



Universidad Católica  
**San Pablo**

## Título de la Tesis

José Carlos Delgado Ramos

Orientador: Prof Dr. Yván Jesús Túpac Valdivia

*Tesis profesional presentada al Programa Profesional de Ciencia de la Computación como parte de los requisitos para obtener el Título Profesional de Lic. en Ciencia de la Computación.*

UCSP- Universidad Católica San Pablo  
Diciembre de 2014

*You have nothing to do but mention the  
quantum theory, and people will take  
your voice for the voice of science, and  
believe anything*

George Bernard Shaw (1938)

# Abreviaturas

**QIEA** Quantum-Inspired Evolutionary Algorithm

**PDD** Densidad Determinada Probabilística

**PDF** Función Densidad Probabilística

# Abstract

---

This thesis presents a set of modifications to the Quantum-Inspired Evolutionary Algorithm (QIEA)- $\mathbb{R}$  algorithm, which are based both on a separation of the action range of the quantum individuals operating, and the implementation of a crossover operator intended to modify quantum individuals rather than classical individuals. The mentioned algorithms are tested on a set of benchmark functions in the search for improvements in both quality data generation and convergence during the execution of the aforementioned.

# Resumen

---

La presente tesis propone un conjunto de modificaciones al algoritmo evolutivo de inspiración cuántica para reales QIEA- $\mathbb{R}$ , las cuales se basan tanto en una segregación de los campos de acción de los individuos cuánticos durante su operación, así como en la implementación de un operador de cruzamiento que modifique a los individuos cuánticos a diferencia de el grueso de algoritmos que concentran dichos operadores en los individuos clásicos. Los algoritmos previamente mencionados han sido probados sobre un conjunto de funciones *benchmark* en la búsqueda de mejoras tanto en la calidad de la data generada, como en la convergencia durante la ejecución de los ya mencionados.

# Índice general

<b>1. Introducción</b>	<b>2</b>
1.1. Motivación y Contexto . . . . .	2
1.2. Planteamiento del Problema . . . . .	2
1.3. Objetivos . . . . .	3
1.3.1. Objetivos Específicos . . . . .	3
1.4. Organización de la tesis . . . . .	3
<b>2. Estado del Arte</b>	<b>5</b>
2.1. Variantes y modificaciones del QIEA . . . . .	5
2.2. Aplicaciones de los Algoritmos Evolutivos de Inspiración Cuántica . . . . .	6
2.3. Consideraciones Finales . . . . .	6
<b>3. Marco Teórico</b>	<b>7</b>
3.1. Computación Cuántica . . . . .	7
3.2. Computación Evolutiva . . . . .	8
3.3. Algoritmos Evolutivos de Inspiración Cuántica . . . . .	11
3.4. Algoritmo Evolutivo de Inspiración Cuántica - $\mathbb{R}$ (QIEA- $\mathbb{R}$ ) . . . . .	13
3.5. Consideraciones Finales . . . . .	15
<b>4. Propuesta</b>	<b>16</b>
4.1. Segregación del campo operativo de los individuos cuánticos en universos .	16
4.2. Operador de cruzamiento para los individuos cuánticos . . . . .	17

4.3. Detalle y nomenclatura de los algoritmos a evaluar . . . . .	17
<b>5. Pruebas y Resultados</b>	<b>19</b>
5.1. Escenarios de prueba . . . . .	19
5.2. Condiciones para las pruebas . . . . .	21
5.3. Resultados . . . . .	21
5.3.1. Resultados sobre la función Arckley . . . . .	23
5.3.2. Resultados sobre la función Booth . . . . .	23
5.3.3. Resultados sobre la función Maytas . . . . .	27
<b>6. Conclusiones y Trabajos Futuros</b>	<b>32</b>
6.1. Problemas encontrados . . . . .	32
6.2. Recomendaciones . . . . .	32
6.3. Trabajos futuros . . . . .	33
<b>Bibliografía</b>	<b>37</b>

# Índice de tablas

4.1. Descripción de los algoritmos a evaluar . . . . .	18
5.1. Descripción de las funciones de prueba . . . . .	19



# Índice de figuras

3.1. Proceso de un algoritmo evolutivo. [Eiben and Smith, 2003] . . . . .	10
3.2. Rotación del ángulo al momento de la aplicación del operador Q-gate . . .	13
5.1. $f(x, y) = -20e^{-0.2\sqrt{0.5(x^2+y^2)}} - e^{0.5(\cos(2\pi x)+\cos(2\pi y))} + e + 20$ . . . . .	20
5.2. $f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$ . . . . .	20
5.3. $f(x, y) = 0.26(x^2 + y^2) - 0.48xy$ . . . . .	20
5.4. Valor promedio de los $n$ -ésimos mejores individuos para $n = 2$ - Función Arckley . . . . .	22
5.5. Valor promedio de los $n$ -ésimos mejores individuos para $n = 4$ - Función Arckley . . . . .	22
5.6. Valor promedio de los $n$ -ésimos mejores individuos para $n = 10$ - Función Arckley . . . . .	23
5.7. Evolución de los valores de los mejores individuos para $n = 3$ - Función Arckley . . . . .	24
5.8. Evolución de los valores de los mejores individuos para $n = 7$ - Función Arckley . . . . .	24
5.9. Evolución de los valores de los mejores individuos para $n = 9$ - Función Arckley . . . . .	25
5.10. Valor promedio de los $n$ -ésimos mejores individuos para $n = 3$ - Función Booth . . . . .	25
5.11. Valor promedio de los $n$ -ésimos mejores individuos para $n = 6$ - Función Booth . . . . .	26
5.12. Valor promedio de los $n$ -ésimos mejores individuos para $n = 9$ - Función Booth . . . . .	26
5.13. Evolución de los valores de los mejores individuos para $n = 2$ - Función Booth . . . . .	27

---

5.14. Evolución de los valores de los mejores individuos para $n = 4$ - Función Booth . . . . .	28
5.15. Evolución de los valores de los mejores individuos para $n = 9$ - Función Booth . . . . .	28
5.16. Valor promedio de los $n$ -ésimos mejores individuos para $n = 2$ - Función Maytas . . . . .	29
5.17. Valor promedio de los $n$ -ésimos mejores individuos para $n = 6$ - Función Maytas . . . . .	29
5.18. Valor promedio de los $n$ -ésimos mejores individuos para $n = 10$ - Función Maytas . . . . .	30
5.19. Evolución de los valores de los mejores individuos para $n = 2$ - Función Maytas . . . . .	30
5.20. Evolución de los valores de los mejores individuos para $n = 4$ - Función Maytas . . . . .	31
5.21. Evolución de los valores de los mejores individuos para $n = 10$ - Función Maytas . . . . .	31

# Capítulo 1

## Introducción

### 1.1. Motivación y Contexto

A pesar de su existencia aún restringida al campo teórico, la computación cuántica ofrece posibilidades que la computación tradicional no considera ni remotamente realizables, considerando en especial las limitaciones actuales en procesamiento que han obligado a volcar la atención hacia la paralelización en búsqueda de un mayor aprovechamiento de los recursos [Wall, 1991].

Sin embargo, es precisamente la no existencia de ordenadores cuánticos funcionales un limitante en este campo. Aún son pocos los algoritmos cuánticos verdaderos que serían capaces de aprovechar las posibilidades de la computación cuántica [Gruska, 2004]. No obstante, se busca aprovechar desde ya algunas de las características que ofrece la computación cuántica. Es aquí donde entran en acción diversos algoritmos de inspiración cuántica, que buscan traer al paradigma tradicional elementos y fenómenos propios del paradigma cuántico con la ya característica búsqueda de un incremento de la performance, calidad de resultados y disminución de tiempos.

Dada la relativa novedad del campo, los algoritmos evolutivos de inspiración genética han sido todavía poco estudiados y explorados, por lo que se avisan grandes posibilidades en lo que concierne a la investigación en el área.

### 1.2. Planteamiento del Problema

Se puede afirmar con certeza que el grueso de soluciones y aplicaciones a partir del QIEA original contemplan actualizaciones de los individuos cuánticos basadas tanto en la evaluación de, como en la interacción entre individuos clásicos generados a partir de éstos [Nowotniak and Kucharski, 2014]. Esta visión es heredera del accionar propio de distintos tipos de algoritmos evolutivos estudiados y evaluados desde tiempo atrás que se pusiera ojos en la inspiración cuántica.

Se puede apreciar también que los individuos cuánticos en sí no son directamente

evaluados, y mucho menos poseen una verdadera interacción entre sí. A pesar de ser la parte esencial de los algoritmos evolutivos de inspiración cuántica como generadores de individuos cuánticos, son estos últimos los que realmente presentan las características propias de los algoritmos evolutivos, tales como operadores de mutación y cruzamiento, entre otras.

Como se verá en el Capítulo 3, uno de los requerimientos para la formulación de un algoritmo de inspiración cuántica es, de existir alguna separación del campo de búsqueda en universos, permitir la existencia de alguna interacción entre los mismos, la cual ha de permitir hallar la solución o ayudar a que los sub-problemas al interior de cada universo puedan ser resueltos [da Cruz, 2007]. Este requerimiento permitiría controlar de forma más eficiente el grado de actualización global y local, y podría verse cumplido mediante una separación de los espacios de acción de los respectivos individuos cuánticos en universos únicos, los cuales puedan compartir mutuamente sus mejoras para incrementar el espacio de búsqueda de cada cual.

## 1.3. Objetivos

El principal objetivo de la presente tesis es la evaluación del desempeño de un QIEA para  $\mathbb{R}$  que presente las características de segregación/interacción entre universos de operación para cada individuo cuántico en operación. La evaluación es comparativa y se hará con respecto al QIEA- $\mathbb{R}$  original. Se busca evaluar y explorar el hecho de poder tener mayor control sobre el grado de localidad y globalidad de la actualización de cada individuo cuántico mediante la segregación del campo operativo de los mismos, así como la incorporación de un operador de cruzamiento que funcione a este nivel.

### 1.3.1. Objetivos Específicos

- Estudiar el algoritmo QIEA- $\mathbb{R}$ , comprender sus características y evaluar posibles escenarios de mejora con respecto al grado de globalidad/localidad de sus elementos.
- Incrementar el grado de control tanto de la globalidad como de la localidad en la actualización de individuos cuánticos mediante el aislamiento de los individuos cuánticos en universos separados, así como mediante la introducción de un operador de recombinación para la mejora de los espacios de búsqueda de cada individuo cuántico.
- Medición del desempeño tanto del algoritmo original como de las propuestas evaluadas mediante la realización de pruebas en funciones *benchmark*.

## 1.4. Organización de la tesis

El capítulo 2 busca hacer una revisión del estado del arte en el área de los algoritmos evolutivos de inspiración cuántica, incluyendo un apartado referido a los algoritmos

evolutivos de inspiración cuántica en  $\mathbb{R}$ . El capítulo 3 contiene el marco teórico de la tesis, introduciendo los conceptos básicos en los campos de la computación cuántica, la computación evolutiva, y los algoritmos evolutivos de inspiración cuántica. El capítulo 4 describe las características elegidas a ser implementadas y evaluadas sobre el algoritmo ya indicado. El capítulo 5 describe los escenarios y las condiciones elegidas para las pruebas, así como los resultados de los mismos. Finalmente, el capítulo 6 contiene las conclusiones obtenidas por la presente tesis, además de indicaciones adicionales y posibles escenarios para la investigación futuras en el campo.

## Capítulo 2

# Estado del Arte

Mientras que los primeros estudios sobre computación cuántica fueron propuestos en la década de los ochenta [Benioff, 1980] [Benioff, 1982], el primer trabajo sobre técnicas de inteligencia computacional de inspiración cuántica recién fue presentado hacia finales de la década de los noventa [Narayanan and Moore, 1996]. A diferencia de los “auténticos” algoritmos cuánticos tales como el algoritmo de Grover [Grover, 1996] o el algoritmo de Shor [Shor, 1994], los algoritmos de inspiración cuántica no requieren de una computadora cuántica para ser implementados, sino que implementan niveles adicionales de aleatoriedad inspirados en conceptos y principios de la computación cuántica, tales como los *qubits*, la *interferencia* o la *superposición de estados* [Nowotniak, 2010].

El primer trabajo que realmente se puede considerar como un QIEA fue presentado en el año 2001 [Han and Kim, 2000], el mismo que presenta el QIEA- $\mathbb{R}$ . Es acá cuando se introducen los conceptos de qubits y superposición de estados, además de la representación de individuos basados en el concepto de qubits. Se tratarán estos conceptos con mayor profundidad en el siguiente capítulo.

