



UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES

**INTELIGENCIA COMPUTACIONAL INSPIRADA EN LA CUÁNTICA, APLICADA A LA
OPTIMIZACIÓN DE LA PRODUCCIÓN DE PETRÓLEO EN EL MODELO DE UN POZO
INTELIGENTE**

MARISOL HERNÁNDEZ BÁEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA

2013



UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES

**INTELIGENCIA COMPUTACIONAL INSPIRADA EN LA CUÁNTICA, APLICADA A LA
OPTIMIZACIÓN DE LA PRODUCCIÓN DE PETRÓLEO EN EL MODELO DE UN POZO
INTELIGENTE**

MARISOL HERNÁNDEZ BÁEZ

Trabajo de grado para optar al título de Ingeniero Electrónico

PhD CARLOS RODRIGO CORREA CELY

DIRECTOR

MSc IVÁN MAURICIO AMAYA

CODIRECTOR

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA

2013

AGRADECIMIENTOS

La autora quiere expresar su agradecimiento a:

- **La Universidad Industrial de Santander (UIS).**
- **A la escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.**
- **PhD Rodrigo Correa Cely, Director del proyecto, por su paciencia y colaboración durante el desarrollo de éste.**
- **MSc Iván Mauricio Amaya, Codirector del proyecto, por sus asesorías, tiempo y disponibilidad para poder concluir.**
- **MSc Morteza Haghghat por su oportuna y desinteresada colaboración.**
- **PhD Diana Hernández Báez por su ayuda, ejemplo y guía, a lo largo de todo el proyecto.**
- **A todas las personas que de una u otra forma participaron en el desarrollo del proyecto de grado.**

DEDICATORIA

**“Que no se pierda en quimeras el destino para el que has sido forjado,
Aunque estés hecho de sueños, tu realidad va más allá de los límites humanos”**

MHB

*A Dios, por bendecirme con sus dones y regalarme desde que tengo conciencia, la presencia de
personas admirables...*

*A mi nonita, por dejarnos la huella imborrable de su corazón bondadoso y lleno de alegría hasta
sus últimos días...*

*A mi padre, por dedicar el esfuerzo de su vida, y apoyarnos para que todos cumpliéramos nuestras
metas...*

*A mi madre, por darnos sin dudar un instante lo mejor de sí misma, para que pudiéramos darlo
todo de nosotros...*

*A Nelson, por mostrarme la valentía que se requiere, para mantenerme siempre en pie y nunca
dejar de creer en mí misma...*

*A Sandra, por enseñarme en gran parte el carácter y la tenacidad que necesité para asumir cada
uno de mis retos...*

*A Alexander, por mostrarme que no hay un límite de rutas alternativas mientras no olvide cuál es
mi norte...*

*A Diana, por ser el ejemplo de cómo Dios lleva nuestros pasos por senderos mágicos según su
voluntad...*

*A todos aquellos que con o sin intención le dieron forma a este sueño que se ha consumado...
Gracias...*

TABLA DE CONTENIDO

INTRODUCCIÓN	12
1. DESCRIPCIÓN DEL TRABAJO DE GRADO	13
1.1. Objetivo General	13
1.2. Objetivos específicos	13
2. CONSIDERACIONES PRELIMINARES	14
2.4. Inteligencia computacional (ICC)	14
2.5. Computación evolutiva (CE)	15
2.6. Algoritmos genéticos (AG)	16
2.7. Representación de soluciones	16
3. DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO HÍBRIDO QABC	19
3.1. Algoritmo evolutivo inspirado en la cuántica (QEA)	19
3.1.1. Descripción del algoritmo (QEA)	20
3.2. Hibridación de algoritmos	21
3.3. Hibrido ABC y QEA (QABC)	22
3.3.1. Algoritmo de optimización de colonia artificial de abejas (ABC)	22
3.3.2. Descripción del algoritmo (ABC)	24
3.3.3. Proceso de hibridación ABC y QEA	24
3.4. Aplicación a los pozos inteligentes	27
3.4.1. Representación de ICVs	28
3.4.2. Estructura de optimización	29
3.5. Resultados Experimentales	33
3.5.1. Función Rosenbrock	33
3.5.2. Función objetivo: Optimización del pozo	39
CONCLUSIONES Y RECOMENDACIONES	44
REFERENCIAS	45

LISTA DE TABLAS

Tabla 1. Técnicas de Inteligencia Computacional Inspirada en Cuántica.	14
Tabla 2. Algoritmos híbridos Inspirados en Cuántica.	15
Tabla 3. Tabla de rotación.	20
Tabla 4. Constante de conversión de unidades Cu.	28
Tabla 5. Definición de parámetros de ambos algoritmos para la función Rosenbrock.	34
Tabla 6. Comparación del mínimo global con 100 iteraciones de ambos algoritmos para la función Rosenbrock.	34
Tabla 7. Comparación de resultados de la optimización del pozo para ambos algoritmos respecto al caso base.	40
Tabla 8. Comparación de resultados de la optimización del pozo para ambos algoritmos respecto al caso base en promedio.	41

LISTA DE FIGURAS

Figura 1. Codificación de soluciones en un algoritmo genético	17
Figura 2. Codificación y función objetivo (fitness)	18
Figura 3. Operación del QEA	21
Figura 4. Posibilidades de Hibridación en un algoritmo evolutivo	22
Figura 5. Comportamiento de las abejas en la búsqueda de néctar.	23
Figura 6. Pseudo-código del híbrido ABC y QEA.	26
Figura 7. Esquema de componentes de un pozo inteligente multilateral	27
Figura 8. Extracto de la definición de parámetros de las ICVs en ECLIPSE	29
Figura 9. Diagrama de flujo de la estructura de optimización	31
Figura 10. Ciclo del agua en un pozo de petróleo.....	32
Figura 10. Extracto de la definición de parámetros en ECLIPSE.....	33
Figura 12. Algoritmo QEA con 100 iteraciones para la función Rosenbrock (Mejor individuo)	35
Figura 13. Algoritmo QEA con 100 iteraciones para la función Rosenbrock (Promedio)	36
Figura 14. Algoritmo QABC con 100 iteraciones para la función Rosenbrock (Mejor Individuo)	37
Figura 15. Algoritmo QABC con 100 iteraciones para la función Rosenbrock (Promedio)	38
Figura 16. Optimización de la producción con el algoritmo QEA en ECLIPSE	42
Figura 17. Optimización de la producción con el algoritmo QABC en ECLIPSE....	43

RESUMEN

TITULO: INTELIGENCIA COMPUTACIONAL INSPIRADA EN LA CUÁNTICA, APLICADA A LA OPTIMIZACIÓN DE LA PRODUCCIÓN DE PETRÓLEO EN EL MODELO DE UN POZO INTELIGENTE*

AUTOR: MARISOL HERNÁNDEZ BÁEZ **

PALABRAS CLAVES: QEA, ABC, QABC, ICV

El presente trabajo de grado en la modalidad de investigación, propone desarrollar una estrategia numérica híbrida fundamentada en un algoritmo inspirado en la cuántica, para resolver como ejemplo, la optimización de la producción mediante la configuración de válvulas de control (Inflow Control Valves - ICV) descritas por medio de la simulación del modelo de un pozo dado mediante una fuente confiable y confidencial en la industria del petróleo.

Una detallada y comprensiva explicación de los fundamentos teóricos sobre algoritmos genéticos (GA), Algoritmos cuánticos evolutivos (QEA), Algoritmos de optimización de colonia de abejas (ABC), es presentada en este estudio, siendo la base del proceso de diseño del algoritmo híbrido (QABC) desarrollado en este trabajo, y su correspondiente aplicación a la configuración de válvulas de control para la optimización de la producción

Los resultados obtenidos con el algoritmo híbrido (QABC) desarrollado en este trabajo, mostraron una mejora significativa en la solución del problema de optimización (aplicado a pozos inteligentes, y la función de prueba Rosenbrock) con respecto a los métodos presentados anteriormente, exhibiendo un tiempo de computación aceptable.

Este trabajo de grado contribuye al desarrollo de nuevas metodologías de hibridación de algoritmos pertenecientes al campo de la Inteligencia computacional aplicados en diferentes procesos a nivel industrial.

*Trabajo de grado

** Facultad de Ingenierías Físico-mecánicas, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.
Director: Carlos Rodrigo Correa Cely. Codirector: Iván Mauricio Amaya Contreras

ABSTRACT

TITULO: QUANTUM INSPIRED COMPUTATIONAL INTELLIGENCE, APPLIED TO THE OIL PRODUCTION OPTIMIZATION IN AN INTELLIGENT WELL MODEL*

AUTOR: MARISOL HERNÁNDEZ BÁEZ**

PALABRAS CLAVES: QEA, ABC, QABC, ICV

In this paper a numerical strategy based on a quantum inspired evolutionary algorithm is proposed to solve a production optimization problem of a smart well model.

This project aims to apply the new algorithm to the production optimization by means of the configuration of Inflow Control Valves (ICV) described in the simulated well model previously obtained by a reliable and confidential source in the oil industry.

A very comprehensive insight into the theoretical fundamentals of genetic algorithms (GA), quantum inspired evolutionary algorithms (QEA), and Artificial be colony algorithm (ABC) is introduced, in order to be the basis for the design process of the hybrid Quantum artificial bee colony (QABC) algorithm developed in this work, and its correspondent application to the Inflow control valves optimization.

The results obtained with the hybrid algorithm developed (QABC), have shown a significant improvement on the final solution for the chosen problems (applied to oil production optimization in intelligent wells, and the best outcome in the test function Rosenbrock), regarding the methods above introduced, within an acceptable computing time.

This final degree project, will contribute to the development of novel algorithm hybridization methodologies belonging to the computational intelligence field, applied in different processes at an industrial level.

* Degree work

** Facultad de Ingenierías Físico-mecánicas, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.
Director: Carlos Rodrigo Correa Cely. Codirector: Iván Mauricio Amaya Contreras

INTRODUCCIÓN

Debido al aumento en la complejidad de la tecnología, han aparecido nuevos y retadores problemas en la teoría de la optimización, acrecentando así la presencia del álgebra en ella. El concepto de la optimización es primordial, pues constituye una técnica cuyo objetivo es mejorar el valor de una variable (maximizar o minimizar). Esto puede aplicarse en: temperaturas, rentabilidad, beneficio, entre otras [6].

Por otra parte, la tecnología informática y de la computación ha jugado un papel crítico en las aplicaciones de las técnicas de optimización, tal y como ocurre en el control óptimo de cohetes y proyectiles. En efecto, en vista de la complejidad de los sistemas en la actualidad, es imposible realizar una implementación eficiente de los métodos de optimización, sin previamente realizar un riguroso trabajo de simulación numérica.

La inteligencia computacional (IC) es una rama de la inteligencia artificial la cual envuelve el estudio de los mecanismos adaptativos para facilitar comportamientos inteligentes en ambientes complejos y cambiantes, estableciendo su rol como una herramienta de optimización y diseño. Por su parte, la computación inspirada en cuántica (ICC), es un campo de investigación caracterizado por ciertos principios de mecánica cuántica, combinados con la IC con el fin de resolver inconvenientes de ingeniería con un comportamiento cíclico y recurrente. Se espera que en consecuencia, las aplicaciones de la ICC ofrezcan un buen desempeño además de ofrecer muchas estrategias de mejoramiento.

Este trabajo pretende mostrar una de las múltiples aplicaciones de la inteligencia computacional inspirada en cuántica en el campo de la ingeniería, sirviendo como guía y base para próximas investigaciones realizadas por el grupo CEMOS y la comunidad universitaria en general. Así mismo, se busca motivar al estudiante o practicante para que domine el potencial y aprecie la elegancia de los métodos algebraicos en la teoría de optimización.

Cada uno de los temas tratados, se muestran de manera concisa a lo largo de este libro. En la primera sección, se presenta una breve descripción del trabajo. En la segunda sección, se contemplan algunos conceptos básicos que sustentan el estudio. Finalmente, en la sección tres, se ejecuta el diseño del algoritmo híbrido y se exponen los resultados obtenidos con su respectivo análisis, en el marco del desarrollo de la investigación y la consecución de los objetivos en busca del impacto deseado.

1. DESCRIPCIÓN DEL TRABAJO DE GRADO

Con este trabajo se pretende desarrollar un software de optimización de la producción mediante la configuración de válvulas de control (Inflow Control Valves - ICV) descritas mediante la simulación del modelo de un pozo dado. Para efectuar dicho procedimiento se elegirá un algoritmo inspirado en la cuántica.

Dado que existen varias técnicas de inteligencia computacional, la selección, profundización en el conocimiento de su principio de funcionamiento, manejo e interpretación, aportará otra alternativa para resolver la función objetivo que represente el modelo con su comportamiento específico. A continuación se enumeran los objetivos aprobados en el plan propuesto para el trabajo de grado:

1.1. Objetivo General

Solucionar un problema en el área de optimización, con un algoritmo de inteligencia computacional inspirado en la cuántica.

