

# Implementación Cuántica de un Algoritmo Genético

## *Quantum Implementation of a Genetic Algorithm*

## *Implementação quântica de um algoritmo genético*

Mauricio Solar <sup>1,\*</sup>, Vicente Figueroa <sup>2</sup>, Francisco Manriquez <sup>3</sup>,  
Francisco Pizarro <sup>4</sup>, Liubov Dombrovskaia <sup>5</sup>

Recibido: 10/10/2024

Aceptado: 10/10/2024

**Resumen.** - Este trabajo proporciona una visión generalizada del estado actual de los algoritmos genéticos cuánticos (QGAs), mostrando los avances realizados en esta área de investigación los últimos 24 años. Los QGAs combinan conceptos de la computación cuántica y los algoritmos genéticos clásicos (CGAs), lo que les permite abordar problemas complejos de optimización y búsqueda de manera eficiente. Se presentan los principales hallazgos y contribuciones de estos algoritmos cuánticos destacando las tendencias y los enfoques más prometedores, así como los desafíos y limitaciones que deben superarse. Se presentan nuevos enfoques y técnicas de implementación de QGAs, incluyendo operadores genéticos cuánticos y esquemas de codificación eficientes que contribuyen a mejorar el rendimiento y la convergencia de los algoritmos. Se comparan los QGAs y otros enfoques similares, como los CGAs y los algoritmos cuánticos puros, destacando las ventajas y desventajas relativas de los QGAs en comparación a sus versiones clásicas. Se muestra también una implementación de QGA utilizando la biblioteca Qiskit. Se presentan la selección de los métodos usados para la generación de la población inicial, el cruzamiento y la mutación de las distintas poblaciones de los circuitos cuánticos simulados en los experimentos realizados, ejemplificando las ventajas significativas que estos pueden traer en comparación con los enfoques clásicos.

**Palabras clave:** Algoritmo Genético Cuántico; Computación Cuántica; Algoritmo Genético Cuántico Híbrido.

---

(\*) Autor de correspondencia

<sup>1</sup> Académico, Universidad Técnica Federico Santa María, mauricio.solar@usm.cl, ORCID iD: <https://orcid.org/0000-0002-4433-4622>

<sup>2</sup> Estudiante postgrado, Universidad Técnica Federico Santa María, vicente.figueroa@usm.cl,  
ORCID iD: <https://orcid.org/0009-0001-2391-8223>

<sup>3</sup> Estudiante postgrado, Universidad Técnica Federico Santa María, francisco.manriquezn@usm.cl,  
ORCID iD: <https://orcid.org/0009-0008-7045-4258>

<sup>4</sup> Estudiante postgrado, Universidad Técnica Federico Santa María, francisco.pizarroj@usm.cl,  
ORCID iD: <https://orcid.org/0009-0004-6317-6414>

<sup>5</sup> Académica, Universidad Técnica Federico Santa María, liuba@inf.utfsm.cl, ORCID iD: <https://orcid.org/0000-0001-6572-9765>

Memoria Investigaciones en Ingeniería, núm. 27 (2024). pp. 220-237  
<https://doi.org/10.36561/ING.27.14>

ISSN 2301-1092 • ISSN (en línea) 2301-1106 – Universidad de Montevideo, Uruguay

Este es un artículo de acceso abierto distribuido bajo los términos de una licencia de uso y distribución CC BY-NC 4.0. Para ver una copia de esta licencia visite <http://creativecommons.org/licenses/by-nc/4.0/>

**Summary.** - This work provides a generalized view of the current state of quantum genetic algorithms (QGAs), showing the advances made in this research field over the last 24 years. QGAs combine concepts from quantum computing and classical genetic algorithms (CGAs), allowing them to address complex search and optimization problems efficiently. The main findings and contributions of these quantum algorithms are presented, highlighting the most promising trends and approaches, as well as the challenges and limitations that need to be overcome. New approaches and implementation techniques for QGAs are presented, including quantum genetic operators and efficient coding schemes that contribute to improving the performance and convergence of the algorithms. QGAs and other similar approaches, such as CGAs and pure quantum algorithms, are compared, highlighting the relative advantages and disadvantages of QGAs compared to their classical versions. An implementation of QGA using the Qiskit library is also shown. The selection of the methods used for the generation of the initial population, the crossing and the mutation of the different populations of the quantum circuits simulated in the experiments carried out are presented, exemplifying the significant advantages that these can bring in comparison with classical approaches.

**Keywords:** Quantum Genetic Algorithm; Quantum Computing; Hybrid Quantum Genetic Algorithm.

**Resumo.** - Este trabalho fornece uma visão generalizada do estado atual dos algoritmos genéticos quânticos (QGAs), mostrando os avanços feitos neste campo de pesquisa nos últimos 24 anos. Os QGAs combinam conceitos da computação quântica e algoritmos genéticos clássicos (CGAs), permitindo que eles abordem problemas complexos de busca e otimização de forma eficiente. As principais descobertas e contribuições desses algoritmos quânticos são apresentadas, destacando as tendências e abordagens mais promissoras, bem como os desafios e limitações que precisam ser superados. Novas abordagens e técnicas de implementação para QGAs são apresentadas, incluindo operadores genéticos quânticos e esquemas de codificação eficientes que contribuem para melhorar o desempenho e a convergência dos algoritmos. QGAs e outras abordagens semelhantes, como CGAs e algoritmos quânticos puros, são comparados, destacando as vantagens e desvantagens relativas dos QGAs em comparação com suas versões clássicas. Uma implementação de QGA usando a biblioteca Qiskit também é mostrada. São apresentadas a seleção dos métodos utilizados para a geração da população inicial, o cruzamento e a mutação das diferentes populações dos circuitos quânticos simulados nos experimentos realizados, exemplificando as vantagens significativas que estes podem trazer em comparação com abordagens clássicas.

**Palavras-chave:** Algoritmo Genético Quântico; Computação Quântica; Algoritmo Genético Quântico Híbrido.

**1. Introducción.** - La computación cuántica y la programación genética son dos áreas de investigación fascinantes que han mostrado un gran potencial para transformar la forma en que se abordan los desafíos computacionales complejos. En (1) se exploran las aplicaciones de la programación genética en el campo de la computación cuántica desde la cual ha habido un crecimiento significativo en la investigación y los avances tecnológicos.

En este artículo se explora la resolución de problemas de búsqueda (2), la optimización cuántica, como también los nuevos enfoques que han mejorado la capacidad de encontrar soluciones óptimas para problemas complejos en ámbitos como la optimización de parámetros cuánticos y la exploración de espacios de soluciones cuánticas (3), (4), (5), (6), (7), (8), (9). En estos trabajos se explora cómo la programación genética y los algoritmos genéticos (GA) han sido utilizados en la optimización de sistemas cuánticos, como la calibración de qubits y la reducción de errores en sistemas de computación cuántica, que son fundamentales para superar los desafíos actuales en la implementación práctica de la computación cuántica.

