Multi-Objective and Quality-Diversity Optimization of Quantum Circuits via Statevector Simulation

Thomas Snell

February 8, 2026

Abstract

Automated quantum circuit synthesis—the task of discovering a gate sequence that realizes a desired unitary transformation—is a combinatorial optimization problem whose difficulty grows exponentially with qubit count. We present a systematic empirical comparison of six optimization strategies applied to this problem: random search, an evolutionary algorithm (EA) based on DEAP, a gradient-based optimizer using continuous relaxation with L-BFGS-B, a deep reinforcement learning (RL) agent using REINFORCE, and two quality-diversity (QD) methods—NSGA-II multi-objective optimization and MAP-Elites. All methods share a common evaluation interface normalized by *fitness-function evaluation budget*, enabling fair comparison. We benchmark across six quantum computing problems on a 3-qubit system with 10 000 evaluations per trial and 10 independent trials per configuration. The REINFORCE-based deep learning optimizer and MAP-Elites achieve the highest mean fitness, both reaching perfect solutions (fitness 1.0) on multiple problems. MAP-Elites additionally discovers structurally diverse circuit portfolios with 79–87% archive coverage, while NSGA-II produces compact Pareto-optimal circuits that trade fidelity for reduced depth and gate count. We introduce QD-specific metrics—archive coverage, QD-score, and Pareto front analysis—to characterize the diversity of discovered circuits, an aspect overlooked by prior single-objective studies.

1 Introduction

Quantum circuit synthesis is the problem of determining a sequence of elementary quantum gates that implements a target unitary operation or prepares a target output distribution from a given input state Shende et al. [2006], Dawson and Nielsen [2005]. It is a foundational subroutine in quantum compilation, and its difficulty scales combinatorially with the number of qubits, the depth of the circuit, and the size of the gate set.

Metaheuristic search methods—particularly evolutionary algorithms (EAs)—have been applied to this problem with encouraging results Williams and Gray [1998], Massey and Clark [2005], Las Heras et al. [2016]. More recently, reinforcement learning has emerged as a powerful alternative: Rietsch et al. [2024] demonstrated that deep RL with Gumbel AlphaZero tree search can synthesize unitaries over the Clifford+T gate set, achieving competitive results for fault-tolerant circuit compilation. Meanwhile, Sun et al. [2026] showed that unsupervised representation learning can dramatically reduce the cost of quantum architecture search by decoupling circuit encoding from the search process itself. Despite these individual advances, systematic comparisons between EAs and other optimization paradigms (gradient-based methods, reinforcement learning) under controlled experimental conditions remain scarce. Sharma and Lau [2025] recently highlighted this gap, noting a lack of standardized benchmarking protocols for quantum optimization methods.

A further limitation of prior work is its exclusive focus on single-objective optimization—maximizing fidelity alone. In practice, quantum circuit designers must balance multiple competing objectives: fidelity, circuit depth (which affects decoherence), gate count (which affects error accumulation), and structural diversity (which provides alternative implementations for different hardware constraints). Quality-diversity (QD) algorithms such as MAP-Elites Mouret and Clune [2015] and multi-objective methods such as NSGA-II Deb et al. [2002] are designed precisely for this setting, yet their application to quantum circuit synthesis remains largely unexplored.

In this work we address both gaps. We design a unified benchmarking framework with budget-normalized evaluation and extend it with two QD methods.

Contributions.

- A modular open-source framework comparing six quantum circuit optimizers, built on a fast NumPy statevector simulator.
- Six optimizer implementations sharing a common interface: random search, DEAP-based EA, continuous-relaxation gradient descent, REINFORCE policy gradient, NSGA-II, and MAP-Elites.
- A benchmark suite of six problems spanning search, arithmetic, transform, and oracle-identification tasks.
- An empirical study with 10 trials per condition ($6 \times 6 \times 10 = 360$ runs), reporting fitness, convergence speed, wall-clock time, circuit complexity, and QD-specific metrics (archive coverage, QD-score, Pareto fronts).

2 Background

2.1 Quantum Circuit Model

We consider circuits on n qubits composed of gates drawn from the set $\mathcal{G} = \{\texttt{I}, \texttt{T}, \texttt{H}, \texttt{CNOT}\downarrow, \texttt{CNOT}\uparrow\}$, where `I` is the identity (no-op), `T` is the $\pi/8$ phase gate, `H` is the Hadamard gate, and the two CNOT variants act on adjacent qubits in opposite directions. A circuit of depth d is represented by an integer matrix $G \in \{0, \dots, 4\}^{d \times n}$, where entry $G_{t,q}$ specifies the gate applied to qubit q at time step t. A preprocessing pass enforces CNOT adjacency constraints and removes self-cancelling consecutive gate pairs.

2.2 Statevector Simulation

Each candidate circuit is evaluated by statevector simulation. Given an input state $|\psi_0\rangle$ (typically the uniform superposition obtained from $|0\rangle^{\otimes n}$), the simulator applies each gate row in sequence via matrix–vector multiplication:

$$|\psi_T\rangle = \prod_{t=1}^{d} \prod_{q=0}^{n-1} U^{(q)}_{G_{t,q}} \, |\psi_0\rangle, \tag{1}$$

where $U^{(q)}_g$ is the $2^n \times 2^n$ matrix embedding gate g on qubit q. Gate matrices are cached in a dictionary keyed by (g, q, n) for reuse. Measurement probabilities are $p_i = |\langle i|\psi_T\rangle|^2$.

