

Modelo de algoritmo evolutivo de inspiración cuántica QIEA- \mathbb{R} con control de interacción entre universos

José Carlos Delgado Ramos

Escuela Profesional de Ciencia de la Computación
Universidad Católica San Pablo

13 de diciembre de 2014

Computación Cuántica

- Campo de investigación con mucho futuro, posibilidades completamente inimaginables para el paradigma tradicional en potencia y velocidad.
- Inexistencia de ordenadores cuánticos utilizable a día de hoy.
- Dificultad en el desarrollo de nuevos algoritmos.

Computación de Inspiración Cuántica

- Busca aprovechar algunas de los elementos propios de la Computación Cuántica a algoritmos de computación tradicional.
- Incorpora conceptos tales como: *qubits*, interferencia y superposición de estados, entre otros.

Algoritmos Evolutivos de Inspiración Cuántica en \mathbb{R}

- Busca generar un conjunto de estados observables continuos y no discretos como el algoritmo base.
- La inspiración de este modelo está en el uso de funciones de onda.

Algoritmos Evolutivos de Inspiración Cuántica en \mathbb{R}

- Los individuos cuánticos son reemplazados: pasan a ser funciones aleatorias de probabilidad $\mathbb{U}_{ij} \in [l_{uj}, u_{uj}]$ donde l_{uj}, u_{uj} marcan los límites superiores e inferiores del intervalo que representa al espacio de búsqueda, respectivamente.

Algoritmos Evolutivos de Inspiración Cuántica en \mathbb{R}

- \mathbb{U}_{ij} puede ser representado por el centro $\mu_{ij} = l_{uj} + u_{ij}/2$ mas el ancho del pulso $\sigma_{ij} = u_{ij} - l_{uj}$.
- El algoritmo evoluciona mejorando los individuos cuánticos mediante la restricción de μ_{ij} y σ_{ij} a las áreas que retornen mejores valores de aptitud, a partir de los mejores individuos clásicos generados en cada iteración.

Problema

- Búsqueda de un mayor control de la localidad/globalidad en la actualización de individuos cuánticos.
- 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.

Problema

- El algoritmo original permite dividir el espacio de búsqueda, mas no genera una verdadera segregación pues la generación y evaluación de los individuos clásicos se sigue dando en un mismo espacio compartido.
- Los individuos cuánticos sólo son actualizados: cualquier operador (recombinación, mutación, etc.) se lleva a cabo a nivel de individuos clásicos únicamente.

Objetivos

- 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*.

Propuesta

- Segregación del espacio compartido para la evaluación y generación de los individuos clásicos en universos.
- Generar interacción entre los universos mediante un operador de recombinación que opere a nivel de individuos cuánticos.
- Se crean derivaciones de QIEA- \mathbb{R} combinando estas dos características, además del propio particionamiento del espacio de búsqueda.

Operador de recombinación

- Evaluación de los individuos cuánticos de acuerdo a la calidad de individuos clásicos generados (promedio ponderado del valor obtenido tras evaluar mediante la función *fitness* correspondiente).

Operador de recombinación

- El espacio de búsqueda del individuo cuántico de peor evaluación (el de promedio de *fitness* lejano al óptimo al que se busca converger) es incrementado al unírsele el espacio de búsqueda del individuo cuántico con la mejor evaluación.
- Operación en las k primeras iteraciones debido a la relativamente rápida degradación del espacio de búsqueda tras una rápida convergencia. El valor de k tiende a ser bajo.

Algoritmos implementados

| Algoritmo | Espacio de búsqueda particionado | Campo de acción de individuos cuánticos segregado en universos | Operador de recombinación 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 |

Escenarios de prueba

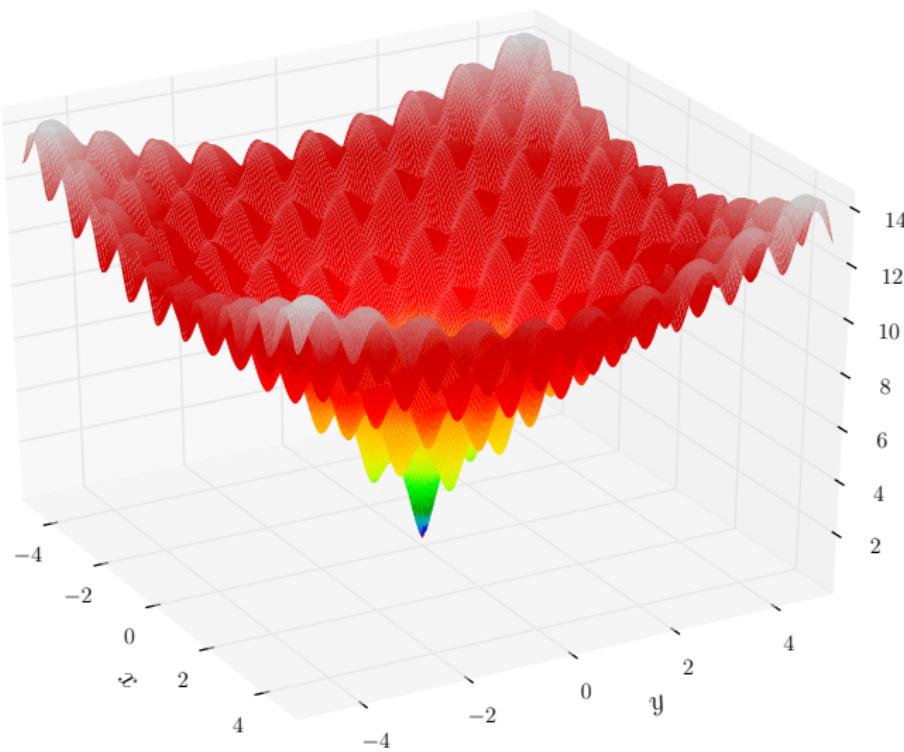
| Función | Fórmula | Valor mínimo | Dominio de búsqueda |
|---------|--|---------------|-------------------------|
| Arckley | $f(x, y) = -20e^{-0.2\sqrt{0.5(x^2+y^2)}} - e^{0.5(\cos(2\pi x)+\cos(2\pi y))} + e + 20$ | $f(0, 0) = 0$ | $-5 \leq x, y \leq 5$ |
| Booth | $f(x, y) = (x + 2y - 7)^2 + (2x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

