Desarrollo de un algoritmo cuántico para planificación de rutas en un espacio discreto tridimensional para la navegación de un Vehículo Aéreo No tripulado (UAV) mediante Recocido Cuántico (QA)
Isaac Oliva González
2026



Universidad Autónoma de Querétaro
Facultad de Informática

Desarrollo de un algoritmo cuántico para planificación de rutas en un espacio discreto tridimensional para la navegación de un Vehículo Aéreo No tripulado (UAV) mediante Recocido Cuántico (QA)

Tesis

Que como parte de los requisitos para obtener el Grado de

Maestro en Ciencias de la Computación

Presenta

Ing. Isaac Oliva González

Dirigido por:

Dr. Hugo Jiménez Hernández

Querétaro, Qro. a 26 de noviembre de 2026

REPOSITORIO INSTITUCIONAL DGBSDI-UAQ

La presente obra está bajo la licencia:
https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es

CC BY-NC-ND 4.0 DEED

Atribución-NoComercial-SinDerivadas 4.0 Internacional

Usted es libre de:

Compartir — copiar y redistribuir el material en cualquier medio o formato

La licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

Bajo los siguientes términos:

Atribución — Usted debe dar crédito de manera adecuada , brindar un enlace a la licencia, e indicar si se han realizado cambios . Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante.

NoComercial — Usted no puede hacer uso del material con propósitos comerciales .

SinDerivadas — Si remezcla, transforma o crea a partir del material, no podrá distribuir el material modificado.

No hay restricciones adicionales — No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

Avisos:

No tiene que cumplir con la licencia para elementos del material en el dominio público o cuando su uso esté permitido por una excepción o limitación aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como publicidad, privacidad, o derechos morales pueden limitar la forma en que utilice el material.

Universidad Autónoma de Querétaro
Facultad de Informática
Maestría en Ciencias de la Computación

Desarrollo de un algoritmo cuántico para planificación de rutas en un espacio discreto tridimensional para la navegación de un Vehículo Aéreo No tripulado (UAV) mediante Recocido Cuántico (QA)

Tesis

Que como parte de los requisitos para obtener el Grado de

Maestro en Ciencias de la Computación

Presenta

Ing. Isaac Oliva González

Dirigido por

Dr. Hugo Jiménez Hernández

Dr. Hugo Jiménez Hernández
Presidente

Dr. Jorge Domingo Mendiola
Secretario

Dra. Sandra Luz Canchola
Vocal

Dra. Ana Marcela Herrera Navarro
Suplente

MC. Daniel Cantón Enríquez
Suplente

Centro Universitario, Querétaro, Qro. México
Noviembre 2026

Agradecimientos

A la Universidad Autónoma de Querétaro y a su Facultad de Informática les agradezco por haberme brindado la oportunidad de formar parte de esta institución, así como por el apoyo recibido para el desarrollo de esta tesis de maestría.

Doy también las gracias a los docentes de la facultad por su enseñanza y acompañamiento durante la maestría, y al Dr. Hugo Jiménez, director de esta tesis, por su guía y apoyo.

Agradezco el financiamiento otorgado por la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) para la realización de este proyecto.

Resumen

Planificar rutas para Vehículos Aéreos No Tripulados (UAV) en entornos tridimensionales es un problema de optimización combinatoria. La complejidad crece de forma exponencial con la escala del entorno, así que enumerar todas las rutas posibles no es viable; se requieren métodos que encuentren soluciones de calidad sin recorrer todo el espacio de búsqueda.

Este trabajo estudia el problema del camino más corto para un par origen-destino (SPSP) sobre un grafo tridimensional ponderado no dirigido. Cada arista recibe un costo que combina la distancia euclidiana y una penalización por ganancia de altitud. El problema se formula como un modelo de Optimización Binaria Cuadrática No Restringida (QUBO), donde las restricciones estructurales de la ruta —conectividad, continuidad y que no se ramifique— pasan a ser penalizaciones cuadráticas dentro de una única función objetivo. Esta representación funciona con sistemas de Recocido Cuántico (QA) y ofrece un marco común para implementar tanto el Recocido Simulado clásico (SA) como un algoritmo de inspiración cuántica que reproduce la dinámica del Recocido Cuántico.

La comparación usa la métrica *Time to Solution* (TTS), que reúne el tiempo de cómputo y la probabilidad de hallar el óptimo global con alta confianza bajo el mismo presupuesto de evaluaciones para ambos algoritmos. Se probaron doce instancias del problema, variando escala del espacio de navegación, tipo de entorno y régimen de penalización por altitud, para identificar en qué condiciones conviene más cada enfoque.

Ambos algoritmos encuentran el óptimo teórico en todas las instancias evaluadas. Al crecer la escala, SA conserva una probabilidad de éxito notablemente más alta: su caída es moderada y sostenida, mientras que el algoritmo cuántico se degrada con mucha más rapidez y deja de ser competitivo en las instancias grandes. Esto se nota en el TTS: SA supera al modelo cuántico en casi todas las instancias y la brecha crece con la escala. La excepción es la instancia más pequeña con espacio de navegación restringido, donde el desempeño de ambos es comparable. Estos resultados permiten identificar cuándo el paradigma cuántico puede competir con el clásico, y sientan bases para escalar estas metodologías hacia hardware cuántico real.

Palabras clave: planificación de rutas, vehículos aéreos no tripulados, recocido simulado, recocido simulado cuántico, QUBO.

Abstract

Path planning for Unmanned Aerial Vehicles (UAV) in three-dimensional environments is a combinatorial optimization problem. Its complexity grows exponentially with scale, so exhaustive enumeration of trajectories is not feasible; methods must find quality solutions without traversing the entire search space.

This work tackles the single-pair shortest path problem (SPSP) on an undirected weighted three-dimensional graph. Each edge cost combines Euclidean distance with a penalty for altitude gain. The problem is cast as a Quadratic Unconstrained Binary Optimization (QUBO) model, where structural path constraints (connectivity, continuity, no branching) become quadratic penalties inside a single objective function. That representation works natively with Quantum Annealing (QA) systems and provides a common ground to implement and compare classical Simulated Annealing (SA) against a quantum-inspired algorithm that mimics QA dynamics via path-integral methods (Simulated Quantum Annealing, SQA).

The comparison relies on the Time to Solution (TTS) metric, which captures both computational time and the probability of reaching the global optimum with high confidence, under the same evaluation budget for both algorithms. Twelve problem instances are tested, varying navigation space scale, environment type, and altitude penalty regime, to see where each paradigm fits best.

Both algorithms reach the theoretical optimum in every instance tested. SA keeps a notably higher success probability as scale increases — its decline is moderate and steady, while the quantum-inspired algorithm degrades far more sharply and falls behind in the largest instances. The TTS metric captures this: SA beats the quantum model in nearly every case, and the gap grows with scale. The exception is the smallest instance with restricted navigation space, where both perform similarly. These findings help clarify when the quantum paradigm can compete with the classical one, and set the stage for scaling these methods toward real quantum hardware.

Keywords: route planning, unmanned aerial vehicles, simulated annealing, simulated quantum annealing, QUBO.

Índice

Resumen i

Abstract ii

Índice iii

Índice de tablas v

Índice de figuras vi

1 Introducción **1**
- 1.1 Descripción del problema . . . 1
- 1.2 Justificación . . . 2
- 1.3 Organización de capítulos . . . 3
- 1.4 Hipótesis . . . 3
- 1.5 Objetivo general . . . 3
  - 1.5.1 Objetivos específicos . . . 3
- 1.6 Alcances y limitaciones . . . 4
- 1.7 Publicaciones . . . 4

2 Antecedentes **5**
- 2.1 Planificación de rutas para vehículos aéreos no tripulados . . . 5
- 2.2 Optimización combinatoria en planificación de rutas . . . 5
- 2.3 Enfoques de solución para planificación de rutas . . . 7
  - 2.3.1 Métodos exactos . . . 7
  - 2.3.2 Enfoques heurísticos y metaheurísticos . . . 9
  - 2.3.3 Enfoques cuánticos . . . 9

3 Marco Teórico **11**
- 3.1 Optimización combinatoria . . . 11
  - 3.1.1 Problemas de optimización con variables binarias . . . 11
  - 3.1.2 Modelo Ising y formulación QUBO . . . 12
- 3.2 Recocido simulado . . . 16
- 3.3 Recocido cuántico . . . 20
  - 3.3.1 Proceso de recocido cuántico . . . 22
  - 3.3.2 Recocido cuántico simulado . . . 26

4 Metodología **30**
- 4.1 Definición del problema de navegación . . . 30
  - 4.1.1 Grafo tridimensional ponderado . . . 31
  - 4.1.2 Problema del camino más corto (3D-SPSP) . . . 32
- 4.2 Formulación QUBO del problema 3D-SPSP . . . 33

4.3 Algoritmos de optimización 35
4.3.1 Recocido simulado 36
4.3.2 Recocido cuántico simulado 36
4.4 Evaluación comparativa 37
4.4.1 Instancias de prueba 37
4.4.2 Configuración experimental 41
4.4.3 Métrica de desempeño: *Time to Solution* 42

5 Resultados y discusión **44**
5.1 Análisis de convergencia 44
5.2 Probabilidad de éxito y escalabilidad 48
5.3 Comparación global mediante *Time to Solution* 49
5.4 Discusión 50
5.4.1 Limitaciones 52

6 Conclusiones y trabajo futuro **54**
6.1 Trabajo futuro 54

Bibliografía **56**

Índice de tablas

2.1 Comparación de enfoques de solución para planificación de rutas 10
3.1 Correspondencia entre recocido cuántico (QA) y recocido cuántico simulado (SQA). 29
4.1 Dimensionalidad del problema QUBO y densidad del grafo para las dos variantes del espacio de navegación en las tres escalas. 40
4.2 Parámetros estructurales y mínimos teóricos de las doce instancias del problema 3D-SPSP. 41
4.3 Parámetros de SA y SQA empleados en los experimentos. 42
5.1 Resultados de convergencia para las doce instancias del problema 3D-SPSP. Las métricas se estiman sobre $R = 100$ ejecuciones independientes por instancia y algoritmo. 45
5.2 Probabilidad de éxito p_s y número mínimo de ejecuciones N requeridas para alcanzar $p_d = 0{,}99$. 49
5.3 Resumen de *Time to Solution* (TTS) para SA y SQA en las doce instancias ($p_d = 0{,}99$). El Factor es la razón $\mathrm{TTS}_{\mathrm{SQA}}/\mathrm{TTS}_{\mathrm{SA}}$. 50

Índice de figuras

2.1 Degeneración en optimización combinatoria. Las rutas $P^{(1)}$ y $P^{(2)}$ son dos configuraciones discretas distintas que conectan el origen x_s con el destino x_t y alcanzan el mismo óptimo global. 6
3.1 Paisaje de energía con múltiples mínimos locales y un mínimo global. . 12
3.2 Estructura de la matriz QUBO y su interpretación como grafo ponderado. 14
3.3 Componentes principales del algoritmo de recocido simulado. 17
3.4 Probabilidad de aceptación del *criterio de Metropolis* para distintas temperaturas ante un mismo incremento de energía Δf_0. 19
3.5 Componentes principales del proceso de recocido cuántico. 23
3.6 Evolución del Hamiltoniano durante el proceso de recocido cuántico. . . 25
3.7 Comparación entre los mecanismos de escape de mínimos locales: activación térmica en el recocido simulado (SA) y tunelización cuántica en el recocido cuántico (QA). 26
3.8 Componentes del algoritmo de recocido cuántico simulado. 27
4.1 Flujo metodológico para la evaluación comparativa de recocido simulado y recocido cuántico simulado aplicados al problema 3D-SPSP. 30
4.2 Instancias para $k = 5$: (a) G_{abierto} con $|V| = 125$ vértices y $|E| = 300$ aristas ($\rho = 0{,}0387$); (b) $G_{\text{restringido}}$ con $|V| = 125$ vértices y $|E| = 217$ aristas ($\rho = 0{,}0281$), donde los obstáculos volumétricos eliminan 83 aristas de la retícula. El vértice fuente s se indica en verde y el destino t en azul. 39
5.1 Distribución de energías por $R = 100$ ejecuciones en las doce instancias del 3D-SPSP. Las filas corresponden a las escalas $k \in \{5, 6, 7\}$ y las columnas a los espacios de navegación G_{abierto} y $G_{\text{restringido}}$. Cada panel contiene cuatro diagramas de cajas: SA y SQA bajo los regímenes de costo $\eta = 0$ y $\eta = 1$. La línea horizontal indica el óptimo teórico $f(\mathbf{x}^*)$. 47
5.2 Evolución de p_s con la escala $k \in \{5, 6, 7\}$ para SA y SQA. Cada curva corresponde a una combinación de espacio de navegación (G_{abierto}, $G_{\text{restringido}}$) y régimen de costo ($\eta \in \{0, 1\}$). 48
5.3 TTS en función de k para SA y SQA en escala logarítmica. Las curvas se muestran para G_{abierto} y $G_{\text{restringido}}$, promediadas sobre $\eta \in \{0, 1\}$. El rango de valores abarca de 0.30 s a 118.87 s. 51
5.4 Rutas óptimas para $k = 5$, $\eta = 1$. a) espacio abierto G_{abierto}; b) espacio restringido $G_{\text{restringido}}$ con obstáculos. El vértice fuente s se indica en verde y el destino t en azul. 51

1. Introducción

Planificar la ruta de un Vehículo Aéreo No Tripulado (UAV) en tres dimensiones es un problema de optimización combinatoria. Los drones se emplean en búsqueda y rescate, vigilancia urbana, reconocimiento y logística (Arshad et al., 2025), escenarios que necesitan trayectorias seguras y eficientes. El reto va más allá de la geometría: el número de rutas posibles crece de forma explosiva con el tamaño del entorno, al punto que los algoritmos determinísticos no pueden barrer el espacio de búsqueda completo. La ruta óptima es la que minimiza distancia, consumo energético o riesgo de colisión.

La computación cuántica abre otra vía. El Recocido Cuántico (QA) aprovecha la superposición y la tunelización para explorar el espacio de soluciones sin seguir una trayectoria secuencial (Kadowaki and Nishimori, 1998). El acceso a hardware cuántico real todavía es limitado: los dispositivos operan con ruido en el régimen NISQ (Noisy Intermediate-Scale Quantum) (Preskill, 2018), pero hay simuladores clásicos que reproducen esta dinámica con buena fidelidad.

Se aborda el problema del camino más corto entre un par origen-destino (SPSP) sobre un grafo tridimensional ponderado. El grafo modela la navegación de un UAV en un espacio discreto, y la formulación que se emplea es QUBO (Optimización Binaria Cuadrática No Restringida) (Glover et al., 2019a). Este modelo sirve de lenguaje común: compatible con hardware cuántico y también con algoritmos clásicos. Sobre esa misma representación corren tanto el Recocido Simulado (SA) como el Recocido Cuántico Simulado (SQA), cada uno con su propia lógica de exploración.

La comparación se apoya en la métrica *Time to Solution* (TTS), que condensa en una cifra el tiempo de cómputo y la probabilidad de dar con una solución de cierta calidad (Rønnow et al., 2014). La pregunta de fondo es qué tanto puede aportar la computación cuántica a problemas de optimización con restricciones espaciales.

1.1. Descripción del problema

Dados un origen s y un destino t en un espacio tridimensional discreto, se busca la ruta que los conecte con el menor costo acumulado. Esto se conoce como el problema del camino más corto para un par origen-destino (SPSP). El costo admite interpretarse como distancia recorrida, energía consumida o riesgo de colisión. Cualquier ruta candidata está obligada a respetar la conectividad y la continuidad que impone la dinámica de vuelo.

El entorno se representa con un grafo no dirigido y ponderado $G = (V, E, C, p)$. Los vértices $v \in V$ son posiciones en $\mathbb{Z}^3$ definidas por $p(v)$, y las aristas $\{i, j\} \in E$ corresponden a las transiciones permitidas entre vecinos, cada una con un costo $c_{ij} \geq 0$. Ese costo conjunta la distancia euclidiana entre los extremos y una penalización adicional cuando la ruta gana altitud, lo que da cuenta de que los ascensos consumen más energía. Una ruta es válida si sus aristas consecutivas pertenecen a E, no repite vértices y todo nodo intermedio tiene grado exactamente dos dentro de la ruta.

Para una retícula de $k \times k \times k$ nodos, el grafo contiene $|V| = k^3$ vértices y $3k^2(k-1)$ aristas. El número de rutas posibles aumenta con k, y para valores moderados recorrer todas las opciones ya no es viable. Bajo estas condiciones, el SPSP es NP-difícil (Pardalos and Jha, 1992). Se necesitan métodos que produzcan buenas soluciones sin examinar el espacio entero.

El SPSP se formula como un modelo QUBO: cada vértice y cada arista del grafo quedan asociados a una variable binaria. La función objetivo acumula los costos de las aristas activas y suma penalizaciones cuadráticas que obligan a trayectorias continuas, sin ciclos y sin ramificaciones. Esta misma formulación alimenta tanto al Recocido Simulado como al Recocido Cuántico Simulado, lo que permite compararlos en igualdad de condiciones mediante la métrica *Time to Solution*.

1.2. Justificación

Dentro de enfoques clásicos de optimización, estos funcionan en instancias pequeñas, pero se vuelven intratables al crecer la escala del entorno tridimensional, las aplicaciones reales de UAVs empleados en espacios donde la combinatoria del problema sobrepasa la capacidad de exploración exhaustiva (Tu et al., 2020; Koritsoglou et al., 2022a).

La computación cuántica emplea una alternativa, el Recocido Cuántico (QA) no explora el espacio de forma secuencial si no usa tunelización cuántica para atravesar barreras de energía que atraparían a un método clásico en un mínimo local. Esta diferencia en el mecanismo de exploración justifica estudiar su aplicación a la planificación de rutas y las condiciones del trabajo orientan la implementación hacia un entorno de simulación clásica, teniendo asi un fundamento conceptual es el mismo.

Al comparar ambos paradigmas exige un marco matemático que ambos compartan sin alterar la codificación del problema. Se eligió QUBO (Punnen, 2022) por tres razones:

- Es el modelo nativo de los sistemas de Recocido Cuántico, así que funciona con hardware cuantico y simuladores sin necesidad de transformaciones adicionales.
- Representa de forma natural las decisiones binarias del problema sin transformaciones intermedias.
- Traduce las restricciones estructurales del camino (conectividad, ausencia de ciclos, continuidad) en penalizaciones cuadráticas dentro de la misma función objetivo.

El desempeño se mide con *Time to Solution* (TTS), que junta tiempo computacional y probabilidad de éxito en una sola cifra. Esto permite comparar métodos que difieren en su naturaleza interna con un presupuesto de evaluaciones equivalente, sin favorecer artificialmente a ninguno.

El trabajo aporta evidencia sobre las condiciones en que el paradigma cuántico resulta competitivo frente al clásico en problemas de optimización combinatoria con

restricciones espaciales, y permite entender sus fortalezas y límites en escenarios de navegación autónoma.

1.3. Organización de capítulos

El trabajo se estructura en seis capítulos. El Capítulo 1, Introducción, plantea el problema de la planificación de rutas para Vehículos Aéreos No Tripulados en espacios tridimensionales discretos, acompañado de la justificación que motiva el estudio, la hipótesis que guía la investigación, los objetivos general y específicos, así como los alcances y limitaciones del trabajo. El Capítulo 2, Antecedentes, ofrece una revisión del estado del arte en torno a la planificación de rutas, la optimización combinatoria y las distintas familias de métodos de solución. El Capítulo 3, Marco Teórico, muestra las bases conceptuales necesarias para el desarrollo del trabajo: la optimización combinatoria, la formulación QUBO, el recocido simulado clásico y el recocido cuántico simulado. El Capítulo 4, Metodología, detalla la representación del entorno mediante un grafo tridimensional, la formulación del problema de camino más corto (3D-SPSP) como un modelo QUBO, los algoritmos de optimización implementados y el diseño experimental seguido para su evaluación. El Capítulo 5, Resultados y discusión, presenta y analiza los hallazgos obtenidos en términos de convergencia, probabilidad de éxito, escalabilidad y tiempo de solución, ofreciendo una interpretación crítica de los mismos. Por ultimo, el Capítulo 6, Conclusiones y trabajo futuro, expone las reflexiones finales y las líneas de continuación del estudio.

1.4. Hipótesis

Si se toma como referencia un algoritmo clásico de recocido simulado que permite encontrar rutas óptimas en un grafo denso y se utilizan las propiedades de la computación cuántica para codificar la información como un modelo cuántico, entonces se podrá desarrollar un algoritmo cuántico de recocido simulado capaz de encontrar rutas óptimas de manera eficiente, incluso cuando el espacio de búsqueda crece exponencialmente.

1.5. Objetivo general

Desarrollar e implementar en un ambiente simulado un algoritmo de recocido cuántico para buscar soluciones eficientes en el problema de planificación de rutas en espacios discretos tridimensionales para la navegación de Vehículos Aéreos No Tripulados (UAV), con el fin de mejorar la eficiencia y efectividad en la búsqueda de soluciones óptimas en comparación con los métodos clásicos tradicionales.

1.5.1. Objetivos específicos

- Implementar el algoritmo en un modelo clásico (recocido simulado), para la resolución de un problema de optimización de planificación de rutas 3D para la navegación de un UAV.

- Investigar las propiedades de los sistemas de Recocido Cuántico para su aplicación en la resolución de problemas de optimización NP-difícil basado en el modelo de Optimización Binaria Cuadrática No Restringida (QUBO).
- Diseñar un algoritmo cuántico adaptado a un sistema de Recocido Cuántico para la resolución de un problema de optimización de planificación de rutas 3D para la navegación de un UAV.
- Implementar el algoritmo cuántico desarrollado en un sistema de Recocido Cuántico mediante hardware cuántico.
- Evaluar el desempeño del algoritmo cuántico contrastándolo con su contraparte clásica para llevar a cabo una comparativa en la resolución del problema de optimización.

1.6. Alcances y limitaciones

Se desarrolla una formulación QUBO para planificación de rutas 3D de un UAV. SA y SQA se implementan y comparan con el mismo presupuesto de evaluaciones, variando escala, espacio de navegación y régimen de costo, bajo la métrica *Time to Solution* (TTS). Los resultados muestran tendencias de escalabilidad y eficiencia relativa entre ambos. El estudio se limita a bit-flip simple, hiperparámetros fijos y pocas escalas. Se considera entornos estáticos con obstáculos fijos y un UAV; no incluye comparación directa con algoritmos exactos ni hardware cuántico real, y explora un solo valor de penalización por ascenso.

1.7. Publicaciones

Como parte del desarrollo de esta tesis, se genero una publicación arbitrada de acceso abierto, donde se presenta una metodología basada en la formulación QUBO y el algoritmo de recocido simulado para la resolución del problema del camino más corto en redes de transporte urbano, teniendo asi su diseño, fundamentos teóricos y evaluación sobre instancias sintéticas y un caso de estudio real.

- Oliva-González, I., Jiménez-Hernández, H. (2026). A QUBO-Driven Simulated Annealing Methodology for Solving the Shortest Path Problem in Urban Transportation Networks. *Algorithms*, 19(5), 352. https://www.mdpi.com/1999-4893/19/5/352

2. Antecedentes

La planificación de rutas en espacios tridimensionales es un problema de optimización combinatoria dentro de la ingeniería de sistemas autónomos, donde los UAVs operan en entornos no estructurados se necesitan algoritmos capaces de manejar espacios de configuración de alta dimensionalidad con restricciones operativas y costos multiobjetivo. En este capítulo se muestra el problema dentro del estado del arte, señalando las fronteras de tratabilidad de los enfoques deterministas y justificar el uso de formulaciones basados en programacion matematica con algoritmos de recocido para resolver problemas de rutas en grafos tridimensionales.

