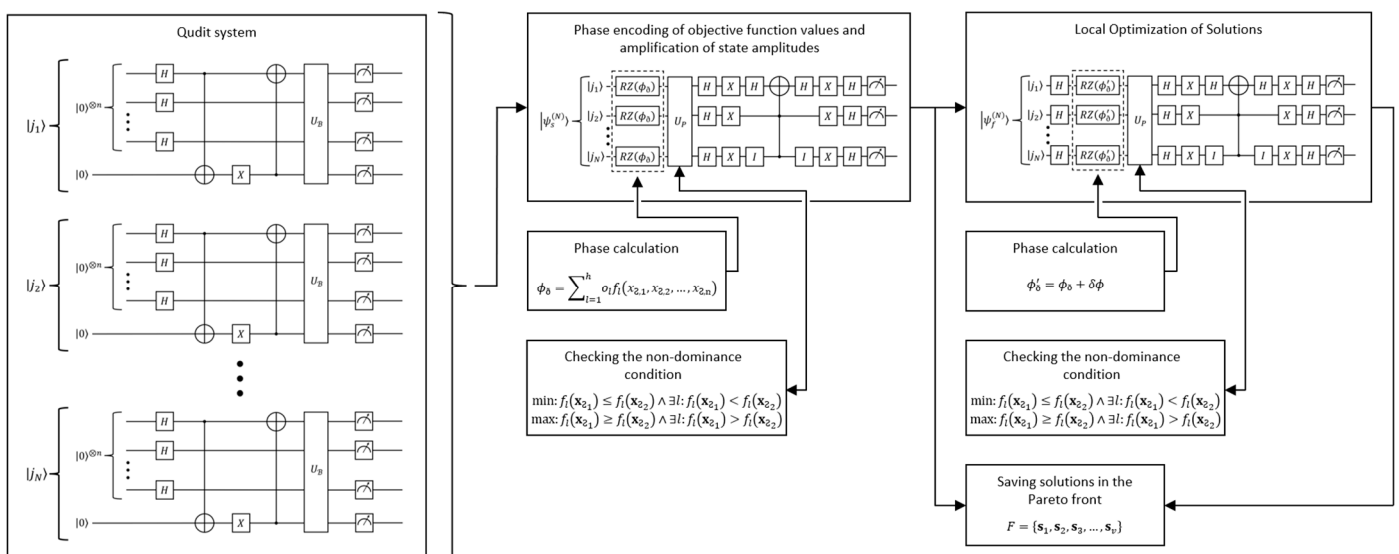


Figure 5. Scheme of the quantum-classical algorithmic system implementing the TF-QIMO algorithm.

In the second stage, Hadamard gates are applied to the resulting system $|\psi_f^{(N)}\rangle$, creating a superposition of all possible states. The application of RZ gates to the sys-

tem $|\psi_f^{(N)}\rangle$ updates the phases ϕ_{δ} to new values ϕ'_{δ} , which are computed for each state $|j_1, j_2, \dots, j_N\rangle_{\delta}$ on the classical computing device. As a result, a neighborhood of candidate solutions is generated around each solution in the Pareto front F . Subsequently, these new solutions are evaluated on the classical computing device to check for non-domination, while the quantum oracle U_P and the Grover diffusion operator identify and amplify each multi-qudit quantum state $|j_1, j_2, \dots, j_N\rangle_{\delta}$ corresponding to a non-dominated solution from the neighborhood. Finally, the new non-dominated solutions are used to update the Pareto front F .

4. Experiments

The performance evaluation of the implemented TF-QMOA algorithm is proposed to be conducted based on the IEEE CEC 2025 benchmark suite [54], specifically using the test problems from the competition on dynamic multiobjective optimization [55].

Multiobjective optimization problems with time-varying properties (Dynamic Multiobjective Optimization Problems, hereinafter—DMOPs) are more complex than static multiobjective problems, which creates significant difficulties for their solution by optimization algorithms [56–58]. This is due to several factors. First, changes in the environment of the optimized system are difficult to detect, and failure to detect such changes may lead to disruption of the Pareto front formation process due to inconsistency of non-dominated solutions for two different optimized environments [59]. Second, the dynamic nature of DMOPs is characterized by irregularity of environmental changes, multimodality, and discreteness of the Pareto front, which greatly complicates the optimization process [60]. Third, the reaction time of optimization algorithms to changes in the environment of the optimized system is an important factor. Time constraints in solving DMOPs require optimization algorithms to maintain a balance between diversity and convergence, enabling them to quickly respond to changes in the environment of the optimized system [61].

The following constrained dynamic multiobjective optimization problems [55] were selected for testing:

1. DMOP T_1

$$\min \begin{cases} f_1(\mathbf{x}) = g(\mathbf{x}) \cdot x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x}) \sqrt{(1 - x_1^2)} \end{cases} \quad (52)$$

subject to the constraint:

$$c_1 = \begin{cases} T_1 \cdot T_2 \leq 0 \\ T_1 = 3 - \cos(\pi t) - \exp(f_1) - 0.3\sin(4\pi f_1) - f_2 \\ T_2 = 4.1 - (1 + f_1 + 0.3f_1^2) - 0.3\sin(4\pi f_1) - f_2 \end{cases} \quad (53)$$

where $g(\mathbf{x}) = 1 + \sum_{i=2}^n x_i^2$; n is the number of variables; $t = \lfloor \frac{\mu}{N \cdot \tau_t} \rfloor / n_t$ where μ is the number of current fitness function evaluations, N is the population size, τ_t is the frequency of environmental changes, n_t is the intensity of environmental changes; $\mathbf{x} \in [0, 1] \times [-1, 1]^{n-1}$.

2. DMOP T_2

$$\min \begin{cases} f_1(\mathbf{x}) = g(\mathbf{x})(x_1 + 0.2\sin(0.5\pi t)\sin(\pi x_1)) \\ f_2(\mathbf{x}) = g(\mathbf{x})(1 - x_1 + 0.2\sin(0.5\pi t)\sin(\pi x_1)) \end{cases} \quad (54)$$

subject to the constraints:

$$\begin{aligned} c_1 &= (f_1 + 2f_2 - 1)(f_1 + 0.5f_2 - 0.5) \geq 0, \\ c_2 &= f_1^2 + f_2^2 - 1.4 \leq 0, \end{aligned} \tag{55}$$

where $g(\mathbf{x}) = 1 + \sum_{i=2}^n ((x_i - \sin(0.5\pi t))^2 + \sin(0.5\pi(x_i - \sin(0.5\pi t))))^2$; n is the number of variables; $t = \lfloor \frac{\mu}{N \cdot \tau_t} \rfloor / n_t$ where μ is the number of current fitness function evaluations, N is the population size, τ_t is the frequency of environmental changes, n_t is the intensity of environmental changes; $\mathbf{x} \in [0, 1] \times [-1, 1]^{n-1}$.

3. DMOP T_3

$$\min \begin{cases} f_1(\mathbf{x}) = g(\mathbf{x}) \cdot x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x})(1 - x_1) \end{cases} \tag{56}$$

subject to the constraints:

$$\begin{aligned} c_1 &= \begin{cases} T_1 \cdot T_2 \geq 0 \\ T_1 = \frac{\sin 0.5\pi t - \cos 0.5\pi t}{\sin 0.5\pi t + \cos 0.5\pi t} \cdot f_1 - f_2 + \frac{-2.2(1 - \cos 0.5\pi t) + 1.3}{\sin 0.5\pi t + \cos 0.5\pi t} \\ T_2 = \frac{\sin 0.5\pi t - \cos 0.5\pi t}{\sin 0.5\pi t + \cos 0.5\pi t} \cdot f_1 - f_2 + \frac{-2.2(1 - \cos 0.5\pi t) + 1.8}{\sin 0.5\pi t + \cos 0.5\pi t} \end{cases} \\ c_2 &= \begin{cases} T_3 \cdot T_4 \geq 0 \\ T_3 = \frac{\sin 0.5\pi t - \cos 0.5\pi t}{\sin 0.5\pi t + \cos 0.5\pi t} \cdot f_1 - f_2 + \frac{-2.2(1 - \cos 0.5\pi t) + 2.6}{\sin 0.5\pi t + \cos 0.5\pi t} \\ T_4 = \frac{\sin 0.5\pi t - \cos 0.5\pi t}{\sin 0.5\pi t + \cos 0.5\pi t} \cdot f_1 - f_2 + \frac{-2.2(1 - \cos 0.5\pi t) + 3.1}{\sin 0.5\pi t + \cos 0.5\pi t} \end{cases} \end{aligned} \tag{57}$$

where $g(\mathbf{x}) = 1 + \sum_{i=2}^n (x_i - \sin(0.5\pi t))^2$; n is the number of variables; $t = \lfloor \frac{\mu}{N \cdot \tau_t} \rfloor / n_t$ where μ is the number of current fitness function evaluations, N is the population size, τ_t is the frequency of environmental changes, n_t is the intensity of environmental changes; $\mathbf{x} \in [0, 1] \times [-1, 1]^{n-1}$.

4. DMOP T_4

$$\min \begin{cases} f_1(\mathbf{x}) = g(\mathbf{x})(x_1 + 0.25\sin(\pi x_1)) \\ f_2(\mathbf{x}) = g(\mathbf{x})(1 - x_1 + 0.25\sin(\pi x_1)) \end{cases} \tag{58}$$

subject to the constraint:

$$\begin{aligned} c_1 &= \left(f_1^2 + f_2^2 - \left(1.3 - 0.45\sin\left(2 \cdot \lfloor 10 \cdot ((t + 1) \bmod 2) - 1 \rfloor \cdot \arctan\left(\frac{f_2}{f_1}\right)^2 \right) \right)^2 \right) \\ &\quad \left(f_1^2 + f_2^2 - \left(1.5 + 0.4\sin\left(4\arctan\left(\frac{f_2}{f_1}\right)^{16} \right)^2 \right) \right) \geq 0, \end{aligned} \tag{59}$$

where $g(\mathbf{x}) = 1 + \sum_{i=2}^n ((x_i - 0.5)^2 - \cos(\pi(x_i - 0.5) + 1))^2$; n is the number of variables; $t = \lfloor \frac{\mu}{N \cdot \tau_t} \rfloor / n_t$ where μ is the number of current fitness function evaluations, N is the population size, τ_t is the frequency of environmental changes, n_t is the intensity of environmental changes; $\mathbf{x} \in [0, 1]^n$.

5. DMOP T_5

$$\min \begin{cases} f_1(\mathbf{x}) = g(\mathbf{x}) \cdot x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x})(1 - x_1) \end{cases} \tag{60}$$

subject to the constraints:

$$\begin{aligned}
 c_1 &= (\sqrt{f_1} + \sqrt{f_2} - 0.95 - 0.5|\sin(0.5\pi t)|)(f_1^{1.5} + f_2^{1.5} - 1.2^{1.5}) \geq 0, \\
 c_2 &= (0.8f_1 + f_2 - (2.5 + 0.08\sin(2\pi(f_2 - f_1)))) \\
 &\left(\left(0.93 + \frac{|\sin(0.5\pi t)|}{3} \right) \cdot f_1 + f_2 - (2.7 + 0.5|\sin(0.5\pi t)| + 0.08\sin(2\pi(f_2 - f_1))) \right) \geq 0,
 \end{aligned}
 \tag{61}$$

where $g(\mathbf{x}) = 1 + \sum_{i=2}^n (|\sin(0.5\pi t)|(x_i - \sin(0.5\pi t))^2 - \cos(\pi(x_i - \sin(0.5\pi t)) + 1)^2)$; n is the number of variables; $t = \lfloor \frac{\mu}{N \cdot \tau_t} \rfloor / n_t$ where μ is the number of current fitness function evaluations, N is the population size, τ_t is the frequency of environmental changes, n_t is the intensity of environmental changes; $\mathbf{x} \in [0, 1] \times [-1, 1]^{n-1}$.