Escenarios de prueba

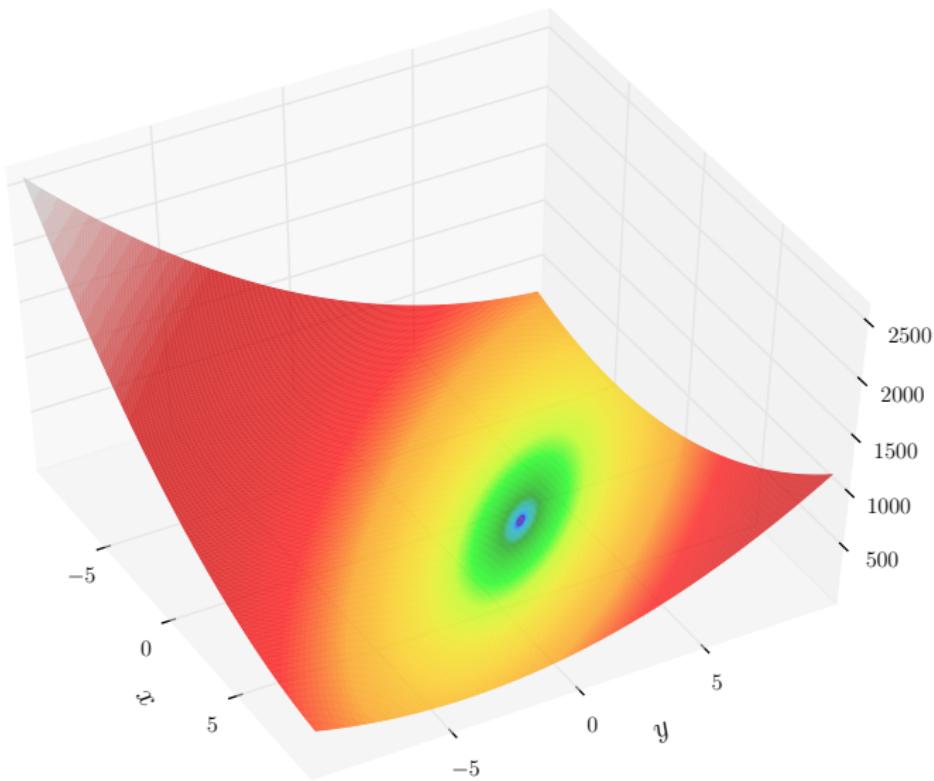
Todas las funciones poseen en común el evaluar dos variables y cumplen las siguientes condiciones:

- Espacios de búsqueda reducidos.
- 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.

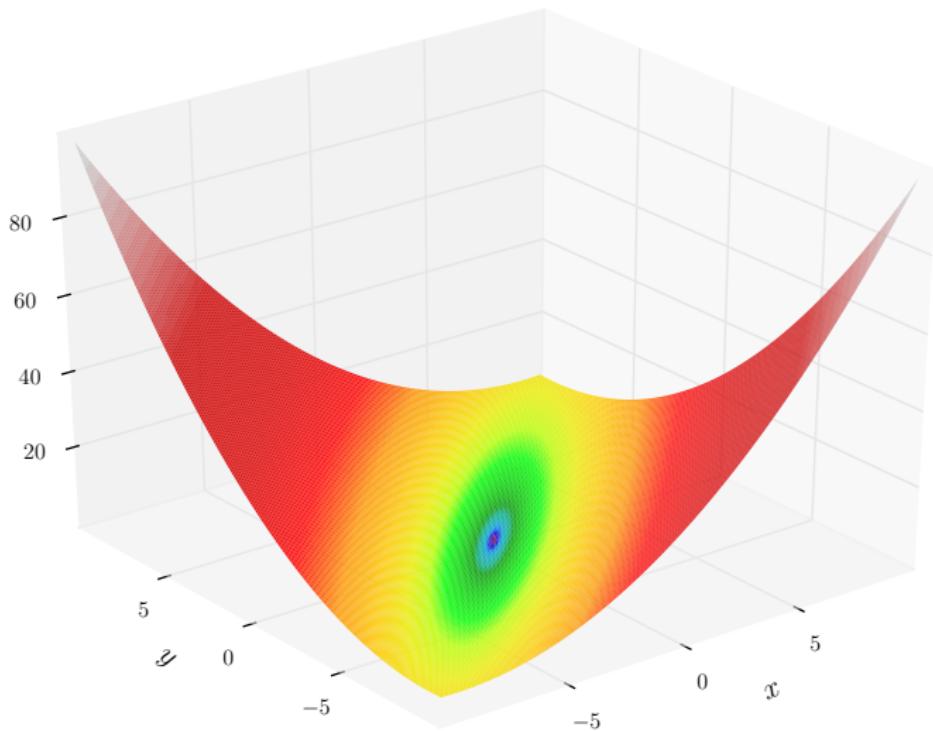
Función Arckley



Función Booth



Función Maytas



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.

- Un espacio de búsqueda p delimitado por l_p y u_p los cuales son los límites del espacio de búsqueda para cada función el cual puede ser único o presentarse particionado.
- Un número de particiones m donde $2 \leq m \leq 10$.

Condiciones para las pruebas

- 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 $1 \leq j \leq 10$

Condiciones para las pruebas

- 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 $1 \leq j \leq 10$
- $k = m - 1$ donde k es la máxima iteración donde se ejecuta el operador de recombinación.

Condiciones para las pruebas

Las características inherentes al algoritmo base seleccionadas para ejecutar en las pruebas son:

- Estrategia $(\mu + \lambda)$ -EE para la selección de la nueva población clásica: reemplazar los λ elementos de X_{t-1} por los λ mejores elementos de la unión $X_{t-1} \cup X_t$. Se toma $\lambda = 0,3$.
- Actualización del ancho de los pulsos mediante la regla de $1/5$, con $\sigma = 0,03$.
- Actualización del centro de los pulsos usando la fórmula $\mu_{ij}(t+1) = \mu_{ij}(t) + \lambda(\mu_{ij}(t) - x_{ij}(t))$ y tomando $\lambda = 0,04$

Condiciones para las pruebas

- Condición de parada: 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.

Lectura de los gráficos

- El primer tipo incluye un promedio ponderado de los $2n$ -ésimos mejores valores *fitness* para un valor n de individuos clásicos generados por individuo cuántico por iteración.
- 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.