2.3 Fitness Function

Fitness is defined as the probability of the target outcome:

$$f(G) = p_{\text{target}}(G), \tag{2}$$

where the target index is problem-dependent (e.g., the marked item in Grover's search). A penalty discourages wasteful circuits: if $f > 0.99$, the fitness is divided by $(1 + 0.1 \cdot b)$ where b is the number of all-identity rows. Fitness values are cached keyed by the byte representation of the gate array concatenated with input/target data.

3 Optimization Methods

All six optimizers implement a common abstract interface:

$$\texttt{optimize(input\_set, target\_set, num\_qubits,} \\ \texttt{time\_steps, evaluation\_budget, seed)} \\ \longrightarrow \texttt{OptimizationResult}$$

returning the best gate array found, its fitness, the number of evaluations consumed, wall-clock time, circuit complexity (count of non-identity gates), and a fitness history trace.

3.1 Random Search (RS)

The simplest baseline: at each of the B budget steps, sample a gate array uniformly at random from $\{0, \dots, 4\}^{d \times n}$, preprocess it, evaluate fitness, and retain the best solution found so far. Random search provides a lower bound on performance and an upper bound on exploration breadth.

3.2 Evolutionary Algorithm (EA)

We use DEAP's `eaSimple` algorithm with tournament selection, two-point crossover, and bit-flip mutation. Individuals are flat integer lists of length $d \cdot n$ that are reshaped into gate arrays for evaluation. The number of generations is derived as $\lfloor B/P \rfloor - 1$, where P is population size and B is the evaluation budget, so total evaluations are approximately B. We use P=50, crossover probability 0.5, mutation probability 0.2, tournament size 3, and per-gene flip probability 0.1.

3.3 Gradient-Based Optimization (Grad)

To apply gradient-based optimization to the discrete gate selection problem, we introduce a *continuous relaxation*. Each gate position (t, q) is parameterized by a logit vector $\boldsymbol{\ell}_{t,q} \in \mathbb{R}^{|\mathcal{G}|}$. A softmax produces mixture weights:

$$w_g = \frac{\exp(\ell_g)}{\sum_{g'} \exp(\ell_{g'})}, \tag{3}$$

and the effective gate matrix at position (t, q) is the convex combination $\tilde{U}_{t,q} = \sum_{g \in \mathcal{G}} w_g \, U_g^{(q)}$. The relaxed fitness is differentiable with respect to the logits, enabling optimization via L-BFGS-B (minimizing $-f$). After convergence, the discrete solution is obtained by taking $\arg\max_g \ell_g$ at each position. We run R=3 random restarts, splitting the budget evenly.

3.4 Deep Learning / REINFORCE (DL)

Since the true fitness involves discrete gate sampling and is non-differentiable, we apply the REINFORCE policy-gradient estimator Williams [1992]. A feedforward neural network π_θ maps the target distribution $\mathbf{t} \in \mathbb{R}^{2^n}$ to logits over $|\mathcal{G}|$ gate types at each of the $d \cdot n$ positions. Gate arrays are sampled from the resulting categorical distributions, evaluated, and the policy is updated with:

$$\nabla_\theta J \approx \frac{1}{K} \sum_{k=1}^{K} (f_k - \bar{f}) \, \nabla_\theta \log \pi_\theta(\mathbf{a}_k), \tag{4}$$

where $\bar{f}$ is an exponential moving average baseline (decay 0.9) and K is the batch size. The network has two hidden layers of 32 units with ReLU activations, trained with Adam (lr $= 10^{-3}$).

3.5 NSGA-II Multi-Objective Optimization

NSGA-II Deb et al. [2002] optimizes three objectives simultaneously:

1. **Maximize fidelity**: target probability $f(G)$.
2. **Minimize active depth**: count of time steps containing at least one non-identity gate.
3. **Minimize gate count**: total non-identity gates.

We use DEAP's $(\mu + \lambda)$ scheme with `selNSGA2` for non-dominated sorting and crowding-distance tie-breaking. The individual representation, crossover, and mutation operators are identical to the single-objective EA. For backward compatibility with the study framework, `optimize()` returns the Pareto-front member with highest fidelity. The full Pareto front is stored for post-hoc analysis.

Table 1: Benchmark problems. The target column describes the desired output distribution. Classical complexity refers to the best classical algorithm for the equivalent task.

Problem	Target	Steps	Classical
Grover	$P(\text{marked}) = 1$	15	$O(N)$
Flip	Bitwise NOT	9	$O(N)$
Inverse	$1/x$ mapping	12	$O(N)$
Fourier	DFT	12	$O(N \log N)$
Deutsch–Jozsa	$P(\lvert 000\rangle) = 0$	12	$O(N/2{+}1)$
Bernstein–Vaz.	$P(\lvert s\rangle) = 1$	12	$O(N)$

3.6 MAP-Elites Quality-Diversity Optimization

MAP-Elites Mouret and Clune [2015] maintains a 2D grid archive indexed by two *behavioral descriptors*:

- **Active depth**: number of time steps with ≥ 1 non-identity gate, discretized into 10 bins.
- **Entanglement density**: ratio of CNOT gates to total non-identity gates, discretized into 10 bins.

Each cell stores the highest-fitness circuit mapping to that (depth, density) region. The algorithm proceeds in two phases: (1) random seeding fills initial cells, then (2) a mutation loop picks a random occupied cell, mutates its circuit (random gene flips at rate 0.15), evaluates the child, and places it in the archive if its cell is empty or the child has higher fitness. No external library is used; the implementation is ∼80 lines of core logic. For the study framework, `optimize()` returns the archive member with highest fidelity.

4 Benchmark Problems

We evaluate on six problems, summarized in Table 1. All operate on n=3 qubits (2^n=8 basis states).

Grover's Search. The input is the all-zeros state (mapped to uniform superposition by the simulator). The target distribution places all probability mass

Table 2: Optimizer hyperparameters.