2.1. Planificación de rutas para vehículos aéreos no tripulados

Los Vehículos Aéreos No Tripulados se usan en aplicaciones como búsqueda y rescate, inspección de infraestructura y logística autónoma. Dentro de estos escenarios la viabilidad del sistema depende de la capacidad computacional para encontrar rutas factibles y óptimas respecto a los recursos limitados.

El problema consiste en determinar una ruta en el espacio de configuración del vehículo que conecte un estado inicial con un objetivo, minimizando asi una función de costo y satisfaciendo restricciones de desigualdad $g(x) \leq 0$ impuestas por la dinámica del UAV y geometría de los obstáculos. En UAVs, estas restricciones tienen cotas de altitud, zonas de exclusión coomo obstaculos y límites de consumo energético.

Para este problema, el espacio continuo se discretiza en un grafo $G = (V, E, C)$, en donde los vértices V muestran posiciones válidas, y las aristas E modelan las transiciones permitidas. La complejidad está en la dimensionalidad del grafo, $|V|$ el cual crece lo suficiente para complicar la enumeración exhaustiva.

En este enfoque, el problema del camino más corto entre un par de nodos (SPSP) es la estructura fundamental. Dado el grafo ponderado G, se busca una secuencia de vértices que minimice el costo total entre un origen $s \in V$ y un destino $t \in V$. La versión tradicional del SPSP se resuelve en tiempo polinomial con algoritmos como Dijkstra (Dijkstra, 1959), pero al añadir restricciones en la trayectoria el problema se vuelve NP-difícil e intratable para instancias grandes (Punnen, 2022).

El SPSP tiene aplicaciones en sistemas de transporte inteligente (Tu et al., 2020), modelado de dinámica peatonal (Koritsoglou et al., 2022b) y optimización de topologías en redes de cómputo (Liu et al., 2021).

2.2. Optimización combinatoria en planificación de rutas

La planificación de rutas en espacios discretizados es un problema de optimización combinatoria (Papadimitriou and Steiglitz, 1998; Korte and Vygen, 2012). A diferencia de lo que ocurre en optimización continua, donde las variables toman valores reales en un intervalo, el conjunto de soluciones factibles Ω es finito y discreto. La solución no es un valor numérico, sino una estructura discreta x: una secuencia ordenada de

aristas seleccionada entre un número enorme de permutaciones que crece factorial o exponencialmente con el tamaño del grafo (LaValle, 2006).

Cada punto $x \in \Omega$ es una ruta completa que satisface las restricciones de conectividad y viabilidad. La "altura" es el valor de la función objetivo $f : \Omega \to \mathbb{R}$, que mide el costo de esa configuración (Reidys and Stadler, 2002). El objetivo es encontrar x^* que minimice globalmente esta función.

Estos espacios combinatorios no son convexos, al ser Ω discreto, no hay derivadas que permitan transiciones entre soluciones vecinas, así que los métodos basados en gradientes ∇f no son aplicables (Blum and Roli, 2003). El espacio tiene mínimos locales, siendo configuraciones x_{loc} con costo menor que sus vecinos inmediatos, pero mayor que el mínimo global x^*. Esta estructura irregular dificulta la convergencia de la búsqueda local simple.

La degeneración se suma a la no convexidad (Gal, 1995) y ocurre cuando la función objetivo no es inyectiva, siendo el conjunto de minimizadores globales $S_{opt} = \{x \in \Omega \mid f(x) = f(x^*)\}$ tiene más de un elemento ($|S_{opt}| > 1$) y el óptimo global no es un único punto, sino un conjunto de soluciones distintas con el mismo costo mínimo (Bertsimas and Tsitsiklis, 1997).

La Figura 2.1 ilustra este fenómeno.

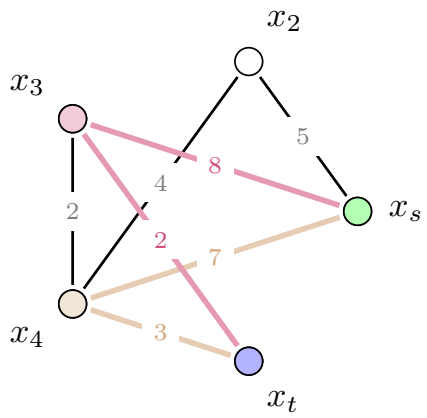


Figura 2.1: Degeneración en optimización combinatoria. Las rutas $P^{(1)}$ y $P^{(2)}$ son dos configuraciones discretas distintas que conectan el origen x_s con el destino x_t y alcanzan el mismo óptimo global.

Dos rutas distintas son indistinguibles para la función de costo. $P^{(1)}$ sigue la secuencia (x_s, x_4, x_t), con costo $7 + 3 = 10$. $P^{(2)}$ sigue (x_s, x_3, x_t), con pesos $8 + 2$, y su costo también es 10. Eso es la degeneración: al menos dos soluciones óptimas distintas.

Para el diseño de algoritmos la degeneración es importante, ya que un método determinista puede sesgarse hacia una sola ruta, pero los enfoques estocásticos, como el recocido simulado (Kirkpatrick et al., 1983), pueden explotar la multiplicidad de estados fundamentales mediante fluctuaciones térmicas o tunelización para mejorar la probabilidad de convergencia.

El reto no es el cómputo aritmético de los costos, sino la naturaleza discreta, no convexa y frecuentemente degenerada del espacio de búsqueda. El navegar eficientemente por estos paisajes es el problema algorítmico y explica por qué se usan métodos estocásticos como el recocido, que sortean mínimos locales y aprovechan la multiplicidad de soluciones óptimas.

2.3. Enfoques de solución para planificación de rutas

Existen varias estrategias para planificación de rutas que aparecen en la literatura, diferenciables principalmente por sus garantías de convergencia, eficiencia asintótica y capacidad para manejar la no convexidad. Esta sección revisa los enfoques más relevantes, desde métodos deterministas, aproximaciones estocásticas y basados en computo cuántico.

2.3.1. Métodos exactos

Los métodos exactos encuentran la solución óptima mediante exploración sistemática del espacio de búsqueda y son referencia para evaluar aproximaciones más escalables y ofrecen optimalidad bajo supuestos específicos sobre la estructura del grafo y la función de costos.

Algoritmos basados en búsqueda en grafos

El algoritmo de Dijkstra (Dijkstra, 1959) encuentra el camino más corto en grafos con costos no negativos propagando mediante etiquetas de distancia, teniendo asi cada nodo visitado quede con una sola vez con su distancia mínima. La complejidad es $O(|V|^2)$ y con $O(|E| + |V| \log |V|)$ cuando se usan colas de prioridad como los *heaps* de Fibonacci (Fredman and Tarjan, 1987). En mallas tridimensionales, con $|E| \approx k|V|$, esa mejora se disminuye y el método pierde aplicabilidad en tiempo real. A* (Hart et al., 1968) mejora a Dijkstra con una heurística $h(n)$ que estimula el costo restante hacia el objetivo, dirigiendo la búsqueda sin perder optimalidad mientras la heurística sea admisible.

Cuando la ruta tiene que respetar restricciones de recursos como combustible, tiempo máximo o nivel de amenaza, el problema se convierte en un *Resource Constrained Shortest Path Problem* (RCSPP) (Irnich and Desaulniers, 2005). A diferencia del SPSP clásico, dos caminos parciales hacia un mismo nodo pueden ser incomparables si uno gasta menos costo pero más recursos, y por eso hay que mantener múltiples etiquetas no dominadas por nodo. Los algoritmos de etiquetado resuelven el RCSPP con programación dinámica, expandiendo etiquetas que representan pares (costo, recursos) y aplicando reglas de dominancia para no saturar el espacio de estados (Tilk et al., 2017; Himmich et al., 2024). El algoritmo de pulso (*pulse algorithm*) es otra alternativa, una búsqueda en profundidad con poda por cotas duales cuyas extensiones bidireccionales han funcionado en redes de hasta 6 millones de nodos (Lozano and Medaglia, 2013;

Cabrera et al., 2020), y trabajos recientes combinan etiquetado bidireccional con A* para el RCSPP logrando mejoras de órdenes de magnitud (Ahmadi et al., 2025).

Métodos basados en geometría computacional

Estos metodos emplean diagramas de Voronoi (Bortoff, 2000) y grafos de visibilidad (Welzl, 1985), los cuales reducen el espacio continuo a una estructura topológica. Los diagramas de Voronoi maximizan el margen de seguridad con los obstáculos, mientras que los grafos de visibilidad obtienen la trayectoria euclidiana más corta tangente a los vértices de los obstáculos, y en ambos casos construir el grafo es costoso y el método se vuelve sensible ante cambios en su entorno. Trabajos recientes emplean diagramas de Voronoi con teoría de consistencia para planificación multi-UAV en entornos compartidos (Chen et al., 2017).

Tambien, otra aproximación geométrica exacta es la descomposición del espacio libre en celdas convexas, donde el espacio de configuración libre se particiona en celdas no superpuestas mediante extensiones de segmentos de obstáculos hasta bordes de celda (descomposición trapezoidal). La trayectoria se obtiene navegando el grafo de adyacencia entre celdas (LaValle, 2006). El método es completo pero su costo crece con la complejidad geométrica del entorno. Una línea más reciente son los grafos de conjuntos convexos (GCS), que formulan el problema como optimización convexa sobre caminos cuyos nodos son regiones convexas contiguas (Marcucci et al., 2024). GCS admite restricciones dinámicas y de recursos dentro de cada región mediante programación convexa, aunque su escalabilidad a espacios tridimensionales densos sigue abierta (Garg et al., 2025).

Programación matemática

Existe otro enfoque el cual es formular la planificación de rutas como un modelo de Programación Matemática (MP), donde la ruta se obtiene resolviendo un sistema de ecuaciones y desigualdades que codifican la minimización y las restricciones.

El enfoque más directo emplea Programación Lineal Entera (ILP), el cual presenta variables binarias que indican la selección de cada arista y restricciones de conservación de flujo garantizando la conectividad, en esta formulación equivale a un problema de flujo de costo mínimo con demanda unitaria. En grafos se necesitan restricciones adicionales para evitar ciclos (Rosa et al., 2025; Yakıcı, 2016).

La Programación Entera Mixta (MILP) añade variables continuas junto con las binarias, lo que permite modelar restricciones más complejas. En UAVs, por ejemplo, se pueden integrar restricciones de autonomía energética con variables continuas que representan el nivel de batería en cada punto, optimizando la geometría de la ruta y la gestión de recursos al mismo tiempo (Arshad et al., 2025; Kumari and Srirangarajan, 2025; Zhang et al., 2025).

Cuando el problema ILP o MILP es demasiado grande para resolverse directamente, se aplican técnicas de descomposición exacta como *Branch and Bound* (B&B) (Land

and Doig, 1960) el cual relaja la integralidad de las variables para obtener cotas inferiores y ramifica sobre variables fraccionarias; *Branch and Cut* (B&C) (Padberg and Rinaldi, 1991) refuerza las relajaciones con planos de corte; y *Branch and Price* (B&P) (Barnhart et al., 1998) combina B&B con generación de columnas, donde los subproblemas de pricing son instancias de RCSPP resueltas por etiquetado (Irnich and Desaulniers, 2005).

2.3.2. Enfoques heurísticos y metaheurísticos

Los métodos aproximados se concentran en obtener rutas factibles, dejando de lado la optimalidad. Las heurísticas aprovechan lo que se conoce del dominio para podar, ya sea descomponiendo la planificación en niveles jerárquicos o integrando las restricciones cinemáticas (curvas de Dubins) directamente en la función de costo, lo que produce rutas válidas en tiempo real pero con una eficacia el cual depende de reglas fijas y sin garantía de convergencia al óptimo global (Liu et al., 2013; Wang et al., 2014; Primatesta et al., 2019).

Las metaheurísticas por su parte son métodos de optimización estocástica que imitan procesos naturales para explorar paisajes no convexos sin quedar atrapadas en el primer mínimo que aparece. Los Algoritmos Genéticos (GA) (Roberge et al., 2012) aplican mecanismos llamaddos selección y mutación, la inteligencia de enjambre (PSO, ACO) (Liu et al., 2016; Cekmez et al., 2016; Ali et al., 2020) simula interacciones colectivas y rastros de feromona, el Recocido Simulado (SA) (Kirkpatrick et al., 1983; Wu et al., 2023) por su parte acepta soluciones de mayor costo con una probabilidad controlada que depende de la temperatura, lo que le permite saltar fuera de mínimos locales. Son robustos frente a no linealidades, pero requieren calibrar hiperparámetros de forma empírica y no ofrecen cotas sobre la distancia al óptimo.

2.3.3. Enfoques cuánticos

Estos enfoques emplean dos propiedades, la superposición, que permite representar 2^N estados con N qubits, y la tunelización. Esta última es relevante en problemas no convexos porque permite que el sistema atraviese barreras de potencial energético en lugar de escalarlas, algo que para los métodos clásicos representa una limitación importante.

El Recocido Cuántico (QA) se basa en la evolución adiabática, un sistema parte de un estado inicial sencillo y se transforma gradualmente hasta alcanzar el estado fundamental de un Hamiltoniano de Ising que codifica el problema a resolver. Llevado a planificación de rutas, el grafo se traduce a una formulación QUBO en la que vértices y aristas se representan como interacciones entre espines. Experimentos con procesadores D-Wave han mostrado que el QA procesa restricciones complejas y tareas de coordinación multi-agente, y en entornos densos ha llegado a superar al recocido simulado (Farhi et al., 2000a; Krauss and McCollum, 2020; Carbone et al., 2023; Stollenwerk et al., 2019; Mohammed et al., 2025).

El Algoritmo de Optimización Cuántica Aproximada (QAOA) usa un enfoque híbrido variacional en computadoras de compuertas cuanticas. Alternan circuitos cuánticos parametrizados con un optimizador clásico para minimizar la energía esperada de un Hamiltoniano de costo, obteniendo resultados que han indicado que es viable para navegación en grafos dirigidos, con estrategias de inicialización híbrida para acelerar la convergencia en dispositivos NISQ (Innan et al., 2025; Liu, 2025; Monbroussou and Kukla, 2021).

La Tabla 2.1 resume las características principales de los enfoques revisados, clasificados de acuerdo a sus garantías de optimalidad, escalabilidad y aplicabilidad práctica.

Tabla 2.1: Comparación de enfoques de solución para planificación de rutas

Enfoque	**Ventajas**	**Limitaciones**
Exactos (Dijkstra, A*, D*/D* Lite, RCSPP, MILP, B&B, B&C, B&P, Lagrangiana)	Garantía de optimalidad global (x^*). Modelado estricto de restricciones. Replanificación eficiente en entornos dinámicos.	Costo computacional exponencial en el peor caso. No escalable a grafos densos.
Heurísticos (RiskA*, Dubins-SAS, descomposición jerárquica)	Eficiencia para tiempo real. Adaptable a cinemática del UAV.	Soluciones subóptimas sin garantías. Poca generalización ante cambios.
Metaheurísticas (GA, PSO, ACO, SA)	Escape de mínimos locales. Robustez en espacios no convexos.	Convergencia no garantizada. Alta sensibilidad a hiperparámetros.
Enfoques Cuánticos (Quantum Annealing, QAOA, Ising/QUBO)	Tunelización y paralelismo cuántico. Potencial en combinatoria NP-difícil.	Hardware ruidoso y limitado (NISQ). Complejidad de mapeo (*embedding*).

3. Marco Teórico

Este capítulo presenta los fundamentos conceptuales y matemáticos de la metodología. Se introducen principios de optimización combinatoria junto con la formulación QUBO, siendo este el marco donde se representa el problema de planificación de rutas. Posteriormente se revisa el recocido simulado como método clásico de referencia, seguido del recocido cuántico y los principios de computación cuántica, finalmente se describe la representación de grafos tridimensionales y la formulación del problema 3D-SPSP.

3.1. Optimización combinatoria

La optimización combinatoria busca seleccionar la mejor configuración entre un conjunto finito de soluciones que suele crecer de forma exponencial. En estos problemas aparecen en situaciones donde es necesario emplear decisiones discretas como asignación de recursos, planificación de operaciones, diseño de redes y, en este caso, determinación de rutas óptimas.

3.1.1. Problemas de optimización con variables binarias

Un problema de optimización combinatoria se formula como:

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{3.1}$$

donde $\mathcal{X}$ representa el espacio de configuraciones factibles (un conjunto discreto finito) y $f : \mathcal{X} \to \mathbb{R}$ es la función objetivo a minimizar.

Una clase importante de estos problemas emplea variables binarias para codificar decisiones y cada variable $x_i \in \{0, 1\}$ representa una elección entre dos alternativas. Un vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \{0, 1\}^n$ codifica una configuración completa de n decisiones binarias, y el espacio de configuraciones factibles $\mathcal{X} \subseteq \{0, 1\}^n$ contiene todas las configuraciones que satisfacen los requisitos del problema. El tamaño de este espacio crece exponencialmente como 2^n, por lo que la enumeración exhaustiva se vuelve inviable incluso con valores moderados de n.

Analizar estos problemas consiste en interpretar la función objetivo $f(\mathbf{x})$ como un paisaje de energía, donde cada configuración $\mathbf{x}$ es un estado del sistema con energía $f(\mathbf{x})$. El problema de optimización se reduce entonces a encontrar el estado fundamental $\mathbf{x}^*$, la configuración de mínima energía:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{3.2}$$

El paisaje de energía de un problema combinatorio suele tener una estructura compleja con varios mínimos locales: configuraciones cuya energía es menor que la de todas las alcanzables mediante una transición elemental, pero no necesariamente la mínima global. Una configuración $\mathbf{x}_L$ es un mínimo local si $f(\mathbf{x}_L) \leq f(\mathbf{x}')$ para toda $\mathbf{x}'$ en su vecindad inmediata.

La Figura 3.1 ilustra esta estructura, donde el paisaje presenta varios mínimos locales de diferentes profundidades junto con el mínimo global.

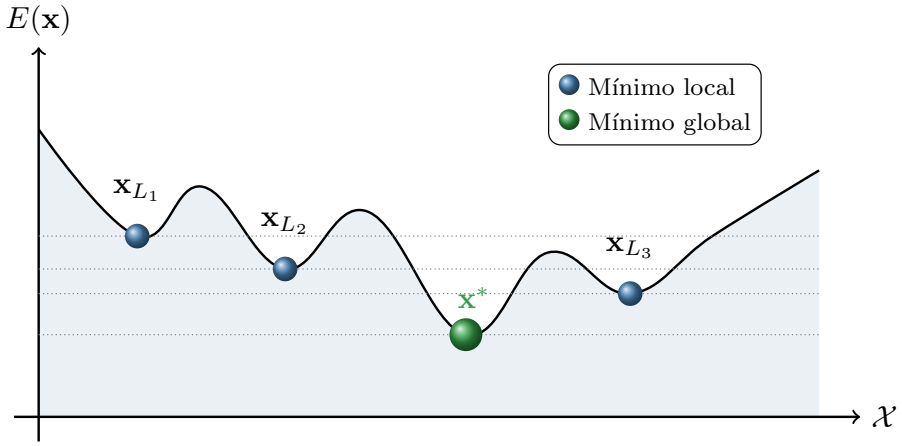


Figura 3.1: Paisaje de energía con múltiples mínimos locales y un mínimo global.

La presencia de numerosos mínimos locales complica estos problemas: un método que solo acepta transiciones hacia menor energía queda atrapado en el primer mínimo local que encuentra, sin garantía de alcanzar el mínimo global. Las barreras energéticas entre mínimos impiden que estrategias puramente descendentes escapen de configuraciones subóptimas.

De ahí surge la necesidad de métodos que superen esas barreras. El recocido simulado lo hace con fluctuaciones térmicas controladas que permiten aceptar, con cierta probabilidad, transiciones hacia estados de mayor energía. El recocido cuántico, en cambio, usa la tunelización para atravesar directamente las barreras.

Ambos enfoques comparten la misma abstracción: una función objetivo sobre configuraciones binarias, representada mediante el modelo Ising o su formulación equivalente QUBO, que se desarrolla en la siguiente sección.

3.1.2. Modelo Ising y formulación QUBO

Resolver problemas de optimización combinatoria requiere una representación matemática que exprese el espacio de configuraciones en términos energéticos. Aquí, el modelo de Ising y la formulación QUBO conectan los sistemas físicos cuánticos con los problemas discretos de optimización.

El modelo Ising se desarrolló para estudiar transiciones de fase magnéticas (McCoy and Wu, 1973; Stinchcombe, 1973) el cual describe sistemas de partículas con espín binario $s_i \in \{-1, +1\}$ y las interacciones entre pares de espines y los campos externos determinan la energía total:

$$H(\mathbf{s}) = -\sum_{i<j} J_{ij} s_i s_j - \sum_i h_i s_i, \tag{3.3}$$

donde J_{ij} cuantifica la fuerza de acoplamiento entre los espines i y j, y h_i representa un campo magnético externo aplicado al espín i. La minimización de $H(\mathbf{s})$ equivale a determinar el estado de menor energía (estado fundamental), analogía directa con los problemas de optimización discreta.

Mediante una transformación $s_i = 2x_i - 1$, con $x_i \in \{0, 1\}$, se obtiene una expresión cuadrática que es equivalente en variables binarias, teniendo asi la formulación QUBO (Quadratic Unconstrained Binary Optimization) (Karp, 2010; Pardalos and Jha, 1992).

Este problema consiste en minimizar una función cuadrática en variables binarias como:

$$f(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} x_i x_j \tag{3.4}$$

donde $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T \in \{0, 1\}^n$ es un vector de n variables binarias de decisión y $Q \in \mathbb{R}^{n \times n}$ una matriz de coeficientes simétrica, es decir, $Q_{ij} = Q_{ji}$.

Las variables binarias cumplen $x_i^2 = x_i$, por tanto la función objetivo puede reescribirse separando terminos lineales y cuadráticos:

$$f(\mathbf{x}) = \mathbf{x}^T Q \mathbf{x} = \sum_{i=1}^{n} Q_{ii} x_i + \sum_{i<j} Q_{ij} x_i x_j \tag{QUBO}$$

En el primer término contiene las contribuciones lineales, cada coeficiente diagonal Q_{ii} representa el costo o beneficio individual de activar la variable x_i. El segundo incluye las interacciones cuadráticas entre pares de variables, donde Q_{ij} (con $i \neq j$) mide el efecto combinado de activar x_i y x_j al mismo tiempo. Esta descomposición permite modelar problemas donde las decisiones individuales tienen costos asociados y hay dependencias entre pares de decisiones.

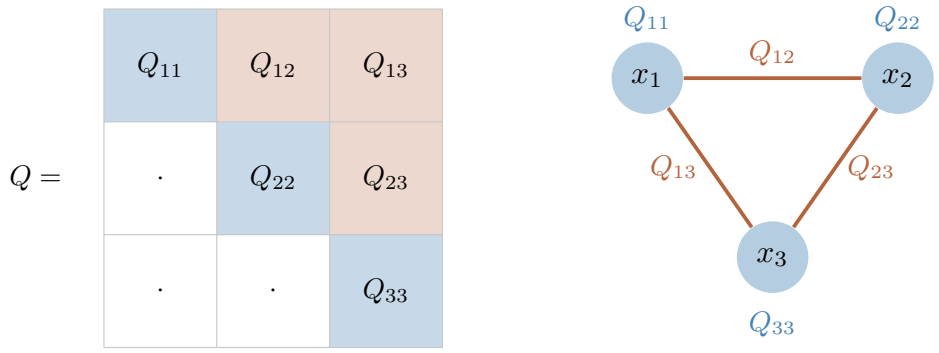
Interpretación de los coeficientes

La matriz Q define el paisaje de energía del problema y puede tratarse como simétrica ($Q_{ij} = Q_{ji}$) y es suficiente con trabajar sobre el triángulo superior, donde cada interacción entre variables se codifica una sola vez con los índices $i < j$.