Resultados sobre la función Arckley - Promedio $2n$ -ésimos mejores valores

Promedio de los $2n$ -ésimos mejores individuos

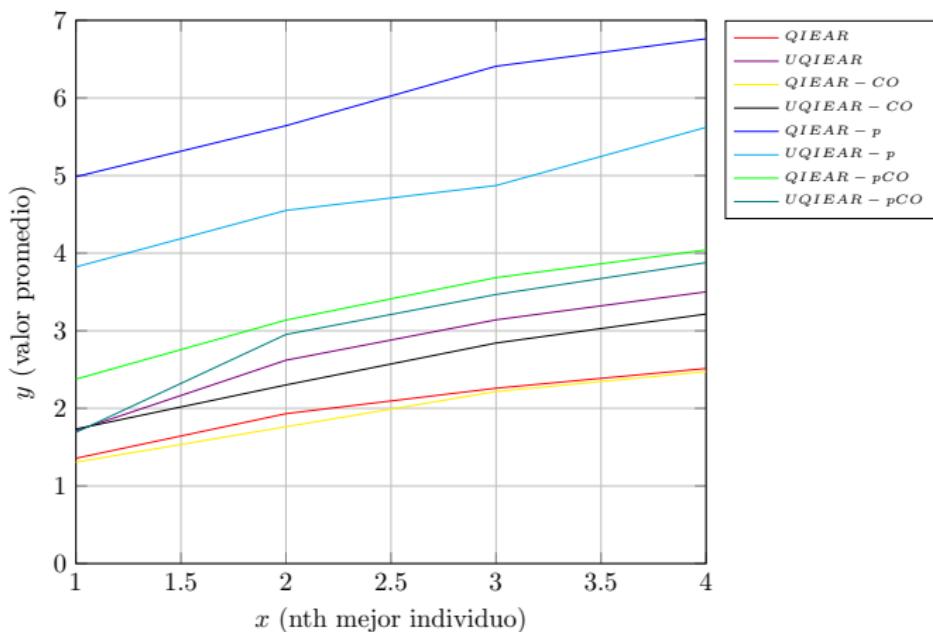


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 2$

Resultados sobre la función Arckley - Promedio $2n$ -ésimos mejores valores

Promedio de los $2n$ -ésimos mejores individuos

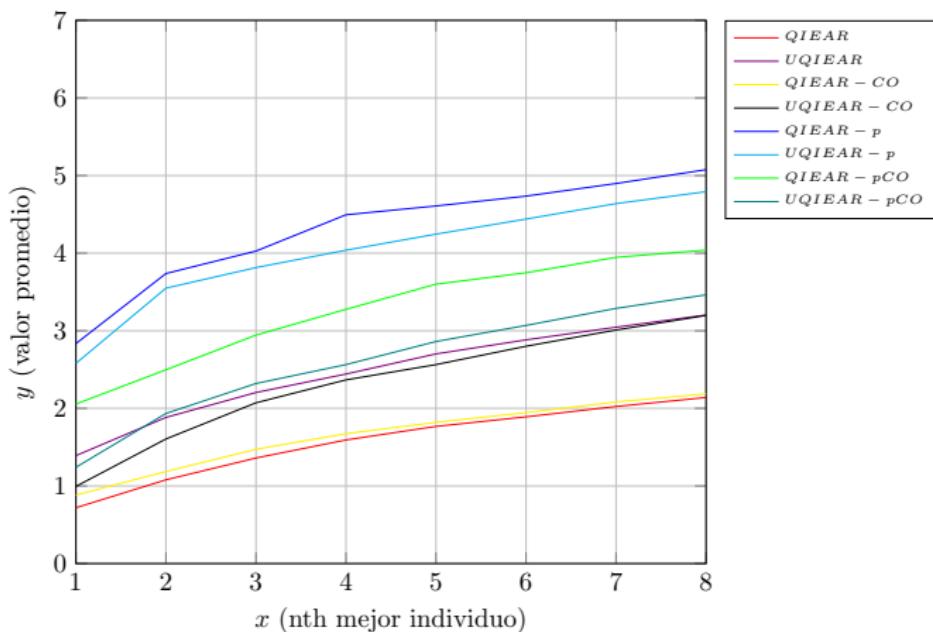


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 4$

Resultados sobre la función Arckley - Promedio $2n$ -ésimos mejores valores

Promedio de los $2n$ -ésimos mejores individuos

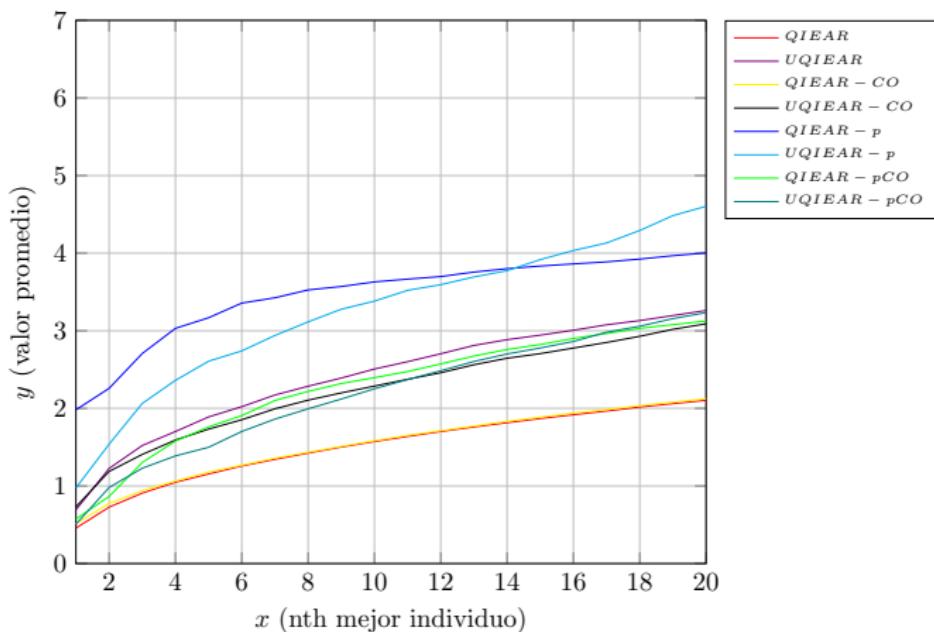


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 10$

Resultados sobre la función Arckley

- Los mejores resultados fueron producidos por el algoritmo original QIEAR®, así como por QIEAR®-CO, ambos algoritmos sin particionar.
- 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.

Resultados sobre la función Arckley - Evolución del mejor valor obtenido por iteración

Evolución por iteración de los mejores individuos

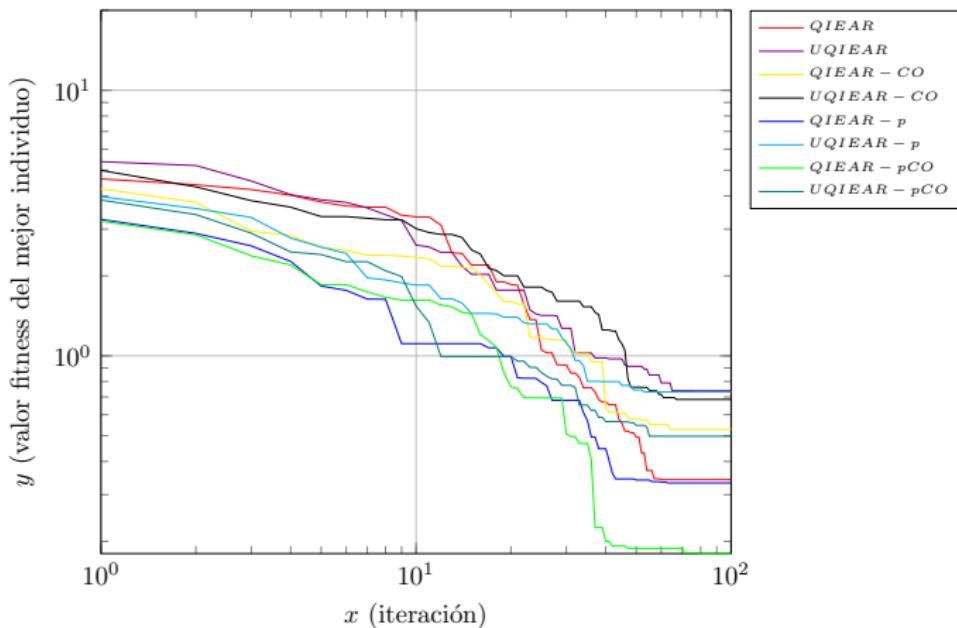


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 3$

Resultados sobre la función Arckley - Evolución del mejor valor obtenido por iteración

