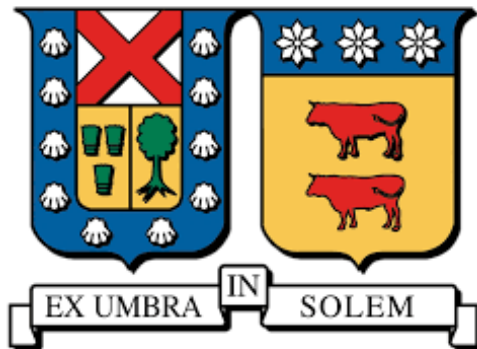


Estado del Arte del Algoritmo Cuántico de Optimización Aproximada (QAOA) con Enfoque en Max-Cut

Mauro Castillo
Javiera Osorio
Mariano Varas Ramos

Segundo semestre de 2025
(Versión Mejorada: 21 de octubre de 2025)



Curso: Introducción a la Computación Cuántica

Profesor: Mauricio Solar

Institución: Universidad Técnica Federico Santa María, Departamento de Informática

Resumen

El Algoritmo Cuántico de Optimización Aproximada (QAOA) representa una de las vías más estudiadas para alcanzar la ventaja cuántica en problemas de optimización combinatoria utilizando hardware de la era NISQ (Noisy Intermediate-Scale Quantum). Sin embargo, las limitaciones inherentes a estos dispositivos, como el ruido, la decoherencia y el número reducido de cúbits, plantean desafíos significativos para su implementación práctica y escalabilidad. Este informe presenta un estado del arte actualizado sobre QAOA, centrándose específicamente en el problema de Max-Cut como el banco de pruebas canónico para evaluar su progreso, variantes metodológicas y rendimiento. Se analizan las principales corrientes de investigación destinadas a superar las barreras del hardware actual, incluyendo: (1) estrategias avanzadas para la optimización de los parámetros variacionales clásicos, (2) mejoras algorítmicas fundamentales como la inicialización caliente (warm-starting) con mezcladores estándar y personalizados, el QAOA Multi-Ángulo (MA-QAOA) y el QAOA Recursivo (RQAOA), potenciado recientemente con aprendizaje por refuerzo, y (3) técnicas de co-diseño hardware-software como el corte de circuitos (circuit cutting) para mitigar errores y abordar problemas de mayor escala. La tesis central es que la viabilidad de QAOA reside en la sinergia de estas estrategias, y Max-Cut ha sido fundamental para validar y comparar su eficacia.

Índice

1. Introducción	4
2. El Algoritmo QAOA: Fundamentos	4
3. El Problema Max-Cut como Banco de Pruebas Canónico	5
4. Desafíos de QAOA en la Era NISQ	5
5. Estado del Arte: Estrategias para Mejorar QAOA	7
5.1. Corriente 1: Optimización de Parámetros	7
5.2. Corriente 2: Mejoras Algorítmicas (Software)	8
5.2.1. Inicialización Caliente (Warm-Starting)	8
5.2.2. QAOA Multi-Ángulo (MA-QAOA)	9
5.2.3. QAOA Recursivo (RQAOA) y Aprendizaje por Refuerzo	10
5.3. Corriente 3: Co-diseño Hardware-Software y Mitigación de Errores	11
5.3.1. Corte de Circuitos (Circuit Cutting)	11
6. Análisis Comparativo y Síntesis	11
7. Conclusiones	11

1. Introducción

La optimización combinatoria aborda problemas cruciales en diversas industrias, desde la logística y finanzas hasta el diseño de fármacos y la ciencia de materiales. Muchos de estos problemas son NP-difíciles, lo que implica que encontrar soluciones óptimas exactas se vuelve computacionalmente intratable para instancias de gran tamaño utilizando algoritmos clásicos [1, 2]. El Algoritmo Cuántico de Optimización Aproximada (QAOA), propuesto por Farhi, Goldstone y Gutmann [3], emerge como un candidato prominente para ofrecer una ventaja cuántica, especialmente en la era de los dispositivos cuánticos ruidosos de escala intermedia (NISQ) [4, 5].

Para evaluar de forma sistemática el potencial y las limitaciones de QAOA, la comunidad científica ha adoptado ampliamente el problema de **Max-Cut** como el **banco de pruebas (benchmark) estándar** [1, 6]. Max-Cut consiste en encontrar una partición de los vértices de un grafo en dos conjuntos disjuntos tal que se maximice el peso total de las aristas que conectan vértices en conjuntos diferentes. Su estructura relativamente simple, combinada con su complejidad NP-difícil, lo convierte en un caso de estudio ideal para comparar el rendimiento de QAOA con algoritmos clásicos de vanguardia, como el algoritmo de Goemans-Williamson (GW) basado en relajaciones semidefinidas (SDP) [2, 6, 7].

Sin embargo, la implementación directa de QAOA en hardware cuántico real enfrenta obstáculos significativos: el ruido inherente a los cúbits, la conectividad limitada entre ellos, los tiempos de coherencia finitos y la dificultad de optimizar los parámetros clásicos del algoritmo [1, 4]. Esta brecha entre la promesa teórica de QAOA y su viabilidad práctica ha catalizado una intensa investigación en múltiples frentes.