6. DMOP T_6

$$\min \begin{cases} f_1(\mathbf{x}) = g(\mathbf{x}) \cdot x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x}) \sqrt{1 - x_1^2} \end{cases}, \tag{62}$$

subject to the constraint:

$$\begin{aligned}
 c_1 &= 1.1 - \left(\frac{f_1}{0.9 + (0.1 + 0.7|\sin(0.5\pi t)|) \cdot \cos\left(5\arctan\left(\left(\frac{f_2}{f_1}\right)^H\right)^4\right)^6} \right)^2 \\
 &- \left(\frac{f_2}{0.9 + (0.8 - 0.7|\sin(0.5\pi t)|) \cdot \cos\left(5\arctan\left(\left(\frac{f_2}{f_1}\right)^H\right)^4\right)^6} \right)^2 \geq 0,
 \end{aligned}
 \tag{63}$$

where $g(\mathbf{x}) = 1 + \sum_{i=2}^n (x_i - \sin(0.5\pi t))^2$; n is the number of variables; $t = \lfloor \frac{\mu}{N \cdot \tau_t} \rfloor / n_t$ where μ is the number of current fitness function evaluations, N is the population size, τ_t is the frequency of environmental changes, n_t is the intensity of environmental changes; $H = \begin{cases} 1, & \sin(0.5\pi t) \geq 0 \\ -1, & \text{else} \end{cases}$; $\mathbf{x} \in [0, 1] \times [-1, 1]^{n-1}$.

7. DMOP T_7

$$\min \begin{cases} f_1(\mathbf{x}) = g(\mathbf{x}) \cdot x_1 + |\sin(0.5\pi t)| \\ f_2(\mathbf{x}) = g(\mathbf{x})(1 - x_1) + |\sin(0.5\pi t)| \end{cases}, \tag{64}$$

subject to the constraint:

$$\begin{aligned}
 c_1 &= \cos(-0.15\pi) \cdot f_2 - \sin(-0.15\pi) \cdot f_1 \\
 &\geq (2\sin(5\pi(\sin(-0.15\pi) \cdot f_2 + \cos(-0.15\pi) \cdot f_1)))^6,
 \end{aligned}
 \tag{65}$$

where $g(\mathbf{x}) = 1 + \sum_{i=2}^n (x_i - |\sin(0.5\pi t)|)^2$; n is the number of variables; $t = \lfloor \frac{\mu}{N \cdot \tau_t} \rfloor / n_t$ where μ is the number of current fitness function evaluations, N is the population size, τ_t is the frequency of environmental changes, n_t is the intensity of environmental changes; $\mathbf{x} \in [0, 1]^n$.

8. DMOP T_8

$$\min \begin{cases} f_1(\mathbf{x}) = g(\mathbf{x})(x_1 + 0.25\sin(0.5\pi t)\sin(\pi x_1)) \\ f_2(\mathbf{x}) = g(\mathbf{x})(1 - x_1 + 0.25\sin(0.5\pi t)\sin(\pi x_1)) \end{cases}, \tag{66}$$

subject to the constraints:

$$\begin{aligned} c_1 &= (4f_1 + f_2 - 1)(0.3f_1 + f_2 - 0.3) \geq 0, \\ c_2 &= \left(1.85 - f_1 - f_2 - (0.3\sin(3\pi(f_2 - f_1)))^2\right)(f_1 + f_2 - 1.3) \leq 0, \end{aligned} \tag{67}$$

where $g(\mathbf{x}) = 1 + \sum_{i=2}^n \left((x_i - \sin(0.5\pi t))^2 + \sin(0.5\pi(x_i - \sin(0.5\pi t)))^2 \right)$; n is the number of variables; $t = \lfloor \frac{\mu}{N \cdot \tau_t} \rfloor / n_t$ where μ is the number of current fitness function evaluations, N is the population size, τ_t is the frequency of environmental changes, n_t is the intensity of environmental changes; $\mathbf{x} \in [0, 1] \times [-1, 1]^{n-1}$.

9. DMOP T_9

$$\min \begin{cases} f_1(\mathbf{x}) = g(\mathbf{x}) \cdot x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x})(1 - x_1) \end{cases} \tag{68}$$

subject to the constraints:

$$\begin{aligned} c_1 &= ((0.2 + 0.5|\sin(0.5\pi t)|) \cdot f_1^2 + f_2 - 2)(0.7f_1^2 + f_2 - 2.5) \geq 0, \\ c_2 &= f_1^2 + f_2^2 - (0.6 + 0.5|\sin(0.5\pi t)|)^2 \geq 0, \end{aligned} \tag{69}$$

where $g(\mathbf{x}) = 1 + \sum_{i=2}^n |x_i - 0.5\sin(2\pi x_1)|$; n is the number of variables; $t = \lfloor \frac{\mu}{N \cdot \tau_t} \rfloor / n_t$ where μ is the number of current fitness function evaluations, N is the population size, τ_t is the frequency of environmental changes, n_t is the intensity of environmental changes; $\mathbf{x} \in [0, 1] \times [-1, 1]^{n-1}$.

10. DMOP T_{10}

$$\min \begin{cases} f_1(\mathbf{x}) = g(\mathbf{x})(x_1 + 0.02\sin((10 - \lfloor 10\sin(0.5\pi t) \rfloor)\pi x_1)) \\ f_2(\mathbf{x}) = g(\mathbf{x})(1 - x_1 + 0.02\sin((10 - \lfloor 10\sin(0.5\pi t) \rfloor)\pi x_1)) \end{cases} \tag{70}$$

subject to the constraint:

$$c_1 = f_1 + f_2 - \sin(0.5\pi t) - \sin(5\pi(f_1 - f_2 + 1))^2 \geq 0, \tag{71}$$

where $g(\mathbf{x}) = 1 + \sum_{i=2}^n (x_i - \sin(0.5\pi t))^2$; n is the number of variables; $t = \lfloor \frac{\mu}{N \cdot \tau_t} \rfloor / n_t$ where μ is the number of current fitness function evaluations, N is the population size, τ_t is the frequency of environmental changes, n_t is the intensity of environmental changes; $\mathbf{x} \in [0, 1] \times [-1, 1]^{n-1}$.

For the purpose of conducting a comparative analysis of the performance of the quantum-inspired algorithm TF-QIMOA, the following algorithms also participate in the testing: the classical multiobjective optimization algorithms NSGA-III, MOEA/D, GDE3 [62–64], the quantum-inspired multiobjective optimization algorithm QI-NSGA-III [65], the dynamic multiobjective evolutionary algorithms DMOEA, MOACO-DCE, MOEA/D-MDDM [30–34], and the hybrid quantum-classical version of the TF-QIMOA algorithm implemented as a quantum-classical algorithmic system (hereinafter—Hybrid TF-QIMOA).

The test results of the optimization algorithms are compared based on the average values of the solution distribution measure (hereinafter—DM), the Pareto dominance indicator (hereinafter—NR) [65], the mutual dominance rate (hereinafter—MDR), the normalized deviation measure (hereinafter—E), the radial coverage metric (hereinafter— Ψ) [66], and the execution time t .

For each optimization algorithm, the population size, the maximum number of generations and the number of variables are set to 100. The frequency of environmental changes τ_t for the test problems is set to 20. The number of sectors for calculating the values of the metric Ψ is equal to 12. The optimization algorithms are executed iteratively over 30 independent runs.

All experiments are conducted on a classical computing device equipped with an Intel Core i9-13980HX processor and 32 GB of RAM. In the Hybrid TF-QIMOA algorithm, the computation of objective function values, the calculation of quantum state phases, the check for solution non-domination, and the storage and update of solutions in the Pareto front are performed on the classical computing device, while all quantum logic circuits are executed on a quantum computing device based on eight superconducting qubits [67].

The results of testing the optimization algorithms on DMOP problems T_1 – T_{10} with environmental change intensities $n_t = 5$, $n_t = 10$ and $n_t = 20$ are presented in Tables 1–3, respectively. The best metric values are highlighted in bold.

Bar charts of the optimization algorithm test results from Tables 1–3 are presented in Figures 6–8. The x -axis represents the performance metrics used to evaluate the optimization algorithms and the y -axis represents the average values of these metrics.

Based on the test results presented in Table 1, it can be seen that the TF-QIMOA algorithm achieves the best values in terms of the DM metric in problems T_1 , T_5 , T_9 , in terms of the NR metric in problems T_3 – T_6 and T_8 – T_{10} , in terms of the MDR metric in problems T_1 , T_2 , T_4 , T_6 , T_9 , T_{10} , in terms of the E metric in problem T_9 , in terms of the Ψ metric in problems T_1 , T_3 , T_5 , T_6 , T_9 , T_{10} . The Hybrid TF-QIMOA algorithm achieves the best values in terms of the NR metric in problems T_1 , T_2 , T_7 , T_9 , in terms of the MDR metric in problems T_5 , T_7 , in terms of the Ψ metric in problems T_2 , T_8 , in terms of execution time t in problems T_1 , T_2 , T_4 , T_7 . These observations indicate that, when using TF-QIMOA, solutions on the Pareto front are distributed most uniformly in 30% of experiments, a larger number of unique non-dominated solutions are identified in 70% of experiments, significant progress in solution dominance across generations is observed in 60% of experiments, solutions are closest to the ideal solution in 10% of experiments, and the largest portion of the objective space of non-dominated solutions is covered in 60% of experiments. In the case of Hybrid TF-QIMOA, a larger number of unique non-dominated solutions is identified in 40% of experiments, significant progress in solution dominance across generations is observed in 20% of experiments, the largest portion of the objective space is covered in 20% of experiments, and Hybrid TF-QIMOA outperforms the other algorithms in terms of average execution time t in 40% of experiments. It should also be noted that the GDE3 algorithm achieves the best values in terms of the DM metric in problems T_2 , T_3 , T_7 , T_{10} , thus demonstrating better solution distribution uniformity in 40% of experiments. However, due to its low performance in identifying unique non-dominated solutions and covering only about one-third of the objective space, this result may indicate GDE3's limited ability to effectively explore the overall solution space and discover diverse solutions. At the same time, the MOEA/D algorithm achieves the best values in terms of the E metric in 80% of experiments (problems T_1 – T_4 , T_6 – T_8 , T_{10}), demonstrating the smallest distance between the solutions found by MOEA/D and the ideal solution.

