Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Genuinely Quantum Algorithms for Bio-inspired Meta-heuristic Optimization

Asimakis Kydros

akydros@csd.auth.gr

Aristotle University of Thessaloniki

Nikolaos L. Tsitsas

Aristotle University of Thessaloniki

Nikos Konofaos

Aristotle University of Thessaloniki

Systematic Review

Keywords: Quantum Algorithms, Meta-heuristic Optimization, Quantum Genetic Algorithm (QGA), Quantum Ant Colony Optimization (QACO), Quantum Particle Swarm Optimization (QPSO)

Posted Date: November 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8123535/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Genuinely Quantum Algorithms for Bio-inspired Meta-heuristic Optimization

Asimakis Kydros[1*], Nikolaos L. Tsitsas[1] and Nikos Konofaos[1]

[1]School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece.

*Corresponding author(s). E-mail(s): akydros@csd.auth.gr;
Contributing authors: ntsitsas@csd.auth.gr; nkonofao@csd.auth.gr;

Abstract

Quantum computation offers a promising pathway toward advancing meta-heuristic optimization, particularly in Nature-inspired paradigms such as genetic algorithms and swarm intelligence. This study conducts a systematic literature review of fifteen major meta-heuristics to assess current efforts toward their quantum realization. Emphasis is placed on identifying genuinely quantum proposals (those that can be implemented on quantum hardware) as opposed to quantum-inspired analogues that remain classically simulable. The review highlights the present landscape, emerging algorithmic strategies, and open challenges that define the path toward truly quantum meta-heuristic optimization.

Keywords: Quantum Algorithms, Meta-heuristic Optimization, Quantum Genetic Algorithm (QGA), Quantum Ant Colony Optimization (QACO), Quantum Particle Swarm Optimization (QPSO)

1 Introduction

Optimization lies at the core of scientific and engineering methodology. Meta-heuristic algorithms, and particularly those inspired by Nature and biological processes, have long served as powerful, domain-agnostic strategies for tackling a wide range of complex problems. The Genetic Algorithm (GA) and Swarm Intelligence (SI)-based techniques, such as Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO), are among the most prominent examples of the abundant utility this class

of optimization algorithms can offer (Holland 1992; Dorigo and Gambardella 1997; Katoch et al. 2021).

At the same time, the rapid progress of Quantum Computing (QC) has opened the possibility of re-imagining classical computation through fundamentally quantum mechanisms. Quantum algorithms have already demonstrated advantages in certain problems, notably in unstructured search and period finding (Grover 1996, 1998; Shor 1994, 1997), motivating researchers to explore whether similar benefits can extend to established optimization strategies. Early attempts on marrying QC with meta-heuristic optimization have produced preliminary mappings of relevant subroutines, predominantly GA-related ones, into quantum operators, albeit often in inconclusive ways (Ge et al. 1998; Rylander et al. 2001; Giraldi et al. 2004).

Despite growing interest, this niche remains small and scattered, surrounded by numerous algorithms labeled as "quantum" that, upon closer inspection, fall under the intrinsically classical philosophy of quantum-inspired computing. While several reviews on quantum optimization covering meta-heuristics exist (Lahoz-Beltra 2016; Montiel Ross 2020; Siddi Moreau et al. 2025; Ur Rehman et al. 2025; Acampora et al. 2026) none examines in detail the genuinely quantum proposals or clarifies their distinction from quantum-inspired counterparts. This lack of analysis can lead to ambiguity regarding the limits of truly quantum meta-heuristics.

This work aims to address that gap by conducting a methodical and thorough study of quantum-implementable algorithms for bio-inspired meta-heuristic optimization. Its goals are to (i) highlight the challenges inherent in translating classical optimization procedures into realizable quantum algorithms, (ii) clarify the distinction between quantum-native and quantum-inspired approaches, and (iii) present, in a comprehensive and structured manner, existing attempts to define quantum analogues for established meta-heuristics. Through this analysis, the study aims to illuminate what is currently achievable within the algorithmic paradigm of QC, assess the potential scale of existing proposals, and inspire further research toward genuinely quantum designs.

The rest of this paper is structured as follows. Section 2 describes the research protocol and methodology. Section 3 discusses current trends in the considered field observed by bibliometric analysis on the indexed data. Section 4 studies the distinction between quantum-implementable and quantum-inspired works. Section 5 examines the state of the art in quantum-native proposals for meta-heuristic optimization, specifically quantum implementations of GA, ACO and PSO. Section 6 consolidates the findings and outlines broader perspectives and future challenges. Further supporting details regarding GA, ACO, PSO, QC, Grover-based approaches and Quantum Annealing (QA) can be found in Appendices A, B, C, D, E and F, respectively.

2 Research protocol

In this section, the outline of the review is presented and the main and secondary questions are defined. The searching procedure employed is explained by deciding on the study parameters, producing appropriate search queries, choosing the best scientific databases, setting filters and detailing the screening process.

Table 1: The 15 meta-heuristics considered in this study, ordered by final deduplicated counts per type. This ordering also indirectly implies popularity in the literature and in the community. Symbols in `NAME` are being used to conserve space. For GA, counts read "346+8", indicating that 8 papers were reached via citation chaining. It similarly follows for PSO.

Meta-heuristic	`NAME`	`ID`	Counts
Genetic Algorithms	"genetic"	"GA or QGA"	346+8
Particle Swarm Optimization	"particle swarm"	"PSO or QPSO"	141+1
Ant Colony Optimization	"ant colony"	"ACO or QACO"	98
Artifical Bee Colony Optimization	"(artificial)? bee colony"[a]	"ABC or QABC"	34
Grey Wolf Optimization	"[gray, grey] wolf"[b]	"GWO or QGWO"	25
Cuckoo Search Algorithms	"cuckoo search"	"CSA or QCSA"	23
Whale Optimization	"whale"	–	21
Bat Algorithm	"bat"	–	16
Firefly Algorithms	"firefly"	–	13
Salp Swarm Optimization	"salp swarm"	–	12
Artificial Fish Swarm Optimization	"(artificial)? fish swarm"[a]	–	8
Krill Herd Algorithms	"krill herd"	–	8
Brain Storm Optimization	"brain storm"	–	5
Dragonfly Algorithms	"dragonfly"	–	5
Grasshopper Optimization	"grasshopper"	–	5

[a] $(\cdot)$? indicates the input is optional (included once or not at all). E.g., "(artificial)? bee colony" unwraps to ("artificial bee colony" OR "bee colony").
[b] $[\cdot, \cdot]$ means one or the other but not both. E.g., "[gray, grey] wolf" unwraps to ("gray wolf" OR "grey wolf").

This work aims to pinpoint, with as much scientific rigor as possible, an answer to the following question:

MQ *"How can bio-inspired meta-heuristic optimization be realized on quantum computers?"*

A vague and quite non-specific question to answer. In order to ensure a structured and transparent process, this study realizes a systematic literature review (Kitchenham 2004) employing the guidelines of the updated PRISMA 2020 protocol (Haddaway et al. 2022). MQ is split into multiple, secondary constituent questions that offer more specificity and facilitate their scientific tackling. Those are the following:

RQ1 *"Which bio-inspired meta-heuristics have already been studied for quantum translation?"*
RQ2 *"Which of those works propose a genuinely quantum design and how can they be differentiated from the quantum-inspired ones?"*
RQ3 *"What are the algorithmic strategies and approaches that have already been studied in the literature for genuinely quantum meta-heuristic optimization?"*
RQ4 *"Which paths hold the greater promise for the future evolution of the field of quantum meta-heuristic optimization?"*

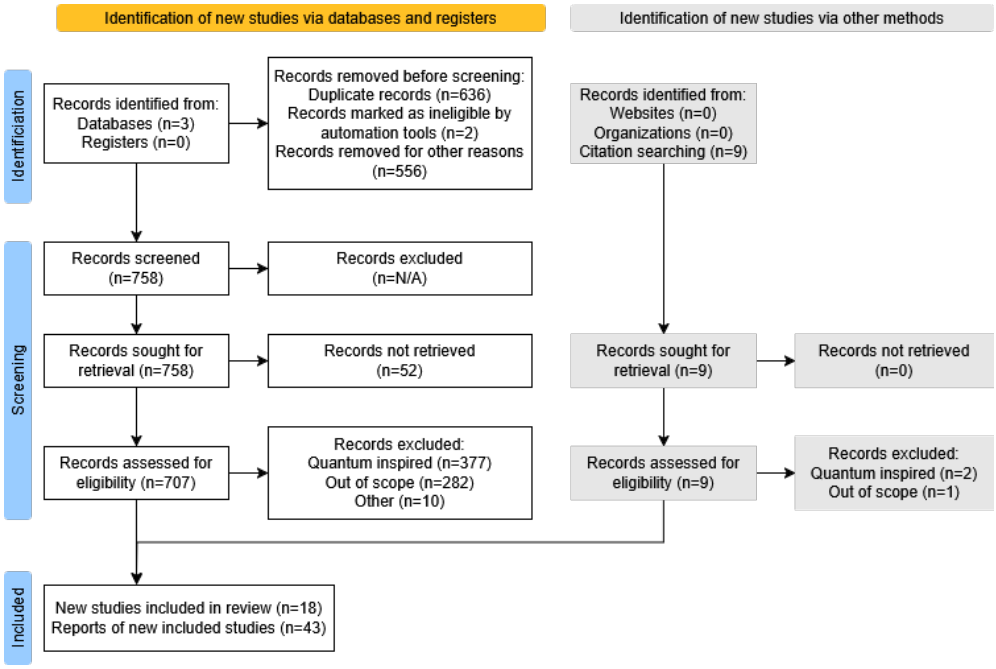


Fig. 1: The PRISMA2020 flow chart describing the literature survey process realized in this work

The scope of this literature review is dictated by RQ1. In order to ensure the inclusion of a satisfactory large part of the field, 15 of the most popular meta-heuristics have been considered. Their names can be found in Table 1. The search flow is summarized in Fig. 1 and explained in the following. Two databases were mainly consulted, Web of Science (WoS) and Scopus. They were elected due to their popularity with the scientific community, their mission of including a wide range of works that are nonetheless demonstrably rigorous, as well as their offering of multiple tools that facilitate searching. For all considered meta-heuristics, these databases were accessed on the 16th of July, 2025.

The principal query format for every meta-heuristic was chosen in such a way that ensured a wide range of works were indexed for each optimization strategy. In this pretext, big literals such as *"quantum genetic algorithm"* were avoided, in favor of *"quantum" AND ("genetic" OR "GA" OR "QGA")*, that enabled flexibility while largely staying on-topic. At the same time, care was given in eliminating a large amount of irrelevant works prior to screening, with the inclusion of QC-related keywords, such as *"quantum circuit"*, *"quantum computation"* and *"quantum algorithm"*, but also with the strict exclusion of quantum-inspired works. The latter was achieved by appending the queries with the negation of a small list of titles that allude to a quantum-inspired approach, some of those being *"quantum-inspired"*, *"quantum-enhanced"* and *"quantum-behaved"*. Thus the main skeleton of the principal queries for WoS and Scopus respectively were defined as:

W1 *TOPIC("quantum" and (NAME or ID)) AND TOPIC("quantum comput*" or "quantum circuit*" or "quantum annealing" or "quantum algorithm*" or "quantum hardware" or "quantum processor*" or qpu or "qiskit" or "d-wave" or "pennylane" or "ibm" or "qasm" or "cirq") AND TOPIC("optimi*") AND NOT TITLE("quantum-inspired" OR "quantum inspired" OR "quantum-enhanced" OR "quantum enhanced" OR "quantum-behaved" OR "quantum behaved" or "quantum assisted" or "quantum-assisted") AND NOT KEY("quantum-inspired" OR "quantum inspired" OR "quantum-enhanced" OR "quantum enhanced" OR "quantum-behaved" OR "quantum behaved" or "quantum assisted" or "quantum-assisted") AND NOT TOPIC("circuit design")*

S1 *ABS-TITLE-KEY(quantum and (NAME or ID)) AND ABS-TITLE-KEY("quantum comput*" or "quantum circuit*" or "quantum annealing" or "quantum algorithm*" or "quantum hardware" or "quantum processor*" or qpu or qiskit or d-wave or pennylane or ibm or qasm or cirq) AND ABS-TITLE-KEY("optimi*") AND NOT TITLE("quantum-inspired" OR "quantum inspired" OR "quantum-enhanced" OR "quantum enhanced" OR "quantum-behaved" OR "quantum behaved" or "quantum assisted" or "quantum-assisted") AND NOT KEY("quantum-inspired" OR "quantum inspired" OR "quantum-enhanced" OR "quantum enhanced" OR "quantum-behaved" OR "quantum behaved" or "quantum assisted" or "quantum-assisted") AND NOT ABS-TITLE-KEY("circuit design")*

The placeholders `NAME` and `ID` are specific to each meta-heuristic. In Table 1, a mapping between the considered meta-heuristics and their `NAME`/`ID` values is given. The authors provide a spreadsheet compiling all the covered queries, along with the discovered data, for further verbosity.

In the above queries, the main goal was to detect works that have each meta-heuristic, in quantum form, as their core topic, while discarding unwanted keywords from the title and keywords, as previously detailed. Initial experimental queries revealed that also removing quantum-inspired-esque keywords from the abstract is undesirable, as it enhanced the danger of discarding papers with sentences similar to *"this is not a quantum-inspired algorithm"*. Furthermore, it was found that works focusing on the problem of quantum or reversible circuit design that included the considered meta-heuristics were strongly linked with quantum-inspired works, and therefore were also skipped.

Not all meta-heuristics enjoy the same popularity. For optimization strategies that indexed few listings under the previous two queries, a supplementary, even more specific query was made on WoS. The threshold for this extra query was arbitrarily set at less than 150 works collectively under application of the inclusion criteria. All meta-heuristics, barring GA and PSO, were therefore enriched by the following query:

W2 *TOPIC("quantum NAME" and ("optimi*" or "algorithm")) AND NOT TOPIC("quantum-inspired" or "quantum inspired" or "quantum-behaved" or "quantum behaved")*

In the above, the aim was to target works exactly referencing quantum versions of the considered strategies. `NAME` is defined as in the two previous queries. Notably, the field of quantum-inspired optimization design is old and larger than the field of realizable quantum algorithms, and a great amount of inspired works traditionally use the format "quantum `NAME`" to title themselves. For this reason, in order to compensate, the exclusion of inspired-related keywords was made more aggressive by being applied on the abstracts as well.

Finally, in order to ensure no relevant but currently unpublished works were being ignored, arXiv was also consulted, to a lesser extent. This is because arXiv mainly includes works that haven't undergone official peer-review process, therefore overly relying on said database had the potential to hinder the quality of the findings and the review overall. The arXiv search engine greatly appreciates simplicity for its queries, thus the search was realized as:

X1 *all:(quantum and `NAME`)*

The above was applied to all meta-heuristics, with the exception of GA that was found to give more qualitative entries under the query *all:("quantum genetic")* and Grey Wolf optimization that required the query be written as *all:(quantum and "grey wolf") or all:(quantum and "gray wolf")*. All arXiv queries were realized on the 29th of July, 2025. Once again, `NAME` is defined as in Table 1.

A relatively lenient number of inclusion criteria was defined for this study. This is because it was deemed important to encapsulate as much of the field as possible, therefore even borderline-relevant works would be desirable.

Considered papers had to satisfy three rules in order to continue to the screening phase. In order of application, these where:

R1 *The work must be either an article published on a journal or a conference paper.*
R2 *The paper must be written in English.*
R3 *The paper must be listed under a field category relevant to computer science and/or algorithmic design, when applicable.* Consult the provided spreadsheet for the category lists enforced for each query.

These filters aimed to reduce the work pile by discarding inaccessible manuscripts or studies on fields alien to this review's focus, instead favoring original research in the area of algorithm design. Collectively, these rules excluded 556 entries from the pile.

Following the application of the above filters, deduplication was performed. The tool employed for it was BibTex Tidy (West and contributors 2025). Deduplication was realized by first grouping and reducing the entry loads per meta-heuristic, and afterwards by lumping all the reduced groups together and reducing again. In total, 636 records were removed. Final reduced count per group can be found in Table 1. In total, 760 works were collected. During screening, an extra 8 works exhbiting relevance for GA and 1 for PSO were identified via citation chaining, raising the total to 769. JabRef (Kopp et al. 2023) was used for the bibliography management, its internal flagger highlighted 2 works as retracted, yielding the final total of 767 works.

The screening process deviated slightly from the norm. It is in the nature of the examined field that real "quantumness" cannot be easily judged by the title, keywords or abstract alone. This is even more true when easy-to-spot cases, like the phrase "quantum-inspired" in the title, are discarded automatically by the selected queries. The discipline of meta-heuristic optimization enriched with quantum ideas has a strong tradition in labeling itself as "quantum" without any extra adjectives or buzzwords, even when this enrichment is analogical or a classical interpretation. For this reason, it was elected to perform a kind of "skim" screening to better judge candidates. Records were skimmed piecemeal for any "telling" points in their proposed algorithmic process that indicate non-implementability in real quantum devices. Concretely, skim searching was performed as follows: every entry that survived the inclusion criteria was manually skimmed with the aim to detect one or multiple of the following cases:

1. Access of theoretical amplitudes that violates the principles of quantum mechanics
   - Update rules that depend on qubit amplitudes
   - Hard-coded angle update lookup tables that depend on amplitudes, directly or indirectly
   - Interpreting chromosomes as "double chains"
   - State-to-problem space transformation via amplitude-dependent functions
   - Operation of quantum gates on a seemingly measured (and thus collapsed) quantum register
   - Measuring a quantum register on each iteration by an amplitude-based classical rule or without explicitly stating reconstruction
   - Presence of the "interference crossover" operator or invoking quantum interference as a quantum operation to be performed
2. Application of the Schrödinger equation for population update, especially if it is based solely on classical calculus and details no methodology to apply it with quantum tools
3. Explicit correlation or strong resemblence to the Quantum Evolutionary Algorithm
4. Mention of the unexpected terms "revolving door operator", "Q-bits" or "quantum non-gate"
5. Absence of multi-qubit gates and thus entanglement
6. No quantum circuit schematic being presented or otherwise described

The above criteria are empirical heuristics that were decided upon subjectively, and after repeated examination of the screenable paperload that highlighted common trends that indicated counterfactual algorithmic design with regards to established QC principles. A more detailed analysis on the above heuristics and how they relate to quantum-inspired works is given in Sec. 4. These heuristics are presented in descending priority order. For works that propose direct access to amplitudes, it is significantly more convincing to conclude that they are not quantum-native than are ones that simply do not contain a quantum circuit, for example. It is important to realize that these deciding factors are informed of quantum computing principles but remain subjective,

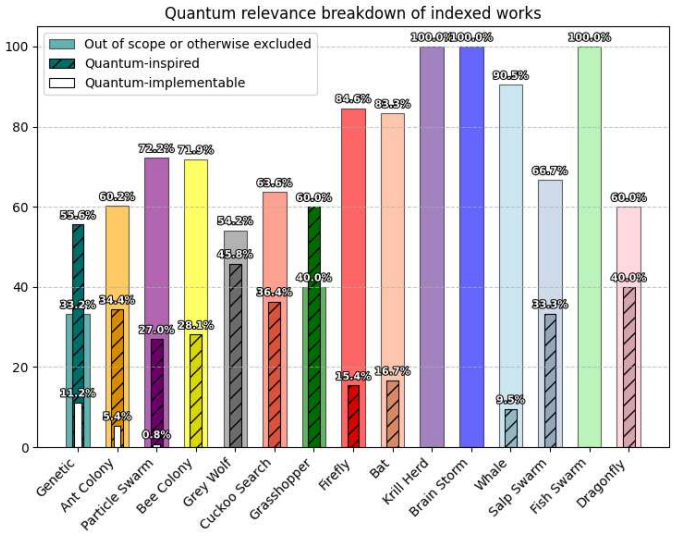


Fig. 2: Relevance comparison of works per meta-heuristic by percentage. GA is the only category not overwhelmed by inspired proposals. The loud presence of inspired proposals in all meta-heuristic categories implies a desire of the community for genuine quantum utility

and as such they will unavoidably reflect the bias of the authors of this work. In order to counteract this slightly, the skim screening process was more lenient for heuristics further down the list. Allowing ambivalent entries to pass to the full screening phase reduced the risk of omitting relevant studies.

The skim screening process sought all 767 papers. Of those, 52 were either lost or behind paywalls. Of the remaining, 379 where judged quantum-inspired in their nature by skimming, while 283 were found out of scope for this study. "Out of scope" included proposals indcluding different meta-heuristics than the 15 considered, pipelines of fully classical versions of meta-heuristics partnered with quantum-inspired versions of others, works that focused on applicability without analyzing the applied algorithm, and instances of a classical optimizer being employed for a quantum-related task, among others. A small group of 10 records were also excluded due to methodological flaws or insufficient analysis. In total, 43 works were deemed relevant, describing 15 standalone algorithms or subroutines and 3 theoretical studies.

3 Bibliometric analysis

In this section, an answer to RQ1 is given, namely *"Which bio-inspired meta-heuristics have already been studied for quantum translation?"*. Trends in the field of quantum meta-heuristic optimization are brought to the surface via data analysis.

Figure 2 pictures the relevance of the collected papers to quantum-implementability per considered meta-heuristic. The overwhelming presence of quantum inspired ideas

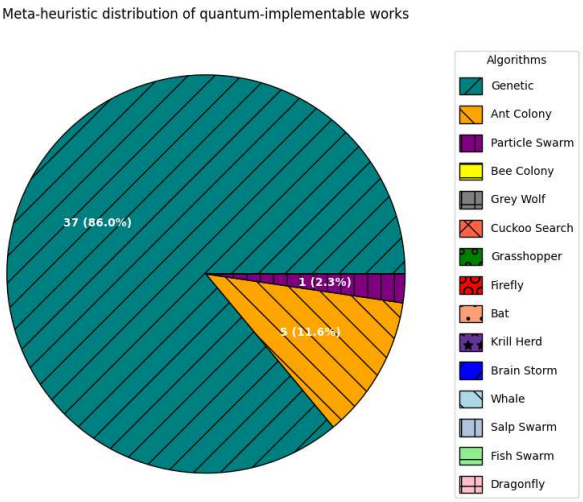


Fig. 3: The distribution of examined meta-heuristics in works proposing real quantum versions as revealed by screening

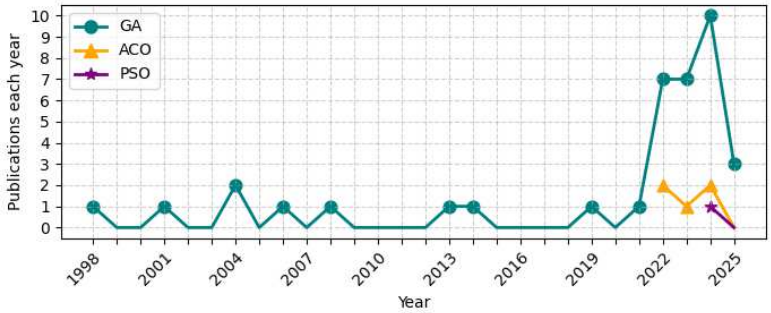


Fig. 4: Works published each year per meta-heuristic. This timeline covers the quantum-implementable studies only. The results indicate that interest on the field is currently rising, as all curves peak on 2024

becomes apparent. This phenomenon implies a strong desire from the community to improve meta-heuristic optimization with quantum ideas, but the limited presence of genuinely quantum attempts comes in contrast to that desire. Given that most attempts appear on GA and a few on ACO, one could conclude that these algorithms are prioritized due to popularity, significance, or even perceived simplicity in translation. Indeed, PSO may present a greater intuition challenge for mapping to a quantum

process, due to its complex state-update mechanisms, especially in the current era of QC where even number addition is non-trivial.

The pie chart of Fig. 3 highlights the distribution of the perceived quantum-implementable works by meta-heuristic. The significant inclination of the community towards GA becomes apparent. This should not be surprising, as in the classical regime, GA is the oldest and arguably the most popular meta-heuristic, enjoying wide application due to strong flexibility and problem adaptability. ACO is the second and, at the time of writing, only other meta-heuristic to exhibit a noticeable attempt at realizing a quantum analogue. As previously suggested, this could be due to popularity of the meta-heuristic in the classical case, or perceived simplicity in its quantum translation. Interestingly, PSO indexes only a single work that explicitly attempts to gauge how the meta-heuristic could be translated as a quantum or hybrid algorithm (Clerc 2024).

Figure 4 visualizes the temporal evolution of the field. Quantum Genetic Algorithm (QGA) had humble beginnings from very early on, before the millennium shift. Then it sporadically made short appearances, until it quickly rose in popularity around 2022. This is also the time Quantum Ant Colony Optimization (QACO) proposals start to appear. One can notice that both meta-heuristics peaked in interest by the QC community very recently, 2024 in fact. The year 2025 also hosted the aforementioned first, and currently only, study on Quantum Particle Swarm Optimization (QPSO). It can therefore be inferred that the field of quantum-realizable meta-heuristic algorithm design is on the rise right now. The foundations are few but strong, and interest is increasing, making this focus area an attractive niche for interested researchers.

4 Distinguishing the Quantum-inspired from the Quantum

In this section, an answer to RQ2 is given, namely *"Which of those works propose a genuinely quantum design and how can they be differentiated from the quantum-inspired ones?"*. Light is shined on which quantum-inspired techniques commonly appear in the literature for meta-heuristic optimization and what constitutes them unimplementable as quantum algorithms. Moreover, some usual signs are identified, that reveal a reportedly "quantum" technique as an inherently classical procedure.

Classical meta-heuristic optimization, for its many benefits, has a tendency of slow or premature convergence and succumbing to local optima (Mitchell 1998). Many methods have been proposed and attempted as a solution to this problem. The introduction of quantum mechanical ideas, with their probabilistic nature and exotic attributes, was also aimed as a convergence improvement. The superposition of quantum systems (see Appendix D) becomes an attractive choice for a feature mapping, as it inherently implies an exponential inner diversity per unit considered in the computation. Furthermore, quantum tunneling effects, a seemingly paradoxical fact of nature, provides a theoretical path for an optimizer through some search space, with avenues of escaping from suboptimal points. It is therefore not surprising that the development of quantum-inspired optimizers enjoys so much attention.