- **Coeficientes diagonales** Q_{ii}: Muestran el *peso individual* de cada variable. Teniendo $x_i^2 = x_i$, el término $Q_{ii}x_i$ solo contribuye cuando $x_i = 1$: si $Q_{ii} < 0$ la energía baja al activar la variable y conviene incluirla, si $Q_{ii} > 0$ la energía sube y conviene excluirla.

- **Coeficientes fuera de la diagonal** Q_{ij} (con $i \neq j$): Codifican la *interacción* entre pares de variables a través del producto $x_i x_j$, que vale 1 solo cuando ambas están activas a la vez. Un $Q_{ij} < 0$ favorece activarlas juntas y un $Q_{ij} > 0$ lo desalienta, lo que permite traducir restricciones del problema en penalizaciones sobre la energía.

En la Figura 3.2 se muestra esa correspondencia entre matriz y grafo ponderado, los nodos son las variables binarias y las aristas las interacciones cuadráticas. Los elementos diagonales (Figura 3.2a) se traducen en pesos sobre los nodos y los de fuera de la diagonal en pesos sobre las aristas (Figura 3.2b).



(a) Matriz Q con elementos diagonales (azul) y elementos fuera de la diagonal (naranja).

(b) Representación como grafo ponderado de variables binarias.

Figura 3.2: Estructura de la matriz QUBO y su interpretación como grafo ponderado.

Incorporación de restricciones mediante penalizaciones

La formulación QUBO no representa restricciones explícitas, pero cuando el problema sí las tiene, estas se incorporan directamente en la función objetivo mediante términos de penalización (Lucas, 2014; Glover et al., 2019a), lo que convierte el problema restringido en uno sin restricciones.

Para $f_0(\mathbf{x})$ la función base sobre $\mathbf{x} \in \{0,1\}^n$ y $g(\mathbf{x}) = 0$ una restricción de igualdad con g lineal en las variables binarias. Esta restricción se convierte en el término cuadrático

$$P(\mathbf{x}) = g(\mathbf{x})^2, \tag{3.5}$$

y la función objetivo penalizada queda como

$$f(\mathbf{x}) = f_0(\mathbf{x}) + \lambda\, P(\mathbf{x}), \tag{3.6}$$

con $\lambda > 0$ como factor de penalización.

Para el término $P(\mathbf{x})$ no es negativo y se anula justo cuando la restricción se cumple. Tienendo un λ suficientemente grande, cualquier violación produce un costo mayor que el de toda configuración factible, y el mínimo global incluye entonces una solución válida del problema.

Múltiples restricciones. Si el problema incluye varias restricciones, cada una se incorpora mediante un término de penalización independiente, teniendo asi un conjunto

de m restricciones de igualdad

$$g_k(\mathbf{x}) = 0, \qquad k = 1, \ldots, m,$$

las cuales genera un término cuadrático $P_k(\mathbf{x}) = g_k(\mathbf{x})^2$ y la función objetivo se crea sumando la función base y la combinación lineal de todas las penalizaciones:

$$f(\mathbf{x}) = f_0(\mathbf{x}) + \sum_{k=1}^{m} \lambda_k \, P_k(\mathbf{x}), \tag{3.7}$$

donde $\lambda_k > 0$ son factores de penalización, que pueden ser iguales o diferenciarse según la importancia de cada restricción.

La expansión algebraica de cada $P_k(\mathbf{x})$ genera términos lineales (en las variables x_i) y cuadráticos (en los productos $x_i x_j$), los cuales se suman a los coeficientes correspondientes de la matriz Q. En particular, si $g_k(\mathbf{x})$ es una función lineal de la forma

$$g_k(\mathbf{x}) = a_{k0} + \sum_{i=1}^{n} a_{ki} x_i,$$

entonces

$$P_k(\mathbf{x}) = \left(a_{k0} + \sum_i a_{ki} x_i \right)^2 = a_{k0}^2 + 2a_{k0} \sum_i a_{ki} x_i + \sum_{i,j} a_{ki} a_{kj} x_i x_j.$$

El término constante a_{k0}^2 no afecta la minimización, los términos lineales contribuyen a los coeficientes diagonales Q_{ii}, y los términos cuadráticos contribuyen a los coeficientes fuera de la diagonal Q_{ij} (con $i \neq j$). De este modo, todas las restricciones se integran de manera homogénea en la misma matriz Q.

Con esto, un problema con múltiples restricciones explícitas se vuelve uno de optimización cuadrática binaria sin restricciones, manteniendo la estructura del modelo QUBO y siendo procesable directamente por los algoritmos de recocido.

Evaluación de configuraciones

Una configuración del sistema está determinada por el vector binario $\mathbf{x} \in \{0,1\}^n$, donde cada componente x_i representa una variable de decisión del modelo QUBO. El estado de cada variable admite una clasificación formal según su activación.

Definición 3.1 (Variable activa e inactiva)
Sea $\mathbf{x} \in \{0,1\}^n$ una configuración del sistema. Una variable x_i se denomina:

- **Activa** si $x_i = 1$, indicando que el elemento asociado a dicha variable está incluido en la solución.
- **Inactiva** si $x_i = 0$, indicando que el elemento asociado está excluido.

Esto permite caracterizar cada configuración mediante el subconjunto de variables que conforman la solución.

Definición 3.2 (Conjunto de variables activas)
Dada una configuración $\mathbf{x} \in \{0,1\}^n$, el *conjunto de variables activas* se define como

$$S(\mathbf{x}) = \{i \in \{1,\dots,n\} : x_i = 1\}.$$

Ya que cada término de la función objetivo involucra al menos una variable como factor, las contribuciones de las variables inactivas ($x_i = 0$) no generan contribucion. El valor de la función objetivo depende entonces del conjunto de variables activas y se puede expresar como

$$f(\mathbf{x}) = \sum_{i \in S(\mathbf{x})} Q_{ii} + \sum_{\substack{i,j \in S(\mathbf{x}) \\ i<j}} Q_{ij}, \tag{3.8}$$

el primer término muestra los coeficientes diagonales de las variables seleccionadas y el segundo las interacciones entre cada par de variables activas a la vez.

Cuando se tienen varias restricciones mediante penalizaciones (Ecuación 3.7), los coeficientes Q_{ii} y Q_{ij} codifican dos tipos de contribuciones, las que corresponden a la función de costo original $f_0(\mathbf{x})$ y las obtenidas por las penalizaciones $\lambda_k P_k(\mathbf{x})$. Dado que ambas contribuciones quedan en la misma matriz Q, la evaluación mediante la Ecuación 3.8 no diferencia entre costo-penalización y el efecto conjunto genera una separación natural en el paisaje de energía, las configuraciones factibles, sin penalizaciones que ocupan las regiones de menor energía, mientras que las que violan alguna restricción acumulan incrementos proporcionales a la magnitud de la violación.

El espacio de búsqueda incorpora 2^n configuraciones posibles, las cuales corresponden a todos los subconjuntos $S \subseteq \{1,\dots,n\}$. Valores moderados de n, se tiene un crecimiento exponencial volviendo asi la enumeración exhaustiva intratable, lo que motiva el uso de métodos heurísticos como el recocido simulado y el recocido cuántico para aproximar el mínimo global de la función objetivo.

3.2. Recocido simulado

El recocido simulado (SA) es una metaheurística el cual esta inspirada en el recocido de metales en metalurgia, un material se calienta y luego se enfría lentamente para eliminar defectos y alcanzar una estructura cristalina de baja energía. La analogía con la optimización combinatoria se da por la función objetivo $f(\mathbf{x})$ representando la energía del sistema, las configuraciones $\mathbf{x} \in \{0,1\}^n$ son estados del paisaje energético, y el objetivo es llegar al estado fundamental $\mathbf{x}^*$ evadiendo los múltiples mínimos locales (Kirkpatrick et al., 1983).

Como se vio en la Figura 3.1, el paisaje de un problema QUBO tiene numerosos mínimos locales $\mathbf{x}_{L_k}$. Un algoritmo unicamente descendente, que solo acepta transi-

ciones hacia menor energía, se puede quedar atrapado en el primer mínimo local. SA propone reoslver esto con un mecanismo probabilístico que acepta, de forma controlada, transiciones hacia mayor energía, logrando escapar de mínimos locales y explorar más configuraciones del espacio $\mathcal{X}$.

El procedimiento se observa mediante cuatro componentes fundamentales que se desarrollan iterativamente, como ilustra la Figura 3.3.

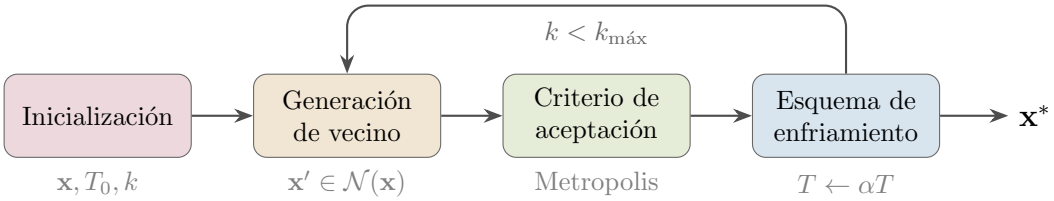


Figura 3.3: Componentes principales del algoritmo de recocido simulado.

1. **Inicialización**

   El algoritmo necesita una configuración inicial $\mathbf{x}$ que sirve como punto de partida para explorar el espacio de configuraciones. Esta configuración puede generarse al azar, asignando a cada variable x_i el valor 0 o 1 con igual probabilidad, o construirse mediante alguna heurística del problema. La configuración inicial se guarda también como la mejor conocida $\mathbf{x}^*$, que se actualiza cada vez que se visita una configuración de menor energía.

   El parámetro de temperatura inicial T_0 debe ser lo bastante alto para permitir una exploración amplia del paisaje energético en las primeras iteraciones. Un criterio práctico es elegir T_0 de modo que la probabilidad de aceptar transiciones con incrementos de energía Δf del orden de magnitud típico del problema sea cercana a 1, con lo que el sistema atraviesa las barreras energéticas durante la exploración inicial.

   Como ejemplo, para un problema con $n = 5$ variables, una configuración inicial generada aleatoriamente y su conjunto de variables activas $S(\mathbf{x})$ podría ser:

   $$\mathbf{x} = (1,\ 0,\ 1,\ 0,\ 1), \qquad S(\mathbf{x}) = \{1,\ 3,\ 5\}$$

2. **Generación de vecinos**

   Desde la configuración actual $\mathbf{x}$, se genera una candidata $\mathbf{x}'$ que pertenece a su vecindad. En problemas con variables binarias la vecindad estándar es el cambio de bit único, $\mathbf{x}'$ se obtiene seleccionando una variable x_i uniformemente al azar e invirtiendo su estado. El conjunto de vecinos de $\mathbf{x}$ se define formalmente como:

   $$\mathcal{N}(\mathbf{x}) = \left\{ \mathbf{x}' \in \{0,1\}^n : \sum_{i=1}^{n} |x'_i - x_i| = 1 \right\} \tag{3.9}$$

   Cada configuración tiene n vecinos. En términos del conjunto de variables activas $S(\mathbf{x})$, invertir una variable x_i corresponde a:

- **Activación**: Si $x_i = 0$, entonces $S(\mathbf{x}') = S(\mathbf{x}) \cup \{i\}$.
- **Desactivación**: Si $x_i = 1$, entonces $S(\mathbf{x}') = S(\mathbf{x}) \setminus \{i\}$.

Como $\mathbf{x}'$ difiere de $\mathbf{x}$ solo en la variable x_i, la diferencia de energía $\Delta f = f(\mathbf{x}') - f(\mathbf{x})$ se obtiene considerando solo los términos de la función QUBO asociados a la variable invertida:

La diferencia de energía $\Delta f = f(\mathbf{x}') - f(\mathbf{x})$ se evalúa directamente como:

$$\Delta f = \mathbf{x}'^T Q\, \mathbf{x}' - \mathbf{x}^T Q\, \mathbf{x} \tag{3.10}$$

Retomando la configuración $\mathbf{x} = (1,\ 0,\ 1,\ 0,\ 1)$ con $S(\mathbf{x}) = \{1, 3, 5\}$, si se selecciona $i = 2$ (activación, $x_2 : 0 \to 1$):

$$\mathbf{x}' = (1,\ 1,\ 1,\ 0,\ 1), \qquad S(\mathbf{x}') = \{1,\ 2,\ 3,\ 5\}$$

Si en cambio se selecciona $i = 3$ (desactivación, $x_3 : 1 \to 0$):

$$\mathbf{x}' = (1,\ 0,\ 0,\ 0,\ 1), \qquad S(\mathbf{x}') = \{1, 5\}$$

3. **Criterio de aceptación**

El paso de una configuracion $\mathbf{x}$ a $\mathbf{x}'$ se decide siguiendo el criterio de Metropolis (Metropolis et al., 1953), inspirado en la mecánica estadística. Con la diferencia de energía Δf, la probabilidad de aceptación se define como:

$$P_{\text{accept}}(\mathbf{x} \to \mathbf{x}') = \begin{cases} 1 & \text{si } \Delta f < 0 \\ \exp\left(-\dfrac{\Delta f}{T}\right) & \text{si } \Delta f \geq 0 \end{cases} \tag{3.11}$$

Las transiciones que reducen la energía ($\Delta f < 0$) se aceptan siempre, por el contrario ($\Delta f \geq 0$) se aceptan con probabilidad que decrece al aumentar Δf o al disminuir T. En la práctica, la decisión se implementa generando un número aleatorio $u \sim \text{Uniforme}(0, 1)$ y aceptando la transición si $u < \exp(-\Delta f/T)$.

La Figura 3.4 muestra cómo cambia esta probabilidad para distintos valores de temperatura. Ante un mismo incremento de energía $\Delta f_0 > 0$, las curvas determinan una probabilidad de aceptación distinta ($p_1 > p_2 > p_3$): a mayor temperatura, mayor es la probabilidad de que u caiga por debajo del umbral y la transición sea aceptada.

Este mecanismo la lugar a dos regímenes de caracteristicos diferentes, controlados por la temperatura T:

- **Alta temperatura**: El factor de Boltzmann $\exp(-\Delta f/T) \approx 1$ para incrementos moderados, lo que favorece una exploración amplia del espacio de soluciones y permite escapar de mínimos locales.

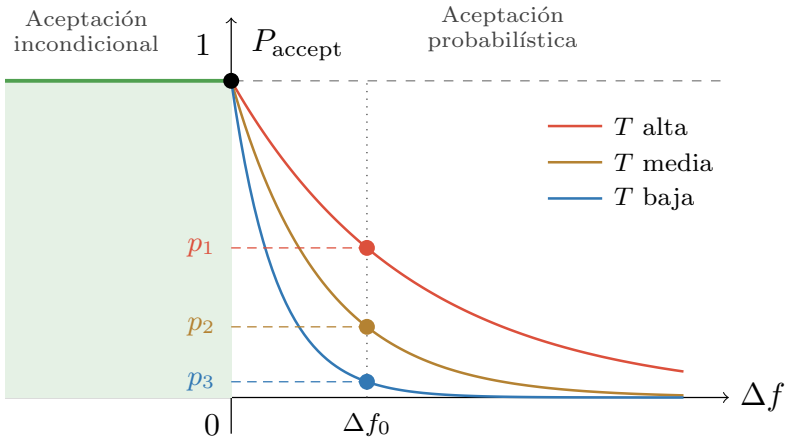


Figura 3.4: Probabilidad de aceptación del *criterio de Metropolis* para distintas temperaturas ante un mismo incremento de energía Δf_0.

- **Baja temperatura**: El factor $\exp(-\Delta f/T) \approx 0$ para $\Delta f > 0$, o que restringe la aceptación de transiciones desfavorables y concentra la búsqueda en regiones de baja energía.

Por ultimo, si la configuración aceptada mejora $f(\mathbf{x}') < f(\mathbf{x}^*)$, se actualiza la mejor configuración conocida $\mathbf{x}^* \leftarrow \mathbf{x}'$.

4. **Esquema de enfriamiento**

   La temperatura T se disminuye gradualmente con las iteraciones, teniendo control en la transición entre exploración y explotación. El esquema más usado es el enfriamiento geométrico:

$$T_{k+1} = \alpha \cdot T_k \tag{3.12}$$

   donde $\alpha \in (0, 1)$ es la tasa de enfriamiento, típicamente entre $[0{,}80, 0{,}99]$.

   El esquema de enfriamiento usa dos parámetros:

   - **Temperatura inicial** T_0: Siendo lo bastante alta para permitir exploración inicial.
   - **Tasa de enfriamiento** α: Con valores cercanos a 1 dan enfriamiento lento (mayor costo computacional, mayor calidad) mientras que valores menores aceleran la convergencia.

   El procedimiento termina al alcanzar el número máximo de iteraciones $k_{\text{máx}}$ y devuelve la configuración $\mathbf{x}^*$ como aproximación al estado inicial.

3.3. Recocido cuántico

El recocido cuántico por su parte aborda la optimización combinatoria desde la mecánica cuántica y para entender su aplicación al modelo QUBO, se establece el marco teórico de la computación cuántica.

Fundamentos de computación cuántica

Feynman (1982) planteó que simular sistemas cuánticos requiere de computadoras basadas en mecánica cuántica, esta idea se fundamento con algoritmos como el de Shor para factorización (Shor, 1994) y el de Grover para búsqueda no estructurada (Grover, 1996). La diferencia con la computación clásica está en propiedades como la superposición, el entrelazamiento, la interferencia cuántica y la tunelización cuántica.

En este enfoque, la información se representa mediante qubits, las unidades fundamentales de información cuántica. Por convención estos sistemas se describen con la notación de Dirac (1981), llamada bra-ket:

$$bra \quad \langle\psi| \quad ket \quad |\psi\rangle\,,$$

donde $|\psi\rangle$ es un vector columna y $\langle\psi|$ es su transpuesto complejo conjugado.

Los estados base de un qubit son:

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}.$$

A diferencia de un bit clásico, un qubit puede estar en superposición, expresado como combinación lineal de los estados base:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

donde $\alpha, \beta \in \mathbb{C}$ son las amplitudes de probabilidad. Al medir, la probabilidad de obtener $|0\rangle$ es $|\alpha|^2$ y la de $|1\rangle$ es $|\beta|^2$, con normalización $|\alpha|^2 + |\beta|^2 = 1$. Esta propiedad permite que un registro de n qubits represente una combinación lineal de los 2^n estados base posibles (Rieffel and Polak, 2000), codificando todas las configuraciones en un único estado cuántico.

Las transformaciones sobre estados de qubits se describen con operadores unitarios. Los operadores de Pauli forman un conjunto fundamental de operadores de un qubit, definidos en la base computacional $\{|0\rangle, |1\rangle\}$ como

$$\sigma^x = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}, \tag{3.13a}$$

$$\sigma^y = \begin{pmatrix}0 & -i\\i & 0\end{pmatrix}, \tag{3.13b}$$

$$\sigma^z = \begin{pmatrix}1 & 0\\0 & -1\end{pmatrix}. \tag{3.13c}$$

El operador σ^x cambia los estados base ($|0\rangle \leftrightarrow |1\rangle$), mientras que σ^z los diferencia asignando autovalores $+1$ y -1 a $|0\rangle$ y $|1\rangle$. Al aplicarse al i-ésimo qubit de un registro de n, se denotan σ_i^x y σ_i^z. Con estos operadores se construyen los Hamiltonianos del recocido cuántico.

La interferencia cuántica permite controlar las amplitudes, la constructiva amplifica la probabilidad de medir estados deseados y la destructiva suprime los indeseados (Nielsen and Chuang, 2010). El entrelazamiento cuántico genera correlaciones entre qubits que no se describen como producto de estados individuales, lo que permite explorar correlaciones globales en el espacio de configuraciones (Einstein et al., 1935; Bell, 1964). La tunelización cuántica atraviesa barreras de potencial sin excitación térmica, un mecanismo útil para escapar de mínimos locales en optimización.

Modelos de cómputo cuántico

Estos principios pueden aprovecharse computacionalmente de distintas formas. Los dos modelos principales son:

- El modelo de Compuertas Cuánticas (Quantum Gate) implementa algoritmos mediante secuencias discretas de compuertas unitarias que manipulan estados de qubits (Nielsen and Chuang, 2010). Funciona bien con algoritmos estructurados como el de Shor (Shor, 1994, 1999) o Grover (Grover, 1996), donde la interferencia se diseña explícitamente.

- La Computación Cuántica Adiabática (Adiabatic Quantum Computing, AQC) reemplaza las operaciones discretas por un proceso continuo de evolución temporal controlada (Farhi et al., 2000b). En vez de aplicar compuertas, el sistema evoluciona bajo un Hamiltoniano que se modifica gradualmente, usando tunelización cuántica para explorar el espacio de configuraciones e interferencia para favorecer estados de baja energía.

Ambos modelos son computacionalmente equivalentes de acuerdo con Aharonov et al. (2008). En AQC resulta natural para optimización combinatoria ya que la función objetivo se codifica como la energía de un sistema físico y la búsqueda del mínimo equivale a determinar el estado de mínima energía. Esta relacion entre problema y dinámica física fundamenta el recocido cuántico que se desarrolla a continuación.

Computación cuántica adiabática

La Computación Cuántica Adiabática (AQC) se basa en el Teorema Adiabático (Born and Fock, 1928), el cual describe que si un sistema arranca en el estado fundamental de un Hamiltoniano y los cambios se hacen lo bastante despacio, el sistema se mantiene en ese estado fundamental a lo largo del tiempo.

Con esto, un problema de optimización se reduce a buscar el estado de mínima energía en un sistema cuántico. Si se tiene una función $f : \{0,1\}^n \to \mathbb{R}$ definida sobre

n variables binarias $\mathbf{x} = (x_1, \ldots, x_n)$, lo que se quiere es hallar la configuración $\mathbf{x}^*$ que entrega el valor más bajo.

Un algoritmo AQC trabaja sobre n qubits en un estado $|\psi(t)\rangle$ y se contruye a partir de tres elementos:

a. **Hamiltoniano inicial** (H_I): describe un sistema cuyo estado fundamental se conoce y es fácil de preparar. Por lo general es una superposición pareja de todas las configuraciones posibles.

b. **Hamiltoniano del problema** (H_P): codifica la función objetivo $f(\mathbf{x})$ de modo que su estado fundamental $|\mathbf{x}^*\rangle$ da la solución óptima, el autoestado con el autovalor más bajo corresponde a $\mathbf{x}^* = \arg\min_{\mathbf{x}} f(\mathbf{x})$.

c. **Camino de evolución adiabática**: una función $s(t)$ que crece de forma monótona de 0 a 1 según el tiempo va de $t = 0$ a $t = T_a$, donde T_a es lo que dura la evolución.

El Hamiltoniano dependiente del tiempo que gobierna la dinámica del sistema se construye como:

$$H(t) = [1 - s(t)]\, H_I + s(t)\, H_P \tag{3.14}$$

En $t = 0$ con $s(0) = 0$, el sistema está en el estado fundamental de H_I. Conforme $t \to T_a$ y $s(t) \to 1$, el Hamiltoniano del problema domina y, si se cumple la adiabaticidad, el sistema converge al estado fundamental de H_P, que codifica la solución $\mathbf{x}^*$.