En este artículo se muestra los avances en el campo de la programación genética aplicada a la computación cuántica. Los avances en la optimización de circuitos, diseño de algoritmos, resolución de problemas, simulación, seguridad y optimización de sistemas cuánticos, han ampliado el espectro de aplicaciones prácticas y están acelerando el desarrollo de la computación cuántica. Entre las aplicaciones se encuentran imágenes médicas (10), visualización de datos (11), comunicaciones (14; 12; 13), arquitectura (15; 16), energía eólica (17), gestión de energía (18), control (19), diagnóstico de fallas (20), uso de criptografía cuántica en imágenes (21). La evolución de programas cuánticos ha llevado al desarrollo de algoritmos de criptografía más robustos y resistentes, impulsando la seguridad en el ámbito cuántico. En la sección 3 se presenta el estado del arte en esta materia.

A partir de los pseudo-códigos y diagramas presentados en (6), (8) y (23), se extrapola una versión con el funcionamiento de estos en las librerías de Qiskit, las cuales se implementan. En la sección 4 se muestra esta implementación usando Qiskit de un Algoritmo Genético Cuántico (QGA) basado en un Algoritmo Genético Cuántico Híbrido (HQGA) presentado en (8) y se presentan los métodos que permiten simular un entorno cuántico para el desarrollo de los mismos circuitos utilizando Python.

**2. Definición de un Algoritmo Genético: Población, Cruzamiento, Mutación.** - Una de las ramas que más prometen en la computación cuántica es la aplicación de algoritmos genéticos a estos mismos. Los GA son técnicas de optimización inspiradas en los principios de la evolución biológica, que han demostrado su eficacia en la resolución de problemas complejos en una amplia gama de áreas, desde la planificación de rutas hasta el diseño de estructuras.

La combinación de GA con computación cuántica, a través de herramientas como Qiskit, permite aprovechar las ventajas de ambas disciplinas. Los QGA tienen el potencial de superar las limitaciones de los CGA, ya que pueden explorar simultáneamente múltiples soluciones en paralelo gracias a los fenómenos cuánticos como la superposición y la interferencia cuántica, al menos para un buen número de casos específicos.

Los QGAs se basan en la codificación de qubits para el almacenamiento de la información. Para esto se puede ocupar cualquier representación conocida basada en bits, ocupando un qubit por cada bit, pero pueden surgir limitaciones evidentes. Si se quiere ocupar la representación de punto flotante de 32 bits, se agotan fácilmente los recursos disponibles, ya que actualmente no se logra soportar una gran cantidad de qubits en un computador cuántico. En un QGA, se codifica un cromosoma como  $m$  secuencias de  $k$  qubits, que representan  $m$  variables codificadas en  $k$  bits según la ecuación

1.

$$q_j^t = [\psi_{11}^t | \psi_{12}^t | \dots | \psi_{1k}^t | \psi_{21}^t | \psi_{22}^t | \dots | \psi_{mk}^t] \quad (1)$$

donde  $q_j^t$  representa el cromosoma  $j$  de la  $t$ -ésima generación. Para la codificación inicial, se codifican los  $(\alpha, \beta)$  de cada  $\psi$  como  $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$  (compuerta de Hadamard), de manera que, en un inicio, la probabilidad de cualquier estado del cromosoma sea el mismo.

**2.1. Rotación.** - En un algoritmo genético, es esencial actualizar los cromosomas de manera que se acerquen al valor esperado. En un QGA, las compuertas de rotación son las que se encargan de esta función. Se define la compuerta de rotación  $U(\theta_i)$ , aplicada a cada par  $(\alpha_i, \beta_i)$  según la ecuación 2.

$$U(\theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \quad (2)$$

Se puede notar que  $U(\theta_i)$  es equivalente a una rotación en el eje  $Y$  en la esfera de Bloch. El ángulo  $\theta_i$  es fijo y se define manualmente, según la estrategia de ajuste, pero normalmente toma valores entre  $0.1\pi$  y  $0.005\pi$ . El algoritmo de rotación depende de tablas de valores fijos, que determinan la dirección de la rotación a aplicar según el fitness del cromosoma actual en relación al mejor. Dependiendo de la tabla afecta a la rapidez de convergencia del algoritmo, por lo que se plantea el método de rotación auto-adaptada. Este último se basa en ir reduciendo el ángulo de rotación según el número de la actual iteración generacional, sin perder la dirección hacia el mejor cromosoma encontrado, según la ecuación 3.

$$\theta_i = \theta_{max} - \left( \frac{\theta_{max} - \theta_{min}}{it_{max}} \right) iter \quad (3)$$

donde  $\theta_{max}$  y  $\theta_{min}$  son valores fijos,  $it_{max}$  es la iteración maximal e  $iter$  es la generación actual. Se puede notar que los valores de  $\theta_i$  disminuyen según avanzan las iteraciones, lo que permite ir reduciendo la función de ajuste (Fitness).

**2.2. Mutación.** - La mutación permite que el algoritmo desarrolle nuevas posibilidades de valores, desviándose del resto de la población, lo que puede ayudar a evitar caer en óptimos locales. La compuerta de mutación invierte los valores de  $(\alpha, \beta)$  en  $(\beta, \alpha)$ , lo que equivale a una compuerta NOT (o bien, compuerta X).

**2.3. Cruzamiento.** - Teóricamente, no se puede hacer un equivalente del cruzamiento en su versión cuántica. Aun así, se puede ocupar en algoritmos basados en computación cuántica. Una forma de hacer cruzamiento es seleccionar una posición en 2 de los cromosomas e intercambiar las secciones posteriores a ese punto.

**2.4. Diagrama del QGA.** - De la literatura se selecciona la arquitectura híbrida HQGA de (8), para implementar elementos de la computación cuántica como elementos de la computación clásica. Específicamente, se orienta la arquitectura para la resolución del conocido problema de la mochila (*Knapsack Problem*), siendo los valores asociados a cada objeto generados aleatoriamente con una distribución uniforme entre 1 y 10. El diagrama de flujo de la Figura 1 describe el funcionamiento del mismo.

### 3. Estado del Arte. -

**Metodología.** - Para buscar trabajos relacionados con QGA en la literatura se usan distintas técnicas, siendo principalmente *Snow Ball* la más común. Se consideraron los trabajos que utilizan directamente como referencia el survey en (1), y una vez obtenidos los trabajos que lo citan se filtra la mayoría para reducir el número de trabajos que no se relacionan directamente con mejoras a los algoritmos genéticos cuánticos. Los trabajos de interés extraídos con este método son (24), (25), (22), (5), (1) y (26).

