Research
A SCIENCE PARTNER JOURNAL

RESEARCH ARTICLE

Quantum-Inspired Fast Algorithm and Circuit Realization for Constrained Combinatorial Optimization Problem

Haosen Chen†, Shailan Deng†, Tian Chen*, and Xiangdong Zhang*

Key Laboratory of Advanced Optoelectronic Quantum Architecture and Measurements of Ministry of Education, School of Physics, Beijing Institute of Technology, Beijing 100081, China.

*Address correspondence to: chentian@bit.edu.cn (T.C.); zhangxd@bit.edu.cn (X.Z.)
†These authors contributed equally to this work.

Citation: Chen H, Deng S, Chen T, Zhang X. Quantum-Inspired Fast Algorithm and Circuit Realization for Constrained Combinatorial Optimization Problem. *Research* 2026;9:Article 1345. https://doi.org/10.34133/research.1345

Submitted 24 December 2025
Revised 1 June 2026
Accepted 9 June 2026
Published 29 June 2026

Copyright © 2026 Haosen Chen et al. Exclusive licensee Science and Technology Review Publishing House. No claim to original U.S. Government Works. Distributed under a Creative Commons Attribution License (CC BY 4.0).

Constrained combinatorial optimization (CCO) problems are prevalent across various fields and represent key challenges in computational science and engineering. Although numerous classical and quantum algorithms have been proposed to tackle these problems, substantial limitations still persist. Classical algorithms exhibit exponential computational complexity growth with scale and persistent vulnerability to local minima traps. Quantum computing, while offering theoretical advantages through global superposition, faces practical barriers such as short decoherence times and current hardware limitations. To address these challenges, we propose a quantum-inspired fast algorithm for solving CCO problems. Our approach enhances global search capability with superposition encoding and avoids constraint-induced local minima via a project–feedback strategy. Particularly, our method aligns with mature electronics manufacturing and demonstrates a proof-of-concept implementation in classical systems, indicating high-efficiency potential for solving constrained optimization problems.

Introduction

Constrained combinatorial optimization (CCO) problems involve finding the optimal configuration satisfying given constraints from a discrete solution space. These problems represent core challenges in computational science and engineering optimization [1,2]. With the surging demand for efficient decision-making in fields like logistics scheduling, chip design, and artificial intelligence, the computational efficiency and stability of solving CCO have been a critical bottleneck in contemporary industrial applications. At their core, CCO problems feature extremely high-dimensional discrete landscapes shaped by the interplay between objective functions and constraint-induced feasibility regions, making efficient navigation of these landscapes a central challenge for both algorithmic and physical computing paradigms.

To overcome these challenges, numerous classical approaches have been proposed. Classical heuristic algorithms [3–7] [e.g., simulated annealing (SA), ant colony optimization (AC), and genetic algorithm] alleviate computational demands through stochastic search and local optimization strategies. While effective in moderate-scale scenarios, these methods often struggle to balance global exploration with rapid convergence, leading to either premature trapping in local minima or increased computational overhead. In parallel, continuous relaxation-based frameworks improve tractability by embedding discrete decision variables into continuous spaces and optimizing a surrogate objective defined over the relaxed domain [8,9]. Such approaches are widely adopted due to their flexibility and compatibility with gradient-based optimization techniques. In these formulations, the search process is primarily guided by the minimization of a fixed relaxed objective, where the geometry of the surrogate landscape determines the optimization trajectory.

Beyond purely algorithmic approaches, recent years have witnessed the emergence of physics-assisted hardware solvers, such as Ising machines implemented via compute-in-memory architectures [10–12] and photonic or electronic oscillator networks [13–19]. By exploiting the intrinsic dynamics of physical systems to perform large-scale parallel state evolution, these platforms offer a promising pathway to alleviate the memory and power limitations of conventional digital computing. Despite rapid progress, practical implementations remain constrained by device noise, parameter variability, and limited programmability, which can hinder solution stability and scalability, particularly in problems involving complex constraints.

Meanwhile, quantum computing offers theoretical potential for efficiently escaping local optima by leveraging the global superposition inherent in quantum systems [20–27]. For instance, quantum annealing searches for the global optimum through adiabatic evolution and has demonstrated promising efficiency in solving low-dimensional sparse problems [28–31]. Nevertheless, its applicability to large-scale real-world applications remains limited due to short decoherence times and constraints of sparse topology mapping. In response, several quantum-inspired algorithms have been developed in recent years to solve combinatorial optimization problems while avoiding the hardware constraints of quantum devices [11,32–35]. However, a common drawback

among these methods is the lack of effective mechanisms for handling constraints, which often leads to infeasible solutions or substantially degraded optimization quality in tightly constrained scenarios. Consequently, developing a unified framework that can integrate scalable computation, robust optimization dynamics, and effective constraint handling remains an important open challenge in solving CCO problems.

In this paper, we propose a quantum-inspired fast algorithm (QIFA) for solving CCO problems. QIFA formulates the search as a quantum-inspired evolution of analog state variables that emulate superposition-like exploration, while a projection–feedback mechanism is intrinsically combined with the evolution to enforce feasibility throughout the search. In this way, exploration and constraint satisfaction co-evolve, establishing a unified dynamical optimization framework in which feasibility and search dynamics remain tightly coupled. Using the Traveling Salesman Problem (TSP) as a benchmark task, we compare the solution performance of our algorithm against various other methods at different scales, demonstrating its high efficiency and superior solving capability. For the tested TSP instances, the algorithm empirically exhibits an approximately linear relationship between city size and the number of iterations required to reach near-optimal solutions. Furthermore, we design and implement the algorithm in analog circuitry. Experimental validation on a 20-city global TSP benchmark confirms the feasibility and effectiveness of our analog circuitry hardware in real-world optimization scenarios, without suffering from inherent quantum limitations such as decoherence or cryogenic requirements. We also discuss the theoretical solving time achievable with current cutting-edge experimental platform technology. Overall, this work provides a novel design paradigm for developing hardware-friendly and scalable constrained optimization strategies.

Results

Theory of QIFA for CCO problems

We consider CCO problems with first-order constraints and discrete variables, typical examples of which include TSP, Knapsack Problem (KP), and Job-shop Scheduling Problem (JSP). Such problems are characterized by 3 key elements: a set of discrete decision variables x, an objective function $f(x)$, and a set of constraints $\{g_k(x) | k \in \{1, \ldots, K\}\}$, where k indexes the constraints and K denotes their total number. The optimization task is to minimize the objective function $f(x)$ over the variables x while satisfying all constraints $g_k(x) = 0$.

Each constraint acts only on a subset of the decision variables, which gives rise to overlapping and irregular feasible regions in the combinatorial configuration space. The coexistence of a global objective and multiple interacting constraints results in a feasible set that is highly structured and fragmented, constituting the intrinsic difficulty of CCO problems. This framework is schematically illustrated in Fig. 1A. The discrete decision variables are depicted as blue circular nodes, each representing a discrete variable $x_1, \ldots, x_n$. These variables jointly determine the value of the global objective function, shown as a light-colored central region labeled $\min_x f(x)$, which represents the shared optimization goal. The effects of the constraints are illustrated by purple shaded regions that surround and restrict different subsets of variables.

It has been proven that the CCO problems can be mapped to the ground state search of Ising models [36]. As illustrated in Fig. 1B, the discrete decision variables are encoded by spin variables

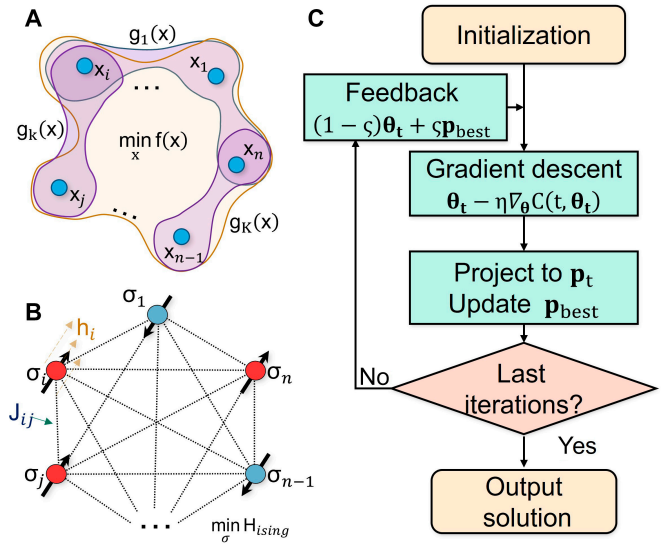


Fig. 1. Framework of QIFA for CCO. (A) A CCO problem consists of a set of discrete decision variables x jointly determining a global objective function $f(x)$ while simultaneously subjecting to multiple constraints $\{g_k(x)\}$. (B) Transformation of the CCO problem into ground state search of Ising model, seeking the spin configurations σ that minimize the Ising Hamiltonian. Parameters from the original problem map to coupling coefficients J_{ij} and external field terms h_i. (C) QIFA's workflow: QIFA operates through an iterative loop consisting of 4 conceptual stages. Starting from an initialization of superposition state variables, the system undergoes gradient-based evolution driven by a time-dependent cost function. The resulting state is then projected onto a feasible discrete configuration, and the historically best solution is updated accordingly. A feedback mechanism guides subsequent evolution toward high-quality solutions, forming a closed-loop process that repeats until the time parameter reaches $t = 1$, at which point the output solution is returned.