The test results of the optimization algorithms under increased environmental change intensity to $n_t = 10$ (Table 2) show that the TF-QIMOA algorithm outperforms the other algorithms in terms of the DM metric in problems T_1 , T_7 , T_9 (30% of experiments), in terms of the NR metric in problems T_3 – T_{10} (80% of experiments), in terms of the MDR metric in problems T_1 , T_5 – T_9 (60% of experiments), in terms of the E metric in problems T_1 , T_5 , T_7 , T_9 (40% of experiments), in terms of the Ψ metric in problems T_1 – T_5 , T_7 , T_{10} (70% of experiments). Meanwhile, the Hybrid TF-QIMOA algorithm achieves the best values in

terms of the NR metric in problems T_1, T_2, T_{10} (30% of experiments), in terms of the MDR metric in problem T_{10} (10% of experiments), in terms of the Ψ metric in problem T_9 (10% of experiments), and in terms of average execution time t in problems T_1, T_4, T_7 (30% of experiments). The GDE3 algorithm outperforms the other algorithms in terms of the DM metric in problems T_2, T_3, T_{10} (30% of experiments) and in terms of average execution time t in problems T_2, T_3, T_5, T_9 (40% of experiments). However, consistent with the results in Table 1, GDE3 demonstrates poor performance in identifying unique non-dominated solutions and insufficient coverage of the objective space of non-dominated solutions. At the same time, the MOEA/D algorithm shows superior convergence performance according to the E metric in problems $T_2-T_4, T_6, T_8, T_{10}$ (50% of experiments).

Table 1. Results of the optimization algorithms on test problems with $n_t = 5$.

Problem	Metric	Hybrid TF-QIMOA	TF-QIMOA	QI-NSGA-III	NSGA-III	MOEA/D	GDE3	DMOEA	MOACO-DCE	MOEA/D-MDDM
T_1	DM	0.1265	0.1156	0.1370	0.2942	0.1932	0.1400	0.1210	0.1185	0.1170
	NR	0.3409	0.2901	0.2935	0.1266	0.1203	0.1278	0.3157	0.3289	0.3351
	MDR	0.0857	0.0966	0.0615	0.0069	-0.0788	0.0420	0.0728	0.0883	0.0896
	E	0.5503	0.5495	0.6010	0.6881	0.1443	0.5688	0.4811	0.4549	0.1505
	Ψ	0.4188	0.4273	0.4167	0.1889	0.1139	0.3333	0.4052	0.3699	0.3823
	t	0.5758	1.0537	1.4986	1.3672	2.7645	1.4746	0.8586	0.9255	1.7814
T_2	DM	0.1307	0.1254	0.1314	0.4020	0.1462	0.0784	0.1289	0.1231	0.1217
	NR	0.7008	0.5421	0.2487	0.1384	0.1372	0.1362	0.6815	0.6932	0.6987
	MDR	0.0651	0.0652	-0.0184	-0.0731	0.0638	0.0137	0.0593	0.0627	0.0641
	E	0.6471	0.4903	0.7723	0.8665	0.1265	0.6218	0.4783	0.4529	0.4364
	Ψ	0.4179	0.4138	0.4167	0.3306	0.0972	0.3333	0.4082	0.4125	0.4168
	t	1.1225	1.5947	2.1014	2.7390	3.3628	1.6782	1.3842	1.4570	1.2690
T_3	DM	0.1323	0.1034	0.1336	0.3214	0.3487	0.0914	0.1078	0.1012	0.0986
	NR	0.1596	0.3129	0.1696	0.1463	0.1301	0.1340	0.2987	0.3054	0.3091
	MDR	0.0010	0.0140	0.0101	-0.0712	0.0594	-0.0050	0.0123	0.0097	0.0131
	E	0.7499	0.6474	0.8524	0.8859	0.3010	0.6300	0.6218	0.5943	0.5789
	Ψ	0.3333	0.3427	0.3278	0.3250	0.1306	0.3333	0.3369	0.3395	0.3418
	t	3.1752	3.7020	3.4482	4.5620	3.8757	2.1301	2.8432	2.9718	2.6854
T_4	DM	0.0713	0.0795	0.0151	0.0802	0.2943	0.1843	0.0217	0.0189	0.0163
	NR	0.1632	0.1832	0.1532	0.1331	0.1445	0.1323	0.1758	0.1794	0.1816
	MDR	0.1180	0.1253	-0.0239	-0.0292	-0.1931	0.0175	0.1089	0.1162	0.1214
	E	0.4279	0.3379	0.5273	0.5280	0.3156	0.6206	0.3287	0.3215	0.3172
	Ψ	0.3667	0.3333	0.4417	0.3333	0.1222	0.3222	0.4291	0.4358	0.4396
	t	1.5145	2.7141	3.6631	4.2374	4.4207	1.7451	2.1386	1.9050	1.9721
T_5	DM	0.0579	0.0434	0.0455	0.1316	0.1553	0.0723	0.0478	0.0449	0.0436
	NR	0.1535	0.1551	0.1359	0.1348	0.1341	0.1290	0.1512	0.1508	0.1537
	MDR	0.0023	0.0020	0.0011	0.0004	0.0000	-0.0035	0.0018	0.0021	0.0021
	E	0.6095	0.6010	0.6042	0.6064	0.6574	0.5737	0.5921	0.5846	0.5793
	Ψ	0.3426	0.3844	0.3411	0.3333	0.2333	0.3333	0.3768	0.3802	0.3829
	t	2.8636	2.1776	3.6563	1.7600	4.0508	2.0266	2.4187	2.5783	2.2075
T_6	DM	0.2748	0.1087	0.3109	0.2966	0.0000	0.2619	0.0892	0.0731	0.0975
	NR	0.1687	0.1956	0.1405	0.1239	0.1332	0.1377	0.1873	0.1912	0.1941
	MDR	0.1335	0.1390	0.1115	-0.0339	0.0727	-0.0625	0.1284	0.1351	0.1378
	E	0.4900	0.4350	0.6545	0.6567	0.0000	0.5650	0.4218	0.4136	0.4072
	Ψ	0.3651	0.3908	0.2944	0.2861	0.0833	0.2944	0.3815	0.3862	0.3894
	t	1.5178	1.6256	1.8971	1.7450	0.8593	1.5565	1.3290	1.4878	1.4032
T_7	DM	0.2115	0.1927	0.2454	0.2548	0.4524	0.0916	0.1834	0.1762	0.1698
	NR	0.1400	0.1364	0.1318	0.1273	0.1396	0.1369	0.1381	0.1392	0.1397
	MDR	0.0468	0.0466	-0.0212	-0.0098	-0.0879	-0.0353	0.0452	0.0461	0.0455
	E	0.5919	0.5784	0.6308	0.6363	0.4894	0.6261	0.5673	0.5598	0.5531
	Ψ	0.3761	0.3815	0.4222	0.3306	0.1444	0.3333	0.4109	0.4172	0.4201
	t	0.5426	0.5925	0.5794	0.5709	1.0248	1.2870	0.6844	0.6387	0.5931
T_8	DM	0.3013	0.3073	0.3833	0.2700	0.0255	0.0805	0.0719	0.0587	0.0642
	NR	0.1489	0.1578	0.1346	0.1259	0.1326	0.1332	0.1553	0.1521	0.1569
	MDR	0.0283	0.0285	0.0696	-0.0639	0.0031	-0.0211	0.0584	0.0627	0.0661
	E	0.5853	0.5005	0.6484	0.7214	0.0318	0.6237	0.4758	0.4926	0.4831
	Ψ	0.4175	0.3333	0.4111	0.3250	0.0861	0.3333	0.4089	0.4162	0.4136
	t	2.9663	2.2034	2.9932	3.9012	0.8415	1.4229	1.8093	1.7354	2.0479

Table 1. Cont.

Problem	Metric	Hybrid TF-QMOA	TF-QMOA	QI-NSGA-III	NSGA-III	MOEA/D	GDE3	DMOEA	MOACO-DCE	MOEA/D-MDDM
T_9	DM	0.1061	0.0972	0.1845	0.2440	0.6592	0.1154	0.1028	0.0983	0.0991
	NR	0.1393	0.1393	0.1355	0.1342	0.1361	0.1375	0.1370	0.1390	0.1388
	MDR	0.0805	0.0904	0.0066	0.0167	0.0027	0.0279	0.0841	0.0876	0.0893
	E	0.5195	0.4367	0.6711	0.7066	0.6585	0.5876	0.4480	0.4415	0.4379
	Ψ	0.3611	0.3745	0.3333	0.3083	0.1639	0.3333	0.3692	0.3738	0.3721
	t	4.3541	5.2427	7.6803	7.9773	3.7936	1.2015	3.1423	3.3873	2.9155
T_{10}	DM	0.2895	0.2492	0.3492	0.3856	0.2945	0.0971	0.2318	0.1847	0.1989
	NR	0.1543	0.1616	0.1420	0.1326	0.1379	0.1329	0.1582	0.1600	0.1605
	MDR	0.0136	0.0249	0.0153	-0.0802	-0.0115	0.0070	0.0203	0.0232	0.0246
	E	0.7980	0.7552	0.8590	0.8604	0.3228	0.6302	0.7386	0.7290	0.7215
	Ψ	0.3466	0.3658	0.3306	0.3306	0.1250	0.3333	0.3579	0.3612	0.3634
	t	1.0163	1.0288	0.9834	1.0171	1.0298	1.3320	1.0579	1.1890	1.0328

Table 2. Results of the optimization algorithms on test problems with $n_t = 10$.