### 2.1. Variantes y modificaciones del QIEA

Debido al poco tiempo de vida de este campo -menos de tres lustros, en realidad-, el área se encuentra aún en crecimiento y sólo existen unos pocos cientos de artículos sobre el tema [Nowotniak and Kucharski, 2014]. Desde el año 2007 se ha dado un incremento notable en la investigación relacionada, concentrando esfuerzos mayoritariamente en aplicaciones, seguido por innovaciones tanto en los operadores genéticos como en las representaciones [Nowotniak, 2010].

En el ámbito de operadores genéticos, ya sean modificados o generalizados, se ha realizado avances tales como la implementación de estrategias de búsqueda de escala mutativa y un método de optimización de caos basado en mapas de Hénon [Teng et al., 2010], una búsqueda en la disminución de operaciones al momento de evaluar la aptitud de los individuos [Malossini et al., 2008], aplicación de rotaciones en los ángulos de las puertas lógicas cuánticas [Zhang et al., 2006], combinación con Cuadrados Latinos [Tsai et al., 2012],

generalización del qubit para una cantidad  $n$  de variables además del operador  $Q$ -gate [Alegría and Túpac, 2014].

Cabe señalar también que en muchos casos se utiliza codificación cuántica binaria de la misma manera que el algoritmo original, mientras que en algunos casos se ha hecho la extensión correspondiente para utilizar valores reales [Qin et al., 2008] [Fan et al., 2007] [Zhao et al., 2009]. Esto, a partir del trabajo desarrollado por [da Cruz, 2007] que propone un algoritmo que genera representaciones en números reales en vez de únicamente representaciones binarias como el algoritmo original.

Si bien la mayoría de algoritmos evolutivos de inspiración cuántica están relacionados estrechamente a los algoritmos genéticos, hay casos en los que se ha llevado dicha inspiración a otros algoritmos evolutivos, tales como a la evolución diferencial [Hota and Pat, 2010] y a la programación genética [Dias et al., 2011] [Dias and Pacheco, 2013].

## 2.2. Aplicaciones de los Algoritmos Evolutivos de Inspiración Cuántica

Al ser derivados de los algoritmos evolutivos, los algoritmos cuántica de inspiración cuántica han sido aplicados en diversos problemas combinatorios y de optimización numérica [Nowotniak, 2010]. Entre algunas de las áreas en las que se han aplicado satisfactoriamente este tipo de problemas, tenemos: procesamiento de imágenes [Talbi et al., 2007], problemas de diseño de redes [Xing et al., 2009], problemas de calendarización de flujo de tienda [Li and Wang, 2007], optimización de sistemas de energía [Miao et al., 2009], generación de instrumentos a escala nanométrica [Dias et al., 2011], entre otros.

Además, los algoritmos representados con números reales han sido exitosamente aplicados a problemas de optimización en ingeniería [Mani and Patvardhan, 2010], modelamiento de opciones de facturación [Fan et al., 2007], optimización de sistemas de energía [Jeong et al., 2010], análisis de datos financieros [Fan et al., 2008], entrenamiento de redes neuronales [Zhao et al., 2009] [Venayagamoorthy and Singhal, 2005], entre otras aplicaciones adicionales.

## 2.3. Consideraciones Finales

El desarrollo de algoritmos evolutivos de inspiración cuántica, al ser un área novedosa y aún poco explorada, tiene todavía mucho que ofrecer. A pesar de ello, ya se ha demostrado su competencia y adaptabilidad a problemas de diversa índole, con garantía de obtención de resultados positivos. Es por esto que se muestran en el presente -y a futuro- como un área de gran potencial y como una alternativa factible para el desarrollo de nuevas soluciones a problemas de distinta índole.

## Capítulo 3

# Marco Teórico

En este capítulo se examinará de forma más profunda las dos áreas de inspiración para la presente tesis: la **computación cuántica** y la **computación evolutiva**.

### 3.1. Computación Cuántica

La computación cuántica es un paradigma de computación que surge en contraposición a la computación clásica, la cual basa su existencia en transistores. En la computación cuántica, se aplican a la computación los principios presentes en la física cuántica, tales como la superposición de estados, entrelazamiento, entre otras [Hey, 1999].

La potencia disponible en la computación clásica está ligada directamente al tamaño de los transistores que componen sus chips. Actualmente, el tamaño de dichos transistores ya está aproximándose a la escala de nanómetros, lo cual limita el aumento de ciclos por reloj para los procesadores, motivo por el cual el foco en la computación clásica ha cambiado para concentrarse en campos como la paralelización [Censor, 1997].

Es aquí donde la computación cuántica muestra sus posibilidades. A diferencia del paradigma tradicional donde la unidad básica es el bit, en la computación cuántica la unidad básica es el *qubit*. La diferencia entre el bit tradicional y el qubit es que mientras el primero sólo puede almacenar un único valor (ya sea este 0 o 1), el segundo puede contener 0, 1 o una superposición de ambos valores, fenómeno al que se llama *superposición de estados*. Es esta superposición de estados la que permite asumir una cantidad simultánea de valores a la vez, y por lo tanto desarrollar una cantidad fantástica de operaciones simultáneas en vez de la única que permite la computación clásica [Nielsen and Chuang, 2010].

Esta diferencia (que se evidencia en la representación de algoritmos cuánticos mediante máquinas de Turing cuánticas en vez de las tradicionales) presenta posibilidades poco imaginables hasta hace poco. Si se lograra construir un computador cuántico y ejecutar en él algoritmos que aprovechen su poder, se podría llegar al extremo de ejecutar tareas de complejidad no-polinomial para un ordenador tradicional (problemas NP-completos). Sin embargo, los avances científicos en el campo todavía no han llegado al punto de poder generar un prototipo práctico de computador cuántico, por lo que sus propiedades



continúan siendo más una posibilidad que un hecho [Han et al., 2001].

Además de ello, cabe resaltar que para aprovechar toda la potencia de procesamiento que un ordenador cuántico podría ofrecer, se deberá desarrollar algoritmos cuánticos que empleen los conceptos relacionados a la computación cuántica. Por ahora, se tienen algunos ejemplos como el algoritmo de Shor (para descomponer un número en factores polinomiales) [Shor, 1994] o el de Grover (para buscar una secuencia no ordenada de datos con  $N$  componentes) [Grover, 1996], entre otros. Basta ver, por ejemplo, el algoritmo de Shor, que demuestra que el poder de la computación cuántica es tal que podría acabar con el campo de la criptografía tradicional al ser capaz de romper algoritmos de encriptación como tales como RSA.

Dado que los algoritmos cuánticos “auténticos” son por ahora inaplicables en la vida real, se ha buscado adaptar distintos conceptos de la computación cuántica a otras áreas para poder aprovechar algunas de sus ventajas en la computación tradicional. De esos intentos surge la computación evolutiva de inspiración cuántica, la cual se explica posteriormente en este capítulo.

La unidad básica de un computador cuántico de dos estados es el llamado bit cuántico o *qubit*, el cual puede encontrarse en un estado ‘0’, un estado ‘1’ o en una superposición de ambos. Se puede representar este estado como:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (3.1)$$

donde  $\alpha$  y  $\beta$  son valores complejos que representan las amplitudes de probabilidad de ambos estados. Por lo tanto,  $|\alpha|^2$  es la probabilidad que el valor del qubit sea de ‘0’ al observarlo, y  $|\beta|^2$  la probabilidad correspondiente para el valor ‘1’. La normalización del estado garantiza que se cumpla:

$$|\alpha|^2 + |\beta|^2 = 1 \quad (3.2)$$

De existir un sistema de  $m$ -qubits, éste puede representar tanta información como  $2^m$  estados a la vez. Sin embargo, al observar el qubit, la superposición del mismo colapsa, asumiendo así un estado único.

## 3.2. Computación Evolutiva

La computación evolutiva es una rama dependiente de la inteligencia artificial, y a su vez, una metodología de optimización inspirada en la evolución biológica y el comportamiento de organismos vivientes [Zhang et al., 2011], y comprende en sí a los algoritmos evolutivos. La computación evolutiva incluye metodologías y variantes tales como: algoritmos genéticos, estrategias evolutivas, programación genética, sistemas de clasificación mediante aprendizaje, evolución diferencial, algoritmo para la estimación de la distribución, inteligencia de enjambre, optimización por colonia de hormigas, entre otros.

Un algoritmo evolutivo es una abstracción de procesos y principios establecidos por el Darwinismo y el Neo-Darwinismo. Por lo general comprenden las siguientes ideas: dada una población de individuos, distintas circunstancias en el ambiente ejercen presión sobre los mismos provocando el fenómeno conocido como selección natural (supervivencia del más apto), causando un incremento en las aptitudes de dichos individuos. Estas circunstancias ambientales son las encargadas de realizar la medición de la aptitud de los individuos.

Dada una función que mide esta calidad y la cual se busca maximizar, se genera aleatoriamente un conjunto de individuos candidatos (que podrían ser elementos en el dominio de la función) a los cuales se les aplica la función ya mencionada para medir y comparar la aptitud de cada individuo por separado. Basándonos en esta medición de aptitud, algunos de los individuos candidatos son escogidos para propiciar la siguiente generación mediante la recombinación y/o mutación. La recombinación es un operador que se aplica a dos o más candidatos-progenitor seleccionados y que permite intermezclar características de ambos produciendo como resultado a un individuo-hijo. La mutación, por otro lado, es la aplicación de un cambio puntual sobre un candidato resultando en uno distinto. Esta generación de nuevos individuos candidatos compite en aptitud -y en algunos casos, edad- con los individuos de la generación anterior por un lugar en la siguiente. Este proceso continua iterando hasta que se evalúa a algún individuo de calidad lo suficientemente alta (hallazgo de una solución) o que se llegue a un límite en el tiempo de computación [Eiben and Smith, 2003].

Vale la pena indicar también que muchos componentes de los algoritmos evolutivos son estocásticos debido a que, a pesar de las mayores posibilidades de supervivencia o reproducción que poseen los mejores individuos, los más débiles mantienen incluso alguna chance de hacerlo también.

---

**Algorithm 1** Pseudocódigo para un algoritmo evolutivo estándar

---

INITIALIZE *population* with random candidate solutions;

EVALUATE each candidate;

**while** NOT termination-condition **do**

    SELECT parents;

    RECOMBINE pairs of parents;

    MUTATE the resulting offspring;

    EVALUATE new candidates;

    SELECT individuals for next generation;

**end while**

---

Debido a que todas las variedades de algoritmos evolutivos siguen los lineamientos formulados tanto en el algoritmo estándar como en la Figura 3.1, las diferencias entre ellas se reducen a detalles técnicos, como la forma de representar las soluciones. Lo ideal es utilizar la representación de datos más adecuada según la naturaleza del problema a resolver. Si bien son varios los paradigmas desarrollados a partir de la computación evolutiva, son tres los más importantes: Estrategias de Evolución, Programación Evolutiva y Algoritmos Genéticos, las cuales evolucionan soluciones para problemas parametrizados. A ellas se les sumaría un cuarto paradigma: Programación Genética, el cual evoluciona los programas computacionales en sí con el fin de solucionar problemas computacionales

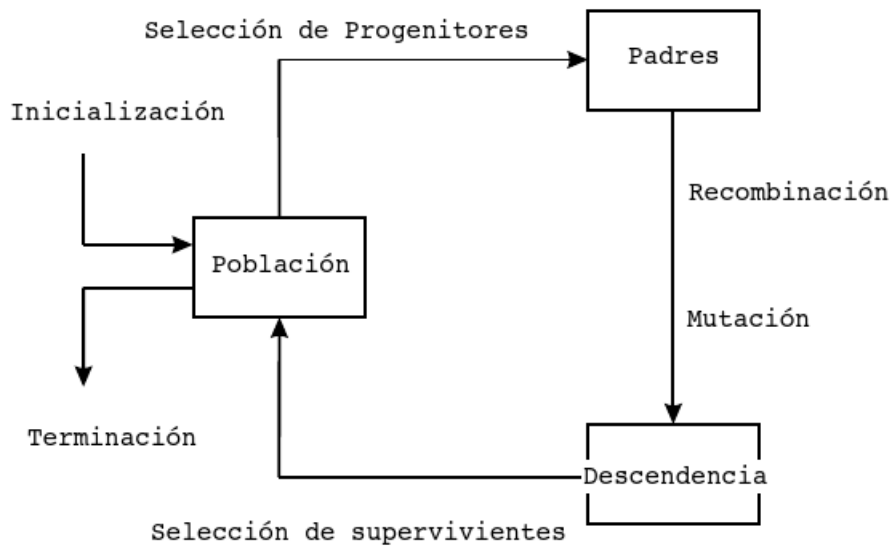


Figura 3.1: Proceso de un algoritmo evolutivo. [Eiben and Smith, 2003]

[Kicinger et al., 2005].adsfasdfads

Es necesario mencionar algunos conceptos propios de los algoritmos evolutivos, entre los cuales se encuentra al **individuo** que es una solución propuesta al problema; la **población** que es el conjunto de individuos a evaluar y evolucionar; la **generación** que es una iteración del algoritmo en el que se evalúa la aptitud de los individuos de la población para posteriormente obtener una población nueva tras realizar cambios aplicando operadores tales como *mutación* o *recombinación*; el **fenotipo** que son los rasgos observables de cada individuo; y el **genotipo** que es la codificación genética factible de convertirse en un individuo.

