



**Universidad**  
Zaragoza

# Computación Cuántica Adiabática para Procesos de Optimización en Finanzas

**Autor:** Guillermo Burges Bruna

**Director:** David González Rojas

**Ponente:** David Zueco Lainez

Facultad de Ciencias

Curso 2024/2025

11 de julio de 2025



# Índice

|  |           |
|--|-----------|
| <b>1. Introducción</b>   | <b>1</b>  |
| <b>2. Marco Teórico</b>  | <b>2</b>  |
| 2.1. Optimización de carteras: Fundamentos teóricos . . . . .                                | 2         |
| 2.2. Formulación computacional: Del modelo financiero al QUBO . . . . .                      | 2         |
| 2.3. Sampler de <i>simulated annealing</i> . . . . .   | 4         |
| 2.4. Computación cuántica adiabática . . . . .   | 5         |
| 2.5. Quantum annealing . . . . .   | 6         |
| 2.6. Modelo de Ising . . . . .   | 7         |
| 2.7. Ordenador cuántico de D-Wave . . . . .  | 8         |
| <b>3. Análisis experimental</b>  | <b>11</b> |
| 3.1. Introducción al problema. Base de datos de <i>Yahoo Finance</i> . . . . .               | 11        |
| 3.2. Generación de carteras de inversión . . . . .   | 12        |
| 3.3. Resolución del problema con <i>simulated annealing</i> . Frontera eficiente . . . . .   | 13        |
| 3.4. Complejidad de una cartera. Exact Solver . . . . .                                      | 16        |
| 3.5. Implementación experimental del algoritmo de <i>quantum annealing</i> . . . . .         | 18        |
| 3.6. Estudio del <i>gap</i> de energías. Cambio de escala de evolución lineal a logarítmica  | 19        |
| 3.7. Complejidad de una cartera en el <i>annealing</i> cuántico . . . . .                    | 21        |
| <b>4. Conclusión. Posibles ampliaciones del modelo</b>                                       | <b>23</b> |
| <b>Anexo</b>   | <b>26</b> |
| <b>A. Demostración del paso de BQM al hamiltoniano de Ising</b>                              | <b>26</b> |
| <b>B. Tablas de resultados</b>   | <b>28</b> |
| B.1. Carteras de empresas utilizadas en el experimento . . . . .                             | 28        |
| B.2. Resultados de la complejidad de las carteras de 4 empresas . . . . .                    | 29        |
| B.3. Evolución de las compras de acciones en el ciclo de <i>annealing</i> cuántico . . . . . | 30        |
| <b>C. Gráficos de Frontera Eficiente</b>   | <b>32</b> |
| C.1. Carteras de 4 empresas . . . . .  | 32        |
| C.2. Carteras de 6 empresas . . . . .  | 35        |
| <b>D. Histogramas de los campos locales y los acoplamientos.</b>                             | <b>37</b> |

- E. Evolución de la energía en el annealing cuántico con pasos distribuidos uniformemente en la escala lineal 41
- F. Evolución del gap de energías entre los primeros excitados y el nivel fundamental con pasos tentativos distribuidos uniformemente en la escala logarítmica 42

# 1. Introducción

La optimización de carteras es una rama fundamental de las finanzas que consiste en seleccionar la mejor combinación de activos financieros (acciones, bonos, etc.) con el objetivo de maximizar el beneficio esperado y minimizar el riesgo de la inversión, bajo ciertas restricciones como el presupuesto disponible o el nivel de aversión al riesgo.

El modelo clásico fue introducido por Harry Markowitz en 1952, formulando el problema como la minimización de una función que combina la media y la varianza de los retornos de los activos. Desde entonces, han surgido múltiples variantes del modelo que abordan el problema desde diferentes enfoques: con múltiples objetivos, en múltiples periodos, con o sin costes de transacción, o incorporando incertidumbre asociada a la estadística del problema.

Este proyecto se centra en un modelo de múltiples objetivos en un único periodo de inversión, cuyo propósito es minimizar el riesgo y maximizar el beneficio esperado, fijando tanto el nivel de aversión al riesgo como el presupuesto total disponible.

A lo largo del trabajo se ha desarrollado una metodología completa que comienza con la adquisición de datos reales de mercado desde *Yahoo Finance* para más de 280 empresas, a partir de los que se han calculado rendimientos y covarianzas mensuales. Con estos datos se han generado múltiples carteras representativas, diferenciadas por su tamaño de codificación, niveles de correlación entre activos y grado de complejidad.

Inicialmente, el objetivo era resolver estas carteras utilizando el ordenador cuántico de D-Wave mediante el enfoque de *quantum annealing*. Sin embargo, por limitaciones técnicas, se ha recurrido a un enfoque clásico basado en el algoritmo de *simulated annealing*. Finalmente, el problema se ha reformulado también como un modelo de Ising cuántico, lo que permite estudiar su comportamiento desde una perspectiva cuántica y comparar los resultados con los obtenidos mediante el enfoque clásico.

El objetivo del trabajo es evaluar la viabilidad de este enfoque, inspirado en modelos físicos, para resolver problemas reales de optimización de carteras. Se analizarán su rendimiento computacional, la calidad de las soluciones obtenidas y las limitaciones prácticas en cuanto a escalabilidad y precisión.

## 2. Marco Teórico

### 2.1. Optimización de carteras: Fundamentos teóricos

El modelo de Markowitz constituye el enfoque clásico para la construcción de carteras eficientes, en el que se busca simultáneamente maximizar la rentabilidad esperada y minimizar el riesgo asociado a la inversión. Introduciendo un parámetro que cuantifica el grado de aversión al riesgo del inversor, es posible seleccionar una cartera óptima dentro del conjunto de carteras eficientes, en función de sus preferencias.

En el contexto del modelo, se consideran  $N$  empresas en las que se pueden invertir. A partir del análisis temporal de la evolución de sus precios, para cada empresa  $i$  se calcula el retorno esperado por acción  $\mu_i$ , definido como la media mensual de los rendimientos históricos, y el riesgo individual como la volatilidad, es decir, la desviación típica  $\sigma_i$  de dichos rendimientos. De forma vectorial, el retorno total por acción se representa mediante el vector  $\mu = \{\mu_i\}$ , mientras que el riesgo total se codifica mediante la matriz de covarianzas  $\Sigma = \{\sigma_{ij}\}$ , donde:

$$\sigma_{ij} = \begin{cases} \sigma_i^2 & \text{si } i = j \\ \rho_{ij}\sigma_i\sigma_j & \text{si } i \neq j \end{cases} \quad (2.1)$$

Siendo  $\rho_{ij}$  la correlación entre los activos  $i$  y  $j$ .

Sea  $B$  el presupuesto total,  $n = \{n_i\}$  el número de acciones adquiridas de cada empresa  $i$  y  $p = \{p_i\}$  el precio medio por acción. El objetivo consiste en encontrar la combinación de valores de  $n_i$  que maximice el retorno total, minimice el riesgo global y respete la restricción presupuestaria. Esto conduce al siguiente problema de optimización [1]:

$$\begin{aligned} \text{Maximizar: } & \mu^T n \\ \text{Minimizar: } & n^T \Sigma n \\ \text{s.a. } & p^T n = B \end{aligned} \quad (2.2)$$

### 2.2. Formulación computacional: Del modelo financiero al QUBO

Un problema QUBO (*Quadratic Unconstrained Binary Optimization*) es una clase de problema de optimización en el que la función objetivo está compuesta exclusivamente por términos lineales y cuadráticos, las variables son binarias y no existen restricciones explícitas fuera de la función objetivo [2]. Todas las condiciones adicionales deben ser incorporadas mediante penalizaciones dentro de dicha función. Formalmente, un problema QUBO puede escribirse como:

$$\min_{x \in \{0,1\}^n} x^T Q x = \sum_{i=1}^n Q_{ii} x_i + \sum_{\substack{i,j=1 \\ i < j}}^n Q_{ij} x_i x_j \quad (2.3)$$

Este tipo de problemas presenta una gran versatilidad y puede aplicarse a una amplia variedad de contextos, dado que numerosos problemas de optimización en ciencias aplicadas, ingeniería o economía involucran variables con dependencias cuadráticas.

En el caso del modelo propuesto de optimización de carteras, las variables no son binarias de forma inmediata, dado que el problema fue inicialmente concebido para ser resuelto como un DQM (*Discrete Quadratic Model*), en el que las variables son enteras y discretas. Esta formulación estaba orientada a ser ejecutada mediante el *solver* híbrido proporcionado por la plataforma cuántica de D-Wave ([3]), y la función del DQM que se debía minimizar es la siguiente:

$$\text{DQM} = - \sum_{i=1}^N \mu_i n_i + \alpha \sum_{i,j=1}^N \sigma_{ij} n_i n_j + \lambda \left( B - \sum_{i=1}^N p_i n_i \right)^2 \quad (2.4)$$

Donde  $\alpha$  es el parámetro que nos indica el grado de aversión al riesgo.

En esta expresión, el primer término corresponde a la maximización del beneficio esperado (lineal), el segundo representa el riesgo total (cuadrático) y el tercero incorpora la restricción presupuestaria mediante un término penalizado, que introduce tanto componentes lineales como cuadráticos en la función objetivo.

Sin embargo, debido a que el acceso al *solver* híbrido fue restringido por la empresa D-Wave a planes de suscripción con costes económicamente inasumibles, se optó por reformular el modelo utilizando una codificación binaria. Esta adaptación permitió expresar el problema como un BQM (*Binary Quadratic Model*), es decir, en términos de una función cuadrática sobre variables binarias, lo que posibilita su resolución mediante técnicas accesibles como el *simulated annealing*, disponible públicamente en la librería de desarrollo de D-Wave.

Para representar en binario el número de acciones que se desea adquirir de cada empresa, se procede del siguiente modo: para cada activo  $i$  se calcula el número máximo de acciones que podría comprarse dedicando la totalidad del presupuesto  $B$  únicamente a dicha empresa. Este valor se define como:

$$n_i^{\max} = \frac{B}{p_i} \quad (2.5)$$

Donde  $p_i$  es el precio medio por acción de la empresa  $i$ . A continuación, se representa  $n_i^{\max}$  en binario, lo que implica que el número de bits necesarios para codificar esta cantidad es:

$$m_i = \lceil \log_2(n_i^{\max}) \rceil + 1 \quad (2.6)$$

De este modo, cada variable entera  $n_i$  asociada al número de acciones queda parametrizada como una suma ponderada de  $m_i$  variables binarias:

$$n_i = \sum_{k=0}^{m_i-1} 2^k b_{i,k} \quad (2.7)$$

Los parámetros de beneficio, riesgo e inversión total resultan, respectivamente:

$$\sum_{i=1}^N \mu_i n_i = \sum_{i=1}^N \mu_i \left( \sum_{k=0}^{m_i-1} 2^k b_{i,k} \right) = \sum_{i=1}^N \sum_{k=0}^{m_i-1} 2^k \mu_i b_{i,k} \quad (2.8)$$

$$\sum_{i,j=1}^N n_i \sigma_{ij} n_j = \sum_{i,j=1}^N \left( \sum_{k=0}^{m_i-1} 2^k b_{i,k} \right) \sigma_{ij} \left( \sum_{l=0}^{m_j-1} 2^l b_{j,l} \right) = \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} 2^k 2^l \sigma_{ij} b_{i,k} b_{j,l} \quad (2.9)$$

$$\sum_{i=1}^N p_i n_i = \sum_{i=1}^N p_i \left( \sum_{k=0}^{m_i-1} 2^k b_{i,k} \right) = \sum_{i=1}^N \sum_{k=0}^{m_i-1} 2^k p_i b_{i,k} \quad (2.10)$$

Por lo tanto, la función del BQM que hay que minimizar acaba siendo:

$$\text{BQM} = - \sum_{i=1}^N \sum_{k=0}^{m_i-1} 2^k \mu_i b_{i,k} + \alpha \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} 2^k 2^l \sigma_{ij} b_{i,k} b_{j,l} + \lambda \left( B - \sum_{i=1}^N \sum_{k=0}^{m_i-1} 2^k p_i b_{i,k} \right)^2 \quad (2.11)$$

Se puede observar que cada término se comporta de forma análoga al DQM. El primero contiene únicamente componentes lineales, el segundo es puramente cuadrático y el tercero introduce tanto términos lineales como cuadráticos. Es importante destacar que, debido a la codificación binaria, los términos asociados a los bits más significativos (aquellos que multiplican potencias de  $2^k$  más altas) tienen un mayor impacto en el valor de la función objetivo.

### 2.3. Sampler de *simulated annealing*

Este *sampler* se basa en el algoritmo de *simulated annealing*, un método heurístico diseñado para encontrar la solución óptima o una muy próxima a ella dentro de un espacio de soluciones de gran tamaño [4]. Su inspiración proviene de un proceso físico característico de los metales, en el que estos se calientan hasta alcanzar un estado de alta energía y desorden (se vuelven maleables), y posteriormente se enfrían de forma controlada con el fin de alcanzar una estructura cristalina más estable y de mínima energía.

Desde un punto de vista computacional, este procedimiento se modela asignando a cada configuración del sistema una energía asociada, determinada en nuestro caso por el valor de la función objetivo del BQM. Esta energía es análoga al modelo de Ising, en el que la energía del sistema viene dada por el Hamiltoniano de Ising:

$$\hat{H}(s_1, \dots, s_N) = \sum_{i=1}^N h_i s_i + \sum_{\substack{i,j=1 \\ i < j}}^N J_{ij} s_i s_j, \quad s_i \in \{-1, 1\} \quad (2.12)$$

Donde  $s_i$  representan los espines del sistema. El cambio de variables binarias a espines se realiza mediante:

$$b_i = \frac{s_i + 1}{2} \quad (2.13)$$

En el modelo de Ising, los espines están unidos mediante interacciones magnéticas de largo alcance, que pueden ser ferromagnéticas ( $J_{ij} < 0$ ) o antiferromagnéticas ( $J_{ij} > 0$ ), y en las cuales los espines tienden a estar alineados u opuestos respectivamente.

En las primeras etapas del algoritmo, cuando la temperatura es elevada, se permite al sistema explorar libremente el espacio de soluciones, incluyendo aquellas que suponen un aumento en la energía. A medida que la temperatura disminuye progresivamente, el sistema tiende a asentarse en regiones del espacio correspondientes a valles de energía más profundos, reduciendo su capacidad para escapar de mínimos locales.

La probabilidad de escapar de los mínimos locales (también denominados estados metaestables) y alcanzar configuraciones de menor energía está relacionada con la altura y la anchura de la barrera energética que separa dichos estados. Debido a las fluctuaciones térmicas propias del sistema a temperaturas elevadas, el algoritmo puede “saltar” estas barreras, permitiendo así la exploración de regiones más profundas del paisaje energético antes de que el sistema se estabilice en un mínimo local.

Cuando la temperatura se aproxima a cero, el algoritmo converge hacia una solución que se encuentra en un mínimo local de las regiones más profundas. Esta solución puede coincidir con el mínimo global, aunque no existe garantía de ello. Por tanto, *simulated annealing* no asegura encontrar la solución óptima, pero proporciona con alta probabilidad soluciones de energía baja, cercanas a la óptima.

Para aumentar la fiabilidad del resultado, el proceso se repite múltiples veces con distintas condiciones iniciales, y se selecciona la mejor solución obtenida entre todas las ejecuciones.

## 2.4. Computación cuántica adiabática

El teorema adiabático en mecánica cuántica establece que, si se tiene un Hamiltoniano dependiente del tiempo  $\hat{H}(t)$  y el sistema se encuentra inicialmente en su estado fundamental, entonces, siempre que dicho Hamiltoniano cambie lo suficientemente lento, el sistema permanecerá en ese estado fundamental durante toda la evolución.

La computación cuántica adiabática se basa en este concepto [5]. Se parte de un Hamiltoniano inicial  $\hat{H}_0$ , del cual se conoce el estado fundamental. Se inicializa el sistema en el estado fundamental, siendo crucial que dicho estado sea fácil de preparar y calcular. A partir de ahí, se realiza una evolución controlada hacia un Hamiltoniano final  $\hat{H}_1$ , cuyo estado fundamental representa la solución del problema que se desea resolver.

Esta interpolación temporal entre ambos Hamiltonianos se describe mediante:

$$\hat{H}(t) = A(t)\hat{H}_0 + B(t)\hat{H}_1, \quad \text{con } B(t) = 1 - A(t) \quad (2.14)$$

con las condiciones  $A(0) = 1$  y  $A(\tau) = 0$ , siendo  $\tau$  el instante final del algoritmo.

En este proceso, el Hamiltoniano debe variar lentamente para evitar transiciones a estados excitados. Para garantizar que se trata de una evolución adiabática, debe cumplirse la siguiente condición:

$$\tau \gg \frac{\max_{0 < s < 1} \left| \langle \psi_1(s) | \frac{\partial \hat{H}(s)}{\partial s} | \psi_0(s) \rangle \right|}{\min_{0 < s < 1} (E_1(s) - E_0(s))^2}, \quad \text{donde } s = \frac{t}{\tau} \quad (2.15)$$

Es decir, la duración  $\tau$  de la transición adiabática debe ser inversamente proporcional al cuadrado del mínimo *gap* de energías entre el estado fundamental y el primer estado excitado.

Además, se deben cumplir otras condiciones: los Hamiltonianos inicial y final no deben conmutar, el estado fundamental no debe ser degenerado, y el *gap* energético ha de ser lo suficientemente grande. Si en algún punto el *gap* se cierra, debe ralentizarse la evolución para garantizar que el sistema permanece en el estado fundamental.

El principio de computación cuántica adiabática constituye la base teórica sobre la que se fundamenta el algoritmo de *quantum annealing*, implementado en sistemas físicos como el ordenador cuántico de D-Wave.

## 2.5. Quantum annealing

El *quantum annealing* es un algoritmo de optimización en el cual, en lugar de superar las barreras energéticas mediante fluctuaciones térmicas (como en el caso del *simulated annealing*), se busca atravesarlas mediante fluctuaciones cuánticas.

Mientras que el *simulated annealing* emplea una temperatura efectiva para explorar el espacio de soluciones, el *quantum annealing* se basa en la introducción de un campo transversal, que induce superposición de estados y permite al sistema cuántico explorar regiones del espacio energético no accesibles clásicamente, favoreciendo así la posibilidad de escapar de mínimos locales mediante el efecto túnel [6].