Evolución por iteración de los mejores individuos

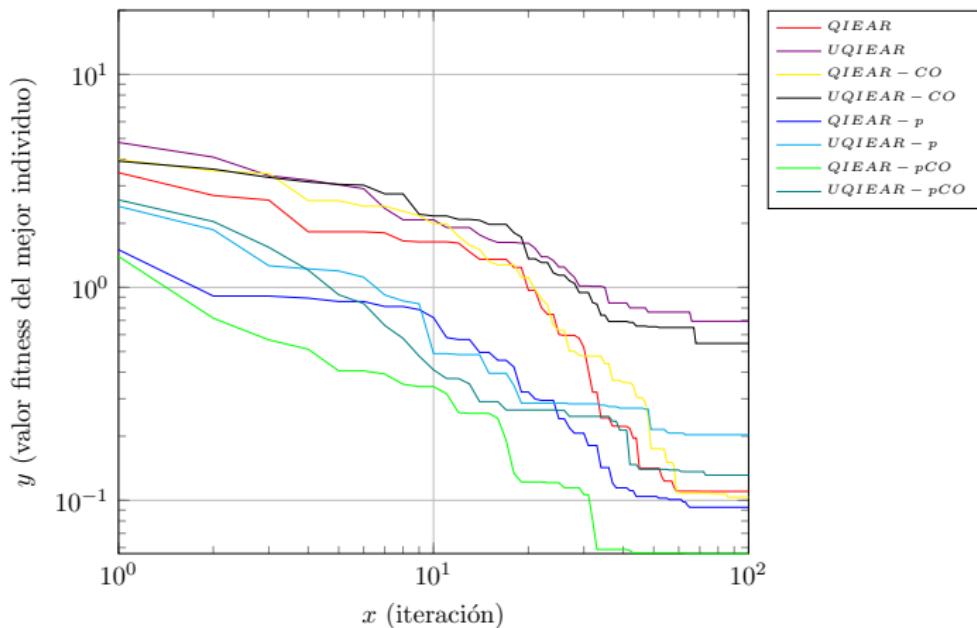


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 7$

Resultados sobre la función Arckley - Evolución del mejor valor obtenido por iteración

Evolución por iteración de los mejores individuos

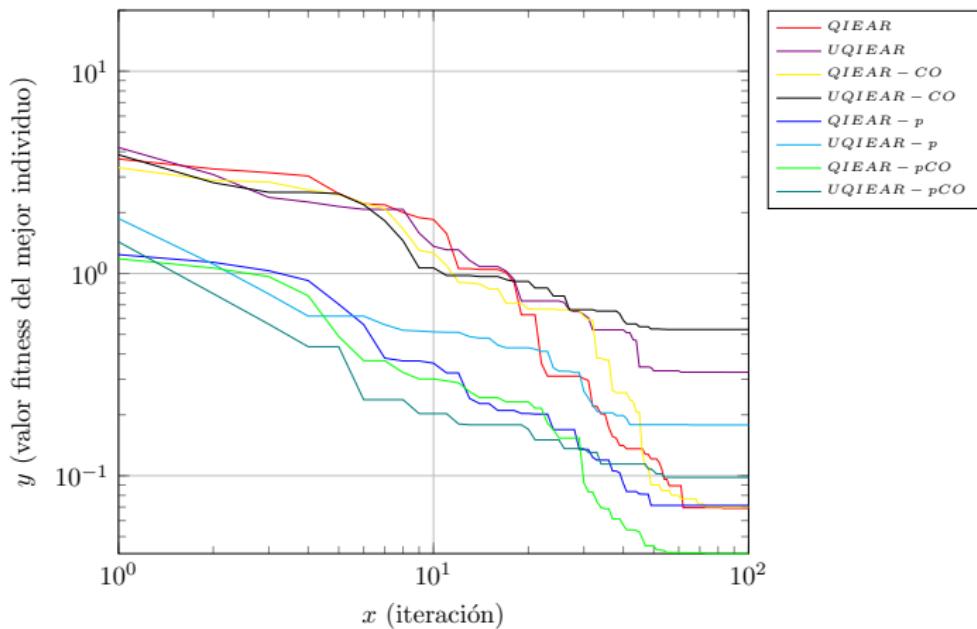


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 9$

Resultados sobre la función Arckley

- El algoritmo QIEA®-pCO supera a los demás tanto en velocidad de convergencia como calidad final de resultados, siendo seguido a corta distancia por QIEA®-p, UQIEA®-pCO y UQIEA®-p. Todos estos algoritmos son particionados.
- Los algoritmos con peores resultados en convergencia son las modificaciones UQIEA® y UQIEA®-CO.

Resultados sobre la función Arckley

- El algoritmo original QIEA® converge mediocremente 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.