Problem	Metric	Hybrid TF-QMOA	TF-QMOA	QI-NSGA-III	NSGA-III	MOEA/D	GDE3	DMOEA	MOACO-DCE	MOEA/D-MDDM
T_1	DM	0.1340	0.1311	0.1412	0.3135	0.4808	0.1352	0.1327	0.1402	0.1491
	NR	0.3645	0.3476	0.3294	0.1253	0.1306	0.1291	0.3528	0.3581	0.3619
	MDR	0.0971	0.0996	0.0477	-0.0517	-0.0077	-0.0184	0.0934	0.0968	0.0992
	E	0.5229	0.4521	0.6038	0.7342	0.4796	0.5808	0.4683	0.4532	0.4597
	Ψ	0.4075	0.4300	0.3658	0.2000	0.1667	0.3306	0.4215	0.4263	0.4289
	t	0.7946	0.9848	1.0729	1.1360	2.6764	1.5293	0.9182	0.9658	0.8363
T_2	DM	0.1374	0.1364	0.1415	0.4355	0.1462	0.0818	0.1295	0.1247	0.1213
	NR	0.7176	0.6056	0.2215	0.1297	0.1372	0.1286	0.6923	0.7041	0.7098
	MDR	0.0466	0.0469	0.0363	-0.0760	0.0638	0.0311	0.0452	0.0467	0.0461
	E	0.5671	0.5053	0.8284	0.9085	0.1265	0.6184	0.4987	0.4892	0.4821
	Ψ	0.4255	0.4325	0.4080	0.3306	0.0972	0.3333	0.4281	0.4308	0.4310
	t	2.7861	2.4136	4.1606	4.2227	2.6906	1.7202	2.2958	2.8137	2.5394
T_3	DM	0.1435	0.1177	0.1559	0.4590	0.3814	0.0893	0.1102	0.1058	0.1021
	NR	0.2225	0.3668	0.2055	0.1401	0.1319	0.1346	0.3519	0.3592	0.3631
	MDR	0.0120	0.0117	0.0105	-0.0167	0.0689	0.0018	0.0108	0.0110	0.0119
	E	0.6981	0.6940	0.8788	0.8816	0.3212	0.6220	0.6817	0.6735	0.6682
	Ψ	0.3382	0.4698	0.3724	0.3278	0.1333	0.3333	0.4581	0.4632	0.4659
	t	2.8770	2.2906	3.6378	3.1879	2.9848	2.1640	2.4188	2.5374	2.3292
T_4	DM	0.0826	0.0763	0.0169	0.0811	0.3311	0.1869	0.0712	0.0653	0.0684
	NR	0.1331	0.1531	0.1445	0.1330	0.1442	0.1374	0.1527	0.1492	0.1518
	MDR	0.0229	0.0214	-0.0111	-0.0186	-0.0135	0.0620	0.0201	0.0214	0.0217
	E	0.5090	0.5280	0.5287	0.5338	0.3873	0.6307	0.5196	0.5124	0.5078
	Ψ	0.3649	0.3688	0.3083	0.3333	0.1306	0.3222	0.3661	0.3630	0.3674
	t	1.6426	2.8727	4.1510	4.9165	4.3832	1.7699	2.0830	2.2416	1.9377
T_5	DM	0.0497	0.0505	0.0441	0.1290	0.1597	0.0766	0.0482	0.0459	0.0473
	NR	0.1326	0.1449	0.1332	0.1340	0.1339	0.1284	0.1421	0.1437	0.1445
	MDR	0.0091	0.0093	0.0036	-0.0066	0.0000	-0.0539	0.0085	0.0091	0.0091
	E	0.5345	0.5064	0.6011	0.6087	0.6697	0.5777	0.5128	0.5083	0.5076
	Ψ	0.3993	0.4172	0.3349	0.3333	0.2361	0.3333	0.4095	0.4159	0.4138
	t	3.0613	3.4427	4.8184	4.8257	3.1969	2.0498	2.8104	2.9533	2.7385
T_6	DM	0.2006	0.1614	0.3542	0.2950	0.0156	0.2855	0.1528	0.1473	0.1431
	NR	0.1419	0.1461	0.1274	0.1246	0.1348	0.1264	0.1438	0.1452	0.1459
	MDR	0.0373	0.0381	0.0268	0.0268	-0.0692	-0.0982	0.0362	0.0375	0.0380
	E	0.4113	0.3982	0.6549	0.7067	0.0245	0.5774	0.3897	0.3831	0.3789
	Ψ	0.3083	0.3838	0.3861	0.2889	0.0889	0.2944	0.3792	0.3845	0.3827
	t	2.1083	2.7657	2.0922	2.5135	0.9061	2.0769	1.8427	1.9671	1.7253
T_7	DM	0.0781	0.0753	0.2633	0.3214	0.4709	0.0884	0.0761	0.0793	0.0772
	NR	0.1939	0.1988	0.1371	0.1364	0.1498	0.1315	0.1978	0.1965	0.1985
	MDR	0.0670	0.0709	-0.0234	-0.0316	-0.1381	-0.0374	0.0687	0.0703	0.0709
	E	0.5133	0.4772	0.7350	0.7343	0.5055	0.6155	0.4839	0.4791	0.4784
	Ψ	0.4178	0.4197	0.3306	0.3333	0.1444	0.3333	0.4165	0.4183	0.4194
	t	0.7703	0.8890	0.8989	0.9321	0.9604	1.3028	0.8489	0.8915	0.8168
T_8	DM	0.3019	0.3147	0.4769	0.3971	0.0264	0.0804	0.2983	0.2917	0.2856
	NR	0.1415	0.1625	0.1321	0.1279	0.1342	0.1327	0.1589	0.1618	0.1607
	MDR	0.0693	0.0726	0.0412	0.0030	-0.0690	-0.0230	0.0671	0.0708	0.0721
	E	0.5330	0.5877	0.8059	0.7714	0.0319	0.6227	0.5684	0.5762	0.5619
	Ψ	0.3194	0.3520	0.4028	0.3139	0.0861	0.3333	0.3951	0.3987	0.4012
	t	2.9001	3.0494	3.2548	3.1722	0.8439	1.4283	2.6148	2.7043	2.5009

Table 2. Cont.

Problem	Metric	Hybrid TF-QMOA	TF-QMOA	QI-NSGA-III	NSGA-III	MOEA/D	GDE3	DMOEA	MOACO-DCE	MOEA/D-MDDM
T_9	DM	0.1226	0.0911	0.2456	0.3006	0.7606	0.1097	0.0962	0.1038	0.0921
	NR	0.1325	0.1343	0.1318	0.1290	0.1324	0.1330	0.1337	0.1341	0.1339
	MDR	0.0096	0.0137	0.0086	0.0083	0.0000	0.0108	0.0132	0.0121	0.0136
	E	0.6278	0.5571	0.6702	0.7217	0.7403	0.5922	0.5684	0.5612	0.5584
	Ψ	0.4147	0.3333	0.3306	0.3333	0.1667	0.3333	0.4079	0.4115	0.4100
	t	6.2769	6.2458	9.6446	13.8408	3.9052	1.1676	5.8162	5.7639	6.6047
T_{10}	DM	0.1565	0.1578	0.3523	0.3722	0.4763	0.0936	0.1489	0.1427	0.1391
	NR	0.1859	0.1859	0.1366	0.1175	0.1434	0.1307	0.1841	0.1823	0.1852
	MDR	0.0795	0.0774	0.0146	0.0036	-0.0670	0.0314	0.0742	0.0783	0.0768
	E	0.7266	0.7082	0.8513	0.8645	0.4985	0.6308	0.7017	0.6942	0.6885
	Ψ	0.3503	0.3908	0.3333	0.3333	0.1444	0.3333	0.3821	0.3864	0.3887
	t	1.2412	1.3231	1.1053	1.1893	0.9054	1.2409	1.1508	1.1397	1.1323

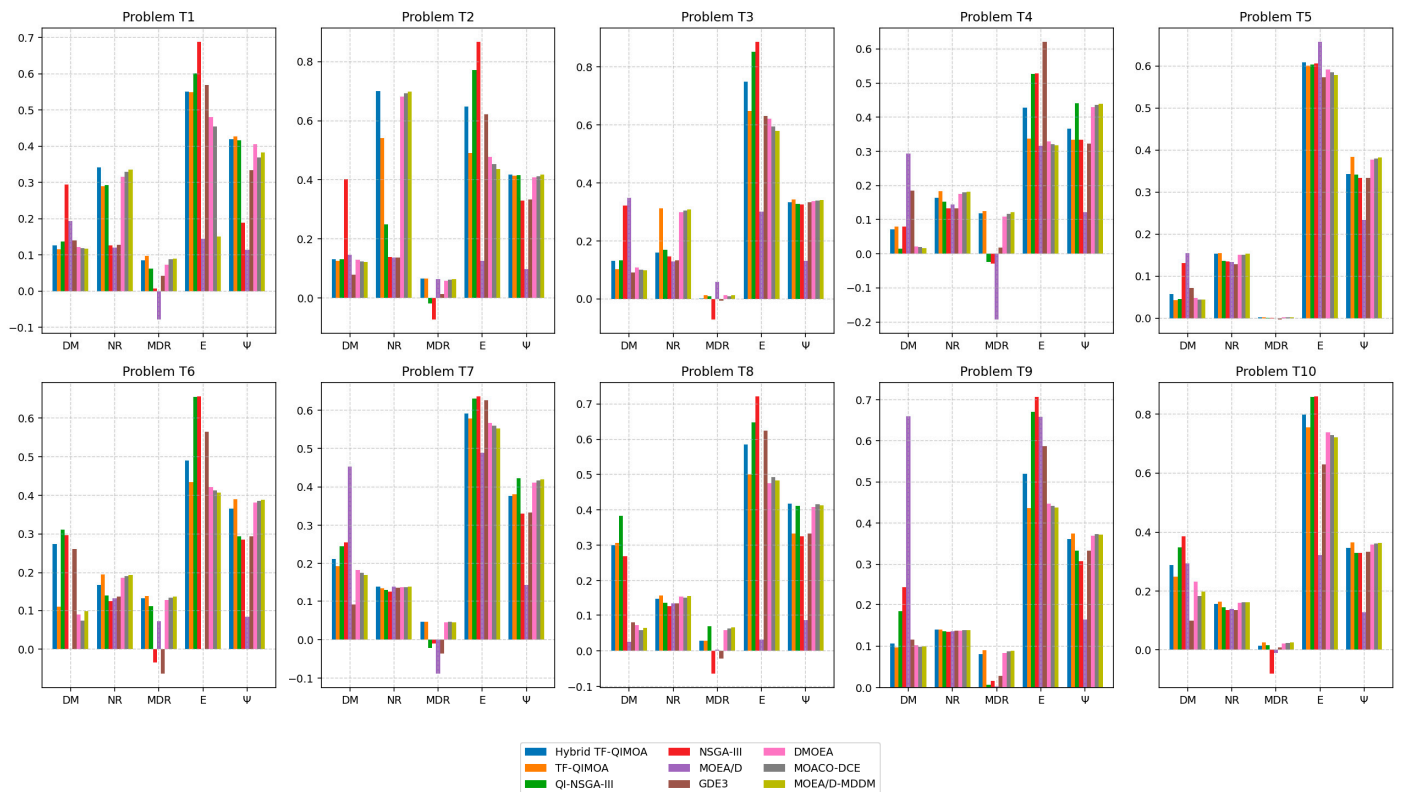


Figure 6. Bar charts of the performance results of the optimization algorithms on test problems with $n_t = 5$.

On the test problems with environmental change intensity $n_t = 20$ (Table 3), the TF-QMOA algorithm outperforms the other algorithms in terms of the DM metric in problems T_1, T_4, T_7 (30% of experiments), in terms of the NR metric in problems T_1-T_5, T_8, T_{10} (70% of experiments), in terms of the MDR metric in problems T_1-T_3, T_8-T_{10} (60% of experiments), in terms of the E metric in problems T_1, T_4, T_9, T_{10} (40% of experiments), in terms of the Ψ metric in problems $T_1-T_5, T_7, T_9, T_{10}$ (80% of experiments). The Hybrid TF-QMOA algorithm achieves the best values in terms of the DM metric in problem T_9 (10% of experiments), in terms of the NR metric in problems T_6, T_7, T_9 (30% of experiments), in terms of the MDR metric in problems T_5, T_7 (20% of experiments), in terms of the E metric in problems T_4, T_5 (20% of experiments), in terms of average execution time t in problems T_1, T_7 (20% of experiments). The GDE3 algorithm outperforms other algorithms in terms of the DM metric in problems T_2, T_3, T_{10} (30% of experiments) and in terms of average execution time t in problems T_2-T_5, T_9, T_{10} (60% of experiments). However, GDE3

continues to exhibit poor performance in identifying unique non-dominated solutions and insufficient coverage of the objective space of non-dominated solutions, which is consistent with its performance on test problems with $n_t = 5$ and $n_t = 10$ (Tables 1 and 2, respectively). Furthermore, in terms of convergence quality (E metric), MOEA/D is dominant, achieving the best metric values in problems T_2, T_3, T_6-T_8 (50% of experiments).

Table 3. Results of the optimization algorithms on test problems with $n_t = 20$.