Inspirándonos en la computación cuántica adiabática, podemos describir la evolución del sistema mediante el siguiente Hamiltoniano dependiente del tiempo:

$$\hat{H}(t) = A(t) \sum_{i=1}^N \hat{\sigma}_x^{(i)} + B(t) \left( \sum_{i=1}^N h_i \hat{\sigma}_z^{(i)} - \sum_{\substack{i,j=1 \\ i < j}}^N J_{ij} \hat{\sigma}_z^{(i)} \hat{\sigma}_z^{(j)} \right) \quad (2.16)$$

El primer término representa el Hamiltoniano inicial  $\hat{H}_0$ . El estado fundamental de este término es:

$$|+\rangle^{\otimes N} = \left( \frac{1}{\sqrt{2}} \right)^N (|0\rangle + |1\rangle) \otimes \cdots \otimes (|0\rangle + |1\rangle) \quad (2.17)$$

Este estado corresponde a una superposición uniforme de todas las configuraciones clásicas posibles del sistema [7]. El operador de Pauli  $\hat{\sigma}_x$  actúa intercambiando los estados  $|0\rangle$  y  $|1\rangle$ :

$$\hat{\sigma}_x |0\rangle = |1\rangle, \quad \hat{\sigma}_x |1\rangle = |0\rangle$$

Desde el punto de vista físico, este Hamiltoniano representa un campo transversal aplicado en la dirección  $x$ , perpendicular a la dirección  $z$  en la que están definidos los estados base  $|0\rangle$  y  $|1\rangle$ . Su presencia induce transiciones entre estados clásicos y permite mantener al sistema en superposición.

El segundo término representa el Hamiltoniano final  $\hat{H}_1$ , que corresponde al modelo de Ising cuántico, donde los valores clásicos  $s_i$  se sustituyen por los operadores de Pauli  $\hat{\sigma}_z^{(i)}$ . Este Hamiltoniano define el paisaje energético del problema, cuyas configuraciones de menor energía se corresponden con las soluciones óptimas.

Así, la transición de  $\hat{H}_0$  a  $\hat{H}_1$  comienza en un estado superpuesto de todos los posibles estados de *qubits*. A medida que  $A(t)$  disminuye y el segundo término del Hamiltoniano gana peso con el aumento de  $B(t)$ , el sistema empieza a explorar los mínimos del paisaje energético. En esta etapa, el sistema puede establecerse transitoriamente en estados meta-estables, pero el campo transversal, mediante los operadores  $\hat{\sigma}_x$ , produce superposición de los estados y permite que se produzca el efecto túnel, atravesando las barreras de energía que separan estos estados de configuraciones más estables.

Finalmente, el campo transversal se reduce hasta anularse, de modo que el sistema colapsa en una configuración clásica que corresponde a una solución del problema. Dicha solución puede ser el óptimo global o, en la mayoría de los casos prácticos, una solución aproximada cercana al óptimo.

## 2.6. Modelo de Ising

En el Hamiltoniano del *quantum annealing*, el término que describe el paisaje energético de soluciones viene dado por el Hamiltoniano del modelo de Ising cuántico:

$$\hat{H}_{\text{ising}} = \sum_{i=1}^N h_i \hat{\sigma}_z^{(i)} + \sum_{\substack{i,j=1 \\ i < j}}^N J_{ij} \hat{\sigma}_z^{(i)} \hat{\sigma}_z^{(j)} \quad (2.18)$$

Donde los términos  $h_i$  y  $J_{ij}$  son, respectivamente, los campos locales y los acoplamientos entre *qubits*.

En nuestro modelo, los índices  $i$  y  $j$  están asociados al número de empresas, no al número de bits. Por tanto, el Hamiltoniano se reformula incluyendo los bits asociados a cada empresa:

$$\begin{aligned} \hat{H}_{\text{ising}} = & \sum_{i=1}^N \sum_{k=0}^{m_i-1} h_{i,k} \hat{\sigma}_z^{(i,k)} + \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} J_{i,j,k,l} \hat{\sigma}_z^{(i,k)} \hat{\sigma}_z^{(j,l)} \\ & + \sum_{i=1}^N \sum_{k=0}^{m_i-1} \sum_{l=k+1}^{m_i-1} J_{i,k,l} \hat{\sigma}_z^{(i,k)} \hat{\sigma}_z^{(i,l)} \end{aligned} \quad (2.19)$$

Para poder pasar del BQM del problema a este Hamiltoniano basta con tener su forma escrita en función de los valores de los espines  $s_i$ , ya que posteriormente se sustituyen por los operadores de Pauli  $\hat{\sigma}_z^{(i)}$ .

Tras descomponer el BQM en sus términos lineales y cuadráticos (véase la demostración en el anexo A), se obtienen las siguientes expresiones para los campos locales y para los acoplamientos entre *qubits*:

$$h_{i,k} = -2^{k-1}\mu_i + 2\alpha \sum_{j=1}^N \sum_{l=0}^{m_j-1} 2^{k-1}2^{l-1}\sigma_{ij} + \lambda p_i 2^k \left( \sum_{j=1}^N \sum_{l=0}^{m_j-1} 2^{l-1}p_i - B \right)$$

$$J_{i,j,k,l} = 2^k 2^{l-1} (\alpha \sigma_{ij} + \lambda p_i^2) \quad (2.20)$$

Los campos locales del sistema recogen tres contribuciones principales. El primer término corresponde al beneficio esperado por unidad de inversión en el activo  $i$ , proyectado sobre los bits que representan la cantidad de acciones adquiridas. A mayor retorno esperado  $\mu_i$ , mayor será la contribución negativa (deseable al minimizar).

El segundo término, ponderado por el parámetro  $\alpha$ , representa la influencia del riesgo global, es decir, las correlaciones  $\sigma_{ij}$  del activo  $i$  con el resto del portafolio. Cuanto mayor sea la correlación y el número de bits asociados, mayor será el campo.

El tercer término, ponderado por  $\lambda$ , se deriva de la penalización asociada al incumplimiento del presupuesto. Aquí intervienen tanto el precio por acción  $p_i$  como las combinaciones binarias de los demás activos. Este término actúa como un sesgo adicional que empuja al sistema a respetar la restricción presupuestaria.

Los acoplamientos están modulados por el término  $\alpha \sigma_{ij}$ , que representa el riesgo compartido entre los activos  $i$  y  $j$ , y por el término  $\lambda p_i^2$ , que deriva de la penalización de presupuesto cruzada entre bits de diferentes activos.

Cabe recalcar también que la presencia de las potencias de 2 pondera la contribución de cada bit (en el caso de los campos locales) o de cada par de bits (en los acoplamientos), según el peso de cada uno en la codificación binaria. Los bits que acompañan a potencias mayores tienen un mayor impacto en la energía total del sistema.

Además, en el contexto del modelo de Ising, los campos locales determinan la preferencia de cada operador  $\hat{\sigma}_z^{(i,k)}$  hacia uno de sus dos posibles autoestados:  $|0\rangle$  o  $|1\rangle$ . Si  $h_{i,k} > 0$ , el campo favorece el estado  $|0\rangle$ , mientras que si  $h_{i,k} < 0$ , favorece el estado  $|1\rangle$ . Esta convención depende de la forma del Hamiltoniano, es decir, si los signos se encuentran invertidos, la interpretación es la opuesta.

Por otro lado, los acoplamientos controlan la interacción entre pares de espines. Si  $J_{i,j,k,l} < 0$ , el sistema favorece que  $\hat{\sigma}_z^{(i,k)} = \hat{\sigma}_z^{(j,l)}$ , y ambos qubits tienden a alinearse (*ferromagnetismo*). Mientras que si  $J_{i,j,k,l} > 0$ , se favorece que  $\hat{\sigma}_z^{(i,k)} \neq \hat{\sigma}_z^{(j,l)}$ , y ambos qubits tienden a adoptar valores opuestos (*antiferromagnetismo*).

## 2.7. Ordenador cuántico de D-Wave

El procesador cuántico de D-Wave (QPU) implementa físicamente el modelo de Ising mediante una red de qubits superconductores interconectados a través de acopladores magnéticos denominados *couplers*.

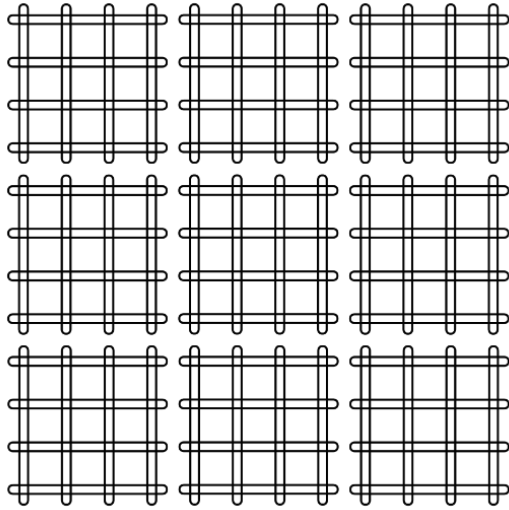
La descripción de la topología del procesador cuántico de D-Wave (QPU) que sigue a continuación se basa en la documentación técnica proporcionada por D-Wave Systems [8].

Primero se va a comenzar describiendo la topología de *Chimera*, cuyas bases fueron implementadas por la generación previa a la actual de ordenadores cuánticos de D-Wave.

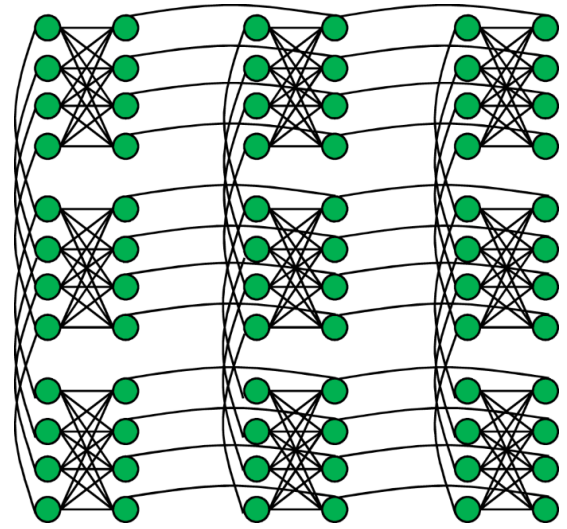
En la generación actual se utiliza la topología de *Pegasus*, que es una evolución de la de Chimera permitiendo mayor conectividad entre qubits.

En la topología *Chimera*, cada *qubit* se implementa como un bucle superconductor orientado vertical u horizontalmente. Estos *qubits* se agrupan en celdas unitarias, que constituyen los bloques básicos de la arquitectura. Cada celda unitaria está compuesta por 8 *qubits*: cuatro dispuestos verticalmente y cuatro dispuestos horizontalmente.

Existen dos formas principales de establecer conexiones entre los *qubits* en la topología de *Chimera*. La primera consiste en los acoplamientos internos, que conectan *qubits* ortogonales dentro de una misma celda unitaria, es decir, *qubits* verticales con *qubits* horizontales. La segunda corresponde a los acoplamientos externos, que conectan parejas de *qubits* paralelos ubicados en la misma fila o columna de la red, permitiendo así la interconexión entre distintas celdas unitarias. En resumen, cada *qubit* tiene 4 conexiones dadas por los acoplamientos internos y 2 dadas por los acoplamientos externos.



(a) Estructura de una celda unitaria en la topología Chimera.



(b) Conectividad entre celdas unitarias en la topología Pegasus.

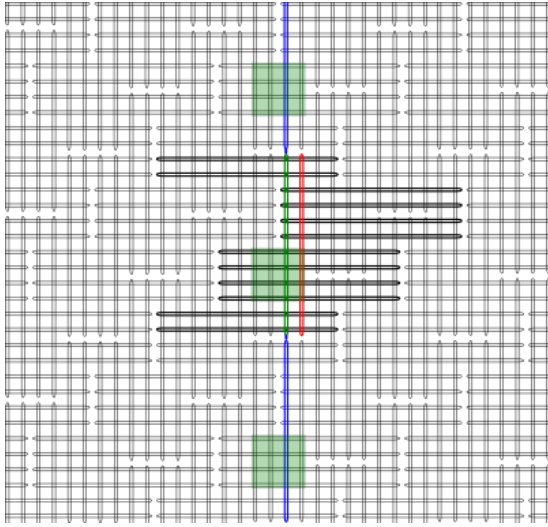
Figura 1: Topologías de conectividad en arquitecturas cuánticas basadas en la topología de Chimera.

Hoy en día la topología usada es la de *Pegasus*. La disposición de los *qubits* es análoga a la de *Chimera*, pero estos no están alineados en una cuadrícula regular, sino que los *qubits* con la misma orientación están parcialmente desplazados. Esto permite una tercera forma de conexión entre *qubits*, llamada *odd couplers*, que consiste en acoplamientos entre *qubits* con la misma orientación y configuración interna. Este tipo de enlace permite establecer conexiones entre *qubits* que, en arquitecturas anteriores, no podían comunicarse directamente.

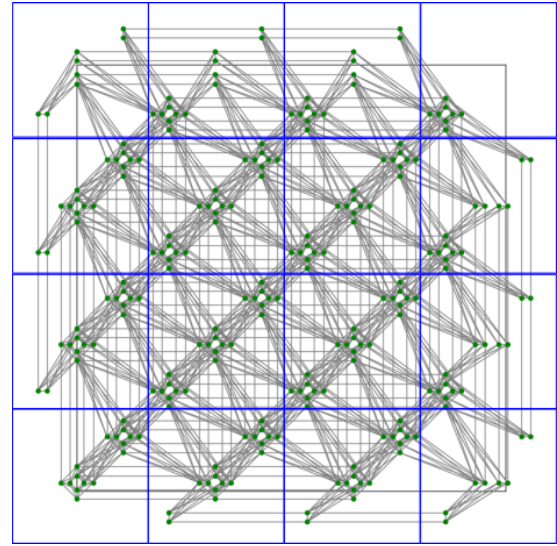
En la figura 2 se muestra un ejemplo de la conectividad total de un único *qubit* (representado en verde). En este caso, el *qubit* está conectado a un total de 15 vecinos: doce mediante acoplamientos internos (*qubits* negros), dos mediante acoplamientos externos (*qubits* azules) y uno adicional mediante un *odd coupler* (*qubit* rojo).

A continuación, se representa la topología de *Pegasus* con las conexiones descritas,

donde los *qubits* son los puntos azules y las líneas grises representan las conexiones entre ellos.



(a) Estructura de una celda unitaria en la topología Pegasus.



(b) Conectividad entre celdas unitarias en la topología Chimera.

Figura 2: Topologías de conectividad en arquitecturas cuánticas basadas en la topología de Pegasus.

Así, al trasladar el modelo de Ising al QPU de D-Wave, los campos locales  $h_i$  se representan físicamente como un campo magnético externo aplicado individualmente a cada *qubit*, que favorece uno de los dos estados posibles (corriente en sentido horario o antihorario). Por otro lado, los acoplamientos  $J_{ij}$  entre *qubits* se implementan mediante interacciones inductivas programables. Dado que estos *qubits* son pequeños circuitos superconductores, dichas interacciones son físicamente equivalentes a la inducción electromagnética entre dos bobinas, como en un transformador. A través de estos acoplamientos, se controla la energía asociada a la unión de los pares de *qubits*, favoreciendo configuraciones ferromagnéticas o antiferromagnéticas según el signo del acoplamiento.

Por último, cabe destacar que, al implementar el algoritmo de *quantum annealing* en un sistema físico real, la condición adiabática ideal no siempre se cumple estrictamente. Una de las principales causas es que, aunque el sistema opera a temperaturas extremadamente bajas (del orden de milikelvins), dicha temperatura no es exactamente nula, lo que implica la presencia de fluctuaciones térmicas residuales. Además, el tiempo del ciclo de *annealing* no puede ser muy grande debido a que el sistema cuántico está expuesto a su entorno, lo que introduce ruido térmico, magnético y eléctrico. Esto implica que, cuanto más tiempo dure el ciclo, más probable es que un *qubit* interactúe con su entorno y pierda el estado de superposición, clave para encontrar una solución válida del problema. Generalmente se suelen obtener resultados altamente cercanos a la solución real del problema, pero hay que tener en cuenta estas imperfecciones a la hora de resolver un problema con el ordenador cuántico de D-Wave.

### 3. Análisis experimental

#### 3.1. Introducción al problema. Base de datos de *Yahoo Finance*

Para poder llevar a cabo un análisis adecuado del problema de optimización de carteras de inversión, es necesario disponer de una base de datos lo suficientemente completa como para extraer la evolución temporal de los precios de las acciones de cada empresa durante un intervalo significativo.

En este proyecto se ha utilizado la base de datos proporcionada por *Yahoo Finance*, una plataforma que ofrece información detallada sobre una gran variedad de activos financieros. Para cada activo, se dispone del precio de apertura y cierre diario, así como de los valores máximos y mínimos alcanzados cada día.

A partir de estos datos, se han seleccionado más de 280 empresas para generar las carteras que serán evaluadas en el modelo (véase el archivo anexo para consultar la lista de empresas).

Para construir el modelo BQM a partir de la información descargada de *Yahoo Finance*, se han calculado los rendimientos mensuales históricos en un intervalo de tiempo determinado. Para ello, se han tomado los precios de cierre del último día hábil de cada mes y se ha calculado el rendimiento mensual de cada acción como el porcentaje de variación con respecto al mes anterior.

De este modo, todos los parámetros necesarios para construir el BQM se obtienen a partir de estas series de rendimientos:

$$\begin{aligned}\mu_i &= \frac{1}{T} \sum_{t=1}^T \mu_i^{(t)} \\ \sigma_{ij} &= \frac{1}{T-1} \sum_{t=1}^T \left( \mu_i^{(t)} - \mu_i \right) \left( \mu_j^{(t)} - \mu_j \right)\end{aligned}\tag{3.1}$$

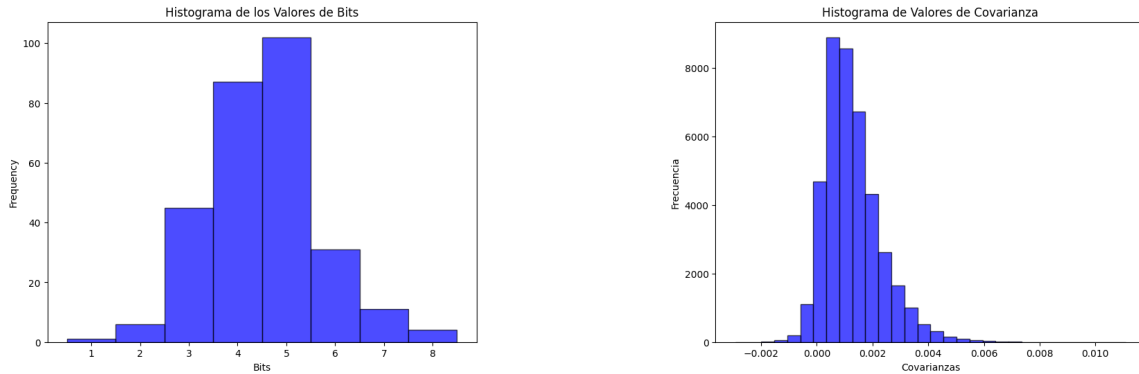
Donde  $T$  representa el número total de meses del intervalo de estudio.

Aplicando la teoría previamente expuesta, se determina el número de bits  $m_i$  necesarios para codificar el número máximo de acciones de cada empresa que podrían adquirirse con el presupuesto total  $B$ . Estos valores, junto con la matriz de covarianzas, son determinantes en la complejidad computacional del problema. A medida que aumenta el número de bits necesarios para representar las decisiones de inversión, también crece la dimensión del modelo QUBO resultante, lo que incrementa de forma significativa el tamaño del espacio de soluciones a explorar.