Resultados sobre la función Booth - Promedio $2n$ -ésimos mejores valores

Promedio de los $2n$ -ésimos mejores individuos

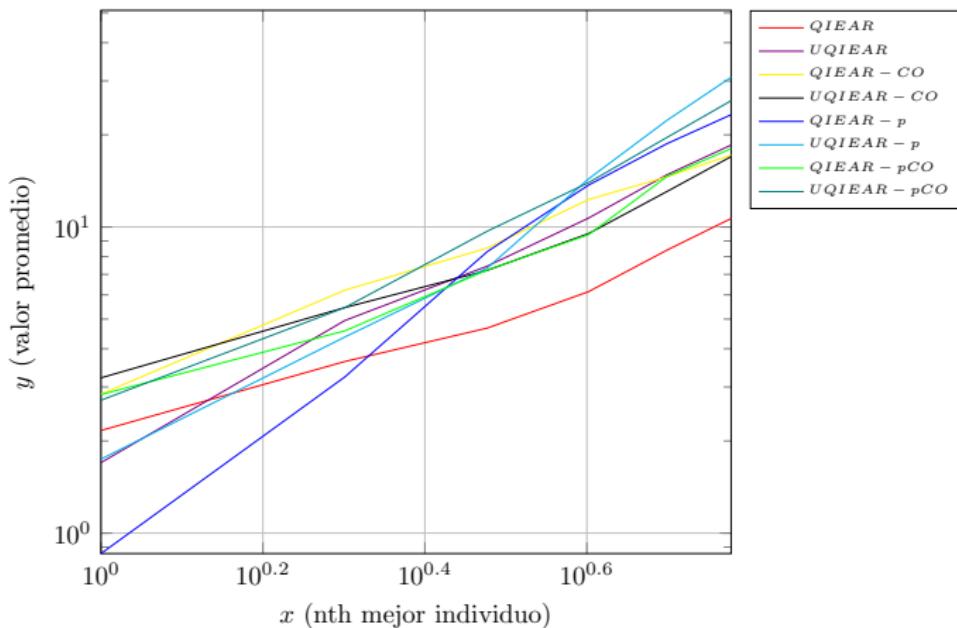


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 3$

Resultados sobre la función Booth - Promedio $2n$ -ésimos mejores valores

Promedio de los $2n$ -ésimos mejores individuos

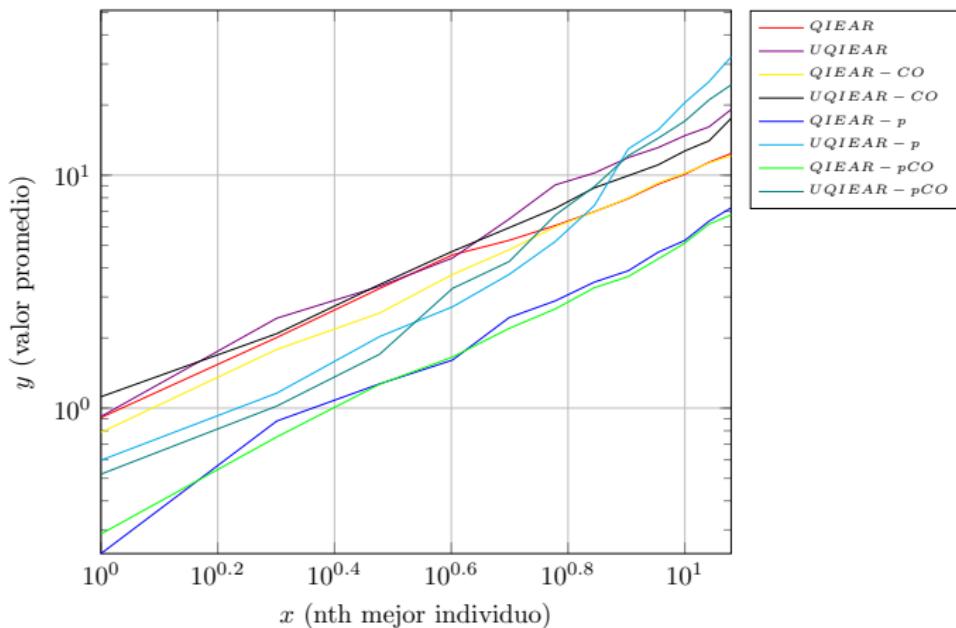


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 6$

Resultados sobre la función Booth - Promedio $2n$ -ésimos mejores valores

Promedio de los $2n$ -ésimos mejores individuos

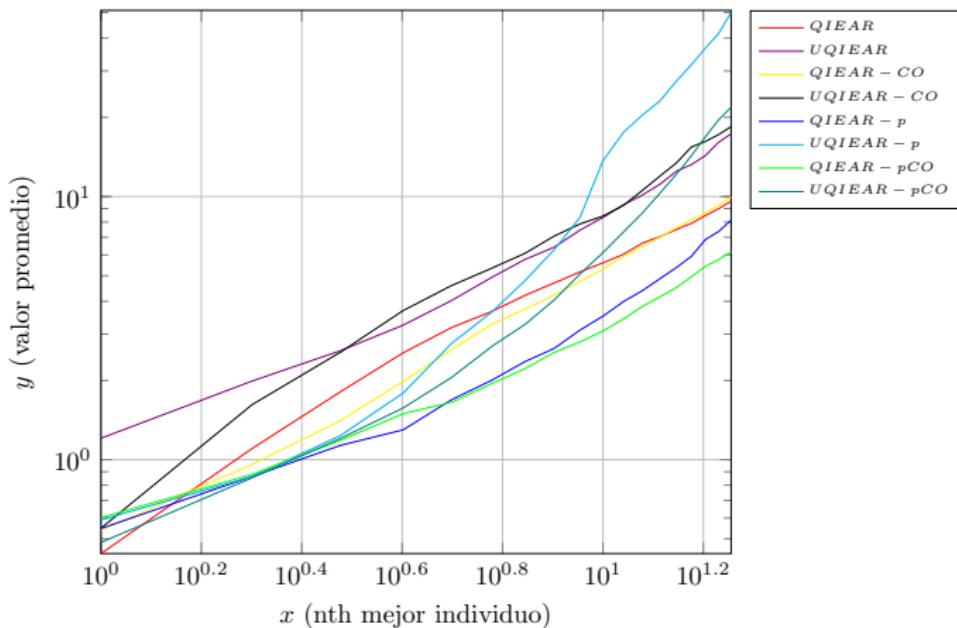


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 9$

Resultados sobre la función Booth

- Los algoritmos segregados en promedio se comportan de forma más mediocre que los algoritmos sin separar. 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 recombinación mejoran en la calidad de datos (QIEAR®-pCO y QIEAR®-CO, en ese orden).

Resultados sobre la función Booth - Evolución del mejor valor obtenido por iteración

Evolución por iteración de los mejores individuos

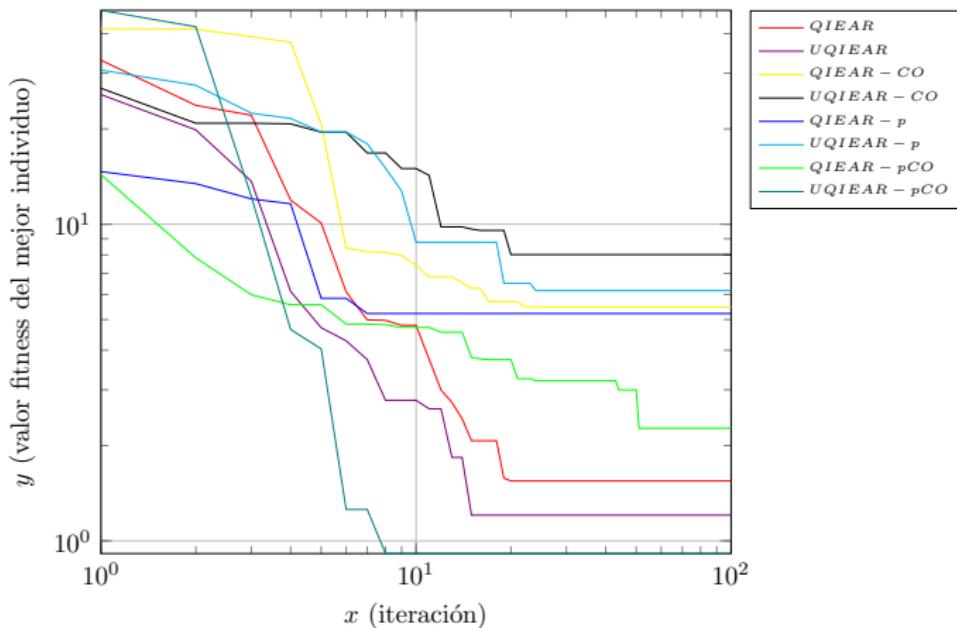


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 2$

Resultados sobre la función Booth - Evolución del mejor valor obtenido por iteración