The field of quantum-inspired meta-heuristic optimization is as old as that of the quantum-implementable algorithms (Narayanan and Moore 1996). Coupled with the fact that QC was still in its infancy back then and poorly understood, this lead to quantum-inspired works often inheriting the label "quantum" while being demonstrably non-quantum algorithms. While many works realize the confusion and properly categorize themselves as non-quantum, this occurrence is still prevalent in the literature. As such, it is important to be aware of what constitutes an algorithm un-realizable on a quantum computer, as Science currently understands them, in order to spot such miscategorizations.

A very popular strategy in defining a quantum-inspired optimizer is to use the Quantum Evolutionary Algorithm (QEA) as its basis template (Han and Kim 2000; Vlachogiannis and Ostergaard 2009; Mousavi et al. 2018; Wang et al. 2023). QEA, originally proposed under the name Genetic Quantum Algorithm (GQA) (Han and Kim 2000) is a GA-like process enriched with elements from QC. The procedure is given in Alg. 1. The main idea of this process is to delegate a single qubit for each gene and then continuously rotate these qubits until the register's collapse converges to optimal chromosome(s). The angle of rotation is selected according to the fitness value of the best measured candidate solution at each iteration.

Algorithm 1: Quantum(-inspired) Evolutionary Algorithm

Data: m the population size, n the chromosome size, f the target fitness function, $\mathcal{T}$ termination criteria, $\mathbf{L}$ a classical routine that generates angles from binary gene data.

Result: An individual that optimizes f.

$t \leftarrow 0$

$Q(t) \leftarrow m$ chromosomes as $\left[\begin{array}{c|c|c|c} \alpha_1^{(0)} & \alpha_2^{(0)} & \cdots & \alpha_n^{(0)} \\ \beta_1^{(0)} & \beta_2^{(0)} & \cdots & \beta_n^{(0)} \end{array}\right] : \alpha_j^* \alpha_j + \beta_j^* \beta_j = 1 \ \forall j$

Measure $Q(t)$ obtaining $P(t)$, a classical population containing m n-sized binary individuals

Calculate fitness values for $P(t)$ classically using f

$b \leftarrow$ The best individual among $P(t)$

repeat

  $t \leftarrow t + 1$

  Measure $Q(t-1)$ obtaining $P(t)$, a classical population containing m n-sized binary individuals

  Calculate fitness values for $P(t)$ classically using f

  Obtain $\Delta\theta_j$ angles per qubit j via $\mathbf{L}$

  Update $Q(t)$ qubit-wise via the rotation gate

$$U_j(\Delta\theta_j) = \begin{bmatrix} \cos\Delta\theta_j & -\sin\Delta\theta_j \\ \sin\Delta\theta_j & \cos\Delta\theta_j \end{bmatrix}$$

  $b \leftarrow$ The best individual among $P(t)$

until $\mathcal{T}$

return b

QEA bases itself on quantum superposition, positing that n-sized chromosomes represent 2^n inner individuals. This has the effect of increasing the population's diversity, effectively sizing it up to $m2^n$ individuals at the explicit cost of $\mathcal{O}(mn)$ computational units, while also allowing for the parallel manipulation of multiple chromosomes simultaneously, by operating on the superposition as a whole. Manipulating the superposition is effected by the operator $U_j(\Delta\theta_j)$, which biases the amplitudes towards a single basis state, to the detriment of the others. Thus it can be argued that QEA is not only able to converge to the optimal bitstring, for correct values $\Delta\theta_j$, but it also favors the two principles of exploration and exploitation at the same time. It is a versatile strategy that enjoys wide application, and it is often paired with classical meta-heuristic formulas that further inform the angles $\Delta\theta_j$, so as to realize meta-heuristics other than GA (Wei and Yu 2023; Zhu et al. 2025; Feng et al. 2024).

Its inherent non-quantumness becomes apparent when diving into the specifics of the algorithm. First and foremost, it can be reasonably inferred that the procedure assumes and invokes no entanglement. This is evident by the chosen chromosome format and the fact that only single-qubit operators are ever utilized, consult Appendix D.2 for more details. Entanglement is what drives parallel computation in a way that promises quantum advantage, so its exclusion casts doubt on the usefulness of a quantum computer for the optimization process. At the same time, the fact that the population always remains $\Theta(mn)$ in size and no entanglement ever appears to change that signals that QEA is efficiently simulable on classical computers. Simulation would therefore be more sensible as it would avoid noise considerations and other problems raised by real QC. Notably, there have been instances of quantum-inspired proposals being translated to quantum circuits (Yoshio Josué Rubio 2021; Rosales et al. 2024; Saini et al. 2025), and experiments show that they perform better in simulation rather than in real devices (Yoshio Josué Rubio 2021; Rosales et al. 2024).

QEA proposals lose quantum implementability when they imply free access of qubit amplitudes. Considering population measurement, remember that the act of observing a quantum system irretrievably collapses its quantum state (see Appendix D.4). Strictly studying Alg. 1 as a quantum process, the very first measurement inside the repeat loop is impossible, as $Q(0)$ was already measured at the initialization phase. The same is true for every repeat iteration, as the update step, described as the application of a quantum operator, cannot happen on $Q(t)$ that has seemingly already collapsed. It could be argued that the operation could happen on the collapsed qubit-encoded population $\tilde{Q}(t) \cong P(t)$, as the qubits do not magically vanish after measurement, they just describe classical information. This is true, but that is clearly not what the algorithm aims to do in order to converge. Indeed, Alg. 1 attempts to iteratively rotate the amplitudes in order to favor collapse towards increasingly more fit bitstrings; therefore, the amplitudes need to survive after every iteration. Another argument could be that these points imply the reconstruction of the previous superposition. If the initial population is created via $U_j(\cdot)$ again[1] then this is very much possible. One can simply keep track of the used angles for each qubit and reuse them. But then, it once again

[1] This is reasonable to assume because state initialization using U_j-like gates is standard practice in QC (called *angle encoding*), and the most common initialization $\alpha_j^{(0)} = \beta_j^{(0)} = \frac{1}{\sqrt{2}}$ is possible via $U_j(\frac{\pi}{4})\,|0\rangle$.

Table 2: A popular implementation of **L** for Alg. 1 as a lookup table. f is the fitness function, b is the current best individual and x is some individual in $P(t)$ to be updated. α_j, β_j are the amplitudes of qubit j for x in $Q(t)$. $\top$ and $\bot$ represent logical truth and falsehood, respectively. The product $|\Delta\theta_j| s(\alpha_j, \beta_j)$ gives the desired angle $\Delta\theta_j$. $s(\alpha_j, \beta_j)$ determines the direction of the rotation angle, its dependency to the qubit amplitudes makes this routine inherently classical. This table was inspired by Han and Kim (2000) and recreated here.

x_i	b_i	$f(x) \geq f(b)$	$\lvert\Delta\theta_j\rvert$	$s(\alpha_j, \beta_j)$			
				$\alpha_j\beta_j > 0$	$\alpha_j\beta_j < 0$	$\alpha_j = 0$	$\beta_j = 0$
0	0	$\bot$	0	0	0	0	0
0	0	$\top$	0	0	0	0	0
0	1	$\bot$	0	0	0	0	0
0	1	$\top$	0.05π	-1	$+1$	± 1	0
1	0	$\bot$	0.01π	-1	$+1$	± 1	0
1	0	$\top$	0.025π	$+1$	-1	0	± 1
1	1	$\bot$	0.005π	$+1$	-1	0	± 1
1	1	$\top$	0.025π	$+1$	-1	0	± 1

becomes sensible to ask, why not just simulate the algorithm classically? A classical computer would allow for measurement procedures that mimic the collapse, that also spare $Q(t)$, thus saving on operations. This is precisely what most QEA-based works utilize or imply (Wang et al. 2005; Jian et al. 2009; Jmal et al. 2019; Garcia and Maureira 2021; Kaur et al. 2022), often defined by the rule in Eq. (1).

$$\texttt{if } \rho \sim \mathcal{U}[0,1) > |\alpha_i^{(t)}|^2 \texttt{ then } x_i \leftarrow 1 \texttt{ otherwise } x_i \leftarrow 0 \tag{1}$$

In Eq. (1), and subsequently throughout this work for similar instances, $\rho \sim \mathcal{U}[0,1)$ means that ρ is selected uniformly at random from the range $[0,1)$. Obviously the algorithm can no longer be a quantum procedure, as accessing amplitudes like this "quantumly" is impossible.

The update step hints at another amplitude accessing instance that is even more violatory of quantum principles. Alg. 1 describes selecting $\Delta\theta_j$ via a classical procedure **L**. This angle somehow needs to be informed of the best instance of the population, and it needs to rotate the amplitudes of each qubit correctly so they favor collapse towards bit values similar to the current best. Many works (Khorsand and Akbarzadeh-T 2005; Duan et al. 2009; Wei and Fujimura 2012; Li et al. 2022; Zhang et al. 2025) elect to choose this angle from a predetermined table, usually outlined similarly to Table 2. This approach cannot be confused for a quantum one, as it demands runtime and perfect knowledge of the amplitudes of each qubit, thus it restricts the scheme to the classical regime. Of course, not all works use this specific table, or even a table at all, as many opt for dynamic selections of $\Delta\theta_j$.

Staying on the topic of amplitude accessing, some works adopt a different view on the tensor-train chromosome encoding format, and choose to interpret it as two

candidate solutions per quantum chromosome, called *double chains* (Li et al. 2011; Chen et al. 2019b; Yumin et al. 2020; Ning et al. 2021). Given that $\alpha_j = \cos\theta_j$, $\beta_j = \sin\theta_j$, the two solutions are presented as in Eq. (2).

$$P_c = (\cos\theta_1, \cos\theta_2, \cdots, \cos\theta_n), \quad P_s = (\sin\theta_1, \sin\theta_2, \cdots, \sin\theta_n) \tag{2}$$

This is often paired with a conversion from qubits towards the problem-specific solution space $\Omega_i = [a_i, b_i]$ per problem variable X_i. The conversion is commonly effected as in Eq. (3),

$$2X_{j;c}^{(i)} = b_i(1 + \alpha_j^{(i)}) + a_i(1 - \alpha_j^{(i)}), \quad 2X_{j;s}^{(i)} = b_i(1 + \beta_j^{(i)}) + a_i(1 - \beta_j^{(i)}) \tag{3}$$

where the jth bits of $X_{i;c}, X_{i;s}$ are computed by the amplitudes of the jth qubit from the qubit collection i. These $X_{i;c}, X_{i;s}$ variables are then used to define $\Delta\theta$. Both of these actions are not allowed in the quantum regime. The amplitudes of a qubit is a quantum-mechanical idea set up to explain the evolution of the system mathematically, they are otherwise physically unknown for a closed system (Nielsen and Chuang 2010).

Other additions to the QEA template can also point to quantum un-implementability. Readers versed in GA-based optimization probably notice the complete lack of a crossover or mutation operations in Alg. 1. This started as a deliberate choice, citing that the rotation step handles the mating step and that the qubit representation itself gives the algorithm all the diversity it could ever need, so additional genetic operators could hinder convergence (Han and Kim 2000). The field eventually moved towards defining genetic subroutines as a further quality enhancement. Crossover and mutation techniques may mimic their classical counterparts, with the former often having two quantum chromosomes exchanging corresponding qubit groups (Mohammed et al. 2012), and the latter being defined as a NOT operator on randomly chosen qubit-genes (Gu et al. 2009; Sun 2022), thus flipping the probabilities of each qubit being observed on $|0\rangle$ or $|1\rangle$ and effectively "flipping" the gene. A popular crossover technique is *interference crossover* (Xing et al. 2009; Miao et al. 2009; Duan et al. 2010; Sun and Ding 2010), shown in Table 3. Recognizing that these crossovers could theoretically be realized as SWAP gates between the parent chromosomes, there is nothing inherently un-implementable about those ideas. However, it is not rare for quantum-inspired works to just describe the operations in written form, or, worse, to unwrap the linear algebra operations into distinct formulas acting on specific amplitudes. This does imply classical thinking; firstly, very few actions in QC are so trivial so as to not warrant definition as an operator or circuit; secondly, once again the amplitudes are not a resource that can be accessed.

Further clues are more subtle. Quantum-inspired works almost universally do not present or analyze circuit forms of their proposals. This is a natural byproduct of classical algorithmic thinking and it ties in with the aforementioned principle of nothing being trivial in QC; it is generally accepted in the field of quantum information science that a proposed algorithm should be designed as a circuit (or, in cases of another computational model being used, an equivalent appropriate template) so as to prove the viability and applicability of said algorithm. If it can be demonstrably mapped to

Table 3: Interference crossover example. This specific instance was inspired by Mohammed et al. (2012) and recreated here. To understand, imagine the rows of the table as the input parent-chromosomes. Each A_i, B_i, etc represents a qubit. By skewering the table from the top along each diagonal, one can generate the four offspring, defined as e.g. $A^{(t+1)} = A_0^{(t+1)} A_1^{(t+1)} A_2^{(t+1)} A_3^{(t+1)} A_4^{(t+1)} A_5^{(t+1)}$.

	↘	↘	↘	↘	↘	↘
$A^{(t)} \rightarrow$	$A_0^{(t+1)}$	$B_0^{(t+1)}$	$C_0^{(t+1)}$	$D_0^{(t+1)}$	$A_4^{(t+1)}$	$B_4^{(t+1)}$
$B^{(t)} \rightarrow$	$D_3^{(t+1)}$	$A_1^{(t+1)}$	$B_1^{(t+1)}$	$C_1^{(t+1)}$	$D_1^{(t+1)}$	$A_5^{(t+1)}$
$C^{(t)} \rightarrow$	$C_4^{(t+1)}$	$D_4^{(t+1)}$	$A_2^{(t+1)}$	$B_2^{(t+1)}$	$C_2^{(t+1)}$	$D_2^{(t+1)}$
$D^{(t)} \rightarrow$	$B_5^{(t+1)}$	$C_5^{(t+1)}$	$D_5^{(t+1)}$	$A_3^{(t+1)}$	$B_3^{(t+1)}$	$C_3^{(t+1)}$

a quantum circuit, then it plays with the rules of QC and thus it is at least theoretically sound. Another detail that can imply quantum-inspired thinking is the usage of terminology for "quantum" segments that is unexpected from the point of view of QC. For example, qubits are usually called "Q-bits" (Wang and Zhou 2010; Xia et al. 2020; Han et al. 2023; Jannu et al. 2024; Alhayan et al. 2025), mutation operators via NOT gates are called "quantum non-gates" (Li et al. 2010; Wang et al. 2011; Xia et al. 2020), and $U_j(\cdot)$, universally known in QC as the R_y rotation gate (Nielsen and Chuang 2010; Miao et al. 2009), is labeled as "the rotation gate" or "the quantum revolving door operator" (Wang et al. 2005; Vlachogiannis and Ostergaard 2009; Gu et al. 2009; Duan et al. 2010; Chen et al. 2019b), among other cases. It is not an uncommon occurrence for quantum-inspired works and it implies a disconnection from true quantum algorithmic thinking. Consider the case of R_y for example; calling it "the rotation gate" is non-specific for QC, as every single-qubit operator describes a rotation in the Hilbert space of the qubit's state vector of some kind (Nielsen and Chuang 2010). Calling it "the revolving door" is ambiguous for the same reason: although a creative analogy, due to it spinning the state vector around the Y axis in Hilbert space, this description is equally applicable to other single qubit gates, such as R_x and R_z.

Another general strategy often encountered in quantum-inspired works is modeling agent position and movement as a wave function evolution (Boushaki et al. 2018; Wang et al. 2018; Khan et al. 2021; Chen et al. 2019a; Dong et al. 2022). This is particularly prevalent in SI works and especially in PSO variants, which is not surprising as the concept of agent evolution and intercommunication is rooted in SI. Under this scheme, each agent is not described by a deterministic position/velocity point, but by a probability cloud yielding to a wave function $\Psi(X, t)$ such that $\int_{-\infty}^{+\infty} |\Psi|^2 \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}z = 1$, where X denotes the agent's position in problem space. The probability density function $Q(X) = |\Psi(X, t)|^2$ then acts as a sampling source for the agent's position in each iteration; essentially a measurement step.

Evolving the wave function translocates the agent's probability cloud as a whole. This can be leveraged to realize an effective exploration/exploitation trade-off, by

placing the peak of Q's distribution around an optimal point. That way, the agent's evolution is biased towards optimality (exploitation), but it still has odds of being measured somewhere else, enhancing diversity (exploration). In the context of swarm optimization, the optimal point can be defined as the location of the currently most fit individual in the swarm, p. In quantum mechanical language, it can be thought of as p creating a potential well centered around itself that acts as an attractor for the entire swarm. Mathematically, this idea is formulated as in Eq. (4).

$$V(X) = -\gamma\delta(X - p) \tag{4}$$

Armed with this idea, the evolution of Ψ can be derived from the stationary Schrödinger equation:

$$\frac{\hbar^2}{2m}\frac{d^2\Psi(X)}{dX^2} + (E - V(X))\Psi(X) = 0 \tag{5}$$

Solving Eq. (5) leads to the evolution of agent i's position X_i being described as

$$X_i^{(t+1)} = g \pm \alpha|X_i^{(t)} - p| \ln(\frac{1}{u}), \quad u \sim \mathcal{U}(0,1) \tag{6}$$

where g is the well center, which, depending on the implementation, could be the current global best p or a mix of the global best and the agent's personal best. The adjustable coefficient α controls the spread of the probability distribution.

Works using this model are easier to identify as quantum-inspired, because they deal overwhelmingly with classical calculus. This study identified no works presenting circuits or mentioning QC-related ideas in conjunction to interpreting agent evolution as a wave function, except for those works that also fall under the QEA prism (Yin et al. 2010). It is safe to assume that these works are quantum-inspired, as implementing Eq. (6) on real quantum devices would be non-trivial and would warrant explicit construction details.

5 State-of-the-art in Quantum Meta-heuristic Optimization

In this section, an answer to RQ3 is given, namely *"What are the algorithmic strategies and approaches that have already been studied in the literature for genuinely quantum meta-heuristic optimization?"*. Existing proposals for QGA, QACO and QPSO are presented and examined, with attention to both the underlying ideas and the techniques already explored in the literature. Their potential limitations are also discussed. From the authors' perspective, meaningful quantum algorithm design should aim to progress beyond the constraints of the Noisy Intermediate-Scale Quantum (NISQ) era in order to fully realize quantum advantage. Accordingly, while considerations of noise robustness are not absent, they remain secondary to this broader perspective. In every subsection, proposals are listed by order of formal publication, except when

their preprint versions are substantively employed in other works in the same subsection (e.g., as inspiration, extension, or refutation), in which case they follow the order of the preprint. The interested reader may consult the appendices for further information regarding key details of this study. For a more thorough dive regarding QC-related concepts, kindly refer to Nielsen and Chuang (2010).

5.1 Quantum Genetic Algorithms

In this section, proposals relating to QGA are considered. The list includes and discusses attempts at a full translation of GA into the quantum realm.

5.1.1 Reduced Quantum Genetic Algorithm

The authors of Udrescu et al. (2006), in an attempt to realize preliminary QGA ideas discussed in Rylander et al. (2001); Giraldi et al. (2004), present a more fervent version of Malossini et al. (2008), wherein genetic mating operators are found unnecessary. The scheme, labeled the Reduced Quantum Genetic Algorithm (RQGA), builds on the idea that the optimal individual, sized n in bits, is already included in the uniform superposition of n qubits, a resource efficiently preparable. This is in sharp contrast to classical GAs that cannot assume the entire search space and thus have to evolve individuals towards creating the optimal. RQGA thus changes the optimization idea from creation to identification.

In order to find the "needle in the haystack", GA is reduced to Grover search; hence the name (Udrescu et al. 2006). The algorithm requires an individual and a fitness register, of n and m qubits each, where m is the chosen bit accuracy for integer fitness values in two's complement representation. It also makes use of a validity bit, helpful in segregating problem-invalid individuals from the rest of the pool. The initialization of those registers follows the pipeline shown in Eq. (7).

$$\begin{aligned} &|0\rangle_{ind}^{\otimes n} |0\rangle_{fit}^{\otimes m} |0\rangle_{valid} \\ &\quad \xrightarrow{H} \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle_{ind} |0\rangle_{fit}^{\otimes m} |0\rangle_{valid} \\ &\quad \xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_{i \text{ valid}} |i\rangle_{ind} |f(i)\rangle_{fit} |1\rangle_{valid} + \frac{1}{\sqrt{2^n}} \sum_{i \text{ invalid}} |i\rangle_{ind} |2^m - 1\rangle_{fit} |0\rangle_{valid} \end{aligned} \tag{7}$$

Essentially, the individual register is prepared to the uniform superposition via Hadamard (H) gates, and then the individual, fitness and valid registers are entangled together via the U_f unitary implementing the fitness function f. This black box must be able to coherently judge chromosome validity, flipping the valid qubit to $|1\rangle$ in valid instances, but also compute $f(i)$ as an integer in two's complement representation (Udrescu et al. 2006). The design of U_f is problem specific, however RQGA applications on the `KNAPSACK` (Ardelean and Udrescu 2022a) and `GRAPH COLORING` (Ardelean and Udrescu 2022b) problems show that it is implementable in

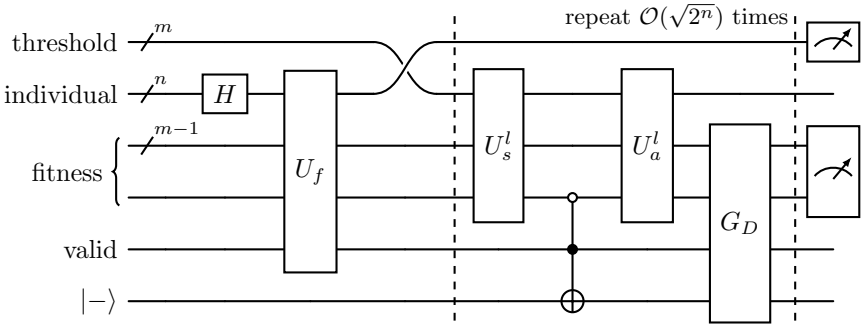


Fig. 5: One loop of RQGA (Udrescu et al. 2006). The search space (Eq. (7)) is encoded via a Hadamard transform and U_f. Then, Grover yields a new fitness threshold. G_D is the standard diffuser, the rest represent the oracle. The operations U_s^l, U_a^l describe the subtraction and increment of the fitness register by the threshold l, respectively. For marking the strict inequality $f(i) > l$, these modules would instead use $l + 1$. RQGA will repeat this entire circuit, typically until the fitness threshold stops improving

those instances using traditional gates such as X, Y, Z, H and Multi-Controlled NOT (MCX).

As a Grover-like scheme, RQGA needs an oracle. The authors propose the usage of an oracle similar to Eq. (E11). The threshold is initialized to a random integer and will be updated after each generation (Udrescu et al. 2006; Ardelean and Udrescu 2022b,a).[2] The oracle implementation ingeniously realizes the idea $f(i) \geq f_x \iff f(i) - f_x \geq 0$ on the circuit level. Concretely, the threshold value is subtracted from the fitness register, and due to parallelism, it is subtracted from every individual's fitness value. Two's complement representation then ensures that positive differences maintain a fitness Most Significant Bit (MSB) of $|0\rangle$, while negative ones, signifying that the individual is below the threshold, have MSB $|1\rangle$. This facilitates $f(i) \geq f_x$ marking by conditioning the phase-flip on the MSB of the fitness register being $|0\rangle$. To further ensure marking of only valid individuals, the phase-flip should also depend on the valid qubit being $|1\rangle$ (Ardelean and Udrescu 2022b,a). The original fitness values can then be reverted by adding back the threshold (Udrescu et al. 2006). Furthermore, hard inequality checks can also be realized similarly, by instead subtracting and adding by the threshold $+1$.

This oracle architecture is pictured in Fig. 5. The required subtraction and addition operators have been studied in the literature (Gossett 1998; Vedral et al. 1996; Cuccaro et al. 2004). A Grover session using this oracle and the standard diffuser G_D leads to a new threshold and an individual achieving it. However, there is no guarantee that this is the solution. Multiple repetitions, with increasingly larger thresholds, are required. The original proposal implies leveraging multiple copies prepared in the state of Eq. (7), where the first copy would utilize the randomly drawn threshold, and

[2] All papers (Udrescu et al. 2006; Ardelean and Udrescu 2022b,a, 2024) suggest drawing the initial threshold uniformly randomly from the integer set $[2^{m+1}, 2^{m+2} - 1]$, ostensibly to restrict search to valid regions. However, since this range lies outside what the m-bit fitness register can represent, and validity is already tracked by a dedicated qubit, the rationale remains ambiguous.

subsequent ones would use the measurement of the previous fitness register (Udrescu et al. 2006). More recent works instead prefer to recycle the same registers, continuously re-initialized to Eq. (7), until the threshold does not improve (Ardelean and Udrescu 2022b, 2024), saving on resources.

RQGA is moderately costly. In terms of quantum resources, it appears to require $\mathcal{O}(n+m)$ qubits. This represents an exponential reduction in memory to classical requirements in the best case, however it excludes any ancillas that may be found necessary depending on gate decompositions. As a Grover-like process, it reportedly requires $\mathcal{O}(\sqrt{2^n})$ oracle calls (Ardelean and Udrescu 2022a, 2024). Given that the construction of the oracle is wholly problem-dependent, its exact cost in elementary quantum gates is not generally knowable. A classical modification attempts to reduce the required number of calls by employing a preliminary run of classical GA to freeze some bits in the chromosomes (Ardelean and Udrescu 2024). The proposed modification introduces the hyperparameter k, by which if k bits are frozen, then k less qubits are required, thus significantly reducing the required oracle calls per threshold update to $\mathcal{O}(\sqrt{2^{n-k}})$. Although a significant achievement, this freezing has the danger of excluding the global best from the considered search space (Ardelean and Udrescu 2024). Another consideration would be that it apparently executes standard Grover (Ardelean and Udrescu 2022a), when the number of marked states t cannot be either 1 or known *a priori* in general. The Boyer-Brassard-Høyer-Tapp algorithm (BBHT) or some similar approach appears more fitting, since a full $\mathcal{O}(\sqrt{2^n})$ iteration session could oscillate the amplitudes of desired states destructively, and instead lead to a measurement of a fitness value that does not improve on the threshold, wasting this entire session. That being said, the overall strategy of RQGA is sound. The quantum realm offers exponentially larger search spaces for cheap. Perhaps the design focus should therefore pivot, from chromosome manipulation to location.