Los algoritmos evolutivos emplean determinadas herramientas comunes en sus distintas variables:

- Una forma de codificar las soluciones. Esta forma varía de tal forma que podemos encontrar el uso de cadenas de alfabetos finitos como en los algoritmos genéticos, el de árboles en la programación evolutiva, o de vectores con valores reales en las estrategias evolutivas, entre otros [Kicinger et al., 2005]
- Una función de aptitud que depende tanto de los individuos como de la forma de evaluarlos.
- Un mecanismo de selección, el cual se basa en la aptitud.
- Un conjunto de operadores para reproducir y alterar a los individuos codificados.

Los algoritmos evolutivos poseen gran cantidad de aplicaciones, entre las que podemos contar problemas de optimización [Coello, 1999] [Zhou et al., 2011], exploración (arte evolutivo) [Romero and Machado, 2008], optimización de procesos químicos [Singulani et al., 2008], entre muchos otros. Presentan gran cantidad de ventajas, entre las cuales podemos mencionar:

- Aplicabilidad en problemas donde no hay otros métodos disponibles, ya sea por presencia de restricciones no lineales, discontinuidad, multi-modalidad, problemas de ruido, etc.
- Adecuados para problemas que requieren múltiples soluciones, debido a la existencia de una población de las mismas.
- Altamente paralelizables.

Por supuesto, los algoritmos evolutivos también presentan algunos inconvenientes, tales como:

- Los efectos que los errores del usuario pueden producir al momento de ajustar parámetros, los cuales pueden resultar en errores o en un desempeño menor que óptimo [Hinterding et al., 1997].
- El ajuste de los parámetros puede tomar tiempo.
- El valor óptimo de los parámetros pueden variar durante la evolución.
- No existe una garantía para hallar soluciones óptimas en un periodo de tiempo determinado, aunque para evitar ello se pueden aplicar pruebas de convergencia asintótica.

### 3.3. Algoritmos Evolutivos de Inspiración Cuántica

Los algoritmos evolutivos de inspiración cuántica buscan aprovechar los paradigmas de la física cuántica para mejorar su rendimiento al resolver problemas. El algoritmo QIEA original fue descrito originalmente en [Han and Kim, 2000] y posteriormente analizado [Han et al., 2001]. Como se puede ver en el capítulo anterior, han surgido buen número de variaciones y aplicaciones, todos basados en el original descrito a continuación. Para la formulación de un algoritmo de inspiración cuántica se debe cumplir con los siguientes requerimientos:

- Tener una representación numérica o un método para convertir en representación numérica.
- Determinar una configuración inicial.
- Definir una condición de finalización.
- Dividir el problema en sub-problemas más simples.
- Identificar el número de universos (estados de superposición).
- Cada sub-problema debe asociarse a un universo.
- Los cálculos deben ser independientes en cada universo.

- Debe haber alguna interacción entre universos, y ésta debe, al menos permitir hallar la solución, o ayudar a que cada sub-problema en cada universo sea capaz de encontrarla.

QIEA emplea una representación denominada Q-bit -la cual equivale al qubit que puede representar ya sea a uno de los valores  $\{0, 1\}$  o a la superposición de ambos-, el cual está definida por dos valores  $[\alpha, \beta]$

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (3.3)$$

donde  $|\alpha|^2 + |\beta|^2 = 1$ , y que hacen referencia a la probabilidad que el Q-bit colapse -ya sea a uno o a otro- de los dos estados posibles al momento de ser observado. De la misma forma se puede definir un vector de  $n$  Q-bits de esta forma:

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \beta_1 & \beta_2 & \dots & \beta_n \end{bmatrix} \quad (3.4)$$

donde también se cumple  $|\alpha_j|^2 + |\beta_j|^2 = 1 \forall j = 0, 1, \dots, n$ .

Para modificar los Q-bits se emplea un operador denominado Q-gate, el cual se encarga de rotar el ángulo y hacerlo inclinarse hacia un valor  $\{0, 1\}$ , de tal manera que asegure el cumplimiento de la ecuación 3.2:

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} \quad (3.5)$$

donde  $\Delta\theta_i, i = 0, 1, \dots, m$  es el ángulo de rotación de cada Q-bit. La magnitud de este valor tiene efecto en la convergencia, pero de ser muy alto se corre el riesgo de diverger o de converger prematuramente hacia un óptimo local.

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = U(\Delta\theta_i) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \quad (3.6)$$

sin perder la característica  $|\alpha'_k|^2 + |\beta'_k|^2 = 1$ .

En la figura 3.2 se puede apreciar cómo las modificaciones en el operador Q-gate permiten que el las probabilidades  $\alpha$  y  $\beta$  varíen sin perjudicar la condición elemental expresada en la ecuación 3.2.

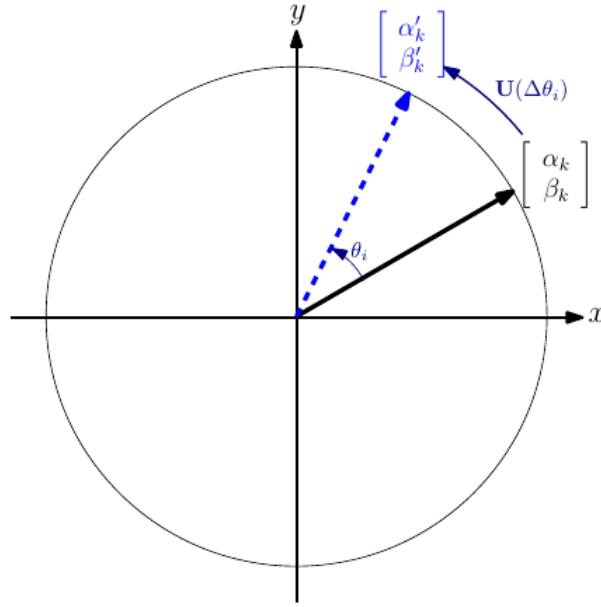


Figura 3.2: Rotación del ángulo al momento de la aplicación del operador Q-gate

### 3.4. Algoritmo Evolutivo de Inspiración Cuántica - $\mathbb{R}$ (QIEA- $\mathbb{R}$ )

La propuesta original de un QIEA para  $\mathbb{R}$  busca generar un conjunto de estados observables continuos y no discretos como el algoritmo base [da Cruz, 2007]. La inspiración de este modelo está en el uso de funciones de onda.

Se calcula el momento de un fotón a partir de su masa al moverse a la velocidad de la luz  $m = hv/c^2$  derivada de la ecuación de equivalencia entre masa y energía  $E = mc^2$ .

$$p = mc = hv/c = h/\lambda \quad (3.7)$$

donde  $\lambda$  es la longitud de onda de la luz.

Al reemplazar  $\lambda = h/p$  en la ecuación utilizada en electromagnetismo donde una onda estacionaria con longitud de onda  $\lambda$  se propaga hacia el lado positivo del eje  $x$  definida como  $\psi(x) = e^{i2\pi x/\lambda}$  donde  $i = \sqrt{-1}$  se obtiene:

$$\psi(x) = e^{ipx/\hbar} \quad (3.8)$$

donde  $\hbar = h/2\pi$ .

Según [Gillespie, 1974], la Densidad Determinada Probabilística (PDD) para la ubicación de una partícula con función de onda  $\psi$  está definida por  $|\psi|^2$ , por lo que al integrar todo el espacio en el que la partícula podría ser encontrada, se obtiene la posibilidad de encontrarla en cualquier lugar del espacio:

**Algorithm 2** Pseudocódigo para el QIEA- $\mathbb{B}$ 


---

```

INITIALIZE  $t \leftarrow 0$ ;
INITIALIZE  $\mathbf{Q}_t$ ;
MAKE  $\mathbf{P}_t$  by observing the states of  $\mathbf{Q}_t$ ;
STORE best solutions among  $\mathbf{P}_t$  into  $\mathbf{B}_t$ ;
while EndCondition = false do
     $t \leftarrow t + 1$ ;
    MAKE  $\mathbf{P}_t$  by observing the states of  $\mathbf{Q}_{t-1}$ ;
    EVALUATE  $\mathbf{P}_t$ ;
    UPDATE  $\mathbf{Q}_t$  using Q-gates;
    STORE best solutions among  $\mathbf{B}_{t-1}$  and  $\mathbf{P}_t$ ; into  $\mathbf{B}_t$ ;
    STORE best solution  $\mathbf{b} \in \mathbf{B}_t$ ;
    if Migration Condition then
        MIGRATE  $\mathbf{b} | \mathbf{b}'_j \leftarrow \mathbf{B}_t$ , locally or globally;
    end if
end while

```

---

$$\int_{-\infty}^{\infty} |\psi|^2 d\tau = 1 \quad (3.9)$$

El concepto de función de onda relaciona probabilísticamente una onda con la localización de una partícula. Como en cada observación de una partícula, ésta asumirá diferentes valores de posición según la probabilidad que tiene de estar ubicada en una determinada región del espacio, se usa este concepto en los modelos de algoritmos evolutivos de inspiración cuántica para  $\mathbb{R}$  al momento de representar los valores de los individuos cuánticos.

**Algorithm 3** Pseudocódigo para QIEA- $\mathbb{R}$ 


---

```

INITIALIZE  $t \leftarrow 0$ ;
INITIALIZE  $\mathbf{Q}_t$  with  $m$  individuals of  $n$  genes;
while  $t \leq T$  do
    GENERATE  $\mathbf{E}_t$  by observing individuals  $\mathbf{Q}_t$ 
    if  $t = 1$  then
         $\mathbf{C}_t = \mathbf{E}_t$ 
    else
        RECOMBINE  $(\mathbf{E}_t, \mathbf{C}_t) \rightarrow \mathbf{E}_t$ 
        EVALUATE  $\mathbf{E}_t$ 
        SELECT  $\mathbf{C}_t \leftarrow k$  best individuals from  $\mathbf{E}_t \cup \mathbf{C}_t$ 
    end if
    UPDATE  $\mathbf{Q}_{t+1}$  with  $m$  best individuals from  $\mathbf{C}_t$ 
     $t = t + 1$ 
end while

```

---

De tal forma, se obtiene una población de individuos cuánticos  $\mathbf{Q}_t = \{q_1, \dots, q_m\}$ , en la generación  $t$ , donde cada individuo cuántico  $q_i$  está formado por  $n$  genes,  $q_{ij} = \{q_{i1}, \dots, q_{in}\}$ , donde a su vez cada gen  $q_{ij}$  está formado por funciones densidades de proba-

bilidad (en vez de variables como en el algoritmo original). Es por esto que se puede definir a cada gen cuántico  $q_{ij}$  perteneciente a un individuo cuántico como una variable aleatoria con función de densidad probabilística  $p_{ij}(x) = \psi *_{ij}(x)\psi(x)$  donde  $\psi_{ij}(x)$  es la función de onda asociada a  $q_{ij}$  del individuo  $q_i$  de la población  $Q_t$  y donde  $\psi *_{ij}(x)$  es el conjugado completo de la función de onda  $\psi_{ij}(x)$ , donde la Función Densidad Probabilística (PDF) debe cumplir la propiedad de normalización  $\int_{-\infty}^{\infty} \psi *_{ij}(x)\psi(x)dx = \int_{-\infty}^{\infty} p_{ij}(x)dx = 1$ . No olvidar también que la PDF ha de ser integrable en la región del dominio de las variables a ser optimizadas para garantizar que se pueda buscar valores en todo el espacio de búsqueda  $\mathbb{X}$ :

Por lo cual, un individuo cuántico puede ser representado de esta manera:

$$q_i = \{q_{i1} = p_{i1}(x), q_{i2} = p_{i2}(x), \dots, q_{in} = p_{in}(x)\} \quad (3.10)$$

Se puede utilizar una distribución uniforme  $U_{ij}(x) \in [l_{ij}, u_{ij}]$  como PDF al definirla como:

$$p_{ij}(x) = \begin{cases} \frac{1}{u_{ij} - l_{ij}} & \text{if } l_{ij} \leq x \leq u_{ij} \\ 0 & \text{else} \end{cases} \quad (3.11)$$

donde  $u_{ij}$  es el límite superior y  $l_{ij}$  el límite inferior del intervalo para el gen cuántico  $q_{ij}$  al colapsar su superposición de estados al ser observado. La definición  $U_{ij}(x)$  respeta la normalización requerida para la PDF y es fácilmente implementable con una función que genere número aleatorios escalada al intervalo  $[l_{ij}, u_{ij}]$  tal como  $U_{ij}(x) = l_{ij} + (u_{ij} - l_{ij})U(x)$  donde  $U(x)$  es un generador de números aleatorios reales en  $[0, 1]$ .