Optimizer	**Key Parameters**
Random Search	—
EA	P=50, p_{cx}=0.5, p_{mut}=0.2, T_s=3
Gradient	3 restarts, L-BFGS-B
DL/REINFORCE	batch 8, hidden 32, lr 10^{-3}, Adam
NSGA-II	P=50, p_{cx}=0.5, p_{mut}=0.2, $(\mu+\lambda)$
MAP-Elites	10×10 grid, 100 seeds, mut. rate 0.15

on a single marked item. The classical equivalent is linear search, requiring $O(N)$ queries on average.

Bitwise Flip. Given discrete binary input states, the target is the bitwise complement.

Modular Inverse. Given continuous-valued input amplitudes, the target amplitudes are $1/x$ (with $1/0 \mapsto 0$), normalized.

Quantum Fourier Transform. The target is the discrete Fourier transform of the input amplitudes.

Deutsch–Jozsa. For a balanced Boolean function, the quantum algorithm should produce zero probability on $|0\ldots0\rangle$. We set the target to uniform probability over all non-zero basis states.

Bernstein–Vazirani. The quantum algorithm should recover a hidden bit string s by concentrating all probability on $|s\rangle$. We use $s = 6$ (binary `110`) for n=3.

5 Experimental Setup

Configuration. All experiments use n=3 qubits, an evaluation budget of B=10,000, and T=10 independent trials per (problem, optimizer) pair. Each trial is seeded deterministically (seed $= 42 + t$ for trial t) and the fitness cache is cleared between trials. The total study comprises $6 \times 6 \times 10 = 360$ optimization runs.

Optimizer Hyperparameters. Table 2 lists the hyperparameters used for each optimizer.

Platform. All experiments run on a single CPU thread using a pure NumPy statevector simulator (no GPU, no Qiskit runtime dependency). PyTorch (CPU-only) is used for the DL optimizer.

Metrics. We report standard metrics per (problem, optimizer) pair, aggregated over 10 trials:

1. **Fitness**: probability of the target outcome, $f \in [0, 1]$.
2. **Convergence speed**: the evaluation index at which the running best fitness first exceeds 95% of the final value.
3. **Wall-clock time**: total seconds elapsed.
4. **Circuit complexity**: number of non-identity gates in the best solution.

For the QD methods, we additionally report:

5. **Archive coverage** (MAP-Elites): percentage of grid cells filled.
6. **QD-score** (MAP-Elites): sum of fitness values across all occupied cells.
7. **Pareto front** (NSGA-II): set of non-dominated solutions in (fidelity, depth, gate count) space.

6 Results

6.1 Fitness

Table 3 reports the mean fitness (± one standard deviation) across 10 trials for each optimizer–problem pair. Figure 1 shows the corresponding bar chart.

DL/REINFORCE achieves the highest or tied-highest mean fitness on five of six problems, reaching perfect fitness (1.0) on Inverse, Fourier, and Deutsch–Jozsa. MAP-Elites matches DL on Inverse and Fourier while achieving the best result on Bernstein–Vazirani (0.991). The EA achieves the best fitness on Grover (0.967) due to its rapid convergence exploiting crossover building blocks. NSGA-II's fidelity is lower overall because its multi-objective pressure distributes the population across the Pareto front rather than concentrating on the highest-fidelity region.

6.2 Convergence

Figure 2 shows the mean convergence curves (best fitness vs. evaluation count) averaged over 10 trials.

The EA converges extremely rapidly—typically within the first 5–70 evaluations—but often to a suboptimal solution, explaining its high variance on several problems. MAP-Elites converges more

Table 3: Mean fitness ± std over 10 trials. Bold indicates the best mean per problem.

Problem	RS	EA	Grad	DL	NSGA-II	MAP-Elites
Grover	0.833 ± 0.077	**0.967 ± 0.052**	0.000 ± 0.000	0.938 ± 0.065	0.768 ± 0.180	0.887 ± 0.027
Flip	0.747 ± 0.162	0.500 ± 0.000	0.050 ± 0.100	**0.985 ± 0.044**	0.535 ± 0.106	0.721 ± 0.226
Inverse	0.885 ± 0.046	0.673 ± 0.224	0.300 ± 0.245	**1.000 ± 0.000**	0.650 ± 0.229	**1.000 ± 0.000**
Fourier	**1.000 ± 0.000**	**1.000 ± 0.000**	0.523 ± 0.031	**1.000 ± 0.000**	0.985 ± 0.044	**1.000 ± 0.000**
Deutsch–Jozsa	0.932 ± 0.069	0.946 ± 0.083	0.125 ± 0.202	**1.000 ± 0.000**	0.797 ± 0.155	0.982 ± 0.036
Bernstein–Vaz.	0.909 ± 0.063	0.724 ± 0.229	0.000 ± 0.000	0.982 ± 0.036	0.691 ± 0.196	**0.991 ± 0.027**