3.3.1. Proceso de recocido cuántico

El Recocido Cuántico (Quantum Annealing, QA) es la implementación práctica de AQC para resolver problemas de optimización combinatoria (Apolloni et al., 1988; Kadowaki and Nishimori, 1998). Retomando la Ecuación (3.14), QA implementa físicamente la interpolación entre Hamiltonianos mediante:

$$H(s) = A(s)\, H_I + B(s)\, H_P \tag{3.15}$$

donde H_I es el Hamiltoniano inicial y las funciones $A(s)$, $B(s)$ generalizan el camino lineal $1 - s(t)$, $s(t)$ de la Ecuación (3.14), lo que da más flexibilidad en el control del proceso. AQC y QA se diferencian en las condiciones de operación, AQC supone un escenario ideal, una evolución perfectamente adiabática con $T_a \to \infty$, sistema aislado del entorno y control exacto de los parámetros, con lo que la convergencia al estado fundamental es determinista, mientras que QA trabaja con las limitaciones reales: el tiempo de recocido es finito, la temperatura no es cero, hay decoherencia por ruido ambiental y los parámetros de control fluctúan. El resultado no es un único estado óptimo garantizado, sino una distribución de probabilidad sobre las configuraciones:

$$P(\mathbf{x}) = |\, \langle \mathbf{x} | \psi(T_a) \rangle \,|^2 \tag{3.16}$$

Esta distribución favorece las configuraciones de baja energía, pero no basta una sola ejecución para dar con $\mathbf{x}^*$. Por eso QA recurre a un protocolo de muestreo múltiple, donde se lanzan varias corridas independientes y se acumula evidencia sobre las soluciones más prometedoras.

El proceso de recocido cuántico consta de cuatro componentes que se repiten a lo largo de N_{lect} lecturas independientes, tal como muestra la Figura 3.5.

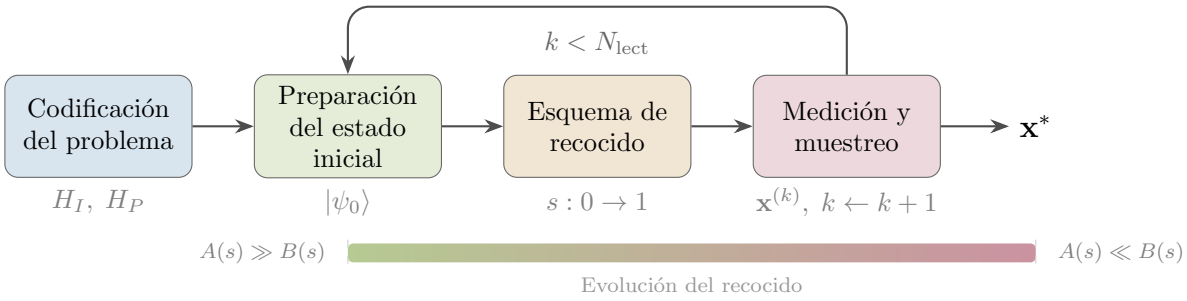


Figura 3.5: Componentes principales del proceso de recocido cuántico.

1. **Codificación del problema**

   Se construyen los Hamiltonianos que definen la dinámica del sistema. El Hamiltoniano inicial H_I introduce las fluctuaciones cuánticas para explorar el espacio de configuraciones mediante el operador σ_i^x (Ecuación 3.13a) en cada qubit:

   $$H_I = -\sum_{i=1}^{n} \sigma_i^x \tag{3.17}$$

   σ_i^x induce transiciones entre $|0\rangle$ y $|1\rangle$ del i-ésimo qubit, generando la superposición necesaria para explorar el espacio de configuraciones.

   El Hamiltoniano del problema H_P codifica la función objetivo QUBO mediante operadores σ_i^z (Ecuación 3.13c), que actúan diagonalmente en la base computacional:

   $$H_P = \sum_{i=1}^{n} h_i \sigma_i^z + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z \tag{3.18}$$

   donde los coeficientes h_i y J_{ij} se derivan de los elementos de la matriz Q mediante la transformación $s_i = 2x_i - 1$ de la Sección 3.1.2. El estado fundamental de H_P es la configuración $\mathbf{x}^*$ que minimiza la función objetivo del problema de optimización.

2. **Preparación del estado inicial cuántico**

El sistema de n qubits se prepara en el estado fundamental del Hamiltoniano inicial H_I, que corresponde a la superposición uniforme de todas las configuraciones posibles:

$$|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle \tag{3.19}$$

Este estado explora sin sesgo: cada una de las 2^n configuraciones tiene igual amplitud de probabilidad. A diferencia de SA, donde la configuración inicial es un vector binario específico $\mathbf{x} \in \{0,1\}^n$, el estado cuántico codifica todas las configuraciones posibles mediante superposición, lo que da un punto de partida global.

3. **Esquema de recocido**

Las funciones $A(s)$ y $B(s)$ controlan cuánto contribuye cada Hamiltoniano durante la evolución. El parámetro $s = t/T_a$ es el tiempo normalizado, con $t \in [0, T_a]$ como tiempo físico y T_a como tiempo total de recocido. Las condiciones de frontera son:

$$\begin{aligned} A(0) \gg B(0) &\implies H(0) \approx H_I \\ A(1) \ll B(1) &\implies H(1) \approx H_P \end{aligned} \tag{3.20}$$

El esquema más común es el lineal, donde $A(s) = 1 - s$ y $B(s) = s$, resultando en:

$$H(s) = (1 - s)\, H_I + s\, H_P \tag{3.21}$$

con lo que se recupera la Ecuación (3.14).

La variacion de s, genera que el espectro de autovalores de $H(s)$ evoluciona continuamente. Los dos niveles de menor energía, el estado fundamental $E_0(s)$ y el primer excitado $E_1(s)$, definen la diferencia $\Delta(s) = E_1(s) - E_0(s)$. Para que el recocido llegue a la solución óptima, el sistema debe permanecer en $E_0(s)$ durante toda la evolución. El punto crítico es donde la diferencia alcanza su mínimo $\Delta_{\text{mín}} = \text{mín}_s\, \Delta(s)$, ahí el sistema es más propenso a transiciones diabáticas hacia $E_1(s)$, lo que afecta la calidad de la solución. Un tiempo de recocido T_a largo reduce la probabilidad de estas excitaciones, aunque en la práctica T_a siempre es finito.

La Figura 3.6 muestra esta dinámica. En $s = 0$ dominan las fluctuaciones cuánticas y el sistema está en el estado fundamental de H_I, explorando el espacio de configuraciones. Conforme s crece, H_P gana influencia y los niveles $E_0(s)$ y $E_1(s)$ se acercan hasta el gap mínimo $\Delta_{\text{mín}}$. Al llegar a $s = 1$, las fluctuaciones cuánticas se suprimen y, si el sistema se mantuvo en $E_0(s)$, converge al estado fundamental de H_P que codifica $\mathbf{x}^*$.

Este esquema es análogo al enfriamiento de SA: mientras SA reduce la temperatura T para pasar de exploración a explotación, QA reduce las fluctuaciones cuánticas mediante $A(s)$, con una transición controlada por la mecánica cuántica en lugar de la termodinámica.

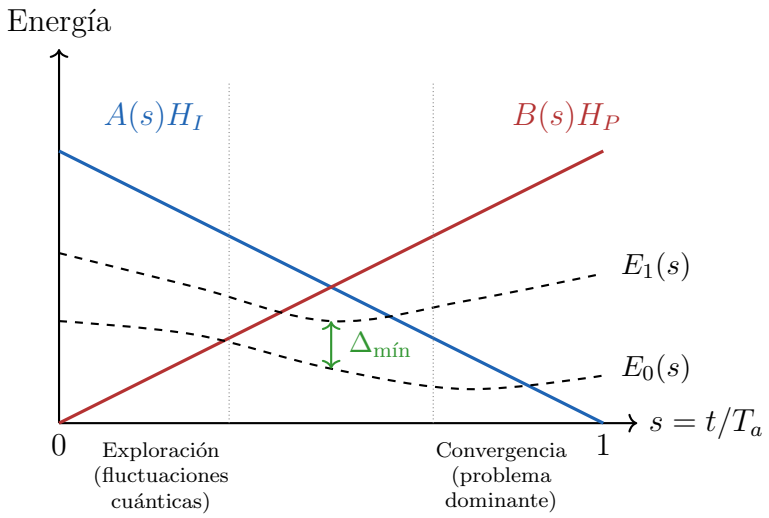


Figura 3.6: Evolución del Hamiltoniano durante el proceso de recocido cuántico.

La diferencia cualitativa está en el mecanismo de escape de mínimos locales, ilustrado en la Figura 3.7. En SA, según el criterio de Metropolis (Ecuación 3.11), la probabilidad de aceptar una transición desfavorable decrece exponencialmente con la altura Δf de la barrera, sin importar su ancho. En QA, las fluctuaciones cuánticas de H_I permiten atravesar barreras mediante tunelización cuántica, cuya probabilidad depende tanto de Δf como del ancho d de la barrera. Las probabilidades de cada mecanismo son

$$P_{\text{SA}} \propto \exp\left(-\frac{\Delta f}{T}\right), \tag{3.22a}$$

$$P_{\text{QA}} \propto \exp\left(-d\sqrt{\Delta f}\right). \tag{3.22b}$$

En SA la trayectoria va sobre la barrera y la probabilidad de escape depende de Δf y T. En QA la trayectoria va a través de la barrera y la probabilidad decrece con el producto $d\sqrt{\Delta f}$.

En barreras estrechas pero altas, algo frecuente en problemas de optimización combinatoria, el mecanismo cuántico puede dar probabilidades de transición mayores que la activación térmica.

4. **Medición y muestreo**

Al terminar la evolución ($s = 1$) se mide el sistema de qubits y el estado $|\psi(T)\rangle$ colapsa a una configuración clásica $\mathbf{x}^{(k)} \in \{0, 1\}^n$. Por temperatura finita, decoherencia y tiempo limitado, una ejecución no garantiza dar con la configuración óptima $\mathbf{x}^*$.

El protocolo estándar de optimización consiste en realizar N_{lect} ejecuciones independientes del proceso de recocido, obteniendo un conjunto de configuraciones candida-

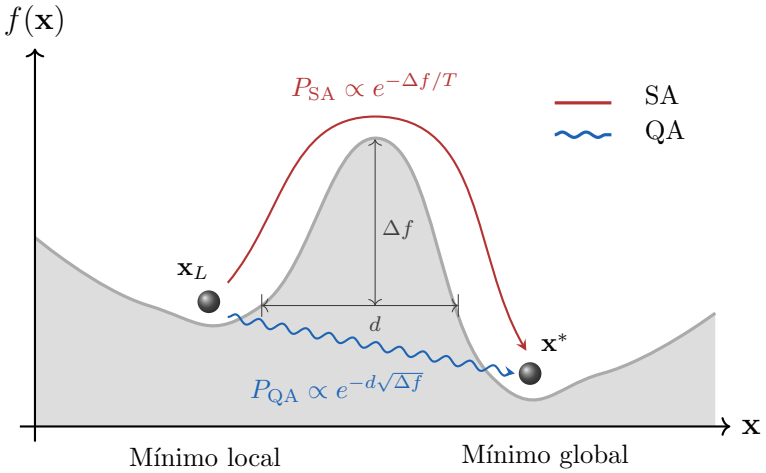


Figura 3.7: Comparación entre los mecanismos de escape de mínimos locales: activación térmica en el recocido simulado (SA) y tunelización cuántica en el recocido cuántico (QA).

tas:

$$\mathcal{R} = \left\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N_{\text{lect}})}\right\}$$

La solución reportada corresponde a la configuración de menor energía observada:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}^{(k)} \in \mathcal{R}} f(\mathbf{x}^{(k)})$$

3.3.2. Recocido cuántico simulado

El Recocido Cuántico Simulado (SQA) es una implementación computacional del recocido cuántico para entornos sin hardware cuántico. En vez de implementar físicamente la evolución de $H(s)$, SQA usa una correspondencia matemática entre el sistema cuántico de n qubits y un sistema clásico extendido que replica sus efectos mediante Monte Carlo cuántico basado en integral de camino (Santoro et al., 2002; Martoňák et al., 2002).

El procedimiento opera mediante cuatro componentes fundamentales, como ilustra la Figura 3.8.

1. **Codificación del sistema extendido**

   El principio de SQA es la descomposición de Suzuki-Trotter (Suzuki, 1976; Trotter, 1959), el cual establece una equivalencia entre la función de partición cuántica del sistema de n qubits bajo $H(s)$ y la de un sistema clásico bidimensional con P réplicas discretas:

$$Z = \text{Tr}\left[e^{-\beta H}\right] \approx \text{Tr}\left[\left(e^{-\frac{\beta}{P}H_P}\, e^{-\frac{\beta}{P}H_I}\right)^P\right] \tag{3.23}$$

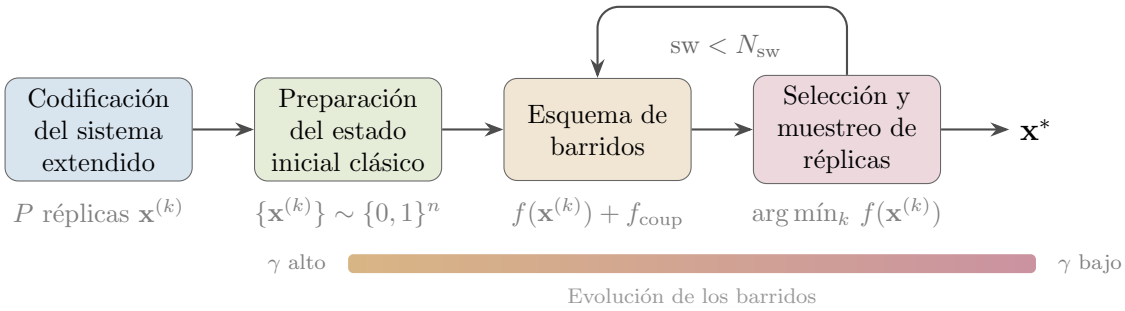


Figura 3.8: Componentes del algoritmo de recocido cuántico simulado.

donde $\beta = 1/T$ es el inverso de la temperatura y el error decrece con $P \to \infty$.

En SQA el problema no se codifica con Hamiltonianos como H_I y H_P como en QA, sino con P réplicas clásicas $\mathbf{x}^{(k)} \in \{0,1\}^n$ $(k = 1, \ldots, P)$ conectadas en anillo con condiciones periódicas de frontera.

2. **Estado inicial clásico**

   Para cada réplica $\mathbf{x}^{(k)}$ se inicializa con una configuración aleatoria en $\{0,1\}^n$. En QA el estado inicial $|\psi_0\rangle$ muestra todas las configuraciones por superposición; aquí cada réplica es una configuración clásica distinta y el conjunto $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(P)}\}$ cumple la misma función de emplear varias zonas del espacio de búsqueda desde el comienzo.

3. **Esquema de barridos**

   Las réplicas evolucionan a lo largo de N_{sw} barridos donde para cada barrido y para cada réplica k se elige una variable $x_i^{(k)}$ al azar y se propone una inversion $(x_i^{(k)} \to 1 - x_i^{(k)})$. La decisión de aceptar o rechazar dependera del efecto sobre dos componentes de la función objetivo extendida (Martoňák et al., 2002).

   El primero es la función objetivo clásica de la réplica k:

$$f\left(\mathbf{x}^{(k)}\right) = {\mathbf{x}^{(k)}}^T Q\, \mathbf{x}^{(k)} \tag{3.24}$$

   El segundo es el término de acoplamiento entre réplicas adyacentes, que aproxima el efecto del campo transversal H_I:

$$f_{\text{coup}}\left(\{\mathbf{x}^{(k)}\}\right) = -\gamma \sum_{k=1}^{P} \sum_{i=1}^{n} \mathbf{1}\left[x_i^{(k)} = x_i^{(k\oplus 1)}\right] \tag{3.25}$$

   donde $k \oplus 1$ denota el vecino adyacente con condiciones periódicas, de modo que $k = P$ se conecta con $k = 1$. El indicador $\mathbf{1}[\cdot]$ vale 1 cuando $x_i^{(k)}$ coincide con la variable correspondiente en la réplica vecina, lo que reduce f_{coup} y favorece que las réplicas adyacentes se mantengan coherentes entre sí, un efecto análogo a la tunelización cuántica de QA. El parámetro $\gamma > 0$ fija la fuerza de este acoplamiento.

En conjunto, forman la función objetivo extendida del sistema:

$$f_{\text{ext}}\left(\{\mathbf{x}^{(k)}\}\right) = \sum_{k=1}^{P} f\left(\mathbf{x}^{(k)}\right) + f_{\text{coup}}\left(\{\mathbf{x}^{(k)}\}\right) \tag{3.26}$$

Como la inversión solo afecta a $x_i^{(k)}$, el cambio total en f_{ext} se separa en dos partes:

$$\Delta f_{\text{ext}} = \Delta f^{(k)} + \Delta f_{\text{coup}}^{(k)} \tag{3.27}$$

El primer término, $\Delta f^{(k)} = f(\mathbf{x}') - f(\mathbf{x}^{(k)})$, mide el cambio en la función clásica de la réplica y es análogo al Δf del recocido simulado (Ecuación 3.10). El segundo captura cómo cambia la coherencia con las réplicas vecinas $k-1$ y $k+1$:

$$\Delta f_{\text{coup}}^{(k)} = -\gamma \sum_{r \in \{k-1,\, k+1\}} \left(\mathbf{1}[x_i' = x_i^{(r)}] - \mathbf{1}[x_i^{(k)} = x_i^{(r)}]\right) \tag{3.28}$$

La transición se acepta según el criterio de *Metropolis* (Metropolis et al., 1953):

$$P_{\text{accept}} = \begin{cases} 1 & \text{si } \Delta f_{\text{ext}} < 0 \\ \exp(-\beta\, \Delta f_{\text{ext}}) & \text{si } \Delta f_{\text{ext}} \geq 0 \end{cases} \tag{3.29}$$

Para el parámetro γ juega el mismo papel que $A(s)$ en QA: cuando es alto favorece la coherencia entre réplicas (análogo a $A(s) \gg B(s)$), y cuando es bajo la búsqueda se concentra en $f(\mathbf{x}^{(k)})$ (análogo a $A(s) \ll B(s)$).

4. **Selección y muestreo de réplicas**

   En los N_{sw} barridos la solución obtenida corresponde a la réplica de menor valor de la función objetivo clásica:

$$\mathbf{x}^* = \arg \min_{k \in \{1,\ldots,P\}} f\left(\mathbf{x}^{(k)}\right) \tag{3.30}$$

   La contribución de f_{coup} se descarta en esta selección debido a que no forma parte de la función objetivo original $f(\mathbf{x})$. Este paso es el análogo clásico de la medición y muestreo en QA: así como el colapso del estado $|\psi(T_a)\rangle$ produce una configuración clásica con probabilidad $|\langle \mathbf{x}^{(k)} | \psi(T_a)\rangle|^2$, la selección de la réplica de menor $f(\mathbf{x}^{(k)})$ extrae la mejor configuración clásica del sistema extendido despues de su equilibración.

SQA por su parte contiene ventajas sobre SA en ciertas clases de problemas (Crosson and Harrow, 2016), y el hardware de recocido cuántico puede superar a las simulaciones clásicas de Monte Carlo (King et al., 2021). Los resultados con SQA son una estimación

conservadora del desempeño alcanzable con hardware cuántico. La correspondencia entre QA y SQA se resume en la Tabla 3.1. La calidad de la simulación depende de tres parámetros: el número de réplicas de Trotter P (fidelidad de la aproximación cuántica), el número de barridos N_{sw} (equilibración del sistema extendido) y la intensidad de fluctuaciones cuánticas γ (balance entre exploración y convergencia, análogo al campo transversal en QA).

Tabla 3.1: Correspondencia entre recocido cuántico (QA) y recocido cuántico simulado (SQA).

Componente	**QA** (hardware)	**SQA** (simulación)
Estado del sistema	$\|\psi(t)\rangle \in \mathcal{H}^{\otimes n}$	P réplicas $\{\mathbf{x}^{(k)}\} \in \{0,1\}^{n\times P}$
Fluctuaciones cuánticas	Campo transversal Γ	Acoplamiento γ entre réplicas
Tunelización	Evolución unitaria bajo H_I	Metropolis sobre E_{total}
Medición	Colapso de $\|\psi(T_a)\rangle$	Réplica de menor energía clásica
Tiempo de recocido	T_a (físico)	N_{sw} barridos

4. Metodología

Este capítulo describe cómo se comparan los algoritmos de recocido simulado y recocido cuántico simulado en el problema de planificación de rutas tridimensionales para UAVs. La Figura 4.1 resume el proceso en cuatro etapas: definición del problema, formulación QUBO, optimización y evaluación comparativa.

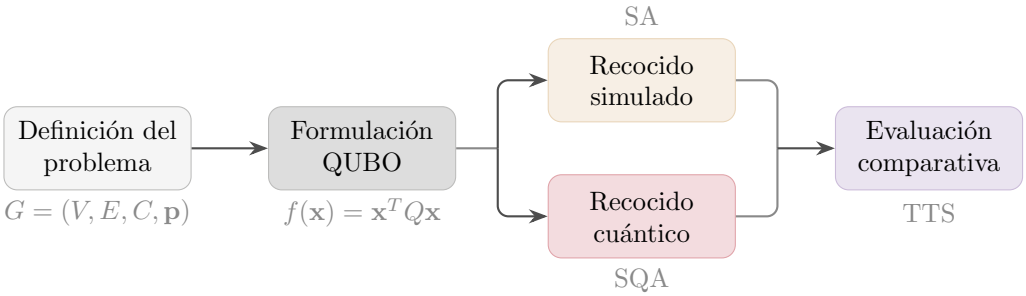


Figura 4.1: Flujo metodológico para la evaluación comparativa de recocido simulado y recocido cuántico simulado aplicados al problema 3D-SPSP.

1. **Definición del problema.** Se establece la representación del entorno de navegación como un grafo ponderado tridimensional y se formaliza el problema del camino más corto (3D-SPSP) sobre esa estructura.

2. **Formulación QUBO.** Se transforma el problema 3D-SPSP en un modelo de optimización binaria cuadrática sin restricciones que codifica el costo de la ruta y las condiciones de factibilidad.

3. **Algoritmos de optimización.** La formulación QUBO resultante se resuelve con dos enfoques: recocido simulado (SA) como método clásico y recocido cuántico simulado (SQA) como método de inspiración cuántica.

4. **Evaluación comparativa.** Se definen las instancias de prueba, los parámetros de cada algoritmo y la métrica *Time to Solution* (TTS) para cuantificar y comparar el desempeño de ambos enfoques.

4.1. Definición del problema de navegación

La planificación de rutas parte de una representación discreta del espacio de vuelo. El entorno tridimensional se modela como un grafo ponderado: los vértices son posiciones accesibles y las aristas son transiciones factibles. De esta forma se convierte en un problema combinatorio sobre un conjunto finito de decisiones binarias.

4.1.1. Grafo tridimensional ponderado

El espacio de navegación se modela mediante un grafo que representa las posiciones accesibles y las conexiones entre ellas.

Definición 4.1 (Grafo tridimensional no dirigido y ponderado)
Un grafo tridimensional no dirigido y ponderado se define como la tupla $G = (V, E, C, \mathbf{p})$, donde:

- $V = \{1, 2, \ldots, n\}$ es un conjunto finito de vértices;
- $E \subseteq \{\{i, j\} \mid i, j \in V, i \neq j\}$ es el conjunto de aristas no dirigidas que representan transiciones factibles entre pares de posiciones;
- $C : E \to \mathbb{R}_{\geq 0}$ es una función de costos que asigna a cada arista $\{i, j\} \in E$ un costo no negativo c_{ij};
- $\mathbf{p} : V \to \{1, \ldots, k\}^3 \subset \mathbb{Z}^3$ es una función de posición que asigna a cada vértice $i \in V$ una coordenada $\mathbf{p}(i) = (x_i, y_i, z_i)$, donde z_i representa la altitud de la posición en el espacio de navegación.

