INFORMATICA, 2026, Vol. 0, No. 0, 1–23

DOI: https://doi.org/10.15388/26-INFOR639

Quantum Genetic Algorithm with Dynamic Encoding Scheme

Jun Suk KIM[1,2], Man-Je KIM[3,*], Chang Wook AHN[1,*]

[1] *Department of AI, Gwangju Institute of Science and Technology, Republic of Korea*
[2] *GIST Institute for AI, Gwangju Institute of Science and Technology, Republic of Korea*
[3] *Department of AI, Chonnam National University, Republic of Korea*
e-mail: junsuk89@gmail.com, jaykim0104@jnu.ac.kr, cwan@gist.ac.kr

Received: January 2026; accepted: July 2026

Abstract. The field of Quantum Genetic Algorithm (QGA) has seen extensive efforts to develop reasonable algorithmic designs mainly due to its potential as well as the intrinsic difficulty to reproduce the necessary processes with the limited amount of quantum resources available, although it still remains one of rather theoretical concepts of approach to achieving quantum optimization. In this paper, we propose a dynamic encoding scheme that combines quantum adaptive search with iterative approximation of the search region. The method reuses the same quantum index register while updating the classical coordinate mapping associated with its basis states, thereby increasing local coordinate resolution without globally refining the entire continuous domain. Through noiseless statevector simulations on benchmark functions, we compare the proposed method with selected QGA variants in terms of final optimization accuracy and simulated qubit usage. The results show improved optimization performance under similar maximum qubit constraints, establishing a simulation-level resource advantage.

Key words: Quantum Algorithm, optimization, Genetic Algorithm.

1. Introduction

The prospect of an idea to implement Evolutionary Algorithm (EA) via quantum computation led to the foundation of a research field called Quantum Evolutionary Algorithm (QEA) (Sofge, 2008) or its main subcategory, Quantum Genetic Algorithm (QGA), along with anticipation that quantum application could dramatically enhance the algorithm's optimization capability. Indeed, one of the key issues in designing an effective evolutionary algorithm is its comparatively high computational cost caused by the considerable number of fitness evaluations, especially for large and complex problems (Goldberg, 1988). Such an intrinsic limitation is mainly due to the algorithm's heuristic nature (Tanabe and Ishibuchi, 2020), and there exists a widespread expectation among the related studies that successful implementation of QGA would vastly benefit from quantum advantages including the parallel computation capability (Udrescu *et al.*, 2006).

*Corresponding authors.

As the initial optimism has subsided, subsequent studies on QGA indicate that there still remains a number of critical issues that hinder efforts to design a valid algorithmic structure for successful application to practical domains, which need to be carefully dealt with before QGA would possibly reach the level that any meaningful comparison with its classical counterparts can be considered (Lahoz-Beltra, 2023). Especially, in the current era of noisy intermediate-scale quantum (NISQ) computing, one of the rooted concerns in designing QGA is the necessity of utilizing qubits effectively due to their scarcity with respect to building a practical algorithmic structure (Nielsen and Chuang, 2004). Proper qubit allocation to represent the data and problem definition is particularly important in the aspect of solving large, complex real-world problems, which in many occasions consist of extensive domains that numerous solution candidates can exist.

Similar to its classical analog, QGA is mostly configured based on binary encoding, i.e. each gene comprises a value of either 0 or 1. The population is thus essentially a binary matrix in size of $M \times N$, where N is the number of individuals and M is the length of each individual as a binary string. The Hadamard operation prepares a q-qubit register in a coherent superposition over 2^q computational-basis states with uniform probability amplitude. These basis states can be used to encode candidate chromosomes, and subsequent quantum operations act on the amplitudes of the whole register. After measurement, sampled bit strings are obtained accordingly with the final probability distribution. In this setting, however, it is not hard to predict that the process of mapping a multi-dimensional continuous domain into quantum metrics could lead to massive qubit requirements or failure of accurate problem and data representation, either of which could be detrimental to the feasibility and practicality of the designed algorithm, not to mention additional difficulties caused by the quantum-inherent phenomena such as the Barren Plateau (McClean *et al.*, 2018).

This paper focuses on the representation problem that arises when QGA is applied to continuous optimization. In conventional binary-encoded QGA, a continuous domain must first be transformed into a finite set of candidate states. A fine global discretization improves coordinate precision, but it also increases the number of qubits required for index, value, comparator, and auxiliary registers. Conversely, a coarse discretization is more qubit-efficient but may fail to represent sufficiently accurate candidate solutions. This precision-resource trade-off becomes more severe as the dimensionality of the problem or the desired numerical precision increases.

The proposed method addresses the aforementioned issue through a dynamic encoding scheme, which, while not to be claimed as a uniquely quantum idea, is introduced to present that classical techniques can be effectively applied to quantum tasks of optimization. The genuinely quantum component of our study lies in how the candidate set inside the current region is represented and searched: a finite grid of coordinate vectors is encoded by quantum index states, and quantum adaptive search is applied to amplify and select the candidate satisfying the current fitness threshold. Instead of assigning qubits to a fixed global discretization of the entire search space, the same quantum index register is reused across generations while its associated coordinate mapping is updated according to the current search region. As the region contracts, the effective spatial density of the

encoded grid increases without requiring a corresponding increase in the number of index qubits.

In summary, the following is a list of the main contribution points of our paper:

- We identify the precision and resource trade-off that arises when the binary-encoded QGA is applied to continuous optimization, where finer global discretization improves coordinate precision but increases the required size of index, value, comparator, and auxiliary registers.
- We propose a dynamic encoding scheme for continuous-domain QGA, in which the same quantum index register is reused across generations while its associated coordinate mapping is updated accordingly with the current search region to increase grid resolution.
- We provide a local approximation argument showing how contraction of the search region reduces the grid spacing and the corresponding discretization error under a Lipschitz-continuity assumption.
- We evaluate the proposed method in noiseless statevector simulations on standard two-dimensional benchmark functions and compare it with the reduced and strictly structured QGA variants. The results suggest improved optimization accuracy under essentially the same maximum simulated-qubit constraint.

The next section introduces a number of notable studies on quantum evolutionary algorithm and briefly explains their outlines and contribution to the field. In the background section, we concisely go over the basics of quantum genetic algorithm, which are necessary for the context of our study. In the subsequent section we identify the qubit utilization issue in QGA and explain our method to resolve it. After that, the experiment section covers the setup, results, and analysis on the comparison made among the QGA techniques, including our method, which is then followed by the conclusion at the end.

2. Literature Review

The idea of combining evolutionary computation with quantum computation has developed through several related but distinct research directions, including quantum-inspired evolutionary algorithms, quantum genetic algorithms, and hybrid quantum-classical optimization. Quantum-inspired evolutionary algorithms were among the earliest attempts to import quantum concepts into evolutionary computation. Han and Kim (2002) proposed a quantum-inspired evolutionary algorithm in which a chromosome is represented by probability amplitudes of quantum-inspired bits, and candidate solutions are obtained by observation and updated by rotation-gate-like rules. This framework improves diversity preservation and compactly represents a probability distribution over candidate solutions, but it is intended to be executed on a classical computer and therefore does not provide genuine quantum speedup. Later studies and surveys expanded this line of work to various combinatorial and numerical problems, but most of the approaches still depend on binary or problem-specific encodings, and continuous optimization requires either real-coded extensions or discretization (Zhang, 2011).