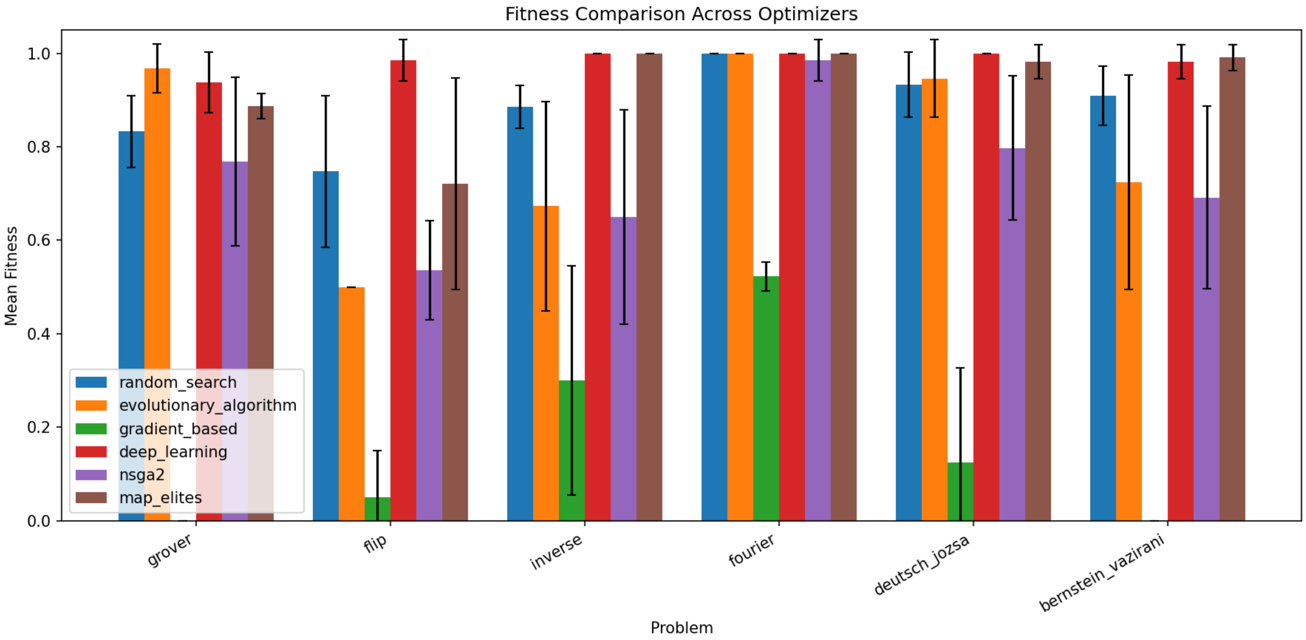

Figure 1: Mean fitness (± 1 std) across 10 trials for each optimizer and problem. DL/REINFORCE and MAP-Elites consistently achieve the highest fitness. NSGA-II trades fidelity for circuit compactness.

slowly but continues improving throughout the budget, benefiting from the diversity of parent circuits in its archive. The DL optimizer exhibits a learning plateau followed by rapid improvement around evaluations 500–800.

6.3 Wall-Clock Time

Figure 3 reports wall-clock times.

The EA is consistently the fastest optimizer (≈2.3–3.8 s), followed by NSGA-II (≈2.5–4.3 s), random search (≈3.7–7.1 s), MAP-Elites (≈4.3–6.6 s), DL (≈9.2–14.1 s), and the gradient optimizer (≈8.6–19.6 s).

6.4 Circuit Complexity

Figure 4 shows the circuit complexity (non-identity gate count) of the best solutions.

NSGA-II produces notably simpler circuits (mean 2.6–11.1 gates) compared to the other methods (mean 10–22 gates). This is a direct consequence of its gate-count minimization objective. The gradient optimizer also produces sparse circuits (2.5–9.4 gates) but for a different reason: most logit positions converge to identity during continuous optimization.

6.5 Quality-Diversity Analysis

MAP-Elites Archive. Figure 5 shows the archive heatmaps for each problem. Table 4 reports archive coverage, QD-score, and best fitness.