Cada individuo cuántico  $q_i$  perteneciente a una población cuántica  $Q_0$  poseería entonces  $n$  genes cuántico  $q_{ij}$ , cada cual representado de dos formas posibles: ya sea por los valores  $l_{ij}, u_{ij}$  o por el centro  $\mu_{ij} = \frac{l_{ij} + u_{ij}}{2}$  mas el ancho de pulso  $\sigma_{ij} = u_{ij} - l_{ij}$ . Ambas representaciones son equivalentes, por lo que un individuo cuántico  $q_i = \{q_{i1}, q_{i2}\}$  con pulsos de ancho 2 y centros posicionados en  $-0.5$  y  $0.5$  sería representado por  $q_i = \{\mu_{i1} = -0.5, \mu_{i2} = 0.5, \sigma_{i1} = 2, \sigma_{i2} = 2\}$ .

### 3.5. Consideraciones Finales

Según se ha visto, el algoritmo QIEA original es sencillo y de implementación relativamente rápida, mientras que el algoritmo para  $\mathbb{R}$  sube ligeramente el listón en términos de inspiración pero añadiendo nuevas capacidades a las posibilidades de la inspiración cuántica. Se tomará este algoritmo como punto de partida para la propuesta que se desarrolla en el siguiente capítulo, el cual busca introducir un nuevo mecanismo de recombinación (*crossover*) en un entorno concurrente de ejecución con más de una población cuántica generadora.



## Capítulo 4

# Propuesta

Como ya fue mencionado en su momento en la introducción de la presente tesis, se busca evaluar el desempeño de un algoritmo QIEA para  $\mathbb{R}$  que presente las características de segregación/interacción entre universos de operación para cada individuo cuántico en operación con respecto al algoritmo original, con el fin de obtener mayor control sobre el grado de localidad y globalidad de la actualización de cada individuo cuántico mediante la segregación del campo operativo de los mismos, así como la incorporación de un operador de cruzamiento que funcione a este nivel.

Se escogió trabajar sobre el algoritmo para  $\mathbb{R}$  debido a la menor cantidad de investigación realizada sobre el mismo a diferencia de la versión original orientada a  $\mathbb{B}$ . Además, cabe considerar la relativa simplicidad de su evaluación, la cual se hará sobre funciones *benchmark* -se verá con más detalle este apartado en el siguiente capítulo-.

A continuación, se describirá con detalle las características a ser tomadas en consideración para la descripción del algoritmo a evaluar.

### 4.1. Segregación del campo operativo de los individuos cuánticos en universos

Como se vió al principio de la sección 3.3 del presente documento, hay algunas condiciones especiales sobre las cuales se pueden proponer algoritmos evolutivos de inspiración cuántica. Se identifica la necesidad de una subdivisión de problemas en universos, los cuales se requiere que sean independientes de los demás. El algoritmo original ejecuta una evaluación conjunta de los distintos individuos clásicos generados, por lo que no satisface esta condición. Incluso si se procede con el particionamiento del espacio de búsqueda, sólo se llegaría a satisfacer parcialmente, porque aún así los individuos clásicos generados permanecen evaluados en forma conjunta y no segregada.

Es por ello que para este caso, se propone de forma inicial la segregación completa de los campos de operación de cada individuo cuántico en universos con nula interacción a nivel de individuos cuánticos entre sí. Ya sea que se particione o no el espacio de

búsqueda, cada universo evaluará sus respectivas generaciones de individuos clásicos de forma independiente.

## 4.2. Operador de cruzamiento para los individuos cuánticos

Si, como se vio en el punto anterior, se limita la capacidad de interacción a nivel de individuos clásicos entre individuos sin ofrecer otra forma de interacción entre universos, se volvería a un escenario en el que se incumplen las condiciones para la propuesta de algoritmos evolutivos de inspiración cuántica mostrados en la tercera sección del anterior capítulo. Esta vez, a diferencia del algoritmo original, el énfasis estaría en la nulidad de la interacción entre universos.

Ya se ha hecho notar con anterioridad en la introducción que el énfasis del cruzamiento en los algoritmos hasta la actualidad sigue recayendo en los individuos clásicos. Por lo tanto, la forma mas lógica de cumplir esta característica sin entrar en conflicto con la condición descrita en la sección anterior, sería introducir un operador de cruzamiento que trabaje directamente entre individuos cuánticos.

Existen diversos modelos de cruzamiento a nivel de individuos clásicos, los cuales ya fueron descritos en el capítulo anterior. Para el presente trabajo se ha elegido trabajar con un operador elitista que, para cada iteración, seleccione tanto al mejor como al peor elemento cuántico. Las características del mejor operador cuántico (que en realidad refiere al espacio de búsqueda del mismo) serán agregados a las del peor elemento cuántico. De esta manera, se busca incrementar la capacidad de generación de individuos clásicos de calidad para los individuos cuánticos de desempeño más discreto.

Como es evidente, no es posible evaluar directamente a un individuo cuántico mediante una función *fitness* como si se hace con los individuos clásicos. Sin embargo, si es posible evaluar a los individuos clásicos generados por cada individuo cuántico. De esta forma, a cada individuo cuántico se le evaluaría la calidad de la generación de individuos clásicos que respondan bien ante las circunstancias a evaluar.

## 4.3. Detalle y nomenclatura de los algoritmos a evaluar

Para efectos de nomenclatura, se ha decidido utilizar el nombre QIEAR para denominar al algoritmo original sobre el cual se realizan las modificaciones ya explicadas. De acuerdo a lo ya expuesto en secciones anteriores, el espacio de búsqueda puede o no estar particionado, por lo que para indicar la existencia de un particionamiento en el mismo, se usará el sufijo  $-p$ .

De la misma manera, se utilizará el prefijo  $U$  para indicar que los espacios de búsqueda para cada individuo cuántico están siendo segregados. Finalmente, el sufijo  $CO$  será

empleado para indicar la implementación del operador de cruzamiento entre individuos cuánticos.

A partir de esto, se evidencia la existencia de dos algoritmos originales y 7 modificaciones al mismo de acuerdo a la incorporación de las dos características novedosas ya detalladas. Una explicación más clara de esto se puede ver en la Tabla 4.1.

Algoritmo	Espacio de búsqueda particionado	Campo de acción de individuos cuánticos segregado en universos	Operador de cruzamiento para individuos cuánticos
QIEAR	No	No	No
UQIEAR	No	Si	No
QIEAR-CO	No	No	Si
UQIEAR-CO	No	Si	Si
QIEAR-p	Si	No	No
UQIEAR-p	Si	Si	No
QIEAR-pCO	Si	No	Si
UQIEAR-pCO	Si	Si	Si

Tabla 4.1: Descripción de los algoritmos a evaluar

## Capítulo 5

# Pruebas y Resultados

### 5.1. Escenarios de prueba

Como se indicó desde un principio, los escenarios para las pruebas son un conjunto de funciones *benchmark* orientadas a problemas de optimización de objetivo único especialmente seleccionadas para evaluar el desempeño de los algoritmos derivados comparativamente con respecto al algoritmo original. Las funciones fueron seleccionadas a partir de [Mishra, 2006] [Silagadze, 2007].

Función	Fórmula	Valor mínimo	Dominio de búsqueda
Arckley	$f(x, y) = -20e^{-0.2\sqrt{0.5(x^2+y^2)}} - e^{0.5(\cos(2\pi x) + \cos(2\pi y))} + e + 20$	$f(0, 0) = 0$	$-5 \leq x, y \leq 5$
Booth	$f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$	$f(1, 3) = 0$	$-10 \leq x, y \leq 10$
Maytas	$f(x, y) = 0.26(x^2 + y^2) - 0.48xy$	$f(0, 0) = 0$	$-10 \leq x, y \leq 10$

Tabla 5.1: Descripción de las funciones de prueba

Todas las funciones poseen en común el evaluar dos variables. Además, otras condiciones que debieron cumplir para ser seleccionadas como escenarios de prueba son:

- Poseer espacios de búsqueda reducidos.
- Poseer mínimos cuya vecindad representen un porcentaje reducido del total del conjunto de soluciones en el dominio de búsqueda.
- Evitar la tendencia a valores mínimos por debajo del mínimo identificado dentro del dominio de búsqueda.

Estas condiciones se explican por la tendencia que presentan tanto el algoritmo QIEA- $\mathbb{R}$  como las distintas derivaciones del mismo que se han evaluado: la tendencia a aumentar desproporcionadamente el espacio de búsqueda conforme se deja de generar mejores individuos. Si se comienza con un espacio de búsqueda amplio, su crecimiento prematuro podría no devolver los mejores resultados; por el contrario, la calidad de las soluciones comienza a decrecer con gran velocidad.

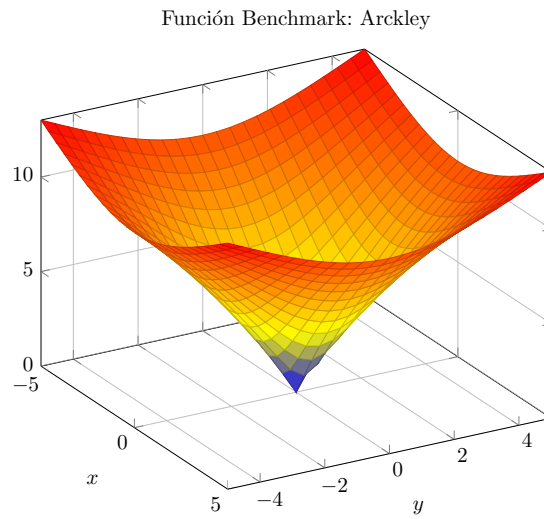


Figura 5.1:  $f(x, y) = -20e^{-0.2\sqrt{0.5(x^2+y^2)}} - e^{0.5(\cos(2\pi x) + \cos(2\pi y))} + e + 20$

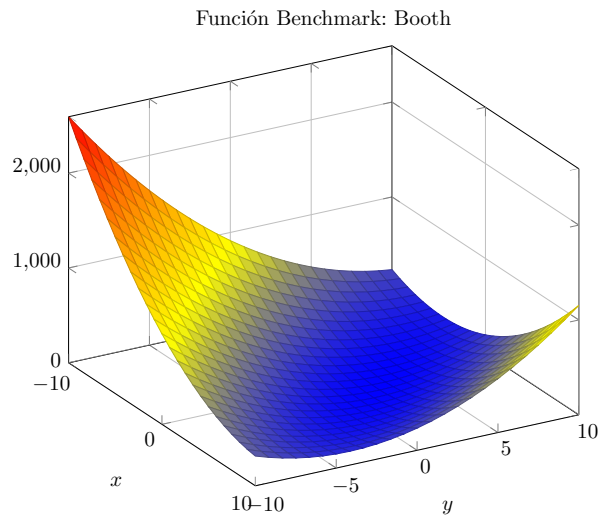


Figura 5.2:  $f(x, y) = (x + 2y - 7)^2 + (2x + y - 5)^2$

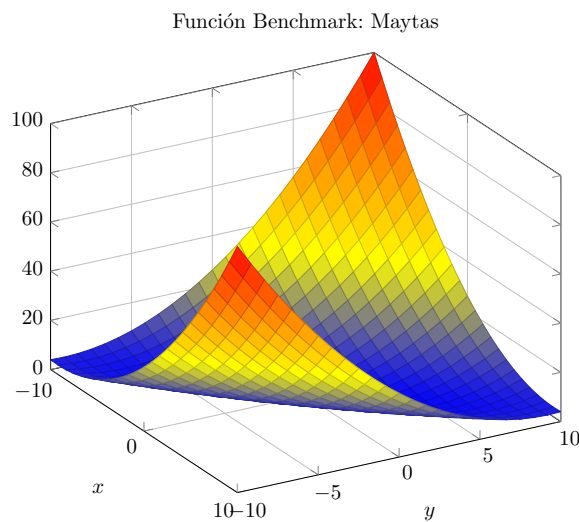


Figura 5.3:  $f(x, y) = 0.26(x^2 + y^2) - 0.48xy$

## 5.2. Condiciones para las pruebas

Las pruebas fueron ejecutadas en un ordenador Acer Aspire 4552 con procesador AMD Phenom™ II N830 Triple-Core Processor y 6 GB RAM DDR3. Todos los algoritmos ya indicados fueron evaluados bajo las siguientes condiciones:

- Un espacio de búsqueda  $p$  delimitado por  $l$  y  $u$  el cual puede ser único o presentarse particionado.
- Un número de particiones  $m$  donde  $2 \leq m \leq 10$ .
- Una población cuántica  $Q_t$  donde  $Q_t = \{q_1, \dots, q_m\}$  en el que cada  $q_i$  representa a un individuo cuántico y donde  $m$  se corresponde con el número de particiones del espacio de búsqueda. En caso que el espacio de búsqueda no esté particionado,  $m$  presenta los mismos límites que el número de particiones.
- Por cada generación, una población clásica  $C_{ij}$  donde  $C_{ij} = \{c_{i1}, \dots, c_{in}\}$  donde cada  $c_{ij}$  representa a un individuo clásico generado por el individuo  $q_i$ , y donde  $1 \leq j \leq 10$

Los algoritmos que implementan el operador de cruzamiento para los individuos clásicos, además, poseen un limitante en el número de iteración tope en las que éste ocurre, y el cual está denotado por  $m - 1$ , lo que significa que a partir de la iteración número  $m$  dicho operador deja de funcionar. Este tope es incluido debido a la sobreexpansión de los propios espacios de búsqueda cuando el operador es ejecutado más de unas pocas veces sobre un mismo individuo cuántico.