1.2. Objetivos específicos

- 1.2.1** Seleccionar una de las técnicas de inteligencia computacional inspirada en la cuántica para ser profundizada. (páginas 16 a 24)
- 1.2.2** Implementar un software tomando como base el algoritmo estudiado y aplicarlo a la optimización del modelo de un pozo inteligente simulado. (páginas 25 a 31)
- 1.2.3** Contrastar los resultados frente a la optimización generada mediante un algoritmo convencional. (páginas 31 a 41)

2. CONSIDERACIONES PRELIMINARES

2.4. Inteligencia computacional inspirada en la cuántica (ICC)

La sociedad de inteligencia computacional de la IEEE, creada por Jim Bezdek, define la ICC como cualquier paradigma computacional biológica, natural, y lingüísticamente motivado, que incluye, pero no se limita a: redes neuronales, sistemas difusos, computación evolutiva (CE), desarrollo mental autónomo, y sistemas híbridos inteligentes que contengan a los ya mencionados [7]. Observando las ventajas, como el comportamiento biológicamente motivado (aprendizaje, razonamiento o evolución), podemos apreciar que paralelamente al procesamiento de la información distribuida y el potencial de manejar las complicaciones del mundo real, sufre inconvenientes como la falta de análisis cualitativo, salidas no repetibles, naturaleza estocástica entre otros. Cabe resaltar, que las mayores ventajas de la ICC, como la amplia aplicabilidad, la robustez para ambientes dinámicos y la capacidad de mejorar por ella misma, establecen su rol como una importante herramienta de optimización y diseño en múltiples campos de la ciencia y la ingeniería.

La teoría de computación cuántica se relaciona con la teoría de computación reversible, trayendo ambas ideas de la teoría de la información clásica, la computación científica, y la física cuántica. La física cuántica fue establecida en el periodo de 1923-1927 gracias a los esfuerzos de Planck, Einstein, Bohr, Heisenberg, Schrodinger y demás. Dicha teoría constituye una estructura matemática para la construcción de teorías físicas. La investigación de principios aplicados de la física cuántica al mejoramiento de la ingeniería de sistemas inteligentes fue impulsada a finales de 1990 [10].

Basados en el artículo “Applications of quantum inspired computational intelligence: a survey” [1], las técnicas de inteligencia computacional inspirada en cuántica se pueden dividir en tres grupos como podemos ver en la Tabla 1. Tenemos la ICC inspirada en las redes neuronales, la ICC inspirada en métodos evolutivos y la ICC inspirada en sistemas difusos. Adicionalmente, podemos encontrar algunas técnicas híbridas en la Tabla 2, cuyo objetivo es combinar las técnicas anteriores para aprovechar sus beneficios mientras mitigan sus desventajas individuales.

Tabla 1. Técnicas de Inteligencia Computacional Inspirada en Cuántica.

Redes Neuronales (RN)	Métodos Evolutivos	Sistemas Difusos
Entrenamiento de RN	Algoritmo genético	Algoritmo de síntesis (cuántica de mediciones, síntesis semántica y lógica difusa)
Algoritmo de entrenamiento Hamiltoniano en RN	Algoritmo evolutivo	
RN de Hopfield	Evolución diferencial	
Mapa auto-organizacional de RN	Algoritmo de optimización de enjambre de partículas	
RN recurrentes	Optimización de colonia de abejas	
Análisis de estabilidad	Algoritmo artificial de enjambre de peces	
Máquinas de soporte vectorial	Sistemas inmunes	
RN de impulsos		

Fuente [1]

Tabla 2. Algoritmos híbridos Inspirados en Cuántica.

Metodología	Comparación
Entrenamiento BPNN usando mejoras de QPSO	BPNN con momentum
Entrenamiento SVM usando paralelo con QGA	GA
Entrenamiento de controlador Difuso/Neural usando doble cadena OGA	GA
Optimización de difuso usando QPSO	FS+PSO, PSO
Algoritmo auto-organizacional = QNN+QGA	FS+PSO, PSO
Auto-organizacional difuso = Interferencia cuántica difusa + QGA	FS+PSO, PSO
QSEA=QEA+PSO	GA, PSO

Fuente [1]

Actualmente, los conocimientos sobre la computación cuántica están siendo aplicados en varios campos, como es el caso del mecanismo Electromagnetism-Like (Inspirado en algunos principios de la teoría del electromagnetismo) que es un algoritmo metaheurístico basado en la atracción y repulsión mecánica, propia de los campos electromagnéticos, Algoritmos codiciosos iterativos, computación mórfica entre otras.

En poco más de una década miles de documentos han reportado aplicaciones exitosas de la ICC, ayudando a los partidarios e investigadores a tener una visión clara del avance científico en esta área hasta la fecha. El gran número de publicaciones que se relacionan con el tema han sido categorizadas basándose en problemas resueltos y las clases de inteligencia computacional.

2.5. Computación evolutiva (CE)

La expresión computación evolutiva designa un amplio conjunto de técnicas heurísticas de resolución de problemas complejos que basan su funcionamiento en un mecanismo análogo a los procesos de la evolución natural. Trabajando sobre un conjunto de soluciones a un problema determinado, la metodología utilizada por estas técnicas se fundamenta en el uso de mecanismos de selección de las mejores soluciones potenciales y de construcción de nuevas soluciones candidatas mediante recombinación de características de las soluciones seleccionadas.

El algoritmo evolutivo trabaja sobre individuos que representan potenciales soluciones al problema, codificados de acuerdo a un mecanismo determinado. Los individuos son evaluados de acuerdo a una función objetivo que toma en cuenta la adecuación de cada solución al problema que se intenta resolver.

La operación del algoritmo evolutivo comienza con una etapa de inicialización de los individuos, que puede ser completamente aleatoria, muestreando al azar diferentes secciones del espacio de soluciones, o guiada de acuerdo a características del problema a resolver. El algoritmo evolutivo podría inclusive tomar como población inicial individuos resultantes como salida de algún otro algoritmo heurístico que permitiera calcular buenas soluciones iniciales aproximadas para el problema.[8]

La evolución propiamente dicha se lleva a cabo en el ciclo que genera nuevos individuos a partir de la población actual mediante un procedimiento de aplicación de operadores estocásticos. En este ciclo se distinguen cuatro etapas:

- Evaluación: etapa que consiste en asignar un valor de adecuación (fitness) a cada individuo en la población. Este valor representa el grado de efectividad de la solución de cada individuo al problema en cuestión, y es utilizado para guiar el mecanismo evolutivo.
- Selección: proceso que determina candidatos apropiados, de acuerdo a sus valores de adecuación, para la aplicación de los operadores evolutivos con el objetivo de engendrar la siguiente generación de individuos.
- Aplicación de los operadores evolutivos: etapa que genera un conjunto de descendientes a partir de los individuos seleccionados en la etapa anterior.
- Reemplazo: mecanismo que realiza el recambio generacional, sustituyendo individuos de la generación anterior por descendientes creados en la etapa anterior.

2.6. Algoritmos genéticos (AG)

Los algoritmos genéticos constituyen una de las técnicas de computación evolutiva más difundidas en la actualidad, como consecuencia de su versatilidad para resolver un amplio rango de problemas. Al constituir un caso de técnica evolutiva, los algoritmos genéticos basan su operativa en una emulación de la evolución natural de los seres vivos, trabajando sobre una población de soluciones potenciales que evoluciona de acuerdo a interacciones y transformaciones únicas.

Los individuos que constituyen la población se esfuerzan por sobrevivir a una selección programada en el proceso evolutivo, inclinada hacia los individuos más aptos, la cual determina aquellos individuos que formarán parte de la siguiente generación.

El grado de adaptación de un individuo se evalúa de acuerdo al problema a resolver, mediante la definición de una función de adecuación al problema, la función objetivo. Bajo ciertas condiciones, el mecanismo definido por los operadores inspirados por la genética natural y la evolución darwiniana llevan a la población a converger hacia una solución aproximada al óptimo del problema, luego de un determinado número de generaciones.

La característica distintiva de los algoritmos genéticos respecto a las otras técnicas evolutivas consiste en el uso fundamental del cruzamiento como operador principal, mientras que la mutación se utiliza como operador secundario tan solo para agregar una nueva fuente de diversidad en el mecanismo de exploración del espacio de soluciones del problema. Inclusive la mutación puede llegar a ser un operador opcional o estar ausente en algunas variantes de algoritmos genéticos que utilizan otros operadores para introducir diversidad.

En general, los algoritmos genéticos se han utilizado para trabajar con codificaciones binarias para problemas de búsqueda en espacios de cardinalidad numerable, aunque su alto nivel de aplicabilidad ha llevado a proponer su trabajo con codificaciones reales, e inclusive con codificaciones no tradicionales, dependientes de los problemas a resolver.

2.7. Representación de soluciones

Los algoritmos genéticos no trabajan directamente sobre las soluciones del problema en cuestión, sino que lo hacen sobre una abstracción de los objetos solución, usualmente denominadas cromosomas por analogía con la evolución natural biológica. Un cromosoma es un vector de genes, mientras que el valor asignado a un gen se denomina alelo.

En la terminología biológica, genotipo denota al conjunto de cromosomas que definen las características de un individuo. El genotipo sometido al medio ambiente se denomina fenotipo. En términos de los algoritmos genéticos el genotipo también está constituido por cromosomas, utilizándose generalmente un único cromosoma por solución individuo del problema. Por ello

suelen utilizarse indistintamente los términos genotipo, cromosoma e individuo. Por su parte, el fenotipo representa un punto del espacio de soluciones del problema.

Dado que un algoritmo genético trabaja sobre cromosomas, se debe definir una función de codificación sobre los puntos del espacio de soluciones, que mapea todo punto del espacio de soluciones en un genotipo, tal como se indica en la ecuación 1. La función inversa de la codificación, denominada decodificación permite obtener el fenotipo asociado a un cromosoma.

$$\text{Codificación: Espacio de soluciones} \rightarrow \text{cromosoma} \quad (1)$$

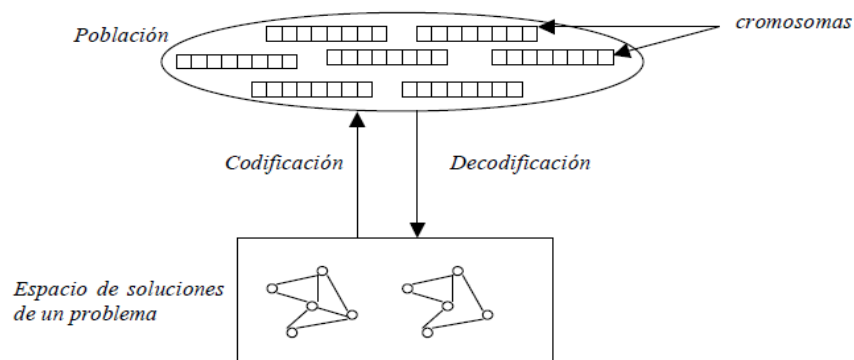
Tomando en cuenta la observación anterior, los mecanismos de codificación de soluciones individuo resultan importantes para el proceso de búsqueda de los algoritmos genéticos. Antes de iniciar la operación del algoritmo, se debe generar una codificación favorable como un mecanismo que permita incorporar el conocimiento específico en la resolución de problemas complejos de optimización. Es útil representar la solución como un conjunto de parámetros unidos en forma de una cadena de valores. Codificaciones dependientes de los problemas se han propuesto con frecuencia, podemos encontrar varios tipos de soluciones, entre los cuales tenemos:

Forma Binaria: Los individuos codifican las variables en el espacio binario usando un conjunto de cardinalidad conocida de valores binarios (ceros y unos) llamado string de bits o bit-string. Cada bit-string representa una solución potencial del problema de acuerdo al mecanismo de codificación predefinido, en general dependiente de la aplicación, como los problemas de configuración de ICV (ver sección 4). La forma binaria opera más rápidamente que otras formas de codificación porque las soluciones solo pueden ser representadas por números enteros, requiere decodificación de variables antes de ser usadas en la función objetivo y son superiores a otras formas de codificación cuando los límites de la solución posible son amplios.

Forma Real: Es usado con menor frecuencia, es un proceso más intuitivo porque no requiere decodificación y se ajusta a problemas donde la solución es un número real, como la optimización de tasas de producción de pozos. Las formas continuas requieren un tamaño de población más grande para buscar en un espacio de soluciones más extenso, y por tanto toma más tiempo. En particular las codificaciones basadas en números reales son útiles para representar soluciones cuando se resuelven problemas sobre espacios de cardinalidad no numerable, como en el caso de determinación de parámetros en problemas de control o entrenamiento de redes neuronales.

La Figura 1 resume gráficamente la relación entre el espacio de soluciones de un problema, la población de cromosomas con la cual trabaja el algoritmo genético y las funciones de codificación y decodificación.

Figura 1. Codificación de soluciones en un algoritmo genético

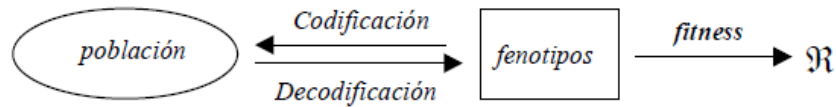


Fuente: [13]

Función de adaptabilidad o función objetivo (fitness):

Todo cromosoma tiene un valor asociado de adaptabilidad que evalúa aptitud del individuo para resolver el problema en cuestión. La función de adaptabilidad es la misma función objetivo del problema, lo cual implica que el cálculo del valor de la adaptabilidad se realiza sobre el fenotipo correspondiente al cromosoma, como se presenta en la Figura 2.

Figura 2. Codificación y función objetivo (fitness)



Fuente: [13]

La función objetivo tiene una influencia importante en el mecanismo del AG. Si bien actúa como una caja negra para el proceso evolutivo, la función objetivo guía el mecanismo de exploración, al actuar representando al entorno que evalúa la bondad de una solución individuo para la resolución del problema.

3. DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO HÍBRIDO QABC

3.1. Algoritmo evolutivo inspirado en la cuántica (QEA)

El algoritmo evolutivo inspirado en la cuántica, se basa en conceptos y principios de la computación cuántica como el bit cuántico (Q-bit) y la superposición de estados. Al igual que los algoritmos evolutivos, el QEA también se caracteriza por la representación de individuos, la función de evaluación y las dinámicas de población. [11]

Es importante notar que aunque QEA se basa en el concepto de la computación cuántica, no es un algoritmo cuántico, sino un novedoso algoritmo evolutivo para una computación clásica.

Se seleccionó este algoritmo como herramienta de optimización para encontrar la configuración óptima de la ICV porque:

- Es un método de búsqueda global aplicable a funciones con óptimo local.
- Es aplicable a funciones discontinuas donde no se pueden obtener derivaciones.
- Funciona bien para computación paralela lo cual incrementa la velocidad de optimización
- Es fácil hacer híbridos con otros algoritmos de optimización.

Conceptos fundamentales:

Q-bit: El QEA usa Q-bit como representación probabilística, definidos como la unidad de información básica varía de 0 a 1. Para K Q-bits, hay 2^k valores posibles.

Cromosomas Q-bit: Pueden representar una superposición lineal de estados (soluciones binarias) en el espacio de búsqueda probabilístico, adoptada para mantener variedad de soluciones y superar la convergencia prematura. Una compuerta rotacional cuántica hace uso completo de la información de los mejores individuos actuales para actualizarlos y evitar estancamiento. QEA usa la compuerta rotacional en mutación y la interferencia total en intercambio genético.

$$q = \begin{bmatrix} \alpha_1 & \dots & \alpha_m \\ \beta_1 & \dots & \beta_m \end{bmatrix} \quad (2)$$

Donde $|\alpha_i|^2 + |\beta_i|^2 = 1 \quad i=1, \dots, m$

M=número de Q-bits y el tamaño de la secuencia de Q-bits del individuo.

$|\alpha_i|^2$ Probabilidad de que el Q-bit sea 0

$|\beta_i|^2$ Probabilidad de que el Q-bit sea 1

Entrecruzamiento: En el proceso de entrecruzamiento, se usa la operación de selección de ruleta para elegir dos cromosomas cuánticos de la generación de padres, y se produce la generación de hijos por entrecruzamiento. Más adelante, los dos mejores individuos se eligen para crear la nueva generación evaluando su adaptabilidad. Esta operación se hace para mejorar la velocidad de convergencia y preservar la información. Usualmente elegimos 0.6 a 0.9 como la probabilidad de entrecruzamiento.

Mutación cuántica: Después del proceso de entrecruzamiento, elegimos el mejor como director de la mutación e implementamos la operación de mutación cuántica usando las reglas mostradas

en la tabla 3. Esta operación también se hace para mejorar la velocidad de convergencia y para incrementar la diversidad de la población. Elegimos 0.01 a 0.2 como la probabilidad de mutación. La mutación se completa a través de la compuerta de rotación cuántica $U(\theta)$, y se actualiza $[\alpha_i, \beta_i]^T$ como:

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \quad (3)$$

Tabla 3. Tabla de rotación.

X_i	$best\ i$	$f(x) > f(best)$	θ_i			
			$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	Falso	0	0	0	0
0	0	Verdadero	0	0	0	0
0	1	Falso	0	0	0	0
0	1	Verdadero	-0.05π	0.05π	$\pm 0.05\pi$	0
1	0	Falso	-0.05π	0.05π	$\pm 0.05\pi$	0
1	0	Verdadero	0.05π	-0.05π	0	$\pm 0.05\pi$
1	1	Falso	0.05π	-0.05π	0	$\pm 0.05\pi$
1	1	Verdadero	0.05π	-0.05π	0	$\pm 0.05\pi$

Fuente: [9]

3.1.1. Descripción del algoritmo (QEA)

Paso 1: Se genera una población cuántica aleatoria, Los $Q(t)$, α_i^0 y β_i^0 $i=1,2,\dots, m$, de todos los $q_j^0 = q_j^t|_{t=0}$, $j=1,2,\dots,n$, son inicializados con $1/\sqrt{2}$. Lo cual significa que un individuo Q-bit, representa la superposición lineal de todos los posibles estados con la misma probabilidad.

Paso 2: En este paso se crean soluciones binarias en $P(0)$ observando los estados de $Q(0)$, donde $P(0) = \{x_1^0, x_2^0, \dots, x_n^0\}$ en la generación $t=0$. Una solución binaria $x_j^0, j=1,2,\dots,n$, es una cadena binaria de tamaño m , la cual está formada mediante la selección de 0 o 1 para cada bit usando la probabilidad de que sea $|\alpha_i^0|^2$ o $|\beta_i^0|^2$ para $i=1,2,\dots, m$, de q_i^0 , respectivamente.

Paso 3: Cada solución binaria se evalúa para obtener el nivel de su función objetivo.

Paso 4: La mejor solución inicial se selecciona entre las soluciones binarias, $P(0)$ y se guarda en $B(0)$, donde $B(0) = \{b_1^0, b_2^0, \dots, b_n^0\}$, y $b_j^0 (= b_j^t|_{t=0})$ es el mismo que x_j^0 en la generación inicial.

Paso 5: En el ciclo **while**, se forman las soluciones binarias en $P(t)$ al observar los estados de $Q(t-1)$ como en el paso 2, y se evalúa cada solución binaria para hallar su valor objetivo (fitness). Debemos notar que x_j^t en $P(t)$ puede ser formado por múltiples observaciones de q_j^{t-1} en $Q(t-1)$ en este caso, x_j^t debe ser reemplazado por $x_{j_l}^t$, donde l es un índice de observación.

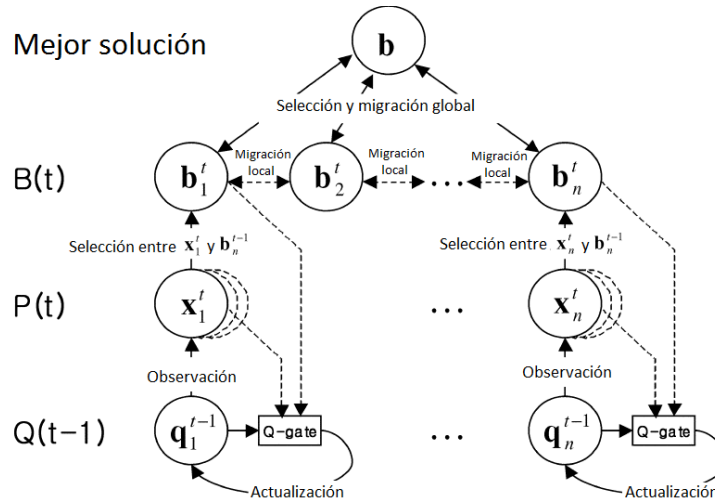
Paso 6: En este paso, se actualizan los individuos Q-bit en $Q(t)$ al aplica las Q-gates definidas en la ecuación 3.

Paso 7. Se seleccionan las mejores soluciones entre $B(t-1)$ y $P(t)$ y se guardan en $B(t)$, y si la mejor solución guardada en $B(t)$ se ajusta más que la mejor solución b , se reemplaza está en b .

Paso 8: Si las condiciones de migración se satisfacen, la mejor solución b migra a $B(t)$, o la mejor entre algunas soluciones en $B(t)$ migra hacia ellos. La condición de migración es un parámetro de diseño, y el proceso de migración definido abajo, puede inducir una variación de las probabilidades de un individuo Q-bit.

Migración: Se realiza al copiar b_j^t en $B(t)$ (migración local) o b en $B(t)$ (migración global)

Figura 3. Operación del QEA



Fuente: [11]

3.2. Hibridación de algoritmos

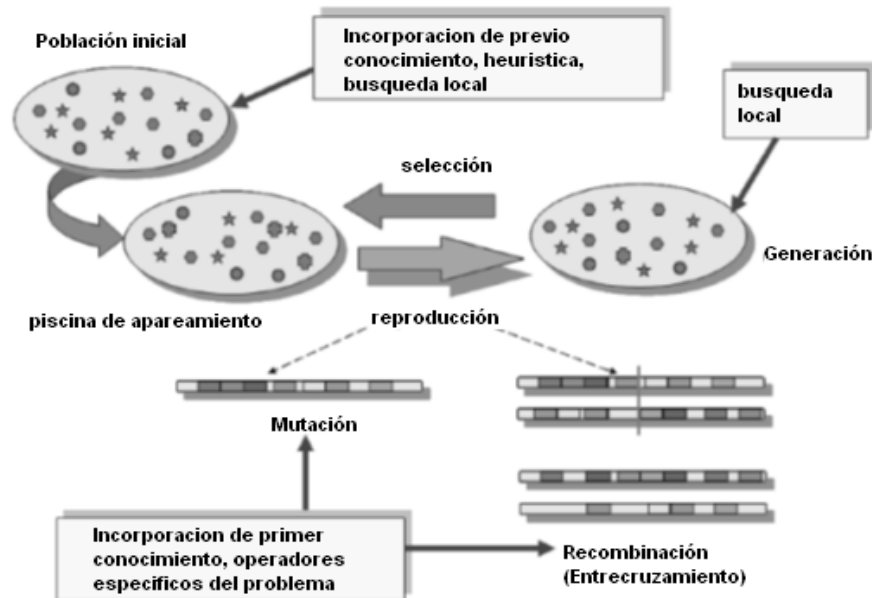
Para la mayoría de problemas un simple algoritmo evolutivo puede ser suficiente para hallar la solución deseada. Como se reporta en la literatura, hay varios tipos de problemas donde directamente un algoritmo evolutivo puede fallar al obtener una solución óptima. Esto muestra claramente la necesidad de la hibridación de algoritmos con otros algoritmos de optimización, heurísticas etcétera. Algunas de las posibles razones para la hibridación son:

1. Mejorar el desempeño del algoritmo evolutivo (ejemplo: velocidad de convergencia)
2. Mejorar la calidad de las soluciones obtenidas mediante el algoritmo evolutivo.
3. Incorporar el algoritmo evolutivo como parte de un sistema más grande.

La Figura 4 muestra algunas posibilidades para la hibridación. Desde la inicialización de la población hasta la generación de la descendencia, hay muchas oportunidades para incorporar otras técnicas/algoritmos. La población puede ser inicializada al incorporar soluciones conocidas, usando heurísticas, o realizando una búsqueda global. Los métodos de búsqueda global deben ser incorporados al interior de los miembros de la población inicial o dentro de la descendencia. Los algoritmos evolutivos pueden ser hibridados usando operadores de otros algoritmos (o por algoritmos) o incorporando conocimiento del dominio específico. El comportamiento de los

algoritmos evolutivos se determina por medio de las relaciones de explotación y exploración mantenidas a través de operación. Los algoritmos evolutivos adaptativos han sido construidos para inducir relaciones explotación/exploración que eviten el problema de convergencia prematura y optimicen los resultados finales [3].

Figura 4. Posibilidades de Hibridación en un algoritmo evolutivo



Fuente: [3]

3.3. Híbrido ABC y QEA (QABC)

Como consecuencia de una exhaustiva investigación acerca de la variedad de los algoritmos encontrados en la ICC y el estudio comparativo de Karaboga, [5] se desarrolla un algoritmo híbrido entre el ABC y QEA partiendo de un esquema planteado por Dian [9].

3.3.1. Algoritmo de optimización de colonia artificial de abejas (ABC)

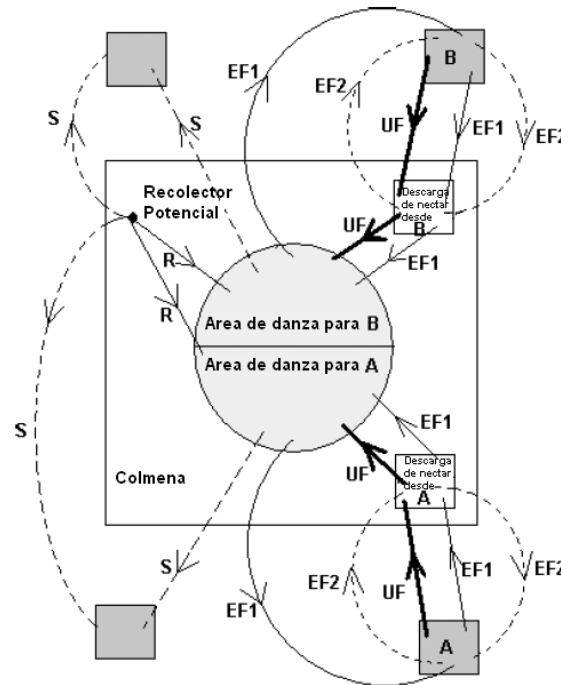
Este algoritmo está basado en el comportamiento de búsqueda inteligente de comida de un enjambre de abejas (Artificial bee colony - ABC). Solo usa parámetros de control comunes: Tamaño de colonia y número máximo de ciclo. Es un método de búsqueda basado en la población, en el cual los individuos llamados posiciones de comida son modificados por las abejas artificiales y su objetivo es descubrir los lugares de las fuentes de comida con más alta cantidad de néctar y finalmente obtener el mejor.