Para recabar más información se utiliza la búsqueda por palabras claves, tales como “*quantum genetic algorithm*” o “*quantum genetic programming*”, entre otros. Mediante esta técnica se encuentran trabajos donde se usa QGA que aportan a problemas en diversas áreas: (19), (16), (15), (7), (12), (2), (21), (27), (17), (28), (10), (4), (3), (29), (23), (6), (30), (31), (32), y (33).

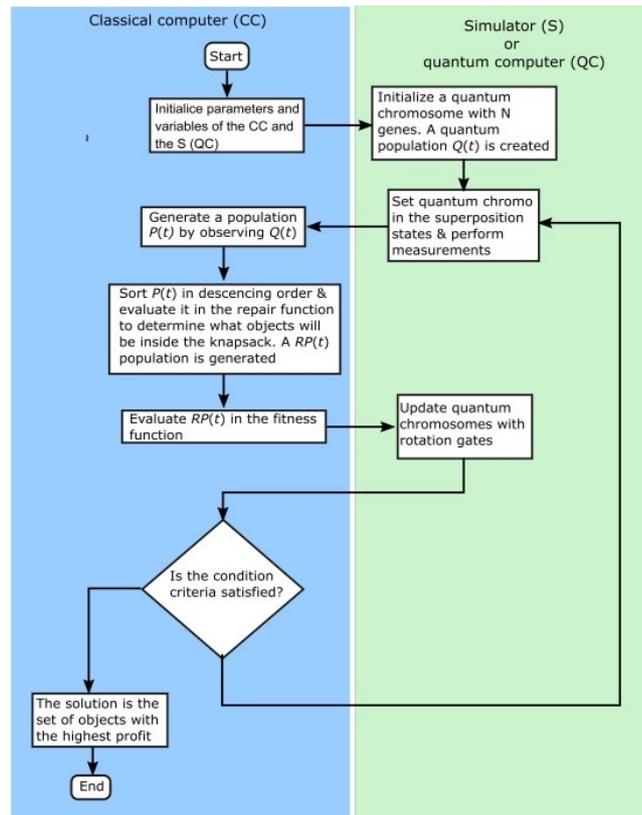


Figura 1. Diagrama de flujo de HQGA.

**3.1. Descripción de trabajos con QGA.** - En esta sección se presenta una breve y resumida descripción de la cobertura y las intenciones de los trabajos publicados, junto con las especificaciones del problema cuántico que aborda cada uno.

En el área de control se encuentra (19) que combina el modelo de nube con el QGA, aprovechando su eficiencia en cálculos complejos para aproximar los órdenes fraccionarios que optimizan el control de un vehículo autónomo submarino (AUV).

También en el área de control en (12) se utiliza una variante de QGA para resolver el problema de optimización, la cual utiliza el gradiente de la función para la rotación cuántica de los qubits, mejorando así su velocidad de convergencia.

En arquitectura se muestra en (16) como se optimizan las dimensiones y materiales de ventanas, techos y murallas de un edificio para disminuir su costo energético. Se utiliza la rotación cuántica como método de evolución genética de las soluciones, aprovechando la capacidad de almacenar información continua de los qubits. En el mismo ámbito, en (15) se aprovecha la capacidad del algoritmo cuántico para hacer cálculos multidimensionales de manera eficiente, para obtener un modelo óptimo al autocalibrar un sistema de medición por medio de imágenes.

En (17) se utiliza un QGA para minimizar costos en la producción de energía eólica. Se utilizan reglas simples sobre la rotación cuántica y una perturbación adicional para evitar que el algoritmo se estanque en un óptimo local. Luego, se compara contra otros algoritmos y se utiliza en un entorno real. En el mismo ámbito energético, en (18) se utiliza un QGA multi-objetivo mejorado para resolver los problemas de contaminación ambiental causados por los automóviles eléctricos en las autopistas de alta velocidad. Se propone una tasa de autoconsumo de energías limpias y estrategias de gestión de la eficiencia energética. Mejora la tasa de convergencia, aumenta la diversidad de la población, disminuye los costos de energía y las emisiones de carbono.

En (20) se propone un método de diagnóstico de fallas en maquinaria rotacional, basado en la máquina de soporte vectorial (SVM) optimizada con QGA. El método se aplica a la detección de fallos en maquinaria rotativa sobre un eje, obteniendo una mayor precisión que los tradicionales SVM y GA (i.e. CGA).

En búsqueda textual en (2) se propone una variante de los QGA (MTSQIGA), donde se calcula el ángulo para la compuerta de rotación comúnmente usada a partir de los resultados de la iteración, en caso de no cumplirse el criterio de longitud impuesto.

En (21) se utiliza un QGA para el proceso de encriptación y desencriptación de imágenes, a partir de la alteración al ángulo de rotación de la compuerta cuántica según reglas definidas en matrices de cada iteración.

También en el ámbito de las imágenes, en (10) se realiza una comparación entre CGA, QGA y PSO (*Particle Swarm Optimization*) para el procesamiento de imágenes médicas. Para QGA se utiliza un algoritmo de rotación simple. Se concluye que PSO obtiene mejores resultados y además converge más rápido a la solución. En segundo lugar, se encuentra QGA, y en último lugar queda CGA.

Una aplicación en comunicaciones se muestra en (13), donde se obtuvieron criterios, métricas y resultados experimentales, en los que se demuestra que el QGA es mejor que el CGA. Se proporciona una discusión detallada sobre la función objetivo, la convergencia al óptimo global y la necesidad de mejorar las soluciones obtenidas en una aplicación de comunicaciones. Similar al trabajo (13), en (14) se presenta un QGA que permite resolver un problema de datos ponderados en una cavidad con un sensor de 3 canales que detecta el ruido debido al impacto de partículas en suspensión. Los resultados se obtienen con una alta capacidad de procesamiento de datos, utilizando una métrica de varianza para la señal de entrada.

En los problemas de optimización aparecen una serie de trabajos.

En (3) se prueba y muestra la efectividad de usar QGA en problemas de optimización combinatoriales, específicamente el problema de la mochila, demostrando cómo este algoritmo logra un rendimiento superior a muchos otros GAs. (3) demuestra y simula el rendimiento de un QGA asistido. Se realiza la prueba haciendo uso de optimización binaria cuadrática sin restricciones (QUBO) durante 100 iteraciones y se compara con su contraparte clásica durante 2000 iteraciones, demostrando de esa manera que el algoritmo cuántico converge a la solución óptima mucho más rápido que el clásico.

(6) busca acelerar el proceso evolutivo y la probabilidad de encontrar una solución óptima en un QGA, determinando el ángulo de rotación, la capacidad adaptativa del mismo y agregando operaciones cuánticas de mutación, logrando así que la eficiencia del algoritmo propuesto sea mayor a un CGA tradicional para resolver problemas de optimización.