Este informe sintetiza el estado del arte de las estrategias desarrolladas para mejorar la aplicabilidad y el rendimiento de QAOA, utilizando Max-Cut como caso de estudio central. Se analizan tres corrientes principales de innovación:

1. **Optimización de Parámetros:** Desarrollo de heurísticas clásicas eficientes para encontrar los ángulos variacionales (γ, β) del circuito QAOA [1, 5].
2. **Mejoras Algorítmicas (Software):** Modificaciones fundamentales del algoritmo, como la inicialización caliente (warm-starting) [2, 6], QAOA Multi-Ángulo (MA-QAOA) [5], y QAOA Recursivo (RQAOA), incluyendo su integración con aprendizaje por refuerzo (RL) [8].
3. **Co-diseño Hardware-Software y Mitigación de Errores:** Técnicas que adaptan el algoritmo a las restricciones del hardware y reducen el impacto del ruido, como el corte de circuitos (circuit cutting) [9, 10].

A través del análisis de estos avances, validados consistentemente en el contexto de Max-Cut, este documento ofrece una perspectiva integral sobre la evolución de QAOA hacia la computación cuántica de utilidad práctica.

2. El Algoritmo QAOA: Fundamentos

El QAOA es un algoritmo híbrido cuántico-clásico diseñado para encontrar soluciones aproximadas a problemas de optimización combinatoria. El problema se codifica en un Hamiltoniano de costo H_C , cuyas autoestados corresponden a las posibles soluciones y cuyos autovalores corresponden a los costos asociados [3]. Para Max-Cut en un grafo $G = (V, E)$ con pesos w_{ij} en las aristas, el Hamiltoniano de costo (maximizar $\langle H_C \rangle$) se define comúnmente como:

$$H_C = \sum_{\langle i,j \rangle \in E} \frac{w_{ij}}{2} (I - \sigma_i^z \sigma_j^z)$$

donde σ_i^z es el operador de Pauli Z actuando sobre el cúbit i (representando el vértice i), e I es la identidad.

El algoritmo prepara un estado variacional $|\psi_p(\vec{\gamma}, \vec{\beta})\rangle$ aplicando alternativamente el operador de evolución del Hamiltoniano de costo $U_C(\gamma_k) = e^{-i\gamma_k H_C}$ y el operador de evolución de un Hamiltoniano mezclador $U_B(\beta_k) = e^{-i\beta_k H_B}$, p veces, sobre un estado inicial, típicamente la superposición uniforme $|+\rangle^{\otimes N}$:

$$|\psi_p(\vec{\gamma}, \vec{\beta})\rangle = U_B(\beta_p)U_C(\gamma_p) \cdots U_B(\beta_1)U_C(\gamma_1)|+\rangle^{\otimes N}$$

El mezclador estándar es $H_B = \sum_{i=1}^N \sigma_i^x$, donde σ_i^x es el operador de Pauli X sobre el cúbit i [3]. El entero p se denomina la ****profundidad**** del QAOA. Los $2p$ parámetros $(\vec{\gamma}, \vec{\beta}) = (\gamma_1, \dots, \gamma_p, \beta_1, \dots, \beta_p)$ son ángulos variacionales que se optimizan clásicamente para maximizar el valor esperado del Hamiltoniano de costo:

$$F_p(\vec{\gamma}, \vec{\beta}) = \langle \psi_p(\vec{\gamma}, \vec{\beta}) | H_C | \psi_p(\vec{\gamma}, \vec{\beta}) \rangle$$

Finalmente, se mide el estado $|\psi_p(\vec{\gamma}^*, \vec{\beta}^*)\rangle$ (con los parámetros óptimos) en la base computacional para obtener una cadena de bits que representa una solución aproximada al problema de optimización [1]. La calidad de la solución se mide a menudo por la ****ratio de aproximación**** $r = F_p(\vec{\gamma}^*, \vec{\beta}^*)/C_{max}$, donde C_{max} es el valor óptimo conocido (o el mejor encontrado).

3. El Problema Max-Cut como Banco de Pruebas Canónico

Max-Cut ha sido adoptado como el benchmark principal para QAOA por varias razones clave:

- **Complejidad NP-difícil:** Aunque su formulación es simple, encontrar la solución óptima es NP-difícil en grafos generales, representando una clase de problemas computacionalmente desafiantes para algoritmos clásicos [4]. Se sabe que es NP-difícil aproximar Max-Cut más allá de un factor de $16/17 \approx 0,941$ [5, 11].
- **Estructura Simple:** El Hamiltoniano de costo H_C para Max-Cut solo involucra interacciones de dos cúbits ($\sigma_i^z \sigma_j^z$) correspondientes a las aristas del grafo, lo que simplifica su implementación y análisis teórico [1].
- **Bien Estudiado Clásicamente:** Existen algoritmos clásicos de aproximación bien conocidos y potentes para Max-Cut, como el algoritmo de Goemans-Williamson (GW) [7], que proporciona una ratio de aproximación garantizada de $\approx 0,878$ para grafos con pesos no negativos. Esto ofrece una línea base sólida contra la cual comparar el rendimiento de QAOA [2, 6].
- **Relevancia Práctica:** Problemas relacionados con Max-Cut aparecen en diversas aplicaciones, como diseño de circuitos, física estadística y segmentación de imágenes [4].