De hecho, el número total de bits que puede manejarse en el problema está limitado, ya que la complejidad computacional tanto del algoritmo de *simulated annealing* como del de *quantum annealing* crece exponencialmente con el número de variables binarias. Esto implica que, a partir de cierto umbral, el tiempo requerido para encontrar soluciones válidas se vuelve inasumible. En la práctica, el algoritmo de *simulated annealing* permite abordar problemas de hasta aproximadamente 28–29 bits en un portátil de gama media, mientras que el de *quantum annealing* presenta una limitación aún más estricta, en torno a los 20–21 bits.

### 3.2. Generación de carteras de inversión

Para poder generar carteras de inversión acordes con los datos extraídos y con los límites computacionales, se presenta un primer histograma que recoge los valores de los bits necesarios para codificar el número máximo de acciones que se pueden adquirir de cada empresa, teniendo un presupuesto de  $B = 1000\text{€}$ . Junto a este se incluye un segundo histograma que muestra la distribución de los valores de covarianza entre todas las empresas consideradas.



(a) Distribución del número de bits para un presupuesto de 1000€.

(b) Distribución de los valores de covarianza entre pares de empresas.

Figura 3: Características estadísticas de la codificación y relaciones entre activos.

Los resultados del primer histograma son fundamentales para diseñar carteras de inversión que se ajusten a las restricciones computacionales del problema, mientras que los valores del segundo permiten identificar correlaciones significativas entre activos, lo que resulta útil para construir carteras con diferentes perfiles de riesgo y comportamiento.

Se observa que el número de bits necesarios para codificar las acciones de las distintas empresas se encuentra en un rango de 1 a 8, concentrándose la mayoría entre 4 y 5 bits. Esto implica que, en la práctica, mediante el algoritmo de *simulated annealing* pueden resolverse carteras de hasta 5 o 6 empresas como máximo, mientras que en el caso del *quantum annealing* el límite operativo se reduce a una media de unas 4 empresas debido a las restricciones computacionales asociadas al número total de bits.

En cuanto a las covarianzas entre acciones de diferentes empresas, los resultados obtenidos son coherentes con el comportamiento esperado del mercado. Se identifican tanto covarianzas positivas, que reflejan una evolución paralela entre los precios de las acciones, como negativas, donde el ascenso en el valor de una acción tiende a coincidir con el descenso de otra. El rango de valores observado oscila entre -0.002 y 0.008, siendo predominantes las covarianzas positivas, lo cual resulta lógico dado que, en situaciones generales del mercado, la mayoría de los activos tienden a moverse de forma similar ante las mismas condiciones económicas.

A partir de estos resultados, se concluye que, para analizar adecuadamente el comportamiento del modelo en distintos contextos financieros, resulta fundamental generar carteras con perfiles de covarianza diversos: alta, baja, nula y negativa.

Para generar carteras con alta covarianza, se utiliza un procedimiento iterativo basado en los valores previamente calculados de covarianza entre pares de empresas. El proceso se

inicia seleccionando el par de empresas con la mayor covarianza del conjunto. A partir de este par inicial, se incorporan nuevas empresas explorando, en orden descendente, aquellos pares que presenten covarianzas elevadas y que contengan una empresa de las ya incluidas en la cartera. En resumen, en cada iteración se busca seleccionar la empresa con mayor covarianza con las presentes en la cartera, excluyendo los pares ya utilizados y evitando incluir elementos duplicados.

Este procedimiento se repite hasta alcanzar el número deseado de empresas en la cartera. Posteriormente, se reinicia el algoritmo con el par de empresas con la siguiente covarianza más alta, y así sucesivamente generando carteras distintas. Se asegura además que cada cartera generada sea única excluyendo las carteras duplicadas.

Para obtener carteras con covarianza negativa, se repite el mismo algoritmo, pero comenzando con el par de empresas cuya covarianza mutua sea la más negativa del conjunto. En cambio, para construir carteras con covarianza nula o muy próxima a cero, se realiza una transformación previa de los datos: se toma el valor absoluto de todas las covarianzas, de forma que el algoritmo seleccione primero aquellas parejas cuya covarianza (positiva o negativa) esté más próxima al cero. Con las carteras de covarianza baja se realiza un procedimiento semejante al de covarianzas nulas, generando carteras con una covarianza alrededor de 0.002 entre las empresas. Esta última es importante ya que hay un mayor número de empresas relacionadas con este valor de covarianza.

Se han generado carteras de 4 y 6 empresas con las distintas características de covarianza descritas: alta, baja, nula y negativa. Dado el presupuesto de 1000 €, se tienen el número de *bits* necesarios para codificar cada cartera, lo que permite conocer de antemano las dimensiones del modelo del problema y predecir la complejidad asociada a su resolución.

En el anexo B.1 quedan especificadas las carteras que se han generado para cada caso, con las empresas y sus correspondientes *bits*.

### 3.3. Resolución del problema con *simulated annealing*. Frontera eficiente

Para la resolución de cada cartera se va a emplear el algoritmo de *simulated annealing*, ya descrito en el marco teórico del problema. El método de resolución de carteras de inversión es el siguiente: primero se escoge la cartera y se construye el BQM de la misma, fijando los parámetros de aversión al riesgo y de penalización de la restricción de presupuesto, enviándolo a continuación al solver de *simulated annealing*. Cada vez que se envía, se resuelve el BQM 1000 veces, comenzando desde diferentes configuraciones iniciales de bits para poder explorar el paisaje de soluciones energéticas. Este procedimiento se repite 100 veces con el objetivo de obtener múltiples soluciones óptimas generadas por el algoritmo. Una vez completadas las ejecuciones, se almacenan las soluciones obtenidas para cada cartera junto a sus respectivas energías (valor del BQM), el beneficio esperado, el riesgo y la inversión total.

El código utilizado en esta etapa se ha extraído y adaptado a partir del repositorio oficial de D-Wave sobre optimización de carteras en un único periodo [3], reformulando el modelo desde su versión original en DQM hacia una representación en BQM compatible con *simulated annealing*.

Desde el punto de vista experimental, resulta especialmente útil representar gráficamente el conjunto de soluciones, situando el riesgo en el eje X y el beneficio en el eje Y. En esta representación, las soluciones con menor energía, es decir, las más favorables según el modelo, deberían situarse en la parte superior izquierda de la gráfica. El resto de las soluciones, con mayor energía, aparecen progresivamente por debajo.

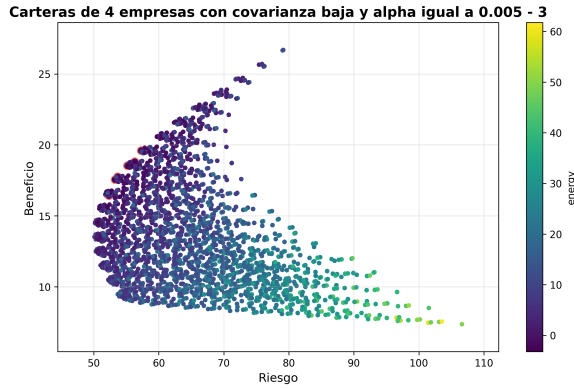
En este tipo de representación se espera observar la denominada frontera eficiente, delimitada por las mejores soluciones obtenidas. En cada ejecución, las mejores soluciones de las 1000 generadas por el *solver* deberían hallarse en dicha frontera o muy próximas a ella. Al repetirse el proceso 100 veces, la frontera eficiente aparece delimitada por el conjunto de mejores soluciones, mientras que el resto de puntos, correspondientes a soluciones subóptimas, se distribuyen por debajo de ella.

Otro factor clave en el análisis experimental son los parámetros libres del modelo. En particular, el coeficiente de aversión al riesgo  $\alpha$  desempeña un papel fundamental en la forma del espacio de soluciones. Valores elevados de  $\alpha$  penalizan fuertemente el riesgo, favoreciendo configuraciones que se concentran en la parte izquierda del plano riesgo-beneficio. A medida que se reduce el valor de  $\alpha$ , el algoritmo permite explorar carteras con mayor riesgo, desplazando progresivamente las soluciones hacia la derecha de la gráfica.

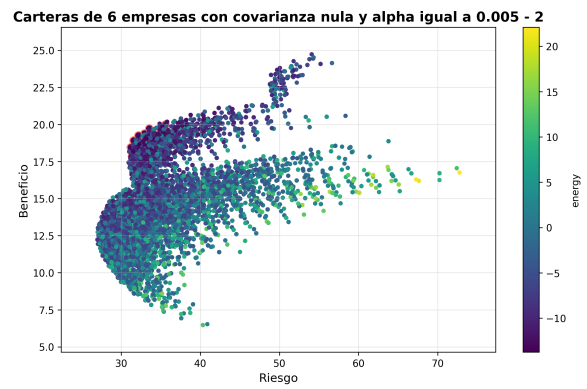
Así, se resuelven las carteras generadas que aparecen en el anexo con  $\lambda = 2.0$ , resolviendo todas a su vez con diferentes grados de aversión al riesgo:  $\alpha_{\text{baja}} = 0.005$ ,  $\alpha_{\text{media}} = 2.0$  y  $\alpha_{\text{alta}} = 20.0$ .

Para representar las soluciones de cada cartera en los gráficos, primero se realiza una preselección de las soluciones cuya inversión total no sea mayor o menor al 0.3% del presupuesto total para carteras con 4 empresas, y del 0.1% para carteras con 6 empresas, con el fin de excluir soluciones energéticamente ineficientes que entorpecen la visualización de las mejores soluciones. Esto es fundamental para presentar un mapa de calor representativo y que se puedan distinguir entre soluciones buenas y las mejores, ya que existen demasiadas soluciones con energías tan elevadas que no deben tenerse ni en cuenta. Además, las mejores soluciones a nivel energético se encuentran rodeadas de rojo para poder distinguir su situación en el gráfico.

Los resultados de todas las carteras representadas en los gráficos se encuentran en el anexo C.1 y C.2. A continuación se muestran dos gráficos que contienen visualmente los resultados más representativos con  $\alpha_{\text{baja}} = 0.005$ : el primero corresponde a una cartera con 4 empresas, y el segundo a una cartera con 6 empresas.



(a) Cartera de 4 empresas.



(b) Cartera de 6 empresas.

Figura 4: Mapas de soluciones riesgo-beneficio obtenidas para carteras con distinto número de empresas.

Se observa el comportamiento esperado: una frontera eficiente bien definida, por encima de la cual no existen soluciones accesibles, y con las configuraciones más óptimas concentradas en la región superior, muy cercanas a la frontera o incluso sobre ella.

A partir de estos resultados se pueden analizar distintos fenómenos de interés. En primer lugar, se examina el desplazamiento de las soluciones más eficientes, marcadas explícitamente en las gráficas. Lo que se aprecia es que, a medida que se incrementa el grado de aversión al riesgo, estas soluciones se trasladan hacia el extremo izquierdo de la frontera eficiente, hasta alcanzar regiones donde la pendiente de dicha frontera tiende a cero, es decir, zonas donde la tangente es prácticamente perpendicular al eje del riesgo.

Otro fenómeno previsto teóricamente y confirmado experimentalmente es que, al disminuir el parámetro de aversión al riesgo  $\alpha$ , incluso manteniendo fija la cartera, el número de soluciones distintas encontradas por el algoritmo de *simulated annealing* aumenta significativamente. Aunque muchas de estas soluciones no sean óptimas, representan nuevos mínimos locales que aparecen en el paisaje energético como consecuencia directa del menor peso asignado al término de riesgo, asociado a configuraciones menos seguras.

Para visualizar este fenómeno, se ha seleccionado una cartera con seis empresas y se han combinado en un mismo gráfico todas las soluciones obtenidas con cada valor de  $\alpha$ .

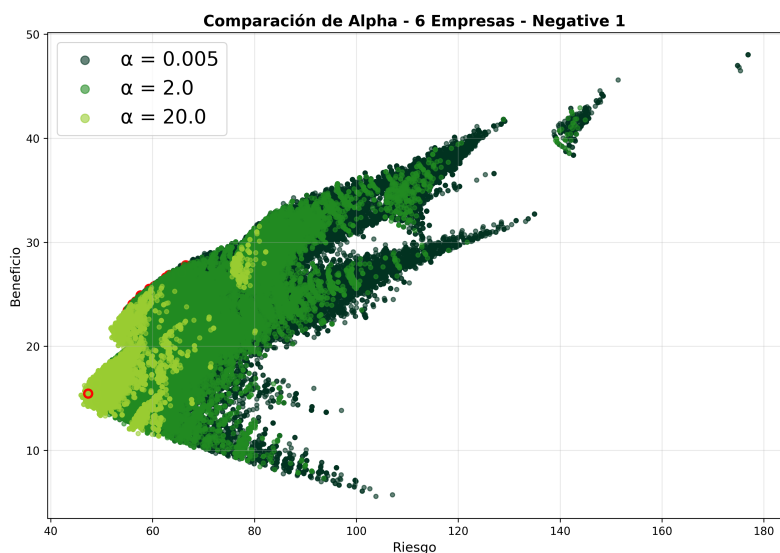


Figura 5: *Soluciones obtenidas para una cartera de 6 empresas con distintos valores de  $\alpha$ .*

En efecto, aquí se ve perfectamente cómo, a medida que disminuye el grado de aversión al riesgo, se exploran un mayor número de soluciones posibles con un riesgo mucho mayor, aumentando el espacio total de posibles soluciones del BQM.

Por último, al analizar carteras con el mismo tipo de covarianza (por ejemplo, todas con baja covarianza) y manteniendo constante el valor de  $\alpha$ , se observa un patrón relevante relacionado con la codificación binaria: a medida que aumenta el número total de bits necesario para representar las acciones posibles de cada empresa, se produce un incremento significativo en el número de soluciones distintas que cumplen con la restricción presupuestaria y, por tanto, aparecen representadas en las gráficas.

Este comportamiento se comprende al analizar el significado práctico de los bits en la codificación binaria. Cuando el número de bits es reducido, las acciones correspondientes presentan un precio elevado, lo que implica que pequeñas variaciones en los bits provocan cambios importantes en la inversión total y, en consecuencia, en la energía del sistema. En cambio, cuando se dispone de un mayor número de bits, el precio de cada acción codificada tiende a ser más bajo y, por tanto, las variaciones discretas en los bits tienen un impacto menor sobre el presupuesto total. Esto permite que muchas más combinaciones distintas de compra pasen el filtro presupuestario, generando un mayor número de soluciones representadas en los gráficos.

### 3.4. Complejidad de una cartera. Exact Solver

Comprender la complejidad computacional asociada a cada cartera es esencial en la práctica, ya que resulta más eficiente estudiar carteras menos complejas antes que aquellas cuya resolución exige un mayor esfuerzo computacional. Desde un punto de vista teórico, una cartera será más compleja cuanto mayor sea el número de bits que la codifican y cuanto mayor sea el número de términos cuadráticos no nulos presentes en la función objetivo del modelo BQM.

En este contexto, podría pensarse que las covarianzas entre los activos financieros son un buen indicador de la complejidad del modelo. No obstante, esta suposición resulta

incorrecta. Las verdaderas contribuciones a la complejidad provienen de las covarianzas entre bits, que se obtienen multiplicando las covarianzas entre empresas por los factores de ponderación binaria, es decir, las potencias de 2 asociadas a cada bit de codificación.

Por lo tanto, incluso cuando dos empresas presentan una baja covarianza, si alguna de ellas requiere un número elevado de bits para representar su número máximo de acciones, los bits más significativos (asociados a potencias de 2 elevadas) pueden generar términos cuadráticos de gran magnitud, lo que incrementa considerablemente la complejidad del problema.

Para poder cuantificar la complejidad computacional de cada cartera se recurre al uso del *Exact Solver* [9]. Este método evalúa de forma exhaustiva todas las posibles configuraciones binarias del modelo, cuyo número total asciende a  $2^{n_{\text{bits}}}$ , siendo  $n_{\text{bits}}$  el número total de variables binarias del BQM. .

Aunque esta estrategia garantiza encontrar la mejor solución posible, su uso práctico está limitado por su elevada complejidad computacional. De hecho, para problemas con más de 23–24 bits, el número de combinaciones crece de forma tan exponencial que su resolución completa se vuelve prácticamente inabordable.

Para evaluar la dificultad de cada cartera, se compara la solución óptima obtenida mediante *Exact Solver* con las soluciones generadas por el algoritmo de *simulated annealing*. En cada ejecución de este último, se recoge la mejor solución encontrada y se compara su energía con la solución real. Si ambas coinciden (o difieren de forma despreciable), se considera que el algoritmo ha logrado encontrar la solución correcta.

Por tanto, el porcentaje de ejecuciones en las que el *simulated annealing* consigue recuperar la solución óptima sirve como una medida de la complejidad del problema: a mayor dificultad, menor será la tasa de aciertos.

Tras tomar estas mediciones con las carteras de 4 empresas, ya que las de 6 contienen demasiados bits, se observa que los resultados más representativos pertenecen a las soluciones con  $\alpha = 2.0$  (véase la tabla del anexo B.2 para los demás resultados).

| Covarianza | Bits | Acierto (%) |
|------------|------|-------------|
| Alta       | 14   | 100         |
|            | 19   | 84          |
|            | 25   | 14          |
| Baja       | 14   | 100         |
|            | 18   | 98          |
|            | 23   | 31          |
| Nula       | 14   | 100         |
|            | 16   | 100         |
|            | 18   | 98          |
| Negativa   | 22   | 21          |
|            | 18   | 98          |
|            | 20   | 77          |
|            | 23   | 66          |

Tabla 1: Porcentaje de aciertos del algoritmo de *simulated annealing* respecto a la solución óptima obtenida con *Exact Solver*

Aquí se ve claramente que los bits son un factor fundamental para la complejidad de una cartera, ya que en todos los casos se ve que a medida que aumentan los bits, aumenta también su complejidad. En cuanto al tipo de covarianza, no se observa ninguna diferencia apreciable debido a lo descrito previamente sobre las potencias de 2 que acompañan a las covarianzas.

Los resultados asociados a  $\alpha = 20.0$  muestran porcentajes de acierto iguales al 100 % o al 0 %. Esto se debe a que, al observar las gráficas correspondientes, el número de soluciones distintas generadas por el algoritmo es significativamente menor que en los casos con menor grado de aversión al riesgo. En estos casos, la mejor solución suele ser única (las 100 mejores son la misma configuración), por lo que el porcentaje de acierto será del 100 % si coincide con la óptima global, o del 0 % en caso contrario. A pesar de este comportamiento, se sigue observando el mismo patrón: la complejidad aumenta con el número de bits necesarios para representar la cartera.

### 3.5. Implementación experimental del algoritmo de *quantum annealing*

Debido a la imposibilidad de acceder al *solver* cuántico de D-Wave, para poder estudiar el comportamiento del modelo en el régimen cuántico se ha adaptado un algoritmo basado en la evolución adiabática. Este algoritmo ha sido creado para que tome como entrada el BQM del problema, traduciendo sus términos lineales y cuadráticos en campos locales y acoplamientos entre *qubits*, respectivamente.