5.1.2 Strictly Structured Quantum Genetic Algorithm

The scheme presented in SaiToh et al. (2013), later dubbed the Strictly-Structured Quantum Genetic Algorithm (SSQGA) (Kim and Ahn 2022b), is another Grover-based QGA, that this time improves upon the previous attempts and re-introduces crossover and mutation in its formulation.

The original proposal of SSQGA (SaiToh et al. 2013) depends on a semi-classical pseudo-random number generator circuit, described mathematically as in Eq. (8).

$$\mathcal{R}:\ \mathcal{R}(|j\rangle_{index}\,|0\rangle^{\otimes n}_{individual}) = \begin{cases} |j\rangle_{index}\,|x_j\rangle_{individual}\,, & j, x_j \neq 0 \\ |0\rangle_{index}\,|0\rangle_{individual}\,, & j = 0 \end{cases} \tag{8}$$

Such an operator $\mathcal{R}$ is implementable as a cascade of c controlled-multi-targeted gates based on pseudo-random numbers. Concretely, supposing the input of this operator is a c-(qu)bit register, each (qu)bit i controls the operation $\bigotimes_{j=0}^{n-1} X^{\rho_{ij}}$ on the n-sized output, where $\rho_i = \rho_0\rho_1\cdots\rho_{n-1}$ is a pseudo-random binary number. This in turn can be decomposed into many Controlled NOT (CNOT)s, an example is shown in Fig. 6. Prepared anew every generation, this engine maps c-sized bitstrings to n-sized ones

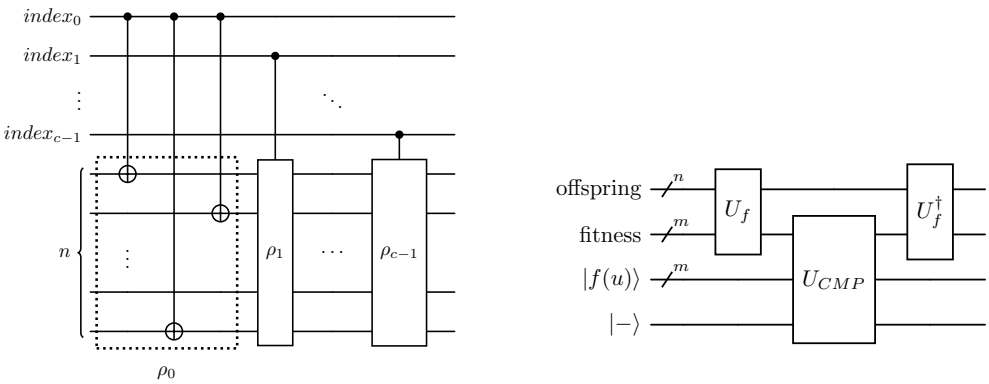


Fig. 6: a The original pseudo-random number generator $\mathcal{R}$ for SSQGA. Each index qubit controls a multi-target gate implementing a random binary number ρ_i. In turn, this can be implemented as many separate CNOTs controlling on the same qubit, one for each bit in ρ_i that is $'1'$. b The oracle architecture described in SaiToh et al. (2013). The oracle operates on the offspring register. The unitary U_f computes the fitness of the offspring in parallel, outputting the value on the ancilla fitness register, sized m for some selected bit accuracy. A comparator then performs the check $f(i) \geq f(u)$, marking satisfying individuals on the ancilla qubit, initialized to $|-\rangle$. Then, U_f is uncomputed

pseudo-randomly, because the bits of the input trigger many overlapping bit-flips on the output, leading to a fairly scrambled result (SaiToh et al. 2013).

Contrary to RQGA, SSQGA considers a sparse population of n-qubit individuals. To achieve this, it connects each chromosome with a c-sized index, such that $2^c \ll 2^n$. Initialization is performed by preparing the index register on the uniform superposition and applying $\mathcal{R}$. With the 0 index vacant, initialization links the current best x^* to it with MCX gates activating on $|0\rangle^{\otimes c}$, in a manner similar to $\mathcal{R}$ (SaiToh et al. 2013). The initialization pipeline looks as in Eq. (9). To facilitate crossover, the initialization step U_{init} prepares two identical versions of state $|\phi\rangle$.

$$
\begin{aligned}
&|0\rangle_{index}^{\otimes c} |0\rangle_{individual}^{\otimes n} \\
&\qquad \xrightarrow{H} \frac{1}{\sqrt{2^c}} \sum_{i=0}^{2^c-1} |i\rangle_{index} |0\rangle_{individual}^{\otimes n} \\
&\qquad \xrightarrow{\mathcal{R}} \frac{1}{\sqrt{2^c}} \sum_{i=1}^{2^c-1} |i\rangle_{index} |x_i\rangle_{individual} + \frac{1}{\sqrt{2^c}} |0\rangle_{index} |0\rangle_{individual} \\
&\xrightarrow{C^{|0\rangle^{\otimes c}}(x^*)} \frac{1}{\sqrt{2^c}} \sum_{i=1}^{2^c-1} |i\rangle_{index} |x_i\rangle_{individual} = |\phi\rangle
\end{aligned}
\tag{9}
$$

The prepared state $|\phi\rangle\,|\phi\rangle$ holds the current population twice. By mentally bisecting and relabeling the individual registers, Eq. (9) is equally reformulated as in Eq. (10).

$$|\phi\rangle^{(1)}\,|\phi\rangle^{(2)} = \frac{1}{\sqrt{2^c}}\sum_{i=0}^{2^c-1}|i\rangle_{index}\,|x_{i;\text{left}}\rangle^{(1)}\,|x_{i;\text{right}}\rangle^{(1)} \otimes \frac{1}{\sqrt{2^c}}\sum_{j=0}^{2^c-1}|j\rangle_{index}\,|x_{j;\text{left}}\rangle^{(2)}\,|x_{j;\text{right}}\rangle^{(2)} \tag{10}$$

Based on this, coherent single-point crossover is effected in an ingenious way; they trace out the $|x^{(1)}_{i;\text{right}}\rangle$, $|x^{(2)}_{j;\text{left}}\rangle$ sub-registers, leaving the implicitly crossed-over state of Eq. (11).

$$|\phi'\rangle = \frac{1}{2^c}\sum_{ij=0}^{2^{2c}-1}|ij\rangle\,|x_{i;\text{left}}x_{j;\text{right}}\rangle_{offspring} \tag{11}$$

Nothing actually occurs in terms of gates. In other words, $U_{crossover} = I$ is the identity map. By this mere relabeling and change of point of view, every possible crossover has been performed in parallel (SaiToh et al. 2013). One can imagine a similar process for a crossover point other than the half-point, or multiple points, by making equivalent reformulations.

Mutation follows the crossover step, again implemented coherently. Affecting only the offspring register, it operates by randomly selecting some of its qubits as targets, others as controls activating on $|1\rangle$, and others as controls activating on $|0\rangle$. Not all qubits need to participate. With those random selections in mind, a multi-control multi-target gate can be defined, where the target qubits are flipped conditioned on the controls. In other words, mutation happens on a handful of qubits for only some individuals, mimicking its classical analogue. Elitism can be reinforced here, by selecting a control condition such that the current best x^* does not satisfy it (SaiToh et al. 2013). This process mimicks classical mutation in that some individual bits are flipped at random; it should be noted, however, that defining random controls has the deliberate effect of mutating the entire subgroup of individuals that satisfy them, in the exact same manner, in parallel. This identical mutation pattern deviates from the classical norm and might have implications on the population's diversity.

With all the parts in place, SSQGA can be formulated as a maximum finding-like process, implementing Amplitude Amplification (AA), for selecting increasingly fit individuals for x^*. Concretely, the amplification starts on the ground state for $2(c+n)$ qubits, with a transformation unitary of $U = U_{mut}U_{crossover}U_{init}$. The required oracle follows the architecture shown in Fig. 6 (SaiToh et al. 2013). The optimization process starts with a randomly selected n-bit x^* and it stops once its evolution surpasses in fitness a sufficiently high predetermined threshold $\mathcal{T}$, $f(x^*) > \mathcal{T}$ (SaiToh et al. 2013). As defined, this procedure reportedly costs $\mathcal{O}(\tilde{N}poly(\log\tilde{N}\log N))$ in time per generation, $\mathcal{O}(poly(\log N))$ in quantum memory and $\mathcal{O}(\log N)$ in classical memory, for $\tilde{N} = 2^c$ and $N = 2^n$, which is found to be quadratically better than the direct classical analogue (SaiToh et al. 2013).

A more recent study on SSQGA (Kim and Ahn 2022b) has found that the original design of $\mathcal{R}$ clogs the optimization process with too many random individuals in comparison to the current best x^*. The proposed improvement is to "freeze" some qubits

in $\mathcal{R}$'s output to identical values as x^*. This naturally enhances the exploitative nature of the algorithm, as the new population shares more genes previously found comparatively optimal. At the same time, diversity persists, as the rest of the output qubits continue to be pseudo-random. It further minimizes the size of the considered superposition, which has direct implications on the required Grover iterations. However, this altered engine structure drastically alters the structure of the overall algorithm.

Algorithm 2: Modified SSQGA (Kim and Ahn 2022b)

Data: Index size c, chromosome size n, problem fitness f, a sufficiently large fitness threshold $\mathcal{T}$.
Result: Optimal individual x^* according to f.

$x^* \leftarrow 0$
Perform a single iteration of the original SSQGA, yielding individual u
$x^* \leftarrow$ `argmax`$(f(u), f(x^*))$
repeat
  $u_0 \leftarrow 0$
  $u_1 \leftarrow 0$
  for $i \leftarrow 0, 1$ **do**
    Create a modified $\mathcal{R}'_i$, where some of the n output qubits are frozen to bit values identical to x^*
    Run AA-Maximum Finding similar to the original proposal. The initial state is $|0\rangle^{\otimes c} |0\rangle^{\otimes n}$, U_{init} uses $\mathcal{R}'_i$ and U_{mut} is the same but applies only to the given register pair
    $u_i \leftarrow$ The result of the above process
  end
  $x^* \leftarrow$ `argmax`$(f(u_0), f(u_1), f(x^*))$
until $f(x^*) > \mathcal{T}$
return x^*

The modified process is shown in Alg. 2. The new pseudo-random generator dictates separate manipulation of the index-individual pairs, excluding crossover once again from the genetic formulation (Kim and Ahn 2022b). Small-scale experiments on Qiskit (Javadi-Abhari et al. 2024) show that the frozen variant accelerates convergence without dropping the solution quality, signifying a positive change (Kim and Ahn 2022b). SSQGA remains one of the most comprehensive and technically well-defined proposals in the field: it manages to define an almost fully quantum algorithm, that leverages established techniques such as AA while preserving and translating traditional GA subroutines in a recognizable way.

5.1.3 Hybrid Quantum Genetic Algorithm

A hybrid quantum-classical GA proposal that is purposed for NISQ devices can be found in Acampora and Vitiello (2021). The presented algorithm, named the Hybrid Quantum Genetic Algorithm (HQGA), delegates chromosome initialization and fitness

evaluation to the classical regime in order to maintain the rest as a quantum process that is affordable and meaningful.

HQGA needs at least three quantum chromosomes in order to operate. At each generation, one will be deemed the elite, and the rest will inherit a portion of its genetic information via quantum entanglement. The quantum chromosomes are constructed as m-sized registers, where m is the classical chromosome size in bits. Each quantum chromosome is initialized via the Hadamard transform, in order to put it in uniform superposition holding all possible 2^m individuals implicitly. Then, each qubit-gene is rotated by an additional $R_y(\pm\delta)$ gate. The purpose of this operation is to tilt each qubit, uniformly randomly, towards a specific bit value; concretely, a positive (negative) sign favors measurement towards $|1\rangle$ ($|0\rangle$). This serves to distinguish the quantum chromosomes between them and add a layer of diversity to the overall algorithm (Acampora and Vitiello 2021). Given that $H = R_y(\frac{\pi}{2})$, one can neatly describe chromosome initialization as $R_y(\frac{\pi}{2} + \delta_j)$ per qubit j, where $\delta_j = \pm\delta$.

Selection in HQGA is realized via measurement of every chromosome register. The collapsed bitstrings undergo classical fitness evaluation, and the one with best fitness is deemed the elite. This selection mechanism is essentially non-uniformly random selection from a pool containing the entire search space. It will be made clear in the following that each bit is iteratively biased in probability towards optimal values, which is why a random selection process can work here. However, it also leaves the door open to non-optimal bit-strings being measured, which has the potential of slowing down convergence.

In order to continue, the quantum chromosomes need to be reconstructed. Per the described initialization process, this is trivially done, as the angles can be kept track of classically. Therefore, all non-elite chromosomes are reconstructed by applying R_y gates with the saved angle sum per each qubit, save of those qubits that will participate in crossover. Reconstruction of the elite is more pronounced. The proposal of Acampora and Vitiello (2021) cites three different elitism strategies. Pure elitism simply reconstructs the elite similarly to the non-elite ones, by rotating its qubits with their saved angles. Reinforced elitism does the same, but it adds a bias $\pm\rho$ to every qubit angle in correspondance to its collapsed value. Specifically, if the elite collapsed to bit value $'1'$ ($'0'$) for gene j, then its reconstruction would be appended by $+\rho$ ($-\rho$). Finally, deterministic elitism reconstructs the collapsed bitstring as a quantum basis state. Concretely, if the elite was measured to $b_1 b_2 \cdots b_{m-1}$, then its deterministic reconstruction would be $|b_1 b_2 \cdots b_{m-1}\rangle$. This is trivially possible with NOT gates on qubits whose corresponding bits are measured to $'1'$. These elitism rules reflect the exploration-exploitation trade-off, as the pure scheme favors diversity by allowing random measurements away from the collapsed bitstring found optimal, while determinism imposes the measured pattern on the subsequent iterations (Acampora and Vitiello 2021).

Whichever the elitism mechanism, the revived elite acts as the driver for crossover. HQGA splits the elite into $\lfloor \frac{m}{n-1} \rfloor$ parts, where n is the quantum population size. Each part is assigned to one of the non-elites. The proposal recommends shuffling the non-elite pool so as to introduce a degree of randomness to this assignment (Acampora

and Vitiello 2021). Having assigned each elite part with a non-elite, crossover is realized by CNOT correlation between part and chromosome. Specifically, if chromosome i was assigned a part of the elite that contains the elite's j qubit-gene, then the jth elite qubit controls the operation, targetting the jth qubit of chromosome i. Figure 7 helps understanding this rule by providing a graphical example. This pattern effects crossover in the sense that the bitstring patterns of the elite are transferred to the correlated individuals. At measurement, each part of the elite will collapse to a set of bit values, and the corresponding individuals will inherit those values on the same bit positions (Acampora and Vitiello 2021). More recent research postulates that, for increased n, this crossover process becomes too distributed, slowing down convergence (Xu et al. 2025). A modification they propose is to define a ratio $P_{targets}$ of target qubits per chromosome size, and apply $mP_{targets}$ CNOTs on every non-elite chromosome. This means that each gene of the elite participates in the crossover more than once in general, leading to a more aggressive genetic spread (Xu et al. 2025). Overall, this technique poses a smart and functional application of entanglement for simulating classical crossover; nevertheless, it also means that, on an "unlucky" measurement of an elite gene, the non-elite entangled gene will inherit the same subpar value.

Finally, qubits among all chromosomes that do not participate on crossover are selected at random to undergo mutation. Qubit mutation is yet again realized by appending the saved angle of each gene with an angle μ. This angle is selected such that its addition inverts the probabilities of the qubit to collapse on its two basis states, specifically $\mu = \pi - 2\Theta_{i,j}^{(g-1)}$, where $\Theta_{i,j}^{(g-1)}$ is the accumulated angle for qubit j of chromosome i at iteration $g-1$. Inverting the amplitudes effectively flips the expected measurement value, reflecting classical mutation, although again the complementary gene value is also generally probable. Selecting qubits to mutate follows the conventional procedure of drawing a random number and comparing it to the predefined mutation rate $P_m \in [0, 1]$ (Acampora and Vitiello 2021).

The algorithm continues the reconstruction-crossover-mutation-selection cycle for a predetermined amount of iterations G (Acampora and Vitiello 2024). Coherence is maintained throughout each iteration because crossover connections and mutation adjustments can be decided classically *a priori*. Experimental testing on noisy and ideal quantum processors demonstrates a capability of high solution accuracy (Acampora and Vitiello 2021). Interestingly, the algorithm performs slightly better on the noisy environment; this can be attributed to noise acting as an additional layer of mutation. The pertrubed effect randomly alters qubit states, potentially enhancing diversity.

HQGA is a procedure that makes efficient use of a moderate amount of quantum resource to convincingly define GA-like optimization. Indeed, the required circuit width can be estimated to $\mathcal{O}(nm)$, which reflects the classical cost but can be viewed as an exponential reduction on the implicit superposed population size $\mathcal{O}(n2^m)$. This compression translates to manageable classical assistance requirements as well. Angle bookkeeping requires memory linear to the circuit width, and fitness evaluations scale only logarithmically with the implicit population, since they happen on the measured chromosomes (Acampora and Vitiello 2021). Equally importantly, the gate count seems to stay roughly similar to the classical, assessable at $\mathcal{O}(mn + mP_{targets})$, while

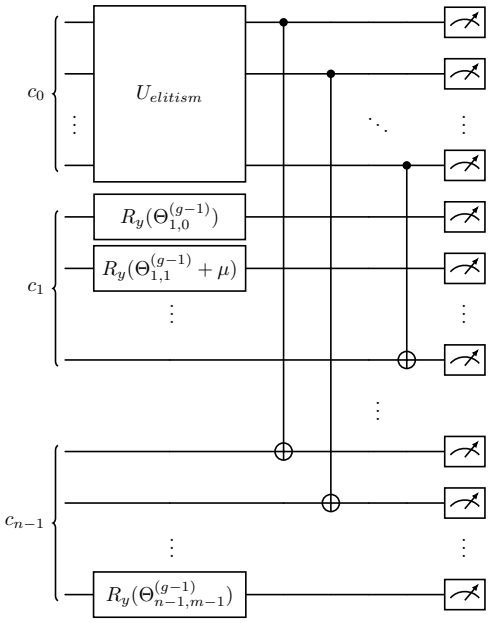


Fig. 7: Iteration g of HQGA (Acampora and Vitiello 2021). The elite chromosome (here c_0) is reconstructed according to the selected principle, while the rest are re-initialized to their prior rotation angles $\Theta_{i,j}^{(g-1)}$. Parts of the elite are assigned to each chromosome, and corresponding gene groups are entangled together. Entangled non-elite genes do not undergo re-intialization. This ensures that, after measurement, corresponding parts between the elite and the individuals share bit values. Non-entangled qubits can participate in mutation. Here, gene $1,1$ gets mutated, and its amplitudes get inverted. Measurement selects the population of generation g. Note that $\Theta_{1,1}^{(g)} = \Theta_{1,1}^{(g-1)} + \mu$ and $\Theta_{1,0}^{(g)} = \Theta_{1,0}^{(g-1)}$, since gene $1,0$ was not manipulated

the circuit depth remains low. Whatever $P_{targets}$ may be, each non-elite gene gets entangled with only a single elite gene, which yields a tunable depth of $\mathcal{O}(P_{targets})$ (Xu et al. 2025). A low depth count is very important for NISQ era computers, as shallow circuits are more resilient to decoherence and noise (Acampora and Vitiello 2021). Still, current qubit resources remain scarce. HQGA struggles to reach its full potential under such harsh memory restrictions. Classical extensions have been put forward to mitigate this, such as shrinking the problem search space via granular computing (Acampora and Vitiello 2023), or applying divide-and-conquer strategies on search space subranges (Acampora and Vitiello 2024).

5.1.4 Quantum genetic algorithm with multiple quantum chromosomes

Another near-completely quantum GA scheme was proposed in Ibarrondo et al. (2024).[3] This scheme too elects to define multiple quantum chromosomes in order to describe the implicit classical population, a choice that aids intuition in introducing quantum operations for crossover and mutation.

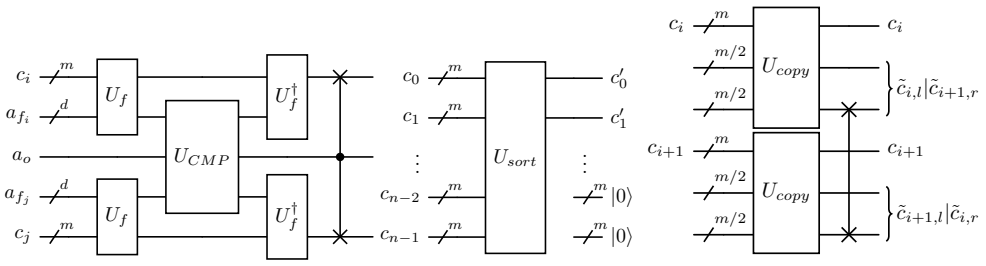
The quantum population, sized n, is initialized to superpositions. However, contrary to the norm, not all quantum chromosomes are set to the uniform superposition. Instead, the registers are prepared to random states drawn from the Haar distribution. This leads to diversity among the quantum population, making the following genetic operations acting on them meaningful, in opposition to just using Hadamard transforms which would lead to evolving copies of the same quantum individual (Ibarrondo et al. 2024).

The selection process aims to realize truncation selection without breaking coherence. The proposal employs the bubblesort algorithm in order to coherently sort the quantum chromosomes in order of fitness, such that the first register holds the fittest member (Ibarrondo et al. 2024). Bubblesort is notoriously inefficient, requiring $\mathcal{O}(n^2)$ comparisons for n elements. Nevertheless, for small n this is not a problem, and its implementation is simpler relative to other schemes. An idea of how it could be realized quantumly is shown in Fig. 8. After performing the sorting, selection can be finalized by resetting the last half of the quantum pool. In Appendix D.5, it is explained why a reset operation cannot be performed coherently. For this reason, this algorithm traces out those registers, effectively ignoring them for the rest of the procedure, and introducing new fresh equivalent registers. This idea maintains coherency, but it also discards information, which leads to the considered sorted population no longer being a pure state. Instead, it is a mixed state; a probabilistic ensemble of multiple versions of the sorted population (Ibarrondo et al. 2024).

The algorithm continues with crossover. The QGA of Ibarrondo et al. (2024) employs the standard single-point crossover on the middle of the chromosomes. This is performed by copying the selected chromosomes on the fresh registers, and then interchanging halves of neighbors among the copied chromosomes via SWAP gates. This process is shown graphically in Fig. 8. Copying is another operation that cannot be performed perfectly and coherently (see Appendix D.5). The proposal benchmarks two known pseudo-clone operations (Alvarez-Rodriguez et al. 2014; Bužek and Hillery 1996). These unitaries realize approximate cloning by maintaining a high fidelity (state similarity) between copied and copiee, with some error margin. The inevitable error could play a constructive role by introducing slight variation in the copied versions of the selected individuals, akin to a secondary mutation (Ibarrondo et al. 2024). Benchmarking reveals that employing one copier or the other leads to a trade off between faster convergence and better solution quality (Ibarrondo et al. 2024).

Mutation finalizes the coherent evolution of the population. In Ibarrondo et al. (2024), mutation mimics the depolarization error of quantum noise (Nielsen and Chuang 2010). Each qubit of every quantum individual is selected for mutation with

[3] First preprint version published in 2022.



(a) (b) (c)

Fig. 8: Subroutines in Ibarrondo et al. (2024). a shows a possible implementation of the bubblesort pair-wise sort step. Two ancilla registers of d qubits each are introduced, and U_f computes the fitness of c_i, c_j in them. Then, a unitary comparator performs the check $\texttt{best}(f(c_i), f(c_j))$, setting an ancilla qubit a_o to $|1\rangle$ if c_i is worse than c_j. The fitness registers are uncomputed and discarded, and a final Fredkin gate performs the potential swap. b describes the entire selection process. U_{sort} executes bubblesort by performing all $\mathcal{O}(n^2)$ pair-wise sorts. Then, the last $n/2$ individuals c_i are traced out, and replaced with fresh registers prepared on the ground state. c depicts the crossover of selected individuals c_i and c_{i+1}. Their copies $\tilde{c}_i, \tilde{c}_{i+1}$ interchange last halves via qubit-wise SWAP gates

probability P_m, the predetermined mutation rate, and undergoes the effect of one Pauli gate $\{X, Y, Z\}$, chosen uniformly at random (Ibarrondo et al. 2024). This is a quantum generalization of the classical mutation idea that considers state-flips along all three quantum degrees of freedom. The idea thus deviates significantly from classical inspiration, but this may prove to be a more fitting choice for the quantum setting.

This cycle continues without breaking coherence, for a fixed number of iterations G. After the last generation, the resulting quantum population is sorted one final time, in order to ensure that the solution can be located in the first register. Measuring that register yields the best solution with high probability, since fit basis states are iteratively honed to the primary chromosomes (Ibarrondo et al. 2024).

Although most genetic operator implementations seem largely in agreement with their traditional classical counterparts, it is important to realize that on this QGA they happen on the qubit level, not on the bit or superposition level as with previous proposals. It remains ambiguous whether shuffling qubits in such a classical manner translates to meaningful mating effects occurring in the bitstrings of their superpositions.