Por tratarse de un grafo no dirigido, se tiene que $c_{ij} = c_{ji}$ para toda arista $\{i, j\} \in E$.

El costo de trayectoria de cada arista $\{i, j\} \in E$ se define como:

$$c_{ij} = \|\mathbf{p}(i) - \mathbf{p}(j)\|_2 + \eta \cdot |z_j - z_i|, \qquad \eta \geq 0 \tag{4.1}$$

donde

$$\|\mathbf{p}(i) - \mathbf{p}(j)\|_2 = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

es la distancia euclidiana entre los extremos de la arista. El segundo término suma una penalización por cambio de altitud entre los puntos $\mathbf{p}(i)$ y $\mathbf{p}(j)$, controlada por el factor $\eta \geq 0$.

Con $\eta = 0$ el costo es solo la distancia euclidiana. Con $\eta > 0$, cada unidad de diferencia de altitud agrega η al costo, lo que refleja el mayor gasto energético del UAV en trayectorias que no son horizontales. Como la penalización depende del valor absoluto $|z_j - z_i|$, el costo resulta simétrico y recorrer una arista en cualquier dirección cuesta lo mismo, en concordancia con el carácter no dirigido del grafo. De este modo, c_{ij} combina la longitud del segmento con la diferencia de altitud de la transición.

La estructura de vecindad e incidencia determina las opciones de movimiento disponibles para el UAV en cada posición, y con ello el espacio de navegación sobre el que se buscan las trayectorias.

Definición 4.2 (Vecindad e incidencia)
Para un vértice $v \in V$, la *vecindad* $\delta(v)$ y el conjunto de *aristas incidentes* $E(v)$ se definen como:

$$\delta(v) = \{u \in V \mid \{v, u\} \in E\}, \qquad E(v) = \{\{v, u\} \in E \mid u \in \delta(v)\}.$$

El *grado* del vértice es $\deg(v) = |\delta(v)|$.

Tener grado elevado implica tener mas transiciones disponibles, uno reducido limita las opciones, como ocurre cerca de los bordes del volumen operacional o en zonas de acceso restringido. En planificación de rutas, $E(v)$ es el conjunto de segmentos de vuelo disponibles en $\mathbf{p}(v)$ como punto de partida, llegada o tránsito.

4.1.2. Problema del camino más corto (3D-SPSP)

En el grafo G se definen dos vértices, el vértice fuente $s \in V$ y el vértice destino $t \in V$, con $s \neq t$. Los demás $V \setminus \{s, t\}$ se denominan vértices intermedios. El objetivo es una secuencia de vértices conectados por aristas que una s con t de forma válida y con costo mínimo.

Tener la secuencia conectada a s con t no se limita para que sea realizable por un UAV. Se necesitan condiciones estructurales que garanticen continuidad sin repeticiones ni ramificaciones.

Definición 4.3 (Ruta válida)
Una ruta P en G desde el vértice fuente s hasta el vértice destino t es una secuencia ordenada de vértices:

$$\mathrm{P} = (v_0, v_1, v_2, \ldots, v_k)$$

donde $v_0 = s$, $v_k = t$, y cada par de vértices consecutivos está conectado por una arista, es decir, $\{v_{i-1}, v_i\} \in E$ para todo $i \in \{1, \ldots, k\}$. Se considera como *ruta válida* si satisface las siguientes *condiciones de factibilidad*:

a. **Conectividad:** Todas las aristas $\{v_{i-1}, v_i\}$ que conectan vértices consecutivos pertenecen al conjunto E;

b. **Ausencia de ciclos:** Los vértices son distintos entre sí, es decir, $v_i \neq v_j$ para todo $i \neq j$;

c. **Ausencia de ramificaciones:** Cada vértice intermedio v_i con $0 < i < k$ participa en exactamente dos aristas de la ruta: una que lo conecta con su predecesor v_{i-1} y otra que lo conecta con su sucesor v_{i+1}.

Estas condiciones de factibilidad definen las rutas admisibles, rutas continuas sin ciclos ni ramificaciones. El dominio de búsqueda se limita al conjunto $\mathcal{P}_{s,t}$ de rutas

válidas, excluyendo saltos entre vértices no adyacentes y recorridos redundantes.

Para poder cuantificar la calidad de una ruta, el costo debe mostrar la distancia obtenida y el esfuerzo energético de los ascensos, caracteristicas importantes en la operación de un UAV en un vuelo. Se define la función de costo que se usará como criterio de optimización.

Definición 4.4 (Costo de una ruta)
El *costo total* de una ruta válida $\mathrm{P} = (v_0, \ldots, v_k)$ se define como la suma de los costos de trayectoria de sus aristas:

$$C(\mathrm{P}) = \sum_{i=1}^{k} c_{v_{i-1}v_i}$$

donde $c_{v_{i-1}v_i}$ está dado por la Ecuación (4.1).

Con ruta válida y costo definidos, el problema de optimización queda establecido. Sea $\mathcal{P}_{s,t}$ el conjunto de rutas válidas que conectan s con t en G. El 3D-SPSP consiste en determinar P^* que minimice el costo acumulado:

$$\mathrm{P}^* = \arg\min_{\mathrm{P}\in\mathcal{P}_{s,t}} C(\mathrm{P}) \tag{4.2}$$

La solución P^* no es necesariamente única (degeneración): pueden existir múltiples rutas con el mismo costo mínimo.

4.2. Formulación QUBO del problema 3D-SPSP

Para resolver el 3D-SPSP con recocido simulado y cuántico hace falta transformarlo en una instancia QUBO compatible con la optimización binaria cuadrática. Siguiendo la penalización cuadrática para múltiples restricciones y la construcción de Krauss and McCollum (2020), se adopta un factor de penalización uniforme λ para las tres condiciones de factibilidad. Los términos de penalización asociados a cada condición se denotan como

$$P_k(\mathbf{x}), \quad k = 1, 2, 3,$$

los cuales se agrupan en el componente de penalización total

$$Q_p(\mathbf{x}) = \sum_{k=1}^{3} P_k(\mathbf{x}).$$

De este modo, la función objetivo toma la forma:

$$Q(\mathbf{x}) = Q_c(\mathbf{x}) + \lambda\, Q_p(\mathbf{x}), \tag{4.3}$$

donde $Q_c(\mathbf{x})$ cuantifica el costo de la trayectoria codificada por $\mathbf{x}$. La formulación del vector de estado que parametriza esta función se establece a continuación.

Definición 4.5 (Vector de estado)
El estado del sistema se representa mediante un vector binario $\mathbf{x} \in \{0,1\}^{|V|+|E|}$, definido como

$$\mathbf{x} = (\underbrace{x_1, \ldots, x_i, \ldots, x_{|V|}}_{|V| \text{ variables de vértice}}, \underbrace{x_{12}, \ldots, x_{ij}, \ldots}_{|E| \text{ variables de arista}})^T,$$

donde $x_i = 1$ indica que el vértice $i \in V$ es visitado por la ruta, y $x_{ij} = 1$ indica que la arista $\{i,j\} \in E$ es recorrida. Los vértices fuente y destino se consideran activos por construcción: $x_s = x_t = 1$.

Componentes de la función objetivo

El componente de costo contiene la suma los costos de la ruta de las aristas seleccionadas

$$Q_c(\mathbf{x}) = \sum_{\{i,j\} \in E} c_{ij}\, x_{ij}. \tag{4.4}$$

Debido a que los c_{ij} ya contienen la distancia geométrica y el efecto del ascenso (Ecuación (4.1)), este componente contiene asi tanto la longitud recorrida como el costo energético asociado a la altitud. El *componente de penalización* se descompone como

$$Q_p(\mathbf{x}) = Q_s(\mathbf{x}) + Q_t(\mathbf{x}) + Q_{\text{cont}}(\mathbf{x}), \tag{4.5}$$

donde cada término propone una condición estructural específica sobre la trayectoria.

- **Restricción del vértice fuente.** Permite que la ruta del UAV inicialice en la posición $\mathbf{p}(s)$ mediante la selección de exactamente una arista incidente:

$$Q_s(\mathbf{x}) = -x_s^2 + \left(x_s - \sum_{j \in \delta(s)} x_{sj} \right)^2 \tag{4.6}$$

  El mínimo $Q_s = -1$ solo se alcanza cuando hay exactamente una arista activa en s. Sin aristas, la trayectoria se desconecta del origen, con más de una la ruta se ramifica.

- **Restricción del vértice destino.** Genera una condición simétrica en la posición objetivo $\mathbf{p}(t)$, permitiendo que exactamente un segmento de vuelo llegue al destino:

$$Q_t(\mathbf{x}) = -x_t^2 + \left(x_t - \sum_{i \in \delta(t)} x_{it} \right)^2 \tag{4.7}$$

  De manera análoga, se tiene $Q_t = -1$ cuando alcanza con una sola arista activa en t. Sin aristas el UAV no llega al destino, y con varias la ruta converge desde distintos frentes, lo que rompe la secuencialidad.

- **Restricción de continuidad.** Impone que cada vértice intermedio activo tenga exactamente dos segmentos de vuelo incidentes, uno de llegada y otro de salida:

$$Q_{\text{cont}}(\mathbf{x}) = \sum_{i \in V \setminus \{s,t\}} \left(2x_i - \sum_{j \in \delta(i)} x_{ij} \right)^2 \tag{4.8}$$

  Para un vértice inactivo ($x_i = 0$), la contribución se descarta solo si ninguna arista incidente está activa, obteniendo que se eviten segmentos flotantes. Para un vértice activo ($x_i = 1$) el mínimo obliga a que la suma de aristas incidentes sea dos, lo que mantiene la continuidad de la trayectoria al pasar por esa posición.

Los tres términos de penalización proponen configuraciones en las que la ruta conecta fuente y destino de forma secuencial, sin ciclos ni ramificaciones, de modo que los mínimos de $Q(\mathbf{x})$ coinciden con el conjunto $\mathcal{P}_{s,t}$ de rutas válidas.

Parámetro de penalización

El parámetro de penalización λ de la Ecuación (4.3) controla qué tan separadas quedan las configuraciones factibles de las infactibles. Si este valor es demasiado bajo, una violación puede costar menos que una ruta válida, si es demasiado alto la convergencia se frena.

Para mantener una escala entre ambos componentes, λ se puede definir en función de los costos del grafo como:

$$\lambda = \sum_{\{i,j\} \in E} c_{ij}. \tag{4.9}$$

De esta manera, la penalización por cualquier violación estructural queda en el orden del costo total de las rutas factibles y las configuraciones infactibles siempre resultan con un costo alto. Debido a que los costos c_{ij} ya dependen de η, el valor de λ se adapta de forma indirecta a la geometría del entorno y al régimen de penalización por altitud, lo que mantiene consistentes ambos componentes en todas las instancias.

4.3. Algoritmos de optimización

Teniendo formulado $Q(\mathbf{x})$, el problema se reduce asi a encontrar la configuración binaria que lo minimice. Se emplean dos algoritmos: recocido simulado (SA) y recocido cuántico simulado (SQA) y se desarrollan sobre $\{0,1\}^{|V|+|E|}$ compartiendo el mismo presupuesto de evaluaciones, aunque con estrategias de exploración distintas y a su vez, obtienen configuraciones vecinas invirtiendo una variable aleatoria y usan el criterio de Metropolis para aceptar o rechazar transiciones, con modificaciones en SQA por el acoplamiento entre réplicas.

4.3.1. Recocido simulado

El Algoritmo 4.1 se desarrolla en una única configuración durante $k_{\text{máx}}$ iteraciones, la configuración inicial se genera al azar y la temperatura se fija en T_0 (línea 1). Para cada iteración se selecciona una variable al azar y se invierte su valor, generando un vecino (líneas 3–4). El cambio en la función objetivo $Q(\mathbf{x})$ se calcula como Δf (línea 5). La transición se acepta mediante el criterio de Metropolis (línea 6): los movimientos que reducen Q se aceptan siempre, y los desfavorables se aceptan con probabilidad $\exp(-\Delta f/T)$. La mejor configuración visitada $\mathbf{x}^*$ se actualiza al encontrar una de menor valor (líneas 8–9). El enfriamiento geométrico $T \leftarrow \alpha \cdot T$ (línea 12) reduce gradualmente la probabilidad de aceptar transiciones desfavorables hasta inducir convergencia.

Algoritmo 4.1 Recocido simulado para QUBO (3D-SPSP)

Entrada: Coeficientes $Q \in \mathbb{R}^{n\times n}$ de la función objetivo, Temperatura inicial T_0, Tasa de enfriamiento $\alpha \in (0,1)$, Iteraciones máximas $k_{\text{máx}}$
Salida: Configuración de mínima energía $\mathbf{x}^*$

1. Generar $\mathbf{x} \in \{0,1\}^n$ aleatoriamente; $\mathbf{x}^* \leftarrow \mathbf{x}$; $T \leftarrow T_0$ ▷ **1.** Inicialización
2. **for** $k = 1$ **to** $k_{\text{máx}}$ **do**
3. Seleccionar $i \sim \text{Uniforme}\{1,\dots,n\}$
4. $\mathbf{x}' \leftarrow \mathbf{x}$ con $x'_i \leftarrow 1 - x_i$ ▷ **2.** Generación de vecino
5. $\Delta f \leftarrow \mathbf{x}'^T Q\,\mathbf{x}' - \mathbf{x}^T Q\,\mathbf{x}$ ▷ Evaluación de Δf
6. **if** $\Delta f < 0$ **o** $\text{Uniforme}(0,1) < \exp(-\Delta f/T)$ **then** ▷ **3.** Criterio de aceptación
7. $\mathbf{x} \leftarrow \mathbf{x}'$
8. **if** $f(\mathbf{x}) < f(\mathbf{x}^*)$ **then**
9. $\mathbf{x}^* \leftarrow \mathbf{x}$
10. **end if**
11. **end if**
12. $T \leftarrow \alpha \cdot T$ ▷ **4.** Esquema de enfriamiento
13. **end for**
14. **return** $\mathbf{x}^*$

4.3.2. Recocido cuántico simulado

El Algoritmo 4.2 empieza construyendo el sistema extendido mediante la aproximación de Suzuki-Trotter (línea 1), que se traduce en P réplicas de configuraciones conectadas en anillo con acoplamiento γ. Cada réplica se inicializa al azar (línea 2). Durante los N_{sw} barridos, para cada réplica k se selecciona una variable al azar y se propone invertir su valor (líneas 5–6). Se evalúa el efecto sobre la función objetivo extendida: $\Delta f^{(k)}$ mide el impacto en $Q(\mathbf{x}^{(k)})$ (línea 7), mientras que $\Delta f^{(k)}_{\text{coup}}$ mide cómo la inversión altera la coherencia con las réplicas vecinas (línea 8). La transición se acepta mediante el criterio de Metropolis aplicado a la suma de ambos términos (líneas 9–10). Al terminar, se reporta la réplica de menor $Q(\mathbf{x}^{(k)})$, ignorando f_{coup} porque solo interviene durante la evolución para simular el efecto cuántico y no pertenece al problema original.

Algoritmo 4.2 Recocido cuántico simulado para QUBO (3D-SPSP)

Entrada: Coeficientes $Q \in \mathbb{R}^{n\times n}$ de la función objetivo, Réplicas de Trotter P, Inverso de temperatura β, Fluctuaciones cuánticas γ, Barridos N_{sw}
Salida: Réplica de mínima energía clásica $\mathbf{x}^*$

Aproximar Z mediante P réplicas conectadas ▷ **1.** Codificación del sistema extendido
Generar $\mathbf{x}^{(k)}$ aleatoriamente para $k = 1, \ldots, P$ ▷ **2.** Preparación del estado inicial clásico
for sw $= 1$ **to** N_{sw} **do**
  for $k = 1$ **to** P **do**
    Seleccionar $i \sim \text{Uniforme}\{1, \ldots, n\}$
    $\mathbf{x}' \leftarrow \mathbf{x}^{(k)}$ con $x'_i \leftarrow 1 - x_i^{(k)}$ ▷ **3.** Esquema de barridos
    $\Delta f^{(k)} \leftarrow \mathbf{x}'^{T} Q\, \mathbf{x}' - \mathbf{x}^{(k)^T} Q\, \mathbf{x}^{(k)}$ ▷ Evaluación de $\Delta f^{(k)}$
    $\Delta f^{(k)}_{\text{coup}} \leftarrow -\gamma \sum_{r\in\{k-1,\,k+1\}} \left(\mathbf{1}[x'_i = x_i^{(r)}] - \mathbf{1}[x_i^{(k)} = x_i^{(r)}]\right)$ ▷ Evaluación de $\Delta f^{(k)}_{\text{coup}}$
    if $\Delta f^{(k)} + \Delta f^{(k)}_{\text{coup}} < 0$ **o** $u < \exp\big(-\beta(\Delta f^{(k)} + \Delta f^{(k)}_{\text{coup}})\big)$ **then** ▷ Criterio de aceptación
      $\mathbf{x}^{(k)} \leftarrow \mathbf{x}'$
    end if
  end for
end for
return $\mathbf{x}^* \leftarrow \arg\min_k f\left(\mathbf{x}^{(k)}\right)$ ▷ **4.** Selección y muestreo de réplicas

4.4. Evaluación comparativa

La comparación entre SA y SQA se puede observar en tres elementos: (i) las instancias de prueba, (ii) la configuración experimental con presupuesto computacional equitativo, y (iii) la métrica *Time to Solution* (TTS), que combina velocidad y probabilidad de éxito en una sola instancia.

4.4.1. Instancias de prueba

Los experimentos incluyen doce instancias del problema 3D-SPSP, construidas a partir de tres factores independientes:

- **Escala** $k \in \{5, 6, 7\}$: determina el tamaño del espacio de navegación y la dimension del problema QUBO.
- **Espacio de navegación**: G_{abierto}, retícula sin restricciones que establece la línea base, y $G_{\text{restringido}}$, espacio con obstáculos volumétricos que representan zonas de vuelo prohibido.
- **Régimen de costo** $\eta \in \{0, 1\}$: con $\eta = 0$ el costo c_{ij} es solo la distancia euclidiana; con $\eta = 1$ se añade una penalización proporcional a la ganancia de altitud según la Ecuación (4.1).

En todas las instancias se comparten el vértice fuente $s = (1, 1, 1)$ y el destino $t = (k, k, k)$, de tal modo que las diferencias en el rendimiento de SA y SQA se relacionen unicamente a la estructura del grafo y al régimen de costo.

Estructura del espacio de navegación

En cada instancia se modela sobre una retícula de $k \times k \times k$ posiciones con separación unitaria, donde cada posición se conecta con sus seis vecinos inmediatos en las direcciones axiales. Las dos variantes del espacio de navegación se distinguen por la presencia o ausencia de obstáculos:

- **Espacio abierto** (G_{abierto}). La retícula no tiene restricciones de acceso, así que todos los vértices y aristas permanecen activos. El número de vértices y aristas queda determinado por $|V| = k^3$ y

$$|E|_{\text{abierto}}(k) = 3k^2(k-1), \tag{4.10}$$

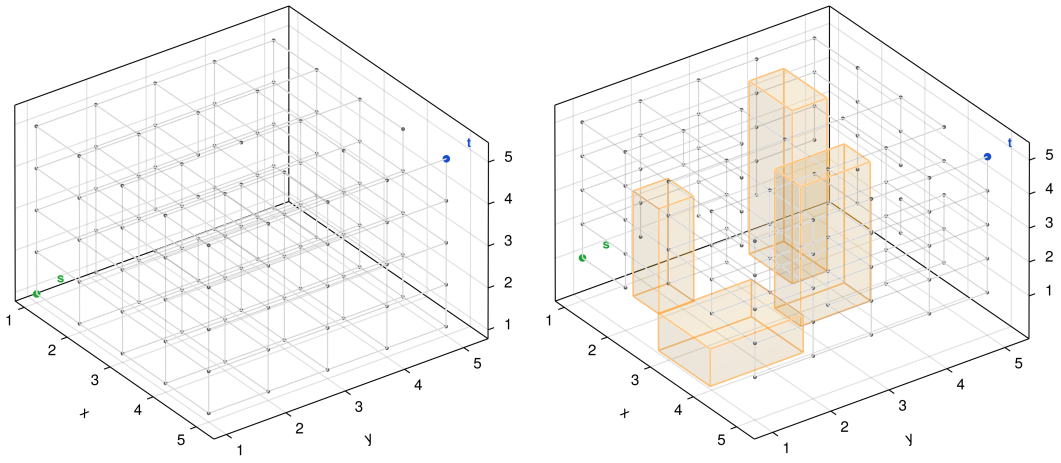
  expresión que surge del conteo de aristas en una malla cúbica regular: para cada una de las tres dimensiones hay $k^2(k-1)$ aristas axiales, sumando $3k^2(k-1)$.

- **Espacio restringido** ($G_{\text{restringido}}$). Se insertan obstáculos volumétricos en ubicaciones específicas de la retícula, eliminando selectivamente las aristas incidentes a regiones de vuelo prohibido. La dimensionalidad del problema QUBO resultante es:

$$n_{\text{restringido}}(k) = k^3 + |E|_{\text{restringido}}(k), \tag{4.11}$$

  donde $|E|_{\text{restringido}}(k)$ es el número de aristas presentes tras eliminar aquellas incidentes a obstáculos.

La dimensionalidad del caso abierto se obtiene de la Ecuación (4.10): $n_{\text{abierto}}(k) = k^3 + 3k^2(k-1)$. La Figura 4.2 muestra ambas variantes para $k = 5$ con $\eta = 0$, ilustrando la diferencia estructural entre la retícula completa y la que tiene obstáculos volumétricos.



(a) (b)

Figura 4.2: Instancias para $k = 5$: (a) G_{abierto} con $|V| = 125$ vértices y $|E| = 300$ aristas ($\rho = 0{,}0387$); (b) $G_{\text{restringido}}$ con $|V| = 125$ vértices y $|E| = 217$ aristas ($\rho = 0{,}0281$), donde los obstáculos volumétricos eliminan 83 aristas de la retícula. El vértice fuente s se indica en verde y el destino t en azul.

Análisis comparativo de densidad y conectividad

La densidad del grafo se define como:

$$\rho = \frac{|E|}{|V|(|V|-1)/2}, \tag{4.12}$$

y da una medida normalizada de conectividad para comparar ambas variantes en las tres escalas. En G_{abierto}, ρ decrece con k por el crecimiento superlineal de las aristas posibles frente a las presentes en la retícula. En $G_{\text{restringido}}$, los obstáculos añaden una reducción que varía con la escala: se eliminan 83 aristas para $k = 5$ (27.7 %), 119 para $k = 6$ (22.0 %), y 245 para $k = 7$ (27.8 %). El impacto relativo se mantiene entre 22 % y 28 %, de modo que ambas variantes conservan una estructura de conectividad comparable. La Tabla 4.1 muestra la dimensionalidad QUBO y la densidad de ambas variantes en las tres escalas.

Energía óptima y validación de instancias

El mínimo teórico de $Q(\mathbf{x})$ para cada instancia se obtiene evaluando la función objetivo en $\mathbf{x}^*$ que codifica la ruta óptima P^*. Como P^* cumple las tres condiciones de factibilidad, el componente de costo recupera $C(\mathrm{P}^*)$, $Q_c(\mathbf{x}^*) = C(\mathrm{P}^*)$. Los términos de penalización alcanzan sus valores mínimos: $Q_s(\mathbf{x}^*) = Q_t(\mathbf{x}^*) = -1$ por la ausencia de

Tabla 4.1: Dimensionalidad del problema QUBO y densidad del grafo para las dos variantes del espacio de navegación en las tres escalas.