En (5) se realizan varias pruebas y se muestran resultados de aplicar algoritmos genéticos en diversos problemas de optimización en computación cuántica. Principalmente se estudia la desigualdad de Bell (Bell-CHSH) y la medida del entrelazamiento entre qubits.

En (7) se propone una implementación de QGA en problemas RCPSP (*Resource-Constrained Project-Scheduling Problem*). Para ello se utiliza cruzamiento cuántico (*quantum crossover*) y mutación cuántica, junto a operaciones de rotación. Además, se estudia su complejidad y se compara su velocidad y eficacia contra algoritmos clásicos encontrados en la literatura.

Ya en el ámbito mismo de los QGA se encuentra (27) que utiliza una mejora de un algoritmo GRS (*Greedy Random Sampling*) y un nuevo GA para solucionar el problema de compilación de circuitos cuánticos. Se concluye que GA es mejor que GRS para instancias de grandes proporciones.

En (28) se realiza una comparación sobre la versión clásica de un algoritmo genético (CGA) y su versión cuántica (QGA), utilizando el problema de la mochila para resultados experimentales. Los resultados muestran que QGA llega a mejores resultados cuando el tamaño del problema es mucho mayor. Además, se plantea una complejidad  $O(N)$  para QGA y  $O(N^2)$  para CGA.

En (25) se propone una versión mejorada del *Quantum Genetic Optimization Algorithm* (QGOA) con el objetivo de acelerar los distintos procesos de un GA. Esta propuesta requiere un número menor de operaciones para identificar la función de ajuste para una sub-población en cada generación, reemplazando los procesos tradicionales por un único proceso cuántico, reduciendo la complejidad a  $O(1)$  en términos de llamadas en el proceso de selección.

Está demostrado que las puertas “adicionadoras” no están permitidas en la computación cuántica, al menos no tal y como se conocen en su versión clásica. En (25) se analizan propuestas tecnológicas para aproximar de forma cuántica las puertas mencionadas, y optimizar las mismas mediante el uso de GAs, llegando a nuevos niveles en términos de eficiencia y confiabilidad.

En (34) se muestra un algoritmo genético celular multi-objetivo con optimización multi-objetivo y un QGA para crear autómatas celulares. Se proporciona un modelado matemático y computacional discreto para el proceso con dos estados posibles. Se diferencia de otros algoritmos meta-heurísticos en que mejora el rendimiento y la próxima generación depende de la célula y su entorno.

En (35) se presenta un GA para la preparación de estados (GASP por sus siglas en inglés) que genera circuitos cuánticos de profundidad relativamente baja para inicializar un computador cuántico en un estado cuántico específico. El método utiliza un conjunto básico de puertas  $R_x, R_y, R_z$  y CNOT y un GA para generar sistemáticamente circuitos para sintetizar el estado objetivo con la fidelidad requerida. GASP puede producir circuitos más eficientes de una precisión determinada con menor profundidad y menor número de puertas que otros métodos.

En (1) se describe cómo simular un computador cuántico para determinar la idoneidad de un algoritmo cuántico en hardware clásico. Se describen tres enfoques diferentes de programación genética para impulsar el simulador para desarrollar nuevos algoritmos cuánticos. Los enfoques son la programación genética estándar basada en árboles, la programación genética del genoma lineal basada en pilas y la programación genética del genoma lineal sin pilas. Se prueban las técnicas en cuatro problemas diferentes: el problema de promesa temprana de 2 bits, el problema de escalamiento mayoritario, el problema de búsqueda en BDs de 4 elementos y el problema and/or de 2 bits. Para tres de estos problemas, los algoritmos descubiertos automáticamente son más eficientes que los algoritmos clásicos para los mismos problemas.

En (26) se propone una nueva representación de un conjunto de operadores genéticos con el objetivo de optimizar distintos circuitos cuánticos mediante programación y GAs, heredando la representación y propiedades de ambos. Considerando que los circuitos cuánticos son una colección de puertas, cada un operando en varias categorías de qubits, con sus respectivos parámetros, es que la representación propuesta puede buscar estos mismos de forma satisfactoria. En particular, se aplican estos circuitos en el problema de la producción del entrelazamiento entre qubits.

En el contexto de la computación cuántica evolutiva, en (29) se propone la implementación cuántica de uno de los procesos en los algoritmos evolutivos: el entrecruzamiento de los cromosomas, dando a entender que este no ha sido implementado de esta forma antes, sino que únicamente ha sido usado clásicamente. Además, propone que el entrecruzamiento se realice de forma paralela entre generaciones, logrando en resumen una aceleración del orden cuadrático por sobre su contraparte clásica.

En (23) se presenta una revisión de distintos GAs implementados en computación cuántica, orientada específicamente a carreras relacionadas con ciencias de la computación, obviando todos los aspectos físicos relacionados con la computación cuántica, con la intención de difundir estos mismos. Se analizan las ventajas y desventajas de cada uno de estos en comparación a sus hermanos clásicos.

En (22) se abarca la teoría y práctica de las bases de los algoritmos evolutivos cuánticos. En un principio se da una introducción a la cuántica (computacional y física), para más adelante hablar sobre los algoritmos evolutivos cuánticos, analizando los principales problemas que se tienen con la implementación de estos en diversos problemas, además de evaluar y mostrar el desempeño de alguno de estos algoritmos.

En (29) se presenta una aproximación simple a los QGAs, analizando los beneficios y las desventajas de estos al aplicar la paralelización cuántica a los CGA.

En (24) pretende lograr una optimización en los algoritmos de comunicación de sistemas cuánticos distribuidos, compuestos por unos cuantos circuitos cuánticos de baja capacidad. Comparando los resultados obtenidos de este algoritmo con otro algoritmo propuesto en un trabajo anterior, esta optimización logra una mejora notable en velocidad.

En (32) se hace uso de *reverse quantum annealing*, una variante de algoritmo de evolución cuántica, que hace uso de búsquedas semi-locales o semi-no locales desde un estado inicial. Este algoritmo permite que los GAs usen fluctuaciones cuánticas para las mutaciones en la evolución de los cromosomas, llamado *Quantum-Assisted Genetic Algorithms* (QAGAs). Este algoritmo es capaz de encontrar buenas soluciones del óptimo global a diferencia del *standard quantum annealing*.

**3.2. Análisis Comparativo.** - La Tabla I muestra las ventajas (columna “Ventaja Comparativa”) de la contribución realizada (segunda columna) por el trabajo de la referencia en la columna “Ref” en el ámbito indicado en la cuarta columna de la tabla.

Los artículos presentados abarcan distintos ámbitos tecnológicos, desde la medicina hasta la arquitectura. Por lo tanto, se considerará para la comparación el aporte o utilización que se le da al algoritmo cuántico y sus resultados comparativos. Los artículos (16) y (15) no se incluyen en la Tabla I porque ambos presentan un algoritmo simple para la convergencia.