This scheme minimizes the need for classical assistance while preserving the defining characteristics of traditional GAs, including all genetic operators. Theoretical analysis and numerical experiments against classical counterparts demonstrate that it can reach high-quality solutions in only a handful of generations, with the dynamics provably converging exponentially in the number of generations (Ibarrondo et al. 2024, 2022). Quantum resource requirements do exceed the limits of near-term hardware. A conservative estimate suggests quantum memory scaling as $\mathcal{O}(nm+nd)$, where m is the

chromosome length and d the fitness accuracy, due to the need for explicit population registers and arithmetic workspace. Although the paper does not provide explicit gate-count asymptotic bounds, a naïve implementation based on reversible comparators and ripple-carry adders suggests a complexity on the order of $\mathcal{O}(n^2 poly(d)+nm)$ (Cuccaro et al. 2004; Ibarrondo et al. 2024), excluding the problem-dependent cost of U_f. Still, this QGA stands as a compelling precedent for near-fully quantum meta-heuristic optimization, making use of arithmetic modules as a solution approach.

5.2 Quantum genetic operators

In this section, proposals relating to QGA are again discussed. The scope includes works that present quantum analogues for specific genetic subroutines, such as initialization, selection, crossover or mutation.

5.2.1 Grover-based quantum selector

Among the earliest proposals is that of Malossini et al. (2008).[4] The authors aim to define a quantum selection mechanism for classical GA by leveraging the Maximum Finding algorithm. This simple idea is described in Alg. 3.

Algorithm 3: Quantum Selector (Malossini et al. 2008)

Data: Fitness unitary U_f, number of iterations k, population size N a power of two, $x \sim \mathcal{U}\{0, 1, \cdots, N-1\}$, $f_x = f(x)$.
Result: An individual x of high fitness.

repeat k **times**
  $|\psi\rangle \leftarrow \sum_{i=0}^{N-1} |i\rangle_{index} |f(i)\rangle_{fitness} |f_x\rangle_{floor}$, prepared using U_f
  $j \leftarrow$ Run BBHT on the index register of $|\psi\rangle$ using Eq. (E11), the target being $f(i) \geq f_x$
  if $f(j) > f_x$ **then**
    $x \leftarrow j$
    $f_x \leftarrow f(j)$
  end
end
return x

The overall strategy depends on Maximum Finding enhancing the amplitudes of individuals that exhibit fitness surpassing an iteratively increasing threshold. In that sense, this quantum selector fulfills a selection role analogous to classical fitness-based choice. Nevertheless, AA as a mechanism is fundamentally distinct: the selection analogy moves away from explicit competition between individuals, instead emerging through the enhancement of the measurement probability of implicitly targeted solutions.

[4] First preprint version published in 2004.

Algorithm 3 is called twice at each GA operation in order to identify two highly performative individuals from the current population. These two "elites" can then mate and their offspring are placed back in the population, substituting for two randomly selected chromosomes (Malossini et al. 2008). One can naturally imagine a more aggressive mating step by selecting more than two parents each time.

Truncation selection and other sorting-based procedures unsurprisingly cost $\mathcal{O}(\tilde{N} \log \tilde{N})$ time for $\tilde{N}$-sized populations, requiring $\mathcal{O}(\tilde{N})$ bit space. The selector of Alg. 3 appears to improve on this in some cases. The quantum memory cost totals $\mathcal{O}(\log N + m)$, where N is the smallest power of two greater or equal to $\tilde{N}$ and m is the desired fitness bit accuracy. In terms of gate count, as analyzed in SaiToh et al. (2013), when the true population size $\tilde{N} \approx N$, the algorithm reduces to standard Grover-style search with query complexity $\mathcal{O}(N/t)$ for t valid solutions,[5] since indices can be used directly and only a small number of selection iterations are needed. However, when $\tilde{N} \ll N$, the algorithm must either perform index-to-chromosome conversion or implement a custom diffuser G_D. Both approaches incur an additional cost of $\mathcal{O}(\tilde{N} \log N)$ per Grover iteration. As a result, the overall time complexity is dominated by this overhead, effectively eliminating the quantum speedup and making the algorithm comparable in cost to classical selection methods. This substantially reduces the algorithm's practical appeal. While historically significant as an early quantum selection proposal, Alg. 3 has largely been superseded by later proposals that achieve similar goals with lower cost.

5.2.2 Variation via Quantum Annealing

QA is a kind of analog QC. It operates by inducing quantum tunneling between physical qubits interconnected in problem-specific ways. This non-classical evolution allows the system to potentially reach optimal points in the described search space that would be hard to reach normally. It therefore incorporates optimization natively.

In the context of QGA implementations, QA has been employed mainly as a mutator. An initial description was given by Coxson et al. (2014), where the proposed GA was largely classical, and QA would accept the population at each iteration in order to perform guided mutation towards more fit candidates. While the presentation claims enviable results ($\times$100 improvement in speed to conventional GA), the details on exactly how QA encodes the population or realizes this guided mutation were not published. For this reason, the utility one can extract from this instance is limited.

A more well-articulated proposal can be found in Quantum-Assisted Genetic Algorithm (QAGA), put forward in King et al. (2019). Contrary to the usual illustration of QA, the idea of *reverse quantum annealing* is presented. Instead of starting the anneal schedule at time $\tau = 0$ and from the uniform superposition, reverse QA starts from a user-defined classical state and at time $\tau = 1$. Then the transverse field is raised gradually, until the system reaches an intermediary state $A(T), B(T)$, $0 < T < 1$. The schedule then freezes for a predetermined amount of time, allowing the "awoken" system to tunnel away from the classical input state, but not too far. Finally, the transverse field starts to reduce in intensity yet again, reaching $\tau = 1$. The resulting

[5] This is relevant because Alg. 3 requires the population size to be a power of two. This cannot be true in general so one would typically pad the true population with dummy values to reach the requirement.

state is randomly different from the initial input, but still recognizably a spawn of the original version. Therefore, reverse QA effects search in the local neighborhood of the input, or, in the context of GA, the desired mutation (King et al. 2019). Reverse QA mutation stands as an interesting concept, with its main bottleneck being the capacity of available quantum annealers. While QA processors may house thousands of physical qubits, due to connectivity constraints and calibration-based deactivations, only a subset forms the usable "working graph" for embedding logical problems (Bernal et al. 2019). The QAGA proposal mentions that the mutation batch hits a maximum of 40 individuals, acutely affecting the total run time or limiting the overall population size by proxy (King et al. 2019).

From a performance perspective, reverse QA replaces the explicit $\mathcal{O}(\#genes)$ mutation step of classical GA with an annealing evolution of approximately fixed duration, covering all the genes of a single individual in parallel. That being said, the practical cost of the procedure is dominated by the probability of obtaining useful post-anneal states, potentially requiring repeated anneal attempts. Consequently, performance for QAGA is more appropriately captured by the *time-to-solution* metric, which incorporates both annealing time and success probability across runs, and may, importantly, scale unfavorably with problem structure. In this sense, the quantum advantage promised by QAGA stems less from asymptotic superiority and more from solution quality, driven by non-local transitions between solution neighborhoods via quantum tunneling (King et al. 2019).

The work in Chancellor (2023) extends the notion of utilizing QA for genetic-like operations, not by introducing a new GA, but by articulating the *inference primitive* formalism. This general framework guides the design of hybrid algorithms for quantum annealers, by treating them as "black-boxes", whose inputs and outputs can be steered by classical post-processing. In contrast to QAGA, this framework conceptualizes how an annealer could perform both crossover and mutation within a single call. While not a concrete QGA proposal, it highlights how QA can be viewed more broadly as a building block for designing meta-heuristic quantum optimizers.

5.2.3 Quantum Mating Operator

The Quantum Mating Operator (QMO) put forward in Acampora et al. (2022a) employs a shallow parameterized quantum circuit, fit for current NISQ devices, to combine the crossover and mutation steps in GA.

The process is described as follows. The algorithm selects a subset of the parents to participate in the mating, while the rest survive as offspring intact. The selected subset is then processed bit by bit, in order to find frequency values corresponding to the amount of times the value '1' appears in each bit position, yielding the set $\{f_j\}$ holding m real-valued frequencies for m-sized chromosomes. These frequencies are upscaled by π. QMO continues by drawing m random numbers from $[0, 1]$, labeled q_j. For each $g_j < P_m$, where P_m is the predetermined mutation rate, πf_j is increased by a random angle from $[0, \pi]$. The final set $\{f'_j\}$ is used as input angles for R_y gates in a quantum circuit consisting of m qubits initialized to the ground state. Measuring this entire circuit gives a single offspring. Repeating the whole process for new mutation draws and angles yields the next offspring, and so on (Acampora et al. 2022a). Figure 9

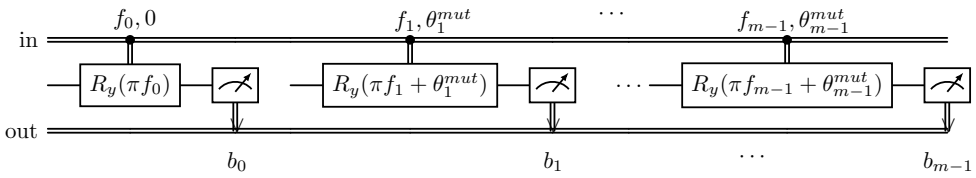


Fig. 9: The QMO circuit as explained in Acampora et al. (2022a). The input register informs each designated qubit of the computed frequency f_j and the mutation angle, if applicable. Measuring qubit j gives the binary gene b_j for the new offspring. The circuit was drawn here in a horizontal manner to preserve space; each frequency is delegated to its own qubit, all of which are operated on in parallel

illustrates an example. Thus QMO fulfills a recombination role in place of crossover and mutation. It stands as an exotic approach, differing fundamentally from classical strategies, since its function is based on quantum state evolution and measurement statistics and rather than explicit substring exchange.

QMO fares well experimentally against conventional mating routines (Acampora et al. 2022a), and it is cheap in terms of quantum resources, requiring just $\mathcal{O}(m)$ qubits and elementary quantum rotations, on the same level as classical requirements. Its constant, singular-step depth and small size also make it fairly robust against quantum noise; on the one hand, the fact that it uses a single quantum step each time that consists wholly of R_y rotations means that it executes extremely fast and that it is ignorant of sign values, making it virtually immune to amplitude damping and sign flip errors, two main error types in QC (Acampora and Schiattarella 2023; Nielsen and Chuang 2010). On the other hand, deliberately inducing noise on the system has the potential to increase convergence quality for highly-constrained problems (Acampora et al. 2023), akin to an extra mutation step.

The scheme also happens to be a fairly limited structure, making it easily modifiable and widely applicable to different instances. QMO has found application in network signal setting design (Acampora et al. 2023) and energy consumption for electric vehicles (Acampora et al. 2024). The original proposal states that selection for QMO can happen at random or via roulette wheel selection, while it can also support elitism by replacing the worst performer of the offspring pool with the best performer of the old population (Acampora et al. 2022a). QMO can also be defined in a distributed fashion in order to support larger sized chromosomes (Acampora and Schiattarella 2023). The idea is to cut the parents in multiple, tractable segments, and run QMO to those in parallel. Then the offspring set can be realized by concatenation of the corresponding parts. In the ideal scenario where there is no noise, this strategy works because there is no entanglement and thus no correlation between the qubits, that would otherwise be lost by leveraging multiple machines. If, however, one wants to utilize quantum noise for implied mutation, as explained previously, this naïve split-and-stitch strategy loses confidence. Finally, it would appear that the work in Martinez et al. (2024) independently came up with a very similar proposal. Their research shows that the scheme might sometimes yield the same individuals as offspring too many times, clogging the convergence flow. They propose a remedy by

applying QMO twice in parallel, each for a different half of the selected parents, with successful results (Martinez E et al. 2024).

It should be noted here that QMO incorporates no entanglement. It was claimed in Sec. 4 that this casts doubt on the effective quantum advantage, and this point is reiterated here. QMO is perfectly simulable classically, a fact that makes true randomness the effective quantum utility being truly leveraged.

5.2.4 Quantum Genetic Sampler

A coherent method to perform selection is proposed in Acampora et al. (2022b), labeled the Quantum Genetic Sampler (QGS). The goal of the selection process is to define a probability distribution that favors individuals based on fitness by performing AA. Details on AA can be found in Appendix E.3.

QGS starts by preparing the current population as a superposition. It is true that arbitrary state preparation is a hard problem (Möttönen et al. 2004; Nielsen and Chuang 2010). However, QGS gets around this hurdle in an ingenious way; the population is kept track of classically, and in the quantum side an m-qubit register, m the chromosome size, is prepared to the uniform superposition. At first this seems to move away from the desired goal; however, one can mentally separate the basis states corresponding to the population $\mathcal{P}$ from the rest, which form the complementary set $\mathcal{Q}$. Therefore population encoding can be realized efficiently as in Eq. (12).

$$|pop\rangle = H^{\otimes m} |0\rangle^{\otimes m} = \frac{1}{\sqrt{2^m}} \sum_{i=0}^{2^m-1} |i\rangle = \frac{1}{\sqrt{2^m}} \sum_{i \in \mathcal{P}} |i\rangle + \frac{1}{\sqrt{2^m}} \sum_{i \in \mathcal{Q}} |i\rangle \tag{12}$$

It will become evident promptly that the existence of the complementary set $\mathcal{Q}$ inside the superposition is not only benign to the algorithm's operation, but also beneficial.

QGS then initiates a truncation selection-like procedure, by sorting the classical population in decreasing order by fitness, so that lower indices hold fitter individuals. The $t \ll 2^m$ first chromosomes resulting from this ordering are overperformers, and the goal now becomes to favor their probability inside the superposition of Eq. (12). This becomes possible via the AA oracle, shown in Eq. (13).

$$O_{QGS} |s\rangle = \begin{cases} e^{\mathrm{i}(\pi - j\beta)} |s_j\rangle , & \text{if } 0 \leq j \leq t-1 \\ |s\rangle , & \text{otherwise} \end{cases} \tag{13}$$

This oracle does not perform the standard sign flip. Instead, it performs a phase rotation with deviation $j\beta$ from the sign flip, where j is the index of the individual in the aforementioned order and $\beta \in [0, \frac{\pi}{t+1}]$ is the step size. This has the effect of fitter individuals being closer to a sign flip and thus more intensely marked, favoring their amplification in the remaining AA process (Acampora et al. 2022b). For clarity, it is highlighted that i in Eq. (13) denotes the imaginary unit.

Performing the Grover operation with this oracle and the standard diffuser G_D for a predetermined amount of times K yields the state shown in Eq. (14),

$$|pop'\rangle = \sum_{j=0}^{t-1} c_{s_j} |s_j\rangle + \gamma \sum_{j=t}^{2^m-t+1} |s_j\rangle \tag{14}$$

with $|c_{s_0}|^2 > |c_{s_1}|^2 > \cdots > |c_{s_{t-1}}|^2 > |\gamma|^2$. Since the oracle favors the overperformers of the current generation, as described, the majority of the amplitude is coalesced around those basis states, in decreasing amounts. In general, AA leaves a leftover nonzero amplitude that gets shared equally to unmarked states, here γ. This is where the utility of QGS shines. Sampling from the state in Eq. (14) will, most of the time, select individuals that are fittest among the current generation, thus maintaining exploitation. However, it also leaves the door open to measuring not only underperformers of the generation, but also foreigners (bitstrings found in the complementary set $\mathcal{Q}$). Thus, the selection process gets enriched with a layer of diversity, favoring exploration (Acampora et al. 2022b).

QGS depends on three hyperparameters, namely t, β and K. The proposal implies that t and β are largely problem-specific, suggesting that $t \ll 2^m$ in order to maintain efficiency, but it also should not be too small. If QGS hones the amplitude to just a couple individuals, it will clog the mating pool with copies of the same individuals and thus hinder convergence (Acampora et al. 2022b). Things become more obscure when it comes to K. In standard AA (case $\beta = 0$), K follows the known theoretical formula, shown in E.1. For $\beta \neq 0$, QGS can deviate from this rule, due to applying more general phase shifts. Experimental tuning by the authors of Acampora et al. (2022b) identifies the setup $t = 3, \beta = \frac{\pi}{8}, K = 3$ as appropriate for the case $\#qubits = 6$. Notably, $K = 3$ matches the theoretical value for $\beta = 0$, although this is not presented as a general rule. One should expect the value of K to be crucially important, because should it lead to a more even amplitude spread, the unmarked region may become too probable and drown out the truncation selection effect.

Contrary to many Grover-based implementations, QGS can be comparatively afforable in quantum resources, particularly with respect to the oracle. The width remains tentatively small at $\mathcal{O}(m)$, and since $t \ll 2^m$, the oracle domain favors a more efficient implementation in general. While an exact per-call gate cost is not reported, the number of calls is tunable at $\mathcal{O}(K)$. Moreover, QGS circumvents the common population state preparation problem, by splitting the superposition implicitly. This does require a complementary classical $\mathcal{O}(N \log N)$ runtime cost, as well as $O(N)$ auxillary bit space, for N-sized populations. This fact makes QGS meet classical costs, limiting potential quantum advantage towards solution quality. Low-scale experiments suggest that the scheme is capable of outperforming conventional selection procedures, in speed, solution quality and overall diversity (Acampora et al. 2022b). Nevertheless, larger-scale empirical validation is required to confirm whether QGS does maintain comparable affordability, convergence quality, as well as robustness to quantum noise, in order to substantiate claims of quantum advantage.

5.2.5 Quantum Population Initializer

The initial population of GA has a notable effect on the convergence process. Especially for constrained problems, depending on the harshness of the constraint, the amount of desirable individuals in the population can be very small. This has effect on both speed and quality of convergence, since from a small sample of desirability, GA will take a while to spread the good genes, and the diversity of good genes is limited. Knowing that an initial population is proper for a given problem is therefore valuable.

The Quantum Population Initializer (QPI), proposed in Kim and Ahn (2022a), aims to employ the quantum counting algorithm (Brassard et al. 1998) to gauge the satisfaction of a given chromosome group and eventually decide on a good initial population. QPI models the problem oracle such as in Eq. (15),

$$O\left|x\right\rangle = \begin{cases} -\left|x\right\rangle, & \text{if } x \text{ satisfies the problem constraints} \\ +\left|x\right\rangle, & \text{otherwise} \end{cases} \tag{15}$$

the idea being that applying Eq. (15) on a given superposition-encoded population will mark all the valid states, allowing the quantum counting algorithm to estimate their number.

The described strategy is shown in Alg. 4. QPI continuously generates random populations and judges their overall satisfaction by counting the number of marked states. Higher number of marks means more adequate individuals. Similar to all Grover-like algorithms, quantum counting costs $\mathcal{O}(\sqrt{n})$ oracle calls for a superposition sized n. QPI then delegates $\mathcal{O}(g\sqrt{n})$ iterations to find a proper population, in order to maintain an overall complexity of $\mathcal{O}(ng)$ in oracle calls. The hyperparameter k is introduced, in order to make this iteration balance adjustable under the complexity limit, totalling the effective oracle calls at $\mathcal{O}(kgn)$ (Kim and Ahn 2022a).

Algorithm 4: Quantum Population Initializer (Kim and Ahn 2022a)

Data: Problem constraints information $\mathcal{C}$, population size n, individual size m, number of satisfactory individuals threshold p, maximum generation count g, iteration hyperparameter k.
Result: A population with at least p individuals satisfying $\mathcal{C}$.

$Q \leftarrow$ a randomly generated population of n m-sized individuals
repeat $\mathcal{O}(kg\sqrt{n})$ **times**
  $\left|Q\right\rangle \leftarrow$ encode Q as a quantum superposition
  Run quantum counting on $\left|Q\right\rangle$ using the oracle of Eq. (15) judged by constraints $\mathcal{C}$, yielding count s
  if $s \geq p$ **then break**
  $Q \leftarrow$ a randomly generated population of n m-sized individuals
end
return Q

It becomes clear from both Eq. (15) as well as $\mathcal{C}$ that the gate cost of the overall algorithm is wholly problem-dependent. That being said, one can generally conclude a qubit count of at least $\mathcal{O}(m)$ for the main population register. By itself, this memory cost represents a spatial reduction to classical needs linear in n, although ancilla qubits for the counting process are probably necessary.

QPI was tested experimentally on the `0-1 KNAPSACK` problem, using different constraints and thresholds. Results showed a significant performance gain under harsh constraints, though at the cost of nearly doubling the runtime (Kim and Ahn 2022a).

The algorithm makes smart use of established quantum processes in order to tackle an important GA problem. However, several limitations remain. Common to other oracle-dependent solutions, whether the oracle can be efficiently mapped to a quantum operator is nontrivial. Another challenge is the requirement of encoding a seemingly arbitrary Q as a quantum superposition in each iteration, again a hard problem in general. Furthermore, it could perhaps benefit the algorithm to keep track of previous population guesses and build on those, instead of generating random ones each time.

5.2.6 Quantum Extreme Value Searching Algorithm

The work in Sabaawi et al. (2023) formulates a process that identifies the co-optimal individuals among a given population. The crux of this idea is the hybrid routine known as Quantum Extreme Value Searching (QEVSA). QEVSA attempts to identify the extreme value (minimum or maximum) that a function f can produce, starting from a rough guess of its range $[lo, hi]$ (Imre 2007). At each step, the midpoint $\frac{lo+hi}{2}$ is utilized to gauge whether there exists an element in f's range less of it. This check is realized by *quantum existence testing*, a special case of quantum counting (Imre 2007). Depending on the outcome, QEVSA then re-examines the correct half sub-range, via classical binary search. The rough sketch is given in Alg. 5.

Algorithm 5: Quantum Extreme Value Searching (Imre 2007)

Data: Initial range guess $[lo, hi]$, the examined function f.
Result: The extremum (w.l.o.g. here the minimum) of f.

$T \leftarrow hi - lo$
$m \leftarrow lo + \lceil \frac{hi-lo}{2} \rceil$
for $i \leftarrow 1$ **to** $\log_2 T$ **do**
  if *QET*(m) *yields true* **then**
    $hi \leftarrow m$
  else
    $lo \leftarrow m$
  end
  $m \leftarrow lo + \lceil \frac{hi-lo}{2} \rceil$
end
return m

Algorithm 5 does not return the argument of the extremum by itself. In Imre (2007), it is reported that QEVSA can be extended by quantum counting and Grover search in order to identify the desired optimizer, however, no further details are given.

The work in Sabaawi et al. (2023) utilizes QEVSA to the benefit of classical GA. The proposal prepares a good initial population by spawning n equally-sized randomly-generated populations, then running QEVSA on all of them to identify the n best individuals in each group. The n "elites" then form the initial population of GA. Afterwards, QEVSA is used again to realize truncation selection, by applying QEVSA $n/2$ times in each iteration. The rest of the algorithm remains classical.

Modifications to the aforementioned QGA scheme allow it to handle constrained problems as well, by appending the constraints to the cost function (Sabaawi et al. 2023). Improvements to QEVSA as well allow it to handle constrained search by incorporating the constraint to the existence testing oracle (El Gaily and Imre 2021).

The complexity analysis in Sabaawi et al. (2023) exhibits some inconsistencies. While the QEVSA subroutine is reported to require $\mathcal{O}(\log G \log^3 \sqrt{2^m})$ operations, where G is the maximum iterations and m the chromosome size, this bound is derived from the constrained version in El Gaily and Imre (2021), which appears to contradict the initially reported cost of $\mathcal{O}(\log T\sqrt{2^m})$ found in Imre (2007). Since QEVSA relies on quantum counting, it inherits the $\mathcal{O}(\sqrt{2^m})$ overhead for number of oracle calls found in all Grover-like algorithms. This cost is not reflected in the reported polylogarithmic bound. Furthermore, the authors in Sabaawi et al. (2023) summarize the complexity cost for this QGA as $\mathcal{O}(Gnm)$. This reflects the complexity cost of classical GA and once again seems unexpected with regards to the cost of QEVSA. The required width appears manageable, at $\mathcal{O}(m + 1/\epsilon)$, where ϵ relates to the desired error in quantum existence testing (Imre 2007).

Overall, while the proposed hybridization introduces an interesting dependence on quantum counting (Brassard et al. 1998), core parts of the cost and the functionality remain ambiguous. In particular, the extension of QEVSA to argument identification is non-trivial: its implementation is intimately tied to solution quality, and it may further increase the computational overhead, potentially limiting the practical advantage of the approach.

5.2.7 Quantum chromosome indexer for selection

In a strategy vaguely similar to QGS, the proposal in Martinez and Zich (2024) aids in the selection step of GA by defining a quantum indexer partly based in Grover search (see Appendix E.1). Prior to the selection process, the classical population is sorted by fitness, such that the last individual is the most fit. A truncation then follows, by discarding the first d individuals and replacing them with random (but problem-valid) chromosomes. The quantum indexer then helps with selection by indicating the index of the chromosome in the population order to place in the parent pool. After a chromosome has been accepted as a parent, it is removed from the population by shifting every individual after it in the order once left-wise.