Estas características hacen de Max-Cut un "laboratorio ideal para probar nuevas ideas, variantes y optimizaciones de QAOA antes de aplicarlas a problemas más complejos o específicos de la industria.

4. Desafíos de QAOA en la Era NISQ

La promesa teórica de QAOA, incluyendo su convergencia al óptimo para $p \rightarrow \infty$ [3], se enfrenta a la dura realidad del hardware NISQ. El trabajo seminal de Guerreschi y Matsuura

(2019) [4] utilizó simulaciones realistas, incluyendo ruido de decoherencia y disipación, compilación para conectividad limitada (cuadrícula 2D) y overhead de enrutamiento (SWAPs), para estimar el costo computacional de QAOA para Max-Cut en grafos 3-regulares.

Metodología y Resultados Clave (Guerreschi & Matsuura [4]):

- **Simulación Realista:** Modelaron el ruido (decoherencia T_2 , disipación T_1) usando la aproximación de "Noise Gates" [12] y compilaron los circuitos QAOA para una arquitectura de cuadrícula 2D, añadiendo SWAPs necesarios.
- **Estimación del Costo Temporal:** Calcularon el tiempo absoluto de ejecución experimental proyectado, considerando tiempos de compuerta (T_G), preparación y medida ($T_P + T_M$), y el número total de ejecuciones requeridas (optimizaciones \times iteraciones \times muestras por estimación). Utilizaron el optimizador Nelder-Mead y 10000 muestras por evaluación de F_p .
- **Comparación con Solver Clásico:** Compararon el tiempo proyectado de QAOA ($p = 4, p = 8$) con el tiempo medido del solver clásico AKMAXSAT [13] en un procesador Intel Xeon Phi.
- **Estimación del Crossover:** Concluyeron que, para superar a AKMAXSAT, QAOA necesitaría "varios cientos de cúbits", estimando el punto de cruce del rendimiento entre cientos y unos pocos miles de cúbits, aunque con incertidumbre debido a la extrapolación desde tamaños pequeños ($N \leq 26$) (ver Fig. 2 en [4]).
- **Cuello de Botella:** Identificaron que la dificultad de optimizar los parámetros $(\vec{\gamma}, \vec{\beta})$, que podría requerir un número exponencial de intentos si se parte de puntos aleatorios, contribuye significativamente al costo total, más allá de la profundidad del circuito cuántico en sí.

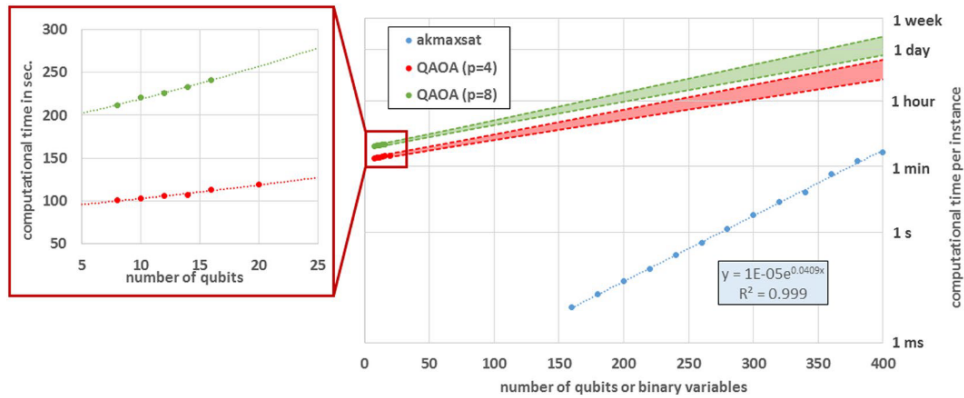


Figura 1: Costo computacional (tiempo por instancia) vs. número de cúbits/variables binarias para QAOA ($p = 4, p = 8$) proyectado experimentalmente y para el solver clásico AKMAXSAT. Se muestra la extrapolación exponencial para QAOA y la exponencial ajustada para AKMAXSAT. El recuadro muestra un zoom en los datos de QAOA. Adaptado de Guerreschi & Matsuura (2019) [4].

Este estudio subrayó los principales desafíos que motivaron las innovaciones posteriores:

- **Ruido y Profundidad Limitada:** Los errores limitan severamente la profundidad p alcanzable [1, 4].
- **Overhead de Compilación:** La conectividad limitada requiere SWAPs que aumentan la profundidad y la exposición al ruido [4].

- **Optimización Clásica Costosa:** Encontrar $(\vec{\gamma}^*, \vec{\beta}^*)$ es computacionalmente intensivo [1, 6, 8].
- **Costo de Muestreo:** Estimar F_p requiere muchas ejecuciones del circuito [1, 4].

5. Estado del Arte: Estrategias para Mejorar QAOA

Frente a los desafíos NISQ, la investigación ha explorado múltiples vías para hacer QAOA más práctico y potente.

5.1. Corriente 1: Optimización de Parámetros

El alto costo de encontrar los parámetros óptimos $(\vec{\gamma}^*, \vec{\beta}^*)$ fue abordado por Zhou et al. (2020) [1].