MAP-Elites achieves 79–87% archive coverage, in-

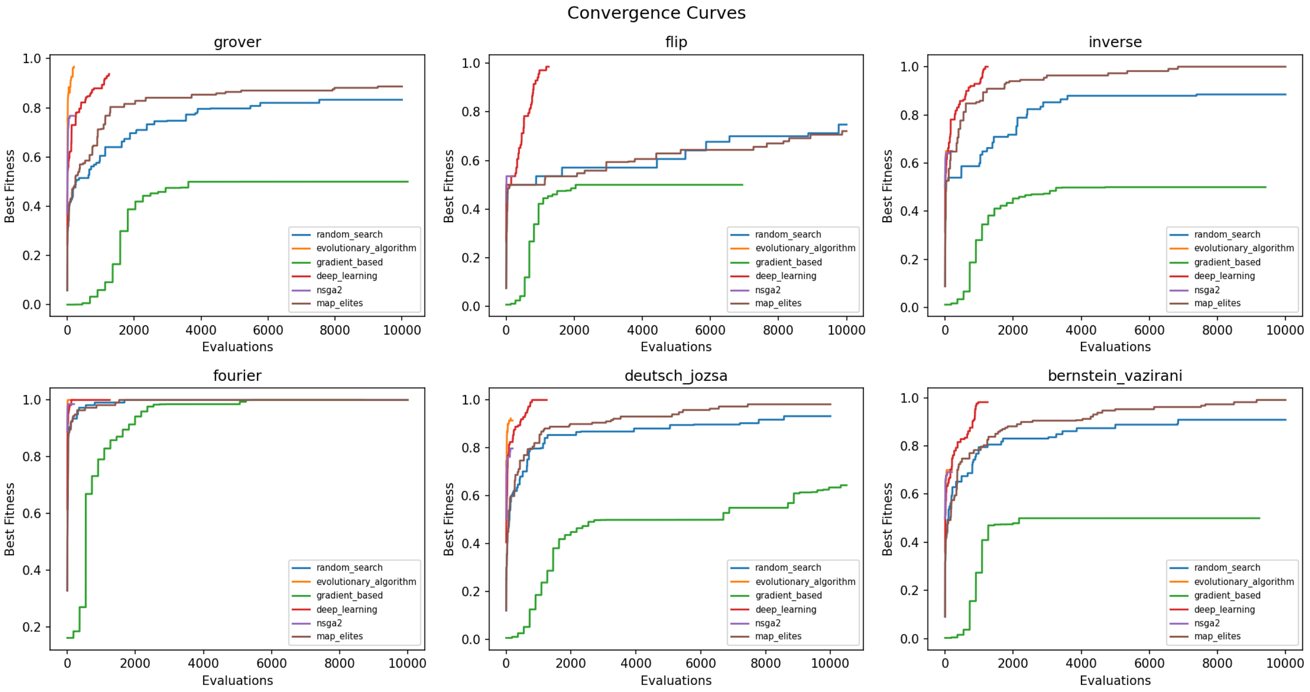

Figure 2: Mean convergence curves across 10 trials. The EA converges fastest but often to a suboptimal solution. MAP-Elites improves steadily throughout the full budget as archive exploration continues.

Table 4: MAP-Elites archive statistics (10×10 grid, last trial).

Problem	**Coverage**	**QD-Score**	**Best** f
Grover	87%	36.5	0.854
Flip	79%	60.1	1.000
Inverse	83%	46.0	1.000
Fourier	84%	65.5	1.000
D–J	84%	48.9	1.000
B–V	86%	43.9	1.000

dicating that the algorithm discovers high-fitness circuits across a wide range of structural configurations. The QD-score (sum of fitnesses across all occupied cells) ranges from 36.5 (Grover) to 65.5 (Fourier), with higher values indicating problems where high fitness is achievable across more of the depth×density space. Fourier has the highest QD-score because many different circuit structures can implement the DFT; Grover has the lowest because the precise interference pattern required constrains viable architectures.

NSGA-II Pareto Fronts. Figure 6 shows the Pareto fronts in (depth, fidelity) space.

The Pareto fronts reveal problem-specific tradeoffs. For Flip, NSGA-II finds circuits achieving perfect fidelity (1.0) at depth 9. For Grover, the front converges to a single depth–fidelity point (0.854 at depth 11), suggesting that reducing depth below this threshold causes a sharp fidelity cliff. For Bernstein–Vazirani, the front includes low-depth solutions (depth 3) at fidelity 0.50 and higher-depth solutions approaching 0.50, indicating that the multi-objective pressure successfully discovered minimal-depth circuits even when they could not reach high fidelity.

7 Discussion

7.1 MAP-Elites as a Practical Circuit Discovery Tool

MAP-Elites emerges as a surprisingly effective optimizer for quantum circuit synthesis, matching or approaching the best single-objective methods in peak fitness while additionally providing a diverse portfolio of circuits. Its archive-based exploration avoids the premature convergence that affects the EA, as parents are drawn from structurally diverse cells rather than a converging population. The high archive coverage (79–87%) demonstrates that viable

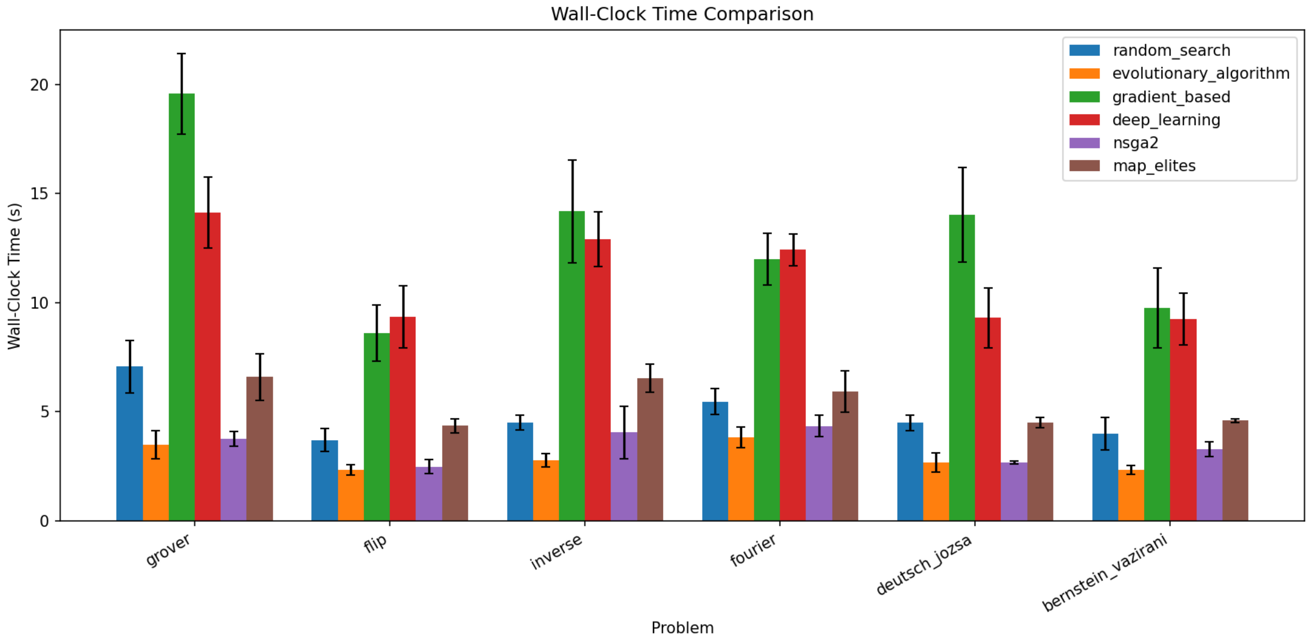

Figure 3: Wall-clock time comparison. The EA is fastest. The gradient and DL optimizers are slowest due to L-BFGS-B restarts and PyTorch overhead, respectively.

circuits exist across a wide range of depth and entanglement configurations, a finding that would be invisible to single-objective optimization.