σ_i whose orientations correspond to candidate configurations x_i of the original problem. The objective function together with constraint penalties are embedded into the energy landscape through coupling coefficients J_{ij} and local-field term h_i, yielding the system Hamiltonian with the following form:

$$H_{ising} = \sum_{ij}^{n} J_{ij}\sigma_i\sigma_j + \sum_{i}^{n} h_i\sigma_i \quad (1)$$

where the ground state of the Hamiltonian corresponds to the optimal feasible solution.

Traditional classical approaches for solving Ising-type optimization problems, such as SA, rely on sequential updates of spin configurations to explore the energy landscape through stochastic perturbations. As the system size increases, the search space grows rapidly, making it increasingly difficult to efficiently identify the global minimum. In contrast, quantum annealing describes the system state as evolving in a global Hilbert space, where a superposition state encodes multiple candidate configurations simultaneously. Beyond this parallel representation, quantum annealing is governed by an adiabatic process in which the system follows the ground state of a time-dependent Hamiltonian, enabling a continuous dynamical exploration of the energy landscape. The potential computational advantage of quantum annealing is often attributed to the combination of superposition-based representation and Hamiltonian-driven evolution, which together provide a search mechanism distinct from stochastic local updates.

To harness the parallelism in practical computation, we construct the QIFA that emulates the dynamical characteristics of quantum adiabatic evolution within a classical framework. The

complete workflow is illustrated in Fig. 1C. The algorithm begins by encoding the superposition states of spins in the Ising model through classical variables. In the QIFA framework, each spin in the Ising model is in a superposition of spin-up and spin-down states. We introduce an analog variable $\theta_i \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ to characterize the orientation of the ith spin. The superposition state of the ith spin is represented as vector S_{θ_i}:

$$S_{\theta_i} = \left[sin\left(\frac{\theta_i}{2} + \frac{\pi}{4}\right), cos\left(\frac{\theta_i}{2} + \frac{\pi}{4}\right)\right]^T \tag{2}$$

Specifically, $[1,0]^T$ denotes the spin-up state, while $[0,1]^T$ denotes the spin-down state. The vector S_{θ_i} represents a rotation on the Bloch sphere parameterized by angle θ_i, mapping the computational basis spin-up and spin-down states to a superposition state with amplitudes given by the sine and cosine terms. Note that the probabilities corresponding to the 2 basis states are the squared moduli of the sine and cosine terms, thereby inherently satisfying the normalization condition. In contrast, the classical continuous relaxation lacks a direct physical correspondence and, consequently, do not impose a normalization requirement [9].

The expectation values of the Pauli operators σ_i^z and σ_i^x, which represent the energy contributions of each spin along the z axis and x axis, respectively, are straightforwardly computed as quadratic forms in the vector S_{θ_i}. Specifically, we have:

$$S_{\theta_i}^T \sigma_i^z S_{\theta_i} = sin\theta_i \tag{3}$$

$$S_{\theta_i}^T \sigma_i^x S_{\theta_i} = cos\theta_i \tag{4}$$

The forms of Eqs. (3) and (4) are consistent with the standard expressions for the expectation values of observables in quantum mechanics. These expressions provide a direct relationship between the spin state and its corresponding energy contribution, which are key to the Ising model's energy landscape.

The complete system state is represented as a tensor product of individual spin states, as follows:

$$\boldsymbol{S_\theta} = S_{\theta_1} \otimes S_{\theta_2} \otimes \ \dots \ \otimes S_{\theta_n} \tag{5}$$

Here, the vector $\boldsymbol{\theta} = \left[\theta_1, \theta_2, \dots, \theta_n\right]^T$ collectively defines the configuration of all spins in the system, with each θ_i representing the orientation of the ith spin. As described in Eq. (2), each S_{θ_i} resides in a 2-dimensional Hilbert space. In Eq. (5), the full system state is constructed by taking the tensor product of n such 2-dimensional spin states, resulting in a composite Hilbert space of dimension 2^n. This collective representation forms the complete system state. In essence, QIFA leverages this high-dimensional Hilbert space to explore multiple potential solutions simultaneously, enabling efficient global search over the solution space.

Afterward, evaluating the state described in Eq. (5), the iterative process is inspired by the quantum adiabatic theorem. The theorem states that if a quantum system starts in the ground state of a simple initial Hamiltonian, and this Hamiltonian is varied sufficiently slowly into a complex final Hamiltonian that encodes the target problem, the system will remain in the instantaneous ground state throughout the evolution, thereby arriving at the optimal solution. In the QIFA framework, this inspires the time-dependent Hamiltonian:

$$H(t) = t\gamma H_z - (1-t)H_x \tag{6}$$

where H_z denotes the matrix representation of the target Hamiltonian, H_x represents the matrix form of the transverse-field Hamiltonian, and γ serves as the relative strength governing the dominance of H_z in the time-dependent system evolution.

$$H_z = \sum_{ij}^n J_{ij}\sigma_i^z\sigma_j^z + \sum_i^n h_i\sigma_i^z \tag{7}$$

and

$$H_x = \sum_i^n \sigma_i^x \tag{8}$$

It can be seen that at $t = 0$, Eq. (6) yields $H(0) = H_x$, whose ground state corresponds to $\boldsymbol{S}_{(\boldsymbol{\theta}=0)}$ as described in Eq. (5), with all θ_i set to zero.

In contrast to quantum annealing, which follows unitary evolution under the Schrödinger equation, this framework simulates the system's progression toward the ground state by updating variable $\boldsymbol{\theta}$ through gradient descent. The time-dependent cost function is defined as the expectation value of the system's Hamiltonian $H(t)$ at time t, which corresponds to the quadratic form of $H(t)$ under the vector $\boldsymbol{S_\theta}$:

$$\begin{aligned} C(t,\boldsymbol{\theta}) &= \boldsymbol{S}_{\boldsymbol{\theta}}^T H(t) \boldsymbol{S}_{\boldsymbol{\theta}} \\ &= \boldsymbol{S}_{\boldsymbol{\theta}}^T \left(t\gamma H_z - (1-t)H_x\right)\boldsymbol{S}_{\boldsymbol{\theta}} \\ &= t\gamma\left(\boldsymbol{z}^T J \boldsymbol{z} + \boldsymbol{z}^T \cdot h\right) - (1-t)\boldsymbol{x}^T \cdot \boldsymbol{1} \end{aligned} \tag{9}$$

where $\boldsymbol{z} = \left[sin\theta_1, sin\theta_2, \dots sin\theta_n\right]^T$ and $\boldsymbol{x} = \left[cos\theta_1, cos\theta_2, \dots cos\theta_n\right]^T$. The gradient of the cost function with respect to analog variables $\boldsymbol{\theta}$ is given by:

$$\nabla_{\boldsymbol{\theta}} C(t,\boldsymbol{\theta}) = t\gamma(2J\boldsymbol{z} + h) \circ \boldsymbol{x} + (1-t)\boldsymbol{z} \tag{10}$$

where $\circ$ denotes element-wise multiplication between vectors.

We update the analog variables $\boldsymbol{\theta}$ via gradient descent (corresponding to the gradient descent module in Fig. 1C):

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-\tau} - \eta \nabla_{\boldsymbol{\theta}} C\left(t, \boldsymbol{\theta}_t\right) \tag{11}$$

where $\boldsymbol{\theta}_{t-\tau}$ is the analog variables obtained at the previous time, τ is the time evolution step size, and η is the step size in gradient descent. By minimizing $C(t,\boldsymbol{\theta})$, this gradient descent process iteratively adjusts $\boldsymbol{\theta}$ to drive the vector $\boldsymbol{S_\theta}$ toward the ground state of $H(t)$ approximately at each step, thereby implementing an approximation of tracking the ground state described by the quantum adiabatic theorem.

At each iteration, a probabilistic projection step is applied to the current vector $\boldsymbol{\theta}_t$ to generate a discrete spin configuration. This projection is implemented through a roulette-wheel sampling strategy, in which spin orientations are selected according to probabilities induced by $\boldsymbol{\theta}_t$, producing a candidate solution $\boldsymbol{p}_t$ that satisfies the problem constraints. The corresponding Hamiltonian value is evaluated for $\boldsymbol{p}_t$, and the best feasible configuration encountered so far is recorded as $\boldsymbol{p}_{best}$ (corresponding to the project-update module in Fig. 1C). This roulette-wheel sampling is analogous to quantum measurement in quantum systems, where the measurement outcome is determined through probabilistic sampling based on the system's quantum state.