La Tabla II muestra las ventajas (columna “Ventaja Comparativa”) de la contribución realizada (segunda columna) por el trabajo de la referencia en la columna “Ref” en el ámbito de los problemas de optimización.

**3.3. Discusión bibliográfica.** - En general, se espera que los QGA tengan un rendimiento mayor a los algoritmos cuánticos y a los CGA, debido a su capacidad de encontrar óptimos locales en pocas iteraciones. Esto al menos, para la mayoría de los problemas que pueden ser tratados cuánticamente.

De la literatura analizada acerca del problema y su contexto se ha logrado profundizar en la extensa aplicación que puede tener la combinación de algoritmos clásicos y cuánticos, permitiendo sacar lo mejor de cada uno para así, no solo sobrepasar a la parte clásica, sino que también a la parte cuántica, aunque esto no se cumple para una gran cantidad de casos de interés.

Muchos de los trabajos de la literatura analizados se centran en estudiar la aplicación de QGA en distintas áreas más que en la investigación de como optimizar el método para que escape de los óptimos locales o tenga una convergencia más rápida.

Por otra parte, existen 3 trabajos que buscan hacer cuántica la parte “clásica” del QGA, implementando en el mismo mutaciones y cruzamientos cuánticos.

Se debe destacar el trabajo de (1) que presenta hasta ese momento los CGA que se implementaron sobre circuitos cuánticos, conociéndose éstos como QGA.

Como parte de los resultados más esperanzadores en esta área se tienen (6), (29) y (32).

Ref	Área	Contribución realizada	Ventaja Comparativa
(19)	Control	Extrapolación de los GA aprovechando su paralelismo y los aplica en nuevo ámbito	Logra salir medianamente del problema de los óptimos locales de los CGA
(12)	Control	Aplica reglas del fitness para ajustar ángulo de convergencia en la rotación cuántica	Realiza análisis numérico y de sensibilidad. Extiende solución a distintos contextos
(2)	Búsqueda	Propone variante de QGA con parámetros del contexto para converger eficientemente	Realiza extenso desarrollo del funcionamiento del algoritmo. Se compara con otros en el estado del arte
(21)	Encriptación	Utiliza el algoritmo como proceso de encriptado	Uso innovador del algoritmo. Realiza análisis de seguridad
(17)	Energía	Usa perturbaciones para no caer en óptimo local	Explica paso a paso (simple) como funciona un QGA
(10)	Procesamiento de imagen	Compara los algoritmos respecto a diversas funciones de entropía	Realiza análisis cualitativo y cuantitativo de los resultados

Tabla I. Contribuciones en diferentes ámbitos.

El año 2013, en (6) se sentaron las primeras bases para hacer cuántico el CGA, naciendo la duda de si se puede llevar a cuántico completamente el CGA, esperando tener en un futuro aplicar un algoritmo cuántico sobre circuitos cuánticos y no uno clásico sobre uno cuántico, tal y como se venía aplicando hace años.

El año 2014 en (29) se realizó la primera implementación cuántica al entrecruzamiento de un GA, siendo esto de total interés, dado que abre paso a un nuevo campo de estudio, permitiendo conocer cómo se comporta un “híbrido” de algoritmo clásico con cuántico.

En 2019, se publica una versión cuántica de las mutaciones clásicas de un GA (32).

**4. Implementación de un Algoritmo Genético Cuántico (QGA).** - El objetivo de esta sección es mostrar la implementación de un QGA utilizando Qiskit, en distintos circuitos cuánticos diseñados específicamente para esta tarea.

```

1 #!pip install qiskit pylatexenc
2 #!pip install more_itertools
3 import math
4 import numpy as np
5 from qiskit import QuantumRegister, QuantumCircuit, Aer, execute
6 from qiskit.quantum_info import Statevector
7 from qiskit.visualization import plot_histogram
8 import pylatexenc
9 import matplotlib.pyplot as plt
10 from more_itertools import locate

```

Se presenta a continuación la metodología usada paso a paso para diseñar y codificar los distintos elementos que componen un GA, tales como la inicialización de la población y los distintos métodos aplicados para el entrecruzamiento y la mutación de los genes en cada población.

Además, se evalúan los experimentos realizados en los circuitos cuánticos diseñados, demostrando como los QGA logran una optimización eficiente de estos mismos.

Ref	Contribución realizada	Ventaja Comparativa
(7)	Utiliza diversos métodos cuánticos para mejorar la convergencia y evitar el óptimo local	Realiza extenso análisis sobre parámetros utilizados. Se compara contra otros 23 algoritmos de la literatura
(27)	Propone nuevo GA para la compilación de hardware cuántico	Profundiza problema y solución. Buen apoyo a investigaciones futuras
(28)	Compara QGA con GA en el problema de la mochila	Analiza resultados respecto a diversos valores en los parámetros del problema
(26)	Presenta nuevo esquema de operadores genéticos para producción de algoritmos cuánticos	Se enfoca especialmente en la producción de entrelazamiento cuántico
(3)	Demuestra la efectividad de QGA para problemas de optimización combinatorios	Compara resultados de QGA con CGA en el problema de la mochila
(29)	Propone nuevo quantum crossover que explota paralelismo cuántico	Análisis teórico cuántico del procedimiento y su contraparte clásica
(23)	Plantea las bases de un real QGA para futuras investigaciones	Explica detalladamente distintos tipos de QGA desde el ámbito computacional
(6)	Propone mejoras a procesos de QGA para mejorar convergencia	Análisis comparativo de CGA y QGA mejorado
(22)	Presentan las bases de QGA y los distintos modelos	Comenta sobre las ventajas y desventajas de QGA
(24)	Plantea un GA para reducir costos en comunicación cuántica	Descripción teórica (simple) de teleportación cuántica y minimizar costos con GA
(32)	Propone QGA asistido con reverse quantum annealing	Análisis teórico del método, comparando con otros algoritmos
(5)	Usa GA para encontrar los máximos valores en la desigualdad de Bell	Compara resultados empíricos con los obtenidos analíticamente
(25)	Utiliza GA para encontrar los protocolos óptimos para una compuerta de adición	Demuestran los resultados de su implementación utilizando el simulador de IBM
(4)	Presenta un desarrollo matemático enfocado en la complejidad del algoritmo	Analiza matemáticamente como funciona un QGA y su complejidad
(30)	Propone QGA asistido con algoritmo de Grover	Compara los resultados entre QGA asistido y CGA
(31)	Presenta de manera simple funcionamiento de un QGA	Comenta ventajas de QGA y GA y dificultades de implementarlo

Tabla II. Contribuciones en el ámbito de la optimización.