This diversity is practically valuable: hardware constraints may favor circuits with specific depth or entanglement characteristics, and a populated archive provides ready-made alternatives without re-running optimization.

7.2 NSGA-II: Compactness at a Cost

NSGA-II's multi-objective formulation successfully produces compact circuits (2.6–11.1 gates vs. 10–22 for other methods), but its fidelity suffers on harder problems (0.535 on Flip, 0.650 on Inverse). The three-way objective trade-off distributes selection pressure across the Pareto front, reducing the effective optimization effort on any single objective. For problems where compact circuits can achieve high fidelity (e.g., Fourier at 0.985), NSGA-II remains competitive.

7.3 The Strength of Random Search

Random search remains a competitive baseline at this scale, outperforming the EA on several problems (Flip, Inverse, Bernstein–Vazirani). With 10 000 independent uniform samples from the gate space, it covers the landscape more broadly than the EA's evolving population, which can converge prematurely.

7.4 Why Gradient Optimization Underperforms

The gradient-based optimizer's poor performance stems from the *discretization gap*: the soft mixture of gate matrices used during optimization does not correspond to any physically realizable circuit. When the continuous solution is discretized via $\arg\max$, the resulting discrete circuit may have drastically different behavior. This effect is severe on 3 qubits, where the gradient optimizer scores near-zero on Grover and Bernstein–Vazirani.

7.5 Threats to Validity

Several factors limit generalizability:

1. The 3-qubit setting is still small; scaling behavior may differ.
2. The gate set includes only five gate types.
3. The fitness function measures only single-outcome probability.
4. Hyperparameters were not extensively tuned.
5. The MAP-Elites behavioral descriptors (depth, entanglement density) were chosen heuristically;

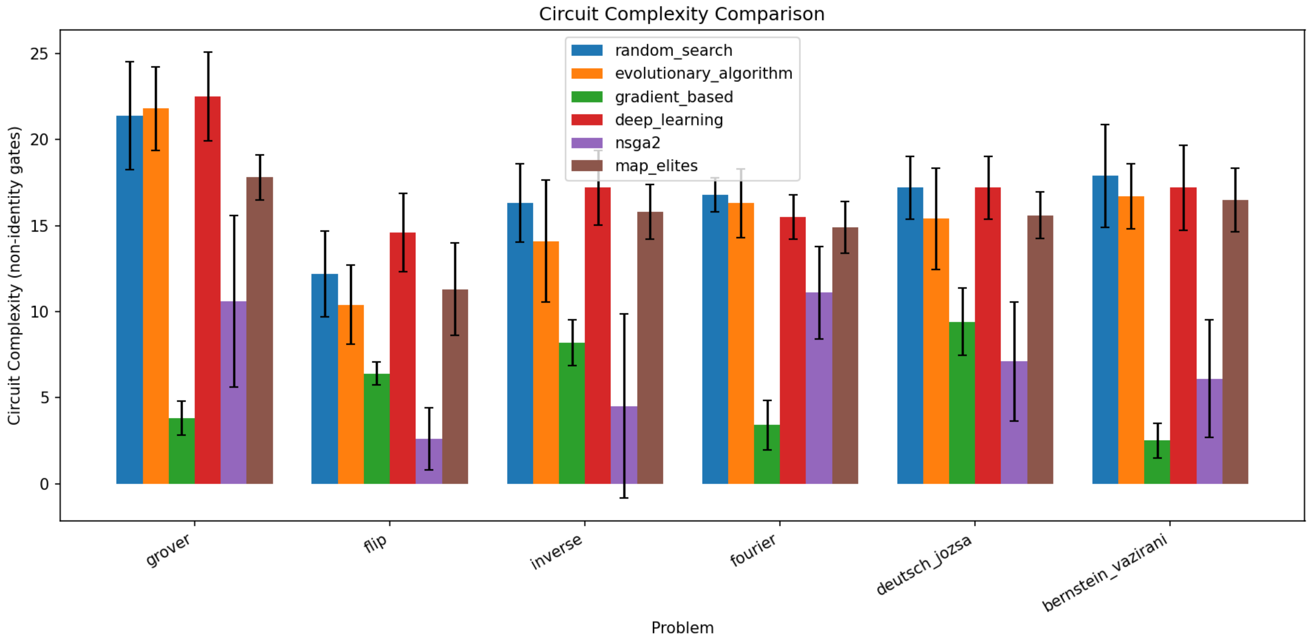

Figure 4: Circuit complexity (non-identity gates) of the best solutions found. NSGA-II produces markedly simpler circuits due to its gate-count minimization objective.

other descriptors may be more informative.

8 Related Work

Evolutionary approaches. Evolutionary approaches to quantum circuit synthesis date to Williams and Gray [1998], who evolved quantum logic circuits using genetic programming. Massey and Clark [2005] explored evolving quantum circuits for a broader class of problems. Las Heras et al. [2016] applied genetic algorithms to quantum circuit optimization with a focus on minimizing gate count.

Quality-diversity. Quality-diversity algorithms Mouret and Clune [2015], Pugh et al. [2016] have been widely applied in robotics and game design but remain largely unexplored for quantum circuit synthesis. Our work represents one of the first applications of MAP-Elites to this domain, demonstrating that the archive-based approach is well-suited to discovering structurally diverse quantum circuits.

Multi-objective quantum optimization. Multi-objective optimization of quantum circuits has received limited attention. Prior work has focused on bi-objective trade-offs (fidelity vs. depth) in variational circuits Cerezo et al. [2021], but systematic application of NSGA-II to discrete circuit synthesis with three objectives (fidelity, depth, gate count) is, to our knowledge, novel.