Metodología y Resultados Clave (Zhou et al. [1]):

- **Patrones de Parámetros:** Mediante búsquedas exhaustivas para Max-Cut en grafos regulares ($N \leq 22, p \leq 50$), identificaron patrones persistentes: γ_i^* tiende a aumentar con i , mientras que β_i^* tiende a disminuir, siguiendo curvas suaves que varían poco entre instancias similares y al pasar de p a $p + 1$ (ver Fig. 2 en [1]).
- **Heurísticas de Inicialización:** Basándose en estos patrones, propusieron estrategias para generar puntos iniciales para la optimización en nivel $p + 1$ a partir de los óptimos del nivel p :
 - **INTERP:** Interpolación lineal de los parámetros de nivel p .
 - **FOURIER:** Reutilizar los coeficientes de una parametrización basada en transformadas de seno/coseno, asumiendo que las componentes de baja frecuencia dominan.
- **Reducción de Costo:** Demostraron numéricamente que estas heurísticas permiten encontrar parámetros cuasi-óptimos con un costo de optimización que escala polinomialmente en p ($O(\text{poly}(p))$), en comparación con el costo $2^{O(p)}$ de múltiples inicializaciones aleatorias (RI) (ver Fig. 3b en [1]).
- **Comparación con Annealing y Mecanismo:** Observaron que el patrón de parámetros recuerda a un protocolo de annealing adiabático, pero demostraron que QAOA puede superar las limitaciones adiabáticas (pequeños gaps espectrales) utilizando mecanismos no adiabáticos ("diabatic bump") aprendidos durante la optimización (ver Fig. 5 y 6 en [1]).

El trabajo de Gaidai & Herrman (2024) [5] también realizó una comparativa de heurísticas para QAOA estándar (Constant, TQA, Interp, Fourier, Greedy, Random), destacando la sorprendente efectividad de la inicialización 'Constant' para sus datos (ver Fig. 1 en [5]).

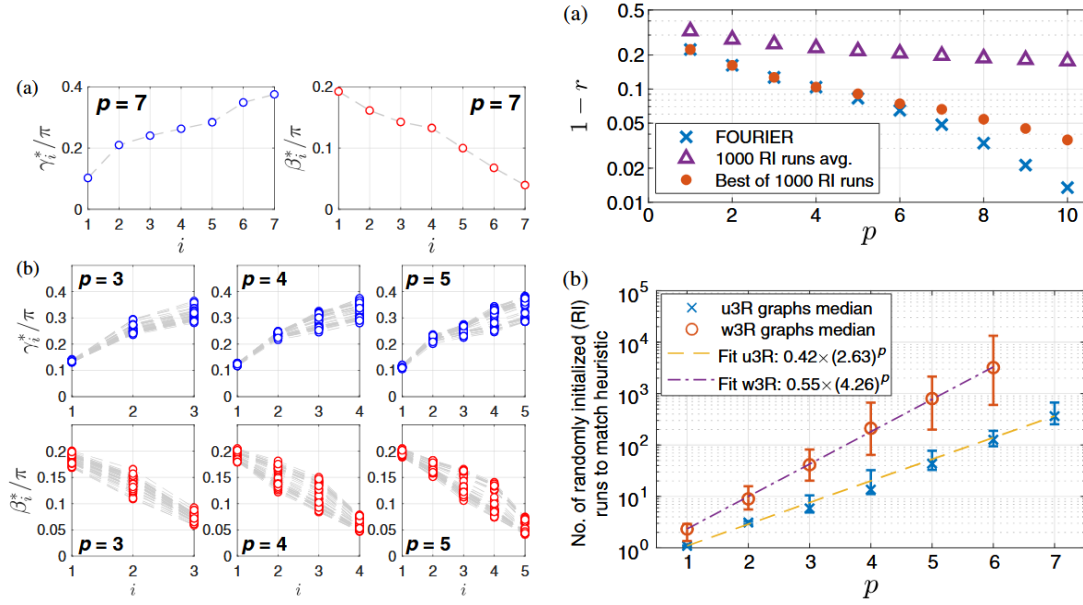


Figura 2: Izquierda: Patrones en parámetros óptimos ($\vec{\gamma}^*$, $\vec{\beta}^*$) para QAOA en grafos u3R ($N=16$, $p=3, 4, 5$). Derecha: Número mediano de ejecuciones con inicialización aleatoria (RI) necesarias para igualar el rendimiento de la heurística FOURIER, mostrando un escalado exponencial. Adaptado de Zhou et al. (2020) [1].

5.2. Corriente 2: Mejoras Algorítmicas (Software)

5.2.1. Inicialización Caliente (Warm-Starting)

Esta técnica busca mejorar la convergencia inicializando el estado cuántico cerca de una buena solución clásica.

