



# Quantum Implementation of a Genetic Algorithm



---

**Mauricio Solar, Vicente Figueroa, Francisco Manriquez,  
Francisco Pizarro, Liuba Dombrovskaja**

Departamento de Informática  
Universidad Técnica Federico Santa María, Chile

1er Taller de Ingeniería de Software Cuántico, CLEI  
Bahía Blanca, 2024



# Contenido

---

- **Introducción**
- **Definición de un Algoritmo Genético (GA)**
- **Estado del Arte**
- **Implementación de un GA Cuántico (QGA)**
- **Conclusiones y observaciones finales**



# Introducción

- **Artículo:** muestra avances en el campo de la programación genética aplicada a la CC. Avances en optimización de circuitos, diseño de algoritmos, resolución de problemas, simulación, seguridad y optimización de sistemas cuánticos (amplio espectro de aplicaciones que están acelerando el desarrollo de la CC).
- **Aplicaciones:** imágenes médicas, visualización de datos, comunicaciones, arquitectura, energía eólica, gestión de energía, control, diagnóstico de fallas, uso de criptografía cuántica en imágenes.
- Evolución de programas cuánticos ha llevado al desarrollo de algoritmos de criptografía más robustos y resistentes, impulsando la seguridad en el ámbito cuántico.
- A partir de pseudo-códigos y diagramas se extrapola una versión con el funcionamiento de estos en las librerías de Qiskit.
- Se muestra implementación (Qiskit) de un Algoritmo Genético Cuántico (QGA) basado en un QGA Híbrido (HQGA) y se presenta los métodos para simular un entorno cuántico para el desarrollo de los mismos circuitos utilizando Python.



# Definición de un AG

- Una de las ramas que más prometen en CC es la aplicación de GA a estos mismos. Los GA son técnicas de optimización inspiradas en los principios de la evolución biológica, que han demostrado eficacia en la resolución de problemas complejos en una amplia gama de áreas, desde la planificación de rutas hasta diseño de estructuras.
- Combinación de GA con CC permite aprovechar ventajas de ambas disciplinas.
- QGA:
  - tiene potencial de superar limitaciones de CGA, ya que pueden explorar múltiples soluciones en paralelo gracias a las propiedades cuánticas como la superposición y la interferencia cuántica, para un buen número de casos específicos.
  - basado en codificación de qubits para el almacenamiento de información puede ocupar cualquier representación conocida basada en bits, ocupando un qubit por bit, pero pueden surgir limitaciones por la disponibilidad de qubits en un computador cuántico.



# Definición de un AG

- QGA: codifica un cromosoma como  $m$  secuencias de  $k$  qubits, que representan  $m$  variables codificadas en  $k$  bits según:

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

- donde  $q_j^t$  representa el cromosoma  $j$  de la  $t$ -ésima generación.
- **Codificación inicial:**  $(\alpha, \beta)$  de cada  $\Psi$  como  $(1/\sqrt{2}, 1/\sqrt{2})$  (compuerta de Hadamard), de manera que la probabilidad de cualquier estado del cromosoma sea el mismo.



# Definición de un AG

## Rotación

- GA: actualizar cromosomas de manera que se acerquen al valor esperado.

- QGA: compuertas de rotación  $U(\theta_i)$  aplicada a cada par  $(\alpha_i, \beta_i)$ :

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

- $U(\theta_i)$ : equivalente a una rotación en eje Y en esfera de Bloch.
- Ángulo  $\theta_i$  fijo se define manualmente (estrategia de ajuste) típico entre  $0.1\pi$  y  $0.005\pi$
- Algoritmo de rotación depende de tablas de valores fijos, que determinan dirección de rotación según fitness del cromosoma en relación al mejor.
- Dependencia de tabla afecta la convergencia  $\Rightarrow$  método de rotación auto-adaptada. Reduce el ángulo de rotación según la generación, sin perder la dirección hacia el mejor cromosoma encontrado, según:

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

- $\theta_{max}$  y  $\theta_{min}$  valores fijos,  $it_{max}$  iteración máxima e  $iter$  es generación actual.
- $\theta_i$  disminuye según avanzan las iteraciones, reduciendo la función de ajuste (Fitness).