Reinforcement learning. Fösel et al. [2021] used deep RL to discover quantum error correction codes, and Moro et al. [2021] applied RL to quantum circuit optimization. Rietsch et al. [2024] applied Gumbel AlphaZero to synthesize unitaries over the Clifford+T gate set.

Architecture search. Sun et al. [2026] proposed a predictor-free framework for quantum architecture search that decouples unsupervised circuit representation learning from the search process.

Benchmarking. Sharma and Lau [2025] provided a systematic comparison of quantum optimization techniques, establishing standardized protocols for fair comparison. Our budget-normalized evaluation protocol follows the same philosophy.

9 Conclusion

We presented a controlled empirical comparison of six optimization methods for quantum circuit synthesis,

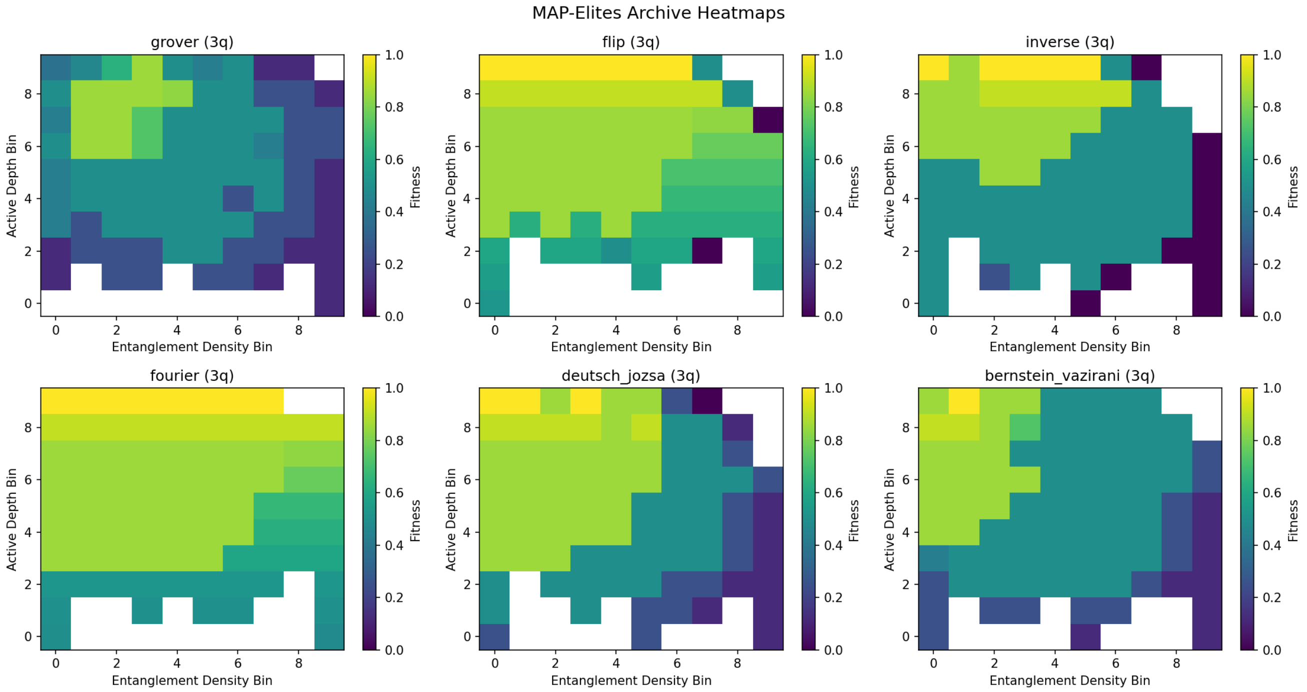

Figure 5: MAP-Elites archive heatmaps. Each cell shows the fitness of the best circuit at that (depth bin, entanglement density bin) coordinate. High coverage indicates that MAP-Elites discovers circuits across a wide range of structural configurations.

including two quality-diversity approaches, evaluated across six problems on a 3-qubit system. Our key findings are:

1. **DL/REINFORCE and MAP-Elites achieve the highest fitness**, both reaching perfect solutions on multiple problems. MAP-Elites matches DL on Inverse and Fourier while providing a diverse portfolio of structurally varied circuits.
2. **MAP-Elites discovers diverse circuit portfolios** with 79–87% archive coverage and QD-scores of 36–66, demonstrating that viable circuits exist across a wide range of depth and entanglement configurations.
3. **NSGA-II produces compact circuits** (2.6–11.1 gates) through explicit depth and gate-count minimization, at the cost of reduced fidelity on harder problems.
4. **The EA offers the best speed–quality trade-off**, converging rapidly at the lowest wall-clock cost.
5. **Gradient-based continuous relaxation consistently underperforms** due to the discretization gap.

These results demonstrate that quality-diversity methods are a promising and underexplored direction for quantum circuit synthesis. Future work should investigate scaling to larger qubit counts, alternative behavioral descriptors for MAP-Elites (e.g., gate-type distribution, symmetry measures), hybrid approaches combining MAP-Elites with learned circuit representations Sun et al. [2026], and integration with hardware-aware compilation to exploit the diverse circuit portfolios on real quantum processors.

References

C. P. Williams and A. G. Gray. Automated design of quantum circuits. In *Quantum Computing and Quantum Communications*, pp. 113–125. Springer, 1998.

P. Massey and J. A. Clark. Evolving quantum circuits. *Genetic Programming and Evolvable Machines*, 6(4):415–449, 2005.

U. Las Heras, U. Alvarez-Rodriguez, E. Solano, and M. Sanz. Genetic algorithms for digital quantum simulations. *Physical Review Letters*, 116(23):230504, 2016.