In CCO problems, the energy landscape shaped by constraints is typically highly rugged, with feasible regions forming isolated basins separated by high-energy barriers. This rugged energy landscape easily traps searches in local minima. To escape these local traps and enhance search efficiency nearby the current optimal solution, we apply feedback adjustment to update variables $\boldsymbol{\theta}_t$ in the direction of the optimal solution (corresponding to the feedback module in Fig. 1C):

$$\boldsymbol{\theta}_t \leftarrow (1-\varsigma)\boldsymbol{\theta}_t + \varsigma \boldsymbol{p}_{best} \tag{12}$$

where ς is the feedback coefficient controlling the strength of this adjustment. This operation acts as a mechanism that integrates discrete optimization feedback into the dynamical evolution.

After completing the feedback process at time t, the time parameter is updated to $t+\tau$. Subsequently, the time-dependent Hamiltonian evolves to $H(t+\tau)$, followed by the executions of gradient descent and project–feedback strategy to analog variables $\boldsymbol{\theta}_{t+\tau}$. This iterative cycle repeats until the time parameter reaches $t=1$, concluding with the historical optimal solution $\boldsymbol{p}_{best}$ as the final output. Through this evolution and feedback, QIFA provides a structured search process that promotes efficient convergence to high-quality feasible solutions.

Numerical results

We take TSP as the benchmark to demonstrate the rapid solving capability of QIFA. TSP is a renowned CCO problem: Given a set of cities and pairwise distances, it seeks the shortest tour visiting each city exactly once and returning to the origin. This problem holds substantial practical relevance across domains including logistics, urban planning, bioinformatics, and chip design. The formal mathematical formulation of TSP, along with its corresponding Ising model Hamiltonian and constrained projection strategies, is detailed in the Methods section.

For city scales ranging from 10 to 100 in increments of 10, we generate random city coordinates with the uniform distribution, producing 50 distinct TSP instances per scale. Firstly, we examine the convergence properties of the QIFA by plotting its average percentage error [$\mathrm{PEav} = \mathrm{avg}\left(\frac{E-E_0}{E_0}\times 100\%\right)$, with E_0 denoting the optimal solution and PEav quantifying solution quality, where lower values indicate superior performance] under varying iterative ratios λ (defined as $\lambda = L/N$, with L being the iteration count and N the city scale). As shown in Fig. 2A, negligible differences between $\lambda = 40$ and $\lambda = 50$ across all scales indicate that QIFA reaches stable convergence within this range.

We further compare QIFA with SA [3], AC [4], and local quantum annealing (LQA) [35]. Figure 2B illustrates the solution quality across scales, with all curves generated under iterative ratios $\lambda = 50$. The red line depicts PEav trend of QIFA, while blue, purple, and green lines represent results of SA, AC, and LQA, respectively. The shaded areas indicate the interquartile range (25% to 75%) of percentage errors for each algorithm. All the results are generated from 100 repeated trials across 50 instances. QIFA consistently achieves lower percentage errors than competing algorithms, with performance advantages becoming more pronounced as the problem scale increases.

Figure 2C and D details the convergence trends of the algorithms at scales of 50- and 100-city, respectively. The results for the remaining city scales are illustrated in Supplementary Information S1 The lines represent the PEav of different approach, while the shaded areas indicate the interquartile range of percentage errors. QIFA rapidly reduces PEav in early iterations and maintains consistently best solution quality compared with other algorithms as iteration count increases, demonstrating substantial advantages through faster convergence rates and superior solution accuracy.

To further quantify scalability, we analyze the number of iterations required to reach near-optimal solutions as a function of city size. As shown in Supplementary Information S2, QIFA empirically exhibits an approximately linear relationship between iteration count and problem scale within the randomly generated TSP instances considered here (city sizes from 10 to 100 with the uniform distribution of city coordinates). In contrast, SA shows a quadratic growth trend under the same solution accuracy, indicating a progressively increasing performance gap as the problem scale increases.

This near-linear scaling behavior can be interpreted through the dynamical structure of QIFA, which integrates 3 key quantum-inspired mechanisms: superposition encoding, adiabatic evolution, and projection–feedback. First, the superposition encoding represents the system state as a product-state parameterization in an exponentially dimensional Hilbert space. This encoding maintains a probabilistic description of multiple candidate configurations, allowing the algorithm to simultaneously explore the combinatorial landscape instead of sequentially evaluating individual states. This global representation substantially enhances the efficiency of the search process. Second, the optimization trajectory is guided by the adiabatic evolution of the system. The time-dependent Hamiltonian evolves gradually from an initial state that explores the entire solution space to the problem-specific Hamiltonian. This mirrors quantum adiabatic evolution, where the system follows the ground state of the time-dependent Hamiltonian, ensuring a smooth transition from exploration to exploitation of the problem landscape. Third, the projection-feedback mechanism plays a role analogous to quantum measurement, converting the continuous system state into feasible discrete configurations through probabilistic sampling. The feedback mechanism continuously updates the solution and adjusts the search direction, ensuring that the optimization remains within feasible regions of the parameter space while improving solution quality. Through the combined effect of these 3 mechanisms, the optimization trajectory evolves smoothly toward low-energy feasible states, avoiding the need for exhaustive enumeration of configurations.

Parameter sensitivity remains a critical challenge in heuristic optimization, often requiring extensive tuning to achieve peak performance and posing a major obstacle to real-world applicability. To systematically evaluate this aspect, we evaluate QIFA on a 20-city TSP benchmark across a wide parameter space, including Hamiltonian relative strength $\gamma \in [0.1,100]$, gradient descent step size $\eta \in [0.1,10]$, and feedback coefficient $\varsigma \in [0.1,0.99]$.

As detailed in Supplementary Information S3, QIFA consistently achieves low error rates (PEav < 0.8%) across all parameter combinations, indicating strong robustness to parameter variations. This robustness can be understood from the roles of the parameters in the dynamical framework. The parameter γ controls the relative dominance of the target Hamiltonian during evolution, influencing the balance between exploration and convergence speed. The step size η governs the magnitude of gradient-driven updates, primarily affecting convergence stability rather than final solution quality within the tested range. The feedback coefficient ς determines the strength of exploitation