Quantum genetic algorithms, distinguished from the quantum-inspired variant above, attempt to reproduce the optimization process of GAs in a quantum computational setting. Malossini *et al.* (2008) proposed a quantum genetic optimization framework, in which quantum procedures are used to accelerate fitness evaluation and selection. Udrescu *et al.* (2006) introduced an implementation of QGA based on Grover's search, leading to the reduced quantum genetic algorithm (RQGA), where the optimization procedure is mainly reduced to quantum selection. This approach is closely related to quantum adaptive search, in which Grover-type amplitude amplification and quantum minimum finding are iteratively used to improve a threshold solution (Dürr and Hoyer, 1996; Grover, 1996; Baritompa *et al.*, 2005). The strength of RQGA and quantum adaptive search is their relatively simple structure and their theoretical quadratic query advantage for finite search spaces. The main limitation, however, is that they require the search space to be encoded as a finite set of candidate states. Therefore, for continuous optimization, the domain must first be discretized, and the number of required qubits increases rapidly correspondingly with dimension and desired coordinate precision.

Another branch of QGA research tries to preserve the classical GA structure more explicitly. SaiToh *et al.* (2014) proposed a semi-classical QGA with quantum crossover and mutation operations, showing that genetic operators can be implemented in a quantum circuit model. Such strictly structured QGA (SSQGA) designs are conceptually important because they mimic the classical GA procedures relatively closely. However, they require additional registers for populations, chromosome copies, pseudo-randomization, and genetic operations such as crossover. As a result, their qubit and circuit-depth requirements are relatively high, which is a serious drawback in the current noisy intermediate-scale quantum era, where quantum resources are limited and noise remains a practical obstacle (Nielsen and Chuang, 2004; McClean *et al.*, 2018; Lahoz-Beltra, 2023).

Recent hybrid studies suggest that resource-aware combinations of quantum and classical search are becoming increasingly important. For example, quantum-assisted genetic sampling uses amplitude amplification as a selection mechanism inside a classical GA, while hybrid RQGA-based methods reduce the quantum search space by fixing part of the chromosome classically (Acampora *et al.*, 2022; Ardelean and Udrescu, 2024). These methods reduce resource demands to a certain degree, but they still largely depend on binary encodings or finite candidate sets. In the broader quantum optimization literature, quantum annealing, QAOA, and variational quantum algorithms provide alternative ways to solve optimization problems by formulating them as Ising, QUBO, Hamiltonian, or parameterized-circuit problems (Farhi *et al.*, 2014; Preskill, 2018). While these approaches appear promising, their applicability to general continuous black-box optimization is indirect and often requires problem reformulation, embedding, or iterative parameter training.

Commonlt, the current QGA-related approaches face a difficulty when applied to continuous optimization under limited qubit availability. Quantum-inspired methods are comparatively flexible in that aspect, but they run classically after all, and SSQGAs preserve crossover and mutation but require the substantial amount of quantum resources. RQGA and quantum adaptive search are relatively qubit-efficient but assume a fixed finite representation of the search space. Those shortcomings imply a precision-resource trade-off:

Table 1
Comparison of optimization approaches related to the proposed method.

Comparison of optimization approaches related to the proposed method				
Name	Method	Encoding	Qubit required	Advantage & disadvantage
Classical GA	Classical heuristic	Binary or real-valued	N/A	Robust; no quantum advantage
Quantum-inspired GA	Classical quantum-inspired heuristic	Simulated probability amplitudes	Non-physical	Compact probability representation; no genuine quantum advantage
RQGA	Grover-based selection	Fixed search based on index states	Relatively low among QGA variants	Expected quadratic speedup; costly for high precision
SSQGA	Circuit-based QGA with GA operators	Quantum population registers	Relatively high among QGA variants	Closely mimics classical GA; large circuit depth
Hybrid QGA	Quantum-classical hybrid	Classical population with quantum-assisted search	Lower than full-QGA	Comparatively practical; limited quantum advantage
Proposed Method	Quantum adaptive search with dynamic encoding	Approximate coordinate grid search	Bounded by RQGA and SSQGA requirement	Effective search precision; multimodal convergence remains an open issue

a coarse discretization uses fewer qubits but loses accuracy, whereas a fine discretization improves accuracy but quickly becomes impractical. The proposed method in this paper intends to address this gap by introducing a dynamic encoding scheme for continuous optimization. Table 1 provides a comparison among the proposed method and the related optimization approaches.

3. Background

This section briefly covers the background information relevant to the scope of this paper. We first explain the basics of QGA and then introduce its two main implementation approaches, reduced QGA and strictly structured QGA.

3.1. *Quantum Genetic Algorithm in General*

A classical GA sets up a population of solution candidates, which are often referred to as chromosomes. Each chromosome is a solution candidate to the given problem, and its fitness value is calculated in terms of its genes that represent values for variables and elements. In QGA, gene values are encoded in the computational-basis states of one or more qubits, and a sequence of encoded gene values is represented by a quantum register. A quantum chromosome can therefore be understood as a register whose computational-basis states correspond to possible chromosome encodings. Before measurement, this register may exist in a coherent superposition of multiple encoded chromosomes, while a classical chromosome is obtained only after measurement. In the computational basis,

a quantum chromosome is basically a state vector formed by tensor products among multiple qubits. At the beginning of computation, the chromosome register of n qubits is initialized as $|\psi\rangle = |0\rangle^{\otimes n}$. Applying Hadamard gates to all qubits prepares the uniform superposition

$$H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} |z\rangle. \tag{1}$$

The register then has support over all binary strings of length n. In QGA, each basis string $|z\rangle$ may be interpreted as an encoded candidate chromosome. Quantum gates transform the probability amplitudes of these encoded candidates coherently, and measurement returns classical bit strings according to the resulting probability distribution.

It is a common practice to implement the quantum selection operator with utilization of Grover's search algorithm (Grover, 1996), one of the relatively well known quantum algorithms for its quadratic speedup capability. Its procedure is divided into two parts, the quantum oracle and the Grover's diffuser. The oracle encodes and stores the problem function and input data, which then can be processed with successive quantum operations. Building a cost-effective oracle itself is a crucial topic with respect to developing any oracle-oriented quantum algorithm, and we will further discuss the related issues in the later part. The Grover's diffuser, coming after the oracle, executes the process of amplitude amplification, which selectively amplifies the probability amplitude of the target state, raising the probability to guarantee its measurement at the end of the computation. Theoretically, the Grover's search can specify the target state with $O(\sqrt{n})$ queries from n items (Dürr and Hoyer, 1996). In QGA, the Grover's search is employed to specify and select chromosomes with the highest fitness values in each generation. A quantum circuit for an overall algorithmic structure for QGA is presented in Fig. 1.