Evolución por iteración de los mejores individuos

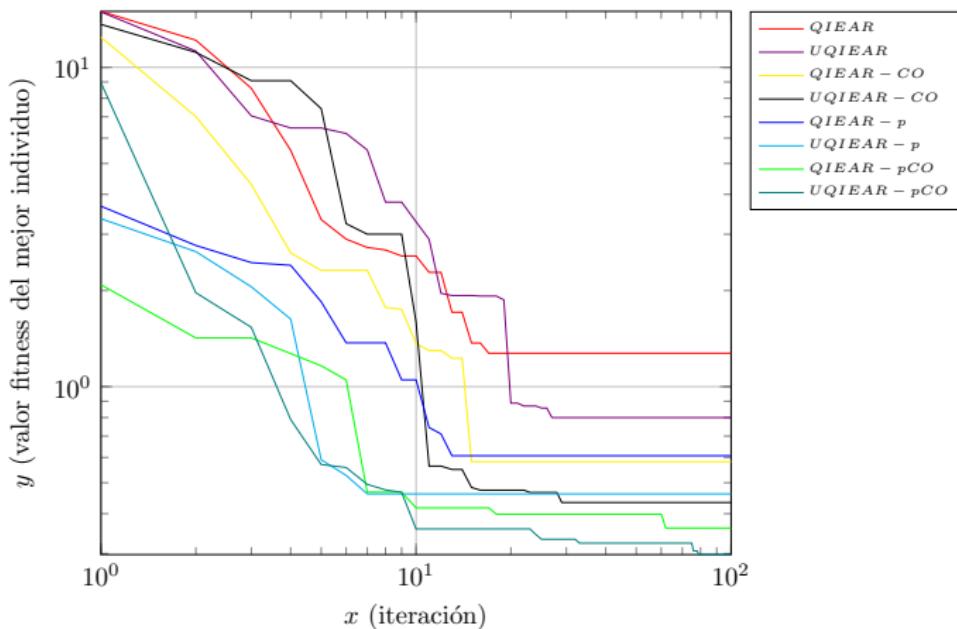


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 4$

Resultados sobre la función Booth - Evolución del mejor valor obtenido por iteración

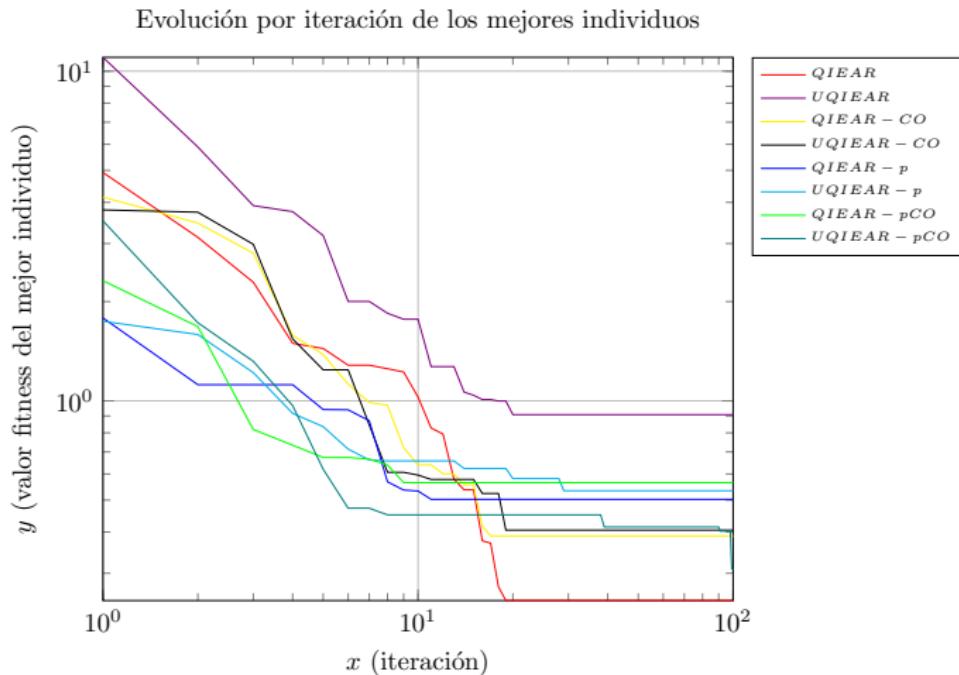


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 9$

Resultados sobre la función Booth

- 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 recombinación es competitivo en comparación con los demás.