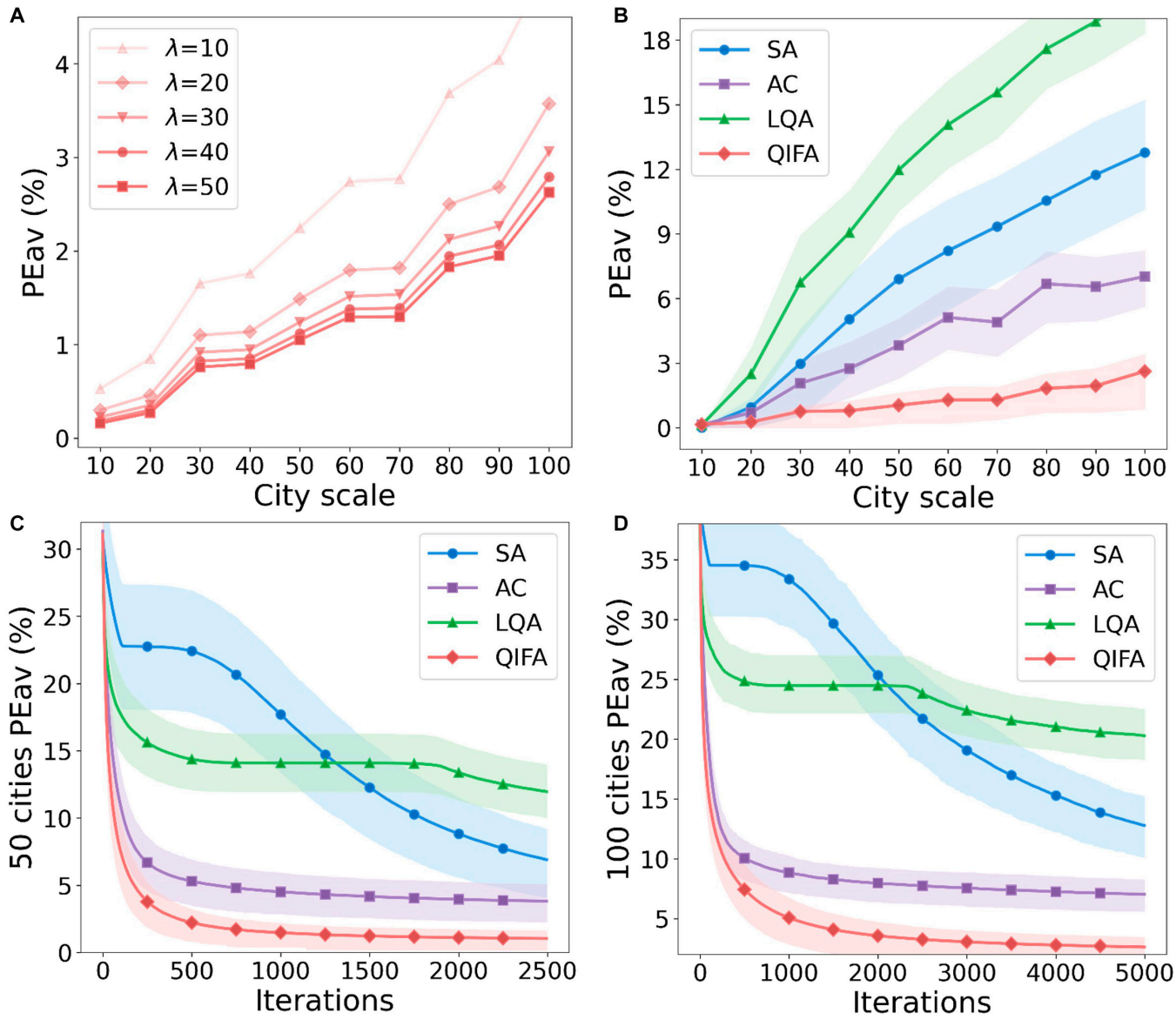


Fig. 2. Numerical results on TSP instances across scales. (A) Solution quality of QIFA under varying iterative ratios λ for city scales from 10 to 100. (B) Solution quality comparison of SA, AC, LQA, and QIFA at city scales from 10 to 100, with iteration counts fixed at 50 times the city scale; lines show PEav from 100 repeated trials across 50 random instances, while shaded areas indicate interquartile ranges of percentage errors. (C) Solution quality versus iteration counts for SA, AC, LQA, and QIFA at 50-city scale, with lines representing PEav from 100 trials over 50 random maps and shaded areas depicting interquartile ranges of percentage errors. (D) Solution quality versus iteration counts at 100-city scale.

toward the best-known configuration, modulating the trade-off between global exploration and local refinement. Together, these parameters influence the convergence dynamics but do not fundamentally alter the structure of the optimization trajectory, which explains the observed robustness of QIFA across a broad parameter regime.

To further validate algorithmic availability, we benchmark QIFA against 20 standard instances from the TSPLIB [37]—a widely adopted dataset for TSP with known optimal solutions. The selected instances are chosen to represent diverse problem characteristics varying city scale from 16 to 105 with different topological structures (Euclidean, geographic, and artificial configurations). As summarized in Supplementary Information S4, QIFA consistently achieves lower percentage errors than the comparison algorithms across the benchmark suite, with performance advantages becoming more pronounced for larger problem sizes. These results indicate that the efficiency and solution quality improvements observed in randomly generated instances extend to widely adopted real-world benchmarks.

To complement the solution quality analysis with computational efficiency, we further evaluate the runtime performance of QIFA and compare it with the Lin–Kernighan–Helsgaun (LKH) heuristic, a highly optimized solver widely used for TSP. All experiments are conducted on a CPU platform (Intel Core i7-9700, 3.00 GHz). To ensure a fair benchmarking setting, QIFA is executed up to its empirical saturation threshold of $\lambda = 50$. The LKH baseline is implemented using the standard Python wrapper (elkai.solve_int_matrix) with default parameters and without modification. The runtime for both methods is measured from the completion of the distance-matrix construction

until the final solution is returned. Runtime statistics are obtained by averaging over 100 independent evaluations. Detailed runtime definitions are provided in Supplementary Information S5. QIFA achieves runtimes on the order of 10^{-1} to 10^{0} s across the tested TSPLIB benchmark instances ranging from 29 to 100 cities, and operates within the same order of magnitude as LKH. Overall, these results demonstrate that QIFA attains high-quality solutions while maintaining high computational efficiency, supporting its practical applicability for combinatorial optimization problems.

The results presented above are simulated using classical computers. When implementing the QIFA on classical computers, spin superposition states require encoding through extensive binary expansions using high and low voltage levels, which fails to substantially embody the properties of quantum superposition. On the other hand, recent investigations have shown that classical electric circuits can be used to simulate various topological physics [38–52], Schrödinger equation [53,54], and quantum algorithms [55], based on the similarity between circuit Laplacian and lattice Hamiltonian. In the following, we explore how to design circuit networks to perform the QIFA, which makes them have processing functions similar to that of quantum computation.

Circuit theory for QIFA and simulation results

The designed classical circuit to implement the full demonstration of QIFA is shown in Fig. 3, which consists of 2 modules, the matrix multiplication module in the blue area and the addition amplifier module in the yellow area.

The matrix multiplication module performs the core matrix–vector multiplication operation, which consists of n input nodes (indexed as $k = 1, 2, \ldots n$) and n output nodes (indexed as $l = 1, 2, \ldots n$). Each input node k connects to an external power signal V_k^{i1}, and resistor R_{kl} bridges every input node k and output node l. R_0 is the fixed resistance. Considering the virtual-short characteristics of operational amplifiers and applying Kirchhoff's law, the voltage V_l^{o1} at output node l satisfies:

$$V_l^{o1} = -\sum_k \frac{R_0 V_k^{i1}}{R_{kl}} \tag{13}$$

The addition amplifier module performs the weighted sum operation, which comprises n node. For each node l, an operational amplifier is connected to corresponding output V_l^{o1} and external power signals V_l^{i2} and V_l^{i3}, incorporating with resistors R_l^1, R_l^2, and R_l^3, respectively. Then, the output voltage V_l^{o2} satisfies:

$$V_l^{o2} = R_0\left(\frac{V_l^{i2}}{R_l^2} + \frac{V_l^{i3}}{R_l^3} - \frac{V_l^{o1}}{R_l^1}\right) \tag{14}$$

By combining the voltages at all nodes, we can define the voltage sequence $\boldsymbol{V} = \left[V_1, V_2, \ldots, V_n\right]^T$, manifesting as a superposition formulation of the voltages. In the matrix multiplication module, the input voltages $\boldsymbol{V}^{i1} = \left[V_1^{i1}, V_2^{i1}, \ldots, V_n^{i1}\right]^T$ correspond to the superposition state of spins as shown in Eq. (3), which is physically realized as $\boldsymbol{V}^{i1} = \sin\boldsymbol{\theta}$. Meanwhile, the second-order spin correlation coefficients J_{kl} of the Ising

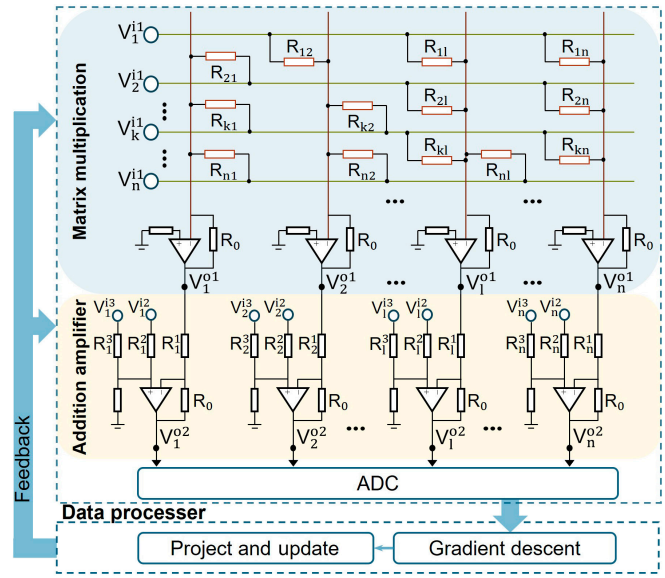


Fig. 3. Schematic diagram of the classical circuit system. Blue: Matrix multiplication module. Yellow: Addition amplification module. The superposition state of spins is physically realized through voltage superposition across the nodes, and the coefficients of the Hamiltonian are modulated by resistors and input voltages. The data processor controls input to the classical circuit, receives ADC outputs as gradients of the cost function, and executes the gradient descent, constrained projection, and feedback adjustment.

model are implemented by modulating resistors $R_{kl} = \frac{R_0}{J_{kl}}$. After the first operational amplifier stage, according to Eq. (13), the output voltage sequence $\boldsymbol{V}^{o1} = \left[V_1^{o1}, V_2^{o1}, \ldots, V_n^{o1}\right]^T$ is:

$$\boldsymbol{V}^{o1} = -J\sin\boldsymbol{\theta} \tag{15}$$

In the addition amplifier module, the physical encoding of the transverse-field terms h_l of the Ising model is achieved through the input voltages as $V_l^{i2} = h_l$ and the superposition state is realized as $V_l^{i3} = \sin\theta_l$, respectively. The resistors are modulated as $R_l^1 = \frac{R_0}{2t\gamma\cos\theta_l}$, $R_l^2 = \frac{R_0}{t\gamma\cos\theta_l}$, and $R_l^3 = \frac{R_0}{(1-t)}$ to scale the coefficients of the Hamiltonian. After the second operational amplifier stage, according to Eq. (14), the final output voltage sequence $\boldsymbol{V}^{o2} = \left[V_1^{o2}, V_2^{o2}, \ldots, V_n^{o2}\right]^T$ is obtained as:

$$\boldsymbol{V}^{o2} = t\gamma(2J\sin\boldsymbol{\theta} + h)\circ\cos\boldsymbol{\theta} + (1-t)\sin\boldsymbol{\theta} \tag{16}$$

This resulting voltage output matches the gradient $\nabla_{\boldsymbol{\theta}}C(t,\boldsymbol{\theta})$ in Eq. (10). After recording the output via analog-to-digital converter (ADC), the data processor executes gradient descent and constrained projection to generate trial solutions and update the historical optimal solution. Subsequently, the processor feeds the updated parameters back to the circuit for the next iteration cycle. This process iterates until the target iteration count is reached.