Los valores de los campos locales y de los acoplamientos son claves para predecir la evolución del sistema a lo largo del *annealing* cuántico. En todas las carteras analizadas, se observa que tanto los campos locales como los acoplamientos presentan rangos de valores similares. Para analizar su distribución, se han generado histogramas de ambas componentes, cuyos resultados se presentan a continuación (véase el anexo D):

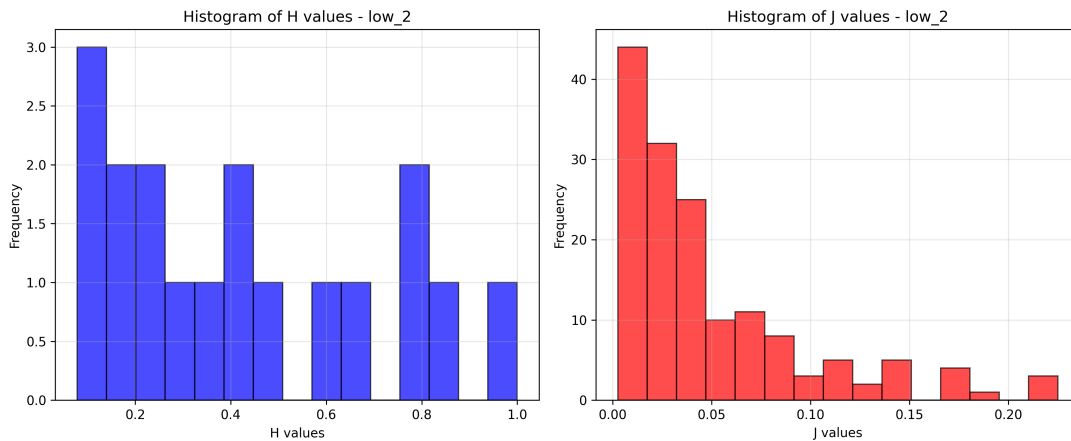


Figura 6: Histogramas de los valores de los campos locales  $h_{i,k}$  (izquierda) y de los acoplamientos  $J_{i,j,k,l}$  (derecha) correspondientes a la segunda cartera con covarianza baja.

Tanto los campos locales  $h_{i,k}$  como los acoplamientos  $J_{i,j,k,l}$  son positivos, lo cual es fundamental para poder explicar lo visto en los siguientes apartados. Teóricamente se ha

explicado antes que si los campos locales son positivos favorecen el estado  $|0\rangle$  y si los acoplamientos son positivos favorecen un estado antiferromagnético entre los qubits.

En cada paso de la evolución, se construye el Hamiltoniano total del sistema modificando los coeficientes de las componentes transversal y longitudinal según el tipo de evolución, y se calcula su espectro energético mediante la diagonalización. A partir del autovector asociado a cada autovalor (estado cuántico), se obtienen los valores esperados de los operadores de Pauli  $\hat{\sigma}_z^{(i,k)}$  y  $\hat{\sigma}_x^{(i,k)}$ . Si el valor esperado es  $\langle \hat{\sigma}_{x,z}^{(i,k)} \rangle < 0$ , se interpreta que el *qubit* se encuentra mayoritariamente en el estado  $|0\rangle$ , por lo que  $b_{i,k} = 0$ . Si  $\langle \hat{\sigma}_{x,z}^{(i,k)} \rangle > 0$ , se asocia a  $b_{i,k} = 1$ . Este procedimiento permite reconstruir las soluciones binarias del problema a partir de la dinámica del sistema cuántico.

### 3.6. Estudio del *gap* de energías. Cambio de escala de evolución lineal a logarítmica

Con el método descrito previamente se puede calcular la energía en cada paso tentativo y obtener la cartera correspondiente al estado fundamental del Hamiltoniano de Ising. Un parámetro fundamental para la correcta evolución del sistema es el *gap* de energías entre el estado fundamental y los primeros estados excitados. Es importante vigilar que este *gap* no se reduzca demasiado, ya que podría provocar transiciones no deseadas al primer estado excitado. En caso de que se detecte un cierre del *gap*, sería necesario reducir el tamaño de los pasos tentativos para garantizar que el sistema permanece en el estado fundamental.

Por lo tanto, es necesario estudiar la evolución del *gap* adiabático para cada cartera, con el fin de identificar regiones críticas en las que podría ser necesario ajustar la dinámica del algoritmo o aplicar un refinamiento en el muestreo para garantizar la validez del proceso adiabático. Se recuerda que el límite computacional se encuentra en los 20–21 bits, por lo que se va a trabajar solo con las carteras que no superen esta cantidad. Para generar el Hamiltoniano de cada cartera se ha empleado un parámetro de aversión al riesgo  $\alpha = 0,005$  con el fin de mantener un espacio de soluciones amplio, y se ha optado inicialmente por una evolución lineal: los pasos tentativos se distribuyen uniformemente en el tiempo total de la evolución del Hamiltoniano.

Experimentalmente se observa un comportamiento similar del *gap* para todas las carteras analizadas (ver anexo E): comienza abierto y se va cerrando progresivamente hasta alcanzar valores muy pequeños en la fase final, donde llega a considerarse prácticamente nulo, ya que la resolución del experimento no permite distinguir su valor real.

Este patrón sugiere que, en el contexto del problema planteado, resulta razonable aumentar la densidad de pasos tentativos en la fase final del proceso adiabático. Para lograrlo, se ha implementado una evolución logarítmica, en la que los pasos se reparten de forma no uniforme: más espaciados al inicio y más concentrados hacia el final.

A continuación, se presenta uno de los resultados obtenidos al aplicar esta nueva estrategia de evolución temporal, representando en la gráfica directamente los valores de diferencias de energía entre los primeros 4 excitados y el estado fundamental (los demás se encuentran en el anexo F).

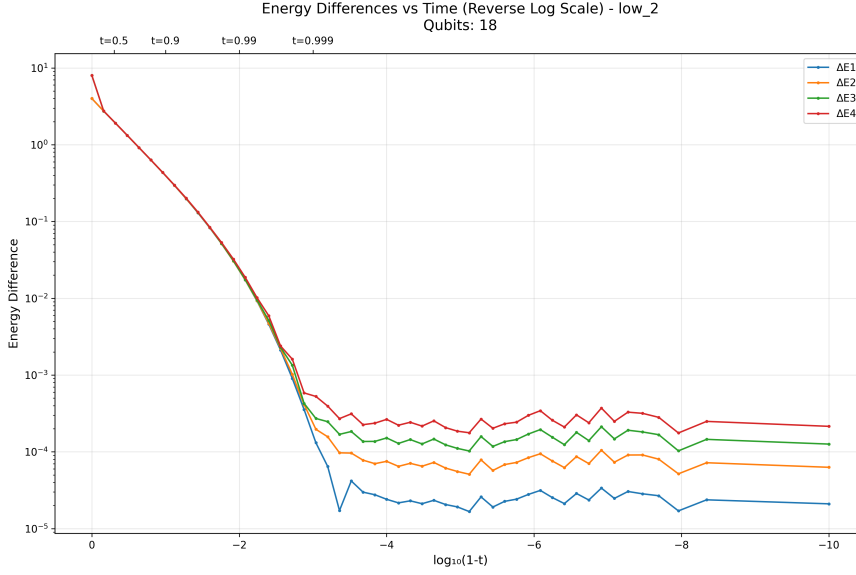


Figura 7: Evolución del *gap* energético con escala logarítmica para una cartera de bajas covarianzas (18 qubits).

Se observa lo siguiente. En el paso inicial, cuando el Hamiltoniano solo tiene componente transversal, la cadena de bits que codifica la solución del problema presenta una configuración aleatoria de ceros y unos, tal como es esperable de un estado en superposición uniforme. A partir del segundo paso, todos los bits adquieren el valor 0. Esto podría verse a simple vista como un cambio de régimen con todos los espines alineados en la misma dirección. Sin embargo, esto se explica debido a la forma de tratar los valores esperados  $\langle \hat{\sigma}_{x,z}^{(i,k)} \rangle$ . Experimentalmente, los campos locales empiezan a tener un mínimo efecto en los valores esperados, y si  $\langle \hat{\sigma}_{x,z}^{(i,k)} \rangle < 0$  se asume directamente que el *qubit* se encuentra en el estado  $|0\rangle$ . Si se realizase un análisis probabilístico en vez de ser determinista, se observaría igualmente una distribución casi aleatoria en los qubits propia del régimen paramagnético, asociado a una componente transversal del Hamiltoniano con un gran peso en el Hamiltoniano total.

Lo esperable sería que, a partir de ese punto, los bits fuesen progresivamente cambiando de valor y “congelándose” conforme se avanza en la evolución adiabática, revelando la solución final de manera gradual. Sin embargo, los bits permanecen congelados en 0, lo cual indica que  $\langle \hat{\sigma}_{x,z}^{(i,k)} \rangle < 0$  hasta que se alcanza una etapa muy avanzada del proceso, justo antes de que el coeficiente del Hamiltoniano transversal desaparezca completamente y el Hamiltoniano total pase a estar dominado por el Hamiltoniano del problema.

Desde un punto de vista físico, esto indica la presencia de dos regímenes diferenciados en la evolución del sistema. En primer lugar, se identifica un régimen paramagnético, dominado por el Hamiltoniano transversal pero influenciado relativamente por los valores de los campos locales. Este régimen abarca la mayor parte de la evolución. Posteriormente, al alcanzar un punto crítico ligado al cierre del *gap* energético y, por tanto, a la cercanía de una posible transición cuántica, se produce un cambio de fase. A partir de ahí, el sistema pasa a un nuevo régimen en el que los espines se reconfiguran para representar una solución válida del problema de optimización planteado. Esta fase es análoga a un régimen spin glass magnético, caracterizado por la frustración característica de estos problemas complejos de optimización y asociada a la incapacidad de satisfacer las tendencias

antiferromagnéticas entre varios *qubits* acoplados los unos con los otros ([10]).

En el marco del problema analizado, estos resultados sugieren además una observación relevante. Al comprobar que los primeros niveles energéticos convergen en el final del proceso, se puede deducir que, incluso en el caso de que el sistema cuántico realizara una transición no deseada hacia un estado excitado lo cual teóricamente podría comprometer la validez del resultado final en el *quantum annealing*, la solución obtenida no diferiría significativamente de la óptima. En otras palabras, el cierre del *gap* energético al final del proceso implica que las diferencias energéticas entre el estado fundamental y los primeros estados excitados son pequeñas. Por tanto, la condición de adiabaticidad no resulta crítica en el contexto específico de este problema.

Cabe destacar que se puede analizar exactamente cuál es la solución de la cartera en cada punto del proceso de *annealing* cuántico. Esto es útil porque permite visualizar la diferencia entre las soluciones del estado fundamental y del primer estado excitado. Aunque estén muy próximas en energía, pueden representar carteras distintas, con compras de acciones diferentes.

Además, también se puede observar cómo cambian las soluciones de la cartera en cada paso del *annealing*. Esto permite ver si, por ejemplo, las soluciones del estado fundamental se mantienen estables o si cambian mucho durante el proceso, lo que da información sobre la estabilidad de la solución óptima. En el anexo B.3 quedan representadas las soluciones obtenidas a lo largo del proceso para dos carteras representativas.

En la cartera con covarianza alta se observa que las compras de acciones varían en cada paso del proceso de *annealing*, sin estabilizarse claramente al final del ciclo. En cambio, en la cartera con covarianza negativa, las soluciones tienden a mantenerse prácticamente constantes en la parte final del proceso.

Esto pone de manifiesto que, incluso cuando el campo transversal es casi nulo, puede seguir influyendo en la dinámica del *annealing* hasta el último instante. Aunque las energías de los estados sean similares, las configuraciones concretas de compra pueden continuar cambiando en algunos casos hasta que el campo transversal se anula por completo.

### 3.7. Complejidad de una cartera en el *annealing* cuántico

Previamente se ha definido la complejidad de una cartera como el porcentaje de soluciones proporcionadas por el *simulated annealing* que coinciden con la solución óptima obtenida mediante el *exact solver*. Un porcentaje bajo indica una mayor complejidad, ya que implica un espacio de soluciones más denso y con energías muy próximas entre sí, lo que dificulta que el algoritmo heurístico identifique el mínimo global.

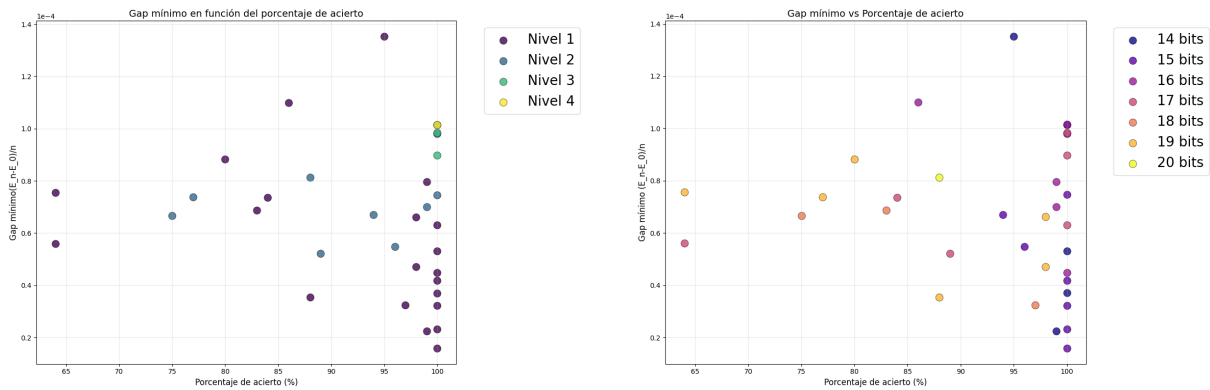
Desde una perspectiva física, este fenómeno puede interpretarse como una *cuasidegeneración* del nivel fundamental en el Hamiltoniano final del sistema: varios estados con configuraciones distintas presentan energías prácticamente iguales a la del estado base. En este contexto, el *gap* entre el estado fundamental y los primeros niveles excitados se reduce notablemente, lo cual incrementa la probabilidad de que el sistema evolucione hacia un estado metaestable en lugar de alcanzar el mínimo absoluto.

Para analizar esta relación se propone lo siguiente: se comienza con un conjunto amplio de carteras de 4 empresas, y se resuelven con un valor de  $\alpha = 2.0$  mediante el algoritmo de *simulated annealing*. A partir de las soluciones obtenidas se calcula el porcentaje de

aciertos respecto a la solución óptima proporcionada por el *exact solver*, lo cual permite parametrizar la complejidad de cada cartera. Análogamente, se resuelven las mismas carteras con el algoritmo de *quantum annealing*, y se registra el *gap* final de energías de los primeros 9 niveles excitados respecto al estado fundamental. Para cada cartera, se divide el valor del *gap* entre el nivel de energía correspondiente (por ejemplo, el *gap* entre el séptimo nivel excitado y el fundamental se divide entre 7), y se selecciona el valor más bajo entre todos ellos.

Este procedimiento permite realizar una comparación equitativa entre todas las carteras, ya que cada una constituye un caso particular: puede ocurrir que una cartera presente los dos primeros niveles excitados muy cercanos al estado fundamental y el resto más alejados, mientras que otra cartera tenga los cinco primeros niveles excitados igualmente próximos. A ojos del *simulated annealing*, ambas situaciones podrían considerarse de complejidad similar, aunque su distribución interna de niveles excitados sea diferente.

Representando gráficamente los resultados obtenidos se observa la siguiente relación:



(a) Coloreado según el nivel de energía seleccionado.

(b) Coloreado según el número de bits de cada cartera.

Figura 8: Gap de energías en función de la complejidad del problema.

Lo que se observa es que los resultados con menor tasa de acierto tienden a estar asociados a *gaps* medios, mientras que conforme la tasa de acierto aumenta, los *gaps* pueden adquirir valores más elevados. No obstante, también pueden adquirir valores bajos del *gap* en carteras cuya tasa de acierto es alta, lo cual indica que la propia topología del espacio de soluciones energéticas desempeña un papel crucial en la complejidad del problema. Si el *gap* de energías es pequeño pero la configuración del paisaje energético favorece la identificación del mínimo global, por ejemplo, si los mínimos locales están muy próximos al mínimo absoluto y no existen barreras de energía significativas entre ellos, entonces el problema puede seguir siendo poco complejo. En consecuencia, aunque puede existir una relación entre el tamaño del *gap* y la dificultad de una cartera, la estructura del espacio de soluciones resulta ser un factor igual o incluso más determinante a la hora de caracterizar su complejidad.

En resumen, vistos los gráficos no se puede sacar una conclusión clara a nivel experimental. Debido a las limitaciones computacionales (no se pueden analizar carteras con más de 20 bits) y a la naturaleza del problema (es complicado obtener carteras con bajos porcentajes de aceptación), tampoco se puede conseguir una estadística buena para abordar este problema.

## 4. Conclusión. Posibles ampliaciones del modelo

En este trabajo se ha planteado el problema de optimización de carteras desde una perspectiva computacional, utilizando datos reales y modelos adaptados a su resolución mediante técnicas tanto clásicas como cuánticas. A partir del modelo de Markowitz, se ha reformulado el problema en forma binaria (BQM), permitiendo su resolución con algoritmos como el *simulated annealing*, y su interpretación en el marco del modelo de Ising cuántico.

Para ello, se han recogido datos financieros reales a través de *Yahoo Finance* y se han construido carteras de inversión con distintas características, prestando especial atención a las covarianzas entre empresas y al número de bits necesarios para representar cada cartera. Estos dos elementos han sido fundamentales para estudiar la complejidad computacional del modelo.

A nivel experimental, se han obtenido numerosas soluciones con *simulated annealing*, analizando cómo influyen parámetros como el grado de aversión al riesgo en el espacio de soluciones. Se ha observado cómo, al aumentar dicho parámetro, se obtienen soluciones más conservadoras (con menos riesgo), y cómo la distribución de los resultados da lugar a una *frontera eficiente* que recoge las soluciones más favorables en términos de riesgo y beneficio.

Además, se ha implementado una simulación del *quantum annealing*, basada en la evolución del Hamiltoniano del modelo de Ising. En este contexto, se ha estudiado el comportamiento del sistema en cada etapa del ciclo adiabático, observando cómo se produce un cambio de fase cuántico al final del proceso, cuando el sistema abandona el estado inicial y adopta una configuración que refleja una posible solución al problema.

También se han discutido los límites computacionales del modelo. Se ha comprobado que el número de bits necesarios para representar las variables tiene un impacto directo en la dificultad de resolver el problema. En particular, con *simulated annealing* el límite práctico está en torno a los 29 bits, y con *quantum annealing*, aún más bajo. Esto impone restricciones en el número de empresas que pueden incluirse en una cartera.

Además, hay un aspecto relevante que no se ha abordado y que podría servir como base para futuras líneas de investigación: la *diversificación*. Este principio fundamental en finanzas consiste en distribuir el capital entre un mayor número de activos con el fin de reducir el impacto que puede tener la evolución negativa de una acción concreta sobre el rendimiento total de la cartera.