Finalmente, es pertinente indicar que la condición de parada para cada una de las pruebas es el número de iteraciones, el cual está fijado en 100 debido a que tras diversas evaluaciones, se comprobó que la evolución a partir del mismo es prácticamente nula debido a la rápida convergencia inherente al algoritmo. Cada algoritmo fue ejecutado un total de 50 veces para cada combinación de condiciones.

## 5.3. Resultados

Para un mejor análisis de los resultados, corresponde explicar los dos tipos de gráficos incluidos a continuación. El primer tipo incluye un promedio ponderado de los  $2m$ -ésimos mejores valores *fitness* para un valor  $n$  de individuos clásicos generados por individuo cuántico por iteración. De esta manera es posible evaluar cuál algoritmo es el que está produciendo mejores valores en promedio y cuál ejerce mayor presión evolutiva sobre los mejores individuos almacenados.

El segundo, por el contrario, muestra la evolución del mejor valor *fitness* por generación para cada algoritmo, el cual se ejecuta a lo largo de las iteraciones del programa (determinado en 100 iteraciones por ejecución, como ya fue indicado en la sección anterior). Acá se busca identificar la velocidad de convergencia de los datos hacia el óptimo deseado.

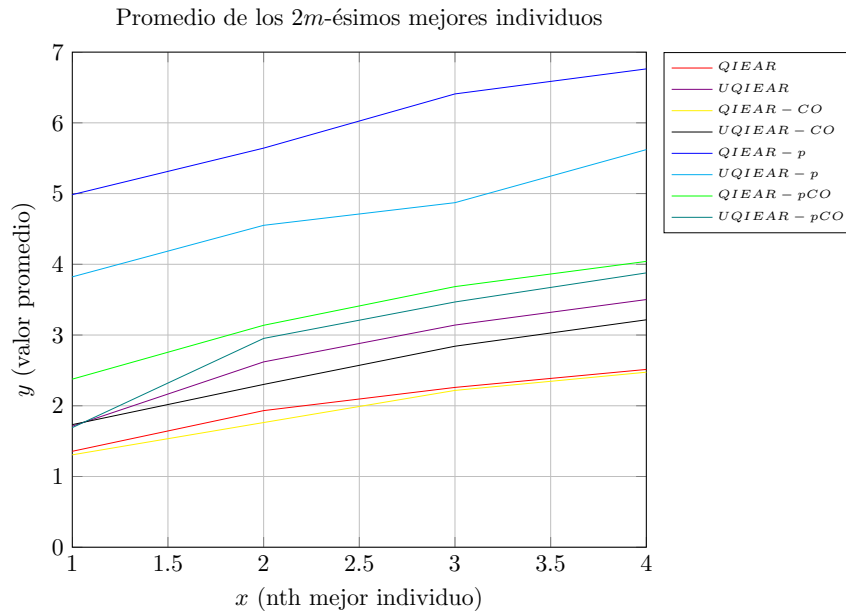


Figura 5.4: Valor promedio de los  $n$ -ésimos mejores individuos para  $n = 2$  - Función Arkley

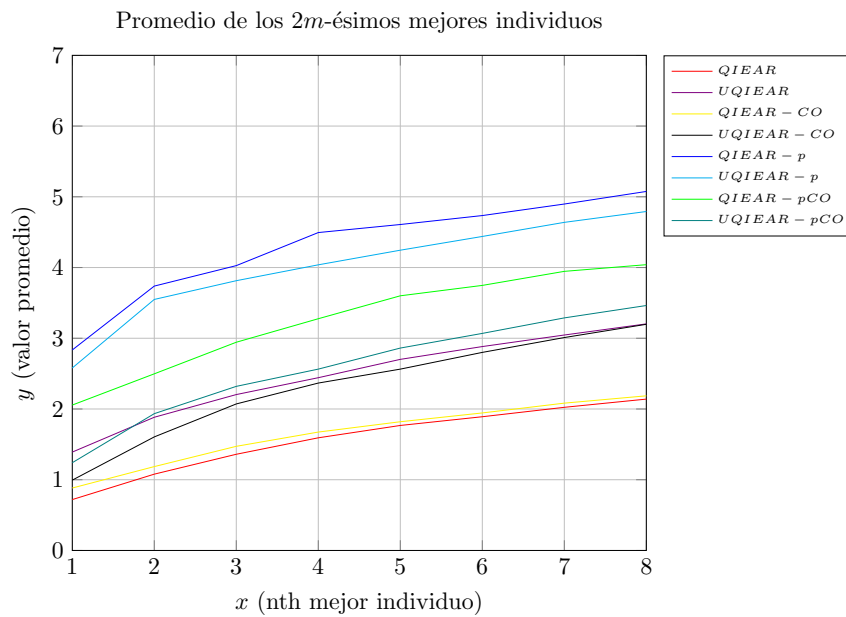


Figura 5.5: Valor promedio de los  $n$ -ésimos mejores individuos para  $n = 4$  - Función Arkley

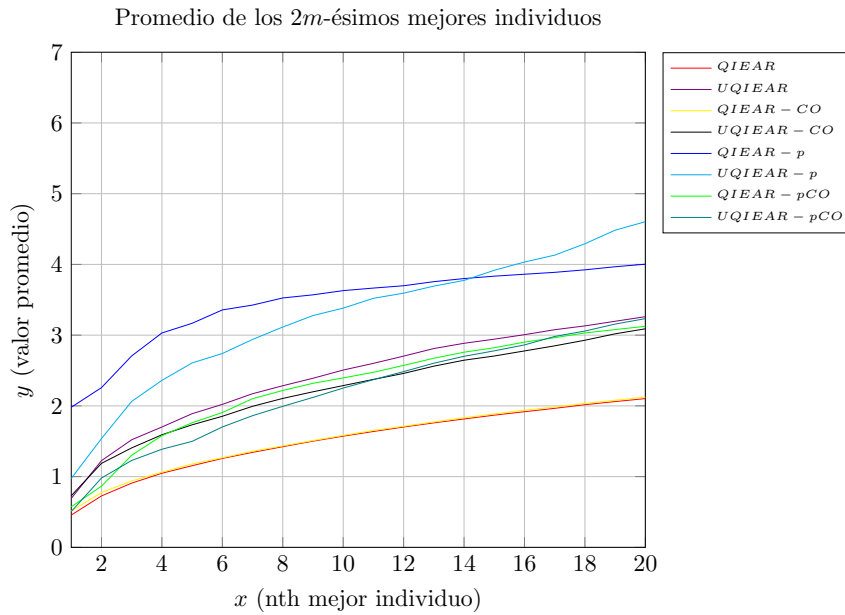


Figura 5.6: Valor promedio de los  $n$ -ésimos mejores individuos para  $n = 10$  - Función Arckley

### 5.3.1. Resultados sobre la función Arckley

De acuerdo a lo que se aprecia en las Figuras 5.4, 5.5 y 5.6, se ve que en promedio, los mejores resultados fueron producidos por el algoritmo original QIEAR, así como por QIEAR-CO, ambos algoritmos sin particionar. Por el contrario, los peores resultados fueron producidos sobre todo por el algoritmo original QIEAR-p y por la variante segregada UQIEAR-p, ambos particionados. En general, la calidad de datos producida por los algoritmos particionados fue inferior a los que ocupaban el dominio entero. Los demás algoritmos obtuvieron resultados intermedios.

En cambio, en las pruebas de velocidad de convergencia (Figuras 5.7, 5.8 y 5.9), se puede ver que el algoritmo QIEAR-pCO supera a los demás tanto en velocidad de convergencia como calidad final de resultados, siendo seguido a corta distancia por QIEAR-p, UQIEAR-pCO y UQIEAR-p. Todos estos algoritmos son particionados. Los algoritmos con peores resultados en convergencia son las modificaciones UQIEAR y UQIEAR-CO. El algoritmo original QIEAR converge mediocrementemente con un bajo número de individuos clásicos generados por individuo, pero al incrementarse éste, los resultados mejoran notablemente pero solamente gracias a una convergencia relativamente tardía en comparación a los mejores en esta prueba.

### 5.3.2. Resultados sobre la función Booth

En la primera prueba para esta función (resultados visibles en las Figuras 5.10, 5.11 y 5.12), se puede ver cómo los algoritmos segregados en promedio se comportan de forma más mediocre que los algoritmos sin segregar. Si bien para  $n = 3$ , los algoritmos originales QIEAR y QIEAR-p muestran mejor promedio de resultados, conforme  $n$  aumenta, los algoritmos con el operador de cruzamiento mejoran en la calidad de datos



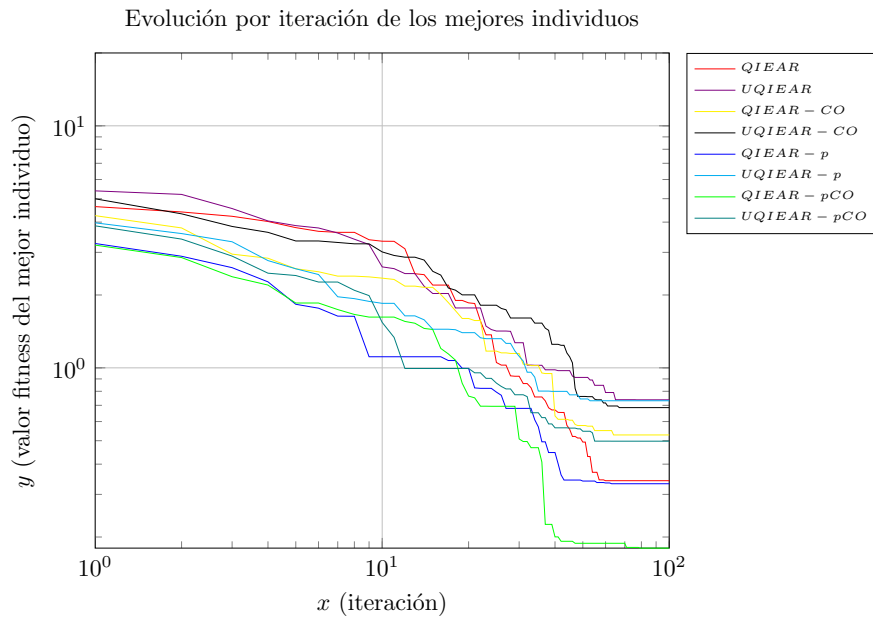


Figura 5.7: Evolución de los valores de los mejores individuos para  $n = 3$  - Función Arkley