Problem	Metric	Hybrid TF-QMOEA	TF-QMOEA	QI-NSGA-III	NSGA-III	MOEA/D	GDE3	DMOEA	MOACO-DCE	MOEA/D-MDDM
T_1	DM	0.1275	0.1265	0.1426	0.2954	0.4744	0.1405	0.1283	0.1271	0.1277
	NR	0.2570	0.2906	0.1940	0.1358	0.1236	0.1406	0.2882	0.2781	0.2849
	MDR	0.0708	0.0833	0.0790	-0.0514	0.0292	-0.0426	0.0765	0.0827	0.0812
	E	0.4556	0.4304	0.6187	0.7011	0.4969	0.5822	0.4418	0.4321	0.4362
	Ψ	0.3823	0.3949	0.3111	0.2444	0.1500	0.3306	0.3876	0.3918	0.3935
	t	1.2671	1.2718	1.6706	1.6275	2.6198	1.4976	1.3401	1.3862	1.4249
T_2	DM	0.1354	0.1124	0.1322	0.5758	0.1823	0.0825	0.1097	0.1021	0.1053
	NR	0.5636	0.5863	0.2068	0.1353	0.1395	0.1336	0.5718	0.5792	0.5831
	MDR	0.1309	0.1319	0.0799	-0.0164	0.1118	0.0566	0.1284	0.1307	0.1315
	E	0.5662	0.5650	0.8045	0.8290	0.1400	0.6213	0.5587	0.5521	0.5476
	Ψ	0.3583	0.3783	0.3747	0.3139	0.1000	0.3333	0.3712	0.3765	0.3749
	t	4.4940	3.5316	4.3609	4.6389	2.7184	1.7066	3.2180	3.3877	3.0904
T_3	DM	0.1142	0.1164	0.1251	0.5501	0.2934	0.0864	0.1123	0.1095	0.1072
	NR	0.1989	0.5726	0.2042	0.1325	0.1369	0.1347	0.5584	0.5693	0.5657
	MDR	0.0100	0.0127	0.0094	-0.0147	-0.3488	-0.0378	0.0115	0.0123	0.0126
	E	0.4715	0.4460	0.8701	0.9371	0.2796	0.6245	0.4472	0.4583	0.4517
	Ψ	0.3333	0.4282	0.3458	0.3333	0.1278	0.3333	0.4238	0.4195	0.4259
	t	3.6063	2.4377	2.9930	3.0833	3.3689	2.1483	2.6882	2.5169	2.8931
T_4	DM	0.0429	0.0118	0.0199	0.0819	0.4168	0.2062	0.0173	0.0142	0.0129
	NR	0.1231	0.1531	0.1489	0.1326	0.1485	0.1387	0.1508	0.1472	0.1521
	MDR	0.0412	0.0427	0.0133	-0.0265	0.0752	0.0691	0.0401	0.0418	0.0425
	E	0.4286	0.4286	0.5284	0.5305	0.4506	0.6237	0.4319	0.4293	0.4289
	Ψ	0.3750	0.3997	0.3500	0.3333	0.1472	0.3222	0.3968	0.3915	0.3952
	t	1.8643	3.6375	3.8496	4.1288	3.7026	1.7628	2.7193	3.1087	2.9118
T_5	DM	0.0672	0.0654	0.0445	0.1284	0.1598	0.0712	0.0581	0.0539	0.0512
	NR	0.1357	0.1461	0.1347	0.1323	0.1350	0.1286	0.1428	0.1449	0.1457
	MDR	0.0037	0.0033	0.0035	0.0012	0.0023	-0.0070	0.0030	0.0035	0.0018
	E	0.5192	0.5232	0.6123	0.6228	0.6511	0.5837	0.5267	0.5214	0.5209
	Ψ	0.3972	0.4093	0.3340	0.3333	0.2417	0.3333	0.4073	0.4021	0.4058
	t	5.6343	4.0988	8.4145	9.7277	3.6405	2.0494	5.0373	4.8321	4.6180
T_6	DM	0.2782	0.1893	0.3836	0.3078	0.0000	0.2133	0.1784	0.1721	0.1689
	NR	0.1513	0.1506	0.1157	0.1292	0.1332	0.1351	0.1505	0.1482	0.1497
	MDR	0.0592	0.0664	-0.1153	0.0677	0.0897	0.0170	0.0631	0.0672	0.0658
	E	0.4786	0.4417	0.7440	0.6737	0.0000	0.5677	0.4471	0.4523	0.4432
	Ψ	0.3167	0.3808	0.3917	0.3028	0.0833	0.3111	0.3852	0.3903	0.3887
	t	2.3315	3.2967	3.7468	3.0152	1.1282	1.4364	2.8964	2.7198	2.5857
T_7	DM	0.0607	0.0586	0.2325	0.2748	0.5639	0.0863	0.0595	0.0598	0.0591
	NR	0.1771	0.1758	0.1378	0.1245	0.1242	0.1314	0.1751	0.1732	0.1763
	MDR	0.0076	0.0068	-0.0538	-0.0337	-0.0099	-0.0188	0.0074	0.0069	0.0067
	E	0.6508	0.6768	0.7726	0.7927	0.6183	0.6298	0.6682	0.6521	0.6579
	Ψ	0.3823	0.3908	0.3333	0.3306	0.1583	0.3333	0.3851	0.3897	0.3882
	t	0.9878	1.0150	1.0146	1.5126	1.3191	1.2825	1.0873	1.1427	1.6051
T_8	DM	0.1958	0.1802	0.4126	0.4546	0.0000	0.0803	0.1639	0.1728	0.1675
	NR	0.1852	0.2039	0.1350	0.1282	0.1332	0.1327	0.1981	0.2028	0.2002
	MDR	0.1142	0.1243	0.1121	0.0886	-0.1317	-0.0282	0.1183	0.1221	0.1235
	E	0.7116	0.7047	0.8148	0.8427	0.0000	0.6195	0.6983	0.6924	0.6878
	Ψ	0.3975	0.3983	0.4139	0.3111	0.0833	0.3333	0.4072	0.4121	0.4105
	t	2.0406	2.3876	3.8422	3.4090	0.8626	1.3767	2.7563	2.6198	2.4829
T_9	DM	0.1032	0.1081	0.2091	0.2721	0.7908	0.1045	0.1063	0.1092	0.1073
	NR	0.1631	0.1528	0.1323	0.1309	0.1452	0.1335	0.1582	0.1619	0.1607
	MDR	0.0045	0.0076	0.0023	-0.0046	0.0000	-0.0355	0.0053	0.0052	0.0061
	E	0.5180	0.5054	0.6485	0.6913	0.7369	0.5910	0.5128	0.5083	0.5081
	Ψ	0.3449	0.3687	0.3333	0.3139	0.1639	0.3333	0.3612	0.3663	0.3648
	t	5.5279	7.2567	10.1820	10.4107	4.8219	1.1843	7.0427	6.8125	6.6395

Table 3. Cont.

Problem	Metric	Hybrid TF-QIMOA	TF-QIMOA	QI-NSGA-III	NSGA-III	MOEA/D	GDE3	DMOEA	MOACO-DCE	MOEA/D-MDDM
T_{10}	DM	0.2610	0.1977	0.3162	0.3627	0.5238	0.0982	0.1813	0.1895	0.1841
	NR	0.1337	0.1431	0.1221	0.1297	0.1383	0.1317	0.1402	0.1418	0.1420
	MDR	0.0195	0.0227	0.0157	−0.0073	0.0040	0.0212	0.0208	0.0211	0.0225
	E	0.5476	0.5273	0.8443	0.8753	0.6032	0.6364	0.5381	0.5320	0.5287
	Ψ	0.3641	0.3758	0.3333	0.3278	0.1583	0.3333	0.3695	0.3728	0.3741
	t	1.7916	1.8286	3.2985	3.1486	1.7666	1.2686	1.9180	1.9874	1.8533

It should be noted, however, that the experimental results of the DMOEA, MOACO-DCE, and MOEA/D-MDDM algorithms demonstrate performance patterns closely resembling those of the TF-QIMOA algorithm across all metrics and environmental change intensities. This similarity can be attributed to the fact that all three methods incorporate adaptive mechanisms inspired by dynamic response strategies, such as memory-based archives and decomposition-based guidance, which align conceptually with the quantum-inspired thermonuclear simulation principles underpinning TF-QIMOA. Nevertheless, despite these structural and behavioral parallels, the quantitative results consistently show that DMOEA, MOACO-DCE, and MOEA/D-MDDM remain slightly inferior to TF-QIMOA in terms of the solution distribution measure, the Pareto dominance indicator, the mutual dominance rate, the normalized deviation measure, and the radial coverage metric.

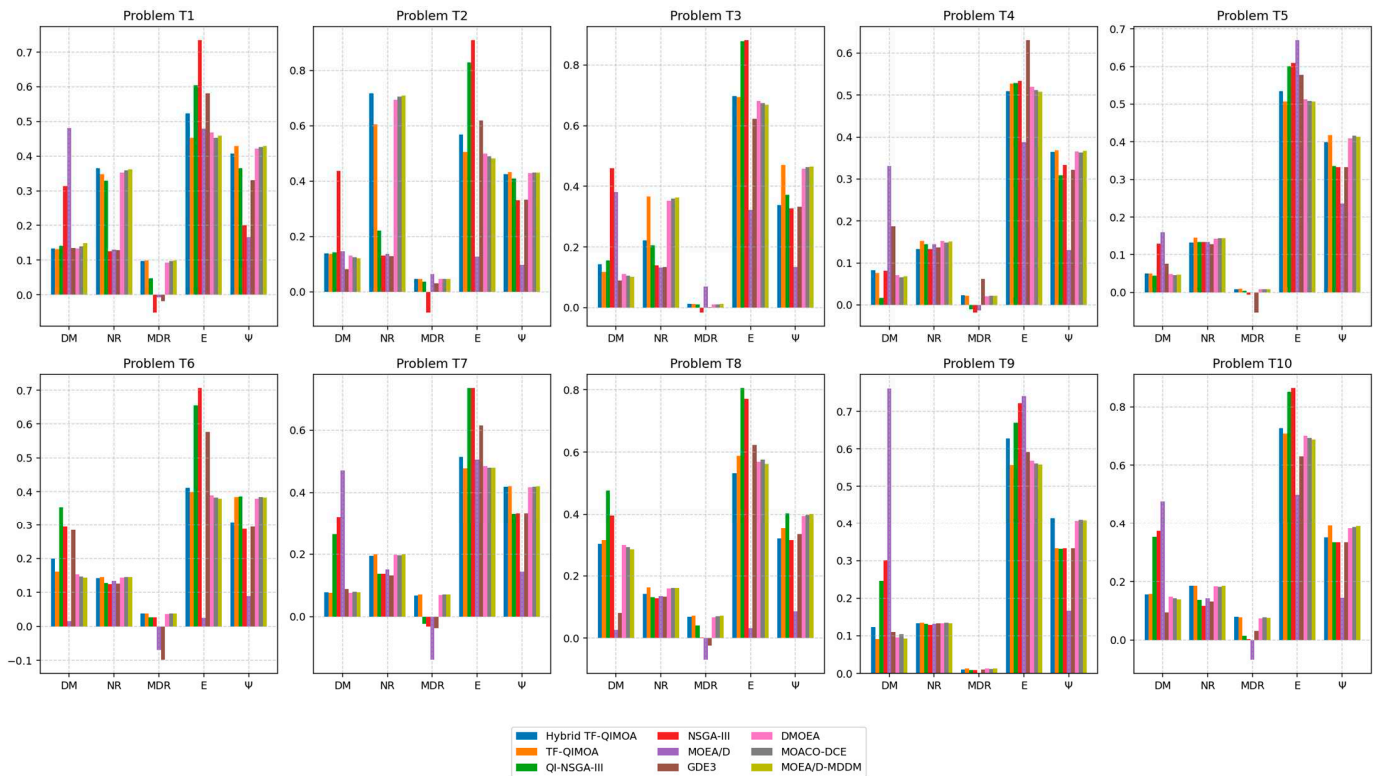


Figure 7. Bar charts of the performance results of the optimization algorithms on test problems with $n_t = 10$.