The quantum indexer is a partial Grover search (Martinez and Zich 2024). Examples are shown in Fig. 10. The circuit operates on $\mathcal{O}(\log |pop|)$ qubits, that are enough to assign an index for each member in the population in superposition. In the original

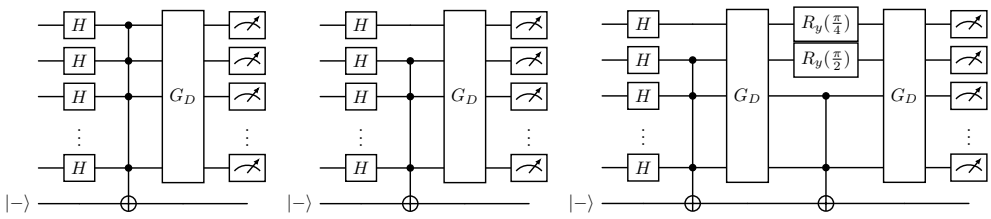


Fig. 10: Quantum indexers for selection proposed in Martinez E et al. (2025). a describes the original proposal of Martinez and Zich (2024) with a MCX oracle targetting $|11\cdots1\rangle$. b shows the first modification; by skipping the topmost qubit, Grover now amplifies towards both $|11\cdots11\rangle$ and $|11\cdots10\rangle$, the best and second-best individuals in terms of fitness. An extension of this modification is shown in c, that starts with the b idea and then performs another rotation, this time skipping another qubit. This would lead Grover to favor all four $|11\cdots100\rangle$, $|11\cdots101\rangle$, $|11\cdots110\rangle$ and $|11\cdots111\rangle$ states. However, to compensate for the fact that the first oracle already favors the last two states, R_y gates are leveraged to bias the amplification away from those values. These circuits were inspired by Martinez E et al. (2025) and recreated here. Note that $|-\rangle = HX\,|0\rangle$

definition of the indexer, the oracle used by Grover is a simple MCX targeting the maximal state $|2^{|pop|}-1\rangle$. The strategy is to start amplifying the register towards the maximal state, but not completely, such that its measurement will favor the most fit chromosome, but not unilaterally. Remember that the sorting places the most fit individual in the last position, so measurement yielding the maximal state points to the fittest member. Allowing other indices to be measured as well increases the diversity of the selection.

Modified oracles, presented in Martinez E et al. (2025), can improve the quality of the selection while maintaining diversity. As shown in Fig. 10, by strategically omitting qubits as controls from the index register allows for the amplified pattern to include more indices that are high in value, which by extension point to fitter individuals. These modifications further enhance the emulation of classical selection, by favoring amplitudes towards index patterns exhibited by fitter individuals. Importantly, the partial application of AA means that the selection effect is more evenly random than one may initially realize. While this is begin explicitly done to enhance exploration, it may also lead to slower convergence speeds.

This idea requires a modest amount of quantum resources. Given $|pop| = N = 2^n$, the required qubits scale as $\mathcal{O}(n)$. The nature of the Grover iteration here happens to be exactly known. The gate cost thus depends on the MCX decomposition model, for which there are several, with varying complexity overheads (Barenco et al. 1995). The most conservative estimate would be to assume a quadratic cost decomposition and zero ancillas, which would push the cost of each Grover iteration in this indexer to no more than $\mathcal{O}(n^2)$. It should be noted, however, that similarly to QGS, this indexer's

operation is reliant on complementary $\mathcal{O}(N \log N)$ classical runtime and $\mathcal{O}(N)$ bit space.

Research on this scheme does not really elucidate how many Grover iterations one should realize before measuring. The original work implies 2 (Martinez and Zich 2024), and subsequent works are more vague. All considered experiments utilize just 4 or 5 qubits (Martinez E. et al. 2024; Martinez and Zich 2024; Martinez E et al. 2025). Given that the effect of a single Grover iteration scales downward with qubit count (Grover 1996; Nielsen and Chuang 2010), it is safe to assume that just 2 iterations is not a universal rule. If the required amount turns out to be too high, it might be the case that even this modest circuit is too expensive for the role it plays.

The indexer has been tested on electromagnetic problems (Martinez E. et al. 2024; Martinez and Zich 2024; Martinez E et al. 2025). As an approach it is distinctive, though its scope is constrained by the limited role of QC, which here serves primarily as a source of genuine randomness.

5.2.8 Entanglement-Aware Quantum Genetic Algorithm

A quantum combination of selection, crossover and mutation steps utilizing entanglement is presented in Haghighi et al. (2025), dubbed the Entanglement-aware Quantum Genetic Algorithm (EAQGA). Contrary to the norm, this algorithm does not maintain a population pool; instead, it keeps track of just the top best chromosomes in each iteration according to fitness. From those two, it generates a quantum circuit that mimics similarities between them while also allowing for random collapses in genes (bits), maintaining diversity.

The scheme starts with a quantum circuit of m qubits, m being the chromosome size, initialized to the uniform superposition $\frac{1}{\sqrt{2}} \sum_{i=0}^{2^m-1} |i\rangle$. This circuit is sampled n times (or, as the paper suggests, n copies of this circuit are measured once in parallel), to produce n initial individuals (Haghighi et al. 2025). Out of those n members, the best x_1 and second best x_2 candidates in terms of fitness are selected, and the rest are discarded. To continue the evolution, x_1 and x_2 are compared in bits in order to find pair correlations; specifically, the algorithm identifies all *positive correlations* in bit pairs between the two elites ($(00)_{x_1}$ and $(11)_{x_2}$ or vice-versa), as well as all *negative correlations* ($(10)_{x_1}$ and $(01)_{x_2}$ or vice-versa). An example is shown in Fig. 11.

The goal of this search is to conclude on a re-initialization pattern for the quantum circuit(s), that favors the values of the two elites and specifically their common patterns. Not all correlation pairs are used at each iteration however. The manner of selection is implied to be probabilistic and problem-specific. In the proposal, the tackled problem was `PORTFOLIO OPTIMIZATION`. Selection thus aimed to prioritize correlation pairs that agreed with the shape of the problem. Concretely, the covariance matrix of the problem was normalized, and agreement was judged by the sign of covariance and the "sign" of the correlation matching; positive (negative) correlation pairs exhibiting positive (negative) covariance were favored by handicapping other instances with a penalty factor. This factor halved the selection probability at early iterations, and gradually recovered it later on, emphasizing problem validity at the start but more aggressive exploration towards the end (Haghighi et al. 2025). This is just one example for realizing pair selection.

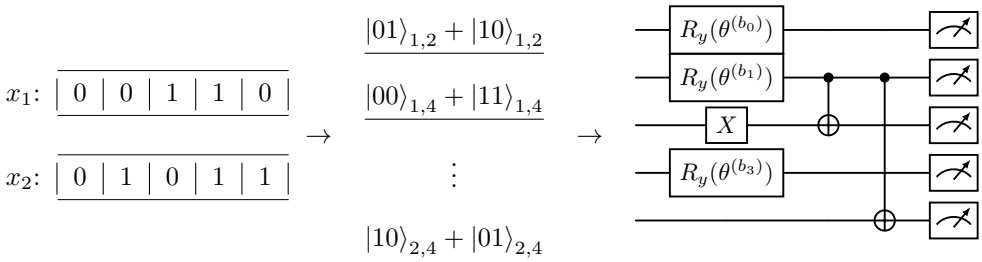


Fig. 11: The circuit re-initialization process for Haghighi et al. (2025). The algorithm identifies the two most fit individuals in the current iteration x_1, x_2, and lists their apparent correlation pairs. Note, for example, that the bits in positions 1 and 2 (counting zero-based left-to-right) are 01 for x_1 and 10 for x_2, thus forming the possible entanglement pair $|01\rangle + |10\rangle$. This is a negative correlation. A positive correlation can be seen between the bit positions 1 and 4. Such possible correlations are compiled together, and a subset of them is selected probabilistically according to the problem. Here, the chosen pairs are underlined, the bit indices described by the kets are written as subscripts, and amplitudes are skipped for simplicity. Having chosen correlation pairs, the circuit that will produce the next population can be created. The pairs $(1,2),(1,4)$ form the chain $(1,2,4)$, implemented as a single R_y and two CNOTs. The remaining bits 0 and 3 gain R_y gates, instead. Sampling from this circuit produces offspring that are both crossed over by the two elite parents, effected by entanglement, but also mutated due to the rotations. This example and the figures were inspired by Haghighi et al. (2025) and recreated here

Having selected the correlation pairs, the circuit is re-initialized by realizing these correlations as entanglement. For each decided pair, once bit position is anointed the driver, and feels the effect of an R_y gate rotated by the angle shown in Eq. (16).

$$\theta^{(0)} = 2\cos^{-1}(\sqrt{p}_\alpha), \quad \theta^{(1)} = 2\cos^{-1}(\sqrt{1-p_\alpha}) \tag{16}$$

In Eq.(16), p_α is a hyperparameter probability, and $\theta^{(b_j)}$ is chosen when x_1 is measured to b_j in position j. Then the pair receives a CNOT operation, with the driver being the control. Instances where chosen correlation pairs form chains, for example (i,j) and (j,k), these pairs are fused, leading to a described entanglement (q_i, q_j, q_k) with one qubit being the driver (Haghighi et al. 2025). Qubits that do not participate in correlation pairs are instead exposed to plain R_y gates utilizing the same angle shown in Eq. (16).

Thus EAQGA continues until termination. The algorithm stops after a predetermined amount of iterations (Haghighi et al. 2025). Since correlation patterns are carried over to the circuit sampler, offspring measured by it will inherit these connections in their bits via entanglement, thus effecting crossover. At the same time, the usage of R_y gates dependent on p_α adds uncertainty, both to the entangled qubits but also to the rest. Therefore mutation is also present, as the bit values of the previous elites are favored but not demanded (Haghighi et al. 2025). Given that the act of

measurement leads to the new elites, one can conclude that this circuit further realizes selection, thus this is a hybrid procedure that executes GA in its entirety. That being said, these analogies should not be taken as absolute. The nature of superposition collapse being probabilistic, this quantum analogue adds a layer of stochasticity to the traditional functionality.

The formulation of EAQGA utilizes at most $\mathcal{O}(nm)$ qubits, should a parallel scheme of n circuits be followed, echoing the classical analogue as n *de facto* acts similar to a population size. Without parallelization, it requires a circuit of $\mathcal{O}(m)$ qubits, sampled $\mathcal{O}(n)$ times. Each circuit re-intialization requires $\mathcal{O}(m)$ elementary quantum operations. This bound is respected due to each consistuent qubit receiving at most one R_y rotation and the number of entangled pairs being linear in the chromosome size (Haghighi et al. 2025). In isolation, it represents a significant reduction on the equivalent classical cost, however the cost of identifying and selecting proper candidate correlation pairs, potentially quadratic in m in cost, might lessen this improvement (Haghighi et al. 2025). It seems to promise positive results; experiments in stock-related problems against classical and inspired GAs shows that EAQGA has the potential to outperform both in simulated and in real environments (Haghighi et al. 2025). It is a quantum enhancement strategy suitable for NISQ devices. However, its dependence on bit-pair identification and selection, attained classically, hides the danger of seriously bogging down the overall performance, neutralizing possible quantum advantages.

5.3 Quantum ant colony optimization algorithms

In this section, algorithms relating to QACO are presented. Due to the limited number of existing proposals, both full algorithms and standalone operators are listed here.

5.3.1 Mrityunjay-Nivedita-Debdeep-Amlan-Subhansu algorithm

The authors in Ghosh et al. (2022) articulate an ambitious proposal for a standalone QACO with their Mrityunjay-Nivedita-Debdeep-Amlan-Subhansu algorithm (MNDAS).[6] MNDAS makes use of a path register and a pheromone register, initialized to $|0\rangle^{\otimes x}$ and $|0\rangle^{\otimes d}$, respectively, along with ancilla qubits. Given $\#paths = N$, $x = \lceil \log_2 N \rceil$ and d is decided classically. The algorithm starts by preparing all possible path configurations in the path register by Hadamard gates, leading to the uniform superposition shown in Eq. (17)

$$|0\rangle_{path}^{\otimes x} |0\rangle |0\rangle |0\rangle_{pheromone}^{\otimes d} \rightarrow \frac{1}{\sqrt{2^x}} \sum_{i=0}^{2^x-1} |i\rangle_{path} |0\rangle |0\rangle |0\rangle_{pheromone}^{\otimes d} \tag{17}$$

The algorithm then performs the exploration step. The rough sketch of the ant exploration submodule, along with the rest of the algorithm, are shown as a circuit in Fig. 12. For a predetermined and constant number of iterations K, all paths are considered for selection and marked for pheromone update. Selection is realized by

[6] First preprint version published in 2020.

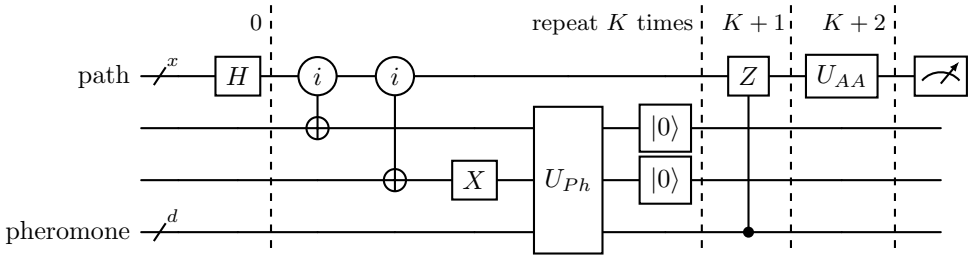


Fig. 12: The quantum circuit for MNDAS. The 0th step prepares the entire path population as uniform superposition inside the path register. Exploration blocks are then executed sequentially, K times. Different, and perhaps multiple, i values are drawn at each step. Step $K+1$ detects the path with maximum pheromone and flips its phase. The CZ gate shown here needs only target a single qubit of the path register. Finally, step $K+2$ increases the likelihood of the marked path through AA. This circuit was inspired by Ghosh et al. (2022) and recreated here

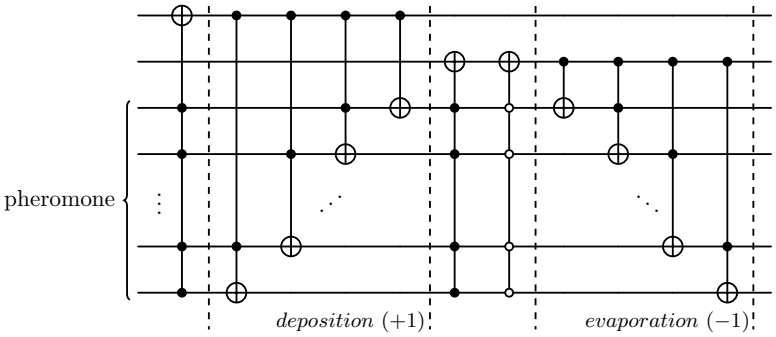


Fig. 13: The implementation of U_{Ph} in MNDAS. The module performs deposition and evaporation on each path pheromone in parallel, conditioned on the ancillas. The MCX gates targetting the ancillas prevent undesired updates, protecting the operator from incrementing the value $|2^d-1\rangle$ (overflow) or decrementing the value $|0\rangle$ (underflow). The maximum pheromone $|2^d-1\rangle$ is also spared from evaporation to speed up convergence. This circuit was inspired by Ghosh et al. (2022) and recreated here

performing the check t mod $weight[i] = 0$, for every iteration t and for all paths i. Each path that satisfies this condition invokes two MCX gates, controlling on the path register for the value i and targetting the two ancillas. The point of this conditional check is to favor selection of shorter (and thus better) paths. Since these K iterations of MNDAS are co-independent, all the checks can happen *a priori* before the start of the quantum algorithm, thus not needing to interrupt its flow.

For each targeted path i, the iteration performs pheromone update, symbolized as the unitary U_{Ph} and defined similarly to Fig. 13. U_{Ph} accepts the ancillas as inputs,

one deciding deposition and the other evaporation. The former is realized as an increment of 1 on the pheromone register, conditioned on the top ancilla, while the latter as the inverted case, realizing decrement by 1 on the same register conditioned on the bottom ancilla. The NOT gate in between the the MCX gates selecting the paths and the pheromone update operator U_{Ph} ensure that selected paths undergo deposition, while unselected ones undergo evaporation. Two complementary checks are performed in other to safeguard the update from overflow and underflow errors, identified as the first and third MCX gates targetting the ancillas, respectively. The second MCX is an additional check that spares the maximum pheromone from evaporation, thus accelerating convergence. After each path, the ancillas are "reset". Appendix D explains why this operation is non-coherent. MNDAS thus requires additional ancillas, $2K$ in fact, to maintain coherence. This drives the quantum memory requirement tentatively at around $\mathcal{O}(\log N + d + K)$.

After K iterations, the path with maximum pheromone is targeted and its phase is flipped. Since each iteration performs at least one deposition, and the maximum pheromone is never evaporated, for suitable K this will always occur. What remains is to enhance the probability of the now marked state, by applying AA on the path register. Details on AA can be found in Appendix E.

MNDAS is a notable proposal, since it manages to define coherently a recognizable ACO procedure in the quantum realm. However, its formulation is questionable since it requires prior knowledge of all path weights. This is intractable in general, but even then, knowing all the weights enables solving the problem by simple linear search, which is cheaper than the reported cost of MNDAS, that being $\mathcal{O}(KN + \sqrt{N})$ (Garcia de Andoin and Echanobe 2022).

5.3.2 Quantum ant colony path selector

The work in Garcia de Andoin and Echanobe (2022) proposes a QACO that focuses on path selection, delegating pheromone updates back to the classical regime. The idea is to introduce a qubit for each of the problem graph's $G = (V, E)$ nodes, and have its basis states represent the decision of either visiting said node ($|1\rangle$) or not ($|0\rangle$). That way, a single bit-string sized $|V|$ describes a specific path through G, and a superposition of $|V|$ qubits can potentially hold all $2^{|V|}$ paths in parallel.

The above scheme requires a single ant-agent, since it can traverse all possible paths in superposition. This QACO assumes that pheromone values are placed on the nodes instead of the edges, and they correspond to selection probabilities for said node (Garcia de Andoin and Echanobe 2022). The quantum ant for iteration j can be described mathematically as in Eq. (18).

$$|ant\rangle_j = \bigotimes_{i=0}^{n-1} R_y(\theta_{i;j}) |0\rangle = \sum_{\text{path } k=0}^{2^n-1} \alpha_{k;j} |k\rangle \tag{18}$$

In Eq. (18), $\theta_{i;j}$ arises from the selection probability of node i for iteration j, and at $j = 0$, $\theta_i = \frac{\pi}{2}$ for all nodes, making all paths initially equiprobable. The accumulated effect of all node-specific pheromone values coalesce to an internal pheromone value

Table 4: The pheromone update table for Garcia de Andoin and Echanobe (2022). x and b represent the collapsed state of this iteration and the current best path, respectively. The table compares bit values i and assigns angle updates $\Delta\theta_i$ for each qubit i. $\top$ and $\bot$ represent logical truth and falsehood, respectively. Assuming the minimization problem, $\leq$ here implies a better fitness. $\Delta\theta_i$ values labeled * flip their sign if $\cos(\frac{\theta_i}{2}) < 0$. The update step is then simply $\theta_i' = \theta_i + \Delta\theta_i$. This table was inspired by Garcia de Andoin and Echanobe (2022) and recreated here.

x_i	0	0	0	0	1	1	1	1
b_i	0	0	1	1	0	0	1	1
$f(x) \leq f(b)$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$
$\Delta\theta_i$	$-0.01\pi^*$	$+0.04\pi$	$-0.05\pi^*$	$+0.07\pi$	$+0.05\pi^*$	-0.07π	$+0.01\pi^*$	-0.04π

for each path k while in superposition, represented as α_k. Therefore, measuring $|ant\rangle_j$ collapses the superposition probabilistically, but biased towards paths that accumulate stronger pheromone. Thus classical selection is mimicked, with a layer of stochasticity that, depending on the speed of accumulation, may have constructive effects on the algorithm's diversity, or lead to excess diversity and slower convergence.

In order to enforce exploitation, the collapsed path is compared to the current best by fitness, and the better one becomes the new best. Pheromone angles are updated via the lookup Table 4, that is reminiscent of aforementioned quantum-inspired works; however, this table is not dependent on amplitudes or unknown values, making it "fair game" for a hybrid algorithm such as this one.

At the same time, the algorithm enhances exploration. An additional set of ancilla qubits drive the exploration process by probabilistically alternating the basis states inside the superposition. This has the effect of scrambling pheromone values, enriching the selection process with diversity by randomly favoring different paths. This can happen in two ways. For unconstrained problems, the exploration qubits are entangled with the ant qubits via CNOT gates; in effect, if a CNOT activates for ant qubit i, all paths flip their decision to visit node i. For constrained problems, there needs to be a safeguard against invalid solutions. In instances where the constraint is, for example, "the number of '1' values in a solution needs to be odd", this quality needs to be maintained for all solutions. A modification to the exploration procedure proposed by Garcia de Andoin and Echanobe (2022) is to use Fredkin gates instead of CNOTs; the point being that if the previous state was valid, then swapping bit values around maintains validity. Whether or not this scalable to more general constraints is questionable. Both exploration schemes are presented graphically in Fig. 14. Contrary to the classical analogue, such an alteration means that two paths have effectively exchanged pheromone values, carrying over to subsequent iterations. If, during convergence, this proves to have been a suboptimal choice, the algorithm will have to spend more time correcting this mistake.

The rest of the algorithm remains classical. After measurement, the collapsed solution undergoes a sanitation process to ensure its validity, by being approximated to the valid solution closest to it in terms of Hamming distance. Termination is decided

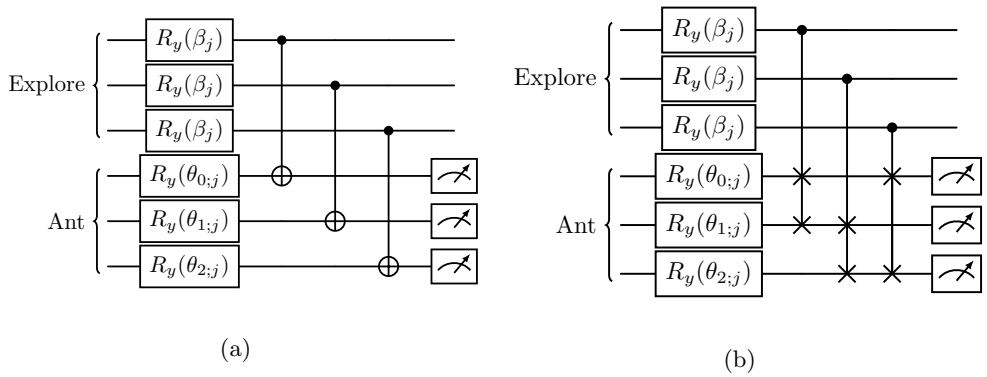


Fig. 14: Path selection in Garcia de Andoin and Echanobe (2022). The ant register initializes to the pheromone angles of iteration j, while the explore qubits initialize to $\beta_j = 2 \arcsin \sqrt{P}_m$, where P_m is the current mutation probability. a shows the exploration scheme for unconstrained problems. The amount of exploration qubits grows linearly with $|V|$. b shows the constrained case. Here, the required exploration qubits grow quadratically to $|V|$, since the possible permutations are $\mathcal{O}(|V|^2)$. The circuits were inspired by Garcia de Andoin and Echanobe (2022) and recreated here

either by the standard iteration cap or by a stagnation counter. This counter increments each time the algorithm does not improve its current best, and after it crosses a certain threshold it determines convergence. Due to the exploration process, the entire search space is always reachable; for this reason, even if this QACO gets stuck in a local optimum, after many iterations it will be able to jump out to a better solution. *Ad infinitum*, this will lead the ant to a global optimum, from which it cannot escape, so the algorithm converges (Garcia de Andoin and Echanobe 2022).

As an ACO selector, this proposal employs a moderate amount of quantum resources. Since the ant register contains $|V| = n$ qubits and only requires single-qubit rotation updates, the main additional overhead arises from the exploration mechanism. For unconstrained problems, the scheme shown in Fig. 14a requires $\mathcal{O}(n)$ qubits and elementary gates. In contrast, the constrained construction proposed by Garcia de Andoin and Echanobe (2022) can be assumed to scale as $\mathcal{O}(n^2)$ in qubits and gates, substituting for the resets and covering the quadratically many pairwise permutations required by the exploration procedure. This remains asymptotically comparable to the memory requirements of many classical ACO variants, although alternative constraint formulations may require even larger circuit widths. At least the quantum memory overhead could likely be reduced by selecting the mutation gates *a priori* through a pseudorandom classical procedure.

5.3.3 Quantum Selector with recursion

A more recent proposal also targets quantum utility on ACO to node selection. The Quantum Selector (QS) proposed in Sarkar et al. (2023, 2024) assumes problem graph $G = (V, E)$ and requires $n = \texttt{max}(\lceil \log_2 |V| \rceil, \lceil \log_2 d \rceil)$ qubits, where d is the maximum

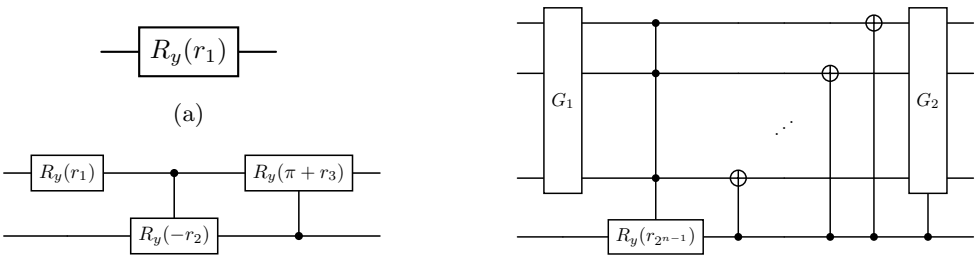


Fig. 15: The ansatz creation pattern for Sarkar et al. (2023). a and b describe the trivial instances where there are 1 and 2 qubits only, respectively. c then defines the recursive algorithm for $n > 2$ instances. G_1 and G_2 are defined by recursively using c again, for the reduced problem of $n - 1$ qubits each. These circuits were inspired by Sarkar et al. (2023) and recreated here

(out)degree in G. The goal of the quantum subroutine is to encode all possible non-visited nodes from current node i as indices in superposition, with their amplitudes encoding selection probabilities. For example, suppose an ant is on the starting node, and has four nodes to move towards. The corresponding edge pheromones lead to selection probabilities $\{P_0, P_1, P_2, P_3\}$, $P_0 + P_1 + P_2 + P_3 = 1$, therefore the desired superposition is the one shown in Eq. (19).

$$|\psi\rangle = \sqrt{P_0}\,|0\rangle + \sqrt{P_1}\,|1\rangle + \sqrt{P_2}\,|2\rangle + \sqrt{P_3}\,|3\rangle \tag{19}$$

Measuring $|\psi\rangle$ on $|1\rangle$, for example, would then decide movement towards the neighbor node labeled 1, with probability P_1, truly at random. This is the main benefit of QS, avoiding pseudo-randomness.