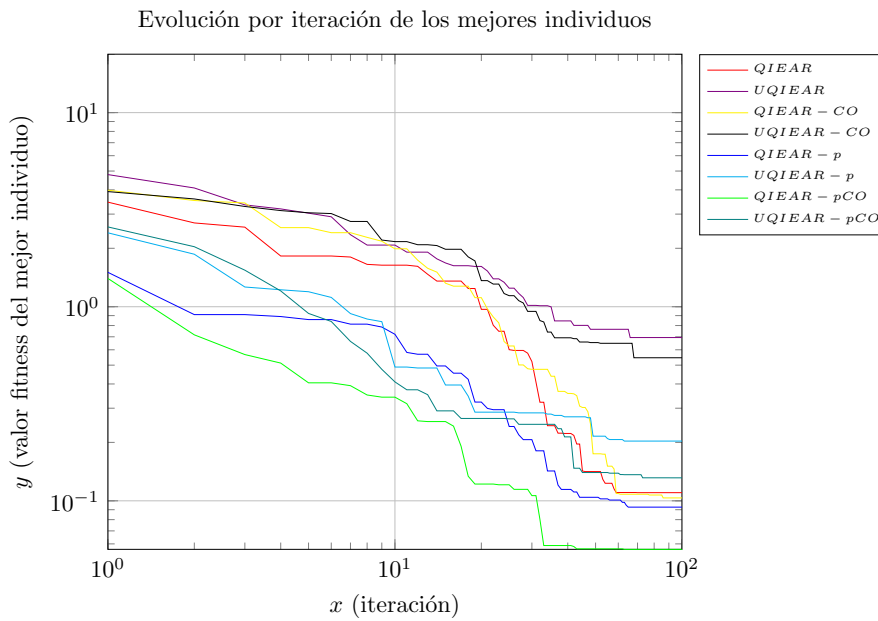


Figura 5.8: Evolución de los valores de los mejores individuos para  $n = 7$  - Función Arkley

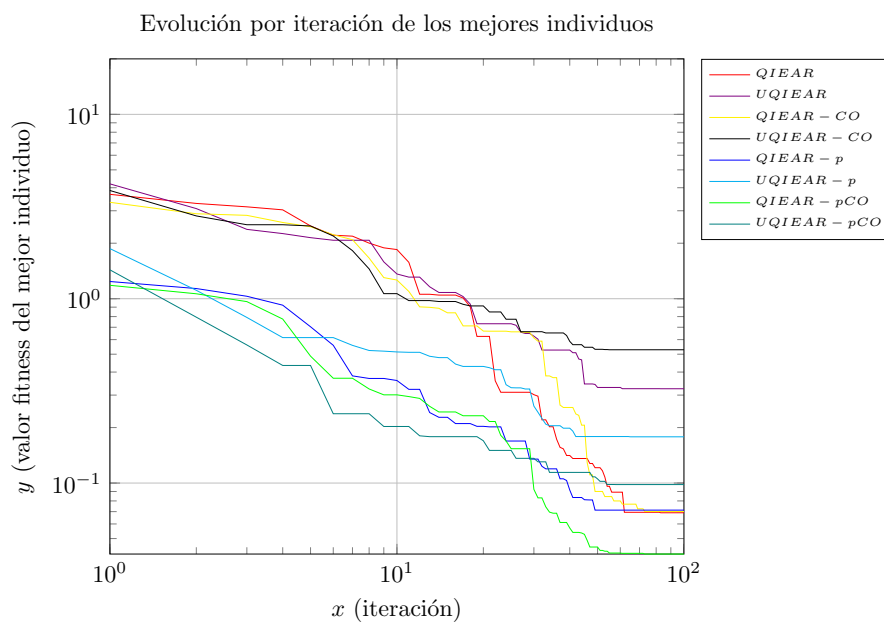


Figura 5.9: Evolución de los valores de los mejores individuos para  $n = 9$  - Función Ackley

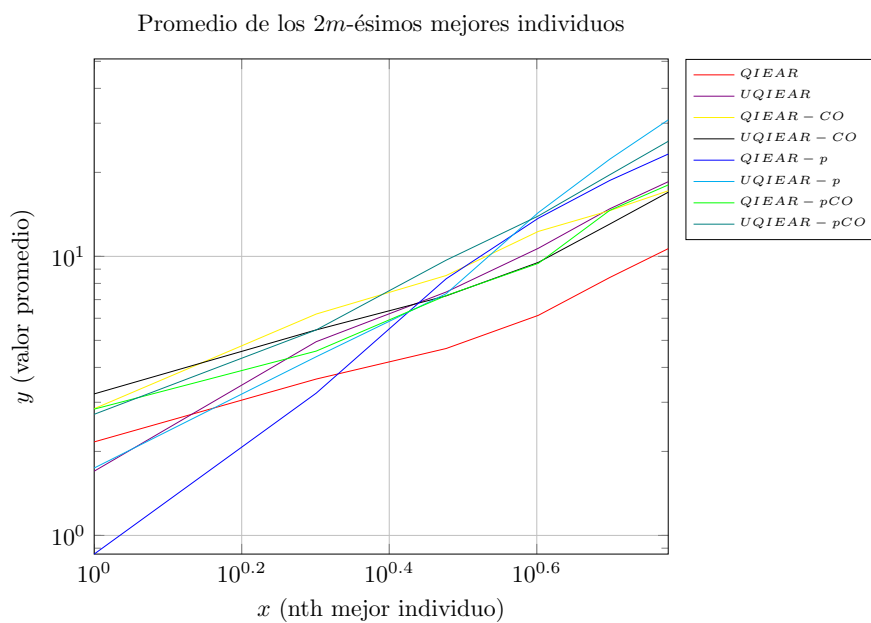


Figura 5.10: Valor promedio de los  $n$ -ésimos mejores individuos para  $n = 3$  - Función Booth

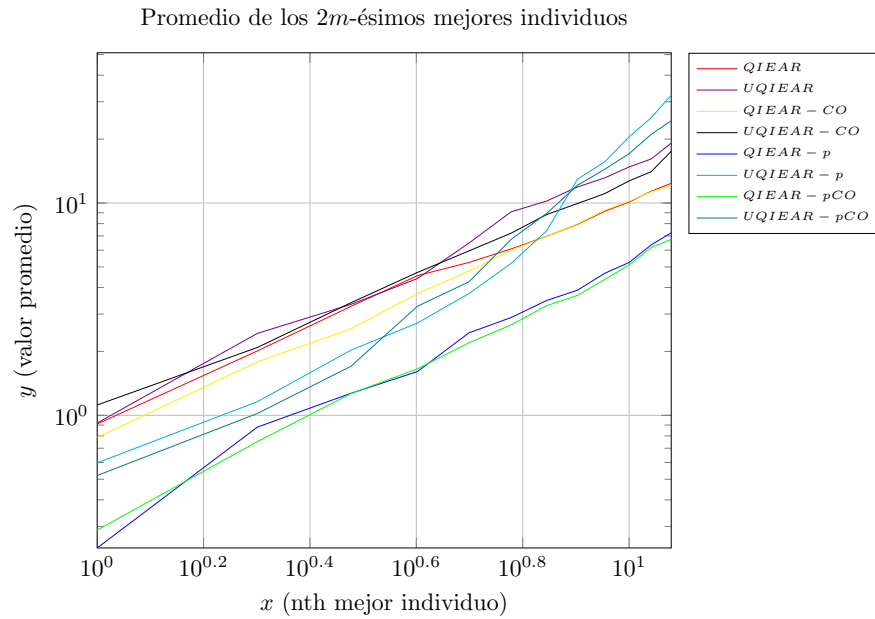


Figura 5.11: Valor promedio de los  $n$ -ésimos mejores individuos para  $n = 6$  - Función Booth

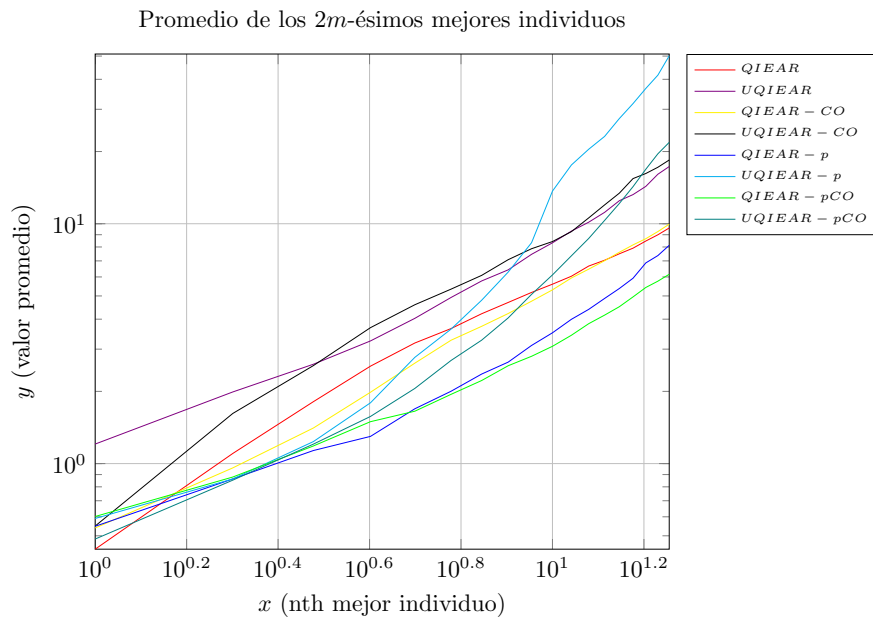


Figura 5.12: Valor promedio de los  $n$ -ésimos mejores individuos para  $n = 9$  - Función Booth

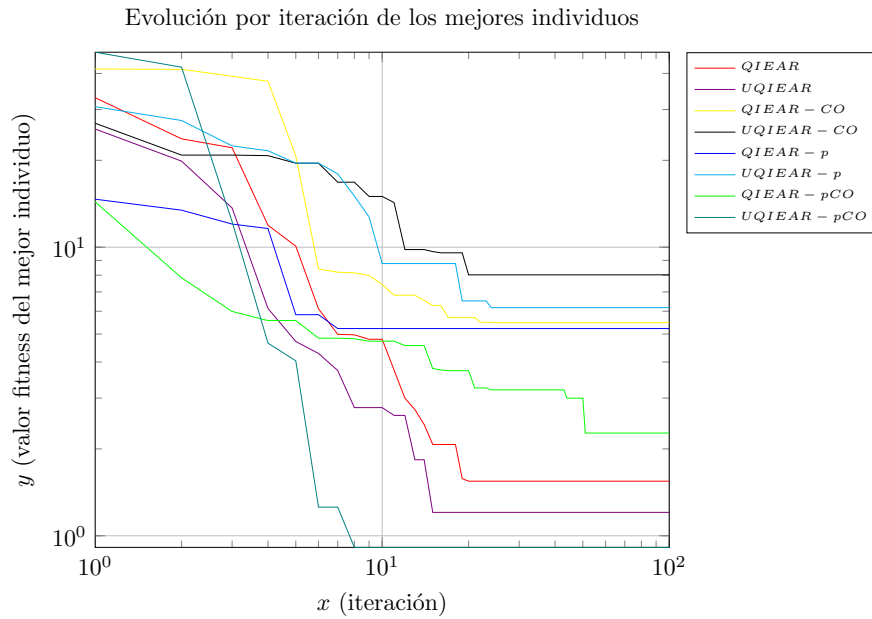


Figura 5.13: Evolución de los valores de los mejores individuos para  $n = 2$  - Función Booth (QIEAR-pCO y QIEAR-CO, en ese orden).

Para las pruebas de convergencia (Figuras 5.13, 5.14 y 5.15), sin embargo, es el algoritmo UQIEAR-pCO seguido de QIEAR-pCO los que presentan mejor balance entre convergencia y calidad de datos. Aunque el algoritmo original QIEAR haya sido el mejor para  $n = 9$ , cabe destacar que fue el peor para  $n = 4$ , con lo que su desempeño fue irregular en esta prueba. En general, el balance de los algoritmos que implementan el operador de cruzamiento es competitivo en comparación con los demás.

### 5.3.3. Resultados sobre la función Maytas

En las pruebas de valores promedio (Figuras 5.16, 5.17 y 5.18), destaca los buenos resultados que obtiene la variante QIEAR-CO, la cual supera los algoritmos UQIEAR-pCO y QIEAR-p que son los evidencian mejor promedio de datos generados en comparación con los demás. Se puede afirmar con certeza que los algoritmos que particionan su espacio de búsqueda logran comportarse mejor que los no particionados.

Finalmente, en las pruebas de convergencia (Figuras 5.19, 5.20 y 5.21) se evidencia una contraposición de resultados separada por el particionamiento del espacio de búsqueda. A pesar que los algoritmos que implementan espacios particionados de búsqueda obtienen una convergencia más rápida, al final evidencian una menor cercanía al mínimo deseado. En todo caso, el algoritmo que ofrece el mejor desempeño promedio para ambos casos son tanto el algoritmo base QIEAR como UQIEAR-pCO debido a figurar por encima del promedio tanto en rapidez de convergencia como en producción de datos cercanos al mínimo.