Metodología y Resultados Clave (Egger et al. [2]):

- **Concepto:** Reemplazar el estado inicial $|+\rangle^{\otimes N}$ por un estado $|\phi^*\rangle$ derivado de la solución c^* de una relajación clásica (e.g., QP convexa o redondeo de SDP).
- **WS-QAOA Continuo:** Para relajaciones convexas (como en optimización de portafolios), mapearon $c^* \in [0, 1]^N$ al estado producto $|\phi^*\rangle = \bigotimes R_Y(\theta_i)|0\rangle$ y adaptaron el mezclador $H_M^{(ws)}$ para que $|\phi^*\rangle$ fuera su estado base, demostrando mejoras significativas a bajo p (ver Fig. 4, 5 en [2]).
- **WS-QAOA Redondeado (para Max-Cut):** Utilizaron una solución binaria d^* obtenida del redondeo aleatorio de GW de la relajación SDP para generar el estado inicial $|\phi^*\rangle$ (con regularización ϵ para evitar los polos). Propusieron un mezclador modificado que permitía recuperar la solución GW con $p=1, \beta_1 = \pi/2, \gamma_1 = 0$, garantizando al menos el rendimiento de GW.
- **WS-RQAOA:** Integraron el warm-start redondeado con RQAOA, usando múltiples cortes GW como puntos de partida para QAOA en cada paso recursivo, mejorando los resultados sobre RQAOA estándar (ver Fig. 3, 8 en [2]).

Metodología y Resultados Clave (Tate et al. [6]):

- **QAOA-Warmest con Mezcladores Personalizados:** Generalizaron el warm-starting a cualquier estado inicial separable $|\psi_0\rangle = \bigotimes |s_{0,j}\rangle$. Definieron un *mezclador personalizado* $H_B = \sum H_{B,j}$ basado en las coordenadas de Bloch (x_j, y_j, z_j) de cada $|s_{0,j}\rangle$, tal que $|\psi_0\rangle$ es el estado de *máxima* energía del mezclador.

- **Garantía de Convergencia:** Probaron (Teorema 1 en [6]) que QAOA-warmest converge al óptimo global para $p \rightarrow \infty$. Esto contrasta con QAOA-warm (warm-start con mezclador estándar), que puede estancarse [2, 6].
- **Warm-Starts Específicos:** Generaron $|\psi_0\rangle$ usando proyecciones de baja dimensión ($k=2, 3$) de la solución SDP de GW, o soluciones de la relajación de Burer-Monteiro (BM-MC_k).
- **Rendimiento a $p = 0$ y bajo p :** Probaron ratios de aproximación constantes no triviales a $p = 0$ para los warm-starts de GW proyectado (Corolario 3 en [6]). Numéricamente, demostraron que QAOA-warmest supera a QAOA estándar, QAOA-warm y al algoritmo GW a profundidades bajas ($p \approx 4 - 8$) (ver Fig. 2, Tabla 2 en [6]), validándolo también en hardware real (Figs. 7-9 en [6]).

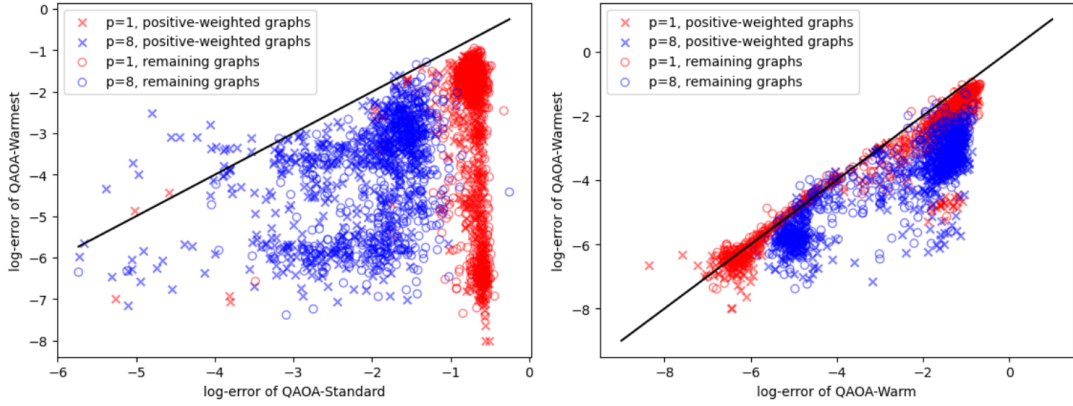


Figura 3: Comparación del log-error de QAOA-warmest vs QAOA estándar (izquierda) y QAOA-warm (derecha) para $p = 1$ y $p = 8$. Puntos bajo la línea negra indican mejor rendimiento de QAOA-warmest. Adaptado de Tate et al. (2023) [6].

5.2.2. QAOA Multi-Ángulo (MA-QAOA)

Esta variante aumenta la expresividad del ansatz QAOA permitiendo más parámetros por capa.

Metodología y Resultados Clave (Gaidai Herrman [5]):

- **Definición:** En lugar de (γ_k, β_k) por capa, MA-QAOA usa $\gamma_{k,ij}$ para cada término $\sigma_i^z \sigma_j^z$ en H_C y $\beta_{k,i}$ para cada σ_i^x en H_B .
- **Reducción de Profundidad ($p > 1$):** Demostraron numéricamente que MA-QAOA alcanza ratios de aproximación (AR) comparables a QAOA estándar con una profundidad p significativamente menor (factores de reducción de hasta 2.5-4) para Max-Cut (ver Fig. 3 y Tabla 2 en [5]).
- **Inicialización 'QAOA Relax':** Propusieron inicializar los parámetros de MA-QAOA usando los parámetros óptimos de QAOA estándar, mostrando mejor rendimiento que la inicialización aleatoria (ver Fig. 2 en [5]).
- **Costo vs Profundidad:** MA-QAOA reduce la profundidad (bueno para NISQ) pero incrementa enormemente el número de parámetros clásicos. Analizando el costo total ($n_{llamadas} \times p$), QAOA estándar puede ser más eficiente para ARs moderadas, pero MA-QAOA alcanza ARs más altas a un costo dado si p es suficientemente grande (ver Fig. 4 en [5]).