**4.1. Definición de la Población.** - En un QGA se codifica un cromosoma como  $m$  secuencias de  $k$  qubits, que representan  $m$  variables codificadas en  $k$  bits que representan un individuo en la población. La población del QGA es inicializada de la siguiente forma:

```

1 def Init_population():
2     # Hadamard gate
3     r2=math.sqrt(2.0)

```

```

4 h=np.array([[1/r2,1/r2],[1/r2,-1/r2]])
5 # Rotation Q-gate
6 theta=0;
7 rot =np.empty([2,2])
8 # Initial popul array (individual x chromosome)
9 i=1; j=1;
10 for i in range(1,popSize):
11     for j in range(1,genomeLength):
12         theta=np.random.uniform(0,1)*90
13         theta=math.radians(theta)
14         rot[0,0]=math.cos(theta);
15         rot[0,1]=-math.sin(theta);
16         rot[1,0]=math.sin(theta);
17         rot[1,1]=math.cos(theta);
18         AlphaBeta[0]=rot[0,0]*(h[0][0]*QuBitZero[0])+rot[0,1]*(h[0][1]*QuBitZero[1])
19         AlphaBeta[1]=rot[1,0]*(h[1][0]*QuBitZero[0])+rot[1,1]*(h[1][1]*QuBitZero[1])
20         # alpha squared
21         qpv[i,j,0]=np.around(2*pow(AlphaBeta[0],2),2)
22         # beta squared
23         qpv[i,j,1]=np.around(2*pow(AlphaBeta[1],2),2)

```

**4.2. Rotación.** - En un QGA las compuertas de rotación son las que se encargan de realizar las rotaciones. Se aplica la compuerta de rotación  $U(\theta)$  a cada par  $(\alpha_i, \beta_i)$  según la ecuación 2. La función de ajuste (Fitness) se obtiene de la siguiente forma:

```

1 # Funcion de Ajuste
2 def fitness_function(x):
3     return sum(x)
4 def numpy_fitness_function(population):
5     return np.sum(population, axis=1)

```

**4.3. Mutación.** - La compuerta de mutación invierte los valores de  $(\alpha, \beta)$  en  $(\beta, \alpha)$ , lo que equivale a una compuerta NOT (o bien, compuerta X).

```

1 def random_mutation(population, gen_length):
2     for i in range(np.random.randint(len(population))):
3         if 0.1 > np.random.randint(100)/100:
4             mutation_index = np.random.randint(len(population))
5             population[mutation_index] = 1 - population[mutation_index]
6     return population

```

**4.4. Cruzamiento.** - Una forma de hacer cruzamiento es seleccionar una posición en 2 de los cromosomas e intercambiar las secciones posteriores a ese punto según lo siguiente:

```

1 def roulette_crossover(population, fitness, gen_length):
2     total_fit = sum(fitness)
3     probs = fitness/total_fit
4     parent_indices = []
5     for i in range(len(probs)):
6         if np.random.randint(0,100)/100 < probs[i]:
7             parent_indices.append(i)
8     for i in range(0, len(parent_indices), 2):
9         parent1 = population[i - 2]
10        parent2 = population[i - 1]
11        crossover_point = np.random.randint(1, gen_length - 1)
12        child1 = np.concatenate((parent1[:crossover_point], parent1[crossover_point
:]))
13
14        child2 = np.concatenate((parent2[:crossover_point], parent2[crossover_point
:]))
15        population[i-2] = child1
16        population[i-1] = child2
17    return population

```

**4.5. Experimentos y Resultados.** - Inicialmente, en el algoritmo de la Figura I se genera un circuito cuántico aprovechando las facilidades que provee Qiskit para simularlos, con tantos qubits según cuantos objetos se quieran en el problema. Luego, al iterar en cada generación se superponen los qubits usando compuertas de Hadamard, para luego colapsar estos y obtener una población clásica con distintos candidatos de soluciones, los cuales son “reparados” para que cumplan con las restricciones duras del problema de no sobrepasar el peso límite de la mochila, sacando objetos aleatoriamente de esta misma.

Finalizando con lo clásico, se evalúa esta población reparada para verificar como está “evolucionando” el algoritmo y de si se alcanzaron los criterios de convergencia de la parte genética. Si no es así, se vuelve a iterar sobre otra generación, pero esta vez con las compuertas rotadas en Y, por una variación de ángulo según qué tan bien se comporta cada individuo  $x$  con el mejor de su generación  $b$ , según se muestra en la Tabla III.

$x_i$	$b_i$	$f(x) \geq f(b)$	$\Delta\theta$
0	0	false	0
0	0	true	0
0	1	false	0
0	1	true	0.05
1	0	false	0.01
1	0	true	0.025
1	1	false	0.005
1	1	true	0.025

Tabla III. Criterio de variación del ángulo de rotación.

Considerando todo lo anterior, se ejecutó la simulación de esta arquitectura para 100 objetos posibles a llevar en la mochila, con pesos aleatorios en un rango de 1 a 10 y valores equivalentes a esos pesos, pero aumentados en 5 unidades. La población se compone de 10 circuitos cuánticos distintos y para una cantidad de generaciones variables se obtuvieron los resultados mostrados en la Tabla IV.

Se aprecia que, independientemente del número de generaciones el algoritmo converge muy rápido a valores cercanos de lo que podría considerarse como el óptimo del problema para los parámetros dados, siendo el peor valor de los mejores ajustes bastante cercano al mejor de todas estas.

Por otra parte, analizando el algoritmo para una instanciación de 10 objetos y 150 generaciones, con el fin de estudiar cómo se comporta el ajuste de forma individual a través de las generaciones, se obtuvo el gráfico de la Figura II que muestra la tendencia del mismo.

Denotando con esto que efectivamente funciona el algoritmo QGA, mejorando a lo largo de las iteraciones de las generaciones, aun pese a las claras variaciones que se tiene entre cada una de estas iteraciones.

**5. Conclusiones y observaciones finales.** - El presente trabajo sobre QGAs permite obtener conocimiento actualizado y generalizado de los avances en esta área de investigación. Considerando la revisión y análisis de la literatura se puede resumir específicamente que los QGAs son una poderosa herramienta para abordar una gran diversidad de problemas complejos, tales como optimización, búsqueda, comunicaciones, entre otros. Permitiendo explorar soluciones de forma eficiente y flexible, hallando soluciones óptimas locales y/o globales, siendo este uno de los problemas que aún existen con este método.

Ejecución	Generaciones	Mejor ajuste
1	5	5787.11
2	8	5858.85
3	14	5932.84
4	18	5929.94
5	25	5902.80
6	30	5906.81
7	35	5933.64
8	40	5922.10
9	50	5911.09

Tabla IV. Resultados de una ejecución de HQGA.