		G_{abierto}			$G_{\text{restringido}}$		
k	$\lvert V\rvert$	$\lvert E\rvert$	n	ρ	$\lvert E\rvert$	n	ρ
5	125	300	425	0.0387	217	342	0.0281
6	216	540	756	0.0233	421	637	0.0172
7	343	882	1225	0.0150	637	980	0.0109

ramificaciones en fuente y destino, y $Q_{\text{cont}}(\mathbf{x}^*) = 0$ por la continuidad en los vértices intermedios. Reemplazando cada componente en

$$Q(\mathbf{x}) = Q_c(\mathbf{x}) + \lambda \left[Q_s(\mathbf{x}) + Q_t(\mathbf{x}) + Q_{\text{cont}}(\mathbf{x}) \right],$$

el mínimo teórico resulta:

$$\begin{aligned} f(\mathbf{x}^*) &= C(\mathrm{P}^*) + \lambda \left[(-1) + (-1) + 0 \right] \\ &= C(\mathrm{P}^*) - 2\lambda. \end{aligned} \tag{4.13}$$

El valor de $C(\mathrm{P}^*)$ se determina para cada instancia con el algoritmo de Dijkstra sobre el grafo G, lo que da una referencia exacta e independiente de los algoritmos de optimización. Este valor sirve como criterio de validación: una ejecución de SA o SQA se considera exitosa si la configuración encontrada cumple $Q(\mathbf{x}) = f(\mathbf{x}^*)$. La Tabla 4.2 reporta $C(\mathrm{P}^*)$ y $f(\mathbf{x}^*)$ para las doce instancias.

Tabla 4.2: Parámetros estructurales y mínimos teóricos de las doce instancias del problema 3D-SPSP.

k	**Instancia**	η	$\lvert V\rvert$	$\lvert E\rvert$	n	$C(\mathrm{P}^*)$	$f(\mathbf{x}^*)$
5	G_{abierto}	0	125	300	425	12	-588
		1	125	300	425	16	-784
	$G_{\text{restringido}}$	0	125	217	342	12	-422
		1	125	217	342	16	-580
6	G_{abierto}	0	216	540	756	15	-1065
		1	216	540	756	20	-1420
	$G_{\text{restringido}}$	0	216	421	637	15	-827
		1	216	421	637	20	-1132
7	G_{abierto}	0	343	882	1225	18	-1746
		1	343	882	1225	24	-2328
	$G_{\text{restringido}}$	0	343	637	980	18	-1256
		1	343	637	980	24	-1732

4.4.2. Configuración experimental

El diseño de experimentos requiere que SA y SQA consuman el mismo número de evaluaciones de $Q(\mathbf{x})$ por ejecución, para que las diferencias reflejen propiedades algorítmicas y no un mayor esfuerzo computacional, para ello define el presupuesto $B = 1{,}600$, del cual se derivan los barridos de cada algoritmo:

$$k_{\text{máx}} = B, \qquad N_{\text{sw}} = \frac{B}{P}, \tag{4.14}$$

de este modo SA ejecuta $k_{\text{máx}} = 1{,}600$ evaluaciones directas, mientras que SQA hace $N_{\text{sw}} = 50$ barridos sobre $P = 32$ réplicas, con un total de $50 \times 32 = 1{,}600$ evaluaciones por ejecución en ambos casos. La Tabla 4.3 contiene los valores.

Para SA, la temperatura inicial $T_0 = 500{,}0$ permite explorar sin restricciones en las primeras iteraciones, debido a que $\exp(-\Delta f/T_0) \approx 1$ para incrementos típicos de $Q(\mathbf{x})$; la tasa $\alpha = 0{,}995$ da un decaimiento geométrico con $T_{k_{\text{máx}}} \approx 0$ al final de las $k_{\text{máx}} = 1{,}600$ evaluaciones, llevando el sistema progresivamente a regiones de baja energía.

Para SQA, $P = 32$ réplicas da una aproximación más precisa de la función de partición cuántica que $P = 16$ (Santoro et al., 2002; Martoňák et al., 2002); $\beta = 5{,}0$ favorece configuraciones de baja energía clásica durante el muestreo; y $\gamma = 3{,}0$ se reduce linealmente hasta cero durante los $N_{\text{sw}} = 50$ barridos, llevando el sistema al régimen clásico de forma progresiva.

Ambos algoritmos se ejecutaron $R = 100$ veces por instancia con inicialización aleatoria independiente en cada corrida, implementados en Julia (Bezanson et al., 2017) sobre un entorno de un solo hilo, de modo que el tiempo de ejecución t_{run} de cada

Tabla 4.3: Parámetros de SA y SQA empleados en los experimentos.

Algoritmo	Parámetro	Símbolo	Valor
SA	Temperatura inicial	T_0	500,0
	Tasa de enfriamiento	α	0,995
	Evaluaciones	$k_{\text{máx}}$	1,600
	Ejecuciones por instancia	R	100
SQA	Réplicas de Trotter	P	32
	Inverso de temperatura	β	5,0
	Campo transverso inicial	γ	3,0
	Barridos	N_{sw}	50
	Ejecuciones por instancia	R	100

corrida es reproducible y comparable entre algoritmos. Este valor, estimado como la media aritmética de las R corridas, constituye uno de los dos insumos de la métrica de desempeño definida a continuación.

4.4.3. Métrica de desempeño: *Time to Solution*

SA y SQA son algoritmos probabilísticos: una ejecución individual no garantiza encontrar $\mathbf{x}^*$ que minimiza $Q(\mathbf{x})$, por lo que el desempeño no puede medirse con el tiempo de una corrida aislada. La métrica *Time to Solution* (TTS) combina el costo por ejecución y la probabilidad de éxito acumulada tras repeticiones independientes, y da un criterio unificado para la comparación (Albash and Lidar, 2018; Mohseni et al., 2022).

El TTS parte de la probabilidad de éxito p_s de una ejecución individual, definida como la fracción de corridas que alcanzan el valor óptimo sobre el conjunto $\mathcal{X}_R = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(R)}\}$ de R ejecuciones independientes:

$$p_s = \frac{\left|\left\{\mathbf{x}^{(r)} \in \mathcal{X}_R \mid f(\mathbf{x}^{(r)}) = f(\mathbf{x}^*)\right\}\right|}{R}, \tag{4.15}$$

donde $f(\mathbf{x}^*)$ es el mínimo teórico de la función objetivo, valor calculado mediante el algoritmo de Dijkstra para cada instancia. Como las ejecuciones son independientes entre sí, la probabilidad de observar al menos una solución óptima tras N ejecuciones es:

$$P(N) = 1 - (1 - p_s)^N, \tag{4.16}$$

donde se obtiene el número mínimo de ejecuciones necesarias para alcanzar una probabilidad objetivo p_d:

$$N(p_d, p_s) = \left\lceil \frac{\log(1 - p_d)}{\log(1 - p_s)} \right\rceil. \tag{4.17}$$

El TTS usa el cociente sin redondeo junto con el costo por ejecución t_{run} para calcular el tiempo total necesario para garantizar el éxito con probabilidad de al menos p_d:

$$\text{TTS}(p_d,\, p_s) = t_{\text{run}} \cdot \frac{\log(1 - p_d)}{\log(1 - p_s)}. \tag{4.18}$$

Un algoritmo con menor p_s pero también menor t_{run} puede ser competitivo frente a uno con mayor p_s pero ejecuciones más costosas, y el TTS captura este balance de forma directa. Los valores se calculan con $p_d = 0{,}99$ para cada combinación de algoritmo e instancia.

5. Resultados y discusión

Se comparan los algoritmosSA y SQA en las doce instancias del problema 3D-SPSP. El análisis cubre tres aspectos: (i) convergencia al óptimo teórico, (ii) probabilidad de éxito según la escala y (iii) comparación global mediante TTS (*Time to Solution*).

5.1. Análisis de convergencia

Para ambos algoritmos se alcanzan el óptimo teórico $f(\mathbf{x}^*)$ en las doce instancias (columna $f_{\text{mín}}$ de la Tabla 5.1). Confirmato que el operador de inversion de bit recorra el espacio de búsqueda lo suficiente para llegar al mínimo global en todos los casos.

La función objetivo $f(\mathbf{x})$ en los experimentos corresponde a la formulación QUBO desarrollada, que muestra el costo de la ruta con penalizaciones por violaciones estructurales. El valor óptimo $f(\mathbf{x}^*)$ se calculó con Dijkstra sobre el grafo de navegación para cada instancia y sirve como referencia para evaluar cada ejecución.

Para cada instancia y algoritmo se realizan $R = 100$ ejecuciones independientes. Sea $\{\mathbf{x}^{(r)}\}_{r=1}^{R}$ el conjunto de configuraciones finales obtenidas, con

$$f^{(r)} = f\left(\mathbf{x}^{(r)}\right) \tag{5.1}$$

el valor de la función objetivo en la r-ésima ejecución. Las métricas de convergencia se definen como:

$$f_{\text{mín}} = \min_{1 \le r \le R} f^{(r)} \tag{5.2a}$$

$$\mathbb{E}[f] = \frac{1}{R} \sum_{r=1}^{R} f^{(r)} \tag{5.2b}$$

$$\sigma_f = \sqrt{\frac{1}{R} \sum_{r=1}^{R} \left(f^{(r)} - \mathbb{E}[f]\right)^2} \tag{5.2c}$$

$$\Delta f = \max_{1 \le r \le R} f^{(r)} - \min_{1 \le r \le R} f^{(r)} \tag{5.2d}$$

La Ec. 5.2a da el mejor valor observado, la Ec. 5.2b la energía media, la Ec. 5.2c la dispersión estadística y la Ec. 5.2d el rango total de la distribución empírica de cada algoritmo. Que $\sigma_f > 0$ y $\Delta f > 0$ en todas las instancias se debe a la naturaleza estocástica de SA y SQA.

Para examinar el comportamiento de cada algoritmo más allá de los promedios, la Figura 5.1 muestra la distribución completa de energías en las $R = 100$ ejecuciones. El eje vertical de cada diagrama es la desviación respecto al óptimo teórico, $f(\mathbf{x}) - f(\mathbf{x}^*)$. A continuación se describe cómo evolucionan estas distribuciones con la escala.

Tabla 5.1: Resultados de convergencia para las doce instancias del problema 3D-SPSP. Las métricas se estiman sobre $R = 100$ ejecuciones independientes por instancia y algoritmo.

k	Instancia	η	Alg.	$f_{\text{mín}}$	$\mathbb{E}[f]$	σ_f	Δf
5	G_{abierto}	0	SA	-588	$-587{,}880$	0,477	2
			SQA	-588	$-587{,}060$	1,286	6
		1	SA	-784	$-783{,}880$	0,477	2
			SQA	-784	$-783{,}080$	1,405	6
	$G_{\text{restringido}}$	0	SA	-422	$-421{,}800$	0,603	2
			SQA	-422	$-421{,}820$	0,575	2
		1	SA	-580	$-579{,}860$	0,652	4
			SQA	-580	$-579{,}720$	0,697	2
6	G_{abierto}	0	SA	-1065	$-1064{,}120$	1,373	6
			SQA	-1065	$-1062{,}160$	2,330	10
		1	SA	-1420	$-1419{,}220$	1,630	8
			SQA	-1420	$-1416{,}940$	2,777	12
	$G_{\text{restringido}}$	0	SA	-827	$-826{,}260$	1,440	6
			SQA	-827	$-824{,}960$	1,989	10
		1	SA	-1132	$-1131{,}080$	1,851	10
			SQA	-1132	$-1129{,}360$	2,460	8
7	G_{abierto}	0	SA	-1746	$-1742{,}740$	3,187	16
			SQA	-1746	$-1739{,}820$	3,675	16
		1	SA	-2328	$-2324{,}180$	4,504	26
			SQA	-2328	$-2320{,}340$	4,908	22
	$G_{\text{restringido}}$	0	SA	-1256	$-1253{,}320$	3,130	14
			SQA	-1256	$-1251{,}660$	2,771	16
		1	SA	-1732	$-1728{,}440$	4,154	18
			SQA	-1732	$-1726{,}500$	3,320	16

- **Escala $k = 5$:** SA concentra las energías mucho más que SQA. Sus diagramas de cajas son compactos alrededor del óptimo, con medianas en cero y bigotes cortos. SQA es mucho más disperso, sobre todo en G_{abierto}: $\sigma_f = 1{,}286$ ($\eta = 0$) y 1,405 ($\eta = 1$), frente a 0,477 de SA.

  En $G_{\text{restringido}}$, la reducción del espacio factible beneficia más a SQA: su desviación ($\sigma_f = 0{,}575$ y 0,697) se acerca a la de SA (0,603 y 0,652). Es el único caso donde ambos tienen dispersiones comparables.

- **Escala $k = 6$:** Las cajas se extiende con respecto a $k = 5$. SA mantiene ventaja, pero la diferencia con SQA crece. En G_{abierto}, SA tiene $\sigma_f \in [1{,}373,\ 1{,}630]$ mientras SQA llega a [2,330, 2,777]. Los bigotes de SQA son más largos y aparecen valores atípicos alejados del óptimo.

  El régimen de costo ($\eta = 1$) no afecta igual a ambos espacios. En SA, σ_f sube en G_{abierto} (de 1,373 a 1,630) pero se mantiene en el mismo orden en $G_{\text{restringido}}$ (1,440 frente a 1,851). En SQA el efecto es menos claro.

- **Escala $k = 7$:** Aquí se notan las limitaciones de SQA con el presupuesto fijo. En G_{abierto} ambos algoritmos muestran dispersiones altas ($\sigma_f > 3{,}1$ para SA, $\geq 3{,}7$ para SQA) y SQA alterna entre encontrar el óptimo y quedarse en mínimos locales.

  En $G_{\text{restringido}}$ la situación se invierte: SQA resulta menos disperso que SA ($\sigma_f =$ 2,771 y 3,320 frente a 3,130 y 4,154). Esto sugiere que al eliminar aristas (de 1225 a 980 variables) mejora la coherencia entre las réplicas de Trotter.

SA tiene menor dispersión que SQA en casi todas las instancias, y la ventaja crece con la escala, pero la concentración de energías no captura del todo la capacidad de cada algoritmo de alcanzar el óptimo en una ejecución. Para eso se emplea la probabilidad de éxito.

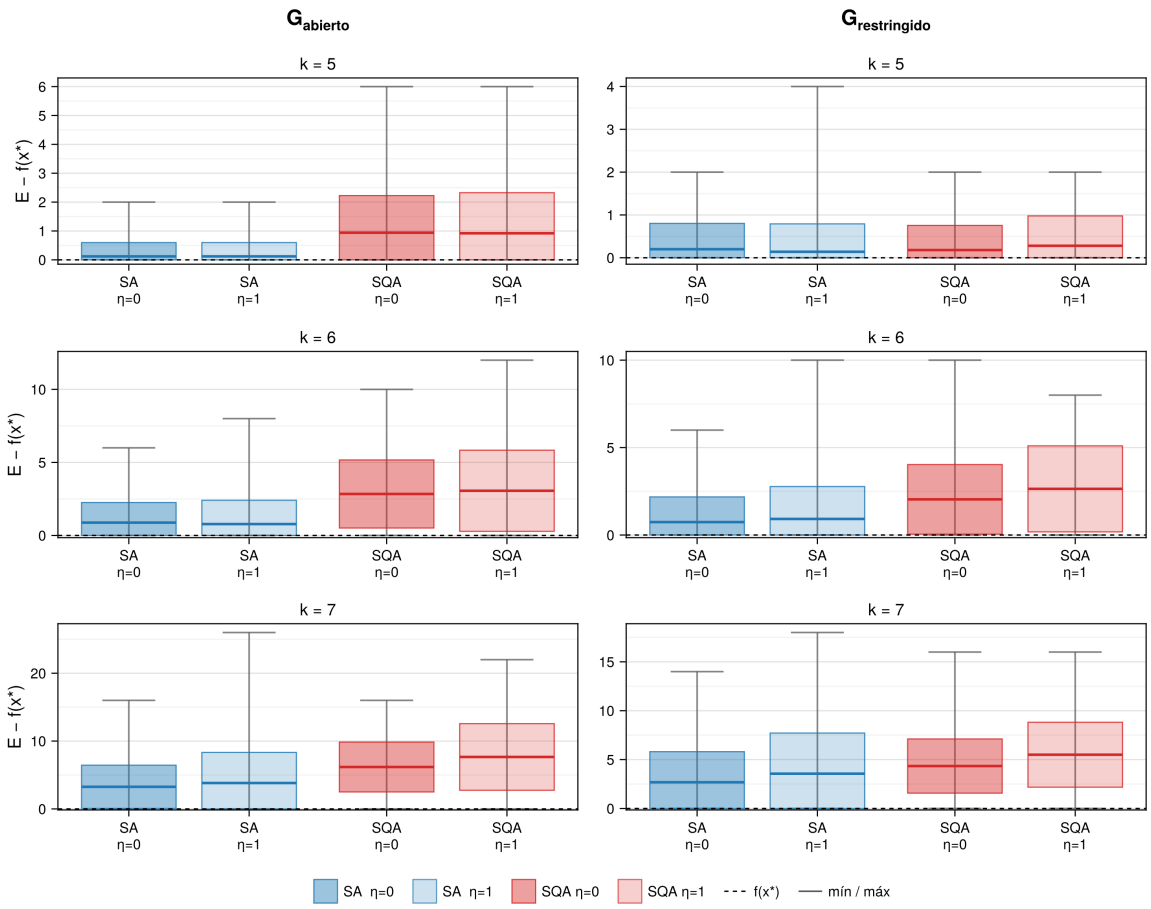


Figura 5.1: Distribución de energías por $R = 100$ ejecuciones en las doce instancias del 3D-SPSP. Las filas corresponden a las escalas $k \in \{5, 6, 7\}$ y las columnas a los espacios de navegación G_{abierto} y $G_{\text{restringido}}$. Cada panel contiene cuatro diagramas de cajas: SA y SQA bajo los regímenes de costo $\eta = 0$ y $\eta = 1$. La línea horizontal indica el óptimo teórico $f(\mathbf{x}^*)$.

5.2. Probabilidad de éxito y escalabilidad

SA supera a SQA en p_s en once de las doce instancias. Más allá de la ventaja puntual, lo que define la comparación es la tendencia con la escala: SA pasa de $p_s \in [0{,}90,\ 0{,}95]$ en $k = 5$ a $[0{,}25,\ 0{,}44]$ en $k = 7$, mientras que SQA cae de $[0{,}60,\ 0{,}91]$ a solo $[0{,}06,\ 0{,}10]$ en el mismo rango, una pérdida de más de un orden de magnitud. La única excepción es $G_{\text{restringido}}$, $\eta = 0$, $k = 5$, donde SQA llega a $p_s = 0{,}91$ frente a 0,90 de SA. La diferencia es mínima, pero alcanza para ver que reducir el espacio factible puede emparejar el desempeño en instancias pequeñas. La Figura 5.2 presenta la evolución de p_s para las cuatro combinaciones.

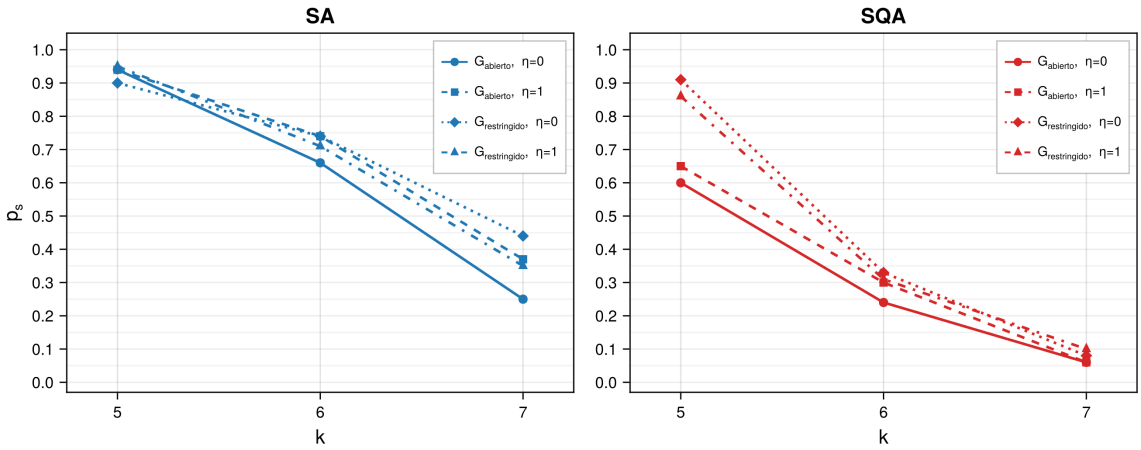


Figura 5.2: Evolución de p_s con la escala $k \in \{5, 6, 7\}$ para SA y SQA. Cada curva corresponde a una combinación de espacio de navegación (G_{abierto}, $G_{\text{restringido}}$) y régimen de costo ($\eta \in \{0, 1\}$).

La Tabla 5.2 muestra los valores de p_s para cada instancia y el número mínimo de ejecuciones N requeridas para alcanzar la probabilidad objetivo $p_d = 0{,}99$.

Efecto del espacio de navegación

En $k = 6$ y $k = 7$, el espacio restringido $G_{\text{restringido}}$ eleva p_s de SA en la mayoría de los casos, porque eliminar aristas acota el espacio de configuraciones y facilita la convergencia. En $k = 5$, el efecto varía según el algoritmo: para SA, $G_{\text{restringido}}$ reduce ligeramente p_s en $\eta = 0$ (0.90 frente a 0.94), mientras que para SQA lo sube de forma notoria (0.91 frente a 0.60). En $k = 7$, $G_{\text{restringido}}$ favorece claramente a SA ($\eta = 0$: 0.44 frente a 0.25) y también, de forma modesta, a SQA (0.08 frente a 0.06 para $\eta = 0$; 0.10 frente a 0.06 para $\eta = 1$), aunque la ganancia absoluta es pequeña dado el deterioro general de SQA en esta escala.

Tabla 5.2: Probabilidad de éxito p_s y número mínimo de ejecuciones N requeridas para alcanzar $p_d = 0{,}99$.

k	Instancia	η	p_s^{SA}	p_s^{SQA}	N^{SA}	N^{SQA}
5	G_{abierto}	0	0.94	0.60	2	6
		1	0.94	0.65	2	5
	$G_{\mathrm{restringido}}$	0	0.90	0.91	2	2
		1	0.95	0.86	2	3
6	G_{abierto}	0	0.66	0.24	5	17
		1	0.74	0.30	4	13
	$G_{\mathrm{restringido}}$	0	0.74	0.33	4	12
		1	0.71	0.31	4	13
7	G_{abierto}	0	0.25	0.06	17	75
		1	0.37	0.06	10	75
	$G_{\mathrm{restringido}}$	0	0.44	0.08	8	56
		1	0.35	0.10	11	44

Efecto de η

El régimen de costo ($\eta = 1$) no tiene un efecto uniforme sobre p_s. En algunos casos mejora la probabilidad de éxito de SA (por ejemplo, $G_{\mathrm{restringido}}$, $k = 5$: 0.95 frente a 0.90; G_{abierto}, $k = 7$: 0.37 frente a 0.25), mientras que en otros la deja igual o la reduce ligeramente. En SQA el efecto es igualmente variable y de menor magnitud. La penalización por ascenso diferencia energéticamente las rutas según su ganancia neta de altitud, lo que reduce la degeneración del óptimo en ciertos paisajes, pero ese efecto se pierde ante la dificultad combinatoria cuando el espacio de búsqueda supera las 10^3 variables.