To validate the theoretical analysis, we perform numerical simulations (using LTspice software) of the circuit implementations for the TSP problems at varying scales, with specific circuit parameters detailed in the Methods section. For example, we conduct comparative evaluations on TSPLIB instances including the 29-city instance (bayg29), 48-city instance (att48), 76-city instance (eil76), and 100-city instance (rd100). Detailed solution progression is provided in Supplementary Information S6. The simulation results consistently align with theoretical

predictions, exhibiting matching convergence trends in solution energy as iterations increase.

Experimental results

To validate the practical problem-solving capability of executing QIFA on classical circuit systems, we select 20 major global cities as a case study for TSP, seeking the shortest tour traversing all cities. The names, locations, and optimal tour for these 20 cities are visualized in Fig. 4A. The specific latitude and longitude coordinates for each city are provided in Supplementary Information S7.

The designed circuit is fabricated on the printed circuit board (PCB) for solving the 20-city global TSP, as shown in Fig. 4B. The upper region features the matrix multiplication module implemented via a 20×20 precision chip resistor network (Fig. 4C), where the resistance values are configured to be inversely proportional to the spin correlation coefficients of the Ising model, thereby enforcing the TSP constraint that each city is visited exactly once. Subsequently, the first-stage operational amplifier circuit (Fig. 4D) converts the output current of the matrix multiplication module into the input voltage V_k^{o1} for the next layer. The lower region constructs the addition amplifier module using 20 operational amplifiers and adjustable resistors, where interconnecting resistance values represent intercity distances in the TSP problem. The second-stage operational amplifier circuit (Fig. 4E) performs addition and amplification to generate the output voltage V_k^{o2} corresponding to the gradient. In the current prototype, parameter updates are performed in a hardware-in-the-loop manner: Circuit outputs are measured at each iteration, and the updated parameters are computed externally and manually tuned into the circuit, enabling iterative execution of the QIFA dynamics. Further details of the board configuration and experimental setup are provided in the Methods section.

Figure 4F compares simulation and experimental result of tour lengths versus iteration counts for QIFA on the 20-city global TSP. The red line corresponds to the average of experimental results, and the blue line corresponds to the average of simulation results, with the error bars indicating the fluctuation ranges. We choose an appropriate density of the points to plot error bars per 100 iterations, which can not only keep the continuity of the results but also make the results visible and accurate. As iterations increase, both simulated and experimental results gradually decrease their fluctuation ranges, exhibiting consistent convergence trends

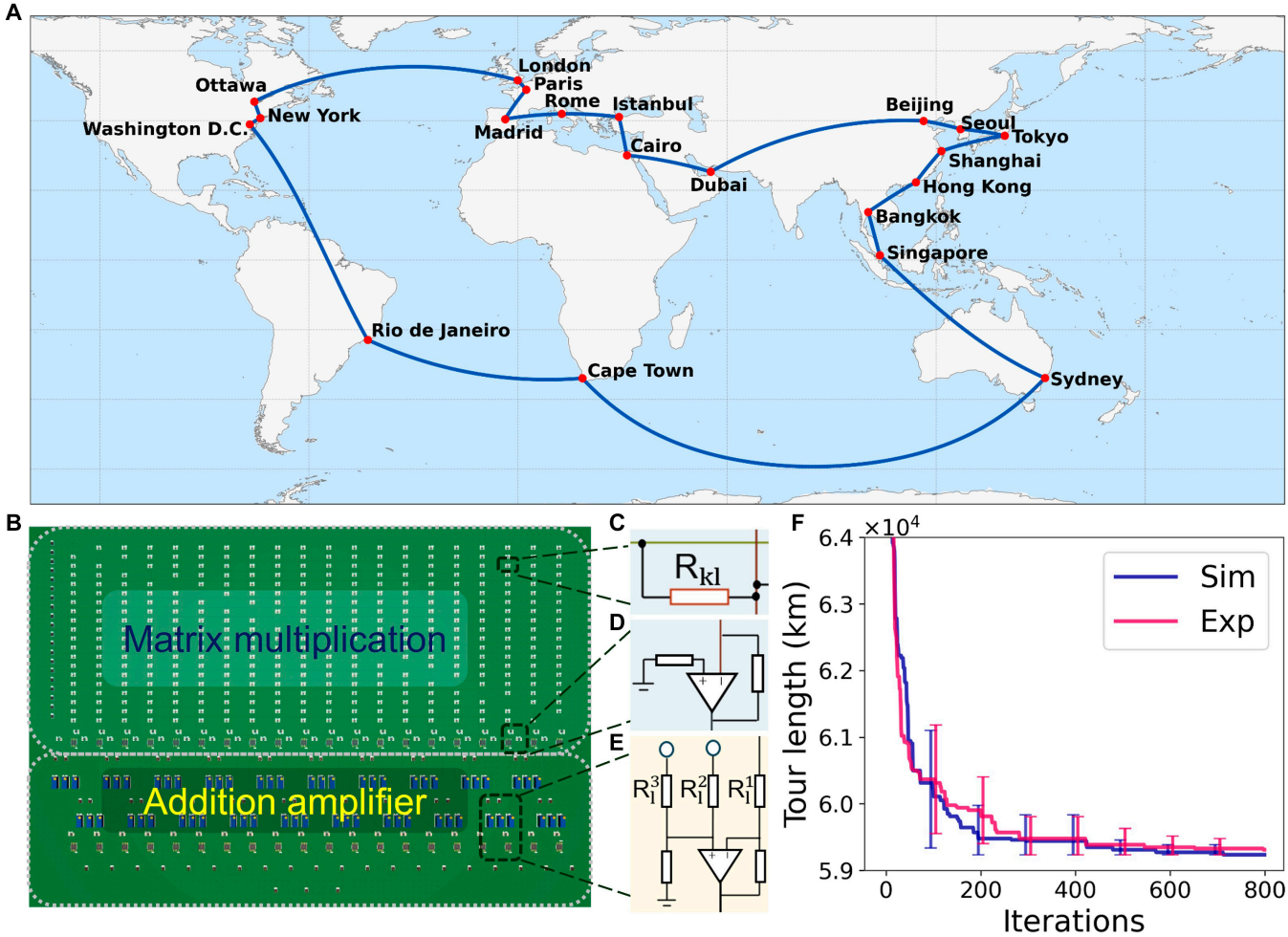


Fig. 4. Experimental validation on 20-city global TSP. (A) Geographical distribution of 20 global cities and the optimal tour (blue lines), with coordinates mapped based on the WGS-84 ellipsoidal Earth model. (B) Layout of PCB hardware: Upper silkscreen area corresponds to the matrix multiplication module; lower section corresponds to the addition amplification module. (C) One precision chip resistor in the network of the matrix multiplication circuit module. (D) One block of the first-stage operational amplifier circuits converts the output currents of matrix multiplication circuit to voltages. (E) One block of the second-stage operational amplifier circuits in the addition amplification circuit module. (F) Simulated and experimental tour lengths versus iterations for QIFA on the 20-city global TSP, with the error bars indicating the fluctuation ranges.

approaching near-optimal tour solutions, and they together prove the effectiveness of our design. We further evaluate the impact of component tolerance on solution quality in Supplementary Information S8. The resulting tour lengths consistently converge across all tolerance levels. Notably, under the 1% tolerance condition—corresponding to the precision grade adopted in our physical experiments—the performance nearly matches that of the ideal tolerance-free scenario, demonstrating that the algorithm remains robust under real-world hardware constraints.

The above experiments demonstrate the availability of implementing QIFA in classical circuits, where the input–output relationship derived using Kirchhoff's law establishes precise correspondence with the gradient of QIFA's cost function. This classical circuit architecture leverages mature electronics manufacturing and control technologies to manifest effects equivalent to quantum state superposition through classical voltage–current superposition. Owing to inherent characteristics of classical circuits, this approach offers notable advantages: It can run in a normal environment without special requirements and has good scalability and stability.