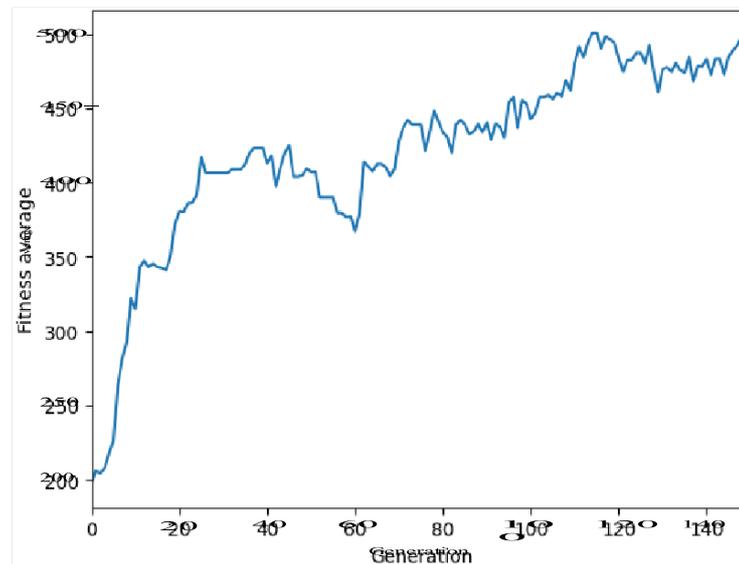


Figura II: Variación del ajuste a través de las generaciones

Los QGAs están presentes hoy en día en diversos tipos de problemas y áreas, partiendo por problemas de optimización combinatorial, planificación, optimización de funciones multimodales, etc; y en áreas tan diversas como arquitectura, medicina, economía, entre otros.

Si bien existen escenarios y dominios de aplicación específicos donde los CGAs o los enfoques cuánticos puros pueden ser más adecuados, los QGAs destacan por su capacidad de manejar problemas complejos y su flexibilidad para adaptarse a diferentes contextos.

Se han experimentado avances significativos, especialmente con respecto a nuevas técnicas, tales como la inclusión de operadores genéticos cuánticos y esquemas de codificación eficiente, ayudando a que los QGA converjan más rápido. De acuerdo a los experimentos realizados con QGAs, se puede decir que, dado que estos algoritmos se ejecutan en una simulación de un entorno cuántico, estos no alcanzan la optimización temporal que se debería obtener al ejecutarlos en un computador cuántico real, teniendo en algunos casos una demora de varios minutos para la generación de resultados. Los algoritmos genéticos convergen rápidamente en las iteraciones iniciales, variando distintos parámetros para ver si existe otro óptimo.

La representación cuántica utilizada en el cromosoma cuántico es crucial para el desempeño del QGA. Una elección adecuada de la codificación y el diseño de compuertas cuánticas optimizadas pueden mejorar la capacidad del algoritmo para encontrar soluciones óptimas. Al contrario, la mala selección de estas compuertas puede provocar variaciones enormes en el algoritmo, escapando de la búsqueda y convergencia hacia un óptimo.

Por último, se destaca la biblioteca Qiskit que proporciona una plataforma sólida y flexible para implementar QGAs. Su conjunto de herramientas y funcionalidades facilita la implementación de las operaciones genéticas y la evaluación de la aptitud en el contexto cuántico.

## Referencias

- [1] Lee Spector, Howard Barnum, Herbert J. Bernstein, Nikhil Swamy. “7: Quantum Computing Applications of Genetic Programming”. The MIT Press. 1999. <https://doi.org/10.7551/mitpress/1110.003.0010>
- [2] Mohammad Mojrián y Seyed Abolghasem Mirroshandel. “A novel extractive multi-document text summarization system using quantum-inspired genetic algorithm: MTSQIGA”. *Expert Systems with Applications* 171 (2021), pp. 114555. issn: 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2020.114555>. url: <https://www.sciencedirect.com/science/article/pii/S0957417420311994>.
- [3] Kuk-Hyun Han y Jong-Hwan Kim. “Genetic quantum algorithm and its application to combinatorial optimization problem”. *Proc. of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*. 2 (2000), pp. 1354-1360, vol. 2. doi: 10.1109/CEC.2000.870809.
- [4] Andrea Malossini, Enrico Blanzieri y Tommaso Calarco. “Quantum Genetic Optimization”. *IEEE Transactions on Evolutionary Computation* 12.2 (2008), pp. 231-241. doi: 10.1109/TEVC.2007.905006.
- [5] Y. Hardy y W.-H Steeb. “Genetic Algorithms and Optimization Problems in Quantum Computing”. *Int. Journal of Modern Physics C -IJMPC* 21 (2010), pp. 1359-1375. doi: 10.1142/S0129183110015890.
- [6] Huaixiao Wang et al. “The Improvement of Quantum Genetic Algorithm and Its Application on Function Optimization”. *Mathematical Problems in Engineering* (2013). doi: 10.1155/2013/730749.
- [7] Hatem M. H. Saad et al. “Quantum-Inspired Genetic Algorithm for Resource-Constrained Project-Scheduling”. *IEEE Access* 9 (2021), pp. 38488-38502. doi: 10.1109/ACCESS.2021.3062790.
- [8] Enrique Ballinas y Oscar Montiel Ross. “Hybrid Quantum Genetic Algorithm for the 0-1 Knapsack Problem in the IBM Qiskit Simulator”. *Computacion y Sistemas* 26 (2022), pp. 725-742. doi: 10.13053/CyS-26-2-4253.
- [9] Arufe L, González MA, Oddi A, Rasconi R, Varela R. “Quantum circuit compilation by genetic algorithm for quantum approximate optimization algorithm applied to MaxCut problem”. *Swarm Evol Comput* 69:101030. (2022) <https://doi.org/10.1016/j.swevo.2022.101030>
- [10] In'és Hilali-Jaghdam et al. “Quantum and classical genetic algorithms for multilevel segmentation of medical images: A comparative study”. *Computer Communications* 162 (2020), pp. 83-93. doi: <https://doi.org/10.1016/j.comcom.2020.08.010>. url: <https://www.sciencedirect.com/science/article/pii/S0140366420318892>.
- [11] Lensen A, Xue B, Zhang M (2021). “Genetic programming for evolving a front of interpretable models for data visualization”. *IEEE Trans Cybern* 51(11):5468–5482. <https://doi.org/10.1109/TCYB.2020.2970198>
- [12] Zhaoyang Huang et al. “Optimal design of load frequency active disturbance rejection control via double-chains quantum genetic algorithm”. *Neural Computing and Applications* 33 (2021). doi: 10.1007/s00521-020-05199-6.
- [13] Durán, C., Carrasco, R., Soto, I. et al. “Quantum algorithms: applications, criteria and metrics. *Complex*”. *Intell. Syst.* 9, 6373–6392 (2023). <https://doi.org/10.1007/s40747-023-01073-9>
- [14] Qian X, Wang S, Li C, Wang G (2019). “Multi channels data fusion algorithm on quantum genetic algorithm for sealed relays”. *J Phys Conf Ser* 1237(2):22130. <https://doi.org/10.1088/17426596/1237/2/022130>
- [15] Xinjian Pan et al. “Self-calibration for linear structured light 3D measurement system based on quantum genetic algorithm and feature matching”. *Optik* 225

(2021), pp. 165749. issn: 0030-4026. doi: <https://doi.org/10.1016/j.ijleo.2020.165749>. url: <https://www.sciencedirect.com/science/article/pii/S003040262031576X>.