# Definición de un AG

---

## Mutación

- Permite que AG desarrolle nuevas posibilidades de valores, desviándose del resto de la población, lo que puede ayudar a evitar caer en óptimos locales.
- Compuerta de mutación invierte valores de  $(\alpha, \beta)$  en  $(\beta, \alpha)$ , que equivale a compuerta NOT (compuerta X).



# Definición de un AG

---

## Cruzamiento

- Teóricamente no se puede hacer un equivalente del cruzamiento en versión cuántica
- Se puede ocupar en algoritmos basados en CC.
- Forma de hacer cruzamiento: seleccionar una posición en 2 cromosomas e intercambiar las secciones posteriores a ese punto.

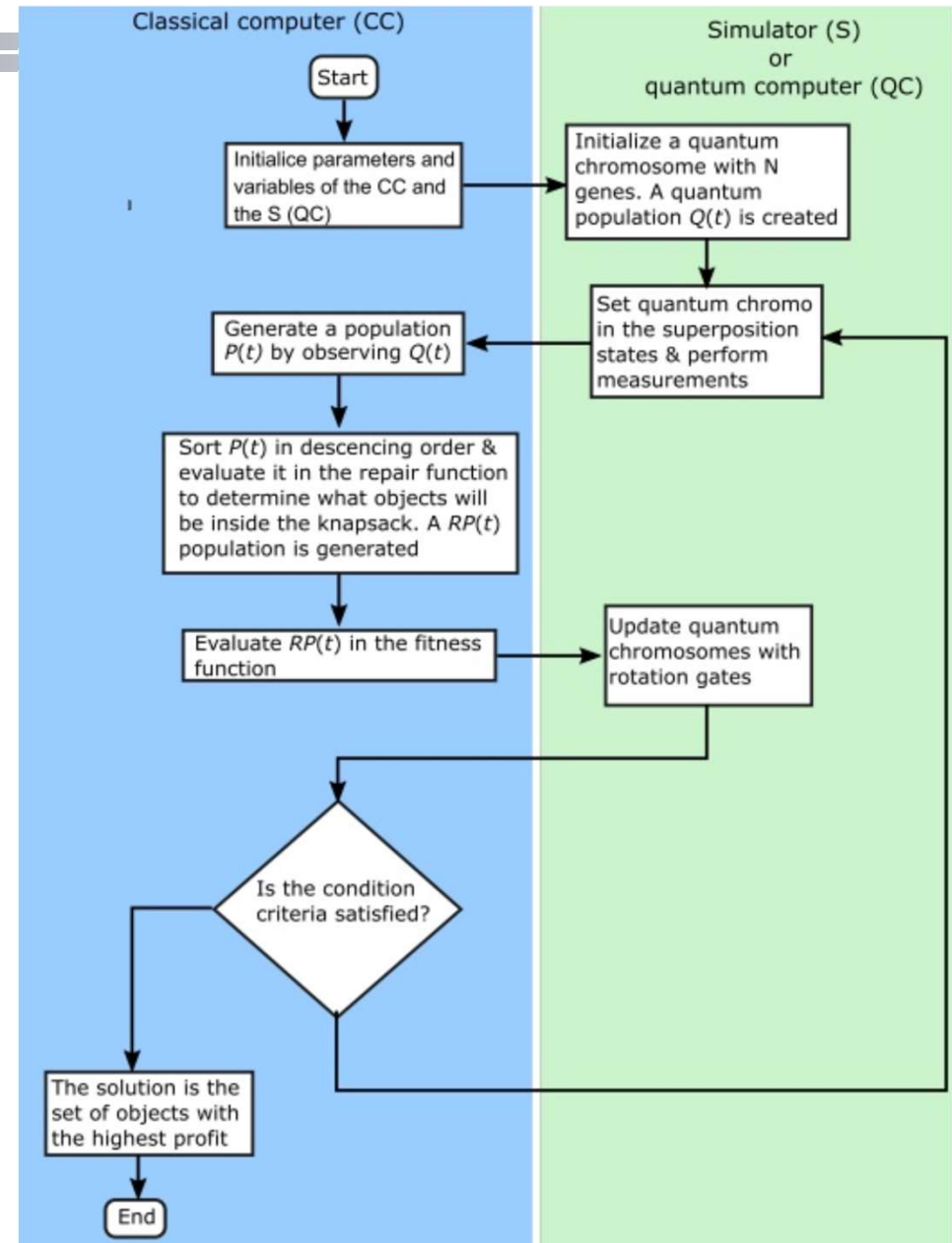




# Definición de un AG

## Diagrama del QGA

- Se selecciona la arquitectura híbrida HQGA (Ballinas y Montiel, 2022), para implementar elementos de la CC como elementos de la computación clásica.
- Arquitectura orientada a la resolución del problema de la mochila (Knapsack), generando aleatoriamente valores asociados a cada objeto con una distribución uniforme entre 1 y 10.