Los roles de las abejas se dividen según la forma de buscar su comida:

- Empleados/Espectadores: buscan dependiendo de su experiencia y de la de los demás, ajustan su posición (búsqueda local)
- Exploradoras: Eligen una fuente de alimento aleatoriamente sin experiencia. Si encuentran una mejor la guardan en la memoria y olvidan la anterior (búsqueda global)

Para lograr introducir el modelo de la selección de búsqueda de comida que lleve a emerger de una inteligencia colectiva del enjambre de abejas, primero necesitamos definir tres componentes esenciales: Fuentes de comida, recolectoras empleadas y desempleadas.

Figura 5. Comportamiento de las abejas en la búsqueda de néctar.



Fuente: [9]

- (1) Fuentes de comidas (A y B en la Figura. 5): Por simplicidad, el rendimiento de una fuente de alimento puede ser representado por una sola cantidad. En la función de optimización de nuestro problema las variables son establecidas mediante la información de la fuente de alimento:
 - Posición: Posible solución.
 - Cantidad de néctar: Función objetivo (calidad-fitness).
- (2) Recolectoras desempleadas: Son aquellas abejas que no tienen conocimiento sobre las fuentes de alimento y continuamente están en la búsqueda de una fuente de alimentación que explotar. Las hay de dos tipos: exploradoras y espectadoras.
- (3) Exploradoras (S en la Figura. 5): Son las abejas que espontáneamente inician la búsqueda de nuevas fuente de alimento sin un conocimiento previo. El porcentaje de abejas exploradoras varía desde 5% a 30% de acuerdo a la información del nido.
- (4) Espectadoras (R en la Figura. 5): Son las abejas que esperan en la colonia y buscan las fuentes de alimento a través de la información que las recolectoras empleadas les comparten, hay una mayor probabilidad de que las espectadoras elijan fuentes más productivas.

(5) Recolectoras empleadas: Se asocian con una fuente de alimento en particular, la cual explotan. Llevan con ellas la información de la productividad de la fuente y comparten esta información con determinada probabilidad. Después de cargar una porción de néctar de la fuente de alimento, se devuelven a la colonia y descargan el néctar al área de alimentos de la colonia. Hay tres posibles opciones para las abejas recolectoras de acuerdo a la cantidad residual de néctar:

- Si la cantidad de néctar disminuye a un nivel bajo o se agota, entonces la abeja abandona la fuente y se vuelve una recolectora desempleada. (UF en la Figura 5).

Si hay suficiente cantidad de néctar aún en la fuente:

- Puede continuar recolectando sin compartir la información de la fuente (EF2 en la Figura 5).
- Puede ir al área de baile y moverse para informar a sus compañeras sobre la fuente de alimento. (EF1 en la Figura 5).

3.3.2. Descripción del algoritmo (ABC)

Los pasos del algoritmo ABC son los siguientes:

Paso 1: Producción de fuentes de alimento iniciales por parte de las abejas empleadas.

Paso 2: REPEAT (se repite los pasos a partir de este punto)

Paso 2.1: Cada abeja empleada va a la fuente de alimento en su memoria y determina una fuente vecina, evalúa su cantidad de néctar y se mueve hacia la colonia

Paso 2.2: Cada espectadora ve el movimiento de las abejas empleadas y elige una de las Fuentes dependiendo de su movimiento. Después de elegir una vecindad alrededor, evalúa su cantidad de néctar.

Paso 2.3: Se determinan las fuentes abandonadas y se reemplazan con las nuevas Fuentes que las exploradoras han descubierto.

Paso 2.4: Se registra la mejor fuente de alimento encontrada hasta ahora.

Paso 3: UNTIL (hasta que los requerimientos sean satisfechos)

3.3.3. Proceso de hibridación ABC y QEA

El QEA provee de nuevas ideas que mejoran el tradicional AG. En primer lugar, la información en un cromosoma cuántico es mayor que en un cromosoma clásico, el tamaño de la población disminuye y la diversidad mejora. Segundo, la operación de mutación ya no es totalmente aleatoria sino que es dirigida por algunas reglas para hacer la siguiente generación mejor e incrementar la velocidad de convergencia.

Aunque el QEA tiene varias ventajas, aún tiene posibilidades de mejorar. En las pruebas, se encuentra que el QEA no siempre puede encontrar la mejor solución del problema. Lo cual significa que el algoritmo todavía tiene una considerable probabilidad de convergencia prematura. Para

resolver este problema, se propone un híbrido de QEA basado en el algoritmo ABC. Los parámetros son iguales que en el QEA básico.

El proceso del híbrido propuesto entre QEA combinado con el algoritmo ABC para la resolución de problemas complejos de optimización puede ser descrito a continuación:

Paso 1: Inicialización de parámetros: Se genera una población cuántica aleatoria, Los $Q(t)$, α_i^0 y β_i^0 $i=1,2,\dots,m$, de todos los $q_j^0 = q_j^t|_{t=0}$, $j=1,2,\dots,n$, son inicializados con $1/\sqrt{2}$. Lo cual significa que un individuo Q-bit, representa la superposición lineal de todos los posibles estados con la misma probabilidad. Se configuran otros parámetros: Probabilidad de entrecruzamiento clásico (P_c), probabilidad de mutación (P_m), máxima circulación de generación (\max_{gen}), número de población cuántica (n), el tamaño del cromosoma (L), y el límite de búsqueda ($Limit$)

Paso 2: Se genera una población clásica usando estos cromosomas cuánticos como la posición inicial de las abejas empleadas y se evalúa la función objetivo en cada individuo.

Paso 3: Se usa la operación de selección de padres en los cromosomas cuánticos por el método de ruleta, se hace el entrecruzamiento con la probabilidad de entrecruzamiento elegida. Seleccionamos las mejores soluciones para la siguiente generación y se actualiza la población cuántica.

Paso 4: Se usa la operación de mutación con la probabilidad elegida, de acuerdo a la mejor posición encontrada y se actualiza la población cuántica. Se produce una nueva población clásica, se evalúa la función objetivo de cada cromosoma, se compara con los anteriores y se elige el mejor para actualizar la población clásica con las nuevas posiciones de las abejas empleadas.

Paso 5: Cada espectadora elige una fuente de alimento dependiendo de la función objetivo de las abejas empleadas, produce una modificación en la posición de la fuente en su memoria y calcula su adaptabilidad ($fitness$). Siempre y cuando esta sea mayor que la anterior, la abeja memoriza la nueva posición y olvida la anterior

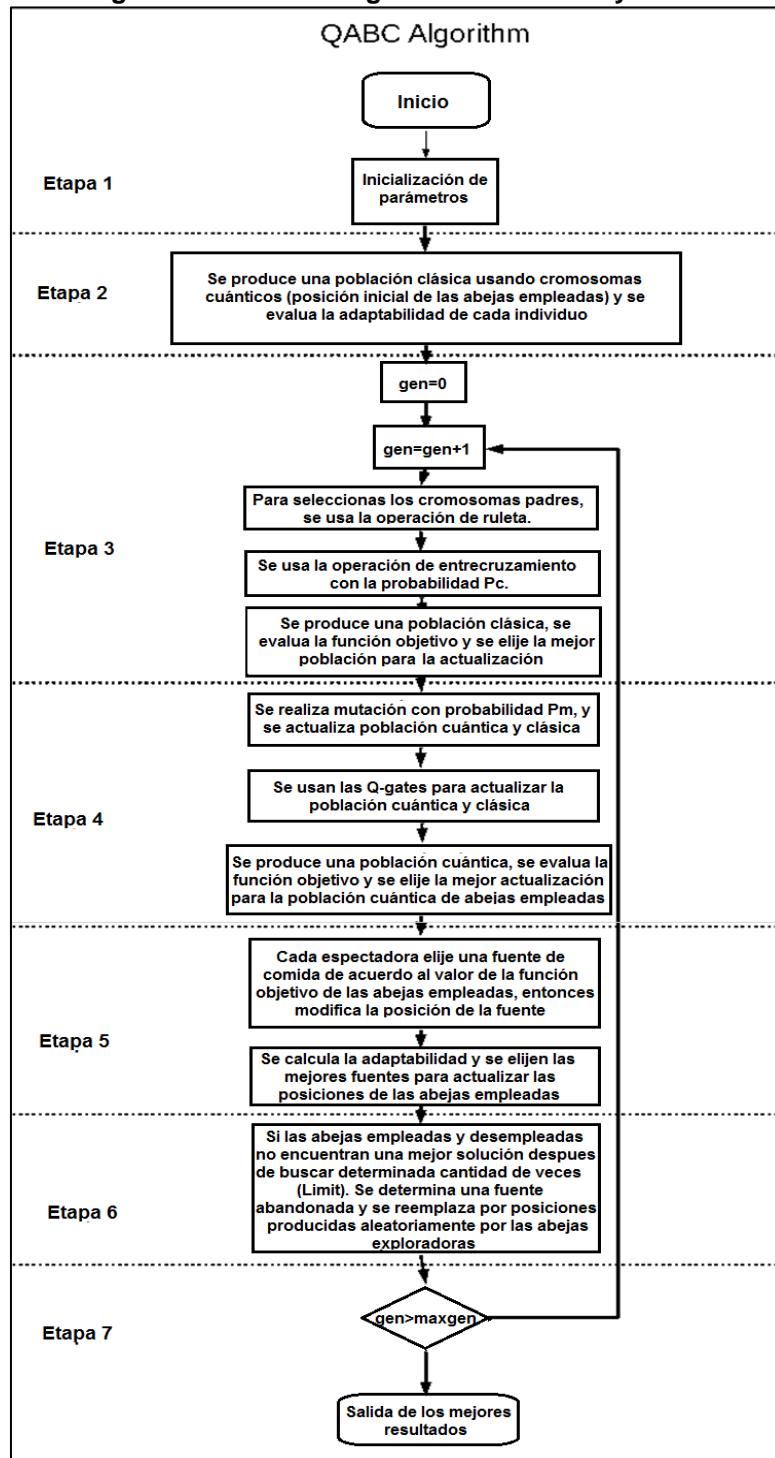
Paso 6: Si una abeja empleada y una desempleada siguen sin encontrar una mejor solución después de buscar cierta cantidad de veces ($Limit$), abandonan la fuente y aleatoriamente producen una nueva para reemplazar la abandonada por las exploradoras artificiales.

Paso 7: Se encuentra la mejor solución y se memoriza. Calculamos el valor promedio de las abejas empleadas.

Paso 8: Si el criterio de parada se satisface, el algoritmo se detiene y entrega la mejor solución, de lo contrario vuelve al paso 3

Se debe recordar que cada población clásica corresponde a una población binaria, por lo cual fue necesario hacer la conversión a decimal en el momento de evaluar la función objetivo, en el proceso que describe la Figura. 6

Figura 6. Pseudo-código del híbrido ABC y QEA.



Fuente: [9] con modificaciones del Autor

3.4. Aplicación a los pozos inteligentes

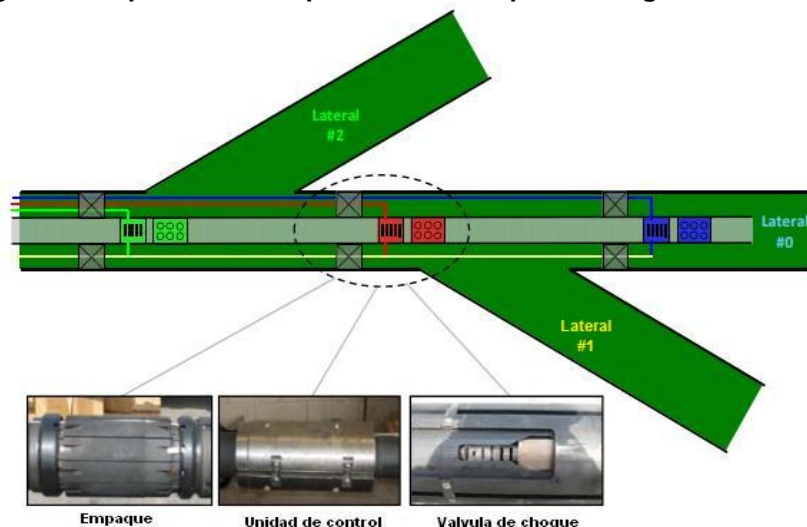
El diseño y planeación de pozos ha avanzado tremendamente durante las últimas dos décadas, desde el uso de pozos verticales convencionales hasta pozos horizontales no convencionales (Non-Conventional-Wells-NCWs) usando tecnología de perforación direccional. El rango de los pozos no convencionales se extiende desde simples pozos horizontales con un solo fondo a pozos complejos multilaterales con múltiples sub-lados (fishbone Wells).

Un pozo “inteligente” es considerado uno de los tipos más avanzados de pozos no convencionales. Suele estar equipado con una completación especial que tiene elementos de empaques y sellos que permiten la segmentación del pozo. Además, posee sensores de presión y temperatura en las válvulas de control (Inflow Control Valves - ICV) instaladas en la tubería de producción, como muestra la Figura 7. Los sensores permiten un monitoreo de presión y temperatura continuo mientras las ICVs proveen la flexibilidad para controlar cada rama del pozo multilateral independientemente. Un pozo inteligente puede ser un pozo multilateral donde cada lado es controlado por una ICV o una perforación simple donde cada segmento es controlado por una ICV. Las ventajas de los pozos inteligentes han sido demostradas en aplicaciones prácticas para producción de yacimientos simples y múltiples. Por su habilidad para controlar la producción de cada lado o segmento a través del ajuste y manipulación de ICVs, los pozos inteligentes pueden mitigar la producción de agua, localizando la tasa de producción óptima y por tanto incrementando el recobro final [2].