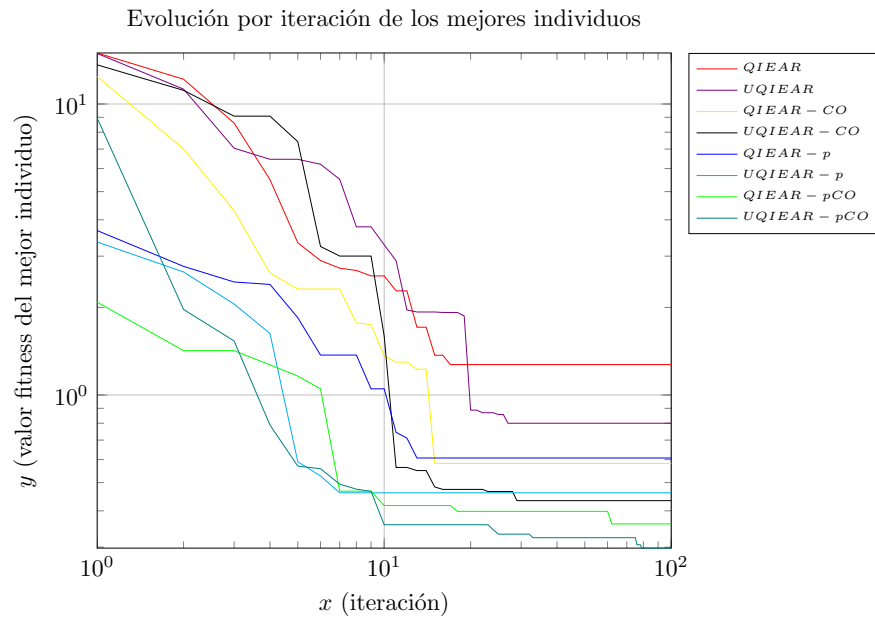


Figura 5.14: Evolución de los valores de los mejores individuos para  $n = 4$  - Función Booth

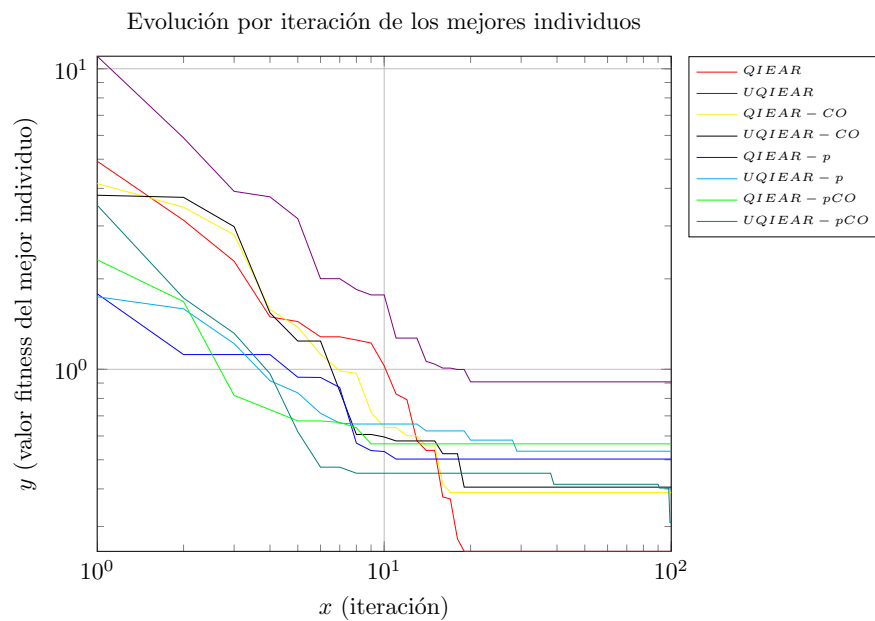


Figura 5.15: Evolución de los valores de los mejores individuos para  $n = 9$  - Función Booth

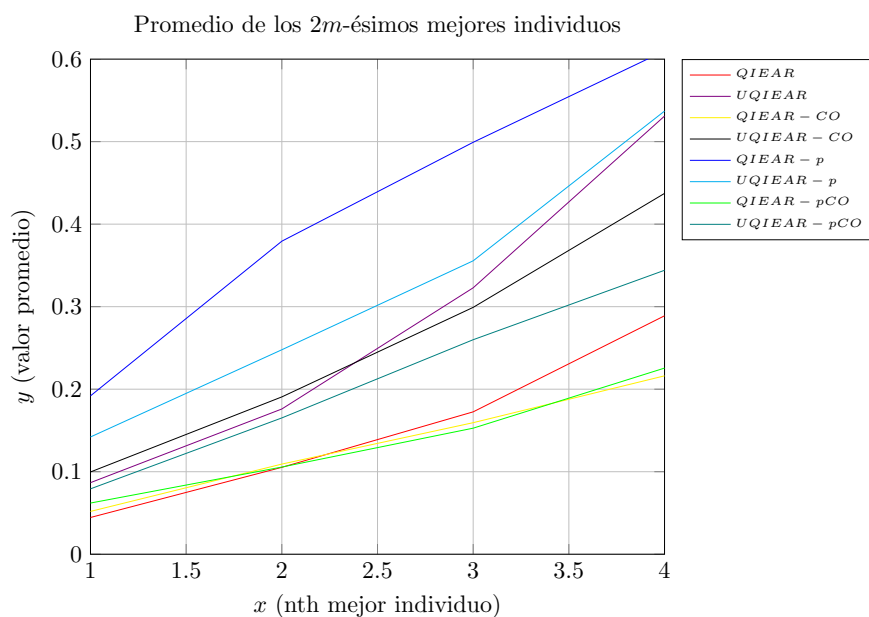


Figura 5.16: Valor promedio de los  $n$ -ésimos mejores individuos para  $n = 2$  - Función Maytas

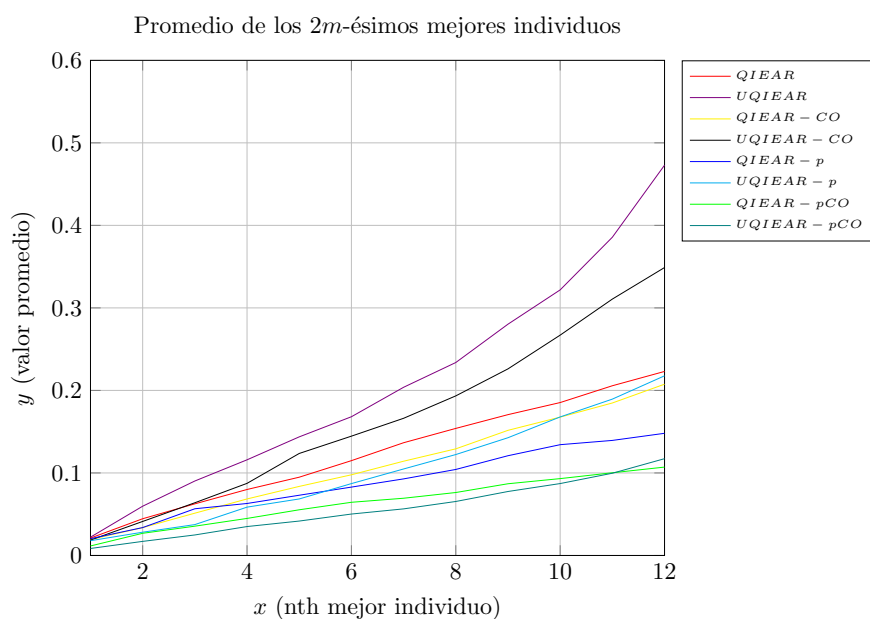


Figura 5.17: Valor promedio de los  $n$ -ésimos mejores individuos para  $n = 6$  - Función Maytas

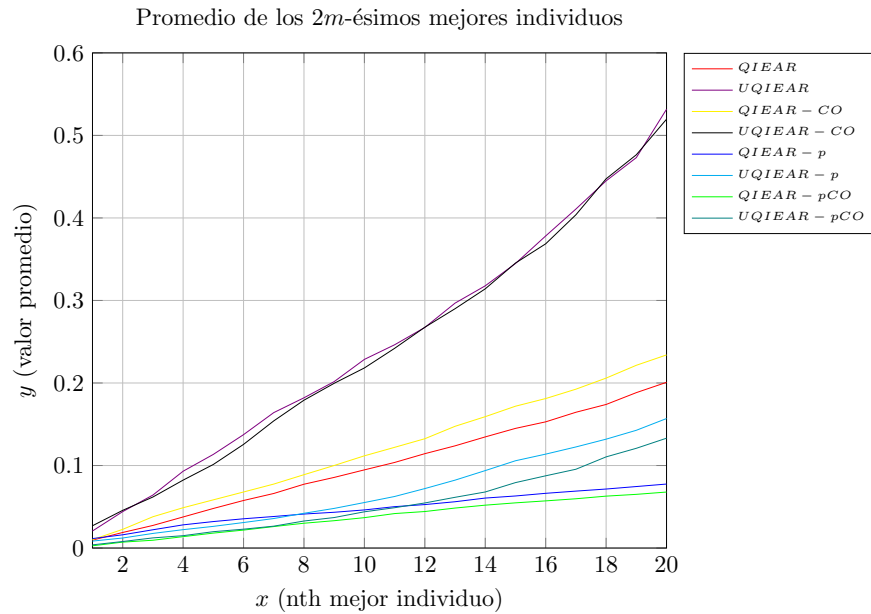


Figura 5.18: Valor promedio de los  $n$ -ésimos mejores individuos para  $n = 10$  - Función Maytas

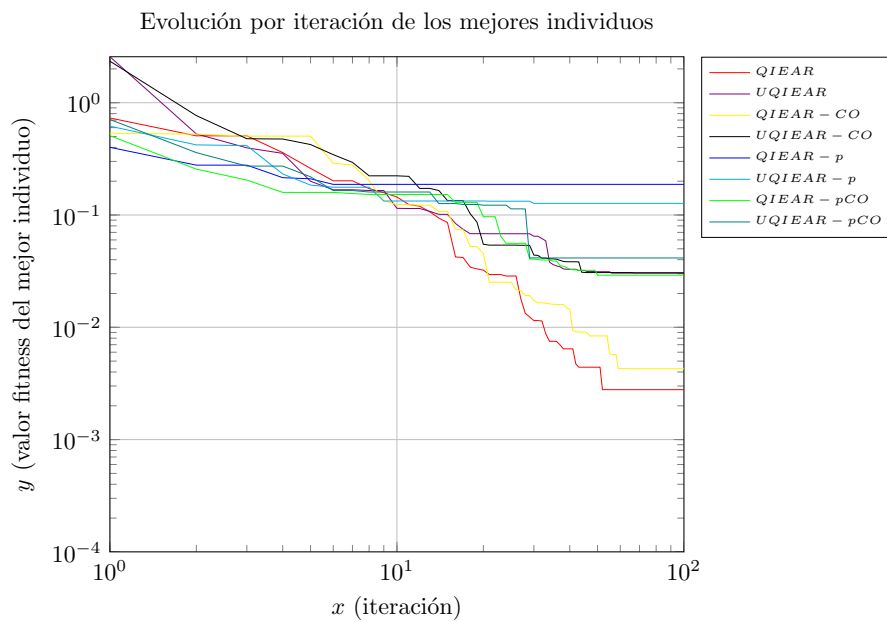


Figura 5.19: Evolución de los valores de los mejores individuos para  $n = 2$  - Función Maytas

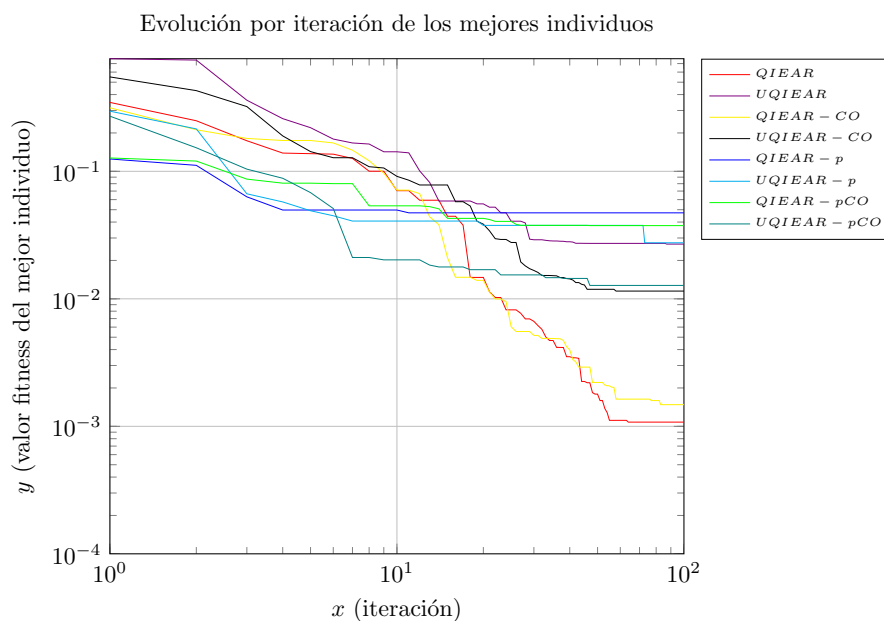


Figura 5.20: Evolución de los valores de los mejores individuos para  $n = 4$  - Función Maytas