# Estado de Arte

## Metodología

- Para buscar trabajos relacionados con QGA en la literatura se usan distintas técnicas, siendo Snow Ball la más común.
- Se considera trabajos que utilizan como referencia el survey en *Spector et al. (99)*, 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 QGA. Se obtuvieron 6 trabajos de interés con este método.
- 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 20 trabajos donde se usa QGA que aportan a problemas en diversas áreas.



# Estado de Arte

## Descripción de trabajos con QGA

- Área de control:
  - *Wan et al. (19)*: 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).
  - *Huang et al. (21)*: 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.
- Arquitectura:
  - *Wang et al. (21)*: optimiza 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.
  - *Pan et al. (21)*: aprovecha 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.



## Descripción de trabajos con QGA

- Energía:
  - *Lee et al. (11)*: utiliza QGA para minimizar costos en la producción de energía eólica. Utiliza 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.
  - *Wang et al. (22)*: 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.
- Diagnóstico de fallas (*Zhu et al., 18*): propone un método de diagnóstico en maquinaria rotacional, basado en SVM optimizada con QGA. Aplica a la detección de fallas en maquinaria rotativa sobre un eje, obteniendo una mayor precisión que los tradicionales SVM y GA (i.e. CGA).
- Búsqueda textual (*Mojrián et al., 21*): propone variante de QGA (MTSQIGA), calculando el ángulo para la compuerta de rotación comúnmente usada a partir de los resultados de la iteración, cuando no cumple el criterio de longitud impuesto.



# Estado de Arte

## Descripción de trabajos con QGA

- Imágenes:
  - *Cheng et al. (20)*: utiliza QGA para 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.
  - *Hilali-Jaghdam et al. (20)*: realiza una comparación entre CGA, QGA y PSO (Particle Swarm Optimization) para el procesamiento de imágenes médicas. QGA utiliza un algoritmo de rotación simple. PSO obtiene mejores resultados y converge más rápido a la solución. Segundo lugar se encuentra QGA, último lugar CGA.
- Comunicaciones:
  - *Durán et al. (23)*: obtuvo criterios, métricas y resultados experimentales, que demuestra que QGA es mejor que CGA. Proporciona discusión sobre función objetivo, convergencia al óptimo global y necesidad de mejorar soluciones obtenidas en la aplicación.
  - *Qian et al. (19)*: presenta 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.



# Estado de Arte

## Descripción de trabajos con QGA

- Optimización (1/5):
  - *Han et al. (2000)*: prueba y muestra la efectividad de usar QGA en problemas de optimización combinatoriales, específicamente el problema de la mochila, logrando un rendimiento superior a muchos otros GAs. Demuestra y simula el rendimiento de un QGA asistido. Realiza la prueba haciendo uso de optimización binaria cuadrática sin restricciones (QUBO) durante 100 iteraciones y compara con su contraparte clásica durante 2000 iteraciones, demostrando que el QGA converge a la solución óptima mucho más rápido que el CGA.
  - *Wang et al. (13)*: 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 una eficiencia mayor a un CGA para resolver problemas de optimización.
  - *Hardy et al. (2010)*: realiza pruebas y muestra resultados de aplicar AG en diversos problemas de optimización en CC. Estudia la desigualdad de Bell (Bell-CHSH) y la medida del entrelazamiento entre qubits.