El simulador de yacimientos usado en este trabajo es ECLIPSE Schlumberger, el cual es usado ampliamente en la industria del petróleo, puesto que puede fácilmente implementar varios tipos de producción y restricciones económicas a través de los comandos existentes. Estas restricciones incluyen el límite económico de producción, el límite de la tasa de producción y el límite de la presión de fondo (BHP). ECLIPSE evalúa la función objetivo y determina que tan eficiente es la configuración de una ICV.

Se establece una comunicación entre ECLIPSE y MATLAB para que ECLIPSE pueda evaluar las configuraciones propuestas de las ICVs y así enviar los resultados de vuelta a MATLAB para continuar el proceso de optimización. [14]

Figura 7. Esquema de componentes de un pozo inteligente multilateral



Fuente: [14]

Representación de ICVs.

Eclipse incluye varias claves que se usan para restringir el flujo del fluido en el yacimiento. Las cuales son: WSEGVAlVE, WSEGLABY, y WSEGFILM. Todas las claves aplican el mismo mecanismo de restricción el cual impone una caída de presión por fricción adicional entre la fase de arena y la tubería. La elección de la clave depende del medio físico, en el cual se obtiene la restricción.

En este estudio, la clave elegida para representar las ICVs fue WSEGVAlVE. Esta clave representa una válvula subcrítica que impone una caída de presión adicional debido al flujo a través de una restricción con un área de corte transversal específica. La caída de presión a lo largo de una ICV se calcula usando un modelo homogéneo de flujo subcrítico a través de una tubería que contiene una restricción:

$$\delta P = \delta P_{\text{cons}} + \delta P_{\text{fric}} \quad (4)$$

Donde δP_{cons} es la caída de presión debido al flujo a través de la restricción y se calcula como:

$$\delta P_{\text{cons}} = C_u \frac{\rho v_c^2}{2C_v^2} \quad (5)$$

C_u , es una constante de conversión de unidades, la cual se muestra en la tabla 4.

Tabla 4. Constante de conversión de unidades C_u .

Sistema de unidades	C_u	Densidad	Velocidad	Presión
METRIC	1,00E-05	kg/m ³	m/s	bar
FIELD	2,16E-04	lbm/ft ³	ft/s	psi
LAB	9,87E-07	gm/cm ³	cm/s	atm
PVT-M	9,87E-06	kg/m ³	m/s	atm

Fuente: [12]

ρ , es la densidad del fluido combinado.

v_c , es la velocidad de flujo de la mezcla a través de la restricción,

C_v , es el coeficiente adimensional del fluido para la válvula.

δP_{fric} , es una caída de presión adicional debida al fluido en el lateral horizontal. Se calcula usando la expresión estándar para la caída de presión por fricción de flujo homogéneo a través de la tubería.

$$\delta P_{\text{fric}} = 2C_u f \frac{L}{D} \rho v_p^2 \quad (6)$$

f , es el factor de fricción de Fanning

L , es la longitud adicional de la tubería en el segmento

D , es el diámetro de la tubería sin restricción

v_p , es la velocidad de flujo de la mezcla a través de la tubería.

Obviamente v_c y v_p dependen de las áreas de corte transversal respectivas a la tubería y la restricción, y de la tasa de flujo volumétrico de la mezcla a través del segmento (q_m).

$$q_m = v_c A_c = v_p A_p \quad (7)$$

La caída de presión total (δP) es una función de la caída de presión por fricción debida al flujo en el lateral horizontal, por lo tanto depende directamente de la configuración de la ICV, la cual se controla cambiando el área transversal de la ICV (A_C).

Las ICVs deben ser diseñadas para cada yacimiento de forma separada, pues cada uno tiene características diferentes de porosidad y permeabilidad, por lo tanto tendrá diferentes tasas de producción.

En la figura 8 se obtiene un fragmento del archivo de ECLIPSE valves.inc, el cual representa en parte las ICVs en este trabajo. La cuarta cifra es el área de la ICV, y se modifica continuamente por medio del algoritmo de MATLAB para cambiar las configuraciones de las ICVs

Figura 8. Extracto de la definición de parámetros de las ICVs en ECLIPSE

```

WSEGVAlV

TWC 7 0.7 1e-006 /
TWC 13 0.7 0.00017001 /
TWC 18 0.7 4.3911e-006 /
TWC 24 0.7 1e-006 /
/
    
```

Fuente: Modelo de pozo confidencial.

Cada columna en las líneas del archivo tiene un significado

1. Nombre del pozo, el cual debe tener definidos previamente sus segmentos. Con la clave WELSEGS.
2. Numero de segmento dentro de la válvula, es un número entero que va desde 2 hasta el NSEGMX (cantidad de segmentos) especificado en la clave WSEGDIMS.
3. El coeficiente adimensional de flujo para la válvula.
4. El área de corte transversal para el fluido en la restricción,

3.4.1. Estructura de optimización

La Figura 9 muestra un diagrama de flujo del proceso de optimización. El cual indica el modo en que MATLAB se comunica con ECLIPSE para enviar y recibir los parámetros de optimización.

Paso 1: Se genera una población la cual se encuentra de forma cuántica, binaria y finalmente se tiene el individuo que se convierte en la cadena decimal correspondiente.

Paso 2: Al decodificar al modo decimal se encuentra el número de configuración de ICV deseado. Por ejemplo, una cadena binaria de nueve bits se convierte en una cadena decimal de tres partes la cual corresponde a tres configuraciones de ICV.

Paso 3: Dado que la mayoría del tiempo de cómputo de la optimización se gasta evaluando la función objetivo (i.e. la simulación del yacimiento), se crea una librería para cada solución posible (configuración de ICV). La librería contiene cualquier configuración ICV que se propone mediante el algoritmo en MATLAB y el resultado de su correspondiente función objetivo. Antes de que las configuraciones de la ICV sean enviadas al simulador, el código de optimización revisa la librería en busca de cualquier configuración de la ICV idéntica realizada antes. Si estas configuraciones

están disponibles, el simulador no tendrá la necesidad de operar y el tiempo de cómputo se ahorrará.

Paso 4: Si las configuraciones de ICV no están disponibles, se escribirán en un archivo de entrada de texto para pasar a Eclipse.

Paso 5: Eclipse opera con las configuraciones de ICV suministradas.

Paso 6: Una vez la simulación termine, el algoritmo de MATLAB lee los resultados simulados. i.e. en este problema son la producción de petróleo acumulado y producción de agua acumulada

Paso 7: Se calcula la función objetivo en MATLAB.

Paso 8: Los pasos anteriores empezando por el paso 2, se repiten para cada individuo en la población antes de que el proceso de selección empiece.

Paso 9: En este momento, las configuraciones de ICV y el valor correspondiente a su función objetivo se guardaran en una librería para usar en el futuro.

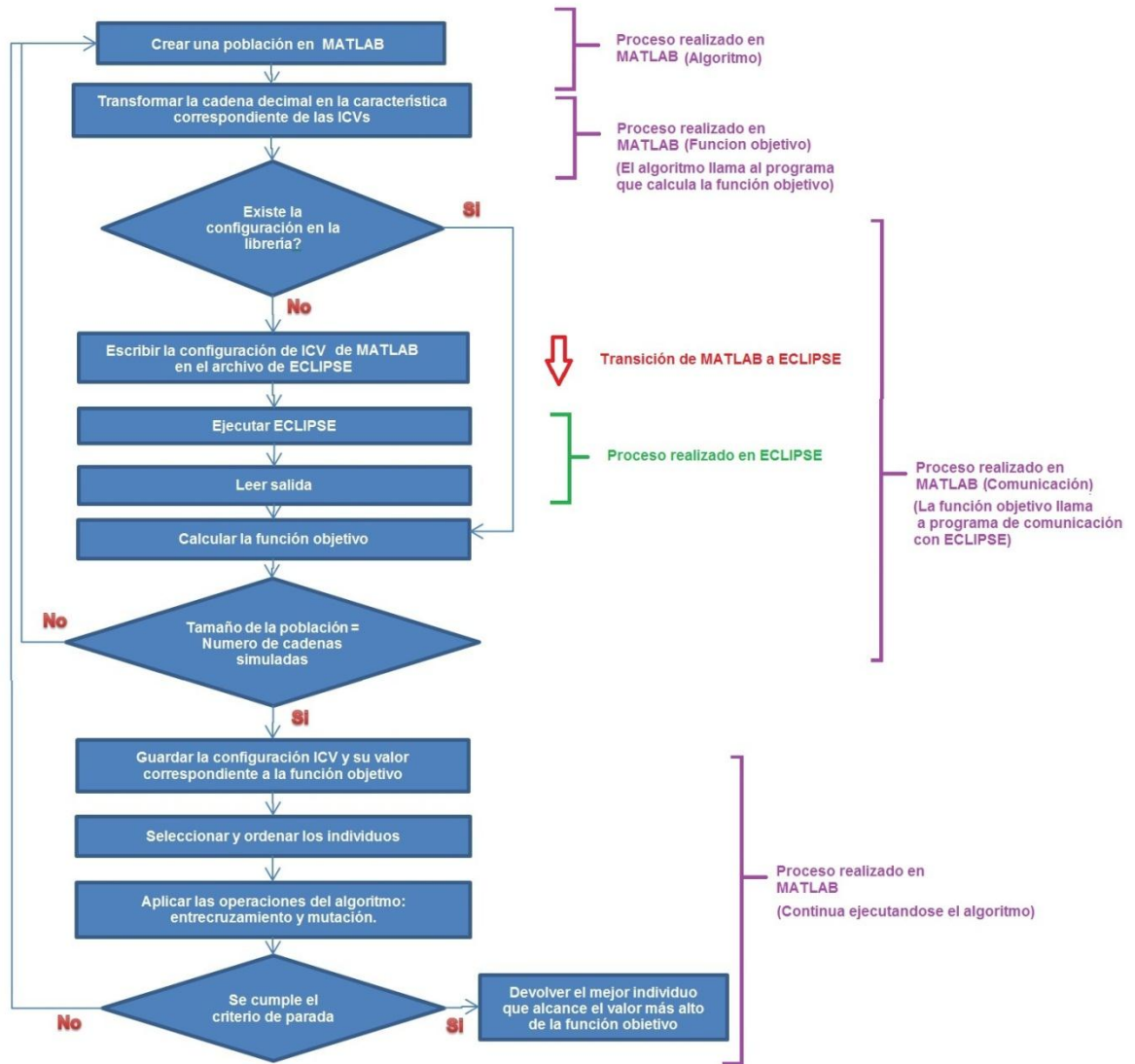
Paso 10. Una vez se simulen todos los individuos, el algoritmo clasifica estos individuos basado en su función objetivo y selecciona los individuos que contribuirán a la creación de la siguiente generación.

Paso 11: Se llevan a cabo todas las operaciones del algoritmo empezando por el entrecruzamiento y la mutación en la etapa correspondiente al QEA y la búsqueda limitada por las abejas exploradoras, las cuales se aplican a los individuos seleccionados para la siguiente generación.

Paso 12: Se repite el proceso completo para las siguientes generaciones hasta que se cumpla con el criterio de terminación.

Paso 13: El algoritmo finalizara si se alcanza el número de generación especificada al principio o si no hay alguna mejora en la función objetivo durante 50 generaciones consecutivas.

Figura 9. Diagrama de flujo de la estructura de optimización



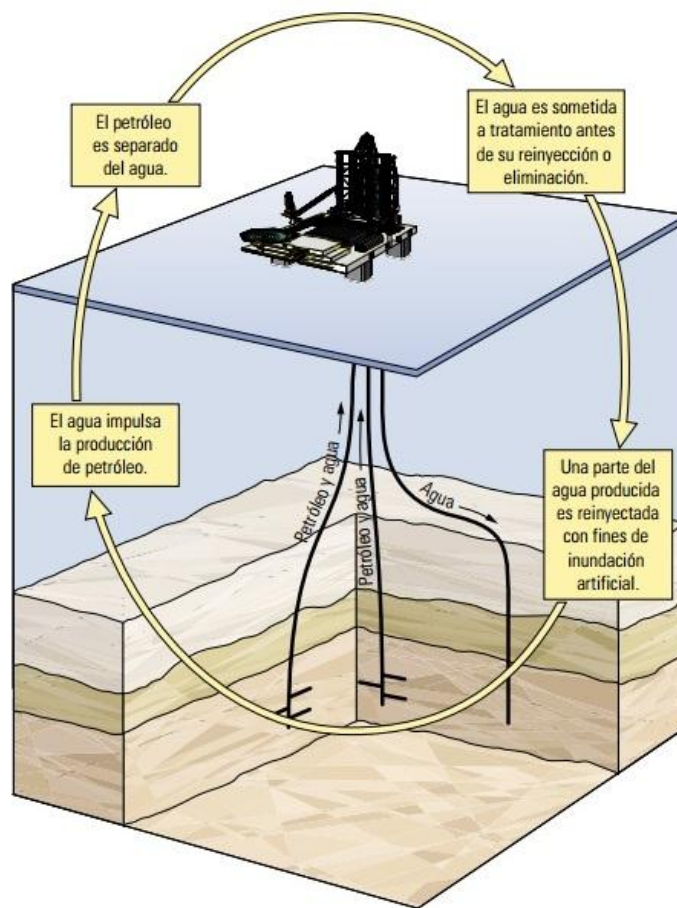
Fuente [14]

Las opciones avanzadas de administración de campos en el software ECLIPSE soportan las claves de accionamiento (ACTION keywords) para yacimientos flexibles y control de pozos. Cantidades definidas por el usuario (User-defined quantities UDQ), argumentos definidos por el usuario (user-defined arguments UDA), y tablas definidas por el usuario (user-defined tables UDT) extienden enormemente esta capacidad. Esto permite, por ejemplo, el control por medio de consideraciones económicas, objetivos del pozo compuesto, y facilidad en la especificación de restricciones. Esto extiende todo el rango de controles tradicionales que existe en ECLIPSE para permitir el modelamiento de yacimientos complejos y controles de pozo.