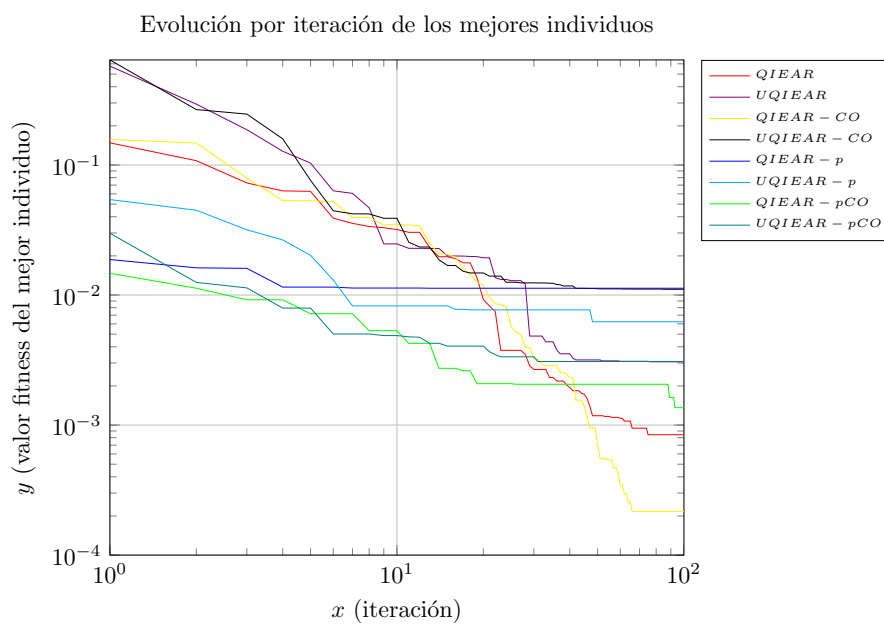


Figura 5.21: Evolución de los valores de los mejores individuos para  $n = 10$  - Función Maytas



## Capítulo 6

# Conclusiones y Trabajos Futuros

El presente trabajo ha propuesto distintas modificaciones al algoritmo original QIEAR con la expectativa por obtener resultados mejores en términos de tiempo de convergencia y calidad de datos generados. En términos generales, los únicos algoritmos que han registrado resultados medianamente consistentes y competitivos a lo largo de todas las pruebas son las variantes que implementan el operador de cruzamiento en espacios particionados. Esto es: QIEAR-pCO y UQIEAR-pCO.

Desafortunadamente, en los casos en los que registran mejoras sobre los algoritmos originales (sean QIEAR y QIEAR-p), ésta no resulta ser muy notoria. Además, hay otra cantidad similar de casos en los que ambos se ven superados, ya sea por uno u otro.

Los demás algoritmos propuestos obtuvieron resultados más irregulares y menos óptimos, por lo que se desaconseja su consideración a futuro. Cabe destacar que los algoritmos con peores resultados en general fueron los que implementaban el particionamiento del espacio de búsqueda sin el operador de cruzamiento.

### 6.1. Problemas encontrados

La segunda parte de este capítulo corresponde a los problemas encontrados. esta sección es muy importante para que los siguientes estudiantes que hagan algo en esta línea no cometan los mismos errores y tu tesis sea un buen peldaño para avanzar más rápido.

### 6.2. Recomendaciones

En esta sección el tesista debe reflejar que la tesis ha permitido adquirir nuevos conocimientos que podrían servir para guiar otros trabajos en el futuro.

---

## 6.3. Trabajos futuros

En base a los puntos anteriores es recomendable que tu tesis también sugiera trabajos futuros. Esta sección es especialmente útil para otras ideas de tesis.

Todo este capítulo no debe ser más de unas 4 páginas.

## Bibliografía

- [Alegria and Túpac, 2014] Alegria, J. M. and Túpac, Y. J. (2014). A generalized quantum-inspired evolutionary algorithm for combinatorial optimization problems.
- [Benioff, 1980] Benioff, P. (1980). The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, 22(5):563–591.
- [Benioff, 1982] Benioff, P. (1982). Quantum mechanical models of turing machines that dissipate no energy. *Physical Review Letters*, 48(23):1581.
- [Censor, 1997] Censor, Y. (1997). *Parallel optimization: Theory, algorithms, and applications*. Oxford University Press.
- [Coello, 1999] Coello, C. A. C. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information systems*, 1(3):269–308.
- [da Cruz, 2007] da Cruz, A. (2007). *Algoritmos evolutivos com inspiração quântica para problemas com representação numérica*. PhD thesis, PhD thesis, Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.(In Portuguese).
- [Dias and Pacheco, 2013] Dias, D. M. and Pacheco, M. A. C. (2013). Quantum-inspired linear genetic programming as a knowledge management system. *The Computer Journal*, 56(9):1043–1062.
- [Dias et al., 2011] Dias, D. M., Singulani, A. P., Pacheco, M. A. C., de Souza, P. L., Pires, M. P., and Neto, O. P. V. (2011). Self-assembly quantum dots growth prediction by quantum-inspired linear genetic programming. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2075–2082. IEEE.
- [Eiben and Smith, 2003] Eiben, A. E. and Smith, J. E. (2003). *Introduction to evolutionary computing*. springer.
- [Fan et al., 2007] Fan, K., Brabazon, A., O’Sullivan, C., and O’Neill, M. (2007). Option pricing model calibration using a real-valued quantum-inspired evolutionary algorithm. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1983–1990. ACM.
- [Fan et al., 2008] Fan, K., Brabazon, A., O’Sullivan, C., and O’Neill, M. (2008). Quantum-inspired evolutionary algorithms for financial data analysis. In *Applications of Evolutionary Computing*, pages 133–143. Springer.

- 
- [Gillespie, 1974] Gillespie, D. T. (1974). *A quantum mechanics primer: An Elementary Introduction to the Formal Theory of Non-relativistic Quantum Mechanics*. John Wiley and Sons.
- [Grover, 1996] Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM.
- [Gruska, 2004] Gruska, J. (2004). Quantum computing 1.
- [Han and Kim, 2000] Han, K.-H. and Kim, J.-H. (2000). Genetic quantum algorithm and its application to combinatorial optimization problem. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 2, pages 1354–1360. IEEE.
- [Han et al., 2001] Han, K.-H., Kim, J.-H., et al. (2001). Analysis of quantum-inspired evolutionary algorithm. In *Proceedings of the 2001 International Conference on Artificial Intelligence*, pages 727–730. Citeseer.
- [Hey, 1999] Hey, T. (1999). Quantum computing: an introduction. *Computing & Control Engineering Journal*, 10(3):105–112.
- [Hinterding et al., 1997] Hinterding, R., Michalewicz, Z., and Eiben, A. E. (1997). Adaptation in evolutionary computation: A survey. In *Evolutionary Computation, 1997., IEEE International Conference on*, pages 65–69. IEEE.
- [Hota and Pat, 2010] Hota, A. R. and Pat, A. (2010). An adaptive quantum-inspired differential evolution algorithm for 0–1 knapsack problem. In *Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on*, pages 703–708. IEEE.
- [Jeong et al., 2010] Jeong, Y.-W., Park, J.-B., Jang, S.-H., and Lee, K. Y. (2010). A new quantum-inspired binary pso: application to unit commitment problems for power systems. *Power Systems, IEEE Transactions on*, 25(3):1486–1495.
- [Kicinger et al., 2005] Kicinger, R., Arciszewski, T., and Jong, K. D. (2005). Evolutionary computation and structural design: A survey of the state-of-the-art. *Computers & Structures*, 83(23):1943–1978.
- [Li and Wang, 2007] Li, B.-B. and Wang, L. (2007). A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(3):576–591.
- [Malossini et al., 2008] Malossini, A., Blanzieri, E., and Calarco, T. (2008). Quantum genetic optimization. *Trans. Evol. Comp*, 12(2):231–241.
- [Mani and Patvardhan, 2010] Mani, A. and Patvardhan, C. (2010). Solving ceramic grinding optimization problem by adaptive quantum evolutionary algorithm. In *Intelligent Systems, Modelling and Simulation (ISMS), 2010 International Conference on*, pages 43–48. IEEE.
- [Miao et al., 2009] Miao, H., Wang, H., and Deng, Z. (2009). Quantum genetic algorithm and its application in power system reactive power optimization. In *Computational Intelligence and Security, 2009. CIS’09. International Conference on*, volume 1, pages 107–111. IEEE.

- 
- [Mishra, 2006] Mishra, S. (2006). Some new test functions for global optimization and performance of repulsive particle swarm method.
- [Narayanan and Moore, 1996] Narayanan, A. and Moore, M. (1996). Quantum-inspired genetic algorithms. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 61–66. IEEE.
- [Nielsen and Chuang, 2010] Nielsen, M. A. and Chuang, I. L. (2010). *Quantum computation and quantum information*. Cambridge university press.
- [Nowotniak, 2010] Nowotniak, R. (2010). Survey of quantum-inspired evolutionary algorithms.
- [Nowotniak and Kucharski, 2014] Nowotniak, R. and Kucharski, J. (2014). Higher-order quantum-inspired genetic algorithms. *arXiv preprint arXiv:1407.0977*.
- [Qin et al., 2008] Qin, C., Liu, Y., and Zheng, J. (2008). A real-coded quantum-inspired evolutionary algorithm for global numerical optimization. In *Cybernetics and Intelligent Systems, 2008 IEEE Conference on*, pages 1160–1164. IEEE.
- [Romero and Machado, 2008] Romero, J. J. and Machado, P. (2008). *The art of artificial evolution: a handbook on evolutionary art and music*. Springer.
- [Shor, 1994] Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134. IEEE.
- [Silagadze, 2007] Silagadze, Z. (2007). Finding two-dimensional peaks. *Physics of Particles and Nuclei Letters*, 4(1):73–80.
- [Singulani et al., 2008] Singulani, A. P., Vilela Neto, O. P., Aurélio Pacheco, M. C., Vellasco, M. B., Pires, M. P., and Souza, P. L. (2008). Computational intelligence applied to the growth of quantum dots. *Journal of Crystal Growth*, 310(23):5063–5065.
- [Talbi et al., 2007] Talbi, H., Batouche, M., and Draa, A. (2007). A quantum-inspired evolutionary algorithm for multiobjective image segmentation. *International Journal of Mathematical, Physical and Engineering Sciences*, 1(2):109–114.
- [Teng et al., 2010] Teng, H., Zhao, B., and Cao, A. (2010). Chaos quantum genetic algorithm based on henon map. *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, 1:922–925.
- [Tsai et al., 2012] Tsai, J.-T., Chou, J.-H., and Ho, W.-H. (2012). Improved quantum-inspired evolutionary algorithm for engineering design optimization. *Mathematical Problems in Engineering*, 2012.
- [Venayagamoorthy and Singhal, 2005] Venayagamoorthy, G. K. and Singhal, G. (2005). Quantum-inspired evolutionary algorithms and binary particle swarm optimization for training mlp and srn neural networks. *Journal of Computational and Theoretical Nanoscience*, 2(4):561–568.
- [Wall, 1991] Wall, D. W. (1991). *Limits of instruction-level parallelism*, volume 19. ACM.
-

- 
- [Xing et al., 2009] Xing, H., Liu, X., Jin, X., Bai, L., and Ji, Y. (2009). A multi-granularity evolution based quantum genetic algorithm for qos multicast routing problem in wdm networks. *Computer Communications*, 32(2):386–393.
- [Zhang et al., 2006] Zhang, G.-x., Li, N., Jin, W.-d., and Hu, L.-z. (2006). Novel quantum genetic algorithm and its applications. *Frontiers of Electrical and Electronic Engineering in China*, 1(1):31–36.
- [Zhang et al., 2011] Zhang, J., Zhan, Z.-h., Lin, Y., Chen, N., Gong, Y.-j., Zhong, J.-h., Chung, H. S., Li, Y., and Shi, Y.-h. (2011). Evolutionary computation meets machine learning: A survey. *Computational Intelligence Magazine, IEEE*, 6(4):68–75.
- [Zhao et al., 2009] Zhao, S., Xu, G., Tao, T., and Liang, L. (2009). Real-coded chaotic quantum-inspired genetic algorithm for training of fuzzy neural networks. *Computers & Mathematics with Applications*, 57(11).
- [Zhou et al., 2011] Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49.