SA resulta más robusto que SQA al aumentar la escala, con una caída de p_s mucho más contenida. Esto se traduce en menos ejecuciones para alcanzar la solución óptima con alta confianza. La siguiente sección combina estos resultados con los tiempos de ejecución para medir el desempeño global con *Time to Solution*.

5.3. Comparación global mediante *Time to Solution*

Los tiempos de ejecución por corrida t_{run} son similares entre SA y SQA en todas las escalas. En $k = 5$, las diferencias relativas son menores al 3 % para $\eta = 1$ en ambos espacios de navegación, aunque llegan al 12 % en G_{abierto}, $\eta = 0$ (0.469 s frente a 0.524 s). En $k = 6$ y $k = 7$ los tiempos son casi iguales entre algoritmos, con diferencias generalmente por debajo del 7 %. Como ambos usan el mismo presupuesto $B = 1{,}600$ evaluaciones, las diferencias en TTS reflejan directamente las de p_s.

Tabla 5.3: Resumen de *Time to Solution* (TTS) para SA y SQA en las doce instancias ($p_d = 0{,}99$). El Factor es la razón $\mathrm{TTS}_{\mathrm{SQA}}/\mathrm{TTS}_{\mathrm{SA}}$.

k	Instancia	η	$t_{\mathrm{run}}^{\mathrm{SA}}$ (s)	$t_{\mathrm{run}}^{\mathrm{SQA}}$ (s)	$\mathrm{TTS}_{\mathrm{SA}}$ (s)	$\mathrm{TTS}_{\mathrm{SQA}}$ (s)	Factor
5	G_{abierto}	0	0.469	0.524	0.768	2.635	3.43
		1	0.278	0.286	0.456	1.255	2.75
	$G_{\mathrm{restringido}}$	0	0.358	0.325	0.717	0.622	0.87
		1	0.194	0.197	0.299	0.461	1.54
6	G_{abierto}	0	0.702	0.716	2.995	12.016	4.01
		1	0.708	0.725	2.420	9.363	3.87
	$G_{\mathrm{restringido}}$	0	0.501	0.497	1.712	5.711	3.34
		1	0.501	0.499	1.863	6.195	3.33
7	G_{abierto}	0	1.572	1.597	25.169	118.870	4.72
		1	1.702	1.592	16.960	118.512	6.99
	$G_{\mathrm{restringido}}$	0	0.958	0.956	7.608	52.819	6.94
		1	0.961	0.957	10.273	41.817	4.07

La razón $\mathrm{TTS}_{\mathrm{SQA}}/\mathrm{TTS}_{\mathrm{SA}}$ (Tabla 5.3) crece en general con la escala: entre 0.87 y 3.43 en $k = 5$, entre 3.33 y 4.01 en $k = 6$, y entre 4.07 y 6.99 en $k = 7$. El único caso donde SQA supera a SA en TTS es $G_{\mathrm{restringido}}$, $\eta = 0$, $k = 5$ (Factor = 0,87), que coincide con la instancia donde ambos tienen probabilidades de éxito casi idénticas. La Figura 5.3 presenta esta tendencia en escala logarítmica.

La Figura 5.4 presenta las rutas óptimas encontradas por SA en la instancia $k = 5$, $\eta = 1$, para ambos espacios de navegación. En G_{abierto} (izquierda), la ruta va en diagonal directa desde el origen $s = (1, 1, 1)$ hasta el destino $t = (5, 5, 5)$. En $G_{\mathrm{restringido}}$ (derecha), los obstáculos volumétricos obligan a rodearlos manteniendo el costo mínimo. Para $k = 6$ y $k = 7$, las rutas óptimas siguen la misma lógica: trayectoria directa en espacio abierto y rodeo de obstáculos en espacio restringido. Ambos se diferencian en el número de vértices intermedios y la longitud de los segmentos.

La métrica TTS captura dos dimensiones del desempeño, tanto velocidad por ejecución como probabilidad de éxito. Salvo en la instancia más pequeña con espacio restringido, SA necesita menos tiempo que SQA para garantizar el óptimo con 99 % de confianza. La ventaja es notable en $k = 7$, donde el TTS de SA es hasta siete veces menor. La siguiente sección discute estos hallazgos y sus limitaciones.

5.4. Discusión

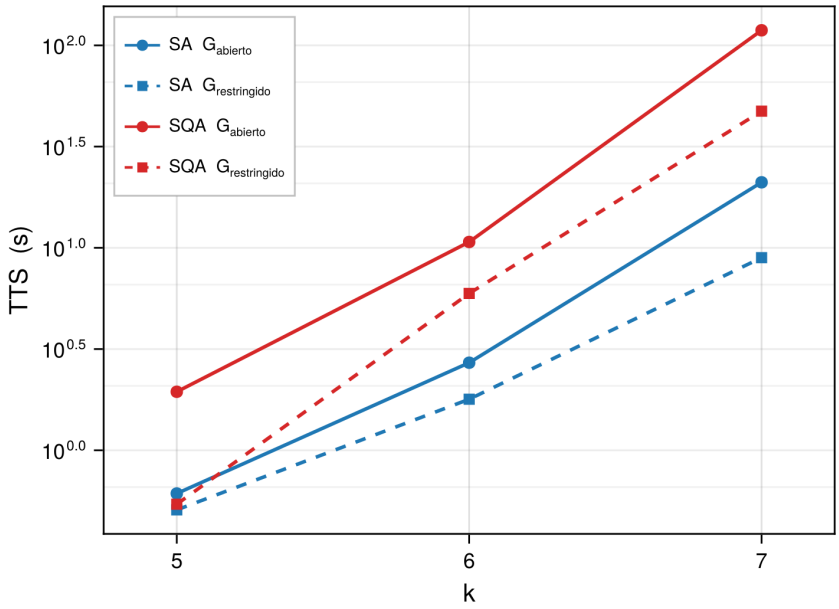
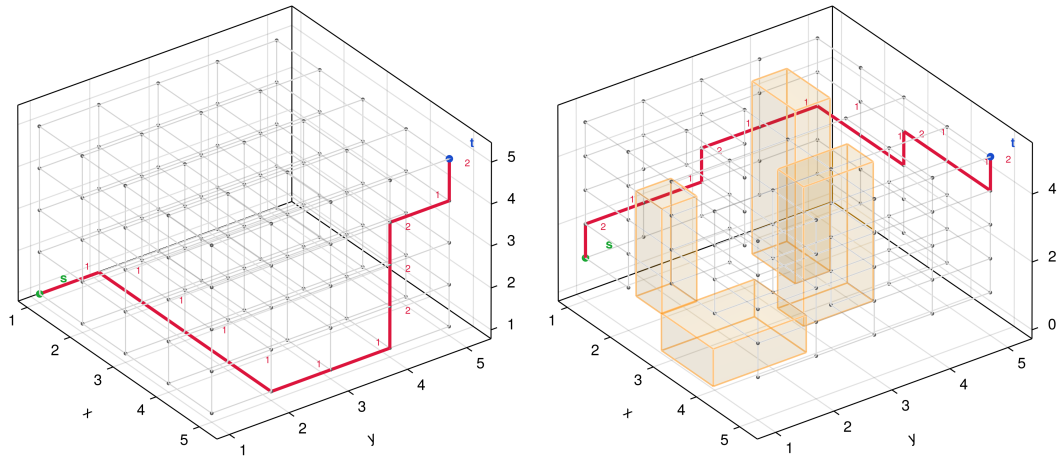


Figura 5.3: TTS en función de k para SA y SQA en escala logarítmica. Las curvas se muestran para G_{abierto} y $G_{\text{restringido}}$, promediadas sobre $\eta \in \{0, 1\}$. El rango de valores abarca de 0.30 s a 118.87 s.



(a) (b)

Figura 5.4: Rutas óptimas para $k = 5$, $\eta = 1$. a) espacio abierto G_{abierto}; b) espacio restringido $G_{\text{restringido}}$ con obstáculos. El vértice fuente s se indica en verde y el destino t en azul.

Ventaja de SA

SA logra un TTS menor en once de las doce instancias, y la diferencia con SQA se extiende con la escala. El factor más alto es 6.99 en $k = 7$, G_{abierto}, $\eta = 1$. La excepción es $k = 5$, $G_{\text{restringido}}$, $\eta = 0$, donde ambos tienen probabilidades de éxito casi iguales y SQA es más rápido por su tiempo de corrida ligeramente menor. Como t_{run} es comparable en todas las instancias, la ventaja de SA es algorítmica: con $k_{\text{máx}} = 1{,}600$ evaluaciones secuenciales y enfriamiento geométrico, SA forma una trayectoria de búsqueda más coherente que el anillo de Trotter de SQA bajo el mismo presupuesto.

SQA a gran escala

La caída de p_s en SQA es mayor: pasa de [0,60, 0,91] en $k = 5$ a [0,06, 0,10] en $k = 7$, una pérdida de más de un orden de magnitud. Con las réplicas y barridos definidos, el anillo de Trotter no logra coordinar la diversidad de copias en un espacio de $n \in [980, 1225]$ variables. SA, al operar secuencialmente, evita este problema de coordinación y se degrada mucho menos en el mismo rango de escala.

Efecto del espacio de navegación

Los obstáculos volumétricos elevan p_s de SA en $k = 6$ y $k = 7$, y benefician a SQA de forma notable en $k = 5$, donde la reducción del espacio factible acerca su desempeño al de SA. A mayor escala, las ganancias absolutas para SQA son pequeñas porque el problema sigue siendo grande incluso con obstáculos: la dimensión QUBO en $G_{\text{restringido}}$, $k = 7$ es $n = 980$ variables, un espacio exigente con el presupuesto disponible.

Efecto de η

La penalización por ascenso no tiene un efecto uniforme sobre p_s. En algunos casos mejora la probabilidad de éxito de SA (por ejemplo, $G_{\text{restringido}}$, $k = 5$: 0.95 frente a 0.90; G_{abierto}, $k = 7$: 0.37 frente a 0.25) y en otros no tiene impacto claro o la reduce ligeramente. El efecto de η depende de la estructura del paisaje de energía de cada instancia, y no se puede generalizar sin considerar el espacio de navegación y la escala.

5.4.1. Limitaciones

Los resultados dependen del operador simple bit-flip empleado, el cual selecciona una variable binaria al azar en cada paso e invierte su estado sobre el espacio de configuraciones. Como el mapeo QUBO penaliza las configuraciones infactibles mediante λ, el bit-flip puede generar configuraciones inválidas durante la búsqueda, y la convergencia depende de que la penalización baste para descartarlas. Esta decisión simplifica la implementación pero muestra a los algoritmos a un paisaje energético con dos escalas (costo y penalización), lo que puede afectar la diversidad del anillo de Trotter en SQA.

Por lo tanto, los resultados reflejan el comportamiento de los algoritmos bajo este operador y el presupuesto fijo $B = 1{,}600$, el cual no deberían generalizarse directamente a otras configuraciones. También, la evaluación emplea tres escalas con dimensiones de $n = 342$ a $n = 1225$, aunque esto deja ver tendencias de escalabilidad, extrapolar a dimensiones mayores requeriría más experimentos, sobre todo para determinar con qué presupuesto SQA podría mostrar competitividad.

6. Conclusiones y trabajo futuro

Este trabajo desarrolló un algoritmo de recocido cuántico simulado (SQA) para planificar rutas en espacios tridimensionales discretos, aplicado al problema del camino más corto (3D-SPSP) de un UAV. El problema se formuló como un modelo QUBO que reúne en una sola función objetivo cuadrática tanto el costo de la trayectoria como las restricciones estructurales de continuidad, ausencia de ciclos y ausencia de ramificaciones. SQA y SA se emplearon sobre esa misma representación, permitiendo una comparacion en condiciones equivalentes, ambos métodos convergieron al óptimo teórico en todas las instancias.

Con un presupuesto fijo de evaluaciones, SA mostró un comportamiento estable, su probabilidad de éxito baja de forma controlada incluso cuando el espacio de soluciones supera las mil variables binarias. SQA, en cambio, reproduce la dinámica del recocido cuántico mediante integral de camino y su probabilidad de éxito cae de forma abrupta al crecer la escala, porque el anillo de Trotter pierde capacidad para coordinar sus réplicas. El enfoque cuántico no queda invalidado por esto, pero los resultados acotan el rango de escalas en el que resulta competitivo con los recursos actuales.

La métrica *Time to Solution* (TTS) combina el tiempo de ejecución con la probabilidad de éxito y permitió comparar el desempeño global. SA superó a SQA en casi todas las instancias y la diferencia crece con la dimensión del problema. La única excepción fue la instancia más pequeña con espacio restringido, donde las réplicas cuánticas alcanzaron un rendimiento comparable al clásico.

De los resultados se extraen dos observaciones. Primera, reducir el espacio de búsqueda (con obstáculos o restricciones) beneficia más al algoritmo cuántico en instancias pequeñas, lo que sugiere que diseñar operadores de perturbación o estrategias de preprocesamiento que reduzcan la dimensión del modelo QUBO que puede ser más efectivo que aumentar el presupuesto de evaluaciones. Segunda, la formulación QUBO del 3D-SPSP funciona como banco de pruebas para futuros algoritmos, al permitir evaluar avances en hardware o simulaciones sobre las mismas instancias.

6.1. Trabajo futuro

Varias líneas de investigación extenderían los hallazgos de este trabajo:

- **Ajuste de hiperparámetros y operadores de perturbación.** El esquema geométrico usado (temperatura en SA, campo transverso en SQA) puede optimizarse con búsqueda sistemática o métodos adaptativos. Ademas, explorar operadores de perturbación que atraviesen configuraciones infactibles de forma controlada podría ampliar la exploración, sobre todo en SQA.

- **Escalabilidad a dimensiones mayores.** Extender el estudio a retículas con $k > 7$ permitiría determinar si la diferencia entre SA y SQA continúa creciendo o si existe un régimen donde el comportamiento cuántico recupere competitividad.

- **Entornos dinámicos y multi-UAV.** Incorporar obstáculos móviles, zonas de exclusión o múltiples vehículos con restricciones de separación podria dar lugar a formulaciones QUBO más complejas, donde las diferencias entre los enfoques podrían mostrarse de forma distinta.

- **Reducción del modelo mediante simetrías.** Las simetrías de la retícula (rotaciones, reflexiones, traslaciones) pueden aprovecharse para reducir la dimensión efectiva del problema QUBO antes de la optimización, lo que podría mejorar el desempeño de ambos métodos, especialmente en SQA.

- **Emplear otros métodos de optimización.** Evaluar metaheurísticas clásicas adicionales como búsqueda tabú o algoritmos genéticos y a su vez algoritmos cuánticos variacionales como QAOA sobre la misma formulación QUBO permitiría poner los resultados de SA y SQA en un contexto más amplio, comparando distintos enfoques en igualdad de condiciones.

- **Incorporación de incertidumbre en los costos.** Extender el modelo a costos no deterministas (viento, consumo variable, mediciones ruidosas) a partir de formulaciones estocásticas o robustas generaría instancias QUBO distintas, y aumentaría la relevancia práctica de la metodología.

Bibliografía

Abbas, A., Ambainis, A., Augustino, B., Bärtschi, A., Buhrman, H., Coffrin, C., Cortiana, G., Dunjko, V., Egger, D. J., Elmegreen, B. G., et al. (2024). Challenges and opportunities in quantum optimization. *Nature Reviews Physics*, pages 1–18.

Aghaee Rad, H. et al. (2025). Scaling and networking a modular photonic quantum computer. *Nature*. Aurora: First scalable, networked photonic quantum computer with 12 qubits, 35 photonic chips, and 13 km of fiber optics.

Aharonov, D., Van Dam, W., Kempe, J., Landau, Z., Lloyd, S., and Regev, O. (2008). Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Review*, 50(4):755–787.

Ahmadi, S., Raith, A., and Jalili, M. (2025). Resource constrained pathfinding with a* and negative weights.

Ahmed, S., Mohamed, A., Harras, K., Kholief, M., and Mesbah, S. (2016). Energy efficient path planning techniques for uav-based systems with space discretization. In *IEEE Wireless Communications and Networking Conference*.

Al Zoobi, A., Coudert, D., and Nisse, N. (2023). Finding the k shortest simple paths: Time and space trade-offs. *ACM Journal of Experimental Algorithmics*, 28:1–23.

Albash, T. and Lidar, D. A. (2018). Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1):015002.

Ali, M., Ahmed, H., Malik, M., et al. (2024). Multicommodity information flow through quantum annealer. *Quantum Information Processing*, 23:313.

Ali, Z. A. et al. (2020). Maximum-minimum ant colony optimization for uav path planning. *Journal of Intelligent & Fuzzy Systems*, 39:163–176.

Altshuler, B., Krovi, H., and Roland, J. (2010). Anderson localization casts clouds over adiabatic quantum optimization. *Proceedings of the National Academy of Sciences*, 107(28):12446–12450.

Anthony, M., Boros, E., Crama, Y., and Gruber, A. (2017). Quadratic reformulations of nonlinear binary optimization problems. *Mathematical Programming*, 162:115–144.

Apolloni, B., Carvalho, C., and De Falco, D. (1989). Quantum stochastic optimization. *Stochastic Processes and their Applications*, 33(2):233–244.

Apolloni, B., Cesa-Bianchi, N., and Falco, D. d. (1988). A numerical implementation of "quantum annealing". In *Stochastic Processes, Physics and Geometry: Proceedings of the Ascona-Locarno Conference*, pages 97–111.

Arora, S. and Barak, B. (2009). *Computational complexity: a modern approach*. Cambridge University Press.

Arshad, J. et al. (2025). Uav path planning and trajectory optimization: A comprehensive survey. *Arabian Journal for Science and Engineering*.

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G., Buell, D. A., et al. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510.

Ashvinkumar, V., Bernstein, A., Cao, N., Grunau, C., Haeupler, B., Jiang, Y., Nanongkai, D., and Su, H.-H. (2024). Parallel, distributed, and quantum exact single-source shortest paths with negative edge weights. In *32nd Annual European Symposium on Algorithms (ESA 2024)*, volume 308 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:15. Dagstuhl–Leibniz-Zentrum für Informatik.

Baek, J., Han, S., and Han, Y. (2019). Energy-efficient uav routing for wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 69:1741–1750.

Ball, P. (2025). Experts weigh in on microsoft's topological qubit claim. *Physics World*. Analysis of controversy surrounding Microsoft's topological qubit announcement.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329.

Bell, J. S. (1964). On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1(3):195.

Belovs, A. (2013). Quantum walks and electric networks. *arXiv preprint arXiv:1302.3143*.

Belovs, A. and Reichardt, B. W. (2012). Span programs and quantum algorithms for st-connectivity and claw detection. In *European Symposium on Algorithms*, pages 193–204. Springer.

Benioff, P. (1980). The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of statistical physics*, 22:563–591.

Benioff, P. (1982a). Quantum mechanical hamiltonian models of turing machines. *Journal of Statistical Physics*, 29:515–546.

Benioff, P. (1982b). Quantum mechanical models of turing machines that dissipate no energy. *Physical Review Letters*, 48(23):1581.

Benioff, P. A. (1982c). Quantum mechanical hamiltonian models of discrete processes that erase their own histories: Application to turing machines. *International Journal of Theoretical Physics*, 21:177–201.

Bennett, C. H., Bernstein, E., Brassard, G., and Vazirani, U. (1997). Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523.

Bennett, C. H., Brassard, G., Crépeau, C., Jozsa, R., Peres, A., and Wootters, W. K. (1993). Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Physical review letters*, 70(13):1895.

Bernstein, E. and Vazirani, U. (1993). Quantum complexity theory. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 11–20.

Berry, D. W. (2014). High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical*, 47(10):105301.

Bertsimas, D. and Tsitsiklis, J. N. (1997). *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA. Explica formalmente la geometría de la optimización y la no unicidad de soluciones (degeneración).

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98.

Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308. Revisión exhaustiva sobre por qué se usan metaheurísticas en paisajes no convexos.

Born, M. and Fock, V. (1928). Beweis des adiabatensatzes. *Zeitschrift für Physik*, 51(3):165–180.

Boros, E. and Hammer, P. L. (2002). Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123:155–225.

Bortoff, S. A. (2000). Path planning for uavs. In *Proceedings of the American Control Conference*, pages 364–368.

Bouchmal, O., Cimolil, B., Stabile, R., Olmos, J. J. V., and Monroy, I. T. (2025). Quantum approximate optimization algorithm applied to multi-objective routing for large scale 6g networks. *Computer Networks*, page 111345.

Bouwmeester, D., Pan, J.-W., Mattle, K., Eibl, M., Weinfurter, H., and Zeilinger, A. (2000). Experimental quantum teleportation. *Nature*, 390:575–579.

Budiyanto, A., Cahyadi, A., Adji, T. B., and Wahyunggoro, O. (2015). Uav obstacle avoidance using potential field under dynamic environment. In *International Conference on Control, Electronics, Renewable Energy and Communications*, pages 187–192.

Cabrera, N., Medaglia, A. L., Lozano, L., and Duque, D. (2020). An exact bidirectional pulse algorithm for the constrained shortest path. *Networks*, 76(2):128–146.

Carbone, A., De Grossi, F., and Spiller, D. (2023). Cutting-edge trajectory optimization through quantum annealing. *Applied Sciences*, 13(23):12853.

Cekmez, U., Ozsiginan, M., and Sahingoz, O. K. (2016). Multi colony ant optimization for uav path planning with obstacle avoidance. In *International Conference on Unmanned Aircraft Systems*, pages 47–52.

Cerezo, M., Verdon, G., Huang, H.-Y., Cincio, L., and Coles, P. J. (2022). Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576.

Chen, X., Li, G., and Chen, X. (2017). Path planning and cooperative control for multiple uavs based on consistency theory and voronoi diagram. In *Chinese Control and Decision Conference*, pages 881–886.

Chiew, S. et al. (2023). Multi-objective optimization and network routing with near-term quantum computers. *arXiv preprint arXiv:2308.08245*.

chih Yao, A. C. (1975). An 0(—e—loglog—v—) algorithm for finding minimum spanning trees. *Information Processing Letters*, 4(1):21–23.

Chuang, I. L., Gershenfeld, N., and Kubinec, M. (1998). Bulk quantum computation with nuclear magnetic resonance. *Physical Review Letters*, 80:3408.

Coffman, V., Kundu, J., and Wootters, W. K. (2000). Distributed entanglement. *Physical Review A*, 61(5):052306.

Coppersmith, D. (2002). An approximate fourier transform useful in quantum factoring. *arXiv preprint quant-ph/0201067*.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2022). *Introduction to algorithms*. MIT press.

Crosson, E. and Harrow, A. W. (2016). Simulated quantum annealing can be exponentially faster than classical simulated annealing. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 714–723. IEEE.

Cui, D., Mehta, A., Mousavi, H., and Nezhadi, S. (2020). A generalization of chsh and the algebraic structure of optimal strategies. *Quantum*, 4:346.

D-Wave Systems Inc. (2024a). D-wave samplers. Biblioteca de samplers clásicos y de recocido cuántico simulado para el kit de desarrollo Ocean.

D-Wave Systems Inc. (2024b). Qubo and ising models — d-wave system documentation.

Daley, A. J., Bloch, I., Kokail, C., Flannigan, S., Pearson, N., Troyer, M., and Zoller, P. (2022). Practical quantum advantage in quantum simulation. *Nature*, 607(7920):667–676.

D'Amato, E., Mattei, M., and Notaro, I. (2019). Bi-level flight path planning of uav formations with collision avoidance. *Journal of Intelligent & Robotic Systems*, 93:193–211.

De Filippis, L., Guglieri, G., and Quagliotti, F. (2012). Path planning strategies for uavs in 3d environments. In *Journal of Intelligent & Robotic Systems*, volume 65, pages 247–264.