Implementing quantum crossover and mutation operators is relatively difficult in QGA, since uncontrolled interactions or unintended intermediate measurements can cause decoherence, collapse, or noise, thereby disturbing the encoded superposition. In addition, operations analogous to crossover and mutation must be implemented coherently through reversible quantum circuits, often requiring additional registers, controlled operations, and

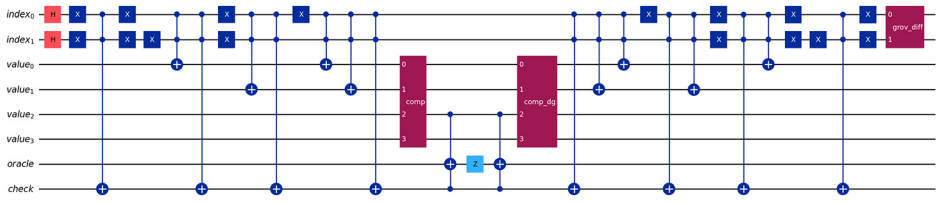

Fig. 1. A quantum circuit that implements a conventional QGA with classical-to-quantum data encoding, a minimum search oracle, and the Grover's diffusion process. The index qubits are put into superposition by the Hadamard gates to generate chromosomes, and a series of the X and CNOT gates is applied to map the data onto the value qubits. The oracle, along with a comparator gate, marks the target state by the CCNOT and Z gates, which run a phase-flip. Then the whole procedure is iterated backwards to "uncompute" the qubits, after which the Grover diffusion is applied to amplify the target state's probability amplitude.

uncomputation (Yanofsky and Mannucci, 2008). Since the processes of crossover and mutation need to address and specify the target chromosomes periodically during computation, reproducing them with quantum computation would require a genuine strategy to bypass the aforementioned limitation.

3.2. *Reduced Variant*

Equation (1) implies that putting n qubits into uniform superposition generates a series of 2^n strings that consists of every possible binary combination of 0s and 1s of length n. Therefore, theoretically, if the sufficient number of qubits could be utilized to map the problem function entirely, the whole procedure would be reduced to simply searching for the optimum in the space where every possible solution candidate exists. Reduced quantum genetic algorithm (RQGA), referred to as quantum adaptive search outside the QGA literature, was introduced inspired by such methodological reduction (Sun and Xiong, 2014).

In RQGA, the only genetic operator involved is the selection, which is reproduced by Grover-BBHT algorithm, an improvement over the original Grover's search. The absence of crossover and mutation here is not a procedural omission, but a defining feature of the reduced variant, whose purpose is to regulate circuit depth and qubit consumption by reducing the genetic process to quantum selection. The Grover's oracle O_f is a boolean operator, i.e. with the registers of input and auxiliary qubits $|x\rangle$ and $|-\rangle$, respectively,

$$O_f |x\rangle |-\rangle = (-1)^{f(x)} |x\rangle |-\rangle , \tag{2}$$

which is accustomed to a black-box function $f(x) : \{0, 1\}^n \xrightarrow{\{} 0, 1\}$, activating a phase flip depending on the evaluation of $f(x)$. Roughly speaking, the Grover's oracle is fed with a question, "Does this state satisfy the given condition (problem function)?" and it returns "yes" or "no" by marking the satisfactory state with a flipped sign. This implies that, for an optimum search problem, a threshold needs to be provided to the oracle to compare with the solution candidates. Indeed, in RQGA, the best solution candidate in each generation is marked as a threshold for the next generation, where it is compared with the next best candidate. Performing this task iteratively would ultimately lead to discovering the near-optimal solutions, negating the necessity of the crossover or mutation procedures as long as the corresponding population is capable of containing the entirety of the possible candidate pool.

3.3. *Strictly Structured Variant*

Strictly structured quantum genetic algorithm (SSQGA) is a rather recently developed approach to implement QGA, which strictly adopts the algorithmic procedure of classical GA. It maps only part of the problem's decision space and utilizes all of the genetic operators, including the crossover and mutation, to search the optimum. The algorithm starts with generating a pseudo-randomizer circuit to initialize the population of c chromosomes

under non-uniform superposition on n qubits, where $c < 2^n$ applies. The purpose of this approach is to reduce the number of solution candidate states to be processed with the Grover's diffuser, decreasing the required number of queries to complete the diffusion.

The crossover operation in SSQGA is enabled by generating two identical populations sets

$$|\psi\rangle = |P_1\rangle \otimes |P_2\rangle = \frac{1}{c} \sum_{i=1}^{c} \sum_{i=1}^{c} |a_1 a_2\rangle \, |x_1 x_2\rangle \,, \tag{3}$$

where the population sets $|P_1\rangle$ and $|P_2\rangle$ are equal, and a_k is an index register to a chromosome register x_k in each population $|P_k\rangle$. The chromosome registers are then halved as $|x_1^{left} x_2^{right}\rangle \, |x_2^{left} x_1^{right}\rangle$, allowing pseudo-swaps between qubits with the same index value by placing a series of CNOT gates. A subsequent study proposes an improved version of this structure by preserving a number of qubits in the chromosome register to significantly reduce the population size. Lastly, the mutation operation in SSQGA is implemented by applying X gates to specific qubits in the chromosome register.

4. Problem Definition and Proposition

In this section, we point out a predicament in the current QGA designs with regard to qubit utilization and explain our proposed approach that addresses the matter.

4.1. *Search Space Presentation*

In practice, QGA is generally built upon the computational basis, which takes the state vector $|\psi\rangle$ of each qubit as

$$|\psi\rangle = \alpha \, |0\rangle + \beta \, |1\rangle \,, \tag{4}$$

where α and β are the probability amplitudes of $|0\rangle$ and $|1\rangle$ that must satisfy $\alpha^2 + \beta^2 = 1$. This implies that, in order to encode classical data into a quantum circuit, it needs to be converted into a form of binary string $q_0 q_1 q_2 \ldots q_n, q_i \in \{0, 1\}$, which then can be mapped into qubits. To allow decoding the result of the quantum computation back to the classical format, an index is assigned to each quantum chromosome, which acts as an indicator that can be referred to for tracking the solution's address in the original data. Consequently, encoding classical data into a quantum circuit requires proper allocation of qubits to the index arrangement and chromosome value mapping. n_{idx} qubits in an index register are put into uniform superposition to generate $2^{n_{idx}}$ indices, while n_{val} qubits in a value register are connected with the index qubits via CNOT gates to represent n_{val} genes accordingly with the corresponding indices.

Since GA is a heuristic optimization algorithm that stochastically chooses solution candidates from the search space, the following lemma holds.

Lemma 1. *Suppose a genetic algorithm that generates an initial population of k chromosomes with a multiset* $X = \{x_1, \ldots, x_k\}$ *of the corresponding fitness values. For a search problem with a sufficiently large, continuous decision space, X is reduced to a set* $X' = \{x'_1, \ldots, x'_s\}$ *where* $x'_i \neq x'_j$ *for all* $i \neq j$ *and* $s = k$.

Proof. Since the decision space is continuous and sufficiently large, the probability that two independently sampled chromosomes correspond to the same point in the search space can safely be assumed to be zero. For a non-degenerate continuous fitness function, the induced fitness values are also almost surely distinct. Hence, for any $i \neq j$,

$$\Pr(x_i = x_j) = 0.$$

Because the population contains only finitely many chromosomes, the probability that any duplicate fitness value appears is also zero. Therefore, with probability one, the multiset

$$X = \{x_1, \ldots, x_k\}$$

contains k distinct elements and can be reduced to a set

$$X' = \{x'_1, \ldots, x'_s\}$$

with $s = k$. Thus, for continuous optimization problems, the generated chromosomes can be assumed to have distinct relative fitness values. Consequently, a value register must be sufficiently large to distinguish the fitness-related representation of all indexed chromosomes. Since n_{idx} qubits generate $2^{n_{\text{idx}}}$ candidate indices and n_{val} qubits represent $2^{n_{\text{val}}}$ distinguishable values, full differentiation requires

$$2^{n_{\text{val}}} \geqslant 2^{n_{\text{idx}}},$$

and therefore

$$n_{\text{val}} \geqslant n_{\text{idx}}.$$ □

In other words, as to continuous function-based problems, one must guarantee $n_{val} \geqslant n_{idx}$ in order to fully differentiate the chromosomes in terms of their relative fitness values. Overall, QGA requires $k \cdot n_{\text{idx}}$ qubits to generate up to $2^{n_{\text{idx}}}$ individuals with a constant $k \geqslant 2$ that could vary depending on the method of encoding and addition of auxiliary qubits for subsequent computations. While the difference in k is trivial asymptotically in terms of the number of qubits, as it is bounded to $O(n_{\text{idx}})$, the difference in terms of the number of chromosomes could be significant, since it grows exponentially with the number of qubits, i.e. it is safe to claim that $O(2^{k_1 \cdot n_{\text{idx}}}) < O(2^{k_2 \cdot n_{\text{idx}}})$ if $k_1 < k_2$. An appropriate management over the numbers of qubits and generable chromosomes thus appears to be a crucial matter in building a practical QGA with respect to its required implementation cost.