[16] Yuxing Wang y Chunyu Wei. "Design optimization of office building envelope based on quantum genetic algorithm for energy conservation". *Journal of Building Engineering* 35 (2021), pp. 102048. issn: 2352-7102. doi:<https://doi.org/10.1016/j.jobe.2020.102048>. url: <https://www.sciencedirect.com/science/article/pii/S2352710220336809>.

[17] Jia-Chu Lee et al. "Quantum genetic algorithm for dynamic economic dispatch with valve-point effects and including wind power system". *Int. Journal of Electrical Power Energy Systems* 33 (2011), pp. 189-197. doi: 10.1016/j.ijepes.2010.08.014.

[18] Wang B, Zhao W, Lin S, Ke J, Wu H (2022). "Integrated energy management of highway service area based on improved multiobjective quantum genetic algorithm". *Dianwang Jishu/Power Syst Technol* 46(5):1742–1751. <https://doi.org/10.13335/j.1000-3673.pst.2021.1610>

[19] Junhe Wan et al. "Fractional-Order PID Motion Control for AUV Using Cloud-Model-Based Quantum Genetic Algorithm". *IEEE Access* 7 (2019), pp. 124828-124843. doi: 10.1109/ACCESS.2019.2937978.

[20] Zhu X, Xiong J, Liang Q (2018). "Fault diagnosis of rotation machinery based on support vector machine optimized by quantum genetic algorithm". *IEEE Access* 6:33583–33588. <https://doi.org/10.1109/ACCESS.2018.2789933>

[21] Guangfeng Cheng, Chunhua Wang y Cong Xu. "A novel hyper-chaotic image encryption scheme based on quantum genetic algorithm and compressive sensing". *Multimedia Tools and Applications* 79 (2020). doi: 10.1007/s11042-020-09542-w.

[22] Susan Stepney y John A Clark. "Evolving quantum programs and protocols". *Handbook of Theoretical and Computational Nanotechnology* 3 (2006), pp. 113-160.

[23] R. Lahoz-Beltra. "Quantum Genetic Algorithms for Computer Scientists". *Computers* 5 (2016), pp. 24. doi: 10.3390/computers5040024.

[24] Mahboobeh Houshmand et al. "An Evolutionary Approach to Optimizing Teleportation Cost in Distributed Quantum Computation". *Int. Journal of Theoretical Physics* 59 (2020). doi: 10.1007/s10773-020-04409-0.

[25] Rui Li et al. "Approximate Quantum Adders with Genetic Algorithms: An IBM Quantum Experience". *Quantum Measurements and Quantum Metrology* 4 (2017), pp. 1. doi: 10.48550/arXiv.1611.07851.

[26] B. I. P. Rubinstein, "Evolving quantum circuits using genetic programming," *Proc. of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, Seoul, Korea (South), 2001, pp. 144-151 vol. 1, doi: 10.1109/CEC.2001.934383.

[27] Riccardo Rasconi y Angelo Oddi. "An Innovative Genetic Algorithm for the Quantum Circuit Compilation Problem". *Proc. of the AAAI Conf. on Artificial Intelligence* 33 (2019), pp. 77077714. doi: 10.1609/aaai.v33i01.33017707.

[28] Zakaria Laboudi y Salim Chikhi. "Comparison of Genetic Algorithm and Quantum Genetic Algorithm". *Int. Arab Journal of Information Technology* 9, (2012).

[29] Akira SaiToh, Robabeh Rahimi y Mikio Nakahara. "A quantum genetic algorithm with quantum crossover and mutation operations". *Quantum Information Processing* 13 (2014), pp. 737. doi: 10.1007/s11128-013-0686-6.

- [30] Jirayu Supasil, Poramet Pathumsoot y Sujin Suwanna. “Simulation of implementable quantumassisted genetic algorithm”. *Journal of Physics: Conference Series* 1719 (2021), pp. 012102. doi: 10.1088/1742-6596/1719/1/012102.
- [31] Bart Rylander, Terence Soule, James A. Foster, Jim Alves-Foss. “Quantum Genetic Algorithms.” *Proc. of the Genetic and Evolutionary Computation Conference (GECCO '00)*, Las Vegas, Nevada, USA, July 8-12, 2000
- [32] James King, Masoud Mohseni, William Bernoudy, Alexandre Fréchet, Hossein Sadeghi, Sergei V. Isakov, Hartmut Neven, Mohammad H. Amin. “Quantum-Assisted Genetic Algorithm”. arXiv:1907.00707, 2019.
- [33] Ruben Ibarrondo, Giancarlo Gatti, Mikel Sanz. “Quantum vs classical genetic algorithms: A numerical comparison shows faster convergence”. *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*. <http://dx.doi.org/10.1109/SSCI51031.2022.10022159>, 10.1109/ssci51031.2022.10022159
- [34] Yu-Fang C, Hao X, Wen-Cong H, Liang Z (2018). “An improved multi-objective quantum genetic algorithm based on cellular automaton”. In: *2018 IEEE 9th Int. Conf. on Software Engineering and Service Sciences*. IEEE, Beijing, pp 342–345. <https://doi.org/10.1109/ICSESS.2018.8663840>
- [35] Creevey, F.M., Hill, C.D. Hollenberg, L.C.L. “GASP: a genetic algorithm for state preparation on quantum computers”. *Sci Rep* 13, 11956 (2023). <https://doi.org/10.1038/s41598-023-37767-w>

**Nota contribución de los autores:**

1. Concepción y diseño del estudio
2. Adquisición de datos
3. Análisis de datos
4. Discusión de los resultados
5. Redacción del manuscrito
6. Aprobación de la versión final del manuscrito

MS ha contribuido en: 1, 2, 3, 4, 5 y 6.

VF ha contribuido en: 1, 2, 3, 4, 5 y 6.

FM ha contribuido en: 1, 2, 3, 4, 5 y 6.

FP ha contribuido en: 1, 2, 3, 4, 5 y 6.

LD ha contribuido en: 1, 2, 3, 4, 5 y 6.

**Nota de aceptación:** Este artículo fue aprobado por los editores de la revista Dr. Rafael Sotelo y Mag. Ing. Fernando A. Hernández Goberti.