Para introducir este concepto en el modelo actual se proponen dos estrategias. La primera consiste en limitar la inversión máxima que se puede realizar en cada empresa mediante un coeficiente de diversificación  $\beta \in [0, 1]$ . Así, el número máximo de acciones permitidas para cada activo pasa a ser:

$$n_i^{\max, \text{new}} = \beta \cdot n_i^{\max, \text{old}}. \quad (4.1)$$

Esta restricción no solo introduce diversificación, sino que además reduce el número de bits necesarios para codificar cada variable.

$$[m_i]_{\text{new}} = [m_i]_{\text{old}} + \log_2 \beta \quad (4.2)$$

Con esta modificación, se podría abordar el estudio de carteras más amplias dentro de los límites computacionales actuales.

La segunda vía consiste en incorporar directamente una penalización por falta de diversificación en la función objetivo del modelo. Esta penalización puede basarse en el índice de Herfindahl-Hirschman (HHI), que mide la concentración de la inversión en los distintos activos [11]. De este modo, se añadiría un término al DQM:

$$F(n_i) = - \sum_{i=1}^N \mu_i n_i + \alpha \sum_{i,j=1}^N \sigma_{ij} n_i n_j + \beta \sum_{i=1}^N \left( \frac{p_i n_i}{B} \right)^2 + \lambda \left( B - \sum_{i=1}^N p_i n_i \right)^2 \quad (4.3)$$

Este enfoque penaliza aquellas carteras en las que el capital está excesivamente concentrado en unos pocos activos, promoviendo soluciones más equilibradas.

Ambas propuestas permiten extender el modelo original hacia escenarios más realistas, acercándolo a situaciones que se presentan con frecuencia en la gestión de carteras en el mundo financiero. Representan, por tanto, una continuación posible del trabajo realizado y un punto de partida para investigaciones futuras.

En definitiva, este trabajo no solo demuestra la viabilidad del uso de técnicas de optimización avanzada en finanzas, sino que también deja abiertas nuevas líneas de desarrollo que se pueden adaptar a las situaciones reales del mercado.

## Referencias

- [1] Botea, A., Bosio, M., & Lee, A. (2020). *Portfolio Optimization using Quantum Annealing*. arXiv preprint arXiv:2012.01121. <https://arxiv.org/abs/2012.01121>
- [2] Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H., & Wang, Y. (2014). The unconstrained binary quadratic programming problem: a survey. *Journal of Combinatorial Optimization*, 28(1), 58–81. <https://arxiv.org/abs/1811.11538>
- [3] D-Wave Systems Inc. (2024). *Portfolio Optimization Example*. GitHub repository. <https://github.com/dwave-examples/portfolio-optimization>
- [4] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671–680. <https://www2.stat.duke.edu/~scs/Courses/Stat376/Papers/TemperAnneal/KirkpatrickAnnealScience1983.pdf>
- [5] Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M. (2000). *Quantum computation by adiabatic evolution*. arXiv. <https://arxiv.org/pdf/quant-ph/0001106>
- [6] D-Wave Systems Inc. (2024). *Quantum Annealing Introduction*. D-Wave Documentation. [https://docs.dwavequantum.com/en/latest/quantum\\_research/quantum\\_annealing\\_intro.html](https://docs.dwavequantum.com/en/latest/quantum_research/quantum_annealing_intro.html)
- [7] Albash, T., & Lidar, D. A. (2014). Adiabatic quantum computation is equivalent to standard quantum computation. *Frontiers in Physics*, 2, 5. <https://www.frontiersin.org/articles/10.3389/fphy.2014.00005/full>
- [8] D-Wave Systems Inc. (2024). *QPU Topologies*. D-Wave Documentation. [https://docs.dwavequantum.com/en/latest/quantum\\_research/topologies.html#qpu-topologies](https://docs.dwavequantum.com/en/latest/quantum_research/topologies.html#qpu-topologies)
- [9] D-Wave Systems Inc. (2024). *ExactSolver — dimod documentation*. [https://test-projecttemplate-dimod.readthedocs.io/en/latest/\\_modules/dimod/reference/samplers/exact\\_solver.html#ExactSolver](https://test-projecttemplate-dimod.readthedocs.io/en/latest/_modules/dimod/reference/samplers/exact_solver.html#ExactSolver)
- [10] Tatsumura, K., & Nishimori, H. (2023). *Spin-glass-like phase in quantum Ising models without frustration*. <https://arxiv.org/abs/2302.04842>
- [11] Sánchez Galán, J. (2016, junio 29). *Índice Herfindahl-Hirschman (IHH)*. Economipedia. <https://economipedia.com/definiciones/indice-herfindahl-hirschman.html>

## A. Demostración del paso de BQM al hamiltoniano de Ising

Se tiene el hamiltoniano:

$$\hat{H}_{\text{ising}} = \sum_{i=1}^N \sum_{k=0}^{m_i-1} h_{i,k} \hat{\sigma}_z^{(i,k)} + \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} J_{i,j,k,l} \hat{\sigma}_z^{(i,k)} \hat{\sigma}_z^{(j,l)} + \sum_{i=1}^N \sum_{k=0}^{m_i-1} \sum_{l=k+1}^{m_i-1} J_{i,k,l} \hat{\sigma}_z^{(i,k)} \hat{\sigma}_z^{(i,l)} \quad (\text{A.1})$$

Así, para poder pasar del BQM del problema al hamiltoniano de Ising basta con tener su forma escrita en función de los valores de los espines  $s_i$ , ya que después sólo hay que sustituirlos por los operadores de Pauli  $\hat{\sigma}_z^{(i)}$ .

Primero, al cambiar de bits a espines se tiene:

$$\begin{aligned} \text{BQM} &= - \sum_{i=1}^N \sum_{k=0}^{m_i-1} (2^k \mu_i) b_{i,k} + \alpha \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} (2^k 2^l \sigma_{ij}) b_{i,k} b_{j,l} + \lambda \left( B - \sum_{i=1}^N \sum_{k=0}^{m_i-1} 2^k p_i b_{i,k} \right)^2 = \\ &= - \sum_{i=1}^N \sum_{k=0}^{m_i-1} (2^k \mu_i) \left( \frac{s_{i,k} + 1}{2} \right) + \alpha \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} (2^k 2^l \sigma_{ij}) \left( \frac{s_{i,k} + 1}{2} \right) \left( \frac{s_{j,l} + 1}{2} \right) \\ &\quad + \lambda \left( B - \sum_{i=1}^N \sum_{k=0}^{m_i-1} 2^k p_i \left( \frac{s_{i,k} + 1}{2} \right) \right)^2 \end{aligned} \quad (\text{A.2})$$

Analizando término a término se tiene que, primero, para el beneficio:

$$- \sum_{i=1}^N \sum_{k=0}^{m_i-1} (2^k \mu_i) \left( \frac{s_{i,k} + 1}{2} \right) = - \sum_{i=1}^N \sum_{k=0}^{m_i-1} (2^{k-1} \mu_i) s_{i,k} + \text{cte.} \quad (\text{A.3})$$

Donde el único término no constante es lineal.

Para el riesgo:

$$\begin{aligned}
& \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} \left( 2^k 2^l \sigma_{ij} \cdot \frac{s_{i,k} + 1}{2} \cdot \frac{s_{j,l} + 1}{2} \right) = \\
& = \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} \left( 2^{k-1} 2^{l-1} \sigma_{ij} \cdot (s_{i,k} s_{j,l} + s_{i,k} + s_{j,l} + 1) \right) = \\
& = \alpha \left[ \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} 2^{k-1} 2^{l-1} \sigma_{ij} s_{i,k} s_{j,l} + 2 \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} 2^{k-1} 2^{l-1} \sigma_{ij} s_{i,k} \right] + \text{cte} = \\
& = \alpha \left[ \sum_{i=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_i-1} 2^{k-1} 2^{l-1} \sigma_{ii} s_{i,k} s_{i,l} + 2 \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} 2^{k-1} 2^{l-1} \sigma_{ij} s_{i,k} s_{j,l} \right. \\
& \quad \left. + 2 \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} 2^{k-1} 2^{l-1} \sigma_{ij} s_{i,k} \right] + \text{cte} = \\
& = \alpha \left[ \sum_{i=1}^N \sum_{k=0}^{m_i-1} (2^{2(k-1)} \sigma_{ii} s_{i,k} s_{i,k}) + 2 \sum_{i=1}^N \sum_{k=0}^{m_i-1} \sum_{l=k+1}^{m_i-1} 2^{k-1} 2^{l-1} \sigma_{ii} s_{i,k} s_{i,l} \right. \\
& \quad \left. + 2 \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} 2^{k-1} 2^{l-1} \sigma_{ij} s_{i,k} s_{j,l} + 2 \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} 2^{k-1} 2^{l-1} \sigma_{ij} s_{i,k} \right] + \text{cte} \tag{A.4}
\end{aligned}$$

Aquí los dos primeros términos dentro del paréntesis son cuadráticos y el tercero es lineal.

Por último, la restricción del presupuesto queda:

$$\begin{aligned}
& \left( B - \sum_{i=1}^N \sum_{k=0}^{m_i-1} 2^k p_i \cdot \frac{s_{i,k} + 1}{2} \right)^2 = \\
& = B^2 + \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} 2^{k-1} 2^{l-1} p_i^2 s_{i,k} s_{j,l} + 2 \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} 2^{k-1} 2^{l-1} p_i^2 s_{i,k} \\
& \quad - \sum_{i=1}^N \sum_{k=0}^{m_i-1} B p_i 2^k s_{i,k} + \text{cte} = \\
& = B^2 + 2 \sum_{i=1}^N \sum_{j=i+1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} 2^{k-1} 2^{l-1} p_i^2 s_{i,k} s_{j,l} + 2 \sum_{i=1}^N \sum_{k=0}^{m_i-1} \sum_{l=k+1}^{m_i-1} 2^{k-1} 2^{l-1} p_i^2 s_{i,k} s_{i,l} \\
& \quad + 2 \sum_{i,j=1}^N \sum_{k=0}^{m_i-1} \sum_{l=0}^{m_j-1} 2^{k-1} 2^{l-1} p_i^2 s_{i,k} - \sum_{i=1}^N \sum_{k=0}^{m_i-1} B p_i 2^k s_{i,k} + \text{cte} \tag{A.5}
\end{aligned}$$

Donde se tiene también términos lineales, cuadráticos y constantes. Al agrupar los términos lineales, los campos locales resultan:

$$h_{i,k} = -2^{k-1} \mu_i + 2\alpha \sum_{j=1}^N \sum_{l=0}^{m_j-1} 2^{k-1} 2^{l-1} \sigma_{ij} + \lambda p_i 2^k \left( \sum_{j=1}^N \sum_{l=0}^{m_j-1} 2^{l-1} p_i - B \right) \tag{A.6}$$

Se hace lo mismo con los términos cuadráticos. Se pueden juntar los términos asociados a  $ij$  con los términos asociados a  $i=j,kl$  ya que son iguales:

$$J_{i,j,k,l} = 2^k 2^{l-1} (\alpha \sigma_{ij} + \lambda p_i^2) \quad (\text{A.7})$$

## B. Tablas de resultados

### B.1. Carteras de empresas utilizadas en el experimento

| Correlación | Empresa 1 | Empresa 2 | Empresa 3 | Empresa 4 | Bits |
|-------------|-----------|-----------|-----------|-----------|------|
| Alta        | CAT       | NFLX      | CF        | ROK       | 14   |
|             | HES       | MS        | WHR       | LEN       | 19   |
|             | PCAR      | AMD       | IVZ       | HST       | 25   |
| Baja        | BLK       | ACN       | CVS       | GE        | 14   |
|             | MCHP      | DHR       | CCL       | PPG       | 18   |
|             | GME       | HAL       | VNO       | AMZN      | 23   |
| Nula        | CLX       | DTE       | CI        | ROK       | 14   |
|             | DUK       | LMT       | T         | NSC       | 16   |
|             | PPL       | LOW       | MCO       | WEC       | 18   |
|             | FMC       | CPB       | NEE       | PBT       | 22   |
| Negativa    | DLTR      | NFLX      | CTRA      | AAL       | 18   |
|             | DAL       | AAL       | AIZ       | NEM       | 20   |
|             | RF        | DAL       | AAL       | NEM       | 23   |

Tabla 2: Carteras de 4 empresas clasificadas por tipo de correlación y número total de bits necesarios.

| Correlación | Emp. 1 | Emp. 2 | Emp. 3 | Emp. 4 | Emp. 5 | Emp. 6 | Bits |
|-------------|--------|--------|--------|--------|--------|--------|------|
| Alta        | ADSK   | ADBE   | EMN    | AMP    | KMX    | HES    | 23   |
|             | NFLX   | MAS    | BEN    | CF     | A      | WYNN   | 26   |
| Baja        | OXY    | ACN    | WHR    | BLK    | TJX    | ORLY   | 22   |
|             | WDC    | FDX    | TXT    | EBAY   | OMC    | BK     | 28   |
| Nula        | DOV    | CAT    | SO     | NSC    | DUK    | HAS    | 24   |
|             | LOW    | PPL    | CHTR   | WEC    | EBAY   | CAG    | 28   |
| Negativa    | NFLX   | CTRA   | AAL    | UAL    | DAL    | NEM    | 29   |

Tabla 3: Carteras de 6 empresas clasificadas por tipo de correlación y número total de bits necesarios.

## B.2. Resultados de la complejidad de las carteras de 4 empresas

| $\alpha$ | Covarianza | Bits | Acierto (%) |
|----------|------------|------|-------------|
| 0.005    | Alta       | 14   | 100         |
|          |            | 19   | 0           |
|          |            | 25   | 0           |
|          | Baja       | 14   | 99          |
|          |            | 18   | 89          |
|          |            | 23   | 59          |
|          | Nula       | 14   | 100         |
|          |            | 16   | 100         |
|          |            | 18   | 79          |
|          |            | 22   | 0           |
|          | Negativa   | 18   | 89          |
|          |            | 20   | 33          |
| 23       |            | 0    |             |
| 2.0      | Alta       | 14   | 100         |
|          |            | 19   | 84          |
|          |            | 25   | 14          |
|          | Baja       | 14   | 100         |
|          |            | 18   | 98          |
|          |            | 23   | 31          |
|          | Nula       | 14   | 100         |
|          |            | 16   | 100         |
|          |            | 18   | 98          |
|          |            | 22   | 21          |
|          | Negativa   | 18   | 98          |
|          |            | 20   | 77          |
| 23       |            | 66   |             |
| 20.0     | Alta       | 14   | 100         |
|          |            | 19   | 0           |
|          |            | 25   | 0           |
|          | Baja       | 14   | 100         |
|          |            | 18   | 100         |
|          |            | 23   | 100         |
|          | Nula       | 14   | 100         |
|          |            | 16   | 100         |
|          |            | 18   | 100         |
|          |            | 22   | 0           |
|          | Negativa   | 18   | 100         |
|          |            | 20   | 100         |
| 23       |            | 100  |             |

Tabla 4: Porcentaje de acierto del algoritmo de simulated annealing para diferentes niveles de  $\alpha$  y carteras de 4 empresas.

### B.3. Evolución de las compras de acciones en el ciclo de *annealing* cuántico

|                       | Cartera de covarianza alta |        |               |               |
|-----------------------|----------------------------|--------|---------------|---------------|
| $1 \times 10^0$       | -28.00                     | -24.00 | [3, 17, 0, 5] | [4, 14, 7, 2] |
| $6.9 \times 10^{-1}$  | -19.51                     | -16.77 | [0, 0, 0, 0]  | [1, 0, 0, 0]  |
| $4.8 \times 10^{-1}$  | -13.83                     | -11.92 | [0, 0, 0, 0]  | [0, 1, 0, 0]  |
| $3.3 \times 10^{-1}$  | -10.02                     | -8.70  | [0, 0, 0, 0]  | [0, 2, 0, 0]  |
| $2.3 \times 10^{-1}$  | -7.47                      | -6.56  | [0, 0, 0, 0]  | [0, 2, 0, 0]  |
| $1.6 \times 10^{-1}$  | -5.76                      | -5.13  | [0, 0, 0, 0]  | [0, 0, 0, 1]  |
| $2.8 \times 10^{-3}$  | -2.33                      | -2.33  | [0, 0, 0, 0]  | [0, 0, 0, 0]  |
| $1.9 \times 10^{-3}$  | -2.32                      | -2.32  | [0, 0, 0, 0]  | [4, 4, 0, 0]  |
| $1.3 \times 10^{-3}$  | -2.32                      | -2.32  | [0, 8, 0, 4]  | [0, 10, 4, 5] |
| $9.1 \times 10^{-4}$  | -2.32                      | -2.32  | [2, 8, 0, 4]  | [2, 8, 0, 4]  |
| $6.3 \times 10^{-4}$  | -2.32                      | -2.32  | [2, 8, 0, 4]  | [4, 22, 1, 0] |
| $4.4 \times 10^{-4}$  | -2.32                      | -2.32  | [0, 0, 0, 5]  | [0, 0, 0, 0]  |
| $3.0 \times 10^{-4}$  | -2.32                      | -2.32  | [4, 10, 0, 5] | [4, 8, 0, 4]  |
| $2.1 \times 10^{-4}$  | -2.32                      | -2.32  | [4, 12, 0, 0] | [0, 8, 0, 0]  |
| $1.4 \times 10^{-4}$  | -2.32                      | -2.32  | [4, 4, 0, 0]  | [4, 8, 0, 0]  |
| $1.0 \times 10^{-4}$  | -2.32                      | -2.32  | [0, 24, 0, 4] | [0, 8, 0, 0]  |
| $6.9 \times 10^{-5}$  | -2.32                      | -2.32  | [4, 4, 0, 0]  | [4, 10, 0, 4] |
| $4.8 \times 10^{-5}$  | -2.32                      | -2.32  | [1, 24, 4, 4] | [4, 8, 0, 0]  |
| $3.3 \times 10^{-5}$  | -2.32                      | -2.32  | [0, 24, 0, 4] | [4, 8, 0, 0]  |
| $2.3 \times 10^{-5}$  | -2.32                      | -2.32  | [0, 16, 0, 3] | [2, 10, 0, 0] |
| $1.6 \times 10^{-5}$  | -2.32                      | -2.32  | [4, 24, 0, 0] | [4, 10, 0, 0] |
| $1.1 \times 10^{-5}$  | -2.32                      | -2.32  | [0, 24, 0, 1] | [0, 10, 0, 4] |
| $7.6 \times 10^{-6}$  | -2.32                      | -2.32  | [0, 24, 0, 4] | [0, 8, 0, 0]  |
| $5.2 \times 10^{-6}$  | -2.32                      | -2.32  | [0, 16, 0, 1] | [0, 8, 0, 0]  |
| $3.6 \times 10^{-6}$  | -2.32                      | -2.32  | [0, 24, 0, 4] | [0, 8, 0, 0]  |
| $2.5 \times 10^{-6}$  | -2.32                      | -2.32  | [4, 4, 0, 0]  | [0, 10, 0, 4] |
| $1.7 \times 10^{-6}$  | -2.32                      | -2.32  | [0, 24, 1, 4] | [2, 8, 0, 0]  |
| $1.2 \times 10^{-6}$  | -2.32                      | -2.32  | [1, 24, 4, 4] | [0, 24, 0, 0] |
| $8.2 \times 10^{-7}$  | -2.32                      | -2.32  | [0, 24, 0, 2] | [0, 2, 0, 0]  |
| $5.7 \times 10^{-7}$  | -2.32                      | -2.32  | [4, 0, 0, 0]  | [0, 10, 0, 4] |
| $3.9 \times 10^{-7}$  | -2.32                      | -2.32  | [0, 24, 0, 4] | [4, 8, 0, 0]  |
| $2.7 \times 10^{-7}$  | -2.32                      | -2.32  | [0, 24, 0, 4] | [4, 8, 0, 0]  |
| $1.8 \times 10^{-7}$  | -2.32                      | -2.32  | [0, 16, 0, 0] | [4, 8, 0, 0]  |
| $1.2 \times 10^{-7}$  | -2.32                      | -2.32  | [0, 16, 0, 1] | [2, 10, 0, 0] |
| $8.1 \times 10^{-8}$  | -2.32                      | -2.32  | [0, 24, 0, 1] | [4, 10, 0, 4] |
| $5.3 \times 10^{-8}$  | -2.32                      | -2.32  | [0, 24, 0, 4] | [6, 8, 0, 0]  |
| $3.4E \times 10^{-8}$ | -2.32                      | -2.32  | [4, 0, 0, 2]  | [0, 16, 0, 4] |
| $2.0 \times 10^{-8}$  | -2.32                      | -2.32  | [4, 4, 0, 0]  | [0, 8, 0, 4]  |
| $1.1 \times 10^{-8}$  | -2.32                      | -2.32  | [1, 24, 4, 4] | [0, 24, 0, 4] |
| $4.5 \times 10^{-9}$  | -2.32                      | -2.32  | [4, 4, 0, 1]  | [0, 2, 0, 0]  |
| 0.0                   | -2.32                      | -2.32  | [1, 24, 4, 4] | [4, 8, 0, 0]  |