Although the current physical setup operates as a proof of concept, the underlying analog circuit architecture is inherently compatible with automated digital control logic for fully integrated scaling. To implement QIFA in practical hardware systems, the aforementioned circuit requires collaboration with a field-programmable gate array (FPGA) for parameter modulations. In typical PCB circuit systems, FPGA clock frequencies operate at ~10 MHz, with unit response times on the order of 1 μs. Each iteration for an N-city TSP problem necessitates $2N$ parameter modulation operations, and the convergence requires approximately $50N$ iterations—totaling $100N^2$ parameter modulation response times. Consequently, FPGA-driven parameter modulation in PCB systems yields computation times of approximately 40 ms for 20-city instances, 250 ms for 50-city instances, and 1 s for 100-city instances. In application-specific integrated circuit (ASIC) systems with clock frequencies reaching ~1 GHz, QIFA is estimated to achieve computation times of approximately 0.4 ms for 20-city instances, 2.5 ms for 50-city instances, and 10 ms for 100-city instances, which is 2 to 3 orders lower than running on Intel Core-i7 CPU with the main frequency of 3 GHz.

Discussion

Several prior studies employing quantum or quantum-inspired approaches have also adopted TSP as a benchmark for validation, and their reported performance on different problem sizes is summarized in Table 1. For instance, quantum annealing implemented on the D-wave quantum annealer achieved a PEav of 2.70% for a 22-city instance, which increased markedly to 25.91% for a 38-city instance, indicating clear scalability challenges [29]. Hybrid quantum-classical evolutionary algorithms combined classical evolution with NISQ devices, reporting a PEav of 7.4% for a 15-city TSP [56]. The quantum approximate optimization algorithm (QAOA), a widely used variational quantum algorithm for combinatorial optimization, reported a PEav of approximately 10% for a 5-city TSP [20]. In addition, quantum machine learning approaches based on equivariant quantum circuits attained PEav values of around 5% and 13% for 10- and 20-city TSP instances, respectively [20]. Quantum-inspired parallel annealing, which is also motivated by adiabatic evolution but does not explicitly incorporate constraint handling, achieved a PEav of about 5% for a 9-city TSP [11]. In addition, deep learning-based methods are a popular and active research approach for solving TSP in recent years [57,58]. Such approaches typically require problem-specific training procedures and considerable training time before deployment. Their computational workflow differs fundamentally from the iterative optimization paradigm considered in this work.

Notably, these quantum and quantum-inspired approaches generally suffer from either limited scalability or insufficient constraint handling, which leads to a rapid degradation of solution quality as the problem size increases. In contrast, the proposed QIFA synergistically combines the global exploration capability inherent in quantum-inspired paradigms with a projection–feedback strategy. This synergy enforces strict constraint satisfaction while enabling efficient navigation of the solution space, resulting in substantially improved solution quality, with a PEav of 0.2% for 20-city instances in experiment result and a PEav of approximately 3% for 100-city instances in numerical study.

On the other hand, although this article primarily focuses on TSP as an example, the QIFA framework demonstrates high flexibility and can be adapted to a wide range of CCO problems. For instance, when addressing KP, it suffices to map the problem onto its corresponding Ising model Hamiltonian, construct a time-dependent cost function for gradient descent, and design constraint-specific projection strategies under capacity limitations, along with feedback adjustments. This approach also achieves high-quality solutions for KP, demonstrating QIFA's effectiveness across various problem types. The detailed QIFA theory and results for KP are presented in Supplementary Information S9 and S10. This structured approach enables seamless deployment across diverse problem domains, supporting QIFA as a versatile and scalable solution for solving a wide array of constrained optimization problems, beyond just TSP.

In summary, this paper proposes QIFA that delivers an efficient framework for the CCO problems. The core innovation is encoding spin superposition states with classical variables, which enables gradient-driven global search with parallel capabilities. Meanwhile, project–feedback strategy accelerates convergence by intensifying exploration in high-quality regions. We demonstrate the feasibility of implementing the QIFA framework in classical circuit systems, whose architecture aligns with mainstream semiconductor manufacturing capabilities. This approach operates robustly in standard environments without quantum-specific conditions, exhibiting scalability and stability. Leveraging ASICs or FPGAs for parallel parameter modulation, QIFA holds promise for real-time optimization in large-scale practical applications, thereby facilitating the deployment of industrial-grade real-time optimization systems.

Methods

The QIFA theory for TSP

TSP can be formulated as a quadratic assignment problem by defining a set of binary variables representing whether each edge belongs to the solution tour. Specifically, for a graph $G = (V, E)$, TSP is expressed as:

$$\min \sum\nolimits_{(i,j)\in E} d_{ij} a_{ij} \tag{17}$$

$$\text{s. t. } \sum\nolimits_{i\in V\setminus\{j\}} a_{ij} = 1, j \in V \tag{18}$$

Table 1. Comparison of QIFA with representative quantum and quantum-inspired algorithms for solving TSP. This table compares the performance of QIFA with other quantum-inspired and quantum algorithms across various TSP problem sizes. The PEav (%) values represent the average percentage error relative to the optimal solution, with lower values indicating better solution quality. The Platform column specifies the hardware or simulation environment used for each algorithm, and the Remarks column provides additional context, including distinguishing features, scalability, and constraint handling capabilities.

Method	Size	PEav (%)	Platform	Remarks
Quantum annealing [29]	22	2.7%	D-wave quantum annealer	Performance degrades with increasing cities
	38	26%		
Hybrid quantum-classical evolutionary algorithm [56]	15	7.4%	Cloud-based NISQ platforms (IBM Quantum, AWS Braket)	Hybrid quantum inspired evolutionary algorithm with NISQ devices
QAOA [20]	5	~10%	Numerical study	Variational quantum algorithm; limited by the number of qubits
Equivariant quantum machine learning [20]	10	~5%	Numerical study	Leveraging the symmetry properties; effective for small-scale problems
	20	~13%		
Quantum-inspired parallel annealing [11]	9	5%	Analog memristor crossbar	Inspired by quantum annealing; no explicit constraint handling
	17	~10%		
QIFA (this work)	20	0.2%	Circuit network	Quantum-inspired Hamiltonian dynamics and projection–feedback
	100	~3%	Numerical study	

$$\sum_{j\in V\setminus\{i\}} a_{ij} = 1, i \in V \tag{19}$$

$$\sum_{i\in S}\sum_{j\notin S} a_{ij} \geq 1, S \subsetneq V, |S| \geq 2 \tag{20}$$

Here, binary variables a_{ij} indicate whether edge E_{ij} belongs to the solution tour, and d_{ij} denotes the length of edge E_{ij}. Equation (17) represents the optimization objective, corresponding to the total length of the solution tour; Eqs. (18) and (19) are equality constraints ensuring exactly one path departing from and terminating at each city, respectively; Eq. (20) is an inequality constraint guaranteeing that the solution contains exactly one ring.

In our algorithm, the inequality constraint [Eq. (20)] is implemented during the constrained projection via the roulette-wheel strategy, while the equality constraints [Eqs. (18) and (19)] are incorporated into the objective term:

$$\begin{aligned} H = &\sum_{(i,j)\in E} d_{ij}a_{ij} + D\sum_i\left(\sum_{j\in V\setminus\{i\}} a_{ij} - 1\right)^2 \\ &+ D\sum_j\left(\sum_{i\in V\setminus\{j\}} a_{ij} - 1\right)^2 \end{aligned} \tag{21}$$

where $D = \max\left\{d_{ij}\right\}$. After variable substitution and rearrangement, Eq. (21) yields the Ising model formulation for TSP:

$$H_z^{TSP} = \sum_{(i,j),(k,l)\in E} J_{ij,kl}\sigma_{ij}^z\sigma_{kl}^z + \sum_{(i,j)\in E}^n h_{ij}\sigma_{ij}^z \tag{22}$$

where

$$\begin{cases} J_{ij,kl} = \dfrac{D}{4}, & i = k \text{ or } j = l, \\ J_{ij,kl} = 0, & otherwise. \end{cases} \tag{23}$$

$$h_{(i,j)} = \frac{d_{ij}}{2} - (n-1)D \tag{24}$$

In QIFA for TSP, the time-dependent Hamiltonian is

$$H^{TSP}(t) = t\gamma H_z^{TSP} - (1-t)H_x^{TSP} \tag{25}$$

We introduce analog variables $\boldsymbol{\theta} = \left[\theta_{11}, \theta_{12}, \ldots, \theta_{ij}, \ldots, \theta_{NN}\right]^T$, where $\theta_{ij} \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ represents the orientation of the ijth spin. The time-dependent cost function corresponds to the quadratic form of $H^{TSP}(t)$ with the vector $\boldsymbol{S}_{\boldsymbol{\theta}}$:

$$C(t, \boldsymbol{\theta}) = \boldsymbol{S}_{\boldsymbol{\theta}}^T H^{TSP}(t)\boldsymbol{S}_{\boldsymbol{\theta}} \tag{26}$$