Las UDQs son funciones matemáticas que proveen extensibilidad significativa en el ejercicio de la lógica condicional. Se puede usar el compendio de vectores para definir uno nuevo. Se puede usar entonces el nuevo como un conjunto de argumentos en las claves. Por ejemplo, se pueden usar las UDQs para establecer parámetros económicos y decidir si es más rentable inyectar vapor de alta o baja calidad.

Para optimizar la operación se debe definir la función objetivo. Para este ejemplo se analiza la rentabilidad, la cual es una operación de maximización. Básicamente se obtiene un valor equivalente a la producción de petróleo y suele perderse dinero con la producción de agua.

Figura 10. Ciclo del agua en un pozo de petróleo.



Fuente: [4]

La figura 10 ilustra claramente el rol del agua en el proceso de producción de petróleo. El agua suele considerarse como un mal necesario en los campos maduros. Si bien a menudo impulsa la producción primaria e interviene en la producción secundaria, al barrer las arenas petrolíferas desplazando al petróleo y generando su flujo. La producción en exceso de agua genera que el proceso de reinyección, su tratamiento y eliminación represente pasivos y costos de producción de petróleo significativos para las compañías productoras de petróleo y gas.

La ecuación que se tendrá en cuenta corresponde a la siguiente:

$$FNPV1 = \sum_p FOPT * Vp - \sum_e FWPT * Ve \tag{8}$$

FOPT: Flujo de producción de petróleo
Vp: Valor de producción de petróleo
FWPT: Flujo de producción de agua
Ve: Valor de producción de agua

En el modelo simulado encontramos el archivo .data el cual contiene una definición completa de características del pozo, junto a la especificación de la función objetivo que muestra la figura 11:

Figura 11. Extracto de la definición de parámetros en ECLIPSE

```
UDQ
-- A COUPLE OF FIELD QUANTITIES
DEFINE FUNPV1 (60*SUM(FOPT)-4*SUM(FWPT)) /
UNITS FUNPV1 USD /
```

Fuente: Modelo de pozo confidencial.

Reemplazando los valores específicos de la función de optimización de la producción del pozo modelado, en la ecuación 4:

$$FNPV1 = \sum_p FOPT * 60 - \sum_e FWPT * 4 \tag{9}$$

Donde 60 será el precio del petróleo y 4 para el costo del agua. Finalmente el objetivo es incrementar la producción de petróleo y reducir la producción de agua.

3.5. Resultados Experimentales

Inicialmente para demostrar su comportamiento, se llevan a cabo experimentos con la función de objetivo Rosenbrock. Se realizan pruebas con los algoritmos QEA y con el híbrido QABC. En ellas, se toman cuatro ejecuciones para las funciones, y los resultados se muestran en las tablas de esta sección.

Los dos algoritmos se han programado en MATLAB y se implementaron en un PC de 3GB de RAM con el sistema operativo Windows 7.

3.5.1. Función Rosenbrock

Como primer ejemplo se toma la función objetivo, dada por:

$$f(x) = \sum_{i=1}^{10} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \tag{10}$$

$$\text{ObjVal} = \text{sum}((100*(\text{Mat2}-\text{Mat1}.\wedge2) .\wedge2+(\text{1}-\text{Mat1}) .\wedge2) ') ' ; \tag{11}$$

Se establecen los parámetros en la tabla 5, de la siguiente manera:

Tabla 5. Definición de parámetros de ambos algoritmos para la función Rosenbrock.

Parámetro	Significado	Algoritmo	
		QEA	QABC
n	Tamaño de la población	20	20
up	Límite superior de las variables	4	4
low	Límite inferior de las variables	-4	-4
pric	Cantidad de cifras significativas de la conversión binaria a decimal	2	2
nvar	Número de variables de la función objetivo	2	2
Pcc	Probabilidad de entrecruzamiento	0,8	0,8
Pm	Probabilidad de mutación	0,02	0,02
maxgen	Número máximo de generaciones	100	100
Limit	Límite de búsqueda, si la solución no mejora.	n/a	30

Fuente: Autor

Se debe recordar en la tabla 5, que el parámetro "limit" es exclusivo de la optimización generada por el algoritmo ABC, por lo tanto no se utiliza este valor en el algoritmo QEA.

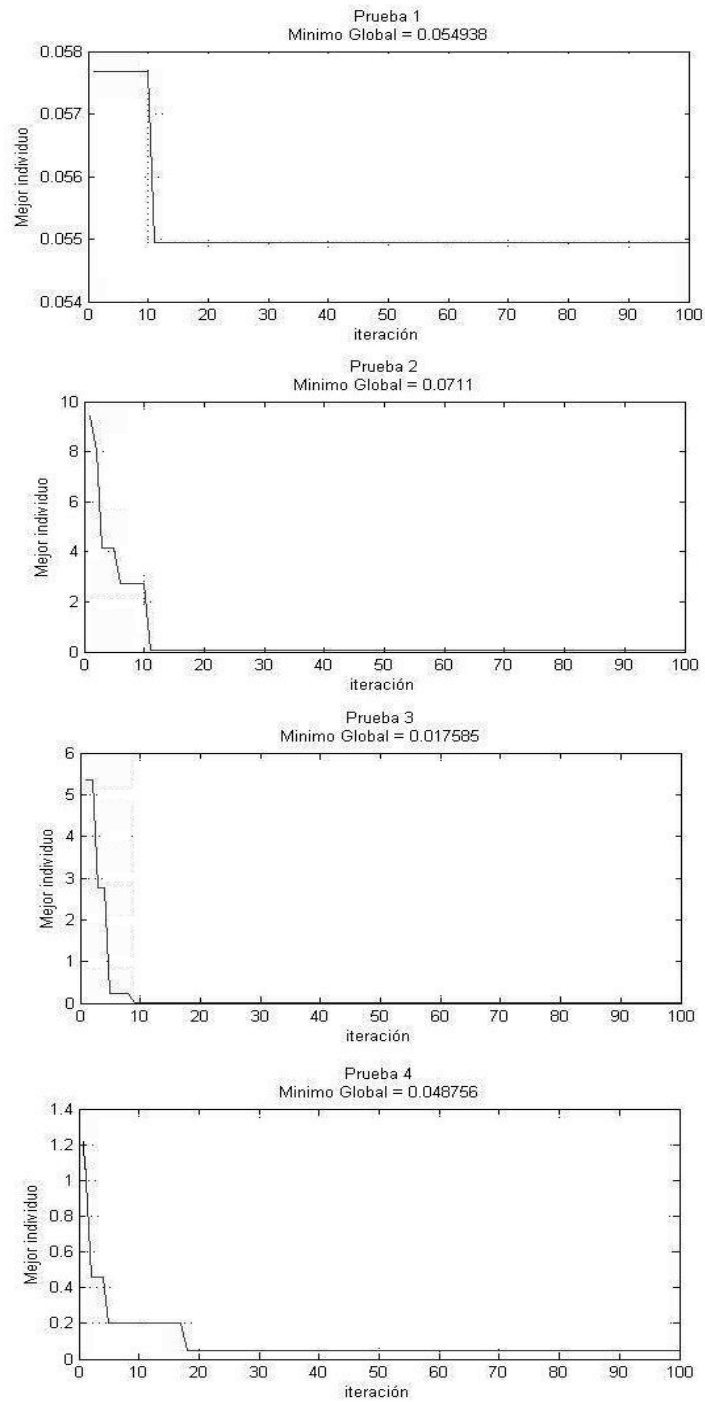
Las figuras 12 y 13 muestran las curvas de evolución del mejor individuo y del promedio para las pruebas realizadas con el algoritmo QEA. Las figuras 14 y 15 muestran las curvas de evolución del mejor individuo y del promedio para las pruebas realizadas con el algoritmo QABC. Finalmente la tabla 6 muestra el mínimo global para cada ejecución y el promedio total de las pruebas. Al comparar el óptimo global en la tabla 6, es evidente que el híbrido propuesto puede encontrar mejores soluciones que el QEA al resolver problemas de optimización. Además, el híbrido tiene un muy buen comportamiento con la habilidad de encontrar la solución óptima y una velocidad de convergencia muy rápida.

Tabla 6. Comparación del mínimo global con 100 iteraciones de ambos algoritmos para la función Rosenbrock.

QEA		QABC	
Iteraciones	100	Iteraciones	100
Ejecución	Mínimo global	Ejecución	Mínimo global
1	0,05494	1	0,00219
2	0,07110	2	0,00948
3	0,01759	3	0,01037
4	0,04876	4	0,00260
Promedio	0,05	Promedio	0,01

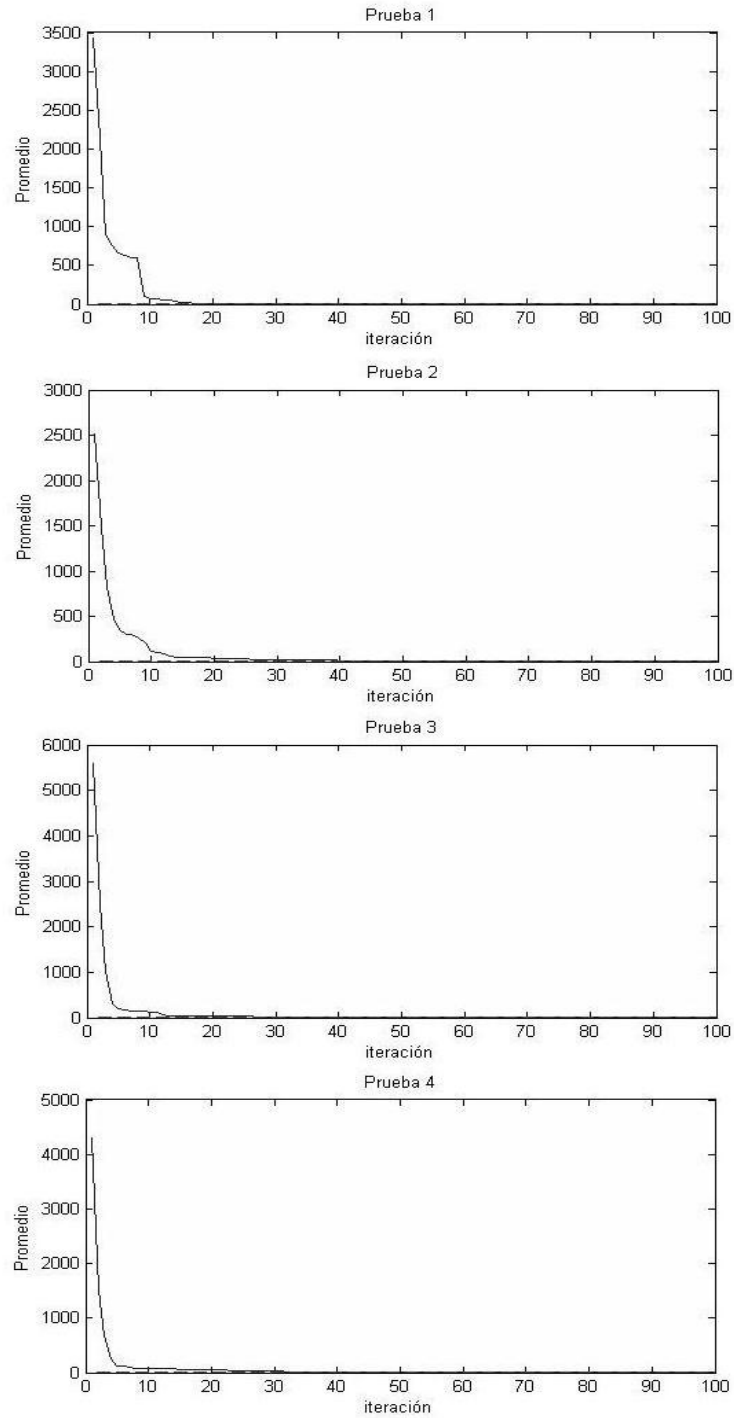
Fuente: Autor

Figura 12. Algoritmo QEA con 100 iteraciones para la función Rosenbrock (Mejor individuo)