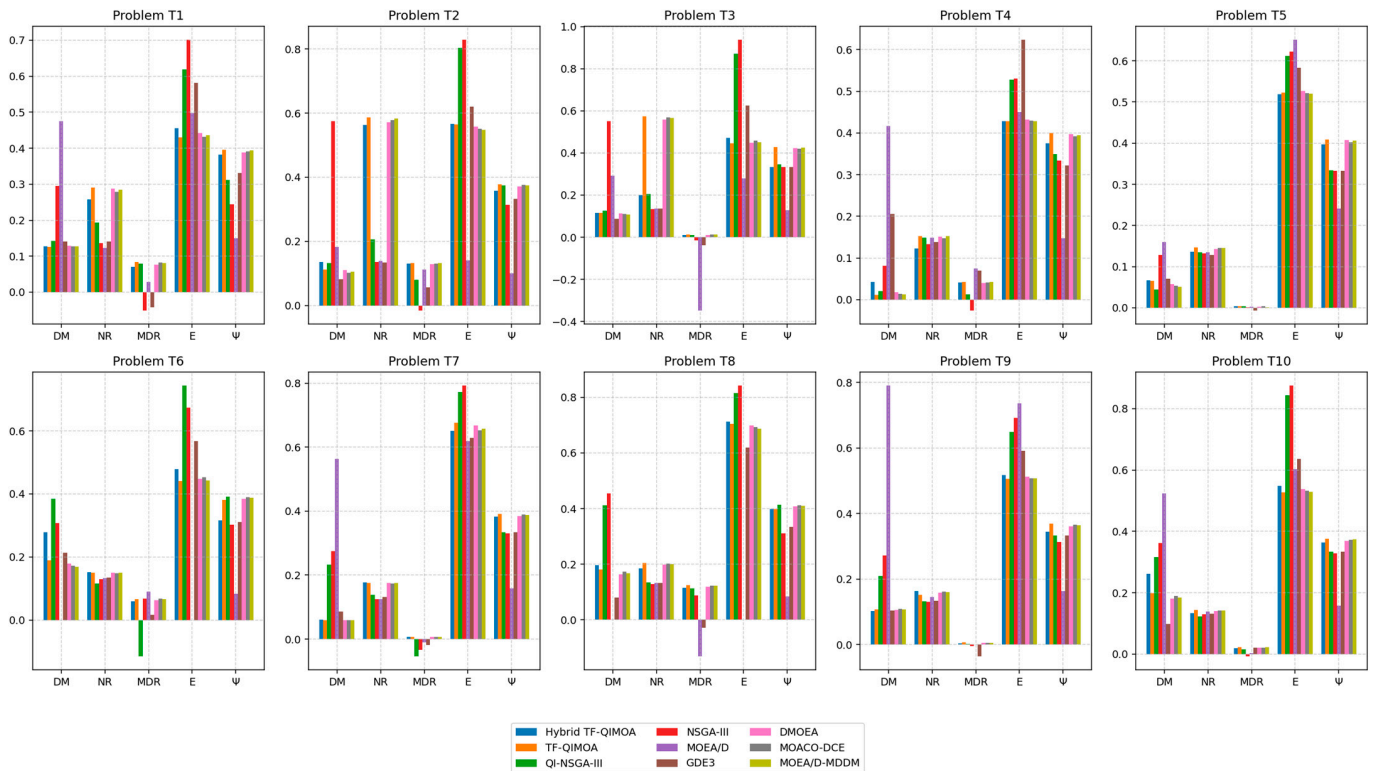


Figure 8. Bar charts of the performance results of the optimization algorithms on test problems with $n_t = 20$.

Tables 4–6 present the results of the Wilcoxon non-parametric statistical test applied to the Pareto dominance indicator distributions for DMOP problems T_1 – T_{10} with environmental change intensity levels $n_t = 5$, $n_t = 10$, and $n_t = 20$, respectively, obtained using the TF-QIMO algorithm, as well as the distributions obtained by the Hybrid TF-QIMO, QI-NSGA-III, NSGA-III, MOEA/D, GDE3, DMOEA, MOACO-DCE, and MOEA/D-MDDM algorithms. The threshold value for p -value is set at 0.05.

The results of the non-parametric Wilcoxon statistical test performed based on the DMOP problems with an environmental change intensity of $n_t = 5$ (Table 4) demonstrate the superiority of the TF-QIMO algorithm over other presented algorithms, except for the QI-NSGA-III algorithm in problem T_1 and the Hybrid TF-QIMO algorithm in problems T_1 , T_2 , and T_7 . In the case of problem T_9 , the TF-QIMO algorithm shows identical performance to the Hybrid TF-QIMO and MOACO-DCE algorithms. When performing the non-parametric Wilcoxon statistical test based on the DMOP problems with an environmental change intensity of $n_t = 10$ (Table 5), the TF-QIMO algorithm demonstrates worse results than the Hybrid TF-QIMO algorithm in problems T_1 and T_2 , and compared to the MOEA/D-MDDM algorithm in problem T_2 . For the DMOEA algorithm, similar effectiveness is observed in problem T_1 . Upon analyzing the results of the non-parametric Wilcoxon statistical test conducted based on the DMOP problems with an environmental change intensity of $n_t = 20$ (Table 6), it is noted that the TF-QIMO algorithm outperforms all other algorithms, except for the Hybrid TF-QIMO algorithm when solving problem T_9 . It should also be noted that the TF-QIMO algorithm shows similar effectiveness to the MOEA/D-MDDM algorithm in problem T_7 and to the DMOEA, MOACO-DCE, and MOEA/D-MDDM algorithms in problem T_9 .

Table 4. Results of the Wilcoxon non-parametric statistical test performed based on the DMOP problems with the intensity of environmental change conditions $n_t = 5$.

Problem	Hybrid TF-QIMOA		QI-NSGA-III		NSGA-III		MOEA/D		GDE3		DMOEA		MOACO-DCE		MOEA/D-MDDM	
	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val
T_1	–	0.0025	–	0.0016	+	0.0000	+	0.0000	+	0.0000	+	0.0001	+	0.0006	+	0.0001
T_2	–	0.0148	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0026	+	0.0014	+	0.0037
T_3	+	0.0047	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0002	+	0.0017
T_4	+	0.0041	+	0.0002	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0005	+	0.0008
T_5	+	0.0003	+	0.0002	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0003
T_6	+	0.0007	+	0.0000	+	0.0000	+	0.0009	+	0.0000	+	0.0008	+	0.0018	+	0.0048
T_7	–	0.0005	+	0.0064	+	0.0000	+	0.0000	+	0.0000	+	0.0005	+	0.0010	+	0.0013
T_8	+	0.0011	+	0.0014	+	0.0001	+	0.0000	+	0.0000	+	0.0043	+	0.0055	+	0.0062
T_9	=	0.0593	+	0.0001	+	0.0000	+	0.0000	+	0.0000	+	0.0106	=	0.0507	+	0.0041
T_{10}	+	0.0039	+	0.0013	+	0.0003	+	0.0000	+	0.0001	+	0.0034	+	0.0052	+	0.0055

Analysis of the experimental results conducted on constrained dynamic multiobjective optimization problems demonstrates the advantages of the TF-QIMOA, Hybrid TF-QIMOA, and GDE3 algorithms, which manifest depending on the characteristics of the optimization problem and the intensity of environmental changes. Despite GDE3’s superiority in terms of average execution time t in 40% of all experiments and in solution distribution uniformity in 33% of all experiments, the algorithm shows unsatisfactory performance in the NR, MDR and Ψ metrics regardless of the intensity of environmental changes. This indicates its inability to fully explore the search space, lack of progress in solution dominance across generations, and poor capability in identifying unique solutions. The TF-QIMOA algorithm demonstrates leadership in the NR metric (73% of all experiments), MDR (60% of all experiments) and Ψ (70% of all experiments), highlighting its superior ability to detect specific non-dominated solutions and cover a larger portion of the objective space under high ($n_t = 5$), moderate ($n_t = 10$) and low ($n_t = 20$) environmental change intensities. It is also worth noting that the Hybrid TF-QIMOA and TF-QIMOA algorithms achieve approximately the same values for the DM, NR, MDR and Ψ metrics in 67% of all experiments. However, Hybrid TF-QIMOA outperforms TF-QIMOA in terms of execution speed in 73% of all experiments.

Table 5. Results of the Wilcoxon non-parametric statistical test performed based on the DMOP problems with the intensity of environmental change conditions $n_t = 10$.

Problem	Hybrid TF-QIMOA		QI-NSGA-III		NSGA-III		MOEA/D		GDE3		DMOEA		MOACO-DCE		MOEA/D-MDDM	
	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val
T_1	–	0.0000	+	0.0001	+	0.0000	+	0.0000	+	0.0000	=	0.0512	+	0.0034	+	0.0007
T_2	–	0.0004	+	0.0000	+	0.0000	+	0.0000	+	0.0002	+	0.0033	+	0.0009	–	0.0012
T_3	+	0.0083	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0065
T_4	+	0.0039	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0027	+	0.0001
T_5	+	0.0001	+	0.0008	+	0.0000	+	0.0004	+	0.0000	+	0.0001	+	0.0009	+	0.0000
T_6	+	0.0001	+	0.0003	+	0.0000	+	0.0000	+	0.0000	+	0.0006	+	0.0056	+	0.0036
T_7	+	0.0000	+	0.0119	+	0.0000	+	0.0000	+	0.0000	+	0.0018	+	0.0004	+	0.0011
T_8	+	0.0005	+	0.0029	+	0.0000	+	0.0000	+	0.0000	+	0.0014	+	0.0106	+	0.0079
T_9	+	0.0097	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0098	+	0.0228	+	0.0030
T_{10}	+	0.0024	+	0.0007	+	0.0000	+	0.0000	+	0.0001	+	0.0010	+	0.0047	+	0.0016

Table 6. Results of the Wilcoxon non-parametric statistical test performed based on the DMOP problems with the intensity of environmental change conditions $n_t = 20$.

Problem	Hybrid TF-QIMOA		QI-NSGA-III		NSGA-III		MOEA/D		GDE3		DMOEA		MOACO-DCE		MOEA/D-MDDM	
	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val	Sign	<i>p</i> -Val
T_1	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0091	+	0.0056	+	0.0013
T_2	+	0.0001	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0025	+	0.0013	+	0.0017
T_3	+	0.0017	+	0.0003	+	0.0000	+	0.0000	+	0.0000	+	0.0001	+	0.0000	+	0.0029
T_4	+	0.0009	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0004	+	0.0016
T_5	+	0.0000	+	0.0006	+	0.0000	+	0.0000	+	0.0000	+	0.0004	+	0.0022	+	0.0018
T_6	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0001	+	0.0076	+	0.0103
T_7	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0014	+	0.0000	=	0.0580
T_8	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0003	+	0.0154	+	0.0135
T_9	−	0.0163	+	0.0001	+	0.0000	+	0.0000	+	0.0000	=	0.0627	=	0.0548	=	0.0594
T_{10}	+	0.0022	+	0.0000	+	0.0000	+	0.0000	+	0.0000	+	0.0019	+	0.0038	+	0.0021

The non-parametric Wilcoxon statistical test yields results that further substantiate these findings by offering a detailed breakdown of algorithmic performance across varying environmental change intensities. Specifically, the TF-QIMOA algorithm consistently exhibits strong performance, particularly excelling in scenarios with moderate and high environmental change intensities, where it demonstrates leadership in key metrics such as NR, MDR, and Ψ . These outcomes align well with the broader analysis indicating the algorithm's robustness in detecting non-dominated solutions and maintaining solution diversity under varying conditions.