The partial derivative of the cost function with respect to parameter θ_{ij} is given by:

$$\frac{\partial C(t,\boldsymbol{\theta})}{\partial \theta_{ij}} = t\gamma\left(\frac{\lambda}{2}\sum_l z_{il} + \frac{\lambda}{2}\sum_k z_{kj} + \frac{d_{ij}}{2} - (N-1)D\right)x_{ij} + (1-t)z_{ij} \tag{27}$$

To obtain constraint-satisfying trial solutions $\boldsymbol{p}_t$ from $\boldsymbol{\theta}$, the constrained projection strategy is configured as follows: Let Y denote the set of visited city indices and i the index of the last visited city. The probability P_{ij} for edge E_{ij} to be chosen in the solution tour via roulette-wheel selection is set to:

$$P_{ij} = \begin{cases} \dfrac{\left(1+z_{ij}\right)^2 d_{ij}^{\beta}}{\sum_{l\notin C}\left(1+z_{il}\right)^2 d_{il}^{\beta}}, & j \notin Y \\ 0, & j \in Y \end{cases} \tag{28}$$

where $z_{ij} = sin\theta_{ij}$ and β is the edge-length heuristic factor. Because the roulette-wheel selection is performed sequentially with the visited set updated after each selection, the procedure explicitly enforces connectivity and prevents the formation of subtours, guaranteeing that the trial solution $\boldsymbol{p}_t$ contains exactly one tour with each city visited exactly once.

The circuit parameters for TSP

For an N-city TSP problem mapped to an Ising model with N^2 spins, the standard QIFA implementation would require N^2 circuit nodes. However, leveraging the sparsity and symmetry revealed in Eq. (27), we reduce hardware complexity by utilizing N circuit nodes to compute the cost function gradient through $2N$ repeated computations per iteration. The experimental configuration operates as follows:

In the ith computation for component $\frac{D}{2}t\gamma\sum_l z_{il}x_{ij}$, Resistors between input node k and output node $l$are modulated as $R_{kl} = \frac{4R_0}{D}$. Resistors in node j are set to $R_j^1 = \frac{R_0}{2t\gamma x_j}, R_j^2 = 0, R_j^3 = 0$. Input voltages are set to $V_j^{i1} = z_{ij}, V_j^{i2} = 0, V_j^{i3} = 0$, yielding output voltage in ith computation $\boldsymbol{V}_{i,j}^{o2} = \frac{\lambda}{2}t\gamma\sum_l z_{il}x_{ij}$. In the subsequent $(n+j)$th computation for component $\left(\frac{\lambda}{2}\sum_k z_{kj} + \frac{d_{ij}}{2} - (N-1)D\right)x_{ij} + (1-t)z_{ij}$, resistors in node i are set as $R_i^1 = \frac{R_0}{2t\gamma x_i}, R_i^2 = \frac{R_0}{t\gamma x_i}, R_i^3 = \frac{R_0}{(1-t)}$. Input voltages are set to $V_i^{i1} = z_{ij}, V_i^{i2} = \frac{d_{ij}}{2} - (N-1)D, V_i^{i3} = z_{ij}$, producing output voltage $\boldsymbol{V}_{n+j,i}^{o2} = t\gamma\left(\frac{\lambda}{2}\sum_k z_{kj} + \frac{d_{ij}}{2} - (N-1)D\right)x_{ij} + (1-t)z_{ij}$ The combined output $\frac{\partial C(t,\boldsymbol{\theta})}{\partial\theta_{ij}} = \boldsymbol{V}_{i,j}^{o2} + \boldsymbol{V}_{n+j,i}^{o2}$ matches Eq. (27). These voltages are recorded via ADC, after which the processor executes gradient descent and project–feedback strategy.

Details for experimental implementation

We utilize EasyEDA software to design the PCBs, which are fabricated at a local foundry. The precision chip resistors used in the circuit are manufactured by RESI Company, featuring a 0603 package with a tolerance of 1%. The precision operational amplifier (model AD8676ARZ-REEL7) operates at ±18V and features a low input bias current (Ib = 2 nA), a low input offset voltage (Vos = 12 μV), and a low input offset current (Ios = 1 nA), all of which have negligible impact on measurement results. We select a 1-kΩ grounding resistor for the non-inverting input terminal. The resistance values of all resistors connected between the inverting input terminal and the input voltage, as well as the feedback resistor R_0, are 1 kΩ. The potentiometer (model 3296W-1-103-8mm) provides an adjustable resistance range of 0 to 10 kΩ.

The system power supply is provided by multiple programmable DC voltage sources of model UDP3305E, which can output a DC voltage ranging from −30 V to +30 V with a resolution of 10 mV. The PCB is connected to the voltage source via a cable with 4-mm lantern plugs on one end and alligator clips on the other.

The measured quantities of our circuit are the output voltages corresponding to each set of input voltages. In our experiment, we use a multimeter (model VC9810A, 1-mV resolution) to collect the output voltage data.

Acknowledgments

Funding: This work was supported by the National Key R&D Program of China (grant no. 2022YFA1404900) and the National Natural Science Foundation of China (12234004).
Author contributions: H.C. provided the theoretical analysis with the help of T.C. S.D. performed the experiments and analyzed the data. X.Z. initiated and designed this research project.
Competing interests: The authors declare that they have no competing interests.

Data Availability

The data that underlie the plots within this paper and other findings of this study are available from the corresponding authors on reasonable request. A demonstration code of QIFA is available at the GitHub repository: https://github.com/crypsis98/QIFA. Additional codes related to this study are available from the corresponding author upon reasonable request.

Supplementary Materials

Supplementary Information S1 to S10
Figs. S1 to S6
Tables S1 and S2

References

1. Papadimitriou CH, Steiglitz K. *Combinatorial optimization: Algorithms and complexity*. Mineola (NY): Dover Publications; 1998.
2. Mohseni N, McMahon PL, Byrnes T. Ising machines as hardware solvers of combinatorial optimization problems. *Nat Rev Phys*. 2022;4(6):363–379.
3. Zhan S, Lin J, Zhang Z, Zhong Y. List-based simulated annealing algorithm for traveling salesman problem. *Comput Intell Neurosci*. 2016;Article 1712630.
4. Stützle T, Hoos HH. MAX–MIN ant system. *Futur Gener Comput Syst*. 2000;16:889–914.
5. Garg H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl Math Comput*. 2016;274: 292–305.
6. Zhou AH, Zhu LP, Hu B, Deng S, Song Y, Qiu H, Pan S. Traveling-salesman-problem algorithm based on simulated annealing and gene-expression programming. *Information*. 2018;10(1):7.
7. Skinderowicz R. Improving ant colony optimization efficiency for solving large TSP instances. *Appl Soft Comput*. 2022;120:Article 108653.
8. Lê-Huu D, Paragios N, Continuous relaxation of MAP inference: A nonconvex perspective. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City (UT): IEEE; 2018. p. 5533–5541.
9. Schuetz MJA, Brubaker JK, Katzgraber HG. Combinatorial optimization with physics-inspired graph neural networks. *Nat Mach Intell*. 2022;4(4):367–377.
10. Hong MC, Cho L-C, Lin C-S, Lin Y-H, Chen P-A, Wang I-T, Tzeng P-J, Sheu SS, Lo WC, Wu C-I, et al. In-memory annealing unit (IMAU): Energy-efficient (2000 TOPS/W) combinatorial optimizer for solving travelling salesman problem. In: *2021 IEEE International Electron Devices Meeting (IEDM)*. San Francisco (CA): IEEE; 2021. p. 21.3.1–21.3.4.