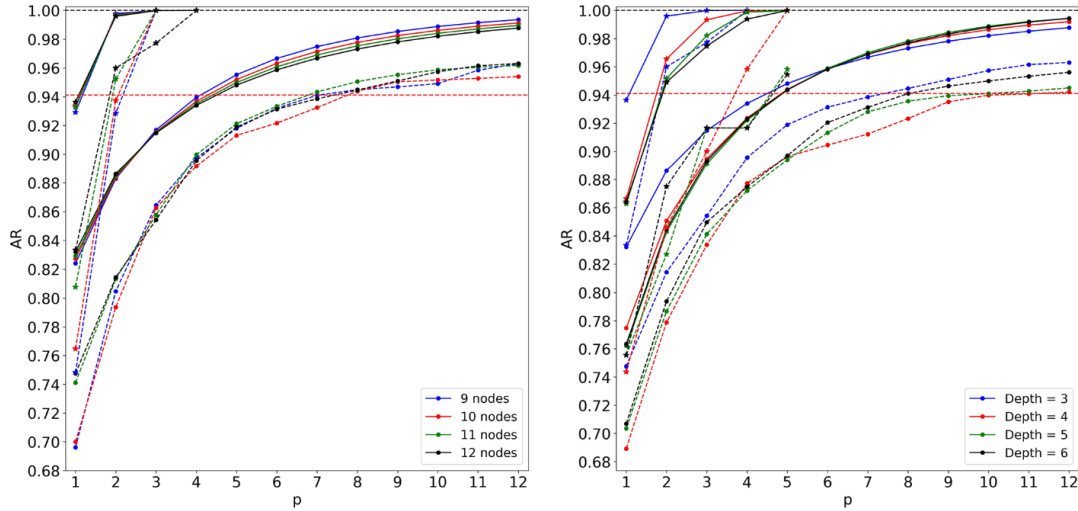


Figura 4: AR vs p para QAOA (círculos) y MA-QAOA (estrellas). Izquierda: Agrupados por N . Derecha: Agrupados por c -depth. Adaptado de Gaidai Herrman (2024) [5].

5.2.3. QAOA Recursivo (RQAOA) y Aprendizaje por Refuerzo

RQAOA reduce iterativamente el tamaño del problema.

Metodología y Resultados Clave (Patel et al. [8]):

- **Limitaciones de RQAOA ($p = 1$):** Identificaron instancias donde RQAOA falla: (i) empates en $\max |\mathcal{M}_{uv}|$ crean ramas en el árbol de búsqueda, y (ii) la elección $\max |\mathcal{M}_{uv}|$ con ángulos óptimos para $\langle H_n \rangle$ puede ser incorrecta (ver Fig. 2 en [8]).
- **RL-RQAOA:** Propusieron usar Aprendizaje por Refuerzo (RL) para aprender una política $\pi_\theta(a|s)$ que elija qué variable eliminar (acción a) en cada estado (grafo s). La política usa softmax sobre $|\mathcal{M}_{uv}|$ con pesos aprendidos β_{uv} , y también aprende los ángulos $(\vec{\alpha}, \vec{\gamma})$ de QAOA (no necesariamente los óptimos para la energía).
- **Mejora de Rendimiento:** RL-RQAOA superó a RQAOA estándar en las instancias difíciles (ver Fig. 5 en [8]) y a un agente RL puramente clásico (RL-RONE), mostrando el valor del componente cuántico (ver Fig. 6 en [8]).

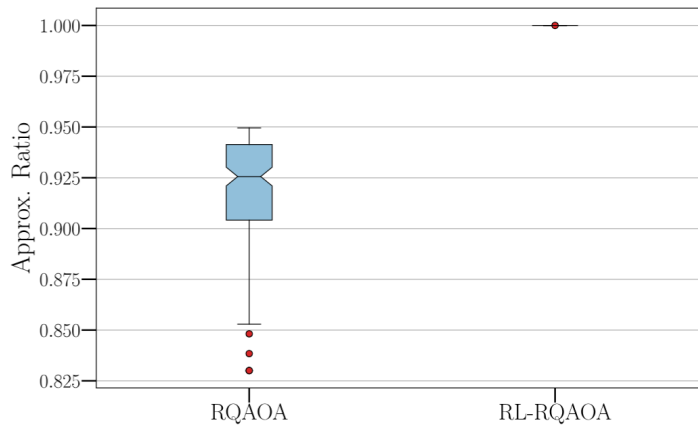


Figura 5: Comparación de AR para RQAOA y RL-RQAOA en instancias difíciles. Adaptado de Patel et al. (2024) [8].