A continuous function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is based upon an indefinite domain, where the infinite amount of input possibilities exists. For solving an optimization problem of such a kind, the methodology of RQGA, that the whole procedure can be reduced to selecting the optimum from a set of all the possible solution candidates, has its limitation, since it is virtually impossible to designate a finite set for input values from a continuous domain. This matter of search space presentation appears to be especially critical in QGA and many other quantum algorithms, where qubits are essentially scarce and their proper distribution to various operations is necessary (Giri and Korepin, 2017). While quantum superposition allows generation of exponentially numerous solution candidates with respect to the number of qubits in the index register, it is important to recall that at least the equal number of qubits needs to be assigned to the value register in order to distinctively present the features of candidates on the decision space, which in most cases consists of real numbers with multiple digits. The fact that n_{val} has n_{idx} as a lower bound for continuous functions infers that increasing the population size raises the minimum number of qubits required for value presentation as well, expanding the overall qubit consumption in two-folds.

4.2. *Search Approximation*

Our proposed method to handle the issue explained in the last subsection is largely based on dimension-wise discretization of the search space, which transforms the coordinates of solution candidates along each axis to countable measures. Although the presented implementation is incorporated into the reduced variant of QGA, the proposed dynamic encoding scheme is not restricted to RQGA, since it modifies the representation of the candidate search space before the selection process and can therefore be combined with other QGA variants that include crossover or mutation operators. This approach of approximation, derived from the following lemma, allows to represent the candidates distinctively with the limited number of available qubits.

Lemma 2. *Suppose a multi-dimensional search space S of a continuous problem function with an infinite set $\mathbb{C}$ of all the possible solution candidates in S. Then there exists a finite subset $C \subset \mathbb{C}$, whose elements are located across S under uniform dispersion with the identical interval between each element pair in every dimension.*

Proof. In practical optimization, a continuous search space is considered over a bounded region. Let

$$S = [a_1, b_1] \times \cdots \times [a_d, b_d] \subset \mathbb{R}^d.$$

For each dimension j, choose a finite number $m_j \geqslant 2$ of grid points and define the interval

$$h_j = \frac{b_j - a_j}{m_j - 1}.$$

Then the set

$$\widetilde{C} = \left\{(a_1 + i_1 h_1, \ldots, a_d + i_d h_d) : i_j \in \{0, \ldots, m_j - 1\}\right\}$$

is finite, satisfies $\widetilde{C} \subset S$, and places candidate points at identical intervals along each dimension. Therefore, every bounded continuous search region admits a finite uniformly dispersed approximation. This justifies the use of a finite coordinate grid to approximate the continuous decision space in the proposed QGA encoding scheme. □

With Lemma 2, the search space can be described by coordinate vectors that are uniformly placed along each dimension, acting as approximate coordinates generated by superposition of the index qubits. $\frac{n_{\text{idx}}}{d}$ qubits are assigned to each dimension to place $2^{\frac{n_{\text{idx}}}{d}}$ vectors, and every intersection between two vectors is regarded as a solution candidate. Overall, $2^{n_{\text{idx}}}$ candidates in total can be generated with n_{idx} qubits in the index register, forming a population. The task of the selection operator in QGA is then to evaluate the fitness of these candidates and specify the best one, which can be represented by the corresponding vectors' binary indices.

It is important to distinguish the role of the adaptive search region from the quantum component of the proposed method. The contraction and expansion of the search range is a classical control mechanism, and similar ideas appear in adaptive grid refinement, pattern search, and trust-region optimization. Our method intends not to emphasize the use of local refinement itself, but the way such refinement is coupled with QGA encoding. At each generation, the computational basis states of the index register are mapped to coordinate vectors in the current search region. Therefore, the same set of quantum states represents different physical coordinates as the region changes. This allows the algorithm to increase the effective coordinate resolution locally without globally increasing the number of qubits to represent the entire continuous domain.

Sporadic placement of coordinate vectors in the continuous search space implies that there is a high probability that the global minimum is not collinear with any of the vectors that form the approximate grid. This loss of information caused by the approximation can be reduced by gradually decreasing the size of the search space.

Lemma 3. *Let $x, x' \in \mathbb{R}^d$ and $y, y' \in \mathbb{R}^d$ be independently and uniformly sampled from the hypercubes $X = [-r, r]^d$ and $Y = [-kr, kr]^d$, respectively, with constants $r > 1$ and $k > 1$. Then the expected Euclidean distance between two random points from the hypercubes satisfies $\mathbb{E}[\|x - x'\|] < \mathbb{E}[\|y - y'\|]$.*

Proof. Let u, u' be independently and uniformly sampled from the unit hypercube $[-1, 1]^d$. Then points sampled from

$$X = [-r, r]^d$$

can be written as

$$x = ru, \qquad x' = ru',$$

while points sampled from

$$Y = [-kr, kr]^d$$

can be written as

$$y = kru, \qquad y' = kru'.$$

By the homogeneity of the Euclidean norm,

$$\|x - x'\| = r\|u - u'\|,$$

and

$$\|y - y'\| = kr\|u - u'\|.$$

Taking expectations gives

$$\mathbb{E}\big[\|y - y'\|\big] = k\,\mathbb{E}\big[\|x - x'\|\big].$$

Since $k > 1$, it follows that

$$\mathbb{E}\big[\|x - x'\|\big] < \mathbb{E}\big[\|y - y'\|\big].$$

Therefore, reducing the size of the search region reduces the expected distance between candidate points. With the same number of coordinate vectors, this increases the effective density of the approximate grid, supporting the proposed iterative contraction strategy. □

Lemma 3 supports a geometric interpretation of region contraction: when the number of grid points is fixed, reducing the search radius increases the density of candidate points in the current region. It does not imply that the global optimum of the original domain remains inside the contracted region; rather, the direct consequence for the proposed encoding scheme is instead a local approximation property: if the relevant optimum lies inside the current search region, then the uniformly encoded grid can approximate the best point in that region more closely as the region radius decreases.