Tabla 5: Evolución de las compras de una cartera de covarianza alta

| <b>Cartera de covarianza negativa</b> |        |        |                 |                |
|---------------------------------------|--------|--------|-----------------|----------------|
| $1-t$                                 | $E_0$  | $E_1$  | Solución 0      | Solución 1     |
| $1 \times 10^0$                       | -36.00 | -32.00 | [2, 49, 11, 0]  | [8, 2, 1, 1]   |
| $6.9 \times 10^{-1}$                  | -25.08 | -22.34 | [0, 0, 0, 0]    | [0, 0, 1, 0]   |
| $4.8 \times 10^{-1}$                  | -17.79 | -15.87 | [0, 0, 0, 0]    | [0, 16, 0, 0]  |
| $3.3 \times 10^{-1}$                  | -12.90 | -11.58 | [0, 0, 0, 0]    | [0, 32, 0, 0]  |
| $2.3 \times 10^{-1}$                  | -9.62  | -8.70  | [0, 0, 0, 0]    | [0, 32, 0, 0]  |
| $1.6 \times 10^{-1}$                  | -7.41  | -6.77  | [0, 0, 0, 0]    | [8, 0, 0, 0]   |
| $2.8 \times 10^{-3}$                  | -2.92  | -2.91  | [0, 0, 0, 0]    | [0, 0, 0, 0]   |
| $1.9 \times 10^{-3}$                  | -2.90  | -2.90  | [0, 0, 0, 0]    | [0, 0, 0, 0]   |
| $1.3 \times 10^{-3}$                  | -2.90  | -2.90  | [16, 56, 8, 0]  | [0, 0, 8, 0]   |
| $9.1 \times 10^{-4}$                  | -2.89  | -2.89  | [24, 56, 8, 0]  | [16, 48, 0, 0] |
| $6.3 \times 10^{-4}$                  | -2.89  | -2.89  | [24, 57, 8, 0]  | [8, 8, 8, 0]   |
| $4.4 \times 10^{-4}$                  | -2.89  | -2.89  | [16, 48, 1, 0]  | [8, 24, 8, 0]  |
| $3.0 \times 10^{-4}$                  | -2.89  | -2.89  | [24, 41, 8, 4]  | [24, 0, 0, 0]  |
| $2.1 \times 10^{-4}$                  | -2.89  | -2.89  | [24, 32, 13, 4] | [16, 32, 0, 0] |
| $1.4 \times 10^{-4}$                  | -2.89  | -2.89  | [8, 48, 0, 0]   | [0, 32, 0, 0]  |
| $1.0 \times 10^{-4}$                  | -2.89  | -2.89  | [8, 32, 0, 4]   | [16, 0, 0, 0]  |
| $6.9 \times 10^{-5}$                  | -2.89  | -2.89  | [8, 44, 2, 4]   | [24, 40, 0, 0] |
| $4.8 \times 10^{-5}$                  | -2.89  | -2.89  | [8, 32, 8, 4]   | [0, 0, 0, 0]   |
| $3.3 \times 10^{-5}$                  | -2.89  | -2.89  | [8, 32, 0, 0]   | [0, 48, 0, 0]  |
| $2.3 \times 10^{-5}$                  | -2.89  | -2.89  | [8, 32, 4, 4]   | [24, 0, 0, 0]  |
| $1.6 \times 10^{-5}$                  | -2.89  | -2.89  | [8, 16, 8, 4]   | [24, 16, 0, 0] |
| $1.1 \times 10^{-5}$                  | -2.89  | -2.89  | [8, 4, 12, 4]   | [24, 0, 0, 0]  |
| $7.6 \times 10^{-6}$                  | -2.89  | -2.89  | [8, 32, 0, 0]   | [0, 32, 0, 0]  |
| $5.2 \times 10^{-6}$                  | -2.89  | -2.89  | [8, 36, 8, 4]   | [24, 16, 0, 0] |
| $3.6 \times 10^{-6}$                  | -2.89  | -2.89  | [8, 0, 8, 4]    | [16, 16, 0, 0] |
| $2.5 \times 10^{-6}$                  | -2.89  | -2.89  | [8, 44, 4, 4]   | [24, 0, 0, 0]  |
| $1.7 \times 10^{-6}$                  | -2.89  | -2.89  | [8, 0, 4, 0]    | [0, 0, 0, 0]   |
| $1.2 \times 10^{-6}$                  | -2.89  | -2.89  | [8, 0, 0, 0]    | [0, 0, 0, 0]   |
| $8.2 \times 10^{-7}$                  | -2.89  | -2.89  | [8, 0, 8, 4]    | [16, 0, 0, 0]  |
| $5.7 \times 10^{-7}$                  | -2.89  | -2.89  | [8, 48, 0, 4]   | [16, 48, 0, 0] |
| $3.9 \times 10^{-7}$                  | -2.89  | -2.89  | [8, 32, 8, 4]   | [16, 0, 0, 0]  |
| $2.7 \times 10^{-7}$                  | -2.89  | -2.89  | [8, 32, 4, 4]   | [16, 48, 0, 0] |
| $1.8 \times 10^{-7}$                  | -2.89  | -2.89  | [8, 16, 8, 4]   | [24, 0, 0, 0]  |
| $1.2 \times 10^{-7}$                  | -2.89  | -2.89  | [8, 0, 8, 4]    | [16, 0, 0, 0]  |
| $8.1 \times 10^{-8}$                  | -2.89  | -2.89  | [8, 48, 12, 4]  | [16, 0, 0, 0]  |
| $5.3 \times 10^{-8}$                  | -2.89  | -2.89  | [8, 4, 14, 4]   | [24, 0, 0, 0]  |
| $3.4E \times 10^{-8}$                 | -2.89  | -2.89  | [8, 32, 10, 4]  | [16, 0, 0, 0]  |
| $2.0 \times 10^{-8}$                  | -2.89  | -2.89  | [8, 32, 10, 4]  | [24, 0, 0, 0]  |
| $1.1 \times 10^{-8}$                  | -2.89  | -2.89  | [8, 32, 6, 4]   | [24, 48, 0, 0] |
| $4.5 \times 10^{-9}$                  | -2.89  | -2.89  | [8, 44, 6, 4]   | [16, 16, 0, 0] |
| 0.0                                   | -2.89  | -2.89  | [8, 4, 6, 4]    | [24, 0, 0, 0]  |

Tabla 6: Evolución de las compras de una cartera de covarianza alta

# C. Gráficos de Frontera Eficiente

## C.1. Carteras de 4 empresas

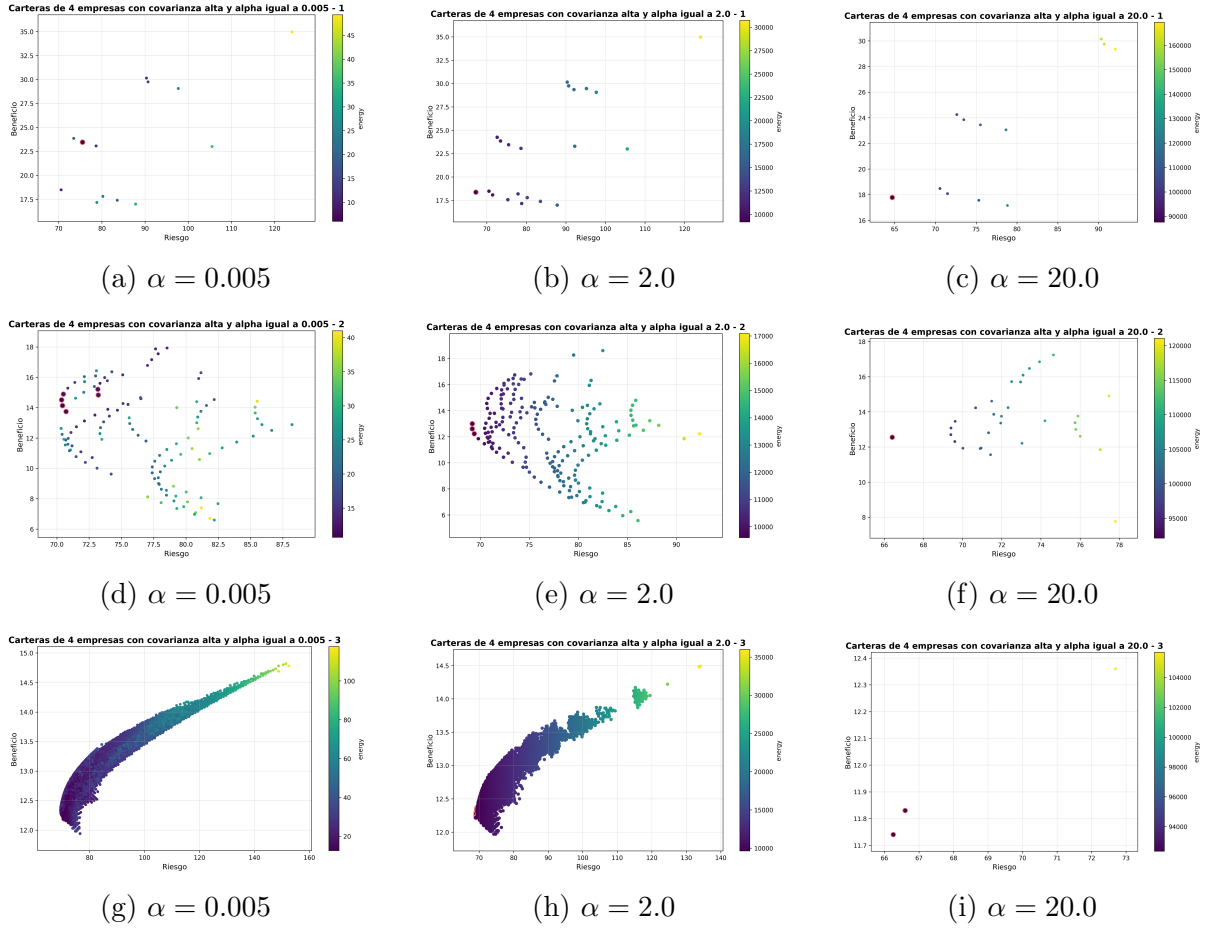
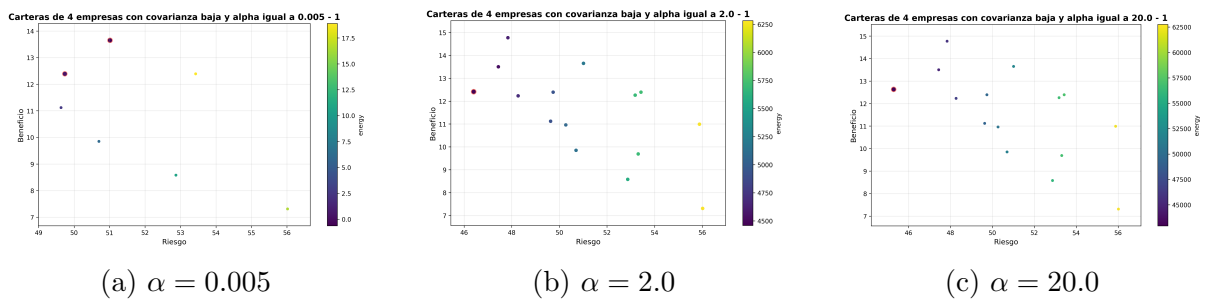


Figura 9: Soluciones de carteras de 4 empresas con covarianza alta, para distintos valores de  $\alpha$ .



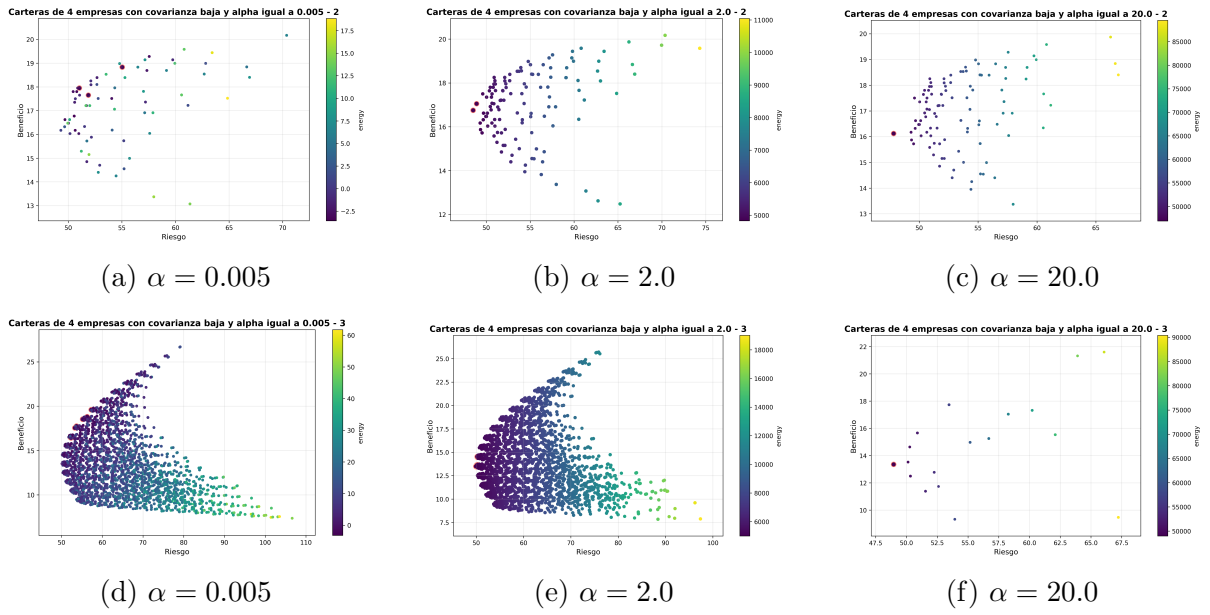
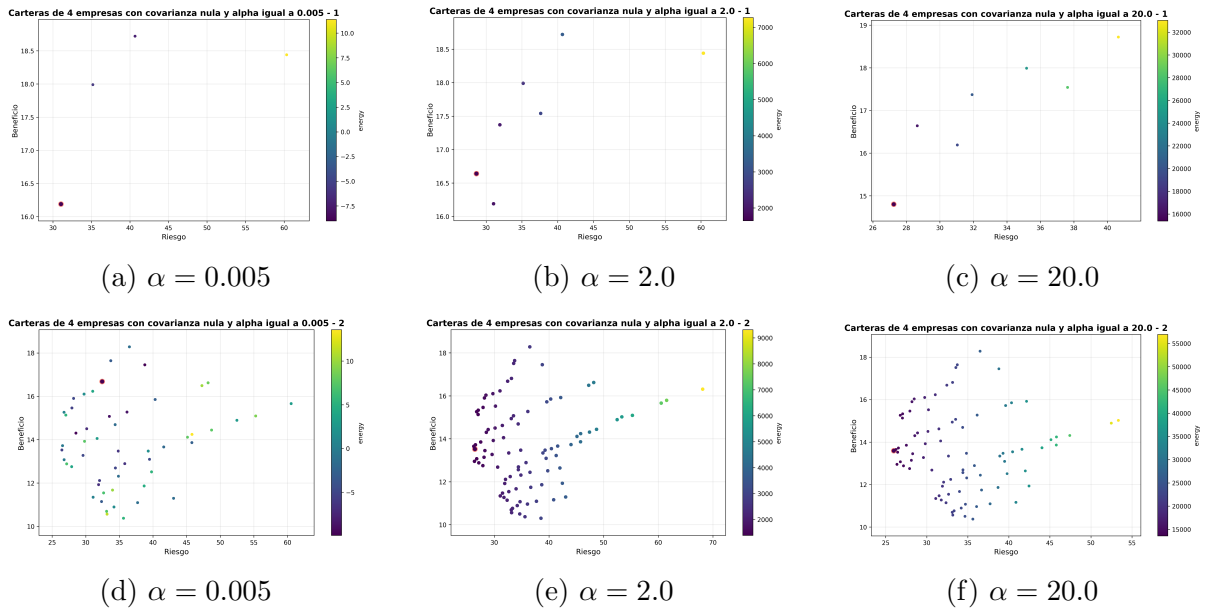


Figura 11: Soluciones de carteras de 4 empresas con covarianza baja, para distintos valores de  $\alpha$ .



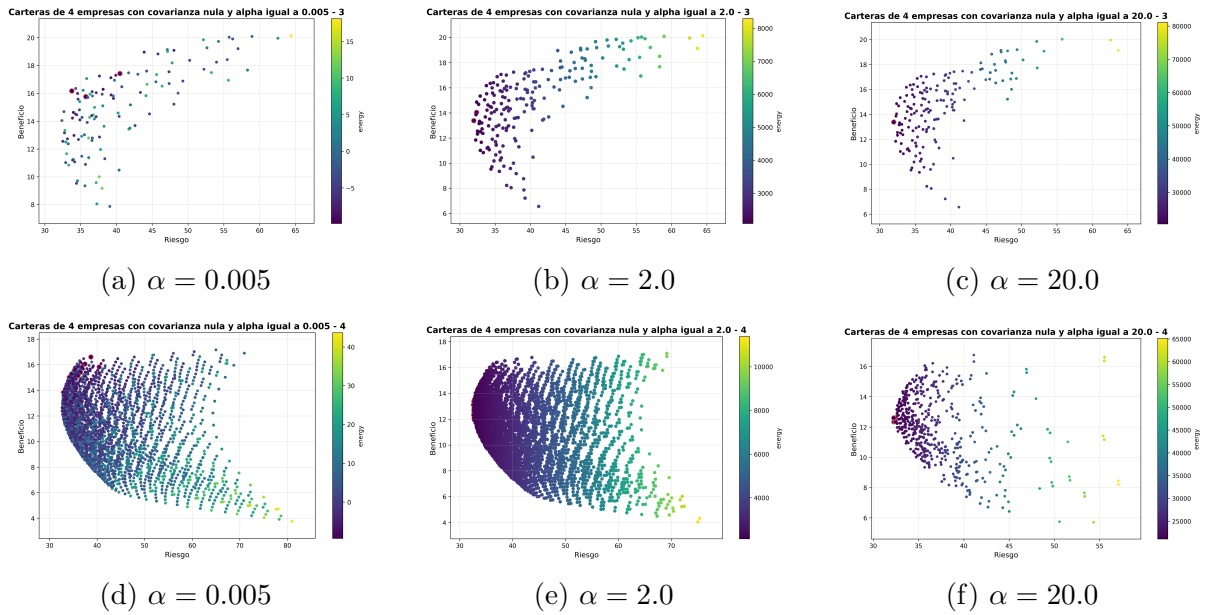


Figura 13: Soluciones de carteras de 4 empresas con covarianza nula, para distintos valores de  $\alpha$ .