5.3. Corriente 3: Co-diseño Hardware-Software y Mitigación de Errores

5.3.1. Corte de Circuitos (Circuit Cutting)

Esta técnica permite simular circuitos grandes en hardware más pequeño.

Metodología y Resultados Clave (Lowe et al. [9], Bechtold et al. [10]):

- **Concepto:** Dividir un circuito grande en fragmentos ejecutables en hardware con menos cúbits, reemplazando las conexiones cortadas por mediciones y preparaciones, y recombinando clásicamente los resultados [9, 14].
- **Mediciones Aleatorias (Lowe et al. [9]):** Propusieron usar mediciones basadas en 2-designs unitarios (e.g., Cliffords aleatorios) para los cortes, logrando un overhead de muestreo $\tilde{O}(4^k/\epsilon^2)$, una mejora cuadrática sobre métodos previos [14] para k cables paralelos (ver Tabla 1, Fig. 4 en [9]).
- **Aplicación a QAOA (Lowe et al. [9]):** Mostraron cómo cortar circuitos QAOA usando separadores de vértices del grafo, con overhead $\approx 2^{O(p\kappa)}$ para separadores de tamaño κ . Realizaron simulaciones a gran escala (hasta 129q simulados en fragmentos de 30q) (ver Figs. 6, 7 en [9]).
- **Validación Experimental y Mitigación de Ruido (Bechtold et al. [10]):** Demostraron en hardware real (IBM) que el corte permite ejecutar QAOA para Max-Cut en problemas mayores a la capacidad del chip. Observaron que el corte puede mejorar los resultados en presencia de ruido al reducir la profundidad efectiva de los fragmentos.

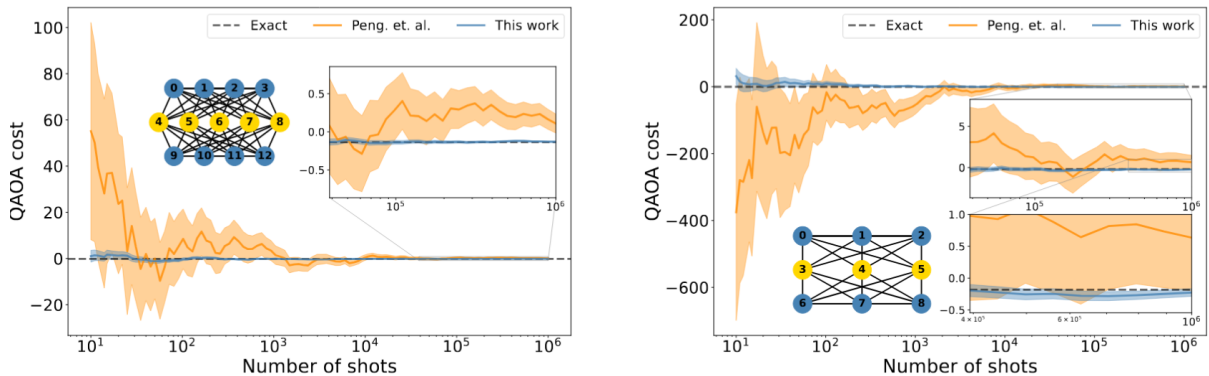


Figura 6: Comparación de varianza del costo QAOA vs tamaño del corte para Lowe et al. ('This work') y Peng et al. Adaptado de Lowe et al. (2023) [9].

6. Análisis Comparativo y Síntesis

La tabla 1 resume las contribuciones clave y su relevancia en el avance de QAOA para Max-Cut.

7. Conclusiones

El estudio de QAOA para el problema de Max-Cut ha evolucionado significativamente desde su propuesta original, transformándose en un ecosistema dinámico de investigación centrado en superar las limitaciones del hardware NISQ. La conclusión inicial de Guerreschi y Matsuura

Cuadro 1: Comparación de artículos clave del estado del arte sobre QAOA para Max-Cut.

Autores (Año)	Enfoque	Contribución Principal	Relevancia para el Campo
Guerreschi & Matsuura (2019) [4]	Simulación con ruido	Establece la necesidad de cientos de cúbits para la ventaja cuántica en implementación ingenua.	Fija la línea base del desafío NISQ.
Zhou et al. (2020) [1]	Optimización Clásica	Descubre patrones en parámetros óptimos y propone heurísticas (INTERP, FOURIER) para encontrarlos eficientemente ($O(\text{poly}(p))$).	Resuelve un cuello de botella práctico clave: el bucle de optimización clásico. Revela mecanismos no adiabáticos.
Egger et al. (2021) [2]	Híbrido Algorítmico	Propone el warm-starting usando soluciones de relajaciones SDP/QP clásicas. Introduce WS-RQAOA con GW.	Introduce sinergia cuántico-clásica, heredando garantías de rendimiento clásicas.
Tate et al. (2023) [6]	Algorítmico / Teórico	Combina warm-starting con mezcladores personalizados que garantizan la convergencia al óptimo global. Supera a GW a bajo p .	Soluciona debilidad del warm-starting (convergencia) y demuestra rendimiento superior a bajo p .
Gaidai & Herrman (2024) [5]	Algorítmico	Introduce MA-QAOA, que reduce la profundidad del circuito a cambio de más parámetros. Propone inicialización 'QAOA Relax'.	Ofrece un trueque (profundidad vs. parámetros) ventajoso para NISQ.
Patel et al. (2024) [8]	Híbrido / IA	Mejora RQAOA utilizando aprendizaje por refuerzo para guiar la eliminación de variables.	Moderniza RQAOA con IA para tomar decisiones más inteligentes, mejorando rendimiento en casos difíciles.
Lowe et al. (2023) [9]	Co-diseño / Teórico	Desarrolla corte de circuitos con mediciones aleatorias que reduce cuadráticamente el costo computacional clásico ($\tilde{O}(4^k)$ overhead).	Mejora significativamente la escalabilidad de una técnica clave de mitigación y adaptación al hardware.
Bechtold et al. (2023) [10]	Co-diseño / Experimental	Demuestra experimentalmente el corte de circuitos para QAOA en hardware real. Observa mejora de resultados por mitigación de ruido.	Valida experimentalmente el corte de circuitos y su potencial beneficio contra el ruido.