In order to achieve the desired state $|\psi\rangle$ at every step, QS relies on the classical algorithm `GetParameters`. Proposed in Sarkar et al. (2023), this routine accepts a set of amplitudes, sized d a power of two, and outputs a set of angles $\{r\}$, sized $d - 1$. If d is not a power of two, then leftover indices gain the amplitude 0. These angles can then be used to realize the desired state. QS uses a recursive process to reach the correct ansatz that hosts the angles. For $n > 2$-qubit circuits, the ansatz is split in three parts: G_1, that deals with the first $2^{n-1} - 1$ angles, G_2, that deals with the last $2^{n-1} - 1$ angles, and T, that applies angle $r_{2^{n-1}}$. The shape of this split is described graphically in Fig. 15. G_1 and G_2 are defined by recursively using this same procedure for the reduced problem they cover. This recursion stops at $r = 1$ or $r = 2$, where the ansätze are trivial.

Armed with a functional QS, the hybrid QACO in Sarkar et al. (2023, 2024) starts from the uniform superposition of all node indices, in order to select a starting point at random. Then, at each movement opportunity, the non-visited nodes neighboring the current position are assigned probabilities using the standard formula (Dorigo and Gambardella 1997). QS builds the corresponding superposition, measures it and iteratively the path is built. At the end of the traversal, all pheromones are evaporated

classically as normal. If the resulting path is valid,[7] then the participant edges undergo classical deposition by increasing in pheromone inversely proportional to their cost. If not, this run is discarded.

The recursive implementation necessitates substantial quantum resource requirements. Given the qubit count n, $\mathcal{O}(2^n)$ angles are required, leading to an overall circuit size exponential in n. This translates to a gate complexity of $\mathcal{O}(N)$, where $N = |V|$. Consequently, the time complexity of this quantum node selection is comparable to the classical counterpart. Notably, the reliance on MCX gates per recursion call might mean increased ancilla qubit counts or circuit depths, depending on the decomposition method (Barenco et al. 1995). This may pose a practical challenge, especially in NISQ computers.

This proposal has enjoyed some application in real life scenarios, specifically in mobile devices (Sarkar et al. 2023), robotics (Sarkar et al. 2024) and patient monitoring for medicine (Tightiz and Yoo 2024). The scheme is also highly parallelizable to many ants; tour creation as specified is an independent and read-only procedure for the main graph, therefore multiple quantum computers could be leveraged simultaneously to execute multiple path findings (Sarkar et al. 2023). While an interesting idea, the approach remains primarily classical in nature, as quantum resources are used mainly for randomness generation. Consequently, this QACO scheme may not fully exploit quantum advantages and could become computationally expensive due to state preparation overheads.

5.4 Quantum particle swarm optimization algorithm

In this section, the single work relating to QPSO is presented.

The work in Clerc (2024) attempts to translate the binary PSO of Kennedy and Eberhart (1995) into the quantum setting by "replacing bits with qubits". The proposed Hybrid Quantum Particle Swarm Optimizer (HQPSO) defines n d-sized quantum registers x_i for a swarm of n d-dimensional particles, to be iteratively manipulated by R_y rotations. Handling both local and global best topologies, the HQPSO circuit further incorporates another n d-sized registers for the personal bests p_i of each particle, or just one d-sized register for the global best g, respectively.

The algorithm starts by randomly initializing classical velocities and quantum rotations for the x_i registers. The main loop of HQPSO measures the x_i registers multiple times, in order to approximate their internal probability distribution. After this sampling process, each particle inherits a representative bit-string for the given iteration, the basis state that it was most frequently measured in. These representatives are then used classically to compute and compare fitness values with the previous personal or global bests; should a representative be found to improve upon the prior best, its parent register, termed x_{best} here, is "copied" onto the corresponding "best" register via CNOT gates. The examples in Fig. 16 highlight this functionality for both local and global best cases.

[7] Validity is problem-specific. For traveling salesman problems, the algorithm could stagnate due to no free neighboring nodes, but the tour can be incomplete. For Hamiltonian cycle problems, an invalid cycle would not have identical start and finish positions (Sarkar et al. 2023).

Classical postprocessing in each iteration handles the velocity and positional updates, per-qubit. Velocity updates utilize quantum state fidelity in order to enforce similarity between each particle x_i and their personal or global best. The update is described as follows.

$$\text{Local: } v_{ij}^{(t+1)} = wv_{ij}^{(t)} + c_{ij}^{(t)}\left(1 - 2\|g_j x_{ij}\|^2\right) + c_{ij}^{(t)}\left(1 - 2\|p_{ij} x_{ij}\|^2\right) \tag{20}$$

$$\text{Global: } v_{ij}^{(t+1)} = wv_{ij}^{(t)} + c_{ij}^{(t)}\left(1 - 2\|g_j x_{ij}\|^2\right) \tag{21}$$

The term w describes the inertial weight hyperparameter, while $c_{ij}^{(t)}$ are uniformly random positive numbers. The positional update then passes the informed velocities through the mapping

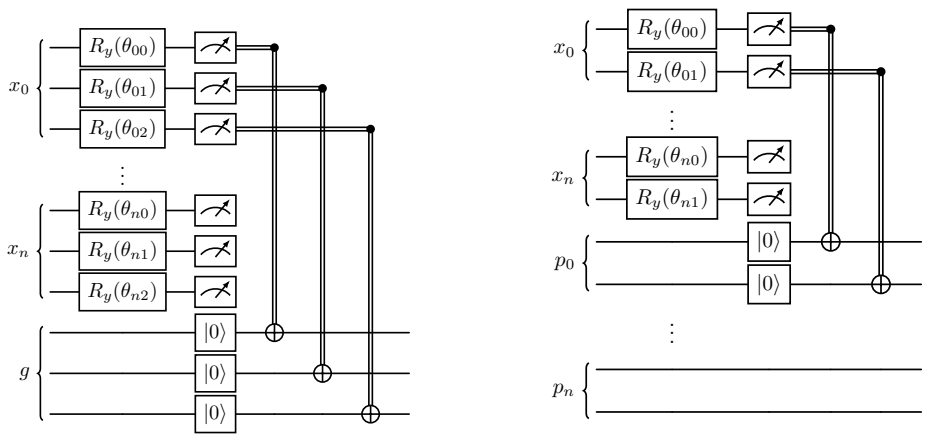
$$\theta_{ij} = \pi\left(\frac{2}{1 + \exp\left(-\lambda v_{ij}\right)} - 1\right) \tag{22}$$

in order to reach the rotation angles θ_{ij} that will drive the particle update quantumly via R_y gates. The proposal suggests setting the hyperparameter λ to 2. By incorporating fidelity inside the velocity update, HQPSO attempts to enhance exploitation, since particles that deviate more from the current best are punished accordingly during the R_y rotation. At the same time, the nature of the rotation biases each qubit state towards one bit-value, but in general leaves a complementary nonzero amplitude to the opposite bit value. In this way, the algorithm also indirectly favors early exploration, as this amplitude will affect the next iteration's measurement step and has therefore the potential to enrich the sampling with more diverse candidates.

The algorithm then repeats the pipeline of multiple measurement, statistical identification of representatives, personal and global best updates, velocity updates and positional rotations, until a user-specified termination criterion is met (Clerc 2024).

HQPSO requires $\mathcal{O}(nd)$ qubits, so its quantum memory needs meet classical overheads. The same holds true for the gate count, which incurs a cost of $\mathcal{O}(nd)$ rotations for the x_i registers, as well as $\mathcal{O}(d)$ CNOTs for the copy, per iteration. As reported in Clerc (2024), Grover-like approaches can show comparable or better performance with exponentially less quantum resources.

The proposal in Clerc (2024) tests the algorithm on the binary restrigin function, reporting unsatisfactory results. Reasons given for this constitute lack of entanglement in the quantum implementation, as well as significant classical assistance in its operation. Indeed, one can identify several limitations in the architecture. As shown in Fig. 16, the CNOT chain copies a collapsed state of the decided x_{best}. However, the algorithm describes handpicking the representative for each quantum individual by statistical sampling. It is thus ambiguous what information the "best" register receives regarding x_{best}, since it is in a classical state that cannot be generally assumed to be the representative. Moreover, the personal and global best registers are never read; if their existence is required solely for value bookkeeping, one might consider classical memory a cheaper alternative. Finally, the fidelity computation is implied to be realized classically, via the approximated distribution. This poses an intractable goal for larger problems, since the distribution grows exponentially with qubit count. Quantum methods for computing state fidelity between registers coherently have been studied



(a) (b)

Fig. 16: Swarm evaluation for HQPSO. a describes the global best version, while b the local best. The illustrated example was inspired by Clerc (2024) and recreated here, for a swarm of n three- (a) and two- (b) dimensional particles. Repeated measurement of each x_i member allows the statistical identification of its most frequent bit-string. Based on that, one of the members is deemed the x_{best}, here x_0. Via qubit-wise CNOT chains, the measured x_{best} register is "copied" onto the corresponding "best" register. Any prior values on the latter are reset. The figures show the measured qubit lines turn into double bit lines in order to save space; the shorthand notation here implies that each qubit is measured onto a classical register, and from that the p_0 and g qubit alterations are then informed.

in the literature, and may pose a compelling alternative for HQPSO (Barenco et al. 1997).

HQPSO stands as a notable first attempt at a genuine QPSO. Its implementation primarily leverages quantum randomness and sets the foundation for future ideas.

6 Perspectives on the future of Quantum Meta-heuristics

This section concludes the text by answering the final research question, RQ4, namely *"Which paths hold the greater promise for the future evolution of the field of quantum meta-heuristic optimization?"*. The discussion that follows draws on the findings of the previous analysis, and shifts the focus to prospective directions, design philosophies, and the balance between near-term pragmatism and long-term ambition.

The study in Clerc (2024) ends with an open question of immense weight for the prospect of the field, paraphrased as *"Is it worth trying to translate an inherently classical procedure into the quantum realm, or is it better to create a new quantum algorithm?"*. This dilemma is echoed throughout this review.

On the one hand, hard-quantum-native translations based on Grover techniques seem to nullify some important classical considerations. For one, encoding the entire search space into exponentially smaller memory expunges the need for generating and maintaining diversity in the population. On the other side of the coin, assuming a proper oracle that faithfully incorporates the fitness optimization check, AA hones the amplitude towards optimal solutions; it needs not invoke mating operators and it cannot be misdirected towards subpar solutions. In that way, the entire exploration-exploitation trade-off disappears, since the search considers every possible case in parallel and reinforces the probability of the desired ones, to the detriment of others. Derivative considerations such as premature convergence similarly lose meaning; the required number of Grover iterations are tied to circuit width, and after they happen, non-targeted solutions are improbable to be measured. In that way, proposals like Malossini et al. (2008) and Udrescu et al. (2006) capture an idea at the heart of QC; mainly, that the focus should be on *identification* of correct solutions, no longer on *creation*. That being said, assuming a "proper oracle" is quite a big ask; whereas in the classical regime, bottlenecks centered around maintaining an adequate exploration-exploitation balance, in the quantum setting this would transfer to resource costs.

On the other hand, completely removing mating operators from the quantum definition raises a valid concern: is the focus even on meta-heuristic optimization anymore? Even middle-ground solutions like SaiToh et al. (2013) that aimed to reintroduce recognizable genetic operators into AA eventually had to part ways with crossover again, due to the Grover mechanism (Kim and Ahn 2022b). The quantum mindset appears to "fight back" the classical one. Conversely, algorithms that insist on recognizable subroutines from classical inspiration, such as Acampora and Vitiello (2021) and Ibarrondo et al. (2024), blur their quantum advantage by defining multiple quantum individuals. Their manipulation is arguably more classical than quantum, and whether the evolution transfers to the "true", internal individuals in their superpositions remains ambiguous. MNDAS (Ghosh et al. 2022) could serve as a balanced example, where quantum ideas are reconciled with recognizable pheromone functions. However, the intractability of that algorithm's classical preprocessing makes this balance less convincing.

An inverse effect tends to emerge for softer quantum-native translations that avoid AA. As the analysis shows, these formulations rely heavily on the probabilistic effect of state collapse, driven through iteratively biased amplitude updates. Superposition then acts more as a persistent source of stochastic diversity, since it generally assigns a nonzero measurement probability to every possible candidate solution. This continuous reintroduction of randomness into the search process via measurement enhances exploration and aleviates concerns regarding premature convergence, but poses the opposite risk: potential for insufficient exploitation pressure and slow convergence toward high-quality solutions. A soft translation seems to be preferrable for partial, hybrid implementations; indeed, the equal prevalence of classical assistance there can help balance the exploitation defecit, enabling the quantum routines to contribute their improvement, typically in sampling and solution quality.

Another dimension concerns the extent to which proposals should adhere to NISQ era limitation. In recent years, proposals have shifted towards shallow circuits, rotation

gates and application to isolated parts of the overall optimizers. While the need for experimental validation on the only currently available computers is understandable, this mindset moves away from true quantum advantage. The most celebrated quantum algorithms, such as Grover's (Grover 1996, 1998) and Shor's (Shor 1994, 1997), were not conceived with hardware limitations in mind; they were designed to solve problems, and hardware followed. It is important for all QC-related algorithmic fields, and especially that of meta-heuristic optimization, to act similarly. Indeed, proposals such as QGS (Acampora et al. 2022b), QPI (Kim and Ahn 2022a) and QS (Sarkar et al. 2023) define convincing versions of isolated parts of GA, but their comparatively costly foundation suggest they could play a larger role in the optimizer. On the other hand, ideas such as Acampora et al. (2022a) and Martinez and Zich (2024), although elegant, are perhaps too simplified, sparking debate on whether they are truly preferable versus the classical counterpart. Recent remarkable progress (Microsoft Azure Quantum et al. 2025; Google Quantum AI and collaborators 2025) encouragingly suggest that QC is entering a more scalable era of Fault-Tolerant Quantum Computers (FTQC). Thus, algorithmic design should be more forward-looking, built with fault tolerance in mind, rather than exclusively constrained to NISQ.

The potential benefit to be gained from a more extensive exploitation of quantum mechanisms becomes more apparent when quantum meta-heuristics are contextualized relative to advances in modern classical optimization. Multi-threaded, multi-core and GPU-accelerated implementations of many meta-heuristic algorithms, including GA, ACO and PSO, have received substantial study, often yielding significant practical improvements in runtime and scalability through distributed and parallel execution (Alba 2005). At first glance, such developments may appear to provide a natural classical analogue to the notion of quantum parallelism discussed throughout this analysis. However, whereas classical parallelism primarily seeks acceleration through the distribution of computation across multiple processing resources, genuinely quantum approaches aim to alter the structure of the search process itself through the exploitation of quantum mechanisms, potentially yielding asymptotic improvements in computational complexity. As discussed in Section 5, properly quantum implementations may, in certain cases, achieve substantially different trade-offs in memory and computational complexity relative to their classical counterparts. Consequently, quantum meta-heuristics should be viewed as an orthogonal algorithmic paradigm to classical parallel implementations, with the potential to access asymptotic regimes inaccessible to classical parallelism alone. A more comprehensive comparative treatment of classical parallel meta-heuristics and quantum optimization frameworks remains an important direction for future dedicated study, but falls beyond the scope of the present review.

Finally, one must ask: which meta-heuristics attract future attention? This review shows that QGA currently dominates, with QACO having made a modest appearance, and the rest essentially untouched. Thus, it would be easy to conclude that QGA holds all the promise. However, looking back at the results of Sec. 3, one can conclude that there is a market for every meta-heuristic to be translated in the quantum realm. It would be unwise, therefore, to ignore them. Every one of the considered meta-heuristics holds value, and the field stands to benefit from diversity. For this

reason, this review identifies three distinct avenues for interested researchers of the field. The first and safest choice is further development of QGA, offering an existing solid foundation of previous works to inspire intuition and orient algorithmic thinking. Second is the continuing inception of QACO, an area of less established background that nevertheless allows for more ambitious and creative work. Lastly, development of any of the remaining meta-heuristics, enhancing the presence of SI in the quantum regime, promises novelty. Among them, QPSO perhaps stands as the most sought after optimizer, and its quantum-native formulation, although challenging, could be a breakthrough. It is clear that not one path is "better"; exploration on all of them is worthwhile, as progress in one inevitably enriches the others.

The authors argue that further progress in the field of quantum meta-heuristic optimization will likely follow two complementary directions. On the "hard quantum" side, qualitative advances may depend on moving beyond direct classical analogues and embracing a more quantum-native and fault-tolerant perspective. In this direction, proposals should aim at defining novel and truly quantum procedures, that meaningfully leverage the unique features of QC, particularly entanglement and quantum parallelism, while building upon established quantum primitives, such as AA, quantum phase estimation and potentially the quantum Fourier transform. The reviewed proposals suggest that the most promising approaches reinterpret optimization through these native quantum mechanisms, rather than attempting to reproduce classical dynamics faithfully. Such designs may therefore progressively move away from traditional concepts such as explicit mating operators or population-based evolutionary dynamics. On the opposite side of the spectrum, the "soft quantum" approach favors hybrid proposals in which QC plays a supporting role, primarily through parallel evaluation and genuine randomness. In such settings, QC may be better viewed as an "exotic multiprocessor", capable of handling specific, selected parallel computations more efficiently, and simultaneously offering a source of high-quality randomness, while the broader optimization logic remains classical. The surveyed literature indicates that, in merging the two computational paradigms, meta-heuristic strategies appear to benefit more when one acts as the dominant framework, while the other complements it selectively, rather than when both participate equally.

From the above, the field of quantum-implementable meta-heuristic algorithms appears poised for significant development, driven by an increasing maturity of QC and the scientific community's growing interest in genuine quantum utility in optimization. The central question of this review, mainly *"How can bio-inspired meta-heuristic optimization be realized on quantum computers?"*, has led to the examination of fifteen classical meta-heuristics for attempted translation into quantum analogues. The analysis reveals that, while most existing efforts remain within the quantum-inspired regime, a distinction subtle to assess, early quantum implementations demonstrate the feasibility and promise of this direction. The leading examples currently include GA, ACO and PSO. Quantum meta-heuristic optimization stands as a compelling area of study for researchers, offering substantial room for novel ideas as the field gradually transitions from quantum-inspired adaptation toward genuinely quantum approaches to optimization.

Acknowledgements. The authors acknowledge the application of the LaTeX package Quantikz (Kay 2023) throughout the paper for the graphical rendering of quantum circuits.

Declarations

- Funding
  No funding was received for conducting this study.
- Competing interests
  The authors of this manuscript declare no competing interests.
- Ethics approval and consent to participate
  Not applicable.
- Consent for publication
  All authors have read and consent to the publication of this manuscript.
- Data availability
  All relevant data to this study are offered freely and openly in:
  https://github.com/asimakiskydros/Quantum-Metaheuristic-Optimization
- Materials availability
  Not applicable.
- Code availability
  Code produced to aid in conducting this study is offered freely and openly in:
  https://github.com/asimakiskydros/Quantum-Metaheuristic-Optimization
- Author contribution
  A.K. conceived the idea for the study, performed the literature search and the data analysis and wrote the manuscript. N.L.T. and N.K. supervised the study's direction and critically revised the work.

Appendix A Genetic Algorithm

The Genetic Algorithm (GA) is one of the most established meta-heuristic optimization methods, inspired by Darwinian evolution and natural selection (Holland 1992; Katoch et al. 2021). In GA, a population of candidate solutions, which are called *chromosomes*, evolves iteratively under operations mimicking biological reproduction. The quality of each chromosome is determined by the *fitness function* $f(x)$, which encodes the problem at hand.

Each iteration of GA, also called a *generation*, is subdivided into multiple phases which simulate the biological evolution of actual chromosomes and typically include *initialization*, the creation of a diverse population of chromosomes, *selection* among those individual chromosomes in some way according to their fitness, and *mating*, which maps the old population into a new one, enriched by the genetic information of the selected individuals. It is common to split the mating step further into *crossover*, which combines segments of selected chromosomes (called *parents*) to produce *offspring*, a process that promotes *exploitation* of existing good traits, and *mutation*, which induces small random changes in the population, promoting the *exploration* of undiscovered traits and helping the algorithm to escape from local optima. Whether

to prioritize exploration or exploitation is a major trade-off in GA, since both are desirable and necessary for optimal convergence but they can contradict each other's effects if a balance is not maintained. An additional optional trait of GAs is *elitism*, which ensures that some high quality individuals from the previous population survive to the next unscathed; this is another method of enhancing exploitation.

Repeated application of the above routines, in a way that maintains a healthy exploitation-exploration balance, gradually converges to the production of a chromosome that optimizes f. GA is a method flexible and widely applicable to many problems, however it often requires careful tuning and may converge slowly or prematurely (Katoch et al. 2021).

Appendix B Ant Colony Optimization

Ant Colony Optimization (ACO) is a prime example of Swarm Intelligence (SI). Specifically, it is an optimization method inspired by the emerging behavior of ant colonies during foraging, by which the ants gradually coalesce into the shortest paths to food sources by communicating through pheromone trails (Dorigo and Gambardella 1997). ACO is especially applicable to problems that can be mapped into path finding through graphs encoding the problem space, by an almost direct analogue to the observed behavior of the biological situation.

In computational form, ants are described as agents. At each iteration of the algorithm, each agent builds a candidate solution by selecting a path through the problem graph, in a process known as *path construction*. The agents form their paths by probabilistically selecting edges reachable from their current position (graph node) based on existing *pheromone intensity* and problem-specific heuristics. All formed paths are evaluated, enriching the pheromone of their constituent edges according to the fitness of the solution described by the path. This is the first part of the *pheromone update* process, called *deposition*, which naturally prioritizes exploitation. The pheromone update step ends with a global evaporation of all edge pheromones by a certain adjustable amount, so as to encourage exploration as well.

By following this construction-update pattern iteratively, eventually high pheromone levels are honed towards paths describing high-quality solutions, converging the procedure towards optimality. Although undeniably a solid and popular optimization strategy, it too depends heavily on parameter tuning and delicate exploration-exploitation balancing (Ojha et al. 2014).

Appendix C Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based optimization method belonging to the broader class of SI techniques. It was originally proposed in Kennedy and Eberhart (1995), inspired by the collective motion observed in flocks of birds and schools of fish. In PSO, a set of candidate solutions, called *particles*, move through the search space by following simple dynamical rules influenced by both their own experience and that of their neighbors.

Each particle is characterized by a position vector $\vec{x}_i$ and a velocity vector $\vec{v}_i$, which determine its trajectory through the problem space. During the optimization process,

each particle keeps track of the best position it has encountered so far, referred to as its *personal best* $\vec{p}_i$, while the swarm collectively tracks a *global best* $\vec{g}$ (or, in some variants, a neighborhood best). At each iteration, particles update their velocities based on a combination of inertia, attraction towards their personal best, and attraction towards the best-known position in their neighborhood, after which their positions are updated accordingly.

This interaction naturally balances exploration and exploitation. The stochastic components of the velocity update encourage exploration of the search space, while the attraction towards personal and global best positions promotes exploitation of known high-quality regions. The topology of the swarm plays an important role in this balance: global-best configurations tend to converge quickly but risk premature convergence, whereas local-best configurations maintain diversity at the cost of slower convergence.

Through repeated updates of particle velocities and positions, the swarm collectively converges towards regions of high fitness. PSO is widely applicable and conceptually simple, but similarly to other meta-heuristics, it requires careful parameter tuning and may suffer from premature convergence or stagnation (Kennedy and Eberhart 1995; Poli et al. 2007).

Appendix D Quantum Computing Fundamentals

In this section, some fundamentals of Quantum Computing (QC) are presented, in order for this work to be self-contained. This is not an extensive introduction. For a more thorough study, the interested reader is directed to Nielsen and Chuang (2010).

D.1 The quantum bit and quantum registers

The fundamental piece of quantum information is the quantum bit (qubit). A qubit is a two-state quantum system, meaning it has two basis states: the ground state and the excited state. These states are labeled as $|0\rangle$ and $|1\rangle$, respectively, making use of Dirac notation (Dirac 1939). This notation signals that a quantum state is different from a classical bit, specifically the former is a vector and the latter is a scalar. The general state of a qubit is written mathematically as in Eq. (D1).

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad |\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \quad \alpha, \beta \in \mathbb{C} \tag{D1}$$

In Eq. (D1), $|\psi\rangle$ is a linear combination of the basis states. This is what is called as *superposition*. The qubit is not in both basis states at the same time, but instead it is in an intermediate state that is "α much" like $|0\rangle$ and "β much" like $|1\rangle$. α, β are called the *amplitudes* of the qubit. These amplitudes must always follow the rule of Eq. (D2)

$$\alpha^*\alpha + \beta^*\beta = 1, \tag{D2}$$

α^* being the conjugate of α.

Using Hopf coordinates, it is also possible to describe the general state of a qubit as in Eq. (D3).

$$|\psi\rangle = e^{\mathrm{i}\delta}\left(\cos\frac{\theta}{2}|0\rangle + e^{\mathrm{i}\phi}\sin\frac{\theta}{2}|1\rangle\right) \tag{D3}$$

In Eq. (D3), $e^{\mathrm{i}\delta}$ is the *global phase* of the state $|\psi\rangle$, while $e^{\mathrm{i}\phi}$ is its *relative phase*. Due to the phenomenon of the collapse of the state during measurement that will be explained later in this section, the global phase has no physically observable impact, therefore it can be (and in most cases is) completely ignored. Relative phase (simply, the phase), on the other hand, can be leveraged while in superposition to the benefit of algorithms. Maintaining ϕ to integer multiples of π induces a sign flip on the amplitude of $|1\rangle$.