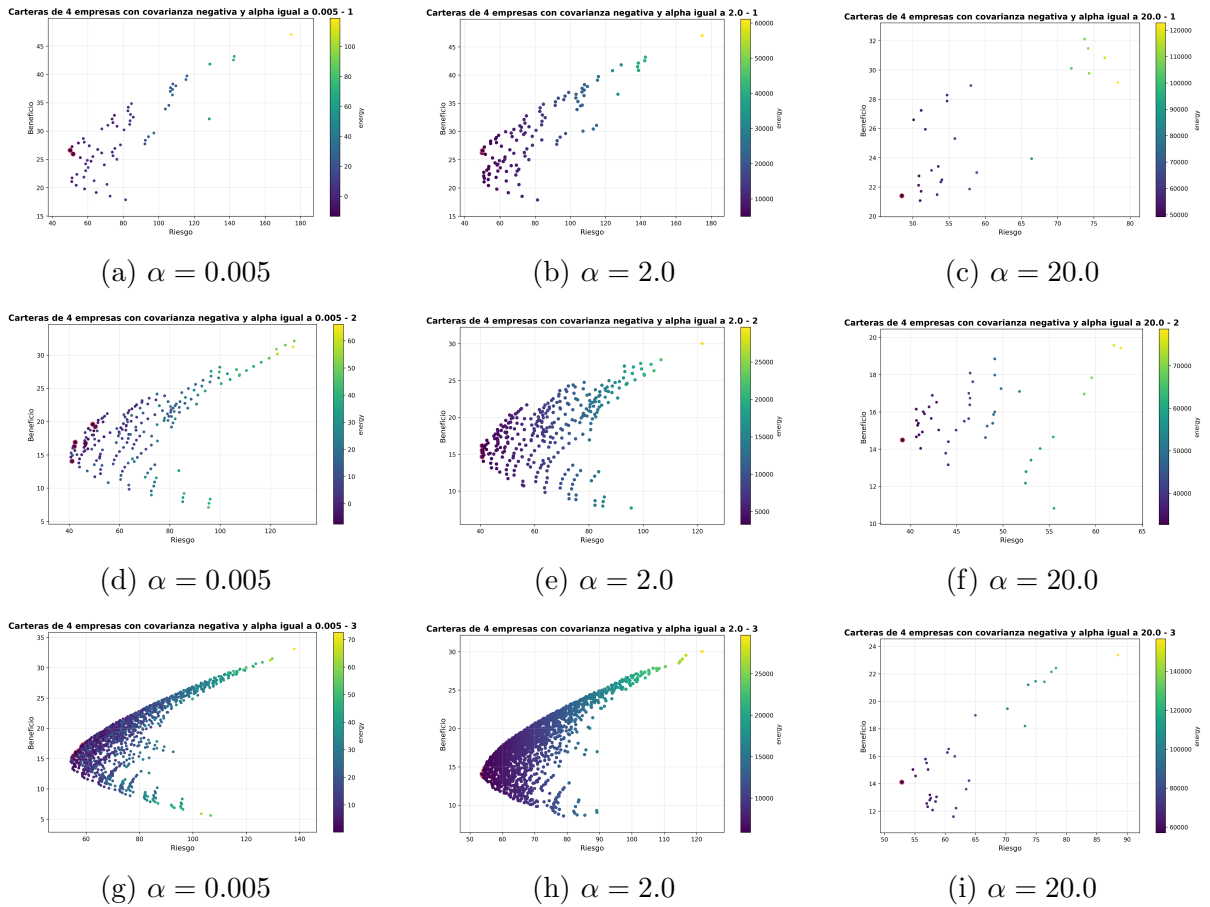


Figura 14: Soluciones de carteras de 4 empresas con covarianza negativa, para distintos valores de  $\alpha$ .

## C.2. Carteras de 6 empresas

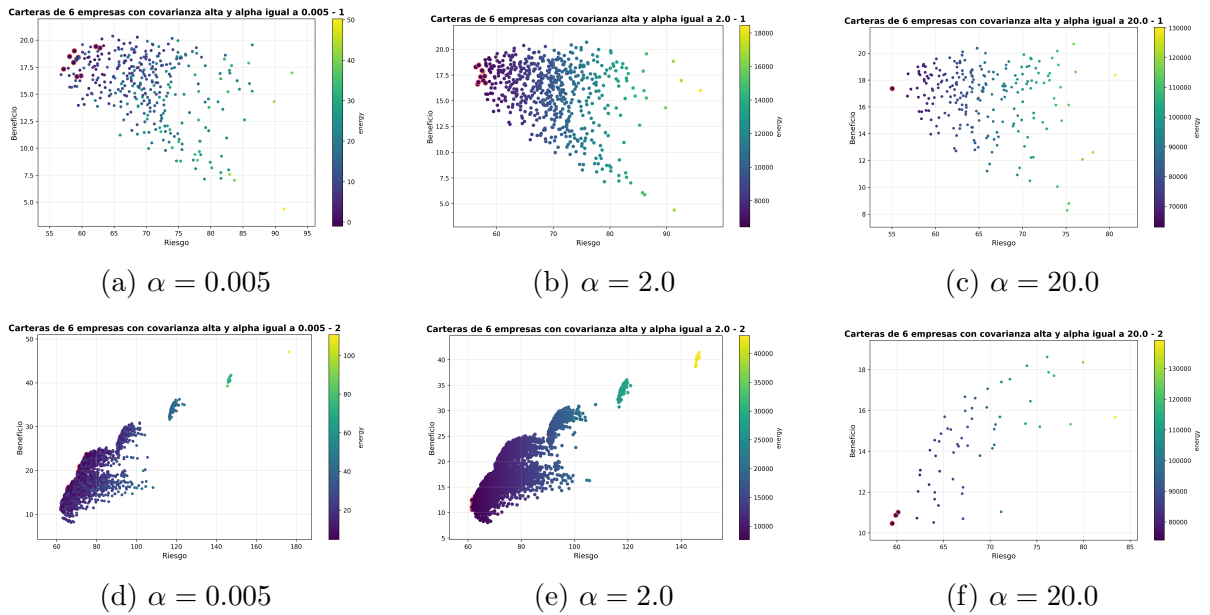


Figura 15: Soluciones de carteras de 6 empresas con covarianza alta, para distintos valores de  $\alpha$ .

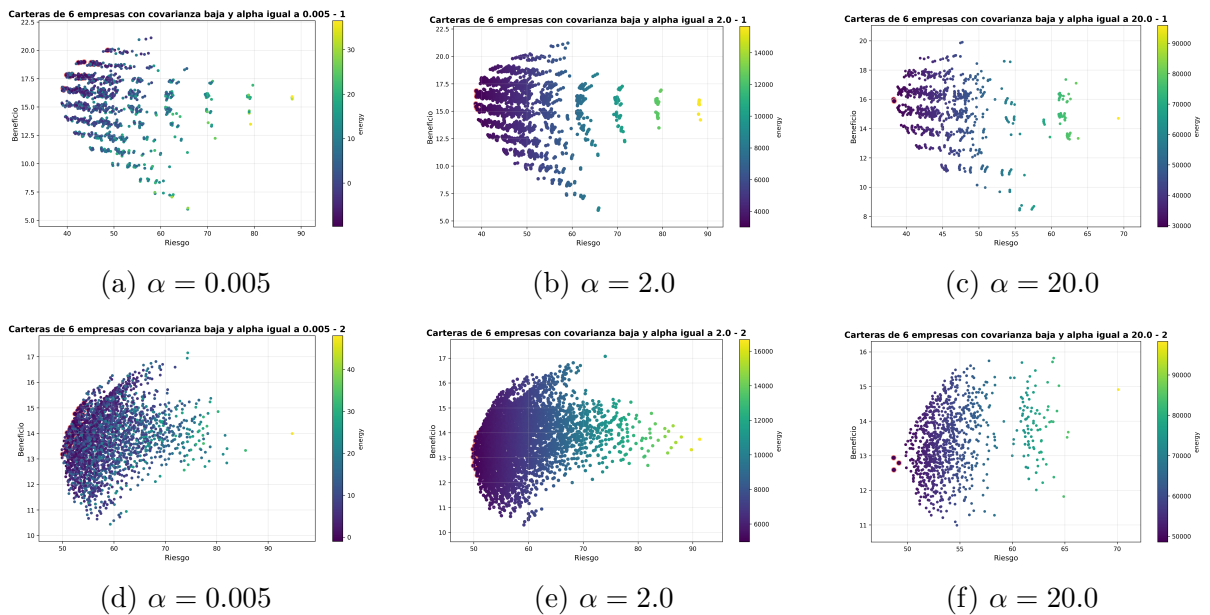


Figura 16: Soluciones de carteras de 6 empresas con covarianza baja, para distintos valores de  $\alpha$ .

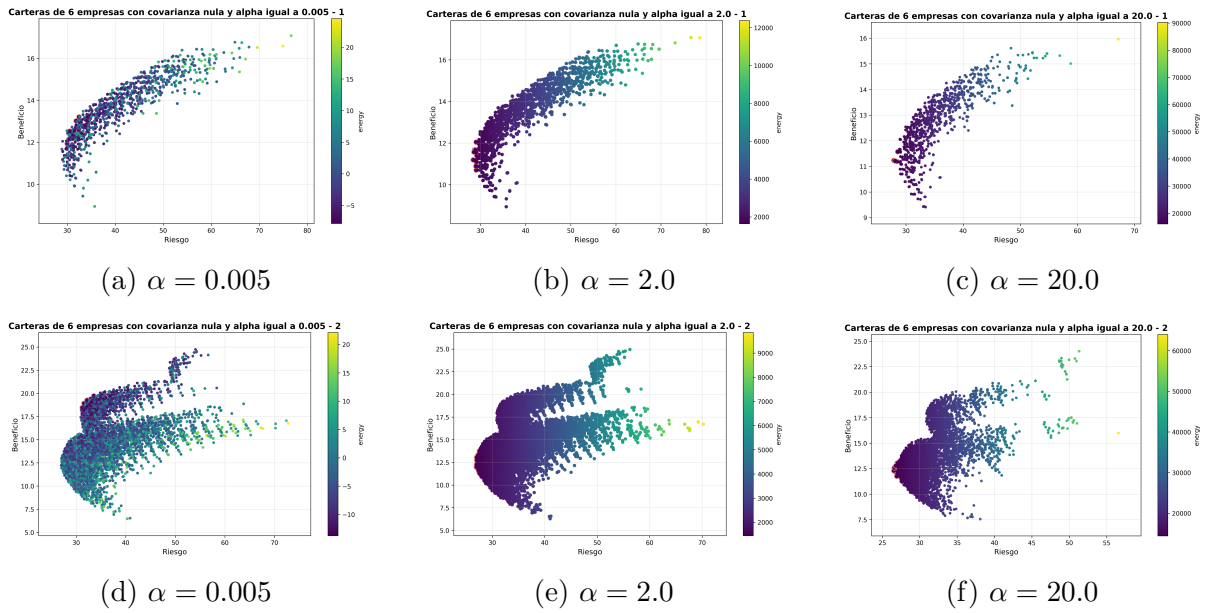


Figura 17: Soluciones de carteras de 6 empresas con covarianza nula, para distintos valores de  $\alpha$ .

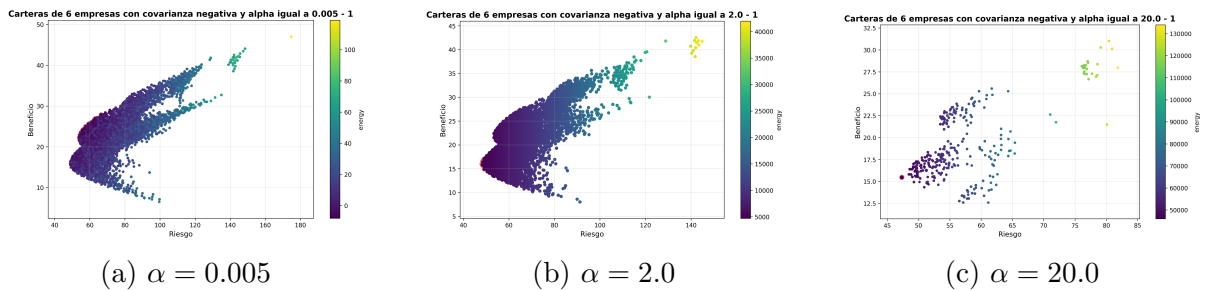


Figura 18: Soluciones de carteras de 6 empresas con covarianza negativa, para distintos valores de  $\alpha$ .

## D. Histogramas de los campos locales y los acoplamientos.

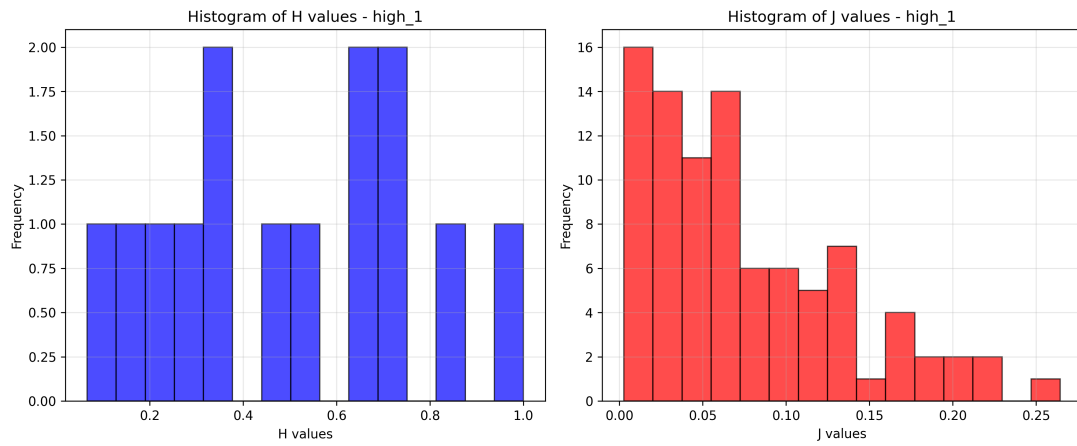


Figura 19: Histogramas de los valores de los campos locales  $h_{i,k}$  (izquierda) y de los acoplamientos  $J_{i,j,k,l}$  (derecha) correspondientes a la primera cartera con covarianza alta.

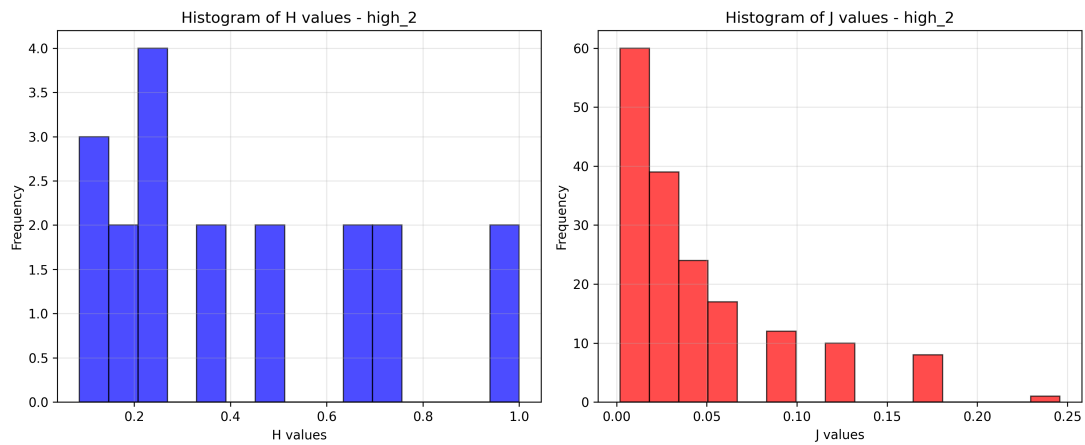


Figura 20: Histogramas de los valores de los campos locales  $h_{i,k}$  (izquierda) y de los acoplamientos  $J_{i,j,k,l}$  (derecha) correspondientes a la segunda cartera con covarianza alta.

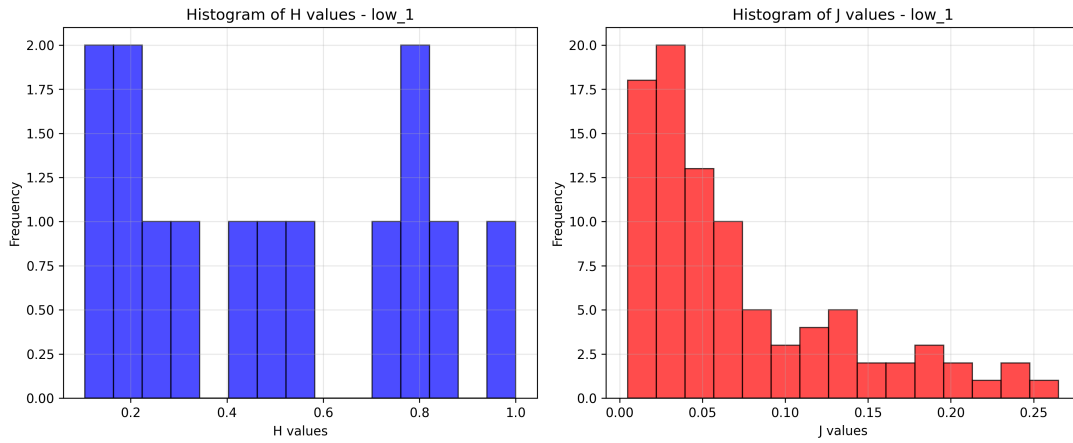


Figura 21: Histogramas de los valores de los campos locales  $h_{i,k}$  (izquierda) y de los acoplamientos  $J_{i,j,k,l}$  (derecha) correspondientes a la primera cartera con covarianza baja.