Proposition 1. *For the local approximation bound, let the search region at generation t be*

$$S_t = [c_t - r_t, c_t + r_t]^d,$$

and let G_t be a uniformly discretized grid over S_t with m coordinate values per dimension. Then the grid spacing is

$$h_t = \frac{2r_t}{m - 1}.$$

For any point $x \in S_t$, there exists a grid point $\hat{x} \in G_t$ such that

$$\|x - \hat{x}\|_2 \leqslant \frac{\sqrt{d}}{2} h_t.$$

If the objective function f is Lipschitz continuous on S_t with constant L, and if x_t^* is the minimum of f over S_t, then the best grid point $g_t^* \in G_t$ satisfies

$$f\left(g_t^*\right) - f\left(x_t^*\right) \leqslant L \frac{\sqrt{d}}{2} h_t.$$

Proof. The uniform grid partitions each coordinate axis into intervals of length h_t. Hence, for any $x \in S_t$, the nearest grid point $\hat{x}$ differs from x by at most $h_t/2$ in each dimension, which gives

$$\|x - \hat{x}\|_2 \leqslant \frac{\sqrt{d}}{2} h_t.$$

Let $\hat{x}_t$ be the nearest grid point to x_t^*. Since g_t^* is the best point in G_t,

$$f\left(g_t^*\right) \leqslant f(\hat{x}_t).$$

By Lipschitz continuity,

$$f(\hat{x}_t) - f\left(x_t^*\right) \leqslant L\|\hat{x}_t - x_t^*\|_2 \leqslant L \frac{\sqrt{d}}{2} h_t.$$

Therefore,

$$f\left(g_t^*\right) - f\left(x_t^*\right) \leqslant L \frac{\sqrt{d}}{2} h_t.$$ □

The bound above is conditional on the relevant optimum remaining inside the current search region. Let x^* denote the global minimum in the original domain and suppose $x^* \in S_t$. After the selection step, the proposed method constructs the next search region as

$$S_{t+1} = [g_t - r_{t+1}, g_t + r_{t+1}]^d, \tag{5}$$

where g_t is the selected candidate. Then x^* remains inside the next search region if and only if

$$\|g_t - x^*\|_\infty \leqslant r_{t+1}. \tag{6}$$

Therefore, contraction is considered safe when the selected candidate is sufficiently close to the global minimum. If this condition is violated, the global optimum is likely to be

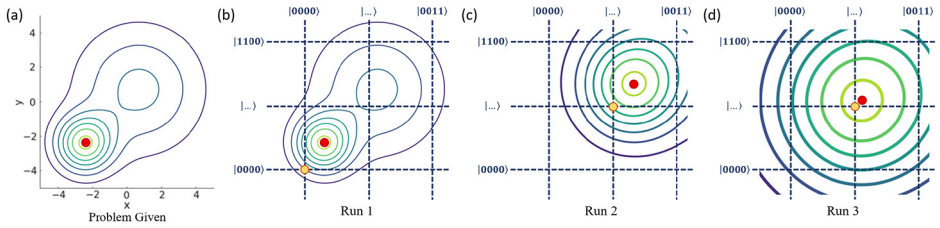

Fig. 2. A graphic of the proposed method's overall procedure. (a) A continuous problem function is given. (b)-(c) An approximate search space is formed by the coordinate vectors placed along each dimension. The solution candidate with the highest fitness is chosen, which becomes the centre of the contracted search space for the next run. (d) Iterating this procedure may eventually lead to a near-optimal solution, terminating the process with an arbitrary condition.

excluded from the next search region and subsequent contractions cannot recover it unless an expansion, restart, or multi-region search mechanism is applied.

We can expect that in a smaller search space, the coordinate vectors will be placed with higher density that probabilistically decreases their Euclidean distances from the minimum inside the space. In other words, theoretically, through repetitive shrinking of the search region, the grid resolution within that region increases, thereby reducing the local discretization error when the target optimum remains inside the region. The best candidate in each generation becomes the centre of the search space for the next generation, which is bounded by other candidate points that are adjacent to the centre candidate on the plane of the approximated space in every dimension. We deem that this methodology is plausible in a sense that the arbitrary local optima are likely to exist within the bounded area due to the approximity of the solution candidates. The overall procedure of the proposed method is pictured in Fig. 2 along with its pseudo-code in Algorithm 1.

4.3. *Coping with Multimodality*

Admittedly, our approach described in the last subsection does not guarantee that the global minimum would remain inside the search space under all circumstances. Let x^* be the global minimum and let g_t be the candidate selected by quantum adaptive search from the approximate grid G_t. If g_t lies in the basin of a local minimum and the contracted region centred at g_t does not contain x^*, then the algorithm becomes restricted to a subregion from which the global minimum is unreachable by further contraction alone. This can occur when the current grid is too coarse to place a candidate sufficiently close to the global basin, or when a local basin contains grid points with better observed fitness values than the available grid points near the global minimum.

We introduce an expansion mechanism to reduce the risk described above. When the best fitness value does not improve for a prescribed number of generations, the search region is expanded in order to recover candidate areas that may have been excluded by previous contractions. This mechanism increases the opportunity to escape from local minima, although it does not by itself guarantee global convergence for arbitrary non-convex or non-Lipschitz functions. A formal global convergence guarantee would require

Algorithm 1 Quantum Adaptive Search in Approximate Region

Input: $f(X)$, QAS
Parameter: num_q, ran, gen, cen, thres
Output: save_best

num_q $\leftarrow$ 2, thres $\leftarrow \infty$, ran $\leftarrow$ 10, cen $\leftarrow$ *random*
while gen **do**
  Generate population P within [cen−ran, cen+ran].
  Run QAS(num_q) on $f(P)$.
  cand_list $\leftarrow$ measure(QAS)
  if cand_list[best] $\leqslant$ thres **then**
    best_cand $\leftarrow$ cand_list[best]
    cen $\leftarrow$ coord[best_cand]
    ran $\leftarrow$ ran / 2^{num_q-1}
  else
    num_q += 1
    if *stuck* **then**
      ran $\leftarrow$ ran $*$ 2^{num_q-1}
    end if
  end if
  save_best $\leftarrow$ best_cand
end while
return save_best

additional techniques, such as repeated coverage of the original domain, a restart policy with nonzero probability of sampling every relevant region, or a multi-region search procedure that maintains several candidate basins simultaneously, which are the commonly suggested practices among various heuristic optimization studies.

In an environment where the abundant number of qubits can be utilized, the aforementioned issue of multimodality could possibly be solved by adopting parallel processing upon multiple intersections of the coordinate vectors. This setting would allow to proceed with multiple search spaces concurrently, and their locations can be decided by the observation counts of the state vectors on multiple measurements of the circuit. Several copies of the circuit that implements our method would be needed to run this procedure.

5. Experiment

This section explains the setup, results, and analysis on the experiment conducted to verify the validity of our proposed method.

5.1. *Setup*

We used IBM Qiskit (Abraham *et al.*, 2019) to set up a quantum simulation environment and to implement our method in it. For result clarification purposes, we chose to

run our program with Qiskit Aer `StatevectorSimulator`, which supports up to 30 simulated qubits in a state vector form. The oracle includes the in-built `IntegerComparator` function that requires n_{val} ancilla qubits, raising the total number of qubits in the index and value registers to $3 \cdot n_{\text{idx}}$. Since two qubits are spared for the task of specifying target states via phase-flip, the maximum possible number for n_{idx} is 9. Such scarcity of qubit indicates that running the simulation upon complex real-world problems would be impractical and hardly produce any meaningful results. Therefore, we decided to conduct the experiment upon four benchmark optimization problem functions, Rosenbrock (Rosenbrock, 1960), Rastrigin (Hoffmeister and Bäck, 1991), Ackley (Ackley, 1987), and Sphere (Picheny *et al.*, 2013). Each function is set as 2-dimensional with up to 4 index qubits assignable to each dimension, generating 16 coordinate vectors at the maximum.