# Estado de Arte

## Descripción de trabajos con QGA

- Optimización (2/5):
  - *Hatem et al. (21)*: propone una implementación de QGA en problemas RCPSP (Resource-Constrained Project-Scheduling Problem). Utiliza cruzamiento cuántico y mutación cuántica, junto a operaciones de rotación. Estudia su complejidad y compara su velocidad y eficacia contra algoritmos clásicos encontrados.
  - *Rasconi et al. (19)*: utiliza una mejora de un algoritmo GRS (Greedy Random Sampling) y un nuevo GA para solucionar el problema de compilación de circuitos cuánticos. GA es mejor que GRS para instancias de grandes proporciones.
  - *Laboudi et al. (12)*: realiza comparación CGA y QGA, utilizando el problema de la mochila. Resultados muestran que QGA llega a mejores resultados cuando el tamaño del problema es mucho mayor. Complejidad  $O(N)$  para QGA y  $O(N^2)$  para CGA.
  - *Li et al. (17)*: propone una versión mejorada del Quantum Genetic Optimization Algorithm (QGOA) con el objetivo de acelerar los distintos procesos de un GA. Requiere un número menor de operaciones para identificar la función de ajuste para una subpoblación en cada generación, reemplazando los procesos tradicionales por un único proceso cuántico, reduce complejidad a  $O(1)$  en términos de llamadas en el proceso de selección.



# Estado de Arte

## Descripción de trabajos con QGA

- Optimización (3/5):
  - *Li et al. (17)*: analizan propuestas tecnológicas para aproximar de forma cuántica las puertas, y optimizarlas mediante el uso de GAs, llegando a nuevos niveles en términos de eficiencia y confiabilidad.
  - *Yu-Fang et al. (18)*: muestra un GA celular con optimización multi-objetivo y un QGA para crear autómatas celulares. 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.
  - *Creevey et al. (23)*: presenta un GA para la preparación de estados (GASP) 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 Rx, Ry, Rz y CNOT y un AG 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.





# Estado de Arte

## Descripción de trabajos con QGA

- Optimización (4/5):
  - *Spector et al. (1999)*: describe cómo simular un computador cuántico para determinar la idoneidad de un algoritmo cuántico en hardware clásico. Describe 3 enfoques de programación genética para impulsar el simulador para desarrollar nuevos algoritmos cuánticos. Los enfoques son: (i) programación genética estándar basada en árboles, (ii) programación genética del genoma lineal basada en pilas y (iii) programación genética del genoma lineal sin pilas. Pruebas en 4 problemas: de promesa temprana de 2 bits, de escalamiento mayoritario, de búsqueda en BDs de 4 elementos y el and/or de 2 bits. Para 3 problemas, los algoritmos descubiertos automáticamente son más eficientes que los clásicos.
  - *Rubinstein (2001)*: propone nueva representación de conjunto de operadores genéticos para optimizar circuitos cuánticos con programación y GAs, heredando la representación y propiedades de ambos. Circuitos cuánticos son colección de puertas, operando en varias categorías de qubits con sus parámetros. En particular, se aplican los circuitos en el problema de la producción del entrelazamiento entre qubits.
  - *SaiToh et al. (14)*: propone la implementación cuántica del cruzamiento, dado que ha sido implementado siempre en forma clásica. Propone que el cruzamiento se realice de forma paralela entre generaciones, logrando una aceleración del orden cuadrático sobre la clásica.



## Descripción de trabajos con QGA

- Optimización (5/5):
  - *Lahoz-Beltra (2016)*: presenta una revisión de distintos GAs implementados en CC, orientada a carreras de ciencias de la computación, obviando aspectos físicos relacionados con la CC. Analiza ventajas y desventajas en comparación a los clásicos.
  - *Stepney et al. (2006)*: abarca la teoría y práctica de las bases de los algoritmos evolutivos cuánticos. Introduce la cuántica (computacional y física), y algoritmos evolutivos cuánticos, analiza principales problemas de implementación en diversos problemas, y evalúa y muestra el desempeño de estos algoritmos.
  - *SaiToh et al. (14)*: presenta una aproximación simple a los QGAs, analizando beneficios y desventajas al aplicar paralelización cuántica a los CGA.
  - *Housmand et al. (20)*: optimiza algoritmos de comunicación de sistemas cuánticos distribuidos, compuestos por circuitos cuánticos de baja capacidad. Compara resultados con algoritmo de trabajo anterior, logrando mejora en velocidad.
  - *King et al. (19)*: usa reverse quantum annealing, una variante de algoritmo de evolución cuántica, que usa búsquedas semi-locales o semi-no locales desde un estado inicial. Permite que los GAs usen fluctuaciones cuánticas para las mutaciones (Quantum-Assisted Genetic Algorithms, QAGAs). Es capaz de encontrar buenas soluciones del óptimo global a diferencia del standard quantum annealing.