Resultados sobre la función Maytas - Promedio $2n$ -ésimos mejores valores

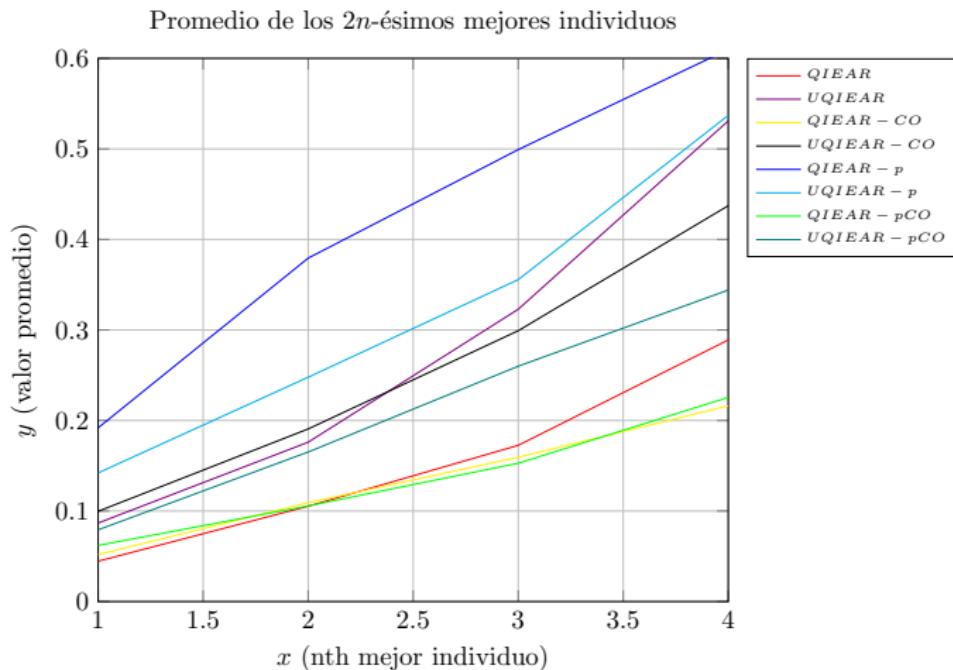


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 2$

Resultados sobre la función Maytas - Promedio $2n$ -ésimos mejores valores

Promedio de los $2n$ -ésimos mejores individuos

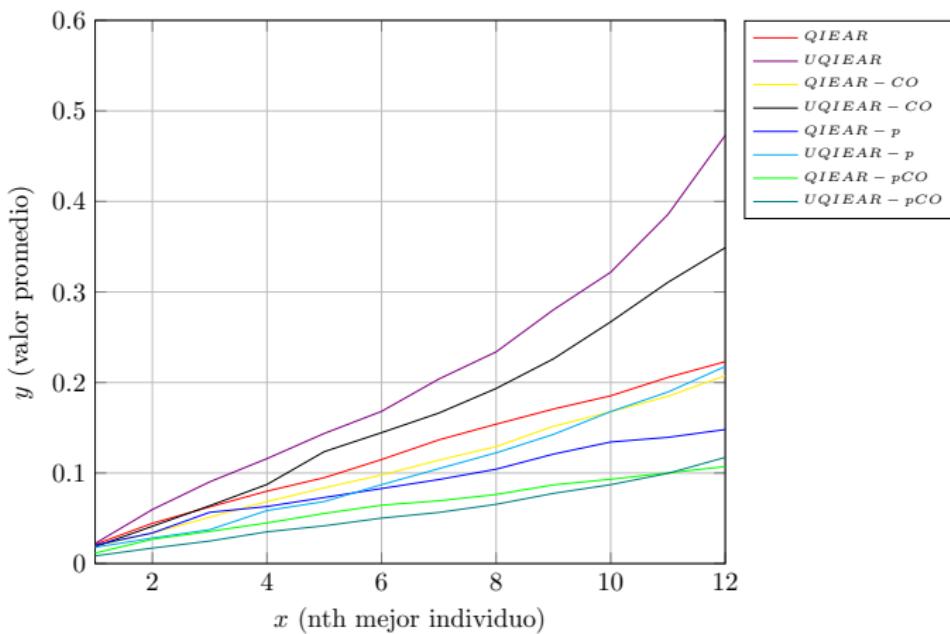


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 6$

Resultados sobre la función Maytas - Promedio $2n$ -ésimos mejores valores

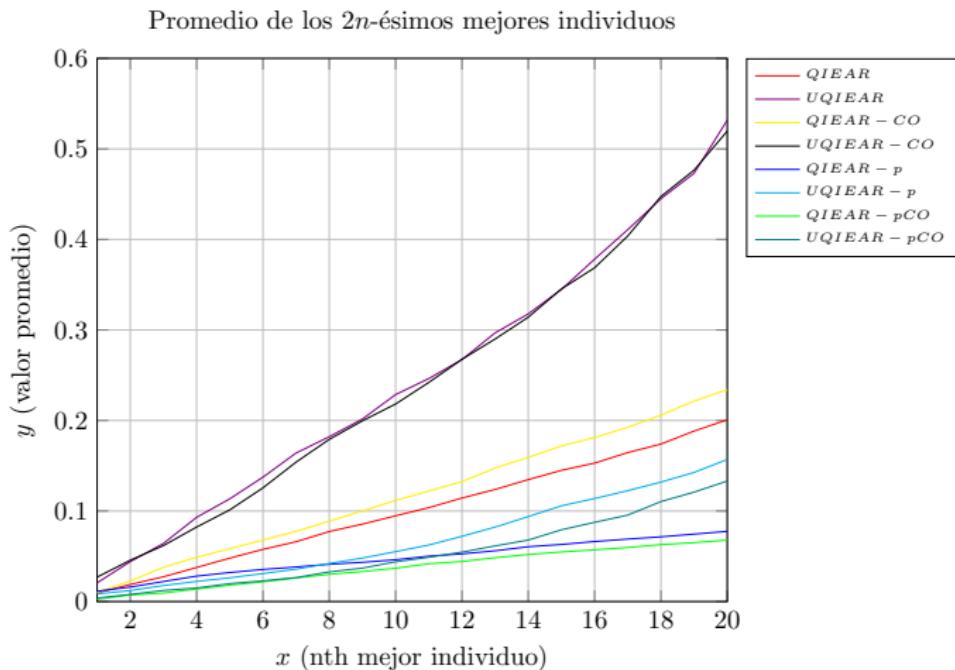


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 10$

Resultados sobre la función Maytas

- 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.

Resultados sobre la función Maytas - Evolución del mejor valor obtenido por iteración

Evolución por iteración de los mejores individuos

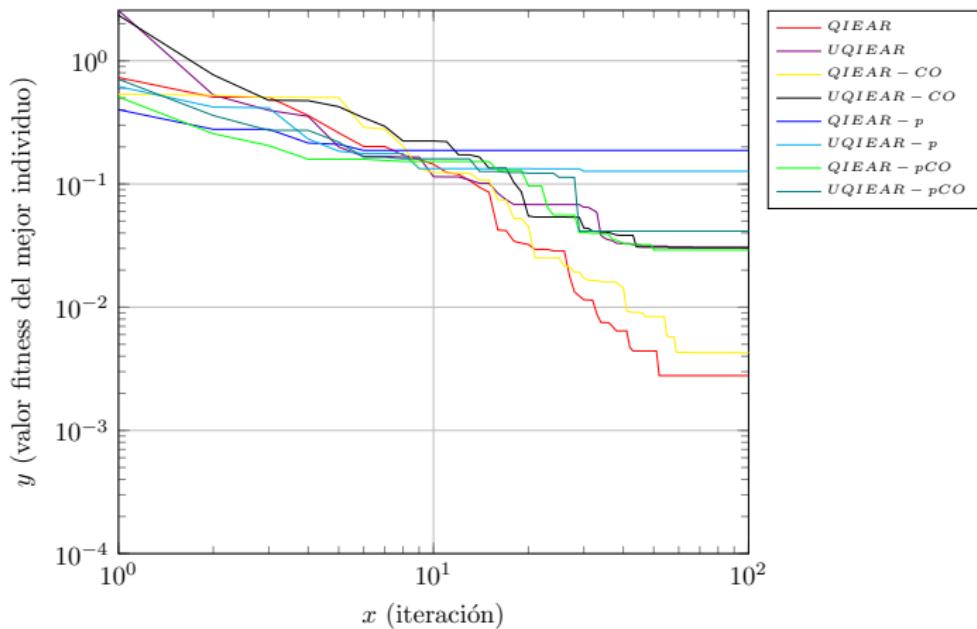


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 2$

Resultados sobre la función Maytas - Evolución del mejor valor obtenido por iteración