In order to measure the comparative performance of the proposed method, we implemented the standard RQGA and SSQGA under the same setting. Our method and RQGA would start with $n_{\text{idx}} = 4$ and add two additional index qubits to their circuits each time the found minimum has remained the same for more than 3 generations. SSQGA, on the other hand, starts with $n_{\text{idx}} = 8$ because it requires two population sets from the beginning. With $n_{\text{idx}} = 4$ initial index qubits, SSQGA would be allowed to generate only 4 chromosomes for each of the initial populations, which we deem is unsuitable for measuring the algorithm's practicality. All three algorithms were given 50 generations to perform the optimization task, and the whole procedure was iterated 10 times to assess their average performance. An elitist strategy is chosen, with which the best candidate in each generation is saved and compared with the next one, replacing it if better.

The benchmark functions used in our experiment generally assume a relative small search space. For example, it is a common practice to evaluate the Rastrigin function on a hypercube of $x_i \in [-5.12, 5.12]$ (Naser *et al.*, 2025). Since one of our aims is to rate the scalable practicality of the quantum algorithms with the limited amount of resources, we set the initial search ranges r relatively larger, by $x_i \in [-10, 10]$ with $r = 10$ and $x_i \in [-20, 20]$ with $r = 20$. For our method, the ranges are continuously changed, depending on the optimization tendency. Each algorithm was tested on both ranges to observe the relative performance in varying size of the search space.

The present experimental evaluation is intended as a proof-of-concept comparison among QGA-based methods under the same simulated-qubit constraint, and it is not designed as a comprehensive benchmarking study against classical continuous optimization algorithms. In particular, we do not intend to include classical adaptive or evolutionary optimizers as baselines for comparison. Therefore, the results reported in this section should be interpreted as evidence of improvement over the selected QGA variants tested here, namely RQGA and SSQGA, rather than as evidence that the proposed method is competitive with classical counterparts.

5.2. *Results*

Figure 3 shows the graphs of optimizing progresses done by our method (titled as QuASAR, an acronym for Quantum Adaptive Search in Approximate Region), RQGA,

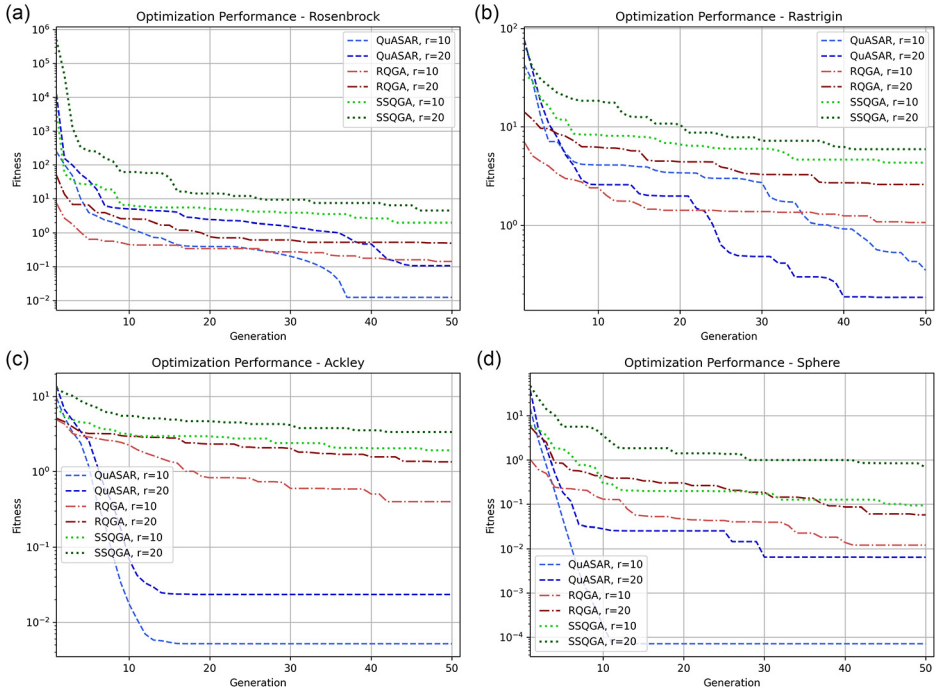

Fig. 3. Graphs of optimization progress on the benchmark functions of (a) Rosenbrock, (b) Rastrigin, (c) Ackley, and (d) Sphere over 50 generations. Dashed blue lines: our method; dashdot red lines: RQGA; dotted green lines: SSQGA. Those with the search range of $r = 10$ are painted with light colours, and the others with $r = 20$ are painted with dark colours.

and SSQGA on the benchmark functions over 50 generations. A logarithmic scale is applied to the Fitness axis to cover the broad range of values. For the functions of Rosenbrock, Ackley, and Sphere, our method with $r = 10$ achieves the lowest minimum found, while for Rastrigin our method with $r = 20$ performs better. Table 2 is provided to list the numerical end results from all three algorithms, each presenting two cases of $r = 10$ and $r = 20$ along with the true global optimum of the functions. The lowest minimum for each function is marked as bold. Note that "Rosen." and "Rastrig." are truncated terms for Rosenbrock and Rastrigin, respectively.

To make a further analysis on the overall qubit requirement of the proposed method and other QGA techniques, we counted the cumulative number of simulated qubits used for each algorithm during optimization. The consumption growth on the Ackley function is plotted on the graph in Fig. 4 as a visual reference, and the final counts are presented in Table 3. Again, the lowest cumulative count for each function is marked as bold, which in every case belongs to our method.

The metric of cumulative qubit usage is used here as a simulator-level measure that reflects how many active qubits are used across generations as the register size changes during the optimization process. The actual hardware-relevant qubit requirement is determined by the maximum active circuit size, which is limited up to 30 qubits for all compared

Table 2
Final best fitness values obtained by the proposed method, RQGA, and SSQGA on the benchmark functions after 50 generations. Values are averaged over 10 independent repetitions. The lowest value for each benchmark function is marked in bold.

End results on fitness				
Methods	Rosen.	Rastrig.	Ackley	Sphere
QuASAR, $r = 10$	**0.0124**	0.3496	**0.0052**	**0.0001**
QuASAR, $r = 20$	0.1068	**0.1852**	0.0234	0.0064
RQGA, $r = 10$	0.1427	1.0625	0.3970	0.0122
RQGA, $r = 20$	0.4909	2.5902	1.3456	0.0580
SSQGA, $r = 10$	1.9780	4.3186	1.9154	0.0937
SSQGA, $r = 20$	4.4878	5.9134	3.3611	0.6832
Global Min.	0.0	0.0	0.0	0.0

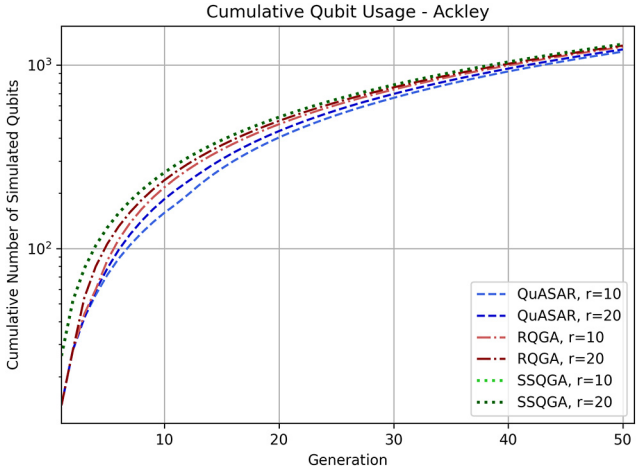

Fig. 4. Cumulative qubit usage for simulated optimization on the Ackley function by our method, RQGA, and SSQGA over 50 generations and 10 repetitions. The line styles and colours are chosen in an identical manner to the graphs in Fig. 3.

methods under the largest tested setting. Therefore, Table 3 should be interpreted together with the final fitness results in Table 2: the main advantage of the proposed method is not a large reduction in maximum qubit count, but an improved optimization accuracy under essentially the same qubit constraint.