# Análisis comparativo

## Contribuciones en diferentes ámbitos:

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



# Análisis comparativo

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



# Discusión Bibliográfica

- Se espera que los QGA tengan un rendimiento mayor a los algoritmos cuánticos y a los CGA, por su capacidad de encontrar óptimos locales en pocas iteraciones.
- Muchos trabajos analizados centran estudio en aplicar QGA en distintas áreas mas que en la investigación de optimizar el método para escapar de óptimos locales o acelerar convergencia.
- Hay 3 trabajos que buscan hacer cuántica la parte “clásica” del QGA, implementando mutaciones y cruzamientos cuánticos.
- Resultados esperanzadores:
  - *Wang et al. (2013)*: sienta las 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 venia aplicando hace años.
  - *SaiToh et al. (2014)*: primera implementación cuántica al cruzamiento de un GA, abriendo paso a un nuevo campo de estudio, permitiendo conocer como se comporta un “híbrido” de algoritmo clásico con cuántico.
  - *King et al. (2019)*: versión cuántica de las mutaciones clásicas de un GA.



# Implementación de un QGA

- Metodología usada paso a paso para diseñar y codificar los elementos que componen un GA, tales como inicialización de la población y los distintos métodos aplicados para el cruzamiento y la mutación de los genes en cada población.
- Se evalúan los experimentos realizados en los circuitos cuánticos diseñados, demostrando como los QGA logran una optimización eficiente.

```
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
```



# Implementación de un QGA

- **Definición de la Población:** QGA codifica un cromosoma como  $m$  secuencias de  $k$  qubits, que representan  $m$  variables codificadas en  $k$  bits que representan un individuo en la población. Inicialización de Población:

```
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)
```



# Implementación de un QGA

- **Rotación:** Compuertas de rotación encargadas de realizar las rotaciones. Se aplica la compuerta de rotación  $U(\theta_i)$  a cada par  $(\alpha_i, \beta_i)$ .
- Obtención de la función de ajuste (Fitness):

```
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)
```





# Implementación de un QGA

- **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
```



# Implementación de un QGA

- **Cruzamiento:** seleccionar una posición en 2 cromosomas e intercambiar las secciones posteriores a ese punto:

```
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        :]))
16        population[i-2] = child1
17        population[i-1] = child2
18    return population
```



# Experimentos y Resultados

- Inicialmente, el algoritmo genera un circuito cuántico (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.
- Finalizando con lo clásico, se evalúa la 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 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  $b$ .

$x_i$	$b_i$	$f(x) \geq f(b)$	$\Delta\theta$
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



# Experimentos y Resultados

- Simulación para 100 objetos (posibles en la mochila), pesos aleatorios en rango de 1 a 10 y valores equivalentes a los pesos aumentados en 5 unidades.
- Población compuesta de 10 circuitos cuánticos para generaciones variables:

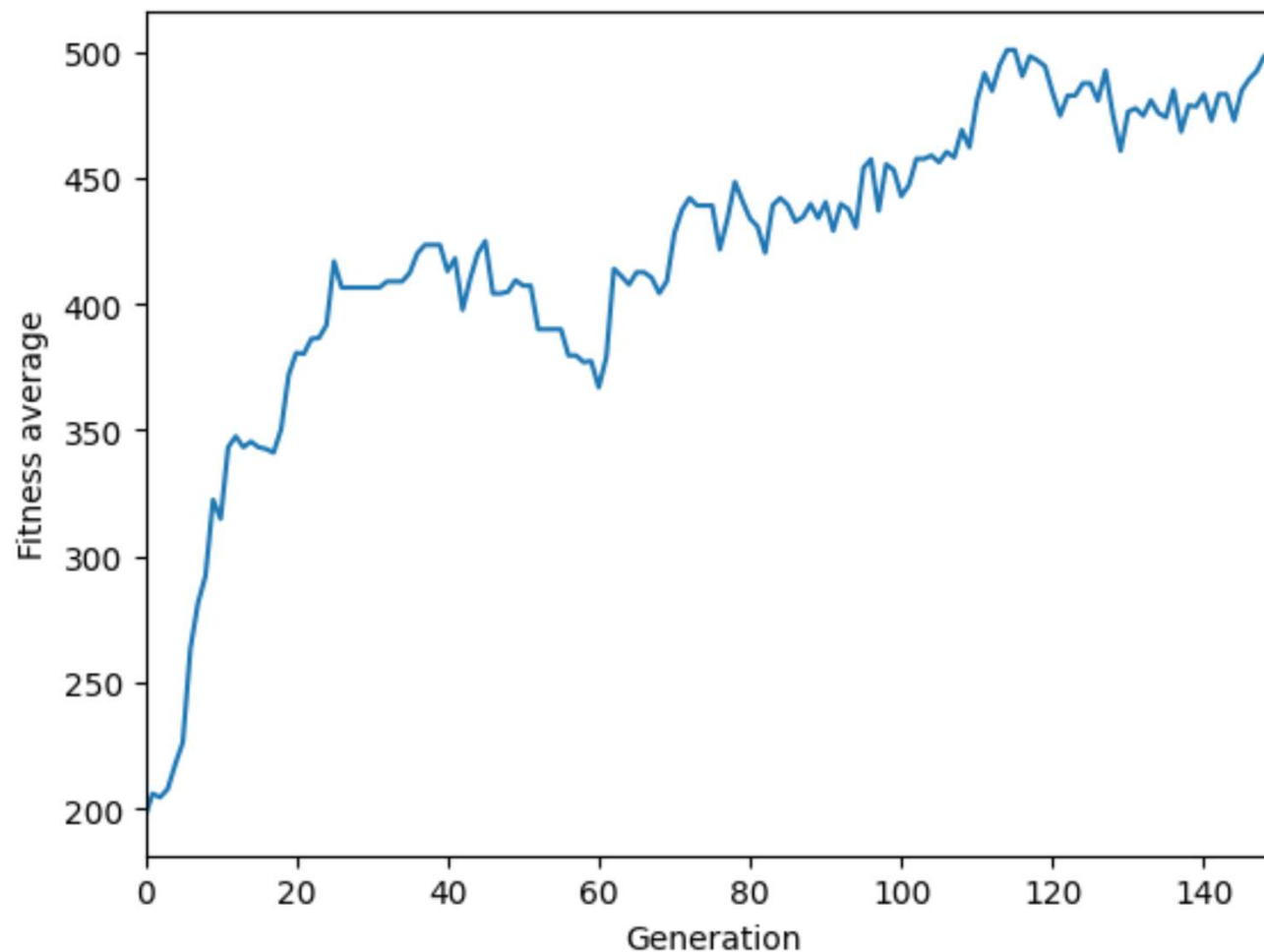
Ejecución	Generaciones	Mejor ajuste
1	5	5787.11
2	8	5858.85
3	14	5932.84
4	18	5929.94
5	25	5902.80
6	30	5906.81
7	35	5933.64
8	40	5922.10
9	50	5911.09

- Independiente del número de generaciones el algoritmo converge muy rápido a valores cercanos al óptimo del problema para los parámetros dados, siendo el peor valor de los mejores ajustes bastante cercano al mejor de todas estas.



# Experimentos y Resultados

- Analizando el algoritmo para 10 objetos y 150 generaciones, para estudiar como se comporta el ajuste de forma individual a través de las generaciones, se muestra la tendencia del mismo.





# Conclusiones

---

- QGAs son una poderosa herramienta para abordar una diversidad de problemas complejos, tales como optimización, búsqueda, comunicaciones, planificación, optimización de funciones multimodales; y en áreas como arquitectura, medicina, economía, entre otros.
- Permitiendo explorar soluciones y hallando soluciones óptimas locales y/o globales, siendo este uno de los problemas que aun existen con este método.
- Existen escenarios y dominios de aplicación específicos donde los CGAs pueden ser más adecuados, los QGAs destacan por su capacidad de manejar problemas complejos y su flexibilidad para adaptarse a diferentes contextos.



# Conclusiones

- 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 GAs convergen rápidamente en las iteraciones iniciales, variando distintos parámetros para ver si existe otro óptimo.
- Representación cuántica utilizada en cromosoma cuántico es crucial para el desempeño del QGA. Elección adecuada de codificación y diseño de compuertas cuánticas optimizadas pueden mejorar la capacidad del algoritmo para encontrar soluciones óptimas. Mala selección de compuertas puede provocar variaciones enormes en el algoritmo, escapando de convergencia hacia un óptimo.