[4] sobre la necesidad de cientos de cúbits para obtener una ventaja cuántica sigue siendo un referente, pero el campo ha respondido con una notable sofisticación técnica.

El estado del arte actual revela que el progreso hacia la utilidad práctica de QAOA no depende de una única solución mágica, sino de un enfoque multifacético que integra avances en distintos niveles:

- La **optimización clásica de parámetros**, antes una barrera potencialmente exponencial, ahora es abordable mediante heurísticas eficientes que explotan la estructura de los parámetros óptimos [1].
- El **algoritmo cuántico** se ha enriquecido con variantes poderosas. El *warm-starting* con mezcladores personalizados (QAOA-Warmest) [6] no solo acelera la convergencia a bajas profundidades, superando incluso a GW, sino que también garantiza la convergencia teórica. *MA-QAOA* [5] ofrece una vía para reducir la profundidad del circuito, crucial en NISQ. *RQAOA*, especialmente potenciado con RL [8], proporciona un mecanismo iterativo para escalar a problemas más grandes.
- La **interacción con el hardware** se aborda proactivamente con técnicas de co-diseño como el *corte de circuitos* [9, 10], que no solo permite ejecutar problemas más grandes en máquinas pequeñas, sino que también puede actuar como una forma de mitigación de errores.

Max-Cut ha sido el catalizador y el campo de pruebas para esta evolución, permitiendo comparar rigurosamente estas estrategias. Aunque la demostración inequívoca de ventaja cuántica a gran escala para problemas de optimización sigue siendo un objetivo a futuro, los avances aquí revisados indican un progreso tangible y una hoja de ruta definida. El futuro de la optimización cuántica con QAOA probablemente radique en la **combinación inteligente y adaptativa** de estas técnicas (optimización de parámetros, variantes algorítmicas, co-diseño/mitigación), creando algoritmos híbridos robustos capaces de extraer valor práctico de los dispositivos cuánticos disponibles, tanto en la era NISQ como más allá.

Referencias

- [1] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Physical Review X*, 10(2):021067, 2020.
- [2] Daniel J Egger, Jakub Mareček, and Stefan Woerner. Warm-starting quantum optimization. *Quantum*, 5:479, 2021.
- [3] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.
- [4] Gian Giacomo Guerreschi and Anne Y Matsuura. Qaoa for max-cut requires hundreds of qubits for quantum speed-up. *Scientific Reports*, 9(1):6903, 2019.
- [5] Igor Gaidai and Rebekah Herrman. Performance analysis of multi-angle qaoa for p41. *Scientific Reports*, 14(1):18911, 2024.
- [6] Reuben Tate, Jai Moondra, Bryan Gard, Greg Mohler, and Swati Gupta. Warm-started qaoa with custom mixers provably converges and computationally beats goemans-williamson’s max-cut at low circuit depths. *Quantum*, 7:1137, 2023.
- [7] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995.
- [8] Yash J Patel, Sofiene Jerbi, Thomas Bäck, and Vedran Dunjko. Reinforcement learning assisted recursive qaoa. *EPJ Quantum Technology*, 11(1):6, 2024.
- [9] Angus Lowe, Matija Medvidović, Anthony Hayes, Lee J O’Riordan, Thomas R Bromley, Juan Miguel Arrazola, and Nathan Killoran. Fast quantum circuit cutting with randomized measurements. *Quantum*, 7:934, 2023.
- [10] Christoph Bechtold, Johanna Barzen, Frank Leymann, Alexander Mandl, Johanna PD Moritz, and Karoline Wild. Investigating the practical application of circuit cutting for qaoa. *Quantum Science and Technology*, 8(4):045020, 2023.
- [11] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001.
- [12] Bálint Bángelo and Dirk-André Deckert. Noise gates for decoherent quantum circuits. *Phys. Rev. A*, 77:032323, Mar 2008.
- [13] A. Kugel. Improved exact solver for the weighted max-sat problem. *Proc. Pragmatics of SAT Workshop (POS-10)*, 8:15–27, 01 2012.
- [14] Tianyi Peng, Aram W. Harrow, Maris Ozols, and Xiaodi Wu. Simulating large quantum circuits on a small quantum computer. *Phys. Rev. Lett.*, 125:150504, Oct 2020.