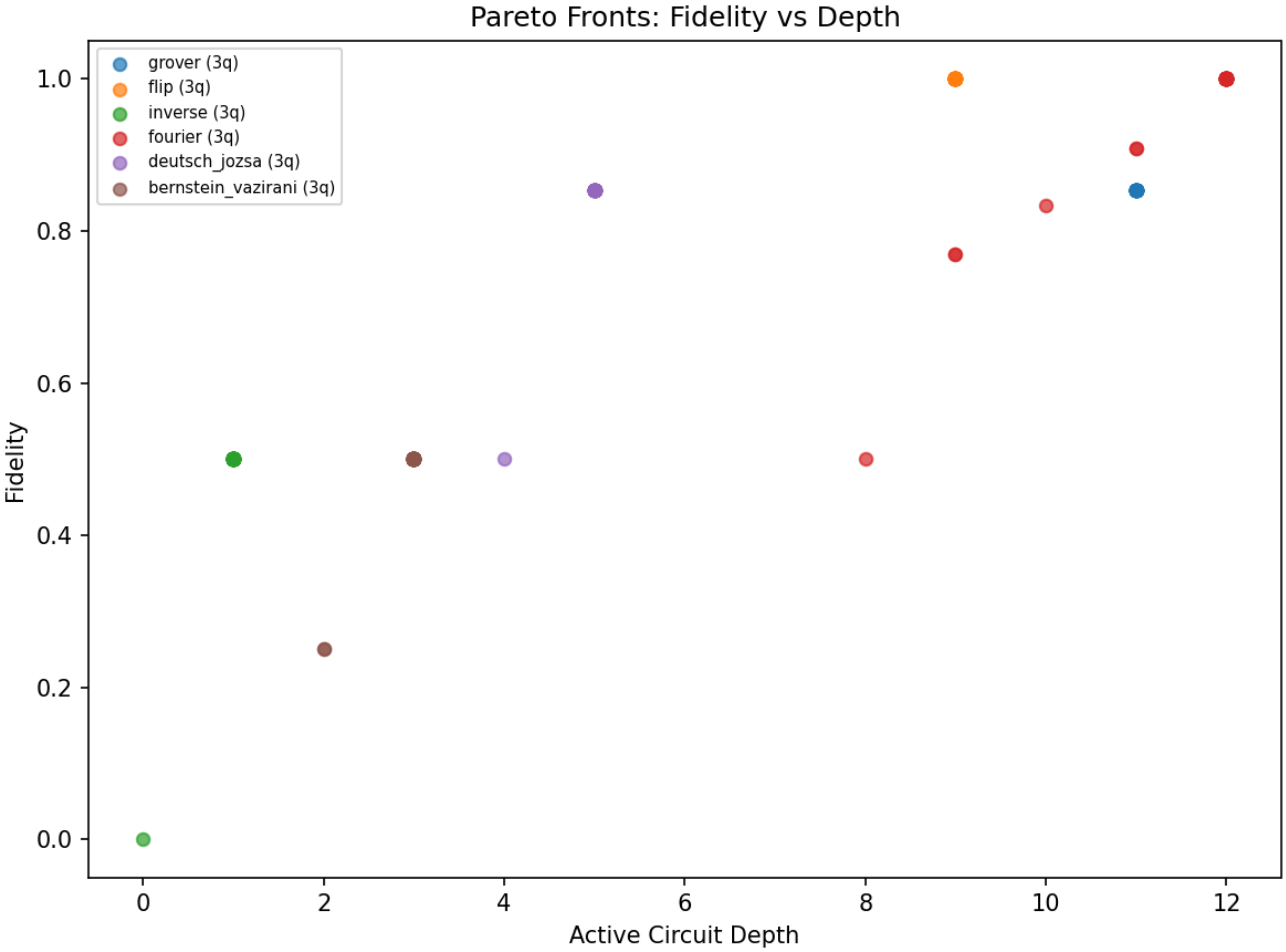

Figure 6: NSGA-II Pareto fronts showing the trade-off between fidelity and active circuit depth. Each point is a non-dominated solution from the final population.

C. M. Dawson and M. A. Nielsen. The Solovay-Kitaev algorithm. *Quantum Information & Computation*, 6(1):81–95, 2005.

V. V. Shende, S. S. Bullock, and I. L. Markov. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010, 2006.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.

M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

T. Fösel, M. Y. Niu, F. Marquardt, and L. Li. Quantum circuit optimization with deep reinforcement learning. arXiv preprint arXiv:2103.07585, 2021.

L. Moro, M. G. A. Paris, M. Restelli, and E. Prati. Quantum compiling by deep reinforcement learning. *Communications Physics*, 4(1):178, 2021.

S. Rietsch, A. Y. Dubey, C. Ufrecht, M. Periyasamy, A. Plinge, C. Mutschler, and D. D. Scherer. Unitary synthesis of Clifford+T circuits with reinforcement learning. In *Proc. IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 824–835, 2024.

M. Sharma and H. C. Lau. A comparative study of quantum optimization techniques for solving combinatorial optimization benchmark problems. arXiv preprint arXiv:2503.12121, 2025.

Y. Sun, Z. Wu, V. Tresp, and Y. Ma. Quantum architecture search with unsupervised representation learning. *Quantum*, 10:1994, 2026.

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

J.-B. Mouret and J. Clune. Illuminating search spaces by mapping elites. arXiv preprint arXiv:1504.04909, 2015.

J. K. Pugh, L. B. Soros, and K. O. Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.