Evolución por iteración de los mejores individuos

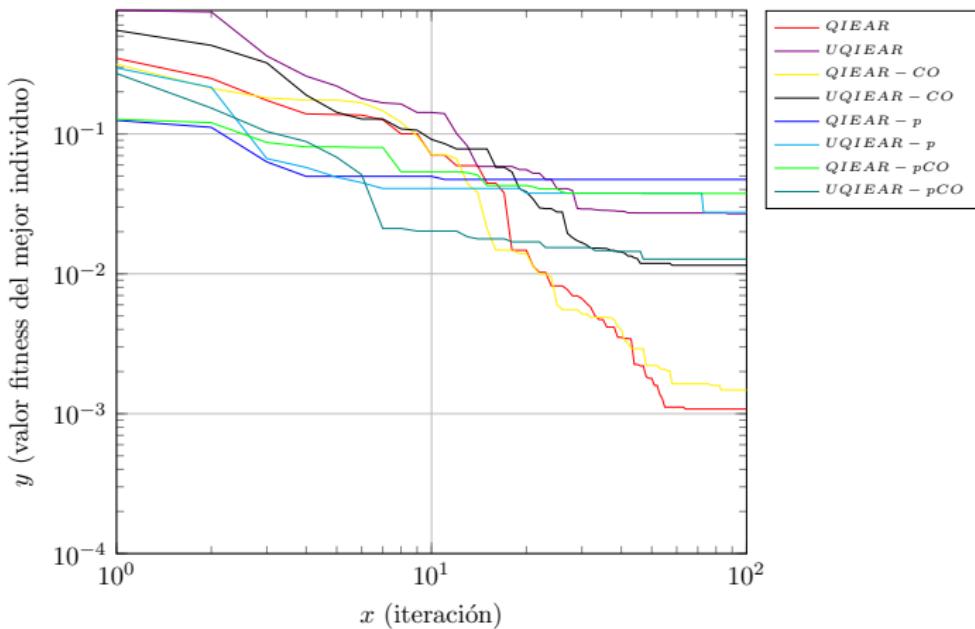


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 4$

Resultados sobre la función Maytas - Evolución del mejor valor obtenido por iteración

Evolución por iteración de los mejores individuos

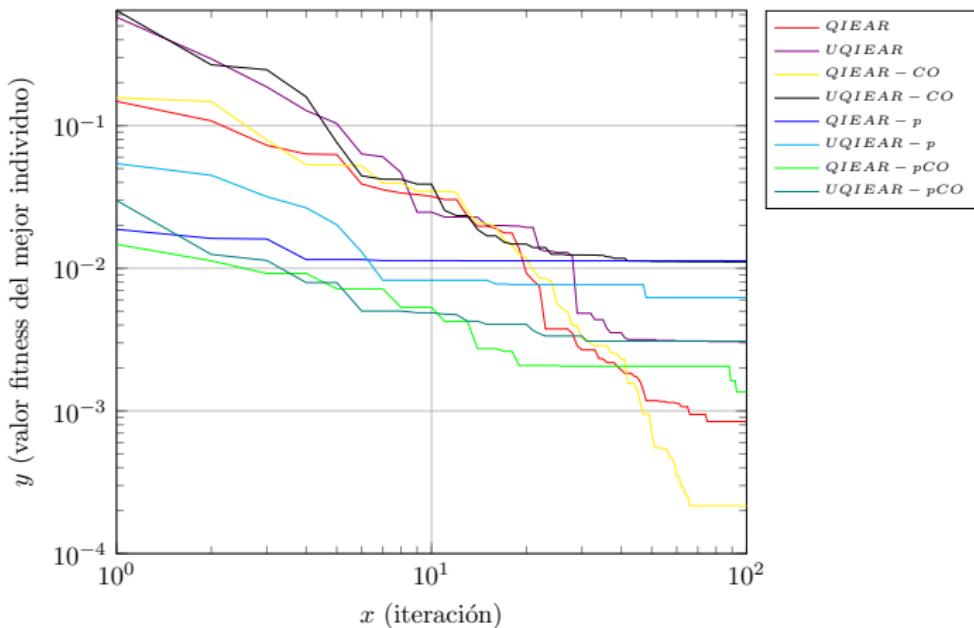


Figura : Valor promedio de los n -ésimos mejores individuos para $n = 10$

Resultados sobre la función Maytas

- A pesar que los algoritmos que implementan espacios particionados de búsqueda obtienen una convergencia más rápida, al final convergen menos hacia el mínimo.
- Mejor desempeño promedio: QIEAR y UQIEAR-pCO debido a figurar por encima del promedio tanto en rapidez de convergencia como en producción de datos cercanos al mínimo.

Conclusiones

- Resultados medianamente consistentes y competitivos: únicamente las variantes que implementan el operador de recombinación en espacios particionados (QIEAR®-pCO y UQIEAR®-pCO), logrando obtener resultados mejores que los algoritmos originales cuando el número de individuos clásicos generados por individuo cuántico por iteración es más bajo.
- Por lo tanto, se puede concluir que el operador de recombinación propuesto es una alternativa válida que reduce la necesidad de generar una cantidad alta de individuos clásicos.

Conclusiones

- Por otro lado, los algoritmos que implementan la segregación en los campos de operación de cada individuo clásico no manifiestan una diferencia sustancial que permita distinguirlos de sus contrapartes no segregadas.
- En los casos en los que un algoritmo con espacios de búsqueda segregados por individuo cuántico destaca -UQIEA®-pCO es un buen ejemplo de este caso-, parece ser más influencia del propio operador de recombinación que de la segregación propiamente dicha.

Conclusiones

- 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 recombinación.

Problemas

- La propia tendencia del algoritmo a sobreincrementar rápidamente sus espacios de búsqueda tras un espacio de tiempo en el que no se hallan mejores soluciones.
- Las propuestas no se han podido evaluar para problemas con espacios de búsqueda mayores a los escogidos en el presente trabajo.

Problemas

- Las pruebas en funciones con mínimos ubicados cercanos en los extremos de los espacios de búsqueda también han representado un reto que no se pudo asumir, debido a la inexistencia de una metodología apropiada para la delimitación de la expansión en el espacio de búsqueda del algoritmo original.
- El comportamiento del algoritmo lo impulsa a buscar más mínimos fuera de dichos límites, por lo que terminarían tendiendo hacia un mínimo no contemplado al interior del espacio de búsqueda.