As a complex vector, $|\psi\rangle$ has a conjugate transpose, which is written as $\langle\psi| = \begin{bmatrix}\alpha^* & \beta^*\end{bmatrix}$. Through this notation, the inner and outer products of two quantum systems can also be defined, shown in Eq. (D4) in that order.

$$\langle\psi|\phi\rangle \triangleq \vec{\psi}^\dagger\vec{\phi}, \quad |\psi\rangle\langle\phi| \triangleq \vec{\psi}\vec{\phi}^\dagger, \quad \psi^\dagger \triangleq (\psi^*)^T \tag{D4}$$

Multiple qubits can be chained together to form a register. The size of the register in terms of the number of basis states and/or amplitudes scales exponentially with the number of qubits inside it. To calculate exactly the state of a register, one uses the tensor product of the qubits, shown in Eq. (D5).

$$|\phi\rangle = |q1\rangle\otimes|q0\rangle = \begin{bmatrix}\alpha_1\\ \beta_1\end{bmatrix}\otimes\begin{bmatrix}\alpha_0\\ \beta_0\end{bmatrix} = \begin{bmatrix}\alpha_1\alpha_0\\ \alpha_1\beta_0\\ \beta_1\alpha_0\\ \beta_1\beta_0\end{bmatrix} = \alpha_1\alpha_0|00\rangle + \alpha_1\beta_0|01\rangle + \beta_1\alpha_0|10\rangle + \beta_1\beta_0|11\rangle \tag{D5}$$

$$|reg_1\rangle = |q_{n-1}q_{n-2}\cdots q_0\rangle = \sum_{i=0}^{2^n-1} c_i|i\rangle, \quad \sum_{i=0}^{2^n-1} c_i^* c_i = 1. \tag{D6}$$

A quantum register is by itself a quantum system, now of more basis states, and as such the squares of its amplitudes c_i must also sum to 1. In Eq. (D6), $|i\rangle$ encodes the basis state that as a binary number forms the number i in decimal. For simplicity, $|q_i\rangle \otimes |q_j\rangle$ can be equally written as $|q_i\rangle|q_j\rangle$ or $|q_iq_j\rangle$.

D.2 Entanglement

Entanglement is a state between two or more quantum systems, by which neither can be described fully without the other. In other words, they are interdependent. Mathematically, any system that cannot be written as a train of tensor products is entangled in some manner. It is imperative to note that such correlation between systems implies that there can no longer be distinct amplitudes for the systems involved. Some famous entangled states (Einstein et al. 1935; Dür et al. 2000) are shown in Eq. (D7).

$$|EPR_1\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \quad |W\rangle = \frac{|001\rangle + |010\rangle + |100\rangle}{\sqrt{3}} \tag{D7}$$

The above are just two examples. Entanglement can take a myriad of forms, depending on how it is realized and what its purpose is. As such, its significance is hard to generalize. It should suffice to note that it enables the instantaneous, indirect, parallel and controlled manipulation of entire registers, and it is that resource in fact that makes the aforementioned superposition so powerful in algorithm design. Without entanglement, any superposition can be efficiently modeled on classical computers implicitly, even if its theoretical size is exponential. To capture this, note the following example:

$$|\psi\rangle = [c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8]^T = \begin{bmatrix}\alpha_1 \\ \beta_1\end{bmatrix} \otimes \begin{bmatrix}\alpha_2 \\ \beta_2\end{bmatrix} \otimes \begin{bmatrix}\alpha_3 \\ \beta_3\end{bmatrix} \tag{D8}$$

In Eq. (D8), the described system $|\psi\rangle$ is sized 3 in qubits, and its complete statevector consists of $2^3 = 8$ complex numbers. Because the degree of entanglement in the system is zero, factoring it in a tensor train reveals a compact representation that requires $3 \cdot 2 = 6$ complex numbers instead, linear in the system size. To obtain the total amplitude of any basis state, one needs only choose the correct amplitude from each qubit and multiply them together, a process again linear in the system size. This of course generalizes to quantum systems of any size, and it applies to state manipulation as well, because as it is explained later on, entanglement is invoked by multiple-qubit operations. Single qubit operations on zero-entanglement systems affect the amplitudes of a single qubit, therefore they can be simulated in constant time.

Thus entanglement is the holy grail of QC and what one typically aims to utilize when transitioning an algorithm into quantum. Its absence from a quantum procedure is generally a bad sign and it might signify that there is no real advantage to leveraging QC in the described process.

D.3 Quantum operators

A quantum operator affects the state of one or multiple qubits, evolving their states inside the space formed by their amplitudes, known as the Hilbert space ($\mathcal{H}$) of the system. Unless they have a specific name, quantum operators are symbolized by the letter U, and are defined mathematically by an underlying matrix. Since quantum operators describe manipulations on a closed quantum system, they are unitary. This means that they are linear ($U(\alpha|0\rangle + \beta|1\rangle) = \alpha U|0\rangle + \beta U|1\rangle \ \forall U$) and that they preserve the inner product ($\forall U \ \exists U^\dagger : U^\dagger U = UU^\dagger = I$). $U^\dagger$ here signifies the Hermitian conjugate (or *dagger*) of U. On a linear algebra level, their defining matrices are square, with side exponential to the system size in qubits, and their determinant equals 1, belonging to the Special Unitary group $SU(2)$. Inner product preservation also directly implies their reversibility, which makes QC a reversible computing model.

Some common quantum operators are listed below. For convenience, b is the binary variable, and the notation $|x; y\rangle$ implies $|x\rangle|y\rangle$ in a conglomeration. Describing their operation for the basis states only suffices, because the effect for general superpositions can be inferred by the aforementioned linearity principle. Note that all gates can be extended with controls.

- Bit flip (NOT or X): $NOT|b\rangle = |1 - b\rangle$
- Phase flip (Z): $Z|b\rangle = (-1)^b|b\rangle$

- Hadamard (H): $H|b\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^b|1\rangle\right)$
- Real rotation (R_y): $R_y(\rho)|b\rangle = \cos\frac{\rho}{2}|b\rangle + (-1)^b \sin\frac{\rho}{2}|1-b\rangle$
- SWAP: $SWAP|\psi;\phi\rangle = |\phi;\psi\rangle$
- CNOT: $CNOT|c;t\rangle = |c;c\oplus t\rangle$
  Performs NOT on a target qubit on the condition that a second (control) qubit has value $|1\rangle$.
- Fredkin (or CSWAP): Similar to CNOT, performs a SWAP between two target qubits on the condition that a third (control) qubit has satisfactory value.
- MCX: The generalization of CNOT for multiple control qubits. It is also known as the Toffoli family, although the name Toffoli for a gate with no other context typically describes a NOT with two controls. Can also be generalized to target-flip multiple qubits and/or perform a different operation than NOT.

The effect of any elemenetary operator happens in constant time, no matter how many qubits it spans.[8] If consequent operators act on disjoint sets of qubits, then they can act in parallel, forming a computational *step*. Therefore, in quantum algorithm design and analysis, memory complexity is expressed in terms of required qubits, and time complexity in terms of required steps.

By definition, once a single operator has affected two or more qubits, they are entangled, in a manner dependent on the operator and their previous state. This need not cover the entire system and in fact it is possible to have an entangled subsystem that is otherwise unentangled as a whole with the rest of the system. Entangled effects can also stack.

D.4 Measurement and state collapse

To extract information from a quantum system, a measurement on it must be performed, on a specific basis. Most often, plain "measurement" refers to measuring over the computational basis $\{|0\rangle, |1\rangle, \cdots, |d-1\rangle\}$. Thus a single binary number is expected as post-measurement output. Following the notation of Eq. (D6), each basis state $|i\rangle$ will appear after measurement of the entire register with probability $c_i^* c_i$, and consequent measurements on that register without resetting it will give the same outcome every time. This fact highlights that the superposition of the register has *collapsed* on the basis state $|i\rangle$, and classical information received is equal in bits to the amount of qubits measured.

Often, useful information is encoded in the probability distribution formed by the final statevector. In other words, one would ideally want to know the final amplitudes. While this is impossible in practice, it is possible to statistically approximate the distribution by repeating the entire algorithm and measurement from start to finish multiple times. This is known as *multi-shot measurement*, a shot being a single measurement iteration. The amount of shots is often independent of the algorithm complexity but intimately tied to the expected error.

[8] What counts as an elementary operator depends on the physical architecture of each quantum computer. For this work, all gates appearing in the itemization list apart from MCX count as elementary, and all other operators must decompose to a combination of those.

A system can also be partially measured, considering only a subsystem. This partially collapses the statevector post-measurement, which, depending on the entanglement of the previous state, may be beneficial as an algorithmic tool. The probability of the measured result is now dependent on the sum of the amplitudes of all basis states that exhibit the received pattern in the selected positions.

D.5 Coherence

Coherence, the natural, "hidden" evolution of the state, closed off from the environment, allows for state manipulation benefiting from the tools previously described. For this reason, maintaining coherence for as long as possible is usually desired. Apart from having to play with the rules of reversibility and unitary operators, coherence also poses more general challenges to algorithm design, which stem from the nature of unitarity and the environment. Some of those are the following.

1. The *No-Cloning theorem* (Wootters and Zurek 1982): $\nexists U : U|\phi;0\rangle = |\phi\rangle|\phi\rangle \ \forall|\phi\rangle$. A perfect copier does not exist. By perfect here, it is meant that the amplitudes of $|\phi\rangle$, although unknown, are transferred to the new state exactly, and no entanglement exists between the two states afterwards. Approximate copiers do exist however (Alvarez-Rodriguez et al. 2014; Bužek and Hillery 1996; Werner 1998).
2. The *No-Deleting theorem* (Kumar Pati and Braunstein 2000): $\nexists U : U|\phi;\phi\rangle = |\phi\rangle|0\rangle \ \forall|\phi\rangle$. This is the time-reversal of the above theorem, and it posits that quantum information is conserved. An eraser of a "quantum variable" doesn't exist either. Nevertheless, it is common for algorithm makers to introduce a "reset" operation as a subroutine. This either means a non-unitary operation, most often containing a measurement which thus makes the state known, or it implies the existence of multiple ancilla qubits, and "reset" means that one forgets about the used qubit and uses a fresh one later on.
3. Quantum noise and errors: Quantum information is volatile, subject to errors due to hardware imperfections, environmental interactions or thermal effects. Over time, the system will decohere, and the information it holds will transition to a classical state. This fact poses a time ceiling on all quantum algorithms. However, since this issue originates and is combated at the hardware level, unless an algorithm is set to run for time on the order of seconds, for theoretical works these considerations can be ignored.

D.6 The circuit model

As a means to better represent and analyze quantum algorithms, most often the circuit model of computation is employed. In it, the time-dependent evolution of all qubits is shown as "wires" passing through "gates", representing the operators. This model facilitates the thought process because it allows for encapsulation tricks and slight notation abuse. Fig. D1 highlights the main symbols. Single lines represent lone qubits, but a bundle of n qubits can also be compiled into a single "bus". Double-lines represent bits. Gates are drawn as squares with their designated label inside, unless

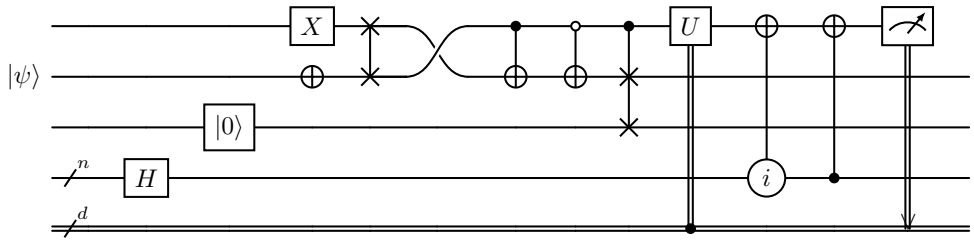


Fig. D1: An example quantum circuit. It compiles the most common notational quirks, its actual operation in this particular instance is not of concern here. From left to right, the symbols represent: state preparation, multi-qubit bundle (bus), Hadamard transform (a Hadamard gate on each of the n wires), nonunitary reset operation (sets the qubit back to $|0\rangle$), NOT, SWAP, line permutation, CNOT, CNOT controlled on $|0\rangle$, Fredkin, classically controlled U, MCX controlled on $|i\rangle$, MCX controlled on $|2^n - 1\rangle$, measurement write

they have a unique symbol. NOT, for example, is also drawn as a round target. If a single-qubit gate is placed ontop of a bus, then that gate is repeated on each singular line in the bus. The case for multi-qubit gates requires context. In Fig. D1, the fourth and fifth steps showcase SWAP gates. In this work, the former implies an actual gate application, while the latter an implicit SWAP, a label permutation. Controls can be visualized in multiple ways depending on the function. Sometimes, it is convenient that a control satisfies on a value other than $|1\rangle$. For example, the open circle signifies satisfaction at $|0\rangle$. For multiple-control gates controlled on buses, a circle with a number or variable can highlight the specific basis state being targeted, while a full or open circle implies activation on the most excited state and the ground state, respectively. Double vertical lines represent classical read-write operations. What is actually read or written should be labeled. The write arrow on measurements is oftentimes omitted. All lines start from the ground state, unless explicitly specified otherwise. Finally, in algorithm design, it is sometimes convenient to delineate the qubit reset as a square with the reset value inside, eg $|0\rangle$. This action is non-unitary and it implies a measurement.

Bear in mind that, although this looks like a reversible logic circuit, and is commonly treated as such, it is purely an analogy. The quantum systems representing the logical qubits are never moved and never "pass" through anything. They are manipulated in place. Similarly, the "gates" aren't physical chips that accept electrical signal and output some other signal, but they are actually manipulations on the quantum systems of their host lines. Their order of operation matters and it goes left-to-right.

Appendix E Grover-based methods

E.1 Grover search

Quantum search, formally known as Grover's algorithm (Grover 1996), is an algorithm of monumental importance to QC, exhibiting the promise of Quantum Supremacy. Usually described as fast searching a desired element inside an unsorted database, it more correctly detects, with high probability, the argument of a function that produces a given output, with only $\lfloor \frac{\pi}{4}\sqrt{N} \rfloor \rightarrow \mathcal{O}(\sqrt{N})$ calls to a black box (oracle) implementing that function, where N is the domain size. Given that nothing is assumed about the structure of the function's domain, classical algorithms can only ever achieve a cost of $\mathcal{O}(N)$ function calls, thus Grover search offers a quadratic improvement.

In order to apply Grover's search, two unitary operators need be defined, an *oracle* and a *diffuser*. The first acts on the uniform superposition and marks those basis states that satisfy some target τ. This target can be, in principle, anything; it can be that the markable states j are numerically above a given threshold, that they give a specific value c for a given function f, or something more complex. This marking is performed as a phase flip on the satisfactory states, typically a sign flip. After the operation of the oracle, the desired states have negative amplitude, while the "garbage" states have positive amplitude. The diffuser then performs an inversion of all the amplitudes around their mean. Mathematically, these operators are described as in Eq. (E9).

$$O_\tau = I - 2 \sum_{j \text{ satisfies } \tau} |j\rangle \langle j| , \quad G_D = 2 |s\rangle \langle s| - I \tag{E9}$$

In the above, $|s\rangle = (H|0\rangle)^{\otimes\sqrt{N}} = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$. As an example, supposing that the oracle marks a single state $|j\rangle$, applying the diffuser immediately afterwards for the first time leads to

$$G_D |s'\rangle = (2|s\rangle\langle s| - I)(\frac{1}{\sqrt{N}} \sum_{i \neq j} |i\rangle - \frac{1}{\sqrt{N}} |j\rangle) = \frac{N-4}{N\sqrt{N}} \sum_{i \neq j} |i\rangle + \frac{3N-4}{N\sqrt{N}} |j\rangle \tag{E10}$$

The desired state $|j\rangle$ increases in amplitude, and thus in probability to be measured. For the rest of the basis states, it had a destructive effect. Thus Grover search hones in the amplitude towards the marked state(s). $\mathcal{O}(\sqrt{N})$ repetitions of this combo prior to measuring maximizes the likelihood that the single-shot final measurement will yield a correct solution (Grover 1996; Karafyllidis 2015).

A major hurdle in this procedure is that, should the satisfactory states be more than one, lets say t many, then the required number of Grover iterations is actually $\lfloor \frac{\pi}{4}\sqrt{\frac{N}{t}} \rfloor \rightarrow \mathcal{O}(\sqrt{N/t})$. It can neither be more nor less than that many, because amplitude evolution follows a $\sin^2$ law (Boyer et al. 1998). Too many iterations and the probability of the desired states will begin to drop again. In most cases, t can hardly be known *a priori*.

Another important remark is that this strategy is $\mathcal{O}(\sqrt{N})$ in function *calls*, not total complexity. In fact, since the function defines the entire problem, its implementation complexity is absent in the general description of Grover search. Whether

the desired function is efficiently implementable as a quantum operator is a different problem in itself.

E.2 BBHT algorithm and Maximum Finding

The BBHT algorithm aims to tackle the aforementioned problem of the unknown value of t in standard Grover search. The idea is to heuristically guess the required amount of Grover iterations, obtain a result, and update the guess if it is not a valid solution (Boyer et al. 1998). Alg. 6 summarizes the procedure.

Algorithm 6: Boyer-Brassard-Høyer-Tapp algorithm

Data: A problem-specific oracle O, the domain size N.
Result: A solution to the problem, found in time $\mathcal{O}(\sqrt{N/t})$.

$m \leftarrow 1$
$\lambda \leftarrow$ some value in $(1, 4/3)$, e.g. $6/5$
repeat forever
  Apply $i \sim \mathcal{U}\{x \in \mathbb{Z} : x \in [0, m)\}$ iterations of Grover search using O, measuring to j
  if j *is a solution* **then return** j
  $m \leftarrow \texttt{min}(\lambda m, \sqrt{N})$
end

This is one way to re-enable the application of Grover search when t is unknown. BBHT also maintains the efficient cost of $\mathcal{O}(\sqrt{N/t})$ (Boyer et al. 1998). It is a widely used algorithm, fit for many situations. One application common in quantum optimization is the *maximum finding* algorithm (Durr and Hoyer 1999). Maximum finding leverages chained repetitions of BBHT to iteratively approximate the maximizer of a problem function f, utilizing the oracle

$$O_l |i\rangle = \begin{cases} -|i\rangle, & f(i) > f(l) \\ +|i\rangle, & \text{otherwise} \end{cases} \tag{E11}$$

Equivalently, the oracle of Eq. (E11) could be defined to mark $f(i) \geq f(l)$, or $f(i) \leq f(l)$ for the minimization problem.

Algorithm 7 identifies, with high probability, the optimizer of a given function. The required iterations (not counting oracle cost or individual BBHT iterations) scale $\mathcal{O}(\sqrt{N})$, specifically by $22.5\sqrt{N} + 1.4 \log_2^2 N$ (Durr and Hoyer 1999).

E.3 Amplitude Amplification

Amplitude amplification (AA) is a direct generalization of Grover's algorithm, that allows benefiting from quadratic complexity gains in problems other than searching (Grover 1998; Brassard and Hoyer 1997). As explained in Appendix E.1, standard quantum search begins from the uniform superposition, which it identifies as the search

Algorithm 7: Dürr-Høyer Maximum Finding algorithm

Data: Oracle following Eq. (E11), initial threshold $l \sim \mathcal{U}\{0, 1, \cdots, N-1\}$, where N is the problem function's domain size.

Result: The minimizer of the problem function with high probability.

repeat $\mathcal{O}(\sqrt{N})$ **times**

  Run BBHT on $\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle_{target} |l\rangle_{threshold}$ using the oracle of Eq. (E11)

  This yielded a measurement j of the target register. Since $f(j) > f(l)$, set j as the new l

end

return l

space. This means that the algorithm starts from the ground state $|0\rangle^{\otimes\sqrt{N}}$ affected by the Welsh-Hadamard transform $H^{\otimes\sqrt{N}}$ before starting Grover iterations. This choice has direct implications on the shape of the diffuser, defined as $G_D = H^{\otimes\sqrt{N}} I_0 H^{\otimes\sqrt{N}}$. This is equivalent to the definition shown in Eq. (E9), leveraging $I_x = I - 2|x\rangle\langle x|$.

Generalizing this exact picture, it is valid to choose the initial state and the transform differently (Grover 1998). For any unitary U and easy-to-prepare initial state $|\gamma\rangle$, the Grover operators can be generalized to $O_\tau = I_{j:j \text{ satisfies } \tau}$, $D = U I_\gamma U^\dagger$. Absorbing all operators into a single one, the amplification operator starting from $|\gamma\rangle$ is shown in Eq. (E12).

$$Q = I_\gamma U^\dagger O_\tau U \tag{E12}$$

The rest of the process remains similar to standard Grover. This generalization proves that the advantage of quantum searching is not inherently tied to the Hadamard transform (Grover 1998), and it facilitates the conversion of a given problem into a search one as a valid quantum algorithmic approach.

Standard Grover's caveats carry over to AA. In particular, apart from the required oracle being implementable as an operator, now the desired transform U and its inverse $U^\dagger$ must also be available.

Appendix F Quantum Annealing

Quantum Annealing (QA) is a hardware-based process utilizing the Adiabatic model of QC, a different computational model to the circuit one. Quantum computers implementing QA center their internal architecture around solving the `ISING MINIMIZATION` problem:

$$\begin{aligned} \texttt{argmin}\ \ E(s) &= \sum_{i \in V} h_i s_i + \sum_{(i,j) \in E} J_{ij} s_i s_j \\ \texttt{s.t.}\ \ s_i &= \pm 1\ \forall i \in V \end{aligned}$$

for a given graph $G = (V, E)$ and an input Hamiltonian H describing biases $h = \{h_i : i \in V\}$ and couplings $J = \{J_{ij} : (i,j) \in E\}$. `ISING` is $\mathcal{NP}$-hard (Barahona 1982),

making it computationally difficult in the classical regime but highly applicable for many different problems that reduce to it (Lucas 2014).

QA employs a strategy reminiscent of classical Simulated Annealing (SA), but instead of thermal fluctuations it uses quantum tunneling effects to drive the optimization (King et al. 2019). The main idea is to start from a simple point and evolve towards the desired but complicated point. By "points", it is meant ground states of Hamiltonians. Since Hamiltonians encode the evolution of a system, and thus represent the problem at hand, ground states of them naturally point to problem solutions. QA realizes this idea by exposing a composite Hamiltonian, shown in Eq. (F13) to a transverse field, that starts with strong intensity, exciting quantum fluctuations in the system, and gradually relaxes.

$$\mathcal{H}(\tau) = A(\tau)\mathcal{H}_{start} + B(\tau)\mathcal{H}_{target} = A(\tau)\sum_i \sigma_i^x + B(\tau)\left(\sum_i h_i\sigma_i^z + \sum_{i,j} J_{ij}\sigma_i^z\sigma_j^z\right) \tag{F13}$$

In Eq. (F13), τ describes the time evolution of the chosen schedule, and it ranges from $\tau = 0$ to $\tau = 1$. A and B are functions implementing the schedule of the annealer, chosen so that $A(0) \gg 0, B(0) = 0$ and $A(1) = 0, B(1) \gg 0$. σ_i^P are Pauli operations on qubit i, specifically

$$\sigma_i^x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_i^z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{F14}$$

The initial Hamiltonian $\mathcal{H}_{start}$ is chosen so that preparing the system on a ground state is easy. Its definition in Eq. (F13) is a common choice because its ground state then becomes $|+\rangle^{\otimes n} = H^{\otimes n}|0\rangle^{\otimes n}$, trivial to prepare (King et al. 2019). By the Adiabatic principle (Born and Fock 1928), if the evolution of Eq. (F13) happens slowly enough (*adiabatically*), then at $\tau = 1$ the system will arrive in a ground state of $\mathcal{H}_{target}$.

D-Wave quantum computers specialize on quantum annealers. The problem graph G can be encoded inside the hardware by the user. The architecture of the machines defines an internal sparse graph that contains thousands of physical qubits and thousands of couplings between them. The user can then program the biases h and couplings J depending on the problem, as well as define the logical qubits of the problem by merging multiple physical qubits together, thus achieving the desired problem graph connectivity (Bernal et al. 2019).

It becomes evident that QA is designed to solve a specific class of problems, but that class is so versatile algorithmically that it is applicable to many domains. For this reason, while it resembles SA in spirit, this study does not consider it as a standalone quantum meta-heuristic optimizer. Rather, it is treated as a quantum hardware primitive, to be used as a subroutine.