11. Jiang M, Shan K, He C, Li C. Efficient combinatorial optimization by quantum-inspired parallel annealing in analogue memristor crossbar. *Nat Commun*. 2023;14(1):5927.
12. Yue W, Zhang T, Jing Z, Wu K, Yang Y, Yang Z, Wu Y, Bu W, Zheng K, Kang J, et al. A scalable universal Ising machine based on interaction-centric storage and compute-in-memory. *Nat. Electron.* 2024;7(10):904–913.
13. Moy W, Ahmed I, Chiu PW, Moy J, Sapatnekar SS, Kim CH. A 1,968-node coupled ring oscillator circuit for combinatorial optimization problem solving. *Nat Electron*. 2022;5(5):310–317.
14. Lo H, Moy W, Yu H, Sapatnekar S, Kim CH. An Ising solver chip based on coupled ring oscillators with a 48-node all-to-all connected array architecture. *Nat Electron*. 2023;6(10):771–778.
15. McMahon PL, Marandi A, Haribara Y, Hamerly R, Langrock C, Tamate S, Inagaki T, Takesue H, Utsunomiya S, Aihara K, et al. A fully programmable 100-spin coherent Ising machine with all-to-all connections. *Science*. 2016;354(6312):614–617.
16. Yamamoto Y, Leleu T, Ganguli S, Mabuchi H. Coherent Ising machines—Quantum optics and neural network perspectives. *Appl Phys Lett*. 2020;117(16):Article 160501.
17. Inagaki T, Haribara Y, Igarashi K, Sonobe T, Tamate S, Honjo T, Marandi A, McMahon PL, Umeki T, Enbutsu K, et al. A coherent Ising machine for 2000-node optimization problems. *Science*. 2016;354(6312):603–606.
18. Shen ZS, Pan F, Wang Y, Men YD, Xu WB, Yung MH, Zhang P. Free-energy machine for combinatorial optimization. *Nat Comput Sci*. 2025;5(4):322–332.
19. Si J, Yang S, Cen Y, Chen J, Huang Y, Yao Z, Kim DJ, Cai K, Yoo J, Fong X, et al. Energy-efficient superparamagnetic Ising machine and its application to traveling salesman problems. *Nat Commun*. 2024;15(1):3457.
20. Skolik A, Cattelan M, Yarkoni S, Bäck T, Dunjko V. Equivariant quantum circuits for learning on weighted graphs. *npj Quantum Inf*. 2023;9(1):47.
21. Marsh S, Wang JB. Combinatorial optimization via highly efficient quantum walks. *Phys Rev Res*. 2020;2(2):Article 023302.
22. Blekos K, Brand D, Ceschini A, Chou CH, Li RH, Pandya K, Summer A. A review on quantum approximate optimization algorithm and its variants. *Phys Rep*. 2024;1068:1–66.
23. Zhou L, Wang ST, Choi S, Pichler H, Lukin MD. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Phys Rev X*. 2020;10(2):Article 021067.
24. Pirnay N, Ulitzsch V, Wilde F, Eisert J, Seifert JP. An in-principle super-polynomial quantum advantage for approximating combinatorial optimization problems via computational learning theory. *Sci Adv*. 2024;10(11):Article eadj5170.
25. Dupont M, Evert B, Hodson MJ, Sundar B, Jeffrey S, Yamaguchi Y, Feng D, Maciejewski FB, Hadfield S, Alam MS, et al. Quantum-enhanced greedy combinatorial optimization solver. *Sci Adv*. 2023;9(45):Article eadi0487.
26. Zhou M, Liu Z, Yin H, Li CL, Xu TK, Chen ZB. Quantum neural network for quantum neural computing. *Research*. 2023;6:0134.
27. Weng C, Gao R, Bao Y, Li BH, Liu WB, Xie YM, Lu YS, Yin HL, Chen ZB. Beating the fault-tolerance bound and security loopholes for byzantine agreement with a quantum solution. *Research*. 2023;6:0272.
28. Hamerly R, Inagaki T, McMahon PL, Venturelli D, Marandi A, Onodera T, Ng E, Langrock C, Inaba K, Honjo T, et al. Experimental investigation of performance differences between coherent Ising machines and a quantum annealer. *Sci Adv*. 2019;5:Article eaau0823.
29. Feld S, Roch C, Gabor T, Seidel C, Neukart F, Galter I, Mauerer W, Linnhoff-Popien C. A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. *Front ICT*. 2019;6:13.
30. Silva C, Aguiar A, Lima PMV, Dutra I. Mapping a logical representation of TSP to quantum annealing. *Quantum Inf Process*. 2021;20(12):386.
31. Kim S, Ahn SW, Suh IS, Dowling AW, Lee E, Luo T. Quantum annealing for combinatorial optimization: A benchmarking study. *npj Quantum Inf*. 2025;11(1):77.
32. Goto H, Tatsumura K, Dixon AR. Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems. *Sci Adv*. 2019;5(4):eaav2372.
33. Goto H, Endo K, Suzuki M, Sakai Y, Kanao T, Hamakawa Y, Hidaka R, Yamasaki M, Tatsumura K. High-performance combinatorial optimization based on classical mechanics. *Sci Adv*. 2021;7(6):eabe7953.
34. Chia NH, Li T, Li HH, Wang C. Quantum-inspired sublinear algorithm for solving low-rank semidefinite programming. arXiv. 2020. https://doi.org/10.48550/arXiv.1901.03254
35. Bowles J, Dauphin A, Huembeli P, Martinez J, Acín A. Quadratic unconstrained binary optimization via quantum-inspired annealing. *Phys Rev Appl*. 2022;18: Article 034016.
36. Lucas A. Ising formulations of many NP problems. *Front Phys*. 2014;2.
37. Reinelt G. TSPLIB—A traveling salesman problem library. *ORSA J Comput*. 1991;3:376–384.
38. Bao J, Zou D, Zhang W, He W, Sun H, Zhang X. Topoelectrical circuit octupole insulator with topologically protected corner states. *Phys Rev B*. 2019;100:Article 201406.
39. Helbig T, Hofmann T, Imhof S, Abdelghany M, Kiessling T, Molenkamp LW, Lee CH, Szameit A, Greiter M, Thomale R. Generalized bulk–boundary correspondence in non-Hermitian topolectrical circuits. *Nat Phys*. 2020;16: 747–750.
40. Hofmann T, Helbig T, Lee CH, Greiter M, Thomale R. Chiral voltage propagation and calibration in a topolectrical Chern circuit. *Phys Rev Lett*. 2019;122:Article 274702.
41. Imhof S, Berger C, Bayer F, Brehm J, Molenkamp LW, Kiessling T, Schindler F, Lee CH, Greiter M, Neupert T, et al. Topolectrical-circuit realization of topological corner modes. *Nat Phys*. 2018;14:925–929.
42. Lee CH, Imhof S, Berger C, Bayer F, Brehm J, Molenkamp LW, Kiessling T, Thomale R. Topolectrical circuits. *Commun Phys*. 2018;1:39.
43. Liu S, Gao W, Zhang Q, Ma S, Zhang L, Liu C, Xiang YJ, Cui TJ, Zhang S. Topologically protected edge state in two-dimensional Su–Schrieffer–Heeger circuit. *Research*. 2019;2019:Article 8609875.
44. Liu S, Shao R, Ma S, Zhang L, You O, Wu H, Xiang YJ, Cui TJ, Zhang S. Non-Hermitian skin effect in a non-Hermitian electrical circuit. *Research*. 2021;Article 5608038.
45. Luo K, Yu R, Weng H. Topological nodal states in circuit lattice. *Research*. 2018;Article 6793752.
46. Ningyuan J, Owens C, Sommer A, Schuster D, Simon J. Time- and site-resolved dynamics in a topological circuit. *Phys Rev X*. 2015;5:Article 021031.
47. Olekhno NA, Kretov EI, Stepanenko AA, Ivanova PA, Yaroshenko VV, Puhtina EM, Filonov DS, Cappello B,

Matekovits L, Gorlach MA. Topological edge states of interacting photon pairs emulated in a topolectrical circuit. *Nat Commun*. 2020;11:1436.
48. Serra-Garcia M, Süsstrunk R, Huber SD. Observation of quadrupole transitions and edge mode topology in an LC circuit network. *Phys Rev B*. 2019;99:Article 020304.
49. Wang Y, Price HM, Zhang B, Chong YD. Circuit implementation of a four-dimensional topological insulator. *Nat Commun*. 2020;11:Article 2356.
50. Zhang W, Zou D, Bao J, He W, Pei Q, Sun H, Zhang X. Topolectrical-circuit realization of a four-dimensional hexadecapole insulator. *Phys Rev B*. 2020;102:Article 100102.
51. Zhang W, Zou D, Pei Q, He W, Bao J, Sun H, Zhang X. Experimental observation of higher-order topological Anderson insulators. *Phys Rev Lett*. 2021;126:Article 146802.
52. Zhang J, Chen T, Deng W, Tong X, Zhang X. Quantum 2-player games and realizations with circuits. *Research*. 2024;7:0480.
53. Zhang H, Chen T, Pan N, Zhang X. Electric-circuit simulation of quantum fast hitting with exponential speedup. *Adv Quantum Technol*. 2022;5(4):2100143.
54. Ezawa M. Electric-circuit simulation of the Schrödinger equation and non-Hermitian quantum walks. *Phys Rev B*. 2019;100.
55. Pan N, Chen T, Sun H, Zhang X. Electric-circuit realization of fast quantum search. *Research*. 2021;2021:9793071.
56. Saini R, Mani A, Prasad M, Bhattacharyya S, Platos J. Evolutionary algorithm for the traveling salesman problem with innovative encoding on hybrid quantum-classical machines. *IEEE Access*. 2025;13:54223–54239.
57. Xin L, Song W, Cao Z, Zhang J. Neuro LKH: Combining deep learning model with Lin–Kernighan–Helsgaun heuristic for solving the traveling salesman problem. *Proc. Adv. Neural Inf. Process. Syst*. 2021:7472–7483.
58. Liu F, Zhang Q, Zhu Q, Tong X, Yuan M. Machine learning assisted multiobjective evolutionary algorithm for routing and packing. *IEEE Trans Evol Comput*. 2024;29(2):317–330.