5.3. *Analysis*

Our method is implemented within the reduced-QGA structure, which deliberately omits the genetic processes of crossover and mutation for the purpose of qubit sparing, and therefore relies primarily on quantum selection. A likely negative effect of such an approach is reduction to an elitist random sampling, since the result in one generation would barely provide any feedback to the next population (Chiesa *et al.*, 2020). Our method com-

Table 3
Mean cumulative qubit usage by the proposed method, RQGA, and SSQGA on the benchmark functions, with 10 repetitions and 50 generations. The metric counts active simulated qubits over generations. Decimal values appear because the results are averaged over repetitions.

Cumulative qubit usage				
Methods	Rosen.	Rastrig.	Ackley	Sphere
QuASAR, $r = 10$	1252.6	**1219**	**1183**	**1206.4**
QuASAR, $r = 20$	**1249.6**	1228.6	1216.6	1226.8
RQGA, $r = 10$	1265.4	1264	1256.2	1264.4
RQGA, $r = 20$	1273	1273.4	1269.4	1271.6
SSQGA, $r = 10$	1300	1300	1300	1300
SSQGA, $r = 20$	1300	1300	1300	1300

pensates such a deficiency by iterative contraction of the search space towards the global minimum, which contributes to an effective exploration over the space with a limited population size. The effect of the proposed approach is relatively more evident in (c) and (d) in Fig. 3, where our method demonstrates a considerable degree of convergence at the early stages of optimization. For the case of $r = 10$ every algorithm generally performs better, and we speculate that even for the Rastrigin function in (b) our method with $r = 10$ would have eventually outperformed the case of $r = 20$ if a longer series of generations were provided.

SSQGA, for both cases of $r = 10$ and $r = 20$, appears to have performed the worst among the test approaches, as we initially expected. The largest contributing factor to this result, to our knowledge, should be the exceptionally small size of its population, which can only contain 16 chromosomes even with the maximum possible number of the index qubits limited by the experiment setup. While there still is a possibility that its performance would dramatically improve if given a sufficient number of qubits for generating the population, its structural constraint that requires two identical sets of population needs to be resolved for the sake of its cost efficiency, especially with regard to the fact that RQGA, which totally lacks the unique and presupposed advantage of having all the possible solution candidates in its search list, still slightly outperforms it.

Fig. 4 and Table 3 show that the amount of qubit usage saved by adopting our method is rather trivial, and it is expected that the difference would be nearly indistinguishable in case large and complex problems are to be solved. Therefore, the resource advantage of the proposed method should be interpreted carefully. The proposed method does not substantially reduce the maximum qubit count required by the underlying RQGA-style circuit; rather, its advantage lies in using a comparable qubit budget to perform a more precise continuous-domain search through dynamic reuse of the same index register over successively refined local regions.

For a fixed value of n_{idx}, the proposed method uses the same selection-oriented circuit structure as RQGA: the same type of index register, value register, comparator ancilla, oracle marking, and Grover-type amplification are employed. The dynamic encoding scheme changes the classical mapping between index states and coordinate values in the current search region, but it does not introduce additional quantum genetic operators or extra

quantum registers. Consequently, its circuit depth, gate count, number of measurements, and simulation runtime are expected to be of the same order as those of RQGA under the same register size and oracle implementation. When the same encoding scheme is incorporated into other QGA variants, its circuit-level resource requirements would likewise follow the requirements of the corresponding base variant.

It is important to note that the presented resource analysis is based on a simulation, not a hardware-oriented experiment. In particular, there are several key factors, including circuit depth, gate counts, oracle gate complexity, finite-shot measurement statistics, total number of circuit executions, and sensitivity to realistic device noise, that need to be assessed to assure the method's optimality and practicality on real quantum devices. Since the experiments were performed using Qiskit Aer StatevectorSimulator, no finite-shot sampling noise or quantum device noise model was included. Runtime on a classical statevector simulator should be further analysed to be used as an evidence of quantum runtime efficiency, since it currently depends on the simulator implementation and the classical hardware environment. These analyses are left for future work.

6. Conclusion

In this paper, we presented a dynamic-encoding quantum genetic algorithm for continuous optimization under limited quantum resources. The central motivation was that existing QGA designs face a fundamental representation problem when applied to continuous domains: a fixed global discretization requires increasingly many qubits as the desired precision and dimensionality grow, while reproducing crossover and mutation operators cause additional register and circuit-depth overhead. The proposed method combines quantum adaptive search with an iteratively approximated search space by, instead of encoding the entire continuous domain at a fixed resolution, generating a finite grid of coordinate vectors within the current search region, applying quantum adaptive search to select the best candidate, and updating the centre and range of the next search region accordingly with the observed optimization progress.

The proposed approach is positioned between reduced QGA and strictly structured QGA. Similar to RQGA, it avoids explicit crossover and mutation in order to reduce qubit usage and relies on Grover-type selection as the main optimization mechanism, while it does not rely on a fixed finite representation of the whole continuous search space. Compared with SSQGA, the proposed method avoids the cost of maintaining duplicated population registers and implementing quantum crossover and mutation. The expansion mechanism further allows the algorithm to respond to stagnation and partially address the risk of premature localization in multimodal landscapes. The simulation results showed that, under the same qubit constraints, the proposed method achieved the best final fitness values among the tested QGA variants on all benchmark functions. Although the cumulative simulated-qubit savings were modest in the present setting, the results indicate that the proposed dynamic-encoding scheme improves final optimization accuracy over the selected QGA variants under similar maximum simulated-qubit constraints. For demonstrating hardware-level efficiency gain, factors such as circuit depth, gate counts, runtime, and noise sensitivity will need to be evaluated.

The main contribution of this work is proposing a qubit-aware encoding strategy for applying quantum adaptive search to continuous optimization. By decoupling effective coordinate precision from a single fixed global discretization, the proposed method provides a technique to utilize limited index and value registers more efficiently. The presented study was conducted in a noiseless simulation environment and on low-dimensional benchmark functions, so the results should be interpreted as proof-of-concept evidence rather than a demonstration of practical quantum advantage, superiority over classical optimizers, or hardware-level resource efficiency. Future work is expected to examine higher-dimensional problems, noisy quantum simulation and hardware execution, more efficient oracle designs, comparison with classical baselines, and theoretical analysis of query complexity and convergence behaviour. These directions are necessary for establishing the proposed dynamic-encoding QGA as a scalable quantum heuristic optimization framework.