References

Acampora G, Schiattarella R (2023) On the effect of quantum noise in quantum genetic algorithms. In: 2023 IEEE Congress on Evolutionary Computation (CEC). IEEE, Chicago, pp 1–8, https://doi.org/10.1109/cec53210.2023.10254078

Acampora G, Vitiello A (2021) Implementing evolutionary optimization on actual quantum processors. Inf Sci 575:542–562. https://doi.org/10.1016/j.ins.2021.06.049

Acampora G, Vitiello A (2023) Improving quantum genetic optimization through granular computing. Granul Comput 8(4):709–729. https://doi.org/10.1007/s41066-022-00348-9

Acampora G, Vitiello A (2024) Improving quantum genetic algorithms through recursive search space exploration. In: 2024 IEEE Congress on Evolutionary Computation (CEC). IEEE, Yokohama, pp 1–8, https://doi.org/10.1109/cec60901.2024.10612000

Acampora G, Schiattarella R, Vitiello A (2022a) Quantum mating operator: A new approach to evolve chromosomes in genetic algorithms. In: 2022 IEEE Congress On Evolutionary Computation (CEC). IEEE, Padua, pp 1–8, https://doi.org/10.1109/cec55065.2022.9870425

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

Acampora G, Chiatto A, De Luca S, et al (2023) Application of quantum genetic algorithms to network signal setting design. In: 2023 IEEE Congress on Evolutionary Computation (CEC). IEEE, Chicago, pp 1–8, https://doi.org/10.1109/cec53210.2023.10254158

Acampora G, Chiatto A, De Luca S, et al (2024) Application of quantum genetic algorithms to connected and electric vehicles energy consumption optimization. In: 2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI). IEEE, Milano, p 530 – 535, https://doi.org/10.1109/rtsi61910.2024.10761791

Acampora G, Chiatto A, Schiattarella R, et al (2026) Quantum artificial intelligence: A survey. Comput Sci Rev 59:100807. https://doi.org/10.1016/j.cosrev.2025.100807

Alba E (2005) Parallel metaheuristics: A new class of algorithms. John Wiley & Sons

Alhayan F, Alshuhail A, Ismail AOA, et al (2025) Artificial intelligence-driven cybersecurity: enhancing malicious domain detection using attention-based deep learning model with optimization algorithms. Sci Rep 15(1):23806. https://doi.org/10.1038/s41598-025-99420-y

Alvarez-Rodriguez U, Sanz M, Lamata L, et al (2014) Biomimetic cloning of quantum observables. Sci Rep 4:4910. https://doi.org/10.1038/srep04910

Garcia de Andoin M, Echanobe J (2022) Implementable hybrid quantum ant colony optimization algorithm. Quantum Mach Intell 4(2):12. https://doi.org/10.1007/s42484-022-00065-1

Ardelean SM, Udrescu M (2022a) Circuit level implementation of the reduced quantum genetic algorithm using qiskit. In: 2022 IEEE 16th International Symposium on Applied Computational Intelligence and Informatics (SACI). IEEE, Timisoara, p 155 – 160, https://doi.org/10.1109/saci55618.2022.9919519

Ardelean SM, Udrescu M (2022b) Graph coloring using the reduced quantum genetic algorithm. PeerJ Comput Sci 8:e836. https://doi.org/10.7717/peerj-cs.836

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

Barahona F (1982) On the computational complexity of ising spin glass models. Journal of Physics A: Mathematical and General 15(10):3241. https://doi.org/10.1088/0305-4470/15/10/028

Barenco A, Bennett CH, Cleve R, et al (1995) Elementary gates for quantum computation. Phys Rev A 52:3457–3467. https://doi.org/10.1103/PhysRevA.52.3457

Barenco A, Berthiaume A, Deutsch D, et al (1997) Stabilization of quantum computations by symmetrization. SIAM Journal on Computing 26(5):1541–1557. https://doi.org/10.1137/S0097539796302452

Bernal DE, Booth KEC, Dridi R, et al (2019) Integer programming techniques for minor-embedding in quantum annealers. arXiv:1912.08314

Born M, Fock V (1928) Beweis des adiabatensatzes. Z Physik 51(3):165–180. https://doi.org/10.1007/BF01343193

Boushaki SI, Kamel N, Bendjeghaba O (2018) A new quantum chaotic cuckoo search algorithm for data clustering. Expert Systems With Applications 96:358–372. https://doi.org/10.1016/j.eswa.2017.12.001

Boyer M, Brassard G, Høyer P, et al (1998) Tight bounds on quantum searching. Fortschritte der Physik 46(4–5):493–505. https://doi.org/10.1002/(sici)1521-3978(199806)46:4/5⟨493::aid-prop493⟩3.0.co;2-p

Brassard G, Hoyer P (1997) An exact quantum polynomial-time algorithm for Simon's problem. In: Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems. IEEE, Washington DC, pp 12–23, https://doi.org/10.1109/istcs.1997.595153

Brassard G, Høyer P, Tapp A (1998) Quantum counting. In: Larsen KG, Skyum S, Winskel G (eds) Automata, Languages and Programming. Springer, Berlin, Heidelberg, pp 820–831, https://doi.org/10.1007/BFb0055105

Chancellor N (2023) Modernizing quantum annealing II: genetic algorithms with the inference primitive formalism. Nat Comput 22(4, Si):737–752. https://doi.org/10.1007/s11047-022-09905-2

Chen R, Dong C, Ye Y, et al (2019a) QSSA: Quantum evolutionary salp swarm algorithm for mechanical design. IEEE Access 7:145582–145595. https://doi.org/10.1109/access.2019.2945842

Chen W, Yang H, Hao Y (2019b) Scheduling of dynamic multi-objective flexible enterprise job-shop problem based on hybrid qpso. IEEE Access 7:127090–127097. https://doi.org/10.1109/access.2019.2938773

Clerc M (2024) What would a quantum PSO be? URL https://hal.science/hal-04472507v1

Coxson GE, Hill CR, Russo JC (2014) Adiabatic quantum computing for finding low-peak-sidelobe codes. In: 2014 IEEE High Performance Extreme Computing Conference (HPEC). IEEE, Waltham, IEEE High Performance Extreme Computing Conference, pp 1–6, https://doi.org/10.1109/HPEC.2014.7040953

Cuccaro SA, Draper TG, Kutin SA, et al (2004) A new quantum ripple-carry addition circuit. arXiv:quant-ph/0410184

Dirac PAM (1939) A new notation for quantum mechanics. In: Mathematical Proceedings of the Cambridge Philosophical Society, Cambridge University Press, p 416–418, https://doi.org/10.1017/s0305004100021162

Dong Y, Hu W, Zhang J, et al (2022) Quantum beetle swarm algorithm optimized extreme learning machine for intrusion detection. Quantum Inf Process 21(1):9. https://doi.org/10.1007/s11128-021-03311-w

Dorigo M, Gambardella L (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1(1):53–66. https://doi.org/10.1109/4235.585892

Duan H, Xing Z, Xu C (2009) An improved quantum evolutionary algorithm based on artificial bee colony optimization. In: Yu W, Sanchez E (eds) Advances In Computational Intelligence, Advances in Intelligent and Soft Computing, vol 61. Springer, Berlin, Heidelberg, pp 269–278, https://doi.org/10.1007/978-3-642-03156-4_27

Duan HB, Xu CF, Xing ZH (2010) A hybrid artificial bee colony optimization and quantum evolutionary algorithm for continuous optimization problems. Int J Neural Syst 20(1):39–50. https://doi.org/10.1142/s012906571000222x

Dür W, Vidal G, Cirac J (2000) Three qubits can be entangled in two inequivalent ways. Phys Rev A 62(6):062314. https://doi.org/10.1103/PhysRevA.62.062314

Durr C, Hoyer P (1999) A quantum algorithm for finding the minimum. arXiv:quant-ph/9607014

Einstein A, Podolsky B, Rosen N (1935) Can quantum-mechanical description of physical reality be considered complete? Phys Rev 47(10):777–780. https://doi.org/10.1103/PhysRev.47.777

El Gaily S, Imre S (2021) Constrained quantum optimization algorithm. In: 2021 20th International Symposium INFOTEH-JAHORINA (INFOTEH). IEEE, East Sarajevo, pp 1–6, https://doi.org/10.1109/infoteh51037.2021.9400679

Feng Y, Wu X, Lou S, et al (2024) Design optimization for pressurized water reactor using improved quantum fish swarm algorithm and intuitionistic linguistic decision-making. Adv Eng Inform 59:102315. https://doi.org/10.1016/j.aei.2023.102315

Garcia J, Maureira C (2021) A KNN quantum cuckoo search algorithm applied to the multidimensional knapsack problem. Appl Soft Comput 102:107077. https://doi.org/10.1016/j.asoc.2020.107077

Ge Y, Watson L, Collins E (1998) Genetic algorithms for optimization on a quantum computer. Unconventional Models Of Computation pp 218–227

Ghosh M, Dey N, Mitra D, et al (2022) A novel quantum algorithm for ant colony optimisation. IET Quantum Communication 3(1, Si):13–29. https://doi.org/10.1049/qtc2.12023

Giraldi GA, Portugal R, Thess RN (2004) Genetic algorithms and quantum computation. arXiv:cs/0403003

Google Quantum AI and collaborators (2025) Quantum error correction below the surface code threshold. Nature 638(8052):920–926. https://doi.org/10.1038/s41586-024-08449-y

Gossett P (1998) Quantum carry-save arithmetic. arXiv:quant-ph/9808061

Grover LK (1996) A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. ACM, New York, p 212–219, https://doi.org/10.1145/237814.237866

Grover LK (1998) Quantum computers can search rapidly by using almost any transformation. Phys Rev Lett 80:4329–4332. https://doi.org/10.1103/PhysRevLett.80.4329

Gu J, Gu X, Gu M (2009) A novel parallel quantum genetic algorithm for stochastic job shop scheduling. J Math Anal Appl 355(1):63–81. https://doi.org/10.1016/j.jmaa.2008.12.065

Haddaway NR, Page MJ, Pritchard CC, et al (2022) PRISMA2020: An R package and Shiny app for producing PRISMA 2020-compliant flow diagrams, with interactivity for optimised digital transparency and open synthesis. Campbell Syst Rev 18(2):e1230. https://doi.org/10.1002/cl2.1230

Haghighi MK, Fortin-Deschênes M, Pere C, et al (2025) EAQGA: A quantum-enhanced genetic algorithm with novel entanglement-aware crossovers. arXiv:2504.17923

Han K, Kim J (2000) Genetic quantum algorithm and its application to combinatorial optimization problem. In: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512), IEEE Congress on Evolutionary Computation, vol 2. IEEE, La Jolla, pp 1354–1360

Han P, Guo Y, Wang P, et al (2023) Optimal orbit design and mission scheduling for sun-synchronous orbit on-orbit refueling system. IEEE Trans Aerosp Electron Syst 59(5):4968 – 4983. https://doi.org/10.1109/taes.2023.3247552

Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press

Ibarrondo R, Gatti G, Sanz M (2022) Quantum vs classical genetic algorithms: A numerical comparison shows faster convergence. In: Ishibuchi H, Kwoh C, Tan A, et al (eds) 2022 IEEE Symposium Series On Computational Intelligence (SSCI). IEEE, Singapore, pp 947–954, https://doi.org/10.1109/ssci51031.2022.10022159

Ibarrondo R, Gatti G, Sanz M (2024) Quantum genetic algorithm with individuals in multiple registers. IEEE Trans Evol Comput 28(3):788–797. https://doi.org/10.1109/tevc.2023.3296780

Imre S (2007) Quantum existence testing and its application for finding extreme values in unsorted databases. IEEE Transactions on Computers 56(5):706–710. https://doi.org/10.1109/tc.2007.1032

Jannu S, Dara S, Thuppari C, et al (2024) Energy efficient quantum-informed ant colony optimization algorithms for industrial internet of things. IEEE Trans Artif Intell 5(3):1077 – 1086. https://doi.org/10.1109/tai.2022.3220186

Javadi-Abhari A, Treinish M, Krsulich K, et al (2024) Quantum computing with Qiskit. arXiv:2405.08810

Jian G, Li-juan S, Ru-chuan W, et al (2009) An improved quantum genetic algorithm. In: Huang T, Li L, Zhao M (eds) Third International Conference On Genetic And Evolutionary Computing. IEEE, Guilin, pp 14–18, https://doi.org/10.1109/wgec.2009.41

Jmal S, Haddar B, Chabchoub H (2019) Apply the quantum particle swarm optimization for the k-traveling repairman problem. Soft Computing 23(23):12547–12560. https://doi.org/10.1007/s00500-019-03805-x

Karafyllidis I (2015) Quantum Computing. Kallipos, Open Academic Editions, undergraduate textbook

Katoch S, Chauhan SS, Kumar V (2021) A review on genetic algorithm: past, present, and future. Multimed Tools Appl 80(5):8091–8126. https://doi.org/10.1007/s11042-020-10139-6

Kaur B, Rathi S, Agrawal RK (2022) Enhanced depression detection from speech using quantum whale optimization algorithm for feature selection. Comput Biol Med 150:106122. https://doi.org/10.1016/j.compbiomed.2022.106122

Kay A (2023) Tutorial on the quantikz package. arXiv:1809.03842

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, pp 1942–1948 vol.4, https://doi.org/10.1109/icnn.1995.488968

Khan AT, Cao X, Li S, et al (2021) Quantum beetle antennae search: a novel technique for the constrained portfolio optimization problem. Sci China Inf Sci 64(5):152204. https://doi.org/10.1007/s11432-020-2894-9

Khorsand A, Akbarzadeh-T M (2005) Quantum gate optimization in a meta-level genetic quantum algorithm. In: International Conference On Systems, Man And Cybernetics, IEEE International Conference on Systems Man and Cybernetics Conference Proceedings, vol 4. IEEE, Waikoloa, pp 3055–3062

Kim JS, Ahn CW (2022a) Quantum strategy of population initialization in genetic algorithm. In: Proceedings Of The 2022 Genetic And Evolutionary Computation Conference Companion, GECCO 2022. ACM, Boston, pp 439–442, https://doi.org/10.1145/3520304.3529010

Kim JS, Ahn CW (2022b) Size-efficient sparse population for strictly structured quantum genetic algorithm. Future Gener Comput Syst 135:159–171. https://doi.org/10.1016/j.future.2022.04.030

King J, Mohseni M, Bernoudy W, et al (2019) Quantum-assisted genetic algorithm. arXiv:1907.00707

Kitchenham B (2004) Procedures for performing systematic reviews. Keele, UK, Keele Univ 33:1–26

Kopp O, Snethlage CC, Schwentker C (2023) JabRef: BibTeX-based literature management software. TUGboat 44(138):441–447. https://doi.org/10.47397/tb/44-3/

tb138kopp-jabref

Kumar Pati A, Braunstein SL (2000) Impossibility of deleting an unknown quantum state. Nature 404(6774):164–165. https://doi.org/10.1038/35004532

Lahoz-Beltra R (2016) Quantum genetic algorithms for computer scientists. Computers 5(4):24. https://doi.org/10.3390/computers5040024

Li J, Qin H, Liu Y, et al (2022) A novel niche quantum ant colony-based clustering optimization approach for wireless auto meter reading network. J Sens 2022(1):1245705. https://doi.org/10.1155/2022/1245705

Li PC, Song KP, Shang FH (2011) Double chains quantum genetic algorithm with application to neuro-fuzzy controller design. Adv Eng Softw 42(10):875–886. https://doi.org/10.1016/j.advengsoft.2011.06.006

Li W, Yin Q, Zhang X (2010) Continuous quantum ant colony optimization and its application to optimization and analysis of induction motor structure. In: 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA). IEEE, Changsha, p 313 – 317, https://doi.org/10.1109/bicta.2010.5645311

Lucas A (2014) Ising formulations of many NP problems. Front Phys 2:5. https://doi.org/10.3389/fphy.2014.00005

Malossini A, Blanzieri E, Calarco T (2008) Quantum genetic optimization. IEEE Trans Evol Comput 12(2):231–241. https://doi.org/10.1109/tevc.2007.905006

Martinez GF, Niccolai A, Zich EL, et al (2024) A novel quantum binary evolutionary algorithm for EM applications: bGA-QCO. In: 2024 IEEE International Symposium on Antennas and Propagation and INC/USNC-URSI Radio Science Meeting (AP-S/INC-USNC-URSI). IEEE, Firenze, pp 35–36, https://doi.org/10.1109/ap-s/inc-usnc-ursi52054.2024.10686199

Martinez GFE, Zich RE (2024) Novel quantum computation based selection operator for genetic algorithms applied to electromagnetic problems. In: 2024 18th European Conference On Antennas And Propagation, EuCAP. IEEE, Glasgow, pp 1–4, https://doi.org/10.23919/EuCAP60739.2024.10501371

Martinez E. GF, Niccolai A, Zich EL, et al (2024) Accelerated quantum selection optimization of a microwave filter. In: 2024 IEEE 1st Latin American Conference on Antennas and Propagation (LACAP). IEEE, Cartagena de Indias, pp 1–2, https://doi.org/10.1109/lacap63752.2024.10876263

Martinez E GF, Zich EL, Niccolai A, et al (2024) An improved quantum crossover operator for binary evolutionary optimization of thinned array antennas. In: 2024 Asian Conference on Communication and Networks (ASIANComNet). IEEE, Bangkok,

pp 1–5, https://doi.org/10.1109/ASIANComNet63184.2024.10811050

Martinez E GF, Niccolai A, Zich RE (2025) Modified oracles for quantum selection applied to electromagnetic problems. In: 2025 19th European Conference on Antennas and Propagation (EuCAP). IEEE, Stockholm, pp 1–4, https://doi.org/10.23919/EuCAP63536.2025.11000033

Miao H, Wang H, Deng Z (2009) Quantum genetic algorithm and its application in power system reactive power optimization. In: 2009 International Conference on Computational Intelligence and Security, vol 1. IEEE, Beijing, p 107 – 111, https://doi.org/10.1109/cis.2009.133

Microsoft Azure Quantum, Aghaee M, Alcaraz Ramirez A, et al (2025) Interferometric single-shot parity measurement in InAs–Al hybrid devices. Nature 638(8051):651–655. https://doi.org/10.1038/s41586-024-08445-2

Mitchell M (1998) An introduction to genetic algorithms. MIT press

Mohammed AM, Elhefhawy N, El-Sherbiny MM, et al (2012) Quantum crossover based quantum genetic algorithm for solving non-linear programming. In: 2012 8th International Conference on Informatics and Systems (INFOS). IEEE, Giza, pp BIO–145–BIO–153

Montiel Ross OH (2020) A review of quantum-inspired metaheuristics: Going from classical computers to real quantum computers. IEEE Access 8:814–838. https://doi.org/10.1109/access.2019.2962155

Möttönen M, Vartiainen JJ, Bergholm V, et al (2004) Quantum circuits for general multiqubit gates. Phys Rev Lett 93:130502. https://doi.org/10.1103/PhysRevLett.93.130502

Mousavi S, Afghah F, Ashdown JD, et al (2018) Leader-follower based coalition formation in large-scale uav networks, a quantum evolutionary approach. In: IEEE INFOCOM 2018 - IEEE Conference On Computer Communications Workshops (Infocom Wkshps). IEEE, Honolulu, pp 882–887, https://doi.org/10.1109/infcomw.2018.8406915

Narayanan A, Moore M (1996) Quantum-inspired genetic algorithms. In: Proceedings of IEEE International Conference on Evolutionary Computation. IEEE, Nagoya, pp 61–66, https://doi.org/10.1109/icec.1996.542334

Nielsen MA, Chuang IL (2010) Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press

Ning T, Wang J, Han Y (2021) Logistics distribution de-carbonization pathways and effect in china: a systematic analysis using vrpsdp model. International Journal Of Low-carbon Technologies 16(4):1404–1411. https://doi.org/10.1093/ijlct/ctab063

Ojha VK, Abraham A, Snášel V (2014) Aco for continuous function optimization: A performance analysis. In: 2014 14th International Conference on Intelligent Systems Design and Applications, pp 145–150, https://doi.org/10.1109/isda.2014.7066253

Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization: An overview. Swarm Intell 1(1):33–57

Rosales SS, Montiel O, Orozco-Rosas U, et al (2024) Comparison of performance of amazon braket using a quantum genetic algorithm. Comput Sist 28(3):1429 – 1448. https://doi.org/10.13053/CyS-28-3-5178

Rylander B, Soule T, Foster J, et al (2001) Quantum evolutionary programming. In: GECCO'01: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. Morgan Kaufmann Publishers Inc., San Francisco, pp 1005–1011, https://doi.org/10.5555/2955239.2955421

Sabaawi AMA, Almasaoodi MR, El Gaily S, et al (2023) Quantum genetic algorithm for highly constrained optimization problems. Infocommun J 15(3):63–71. https://doi.org/10.36244/1cj.2023.3.7

Saini R, Mani A, Prasad MS, et al (2025) Evolutionary algorithm for the traveling salesman problem with innovative encoding on hybrid quantum-classical machines. IEEE Access 13:54223–54239. https://doi.org/10.1109/access.2025.3554690

SaiToh A, Rahimi R, Nakahara M (2013) A quantum genetic algorithm with quantum crossover and mutation operations. Quantum Inf Process 13(3):737–755. https://doi.org/10.1007/s11128-013-0686-6

Sarkar M, Pradhan J, Singh AK, et al (2023) A novel fast path planning approach for mobile devices using hybrid quantum ant colony optimization algorithm. arXiv:2310.17808

Sarkar M, Pradhan J, Singh AK, et al (2024) A novel hybrid quantum architecture for path planning in quantum-enabled autonomous mobile robots. IEEE Transactions On Consumer Electronics 70(3):5597–5606. https://doi.org/10.1109/tce.2024.3423416

Shor PW (1994) Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science. IEEE, Santa Fe, pp 124–134, https://doi.org/10.1109/sfcs.1994.365700

Shor PW (1997) Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing 26(5):1484–1509. https://doi.org/10.1137/s0097539795293172, https://doi.org/10.1137/S0097539795293172

Siddi Moreau G, Pisani L, Profir M, et al (2025) Quantum artificial intelligence scalability in the NISQ era: Pathways to quantum utility. Advanced Quantum Technologies 8(10):2400716. https://doi.org/10.1002/qute.202400716

Sun Y (2022) Parallel improved quantum evolutionary algorithm for complex optimization problems. In: Qiu M, Gai K, Qiu H (eds) Smart Computing and Communication. Springer, Cham, pp 254–264, https://doi.org/10.1007/978-3-030-97774-0_23

Sun Y, Ding M (2010) Quantum genetic algorithm for mobile robot path planning. In: 2010 Fourth International Conference on Genetic and Evolutionary Computing. IEEE, Shenzhen, pp 206–209, https://doi.org/10.1109/icgec.2010.58

Tightiz L, Yoo J (2024) Quantum-fuzzy expert timeframe predictor for post-tavr monitoring. Mathematics 12(17):2625. https://doi.org/10.3390/math12172625

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. ACM, New York, pp 71–82, https://doi.org/10.1145/1128022.1128034

Ur Rehman J, Ulum MS, Shaffar AW, et al (2025) Evolutionary algorithms and quantum computing: Recent advances, opportunities, and challenges. IEEE Access 13:16649–16670. https://doi.org/10.1109/access.2025.3530952

Vedral V, Barenco A, Ekert A (1996) Quantum networks for elementary arithmetic operations. Phys Rev A 54:147–153. https://doi.org/10.1103/PhysRevA.54.147

Vlachogiannis JG, Ostergaard J (2009) Reactive power and voltage control based on general quantum genetic algorithms. Expert Systems With Applications 36(3, 2):6118–6126. https://doi.org/10.1016/j.eswa.2008.07.070

Bužek V, Hillery M (1996) Quantum copying: Beyond the no-cloning theorem. Phys Rev A 54:1844–1852. https://doi.org/10.1103/PhysRevA.54.1844

Wang F, Xie K, Han L, et al (2023) Research on support vector machine optimization based on improved quantum genetic algorithm. Quantum Inf Process 22(10):380. https://doi.org/10.1007/s11128-023-04139-2

Wang J, Zhou R (2010) A novel quantum genetic algorithm for PID controller. In: Huang DS, Zhao Z, Bevilacqua V, et al (eds) Advanced Intelligent Computing Theories and Applications. Springer, Berlin, Heidelberg, p 72–77, https://doi.org/10.1007/978-3-642-14922-1_10

Wang J, Zhong D, Adeli H, et al (2018) Smart bacteria-foraging algorithm-based customized kernel support vector regression and enhanced probabilistic neural network for compaction quality assessment and control of earth-rock dam. Expert Systems

35(6):e12357. https://doi.org/10.1111/exsy.12357

Wang L, Tang F, Wu H (2005) Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation. Applied Mathematics And Computation 171(2):1141–1156. https://doi.org/10.1016/j.amc.2005.01.115

Wang T, Wang SD, Zhang WM, et al (2011) Pattern synthesis of array antennas with a kind of quantum particle swarm optimization algorithm. Advances in Intelligent and Soft Computing 129:123 – 130. https://doi.org/10.1007/978-3-642-25986-9_19

Wei P, Yu W (2023) Improved quantum artificial bee colony algorithm-optimized artificial intelligence models for suspended sediment load predicting. IEEE Access p 1. https://doi.org/10.1109/access.2023.3333051

Wei X, Fujimura S (2012) Multiupdate mode quantum evolutionary algorithm and its applications to combination and permutation problems. Ieej Transactions On Electrical And Electronic Engineering 7(2):166–173. https://doi.org/10.1002/tee.21712

Werner RF (1998) Optimal cloning of pure states. Phys Rev A 58:1827–1832. https://doi.org/10.1103/PhysRevA.58.1827

West P, contributors (2025) Bibtex-Tidy: Cleaner and formatter for bibtex files. URL https://github.com/FlamingTempura/bibtex-tidy, Version 1.14.0. Accessed 29 July 2025

Wootters WK, Zurek WH (1982) A single quantum cannot be cloned. Nature 299:802–803. https://doi.org/10.1038/299802a0

Xia G, Han Z, Zhao B, et al (2020) Local path planning for unmanned surface vehicle collision avoidance based on modified quantum particle swarm optimization. Complexity 2020(1):2095426. https://doi.org/10.1155/2020/3095426

Xing Z, Duan H, Xu C (2009) An improved quantum evolutionary algorithm with 2-crossovers. In: Yu W, He H, Zhang N (eds) Advances In Neural Networks - ISNN 2009, vol 5551. Springer, Berlin, Heidelberg, pp 735–744, https://doi.org/10.1007/978-3-642-01507-6_83

Xu L, Wang X, Wang Z, et al (2025) Hybrid quantum genetic algorithm for structural damage identification. Computer Methods in Applied Mechanics and Engineering 438:117866. https://doi.org/10.1016/j.cma.2025.117866

Yin Q, Li W, Zhang X, et al (2010) Continuous quantum particle swarm optimization and its application to optimization calculation and analysis of energy-saving motor used in beam pumping unit. In: 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA). IEEE, Changsha, pp 1231—-1235, https://doi.org/10.1109/bicta.2010.5645153

Yoshio Josué Rubio OHMRCynthia Olvera (2021) Quantum-inspired evolutionary algorithms on IBM quantum experience. Engineering Letters 29(4):1573

Yumin D, Wu M, Zhang J (2020) Recognition of pneumonia image based on improved quantum neural network. IEEE Access 8:224500–224512. https://doi.org/10.1109/access.2020.3044697

Zhang Y, Zhao J, Jia Y, et al (2025) An improved adaptive quantum genetic algorithm as classical optimizer for the quantum approximate optimization algorithm on maxcut problem. Quantum Inf Process 24(7):222. https://doi.org/10.1007/s11128-025-04776-9

Zhu Y, Zhang M, Huang Q, et al (2025) Secretary bird optimization algorithm based on quantum computing and multiple strategies improvement for kelm diabetes classification. Sci Rep 15(1):3774. https://doi.org/10.1038/s41598-025-87285-0