Debnath, S., Linke, N. M., Figgatt, C., Landsman, K. A., Wright, K., and Monroe, C. (2016). Demonstration of a small programmable quantum computer with atomic qubits. *Nature*, 536(7614):63–66.

Deutsch, D. (1985). Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117.

Deutsch, D. and Jozsa, R. (1992). Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558.

Dewangan, R. K., Shukla, A., and Godfrey, W. W. (2019). Three dimensional path planning using grey wolf optimizer for uavs. *Applied Intelligence*, 49:2201–2217.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Dirac, P. A. M. (1981). *The principles of quantum mechanics*. Number 27. Oxford university press.

Dong, Z., Chen, Z., Zhou, R., and Zhang, R. (2011). A hybrid approach of virtual force and a* search algorithm for uav path re-planning. In *IEEE Conference on Industrial Electronics and Applications*, pages 1140–1145.

Driscoll, J. R., Gabow, H. N., Shrairman, R., and Tarjan, R. E. (1988). Relaxed heaps: An alternative to fibonacci heaps with applications to parallel computation. *Communications of the ACM*, 31(11):1343–1354.

Du, Z. and Li, H. (2024). Research on application of improved quantum optimization algorithm in path planning. *Applied Sciences*, 14(11):4613.

Duan, R., Lyu, K., Wu, H., and Xie, Y. (2018). Single-source bottleneck path algorithm faster than sorting for sparse graphs.

Duan, R., Mao, J., Mao, X., Shu, X., and Yin, L. (2025). Breaking the sorting barrier for directed single-source shortest paths. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC)*, pages 36–44, New York, NY, USA. ACM.

Dürr, C., Heiligman, M., HOyer, P., and Mhalla, M. (2006). Quantum query complexity of some graph problems. *SIAM Journal on Computing*, 35(6):1310–1328.

Einstein, A., Podolsky, B., and Rosen, N. (1935). Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10):777.

Eppstein, D. (1998). Finding the k shortest paths. *SIAM Journal on Computing*, 28(2):652–673.

Farhi, E., Goldstone, J., Gutmann, S., and Sipser, M. (2000a). Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*.

Farhi, E., Goldstone, J., Gutmann, S., and Sipser, M. (2000b). Quantum computation by adiabatic evolution.

Feynman, R. P. (1982). Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488.

Fisher, M. L. (2004). The lagrangian relaxation method for solving integer programming problems. *Management Science*, 50(12_supplement):1861–1871.

Fredman, M. L. and Tarjan, R. E. (1987). Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615.

Fu, Z., Yu, J., Xie, G., Chen, Y., and Mao, Y. (2018). A heuristic evolutionary algorithm of uav path planning. *Wireless Communications and Mobile Computing*.

Gabow, H. N. and Tarjan, R. E. (1988). Algorithms for two bottleneck optimization problems. *Journal of Algorithms*, 9(3):411–417.

Gal, T. (1995). *Postoptimal Analyses, Parametric Programming, and Related Topics*. Walter de Gruyter, Berlin, 2 edition. Obra de referencia sobre el análisis de la degeneración en programación matemática.

Garg, K., Choset, H., and Pavone, M. (2025). Graphs of convex sets with applications to optimal path planning. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Girija, A. S. and Ashok, S. (2019). Pso-apf based path planning of multiple uavs in dynamic environment. *International Journal of Engineering and Advanced Technology*, 8:2277–2282.

Glover, F., Kochenberger, G., and Du, Y. (2019a). Quantum bridge analytics i: a tutorial on formulating and using qubo models. *4OR*, 17:335–371.

Glover, F., Kochenberger, G., and Du, Y. (2019b). A tutorial on formulating and using qubo models. *arXiv preprint arXiv:1811.11538*.

Google Quantum AI (2024). Quantum error correction below the surface code threshold. *Nature*. Google Willow: 105-qubit processor demonstrating exponential error reduction.

Grilo, A. B., Malavolta, G., Walter, M., and Zhang, T. (2025). Computational monogamy of entanglement and non-interactive quantum key distribution. *arXiv preprint arXiv:2510.00791*.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219.

Hammer, P. L. and Rudeanu, S. (1968a). *Boolean Methods in Operations Research and Related Areas*. Springer, Berlin.

Hammer, P. L. and Rudeanu, S. (1968b). *Boolean Methods in Operations Research and Related Areas*. Springer, Berlin.

Handler, G. Y. and Zang, I. (1980). A dual algorithm for the constrained shortest path problem. *Networks*, 10(4):293–309.

Hansen, P., Jaumard, B., and Mathon, V. (1993). State-of-the-art survey—constrained nonlinear 0–1 programming. *ORSA Journal on Computing*, 5:97–119.

Harrow, A. W., Hassidim, A., and Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

He, H. et al. (2024). Quantum annealing and graph neural networks for solving tsp with qubo. *arXiv preprint arXiv:2402.14036*.

Himmich, I., El Hallaoui, I., and Soumis, F. (2024). A multiphase dynamic programming algorithm for the shortest path problem with resource constraints. *European Journal of Operational Research*, 315(2):470–483.

Horodecki, R., Horodecki, P., Horodecki, M., and Horodecki, K. (2009). Quantum entanglement. *Reviews of modern physics*, 81(2):865–942.

Hua, R. and Dinneen, M. J. (2020). Improved qubo formulation of the graph isomorphism problem. *SN Computer Science*, 1:1–18.

IBM (2024). Ibm launches its most advanced quantum computers, fueling new scientific value and progress towards quantum advantage. *IBM Newsroom*.

Innan, N., Kashif, M., Marchisio, A., Gan, Y.-S., Barbaresco, F., and Shafique, M. (2025). Quav: Quantum-assisted path planning and optimization for uav navigation with obstacle avoidance. *arXiv preprint arXiv:2508.21361*.

IonQ (2025a). Ionq announces innovations in compact, room-temperature quantum computing through novel extreme high vacuum (xhv) technology.

IonQ (2025b). Ionq's 2025 roadmap: Toward a cryptographically relevant quantum computer by 2028. Roadmap projecting 20,000 physical qubits (1,600 logical qubits) by 2028 and 2,000,000 physical qubits by 2030.

Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*, pages 33–65. Springer, New York.

Jansen, S., Ruskai, M.-B., and Seiler, R. (2007). Bounds for the adiabatic approximation with applications to quantum computation. *Journal of Mathematical Physics*, 48(10):102111.

Jeffery, S., Kimmel, S., and Piedrafita, A. (2023). Quantum algorithm for path-edge sampling. *arXiv preprint arXiv:2303.03319*.

Johnson, M. W., Amin, M. H., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A. J., Johansson, J., Bunyk, P., et al. (2011). Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198.

Johnson, M. W. e. a. (2011). Quantum annealing with manufactured spins. *Nature*, 473:194–198.

Kadowaki, T. and Nishimori, H. (1998). Quantum annealing in the transverse ising model. *Physical Review E*, 58(5):5355.

Karp, R. M. (2010). *Reducibility among combinatorial problems*. Springer.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5:90–98.

Kim, Y.-D., Gu, D.-W., and Postlethwaite, I. (2008). Real-time path planning with limited information for autonomous unmanned air vehicles. *Automatica*, 44:696–712.

King, A. D., Raymond, J., Lanting, T., Harris, R., Zucca, A., Altomare, F., Berkley, A. J., Boothby, K., Brazier, S., Bunyk, P., et al. (2021). Scaling advantage over path-integral Monte Carlo in quantum simulation of geometrically frustrated magnets. *Nature Communications*, 12(1):1113.

Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Koashi, M. and Winter, A. (2004). Monogamy of quantum entanglement and other correlations. *Physical Review A*, 69:022309.

Kochenberger, G., Glover, F., Alidaee, B., Lewis, K., Lu, Z., Wang, H., and Wang, Y. (2014). The unconstrained binary quadratic programming problem: a survey. *Journal of Combinatorial Optimization*, 28:58–81.

Kochenberger, G., Glover, F., Alidaee, B., and Rego, C. (2005). An unconstrained quadratic binary programming approach to the vertex coloring problem. *Annals of Operations Research*, 139:229–241.

Koenig, S. and Likhachev, M. (2002). D* lite. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI)*, pages 476–483.

Koenig, S., Likhachev, M., and Furcy, D. (2004). Lifelong planning a*. *Artificial Intelligence*, 155(1–2):93–146.

Koritsoglou, K., Tsoumanis, G., Patras, V., and Fudos, I. (2022a). Shortest path algorithms for pedestrian navigation systems. *Information*, 13(6).

Koritsoglou, K., Tsoumanis, G., Patras, V., and Fudos, I. (2022b). Shortest path algorithms for pedestrian navigation systems. *Information*, 13(6).

Korte, B. and Vygen, J. (2012). *Combinatorial Optimization: Theory and Algorithms*. Springer, Berlin, Heidelberg, 5 edition. Texto fundamental sobre algoritmos y estructuras discretas.

Kothari, M. and Postlethwaite, I. (2013). A probabilistically robust path planning algorithm for uavs using rapidly-exploring random trees. In *Journal of Intelligent & Robotic Systems*, volume 71, pages 231–253.

Krauss, T. and McCollum, J. (2020). Solving the network shortest path problem on a quantum annealer. *IEEE Transactions on Quantum Engineering*, 1:1–12.

Krauss, T., McCollum, J., Pendery, C., Litwin, S., and Michaels, A. J. (2020). Solving the max-flow problem on a quantum annealing computer. *IEEE Transactions on Quantum Engineering*, 1:1–10.

Kumari, S. and Srirangarajan, S. (2025). Multi-uav path planning for connectivity-based sweep coverage. *Ad Hoc Networks*, 178:103966.

Land, A. H. and Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520.

LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, Cambridge. La referencia estándar para planificación de movimiento y espacios de configuración.

Liu, J. (2025). Quantum grid path planning using parallel qaoa circuits based on minimum energy principle. *arXiv preprint arXiv:2510.07413*.

Liu, J., Gu, H., Wei, W., Chen, Z., and Chen, Y. (2021). An efficient shortest path algorithm for content-based routing on 2-d mesh accelerator networks. *Future Generation Computer Systems*, 114:519–530.

Liu, W., Zheng, Z., and Cai, K.-Y. (2013). Bi-level programming based real-time path planning for unmanned aerial vehicles. *Knowledge-Based Systems*, 44:34–47.

Liu, Y., Zhang, X., Guan, X., and Delahaye, D. (2016). Adaptive sensitivity decision based path planning algorithm for unmanned aerial vehicle with improved particle swarm optimization. *Aerospace Science and Technology*, 58:92–102.

Lloyd, S., De Palma, G., Gokler, C., Kiani, B., Liu, Z.-W., Marvian, M., Tennie, F., and Palmer, T. (2020). Quantum algorithm for nonlinear differential equations. *arXiv preprint arXiv:2011.06571*.

Lozano, L. and Medaglia, A. L. (2013). On an exact method for the constrained shortest path problem. *Computers & Operations Research*, 40(1):378–384.

Lu, C. and ... (2020). Satellite-based entanglement distribution over 1200 kilometers. *Science*, 356:1140–1144.

Lucas, A. (2014). Ising formulations of many np problems. *Frontiers in Physics*, 2:5.

Luo, Y. et al. (2018). Deep-sarsa based multi-uav path planning in dynamic environment. In *Chinese Control Conference*, pages 10055–10060.

Lv, Z., Yang, L., He, Y., Liu, Z., and Han, Z. (2017a). 3d environment modeling with height dimension reduction and path planning for uav. In *2017 9th International Conference on Modelling, Identification and Control (ICMIC)*, pages 734–739. IEEE.

Lv, Z., Yang, L., He, Y., Liu, Z., and Han, Z. (2017b). 3d environment modeling with height dimension reduction and path planning for uav. In *2017 9th International Conference on Modelling, Identification and Control (ICMIC)*, pages 734–739. IEEE.

Lyu, D., Chen, Z., Cai, Z., and Piao, S. (2021). Robot path planning by leveraging the graph-encoded floyd algorithm. *Future Generation Computer Systems*, 122:204–208.

Mac, T. T., Copot, C., Hernandez, A., and De Keyser, R. (2016). Improved potential field method for unknown obstacle avoidance using uav in indoor environment. In *IEEE International Symposium on Applied Machine Intelligence and Informatics*, pages 345–350.

Manin, Y. I. (1980). Vychislimoe i nevychislimoe (computable and noncomputable), moscow: Sov.

Marcucci, T., Petersen, M., von Stryk, O., and Tedrake, R. (2024). Shortest paths in graphs of convex sets. *SIAM Journal on Optimization*, 34(1):507–532.

Martoňák, R., Santoro, G. E., and Tosatti, E. (2002). Quantum annealing by the path-integral Monte Carlo method: The two-dimensional random Ising model. *Physical Review B*, 66(9):094203.

McCoy, B. M. and Wu, T. T. (1973). *The two-dimensional Ising model*. Harvard University Press.

McGeoch, C. C. (2014). *Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice*. Morgan & Claypool.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Microsoft Azure Quantum (2025). Majorana 1: World's first quantum processor powered by topological qubits. *Nature*. First topological quantum processor with 8 qubits on chip designed to scale to 1 million qubits.

Mohammed, M. Q., Meeß, H., and Otte, M. (2025). Review of the application of quantum annealing-related technologies in transportation optimization. *Quantum Information Processing*, 24(9):296.

Mohseni, N., McMahon, P. L., and Byrnes, T. (2022). Ising machines as hardware solvers of combinatorial optimization problems. *Nature Reviews Physics*, 4(6):363–379.

Monbroussou, L. and Kukla, R. (2021). Quantum potential speed-up of combinatorial optimization solvers for auv path planning. In *OCEANS 2021: San Diego–Porto*, pages 1–9. IEEE.

Montañez-Barrera, J. and Michielsen, K. (2025). Toward a linear-ramp QAOA protocol: Evidence of a scaling advantage in solving some combinatorial optimization problems. *npj Quantum Information*, 11:82.

Mooney, G. J., White, G. A., Hill, C. D., and Hollenberg, L. C. (2021). Whole-device entanglement in a 65-qubit superconducting quantum computer. *Advanced Quantum Technologies*, 4(10):2100061.

Morita, S. and Nishimori, H. (2008). Mathematical foundation of quantum annealing. *Journal of Mathematical Physics*, 49(12):125210.

Nielsen, M. A. and Chuang, I. L. (2010). *Quantum computation and quantum information*. Cambridge university press.

Padberg, M. and Rinaldi, G. (1991). A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100.

Papadimitriou, C. H. and Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, New York. Referencia clásica sobre la teoría de optimización combinatoria y complejidad.

Pardalos, P. M. and Jha, S. (1992). Complexity of uniqueness and local search in quadratic 0–1 programming. *Operations research letters*, 11(2):119–123.

Pawłowski, M. (2010). Security proof for cryptographic protocols based only on the monogamy of bell's inequality violations. *Physical Review A*, 82:032313.

Pettie, S. and Ramachandran, V. (2005). A shortest path algorithm for real-weighted undirected graphs. *SIAM Journal on Computing*, 34(6):1398–1431.

Portmann, C. and Renner, R. (2022). Security in quantum cryptography. *Reviews of Modern Physics*, 94(2):025008.

Preskill, J. (2018). Quantum computing in the nisq era and beyond. *Quantum*, 2:79.

Primatesta, S., Guglieri, G., and Rizzo, A. (2019). A risk-aware path planning strategy for uavs in urban environments. *Journal of Intelligent & Robotic Systems*, 95:629–643.

Punnen, A. P. (2022). The quadratic unconstrained binary optimization problem. *Springer International Publishing*, 10:978–3.

Qian, X., Peng, C., Dong, C., and Xiang, Z. (2015). Dynamic obstacle avoidance path planning of uavs. In *Chinese Control Conference*, pages 8860–8865.

Reagor, M., Osborn, C. B., Tezak, N., Staley, A., Prawiroatmodjo, G., Scheer, M., Alidoust, N., Sete, E. A., Didier, N., da Silva, M. P., et al. (2018). Demonstration of universal parametric entangling gates on a multi-qubit lattice. *Science advances*, 4(2):eaao3603.

Reidys, C. M. and Stadler, P. F. (2002). Combinatorial landscapes. *SIAM Review*, 44(1):3–54. Artículo seminal sobre la teoría de paisajes de fitness en optimización.

Research, I. (2016). Ibm quantum experience: A cloud-based quantum computing platform.

Research, I. (2021). Ibm quantum breaks the 100-qubit processor barrier.

Rieffel, E. and Polak, W. (2000). An introduction to quantum computing for non-physicists. *ACM Computing Surveys*, 32(3):300–335.

Roberge, V., Tarbouchi, M., and Labonté, G. (2018). Fast genetic algorithm path planner for fixed-wing military uav using gpu. *IEEE Transactions on Aerospace and Electronic Systems*, 54(5):2105–2117.

Roberge, V., Tarbouchi, M., and Labonté, G. (2012). Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning. *IEEE Transactions on Industrial Informatics*, 9:132–141.

Roland, J. and Cerf, N. J. (2002). Quantum search by local adiabatic evolution. *Physical Review A*, 65(4):042308.

Rønnow, T. F., Wang, Z., Job, J., Boixo, S., Isakov, S. V., Wecker, D., Martinis, J. M., Lidar, D. A., and Troyer, M. (2014). Defining and detecting quantum speedup. *science*, 345(6195):420–424.

Rosa, R. L., Saadi, M., Rodríguez, D. Z., Gew, L. T., Nordin, R., Ali, R. A. R., and Long, C. M. (2025). Integer linear programming for optimizing drone-based delivery routes. *Engineering Journal*, 29(11):23–38.

Rosenberg, I. (1975). Quadratic unconstrained binary optimization model.

Santoro, G. E., Martoňák, R., Tosatti, E., and Car, R. (2002). Theory of quantum annealing of an Ising spin glass. *Science*, 295(5564):2427–2430.

Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee.

Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332.

Simon, D. R. (1997). On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483.

Stentz, A. (1995). The focussed d* algorithm for real-time replanning. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1652–1659.

Stinchcombe, R. (1973). Ising model in a transverse field. i. basic theory. *Journal of Physics C: Solid State Physics*, 6(15):2459.

Stollenwerk, T., O'Gorman, B., Venturelli, D., Mandra, S., Rodionova, O., Ng, H., Sridhar, B., Rieffel, E. G., and Biswas, R. (2019). Quantum annealing applied to de-conflicting optimal trajectories for air traffic management. *IEEE transactions on intelligent transportation systems*, 21(1):285–297.

Suzuki, M. (1976). Generalized trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Communications in Mathematical Physics*, 51(2):183–190.

Tasseff, B., Albash, T., Morrell, Z., Vuffray, M., Lokhov, A. Y., Misra, S., and Coffrin, C. (2024). On the emerging potential of quantum annealing hardware for combinatorial optimization. *Journal of Heuristics*, 30(5):325–358.

Team, S. R. (2023). Quantum computer hardware: Architectures and applications. *SpinQta News*.

Tilk, C., Rothenbächer, D., Gschwind, T., and Irnich, S. (2017). Bidirectional labeling for the resource constrained shortest path problem. *European Journal of Operational Research*, 261(2):415–426.

Toner, B. F. (2009). Monogamy of non-local quantum correlations. *Proceedings of the Royal Society A*, 465:59–69.

Trotter, H. F. (1959). On the product of semi-groups of operators. *Proceedings of the American Mathematical Society*, 10(4):545–551.

Tu, Q., Cheng, L., Yuan, T., Cheng, Y., and Li, M. (2020). The constrained reliable shortest path problem for electric vehicles in the urban transportation network. *Journal of cleaner production*, 261:121130.

Wang, Y. and Liu, J. (2024). A comprehensive review of quantum machine learning: from nisq to fault tolerance. *Reports on Progress in Physics*.

Wang, Z., Liu, L., Long, T., Yu, C., and Kou, J. (2014). Enhanced sparse a* search for uav path planning using dubins path estimation. In *Proceedings of the Chinese Control Conference*, pages 738–742.

Welzl, E. (1985). Constructing the visibility graph for n line segments in o(n^2) time. *Information Processing Letters*, 20:167–171.

Wesołowski, A. and Piddock, S. (2024). Advances in quantum algorithms for the shortest path problem. *arXiv preprint arXiv:2408.10427*. bounded-error quantum algorithms in adjacency list model: $\tilde{O}(l^2\sqrt{m})$ and $\tilde{O}(l\sqrt{m})$.

Williams, V. V. (2010). Nondecreasing paths in a weighted graph or: How to optimally read a train schedule. *ACM Trans. Algorithms*, 6(4).

Wu, H., Chang, H., and Tian, X. (2023). Trajectory planning method for uav inspection of transmission towers based on simulated annealing algorithm. *Journal of Electronics and Information Science*, 8(5):52–57.

Wu, J., Wang, H., Li, N., Peng, Y., and Yu, H. (2018). Path planning for solar-powered uav in urban environment. *Neurocomputing*, 275:2055–2065.

Wu, K. et al. (2017). Real-time three-dimensional smooth path planning for unmanned aerial vehicles in completely unknown cluttered environments. In *IEEE Region 10 Conference*.

WU, X., XIAO, B., WU, C., GUO, Y., and LI, L. (2022). Factor graph based navigation and positioning for control system design: A review. *Chinese Journal of Aeronautics*, 35(5):25–39.

Xanadu and HyperLight (2025). Xanadu and hyperlight unveil advances in photonic chips, setting new quantum computing benchmarks. *The Quantum Insider*. Ultra-low-loss TFLN photonic chips with waveguide losses below 2 dB/m.

Xie, J. et al. (2020). Marer q-learning for uav path planning. *IEEE Access*, 8:122757–122771.

Yakıcı, E. (2016). Solving location and routing problem for uavs. *Computers & Industrial Engineering*, 102:294–301.

Yan, C. et al. (2019). Dueling double deep q networks for uav path planning. *Chinese Control and Decision Conference*, pages 2292–2297.

Yan, C. and Xiang, X. (2018). An improved q-learning algorithm for path planning of uav. In *Chinese Control and Decision Conference*, pages 1526–1530.

Yang, K., Gan, S. K., and Sukkarieh, S. (2013). A gaussian process-based rrt planner for the exploration of an unknown and cluttered environment with a uav. *Advanced Robotics*, 27:431–443.

Yen, J. Y. (1971). Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716.

Yi, H.-C., You, Z.-H., Huang, D.-S., and Kwoh, C. K. (2022). Graph representation learning in bioinformatics: trends, methods and applications. *Briefings in Bioinformatics*, 23(1):bbab340.

Yu, N. and Ying, M. (2018). Exact simulation of quantum adiabatic evolution by the schrödinger equation. *Quantum Information Processing*, 17(4):1–16.

Zeng, L., Zhang, J., and Sarovar, M. (2016). Schedule path optimization for adiabatic quantum computing and optimization. *Journal of Physics A: Mathematical and Theoretical*, 49(16):165305.

Zhang, C. et al. (2014). Cooperative geometric learning algorithm for multi-uav path planning. *Journal of Intelligent & Robotic Systems*, 76:123–139.

Zhang, C., Huang, N., and Wu, C. (2025). Autonomous coverage path planning model for maritime search and rescue with uav application. *Journal of Marine Science and Engineering*, 13(9).

Zielewski, M. R. and Takizawa, H. (2022). A method for reducing time-to-solution in quantum annealing through pausing. In *International Conference on High Performance Computing in Asia-Pacific Region*, HPCAsia '22, page 137–145, New York, NY, USA. Association for Computing Machinery.

Zurek, W. H. (2003). Decoherence and the transition from quantum to classical—revisited. In *Quantum Decoherence: Poincaré Seminar 2005*, pages 1–31. Springer.