Acknowledgements

This work was supported by Basic Science Research Program of the NRF of Korea funded by the Republic of Korea government (Ministry of Science and ICT) (RS-2024-00347902), and by Regional Innovation System & Educatio n(RISE) Glocal University 30 Program of the Gwangju RISE Center funded by the Republic of Korea government (Ministry of Education) and Gwangju Metropolitan City, Republic of Korea ((2026-RISE(Glocal University 30)-05-011)).

References

Abraham, H., AduOffei, Agarwal, R., et al. (2019). Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2562110.

Acampora, G., Schiattarella, R., Vitiello, A. (2022). Using quantum amplitude amplification in genetic algorithms. *Expert Systems with Applications*, 209, 118203. https://doi.org/10.1016/j.eswa.2022.118203.

Ackley, D.H. (1987). *A Connectionist Machine for Genetic Hillclimbing*. PhD dissertation, University of Michigan, Ann Arbor, MI.

Ardelean, S., Udrescu, M. (2024). Hybrid quantum search with genetic algorithm optimization. *PeerJ Computer Science*, 10, e2210. https://doi.org/10.7717/peerj-cs.2210.

Baritompa, W.P., Bulger, D.W., Wood, G.R. (2005). Grover's quantum algorithm applied to global optimization. *SIAM Journal on Optimization*, 15(4), 1170–1184. https://doi.org/10.1137/040605072.

Chiesa, M., Maioli, G., Colombo, G.I., Piacentini, L. (2020). GARS: Genetic Algorithm for the identification of a robust subset of features in high-dimensional datasets. *BMC Bioinformatics*, 21(51). https://doi.org/10.1186/s12859-020-3400-6.

Dürr, C., Hoyer, P. (1996). A Quantum Algorithm for Finding the Minimum. *CoRR*, arXiv:quant-ph/9607014.

Farhi, E., Goldstone, J., Gutmann, S. (2014). A Quantum Approximate Optimization Algorithm. arXiv: Quantum Physics. https://api.semanticscholar.org/CorpusID:118149905.

Giri, P.R., Korepin, V.E. (2017). A review on Quantum Search Algorithms. *Quantum Information Processing*, 16(12), 1–36. https://doi.org/10.1007/s11128-017-1768-7.

Goldberg, D.E. (1988). *Genetic Algorithms in Search, Optimization and Machine Learning*, 13 ed. Addison-Wesley Professional.

Grover, L.K. (1996). A fast Quantum Mechanical Algorithm for database search. In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96*. Association for Computing Machinery, New York, NY, USA, pp. 212–219. https://doi.org/10.1145/237814.237866.

Han, K.-H., Kim, J.-H. (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, 6(6), 580–593. https://doi.org/10.1109/TEVC.2002.804320.

Hoffmeister, F., Bäck, T. (1991). Genetic Algorithms and evolution strategies: similarities and differences. In: Schwefel, H.-P., Männer, R. (Eds.), *Parallel Problem Solving from Nature*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 455–469.

Lahoz-Beltra, R. (2023). The Conquest of Quantum Genetic Algorithms: The Adventure to Cross the Valley of Death. https://arxiv.org/abs/2401.08631.

Malossini, A., Blanzieri, E., Calarco, T. (2008). Quantum Genetic Optimization. *IEEE Transactions on Evolutionary Computation*, 12(2), 231–241. https://doi.org/10.1109/TEVC.2007.905006.

McClean, J., Boixo, S., Smelyanskiy, V., Babbush, R., Neven, H. (2018). Barren Plateaus in quantum neural network training landscapes. *Nature Communications*, 9, 4812. https://www.nature.com/articles/s41467-018-07090-4.

Naser, M.Z., Al-Bashiti, M.K., Tapeh, A.T.G., Naser, A., Kodur, V., Hawileh, R., Abdalla, J., Khodadadi, N., Gandomi, A.H., Eslamlou, A.D. (2025). A review of benchmark and test functions for global optimization algorithms and metaheuristics. *Wiley Interdisciplinary Reviews: Computational Statistics*, 17(2), 70028. https://doi.org/10.1002/wics.70028.

Nielsen, M.A., Chuang, I.L. (2004). *Quantum Computation and Quantum Information* 1st ed. Cambridge University Press.

Picheny, V., Wagner, T., Ginsbourger, D. (2013). A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48, 607–626. https://doi.org/10.1007/s00158-013-0919-4.

Preskill, J. (2018). Quantum Computing in the NISQ era and beyond. *Quantum*, 2, 79. https://doi.org/10.22331/q-2018-08-06-79.

Rosenbrock, H.H. (1960). An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3), 175–184. https://doi.org/10.1093/comjnl/3.3.175.

SaiToh, A., Rahimi, R., Nakahara, M. (2014). A quantum genetic algorithm with quantum crossover and mutation operations. *Quantum Information Processing*, 13, 737–755.

Sofge, D.A. (2008). Prospective algorithms for quantum evolutionary computation. In: *Proceedings of the Second Quantum Interaction Symposium (QI-2008)*. College Publications, Oxford, UK, pp. 26–28.

Sun, Y., Xiong, H. (2014). Function optimization based on quantum genetic algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, 7, 144–149. https://doi.org/10.19026/rjaset.7.231.

Tanabe, R., Ishibuchi, H. (2020). A review of evolutionary multimodal multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 24(1), 193–200. https://doi.org/10.1109/TEVC.2019.2909744.

Udrescu, M., Prodan, L., Vlăduţiu, M. (2006). Implementing quantum genetic algorithms: a solution based on grover's algorithm. In: *Proceedings of the 3rd Conference on Computing Frontiers, CF '06*. Association for Computing Machinery, New York, NY, USA, pp. 71–82. https://doi.org/10.1145/1128022.1128034.

Yanofsky, N.S., Mannucci, M.A. (2008). *Quantum Computing for Computer Scientists*, 1st ed. Cambridge University Press.

Zhang, G. (2011). Quantum-inspired evolutionary algorithms: a survey and empirical study. *Journal of Heuristics*, 17(3), 303–351. https://doi.org/10.1007/s10732-010-9136-0.

J.S. Kim received his bachelor's degree in Physics from University of Illinois in Urbana-Champaign, USA, in 2016 and received his master's and doctoral degrees in quantum computing from Gwangju Institute of Science and Technology (GIST), Republic of Korea, in 2019 and 2025, respectively. He is currently working as a post-doctoral researcher in the Department of AI and the Institute for AI in GIST. His main topics of research are quantum heuristic optimization, quantum reinforcement learning, and quantum AI.

M.-J. Kim is an associate professor in the Department of AI at Chonnam National University, Republic of Korea. He received the PhD degree from the Artificial Intelligence Graduate School, Gwangju Institute of Science and Technology (GIST), Republic of Korea in 2023. His research interests span game artificial intelligence, reinforcement learning, human-computer interaction, and machine learning. Kim is recognized for his work on evolutionary reinforcement learning algorithms and AI systems that integrate human-centric decision-making frameworks.

C.W. Ahn is a professor in the Department of AI at Gwangju Institute of Science and Technology (GIST), Republic of Korea. He received a PhD degree from the Department of Information and Communications at GIST in 2005. From 2005 to 2007, he worked in Samsung Advanced Institute of Technology, Korea. From 2007 to 2008, he was a research professor at GIST. From 2008 to 2016, he was an assistant/associate professor at the Department of Computer Engineering, Sungkyunkwan University (SKKU), Republic of Korea. His research interests include genetic algorithms/programming, multi-objective optimization, neural networks, and quantum machine learning.