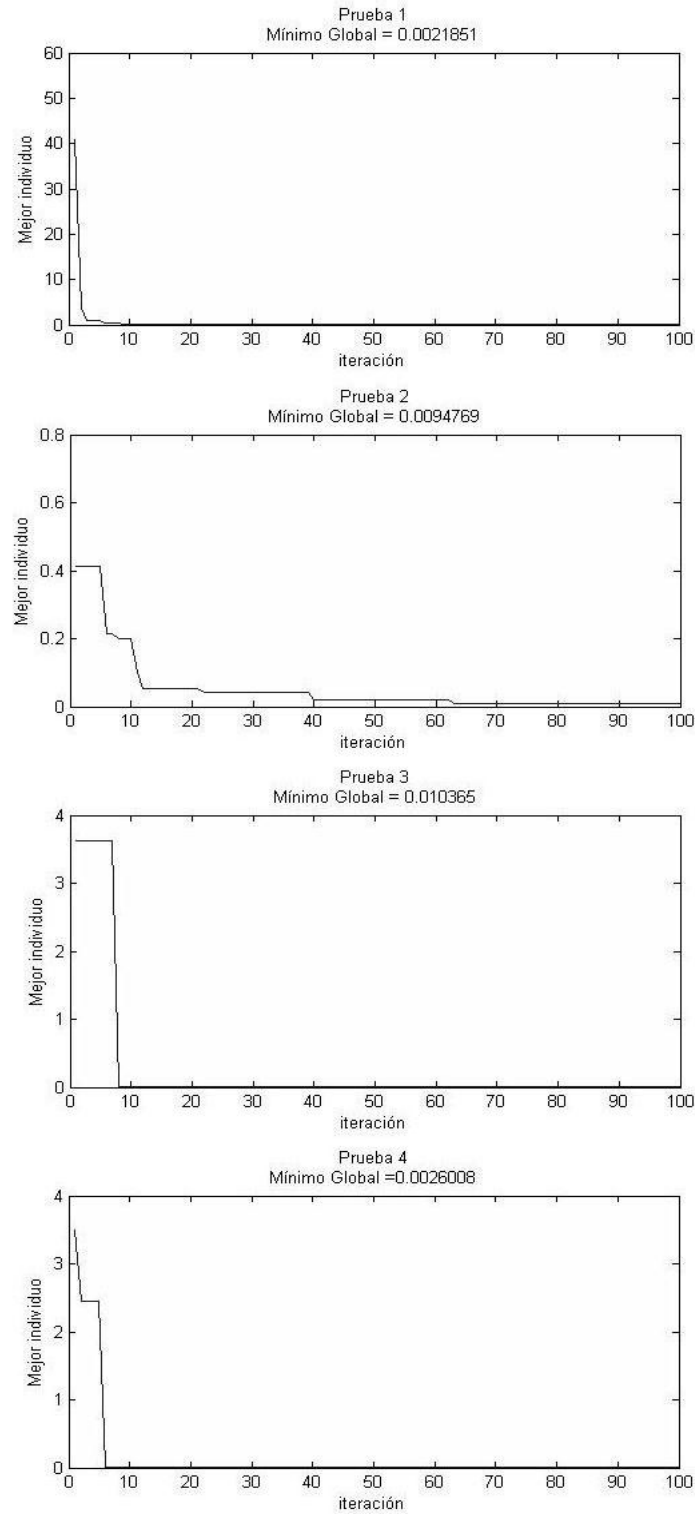
Fuente: Autor

Figura 13. Algoritmo QEA con 100 iteraciones para la función Rosenbrock (Promedio)



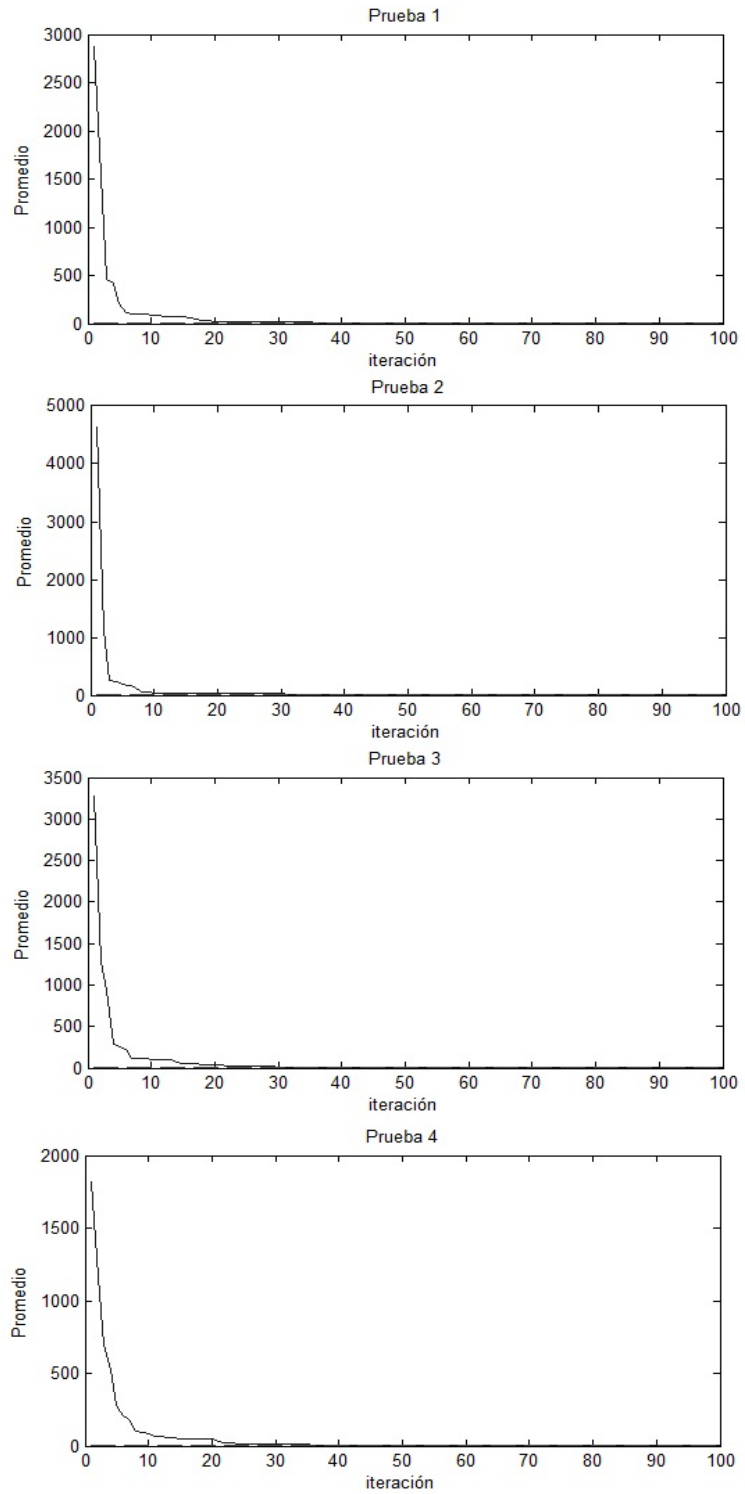
Fuente: Autor

Figura 14. Algoritmo QABC con 100 iteraciones para la función Rosenbrock (Mejor Individuo)



Fuente: Autor

Figura 15. Algoritmo QABC con 100 iteraciones para la función Rosenbrock (Promedio)



Fuente: Autor

3.5.2. Función objetivo: Optimización del pozo

Hemos definido anteriormente en la ecuación 9 la función objetivo a evaluar, pero a continuación nombramos una serie de consideraciones preliminares.

- Ya que el cálculo del mejor valor se evalúa como un mínimo en ambos algoritmos, y estamos haciendo una operación de maximización, la función de MATLAB que enlaza el algoritmo con el software ECLIPSE se multiplicará por menos uno para obtener los valores apropiados.
- Las restricciones en las ICVs del modelo nos llevan a acordar con anterioridad que su rango de acción estará variando entre límites de 1 a 7, en donde 1 corresponde al cierre, y 7 a la apertura de las válvulas.
- Se establecen los parámetros de la siguiente manera: $n = 10$, $L = 80$, $\text{bit}_n = 3$, $P_{cc} = 0.9$, $P_m = 0.02$, $\text{maxgen} = 100$, $\text{Limit} = 30$. Y como hemos dicho anteriormente los límites predefinidos serán $\text{low} = 1$, $\text{up} = 7$.
- Al realizar el análisis de la función objetivo que maneja la optimización del modelo, se toma como base una configuración en la cual todas las válvulas están abiertas.
- Se analiza la rentabilidad de la mejor configuración de las ICVs durante un periodo de 20 años.

Ya que el objetivo es producir la mínima cantidad de agua, que reduzca el costo de la separación de su separación del petróleo mediante una configuración óptima de ICVs. En las tablas 7 y 8 se encuentran los valores de comparación de ambos algoritmos. Los resultados muestran que el algoritmo QABC tiene mejorías considerables respecto al algoritmo QEA.

Las curvas de evolución de las figuras 16 y 17 si bien no reflejan a simple vista la mejoría en la optimización, para un mejor entendimiento se subrayan los lugares de mayor optimización, denotando la diferencia que se encuentra entre el resultado del algoritmo respecto al resultado del ejemplo base, siendo extraídos de la tabla 8. En el eje Y se puede ver apreciar los valores alcanzados de rentabilidad para cada prueba respecto al caso base.

La evolución de la optimización del algoritmo QEA como se ve en la figura 16, no es tan eficiente, como lo es en el algoritmo QABC de la figura 17. El cual tendrá un aumento considerable al concluir el periodo que se tiene en cuenta en la prueba

Se puede notar que el comportamiento de búsqueda del QABC tiene un proceso de ajuste un poco más uniforme, pues las pruebas realizadas describen una evolución similar, a diferencia del caso de la prueba 3 en los primeros años, y la prueba 2 en los últimos años, en el algoritmo QEA de la figura 16.

Las pruebas 3 y 4 del QABC tienen el mayor valor de optimización de todas las pruebas, y se destacan en la figura 17, sin embargo la prueba 2 también tendrá un comportamiento no uniforme en los últimos años, sin llegar a ser la prueba con mayor rendimiento.

Tabla 7. Comparación de resultados de la optimización del pozo para ambos algoritmos respecto al caso base.

Tiempo	FUNPV1 - USD(*10**3)									
	Año	Base	QEA				Híbrido			
			1	2	3	4	1	2	3	4
0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	
1	21.480,7	21.480,7	21.579,4	21.522,1	21.512,1	22.463,9	22.440,7	22.386,5	22.411,7	
2	42.533,5	42.647,9	42.628,0	42.745,3	42.775,9	42.395,9	43.407,0	43.624,3	43.864,5	
3	62.543,7	62.610,9	62.707,5	63.231,7	62.952,4	63.431,5	63.211,0	63.303,9	63.465,9	
4	81.831,2	82.356,2	81.842,0	83.836,5	81.995,0	84.825,6	84.989,5	85.704,7	85.444,0	
5	100.864,3	101.937,7	101.983,1	103.632,8	101.657,7	105.943,0	105.453,3	104.427,1	104.928,5	
6	119.738,1	120.369,2	120.644,9	120.612,2	120.732,5	120.972,8	121.759,5	121.573,4	120.886,6	
7	138.058,6	138.525,7	138.828,1	138.737,2	138.658,7	139.678,3	139.596,9	139.666,2	139.689,6	
8	155.308,9	155.474,0	156.065,5	155.611,6	155.929,3	156.596,6	156.283,5	156.778,6	156.561,4	
9	170.994,3	171.061,2	171.108,4	171.101,3	171.329,5	172.468,7	171.842,5	172.186,2	172.163,4	
10	185.018,3	185.085,2	185.553,7	185.247,2	185.114,6	186.023,6	186.270,3	186.774,4	187.025,0	
11	197.597,4	197.644,0	198.350,9	198.393,1	198.029,1	198.813,4	199.260,5	199.451,9	199.680,5	
12	208.980,0	209.378,8	209.141,0	209.336,1	208.996,2	209.860,1	210.302,5	209.712,6	210.138,9	
13	219.371,0	220.093,4	220.088,7	219.912,9	220.434,2	220.757,6	221.268,5	221.745,1	221.120,5	
14	228.891,9	228.939,5	228.954,8	229.238,9	229.435,2	229.476,7	229.492,3	229.091,1	229.648,6	
15	237.669,2	237.675,7	237.914,6	237.903,3	237.832,7	238.490,9	237.828,4	238.676,3	238.795,6	
16	245.644,2	246.304,9	245.678,4	245.887,0	245.683,1	246.322,5	246.403,8	246.769,6	246.662,5	
17	253.032,7	254.825,5	258.057,2	253.455,1	253.989,8	255.216,5	258.661,9	255.764,0	254.765,8	
18	259.862,5	260.943,3	264.118,2	260.856,4	260.084,4	261.009,3	264.744,0	262.647,0	261.303,9	
19	266.155,6	266.568,5	267.393,6	267.398,1	266.734,4	268.058,2	270.382,1	267.184,6	266.822,0	
20	271.984,4	271.995,3	272.132,2	272.782,7	272.551,6	272.974,3	273.984,1	275.911,0	274.924,4	