However, the statistical tests reveal differences in algorithmic behavior at environmental change intensities $n_t = 20$. Here, while the TF-QIMOA algorithm maintains its competitive edge overall, it occasionally encounters rivals like the Hybrid TF-QIMOA and MOEA/D-MDDM algorithms, especially in problems T_7 and T_9 . This observation underscores the importance of adaptability, as even the most effective algorithms may face limitations in specific problem contexts or under particular environmental dynamics. Moreover, the statistical results highlight the complementary strengths of the Hybrid TF-QIMOA algorithm, which surpasses TF-QIMOA in execution speed in a majority of cases. This suggests that while TF-QIMOA may offer superior solution quality, the Hybrid variant provides a trade-off by optimizing computational efficiency.

5. Discussion

The performance evaluation of the implemented algorithms, TF-QIMOA and Hybrid TF-QIMOA, shows that, regardless of the degree of environmental change intensity in the optimized system, TF-QIMOA consistently achieves excellent results in its ability to identify unique non-dominated solutions, ensure uniform distribution of solutions along the Pareto front, maintain stable improvement in solution quality across generations, and provide high accuracy in approximating the ideal solution. This ensures good coverage of all possible trade-offs between objectives and confirms the algorithm's high adaptability and convergence quality. At the same time, Hybrid TF-QIMOA demonstrates performance comparable to TF-QIMOA in terms of most quality metrics, while outperforming TF-QIMOA in execution time. The first observation is attributed to the structural similarity between the quantum-inspired algorithm TF-QIMOA and the quantum-classical algorithmic system implementing it, whereas the second is explained by the physical properties and computational capabilities of the quantum processing unit. The obtained results demonstrate the high performance of the developed algorithms and indicate their suitability for solving multiobjective optimization problems in both scientific research and practical applications.

Moreover, the algorithms' effectiveness in dynamic multiobjective optimization is particularly noteworthy due to their unique capability to address the complexities of time-varying objectives and constraints, which are frequently encountered in numerous real-world applications. These challenges often arise in fields such as financial portfolio management, where market conditions fluctuate unpredictably, resource allocation in cloud computing systems, where demand shifts dynamically, or environmental monitoring, where external factors like weather patterns change constantly. TF-QIMOA and Hybrid TF-QIMOA stand out not only because they can swiftly adapt to these ever-changing environments but also because they maintain high solution quality throughout this process.

Despite their effectiveness in solving dynamic multiobjective optimization problems, the TF-QIMOA and Hybrid TF-QIMOA algorithms have certain limitations regarding their applicability. In the case of TF-QIMOA, difficulties may arise as the number of objectives increases, potentially requiring additional computational resources for data processing on classical computing devices and systems. Although Hybrid TF-QIMOA demonstrates improved execution time due to the use of quantum logic circuits, it is constrained by the current technical limitations of real quantum computing systems, including quantum devices with limited qubit connectivity and high noise levels [68,69], which may negatively affect result accuracy. Furthermore, it is important to note that both algorithms require careful parameter tuning—such as the number of qudits, the number of qudit levels, the qudit system temperature, and convergence threshold settings—posing challenges when applied to problems with a priori unknown characteristics.

6. Conclusions

This study introduces a novel quantum-inspired algorithm TF-QIMOA, designed for numerical multiobjective optimization by uniquely integrating the high-dimensional state space of qudits with the dynamic, energy-driven principles of controlled thermonuclear fusion. The core contribution lies in demonstrating that the conceptual fusion of quantum information science and plasma physics can yield a powerful and coherent metaheuristic, capable of efficiently navigating complex, high-dimensional Pareto fronts where classical and existing quantum-inspired methods often struggle.

Experimental evaluation on dynamic multiobjective benchmark problems provides compelling evidence of the algorithm's practical significance. TF-QIMOA consistently outperforms leading classical and quantum-inspired baselines across key performance metrics: it identifies a greater number of unique non-dominated solutions, ensures superior uniformity along the Pareto front, maintains stable improvement in solution quality across generations and achieves higher accuracy in approximating the ideal solution. By delivering performance parity with classical execution while being architected for quantum readiness, the hybrid implementation demonstrates a flexible, future-adaptive design capable of evolving alongside quantum hardware advancements.

The theoretical value of this work is found in its successful translation of abstract physical phenomena—qudit superposition, particle interaction, energy release and plasma cooling—into a structured and effective computational paradigm. This provides a new template for designing metaheuristics that are not merely labeled «quantum-inspired», but are fundamentally grounded in the rich dynamics of physical systems. Practically, TF-QIMOA offers a robust tool for tackling real-world multiobjective problems in fields like logistics, energy systems and engineering design, where the ability to find diverse, high-quality trade-off solutions under dynamic conditions is critical.

Looking ahead, future research will focus on extending the algorithm's capabilities to ultra-high-dimensional problems, particularly those arising in deep learning hyperparameter optimization and large-scale systems engineering. This will involve optimizing the

interaction mechanisms and cooling schedules for extreme-scale scenarios. Another critical direction is enhancing the algorithm's robustness in noisy and uncertain environments—investigating its resilience to stochastic noise in objective function evaluations and dynamically shifting constraint boundaries, which is essential for deployment in real-world non-stationary settings [70], such as financial markets or adaptive manufacturing. Finally, the hybrid version of TF-QIMOA will be ported to real noisy intermediate-scale quantum processors. This will require exploring the impact of hardware-specific constraints—such as limited qubit connectivity, gate fidelity, and coherence times—and developing tailored error mitigation strategies for the fusion-inspired interaction and amplitude amplification subroutines.

Author Contributions: Conceptualization, methodology: L.D. and V.M.; software, resources, testing, V.M.; investigation, formal analysis, visualization: L.D. and V.M.; writing—original draft preparation, V.M.; writing—review and editing, L.D.; supervision, L.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data used in the research has been provided in the manuscript, and any additional information desired will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Grushetskaya, Y.; Sips, M.; Schachtschneider, R.; Saberioon, M.; Mahan, A. HPEXplorer: XAI Method to Explore the Relationship Between Hyperparameters and Model Performance. In Proceedings of the ECML PKDD 2024, Vilnius, Lithuania, 9–13 September 2024.
2. Arrazola, J.; Delgado, A.; Bardhan, B.; Lloyd, S. Quantum-inspired algorithms in practice. *Quantum* **2020**, *4*, 307. [[CrossRef](#)]
3. Abs da Cruz, A.V.; Vellasco, M.M.B.R.; Pacheco, M.A.C. Quantum-inspired evolutionary algorithm for numerical optimization. In Proceedings of the 2006 IEEE International Conference on Evolutionary Computation, Vancouver, BC, Canada, 16–21 July 2006.
4. Hao, T.; Huang, X.; Jia, C.; Peng, C. A quantum-inspired tensor network algorithm for constrained combinatorial optimization problems. *Front. Phys.* **2022**, *10*, 906590. [[CrossRef](#)]
5. Kumar Mandal, A.; Chakraborty, B. Quantum computing and quantum-inspired techniques for feature subset selection: A review. *Knowl. Inf. Syst.* **2024**, *67*, 2019–2061. [[CrossRef](#)]
6. Hakemi, S.; Houshmand, M.; KheirKhah, E.; Hosseini, S. A review of recent advances in quantum-inspired metaheuristics. *Evol. Intell.* **2022**, *17*, 627–642. [[CrossRef](#)]
7. Zhang, G. Quantum-inspired evolutionary algorithms: A survey and empirical study. *J. Heuristics* **2011**, *17*, 303–351. [[CrossRef](#)]
8. Iovane, G. Quantum-Inspired Algorithms and Perspectives for Optimization. *Electronics* **2025**, *14*, 2839. [[CrossRef](#)]
9. Wang, Y.; Hu, Z.; Sanders, B.; Kais, S. Qudits and High-Dimensional Quantum Computing. *Front. Phys.* **2020**, *8*, 589504. [[CrossRef](#)]
10. Maslennikov, V.V.; Demidova, L.A. Modification of a Quantum-inspired Genetic Algorithm for Numerical Optimization Using Qudit under Conditions of Simulating Quantum Decoherence. *Comput. Nanotechnol.* **2024**, *11*, 58–85. [[CrossRef](#)]
11. Han, K.; Kim, J. Genetic quantum algorithm and its application to combinatorial optimization problem. In Proceedings of the 2000 Congress on Evolutionary Computation, La Jolla, CA, USA, 16–19 July 2000; Volume 2, pp. 1354–1360.
12. Vivek, Y.; Ravi, V.; Krishna, P.R. Quantum-inspired evolutionary algorithms for feature subset selection: A comprehensive survey. *Quantum Inf. Process.* **2025**, *24*, 196. [[CrossRef](#)]
13. Hua, D.; Chang, C.; Liu, S.; Liu, Y.; Ma, D.; Hua, H. Quantum-Inspired Multi-Objective Optimization Framework for Dynamic Wireless Electric Vehicle Charging in Highway Networks Under Stochastic Traffic and Renewable Energy Variability. *World Electr. Veh. J.* **2025**, *16*, 221.
14. Nikandish, R.; O'Dwyer Boyle, A. A Hybrid Quantum-Classical Generative Adversarial Network for Near-Term Quantum Processors. *IEEE Access* **2024**, *12*, 2169–3536.
15. Pathak, S.; Mani, A.; Sharma, M.; Chatterjee, A. Decomposition Based Quantum Inspired Salp Swarm Algorithm for Multiobjective Optimization. *IEEE Access* **2022**, *10*, 105421–105436. [[CrossRef](#)]
16. Khan, A. Training Learning Weights of Elman Neural Network Using Salp Swarm Optimization Algorithm. *Procedia Comput. Sci.* **2023**, *225*, 1974–1986. [[CrossRef](#)]