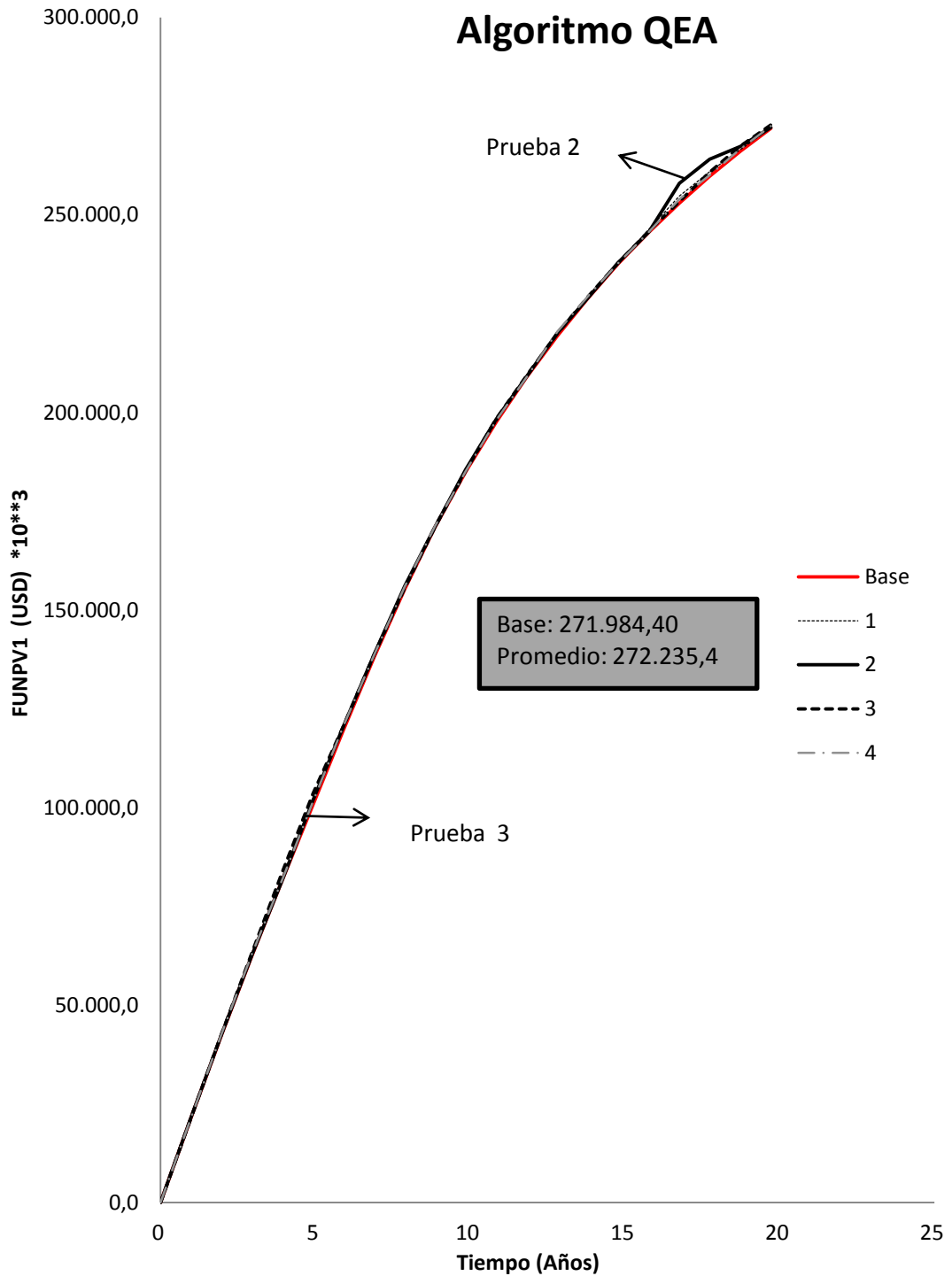
Fuente: Autor

Tabla 8. Comparación de resultados de la optimización del pozo para ambos algoritmos respecto al caso base en promedio.

Tiempo	FUNPV1 - USD(*10**3)		
Año	Base	QEA	Hibrido
0	0,0	0,0	0,0
1	21.480,7	21.523,6	22.425,7
2	42.533,5	42.699,3	43.322,9
3	62.543,7	62.875,6	63.353,1
4	81.831,2	82.507,5	85.241,0
5	100.864,3	102.302,8	105.188,0
6	119.738,1	120.589,7	121.298,1
7	138.058,6	138.687,4	139.657,8
8	155.308,9	155.770,1	156.555,0
9	170.994,3	171.150,1	172.165,2
10	185.018,3	185.250,2	186.523,3
11	197.597,4	198.104,3	199.301,6
12	208.980,0	209.213,0	210.003,5
13	219.371,0	220.132,3	221.222,9
14	228.891,9	229.142,1	228.677,2
15	237.669,2	237.831,6	238.447,8
16	245.644,2	245.888,4	246.539,6
17	253.032,7	255.081,9	256.102,1
18	259.862,5	261.500,6	262.426,1
19	266.155,6	267.023,6	268.111,7
20	271.984,4	272.365,4	274.448,5

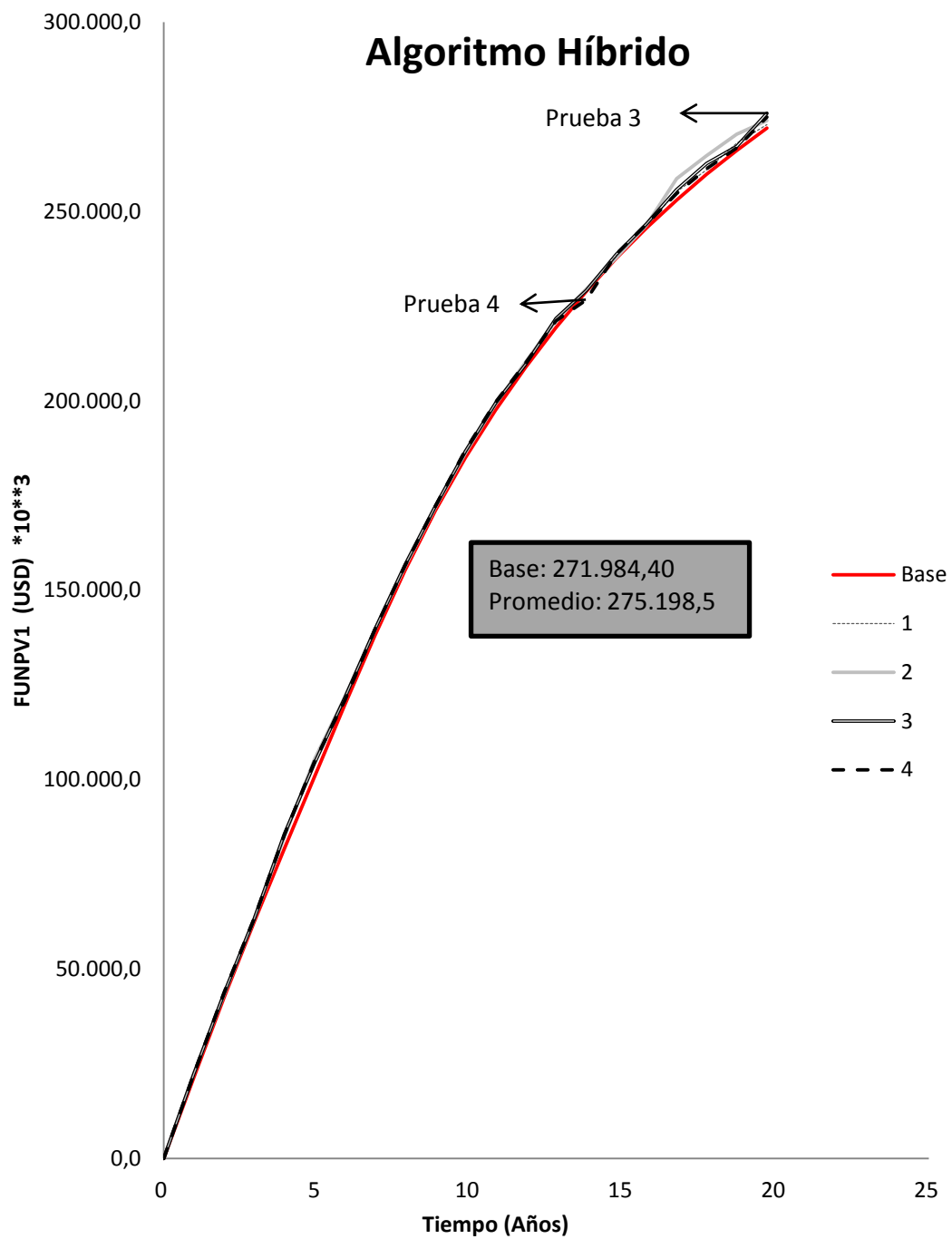
Fuente: Autor

Figura 16. Optimización de la producción con el algoritmo QEA en ECLIPSE



Fuente: Autor

Figura 17. Optimización de la producción con el algoritmo QABC en ECLIPSE



Fuente: Autor

CONCLUSIONES Y RECOMENDACIONES

1. En este trabajo se desarrolló un algoritmo híbrido llamado (Quantum Artificial be colony algorithm -QABC) para la solución de problemas de optimización. Luego de ejecutar una serie de pruebas, se compararon los datos con aquellos obtenidos a través del algoritmo (Quantum inspired evolutionary algorithms - QEA), encontrando que el algoritmo desarrollado es más práctico y efectivo al momento de resolver problemas complejos, y también es un método viable para el problema de optimización de la producción de pozos inteligentes.
2. Se encontró que la principal mejora del algoritmo QABC, respecto al QEA, obedece al perfeccionamiento de la búsqueda local, por medio de la simulación del comportamiento de las abejas espectadoras, y exploradoras las cuales permiten mutaciones específicas para cada posición, impidiendo que alguna de las posibilidades se pierda en la búsqueda, dando como resultado una respuesta más óptima.
3. Por medio de una función usualmente aplicada en la validación de métodos numéricos y algoritmos computacionales, como lo es la función Rosenbrock, se apreció un desempeño sobresaliente del algoritmo híbrido QABC, el cual consistentemente mejora la proximidad al óptimo global.
4. Las pruebas propuestas para la optimización de la producción, demostraron que ambos algoritmos logran maximizar la rentabilidad. Sin embargo siempre se encuentra que el algoritmo híbrido logra encontrar las mejores configuraciones que generan un aumento de la producción de petróleo, con menores costos. Se debe tener en cuenta que parte de los resultados y las herramientas se mantienen en confidencialidad, debido a su uso industrial y a las normas que rigen la adquisición de datos de los pozos de petróleo en general.
5. En la actualidad el mejoramiento de las técnicas de manejo de la producción de agua permite minimizar el volumen de agua llevado a la superficie, convirtiendo el excedente de agua producida de residuo en recurso reutilizable. Por consiguiente, el método de configuración eficiente de las (Inflow Control Valves – ICV) por medio de un algoritmo de optimización, responde a una necesidad primordial en la industria y podría ser considerado un análisis más profundo a otras posibles aplicaciones relacionadas como objeto de trabajos de grado futuros.
6. El tiempo de cómputo de la optimización de la producción obtenido en un computador de uso personal, puede tardar entre ocho y diez horas. En pro de disponer de un mayor rango de datos y realizar análisis más profundos. Se recomienda optar por incorporar el sistema operativo y las licencias necesarias para aprovechar la capacidad del Cluster (computadores usados en conjunto mediante la utilización de un hardware común, que se comportan como si fuesen una única computadora) que se encuentra en el edificio del CENTIC.
7. Se concluye finalmente, que los objetivos previstos han sido alcanzados, pues se ha desarrollado satisfactoriamente un híbrido basado en un algoritmo inspirado en la cuántica que incrementa las habilidades de búsqueda de la mejor respuesta, el cual, al momento de ser aplicado a la optimización de la producción del modelo de un pozo inteligente, logró demostrar su viabilidad y eficiencia.

REFERENCIAS

- [1] A. Manju and M. J. Nigam, "Applications of quantum inspired computational intelligence: a survey," *Artificial Intelligence Review*, May 2012.
- [2] B. Yeten, "Optimum deployment of nonconventional wells," Ph.D dissertation, Stanford University, 2003.
- [3] C. Grosan and A. Abraham, "Hybrid Evolutionary Algorithms: Methodologies , Architectures , and Reviews," Springer-Verlag Berlin Heidelberg publisher, vol. 17, 2007.
- [4] D. B. Burnett, T. J. Feeley, L. N. De, M. Hightower, and M. Lavery, "Manejo de la producción de agua: De residuo a recurso.," *Oilfield Review*, 2004.
- [5] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, Aug. 2009.
- [6] E. Zuazua, "Teoría matemática del control: Motor del desarrollo científico, tecnológico y social." *Lecture Notes in applied mathematics*, Madrid, 1995.
- [7] G. K. Venayagamoorthy, "Advanced Computational Intelligence for Identification, Control and Optimization of Nonlinear Systems," 2007. [Online]. Available: http://ewh.ieee.org/cmte/cis/mtsc/ieeecis/tutorial2007/Ganesh_Kumar_Venayagamoorthy_2007.pdf.
- [8] G. Ochoa, "Introducción a la Computación Evolutiva y la Morfogénesis Artificial," *Lecture Notes in in Computer Science*, University of Nottingham, Universidad Simón Bolívar, 2003.
- [9] H.-B. Duan, C.-F. Xu, and Z.-H. Xing, "A hybrid artificial bee colony optimization and quantum evolutionary algorithm for continuous optimization problems.," *International journal of neural systems*, vol. 20, no. 1, Feb. 2010.
- [10] J. M. Sánchez, *Historia de la física cuántica*. Madrid: Libro de Editorial Dracontos, 2001.
- [11] K. Han and J. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, Dec. 2002.
- [12] R. Manual, T. Description, B. D. List, C. D. List, U. Guide, T. Description, N. W. Modeling, R. Notes, and I. Guide, "Simulation Software Manuals 2009 . 1."
- [13] "Técnicas de computación evolutiva," 1984. [Online]. Available: <http://www.fing.edu.uy/inco/cursos/geneticos/ae/CE.pdf>.
- [14] Z. M. Alghareeb, S. Aramco, and R. N. Horne, "Proactive Optimization of Oil Recovery in Multilateral Wells Using Real Time Production Data," *SPE journal*, vol. 124999, 2009.