17. De Llano, M.; Salazar, A.; Solís, M. Two-dimensional delta potential wells and condensed-matter physics. *Rev. Mex. Física* **2005**, *51*, 626–632.
18. Sremath Tirumala, S. A Quantum-Inspired Evolutionary Algorithm Using Gaussian Distribution-Based Quantization. *Arab. J. Sci. Eng.* **2017**, *43*, 471–482. [[CrossRef](#)]
19. Nogueira, M.; Cotta, C.; Fernández-Leiva, A. An Analysis of Hall-of-Fame Strategies in Competitive Coevolutionary Algorithms for Self-Learning in RTS Games. *Learn. Intell. Optim.* **2013**, 7997, 174–188.
20. Wang, Y.; Wang, W.; Ahmad, I.; Tag-Eldin, E. Multi-Objective Quantum-Inspired Seagull Optimization Algorithm. *Electronics* **2022**, *11*, 1834. [[CrossRef](#)]
21. Kazimipour, B.; Omidvar, M.N.; Li, X.; Qin, K. A Novel Hybridization of Opposition-based Learning and Cooperative Co-evolutionary for Large-Scale Optimization. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation, Beijing, China, 6–11 July 2014.
22. Zhang, Y.; Zhu, Z.; Ning, W.; Fathollahi-Fard, A.M. An Improved Optimization Algorithm Based on Density Grid for Green Storage Monitoring System. *Sustainability* **2022**, *14*, 10822. [[CrossRef](#)]
23. Pan, H.; Lu, C. Solving the Independent Domination Problem by the Quantum Approximate Optimization Algorithm. *Entropy* **2024**, *26*, 1057. [[CrossRef](#)]
24. Khan, A.A.; Hussain, S.; Chandra, R. A Quantum-Inspired Predator–Prey Algorithm for Real-Parameter Optimization. *Algorithms* **2024**, *17*, 33. [[CrossRef](#)]
25. Jansen, D.; Heightman, T.; Mortimer, L.; Perito, I.; Acín, A. Qudit inspired optimization for graph coloring. *arXiv* **2024**, arXiv:2406.00792. [[CrossRef](#)]
26. Wei, Z.; Huang, C.; Wang, X.; Han, T.; Li, Y. Nuclear Reaction Optimization: A Novel and Powerful Physics-Based Algorithm for Global Optimization. *IEEE Access* **2019**, *7*, 66084–66109. [[CrossRef](#)]
27. Wang, Y.; Ma, Y.; Li, Q. A dynamic multi-objective optimization evolutionary algorithm based on classification of environmental change intensity and collaborative prediction strategy. *J. Supercomput.* **2025**, *81*, 54. [[CrossRef](#)]
28. Hou, Y.; Qin, X.; Han, H.; Wang, J. Multiobjective Ant Colony Optimization Algorithm Based on Dynamic Constraint Evaluation Strategy for Highly Constrained Optimization. *IEEE Trans. Cybern.* **2025**, *55*, 4570–4582. [[CrossRef](#)]
29. Gao, K.; Xu, L. A new prediction-based evolutionary dynamic multiobjective optimization algorithm aided by Pareto optimal solution estimation strategy. *Appl. Soft Comput.* **2024**, *165*, 112022. [[CrossRef](#)]
30. Hodgson, P. The Hydrogen Bomb: A Scientist’s Description. *Blackfriars* **1955**, *36*, 156–160. [[CrossRef](#)]
31. Xu, X.; Qi, X.; Deng, L.; Chen, A.-X.; Wang, H.; Qian, Y. Uncertainty analysis of the nuclear liquid drop model. *Chin. Phys. C* **2025**. [[CrossRef](#)]
32. Lee, S. Thermonuclearizing the plasma focus—Converting plasma focus fusion mechanism from beam-gas target to thermonuclear. *Plasma Phys. Control. Fusion* **2022**, *64*, 082001. [[CrossRef](#)]
33. Godes, A.; Shablov, V. Lawson criterion for different scenarios of using D-3He fuel in fusion reactors. *Nucl. Energy Technol.* **2023**, *9*, 207–214. [[CrossRef](#)]
34. Chen, Z.; Zhao, Z.; Lyu, Z.; Li, J.; Pan, D. Numerical simulation of deuterium-tritium fusion reaction rate in laser plasma based on Monte Carlo-discrete ordinate method. *Acta Phys. Sin.* **2019**, *68*, 215201. [[CrossRef](#)]
35. Meaney, K.; Paris, M.; Hale, G.; Kim, Y.; Kuczek, J.; Hayes, A. Deuterium-tritium fusion γ -ray spectrum at MeV energies with application to reaction-in-flight inertial confinement fusion measurements. *Phys. Rev. C* **2024**, *109*, 034003. [[CrossRef](#)]
36. Fechtal, I. Simulation of Plasma-Projectile Interaction in the Tokamak Thermonuclear Reactor and Plasma Panels (PDP). Ph.D. Thesis, Hassan II University of Casablanca, Casablanca, Morocco, 2024.
37. Williams, F. The Density Matrix and Partition Function in Quantum Statistical Mechanics. *Prog. Math. Phys.* **2003**, *27*, 307–315.
38. Ju, C.; Miranowicz, A.; Barnett, J.; Chen, G.; Nori, F. Heisenberg and Heisenberg-Like Representations via Hilbert Space Bundle Geometry in the Non-Hermitian Regime. *Phys. Rev. A* **2025**, *111*, 052213. [[CrossRef](#)]
39. Dobran, F. Fusion energy conversion in magnetically confined plasma reactors. *Prog. Nucl. Energy* **2012**, *60*, 89–116. [[CrossRef](#)]
40. Fernández, F. *Introduction to Perturbation Theory in Quantum Mechanics*; CRC Press: Boca Raton, FL, USA, 2000.
41. Keebaugh, C.; Marshman, E.; Singh, C. Challenges in sensemaking and reasoning in the context of degenerate perturbation theory in quantum mechanics. *Phys. Rev. Phys. Educ. Res.* **2024**, *20*, 020139. [[CrossRef](#)]
42. Bhattacharjee, J.; Ray, D. Time-dependent perturbation theory in quantum mechanics and the renormalization group. *Am. J. Phys.* **2016**, *84*, 434–442. [[CrossRef](#)]
43. Raquel, C.; Naval, P. An effective use of crowding distance in multiobjective particle swarm optimization. In Proceedings of the GECCO 2005—Genetic and Evolutionary Computation Conference, Washington, DC, USA, 25–29 June 2005; pp. 257–264.
44. Bova, F.; Goldfarb, A.; Melko, R. Commercial applications of quantum computing. *EPJ Quantum Technol.* **2021**, *8*, 2. [[CrossRef](#)]
45. Joe, M.; Parkavi, K. Integration of quantum-inspired algorithms in circuit technologies for enhanced computational efficiency. *ICTACT J. Microelectron.* **2025**, *10*, 1952–1956. [[CrossRef](#)]
46. Brizuela, D.; Uria, S. Hybrid classical-quantum systems in terms of moments. *Phys. Rev. A* **2024**, *109*, 032209. [[CrossRef](#)]

47. Rallis, K.; Liliopoulos, I.; Tsipas, E.; Varsamis, G.; Melissourgios, N.; Karafyllidis, I.; Sirakoulis, G.; Dimitrakis, P. Hardware-level Interfaces for Hybrid Quantum-Classical Computing Systems. *arXiv* **2025**, arXiv:2503.18868.
48. Nikoloska, I.; Simeone, O. Training Hybrid Classical-Quantum Classifiers via Stochastic Variational Optimization. *IEEE Signal Process. Lett.* **2022**, *29*, 977–981. [[CrossRef](#)]
49. Pagni, V.; Huber, S.; Epping, M.; Felderer, M. Fast Quantum Amplitude Encoding of Typical Classical Data. *arXiv* **2025**, arXiv:2503.17113. [[CrossRef](#)]
50. Goscinski, A. The parallel Grover as dynamic system. *arXiv* **2018**, arXiv:1808.03347. [[CrossRef](#)]
51. Al-Bayaty, A.; Perkowski, M. A concept of controlling Grover diffusion operator: A new approach to solve arbitrary Boolean-based problems. *Sci. Rep.* **2024**, *14*, 23570. [[CrossRef](#)]
52. Facchi, P.; Ligabò, M.; Viesti, V. Robustness of quantum symmetries against perturbations. *J. Phys. A Math. Theor.* **2025**, *58*, 125305. [[CrossRef](#)]
53. Melo, A.; Beugnot, G.; Minganti, F. Variational Perturbation Theory in Open Quantum Systems for Efficient Steady State Computation. *arXiv* **2025**, arXiv:2504.00085. [[CrossRef](#)]
54. IEEE CEC 2025 IEEE Congress on Evolutionary Computation. In *IEEE Computational Intelligence Magazine*; IEEE: New York, NY, USA, 2014; Available online: <https://ieeexplore.ieee.org/document/10709777> (accessed on 14 July 2025).
55. Zou, J.; Jiang, S.; Hou, Z.; Yu, X.; Hu, Y.; Yang, S. Dynamic Multiobjective Optimisation. In Proceedings of the 2025 IEEE Congress on Evolutionary Computation (CEC 2025), Hangzhou, China, 8–12 June 2025; Available online: <https://yxz996.github.io> (accessed on 14 July 2025).
56. Jiang, S.; Zou, J.; Yao, X. Evolutionary Dynamic Multi-Objective Optimization: A Survey. *ACM Comput. Surv.* **2022**, *55*, 1–47.
57. Song, W.; Liu, S.; Wang, X.; Guo, Y.; Yang, S.; Jin, Y. Learning to Guide Particle Search for Dynamic Multiobjective Optimization. *IEEE Trans. Cybern.* **2024**, *54*, 5529–5542. [[CrossRef](#)]
58. Jiang, X.; Chen, Q.; Ding, J.L.; Zhang, X. Dual-Population Evolution Based Dynamic Constrained Multiobjective Optimization with Discontinuous and Irregular Feasible Regions. *IEEE Trans. Emerg. Top. Comput. Intell.* **2025**, *9*, 1352–1366. [[CrossRef](#)]
59. Yu, K.; Zhang, D.; Liang, J.; Chen, K.; Yue, C.; Qiao, K.; Wang, L. A Correlation-Guided Layered Prediction Approach for Evolutionary Dynamic Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2022**, *27*, 1398–1412. [[CrossRef](#)]
60. Hu, Y.; Zheng, J.; Jiang, S.; Zou, J. Handling Dynamic Multiobjective Optimization Environments via Layered Prediction and Subspace-Based Diversity Maintenance. *IEEE Trans. Cybern.* **2021**, *53*, 2572–2585. [[CrossRef](#)]
61. Hu, Y.; Ou, J.; Suganthan, P.; Pedrycz, W.; Wang, R.; Zheng, J.; Zou, J.; Song, Y. Dynamic Multi-Objective Optimization Algorithm Guided by Recurrent Neural Network. *IEEE Trans. Evol. Comput.* **2024**, *1*, 1–14. [[CrossRef](#)]
62. Doerr, B.; Wietheger, S. A Mathematical Runtime Analysis of the Non-dominated Sorting Genetic Algorithm III (NSGA-III). In Proceedings of the Genetic and Evolutionary Computation Conference Companion, Melbourne, Australia, 14–18 July 2024.
63. Qi, Y.; Ma, X.; Liu, F.; Jiao, L.; Sun, J.; Wu, J. MOEA/D with adaptive weight adjustment. *Evol. Comput.* **2014**, *22*, 231–264. [[CrossRef](#)] [[PubMed](#)]
64. Kukkonen, S.; Lampinen, J. GDE3: The third evolution step of generalized differential evolution. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2–5 September 2005.
65. Demidova, L.A.; Maslennikov, V.V. Application of multilevel quantum systems for parallel evaluation of solutions in multiobjective optimization problems. *Vestn. Ryazan State Radio Eng. Univ.* **2025**, *92*, 57–76. [[CrossRef](#)]
66. Halim, H.; Ismail, I.; Das, S. Performance assessment of the metaheuristic optimization algorithms: An exhaustive review. *Artif. Intell. Rev.* **2021**, *54*, 2323–2409. [[CrossRef](#)]
67. Quantum Computer Company & Practical Computing Solutions. SpinQ. Available online: <https://www.spinquanta.com/products/quantum-computing-software> (accessed on 26 July 2025).
68. Zheng, M.; Zeng, J.; Yang, W.; Chang, P.; Lu, Q.; Yan, B.; Zhang, H.; Wang, M.; Wei, S.; Long, G. Quantum-classical hybrid algorithm for solving the learning-with-errors problem on NISQ devices. *Commun. Phys.* **2025**, *8*, 208. [[CrossRef](#)]
69. Pasupuleti, M.K. Quantum Code and Connectivity: Next-Generation Algorithms, Cryptography, and Secure Quantum Networks. *Int. J. Acad. Ind. Res. Innov. (IJAIRI)* **2025**, *5*, 95–107. [[CrossRef](#)]
70. Kulikov, G.V.; Dang, X.K.; Lelyukh, A.A. Optimization of signal constellations with amplitude-phase shift keying in communication channels with non-fluctuating interference. *Russ. Technol. J.* **2025**, *13*, 76–88. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.