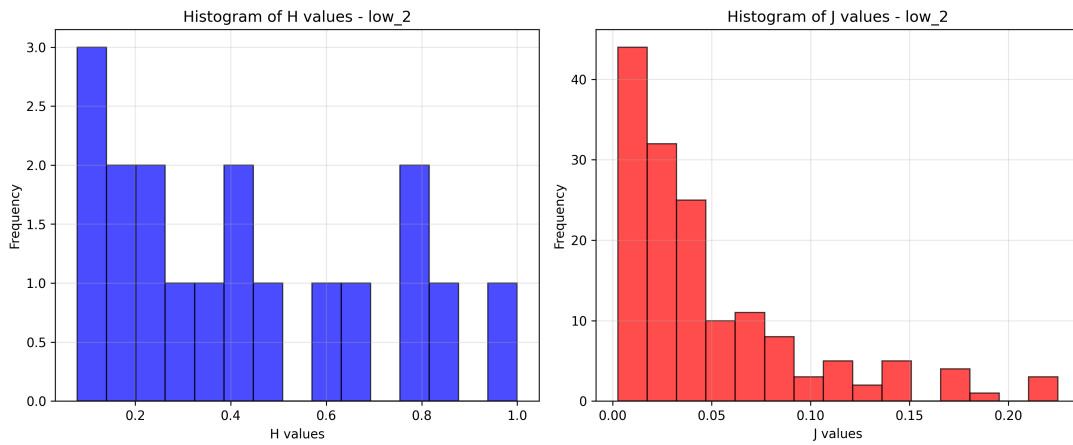


Figura 22: Histogramas de los valores de los campos locales  $h_{i,k}$  (izquierda) y de los acoplamientos  $J_{i,j,k,l}$  (derecha) correspondientes a la segunda cartera con covarianza baja.

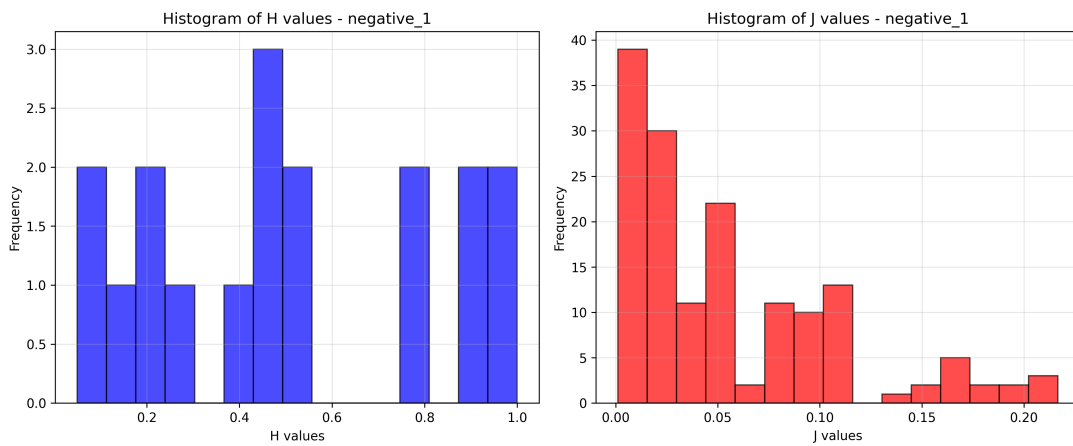


Figura 23: Histogramas de los valores de los campos locales  $h_{i,k}$  (izquierda) y de los acoplamientos  $J_{i,j,k,l}$  (derecha) correspondientes a la primera cartera con rendimientos negativos.

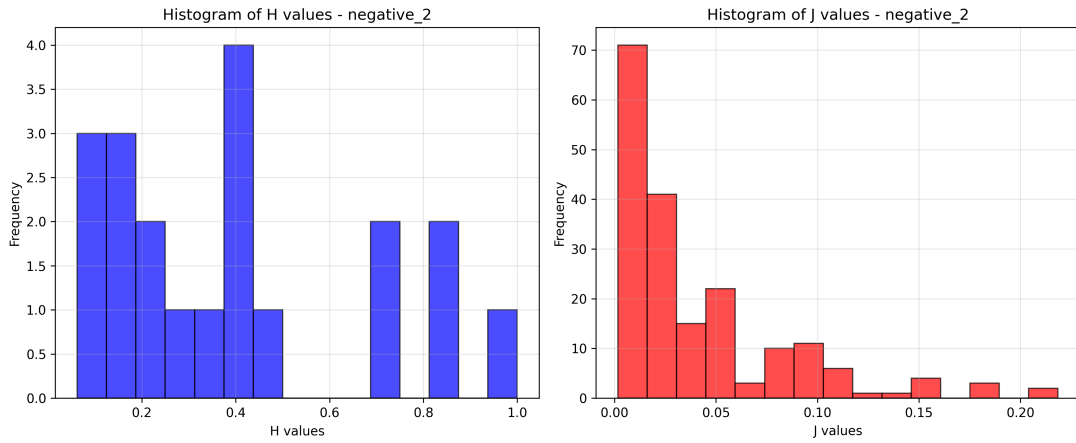


Figura 24: Histogramas de los valores de los campos locales  $h_{i,k}$  (izquierda) y de los acoplamientos  $J_{i,j,k,l}$  (derecha) correspondientes a la segunda cartera con rendimientos negativos.

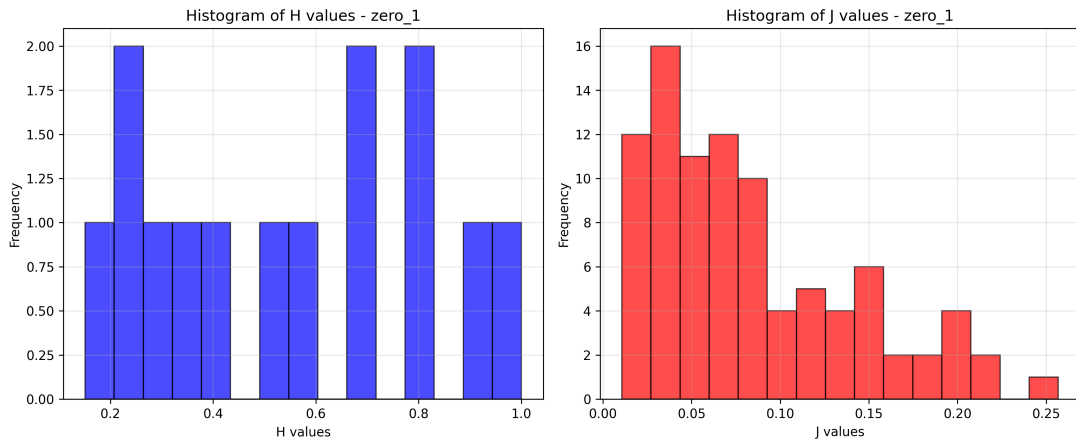


Figura 25: Histogramas de los valores de los campos locales  $h_{i,k}$  (izquierda) y de los acoplamientos  $J_{i,j,k,l}$  (derecha) correspondientes a la primera cartera con rendimientos nulos.

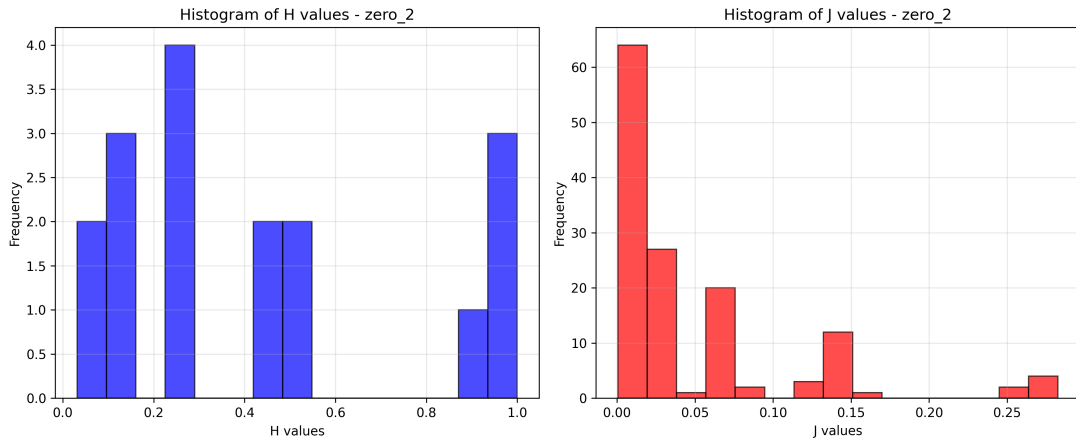


Figura 26: Histogramas de los valores de los campos locales  $h_{i,k}$  (izquierda) y de los acoplamientos  $J_{i,j,k,l}$  (derecha) correspondientes a la segunda cartera con rendimientos nulos.

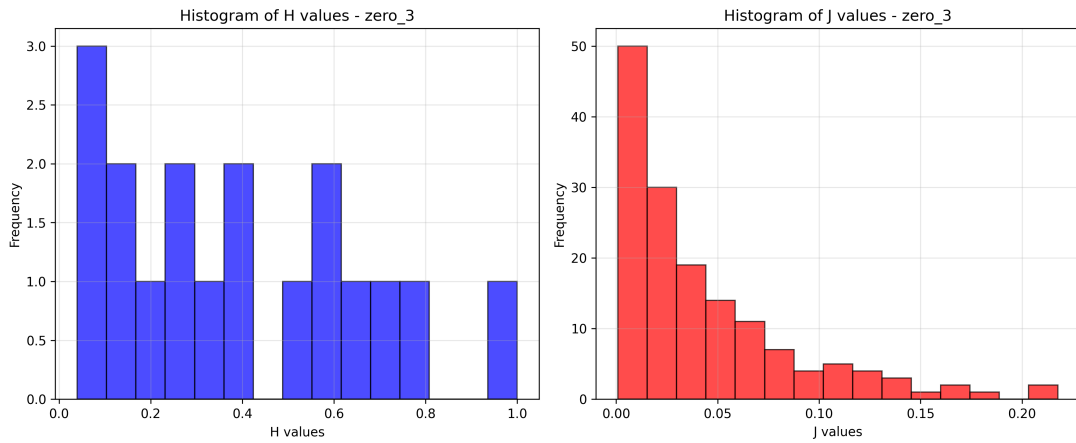
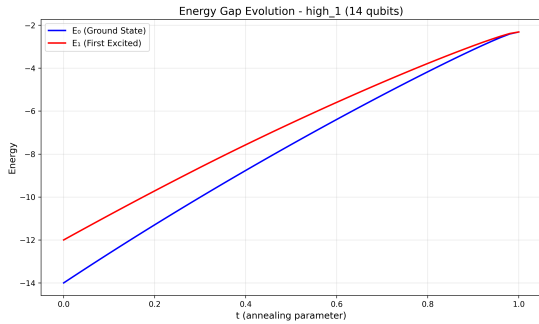
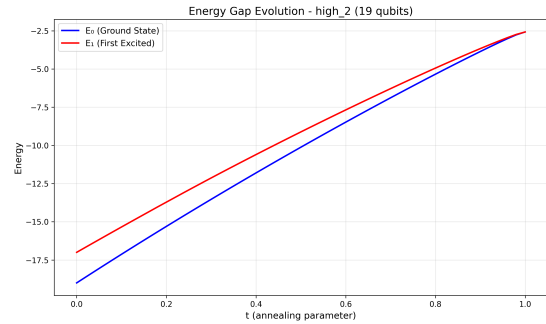


Figura 27: Histogramas de los valores de los campos locales  $h_{i,k}$  (izquierda) y de los acoplamientos  $J_{i,j,k,l}$  (derecha) correspondientes a la tercera cartera con rendimientos nulos.

## E. Evolución de la energía en el annealing cuántico con pasos distribuidos uniformemente en la escala lineal

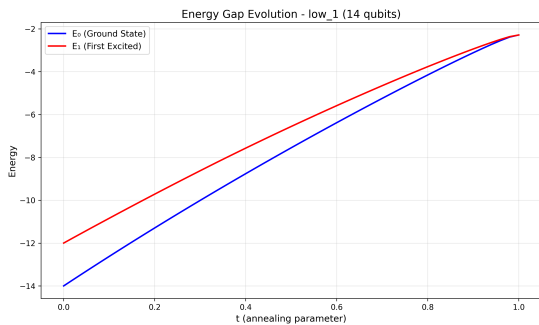


(a) 14 bits

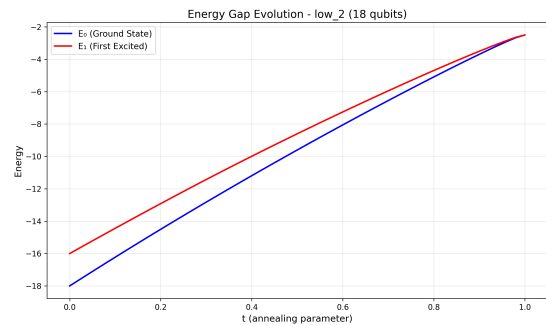


(b) 19 bits

Figura 28: Evolución del estado fundamental y el primer excitado para carteras con covarianza alta.

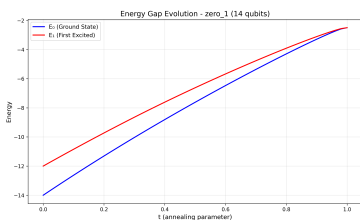


(a) 14 bits

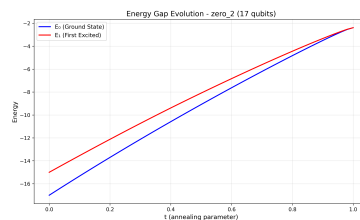


(b) 18 bits

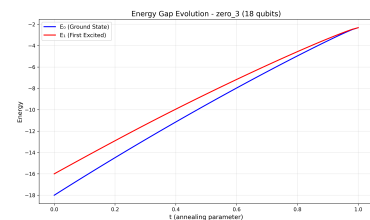
Figura 29: Evolución del estado fundamental y el primer excitado para carteras con covarianza baja.



(a) 14 bits

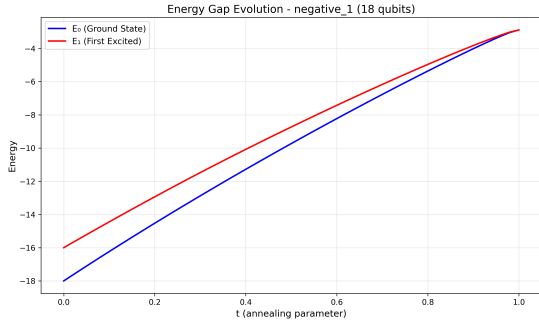


(b) 16 bits

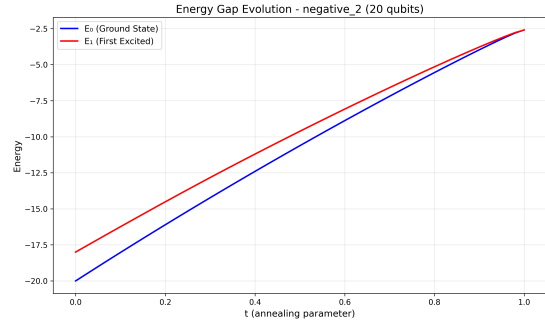


(c) 18 bits

Figura 30: Evolución del estado fundamental y el primer excitado para carteras con covarianza nula.



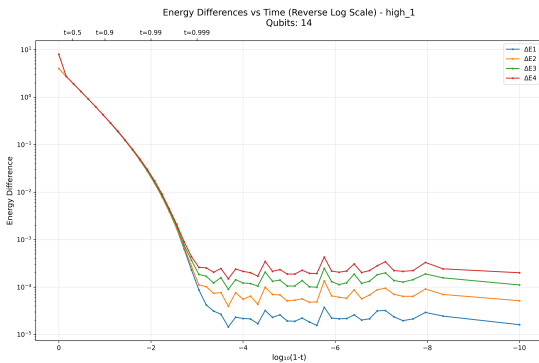
(a) 18 bits



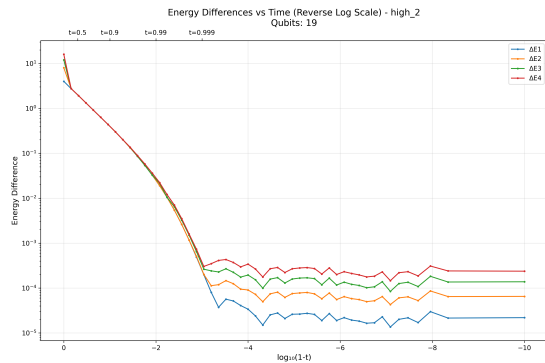
(b) 20 bits

Figura 31: Evolución del estado fundamental y el primer excitado para carteras con covarianza negativa.

## F. Evolución del gap de energías entre los primeros excitados y el nivel fundamental con pasos tentativos distribuidos uniformemente en la escala logarítmica

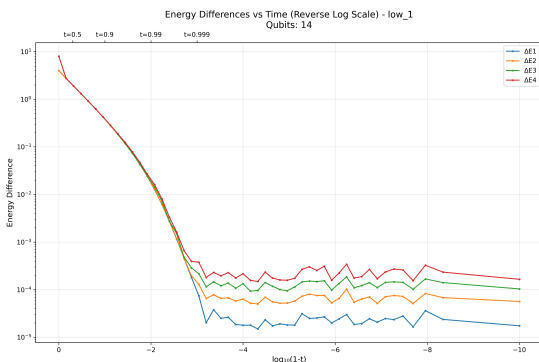


(a) 14 bits

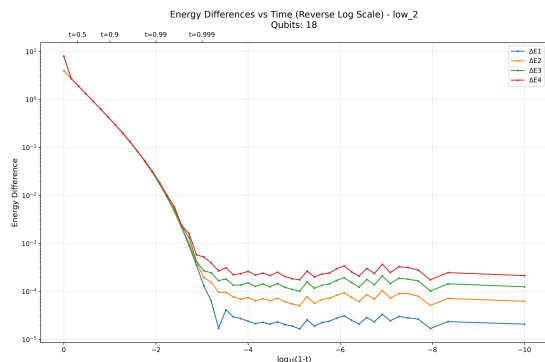


(b) 19 bits

Figura 32: Evolución del gap para carteras con covarianza alta.

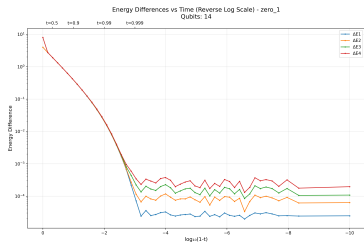


(a) 14 bits

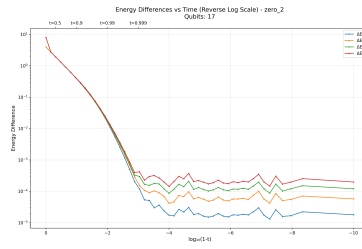


(b) 18 bits

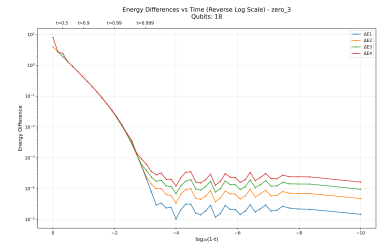
Figura 33: Evolución del gap para carteras con covarianza baja.



(a) 14 bits

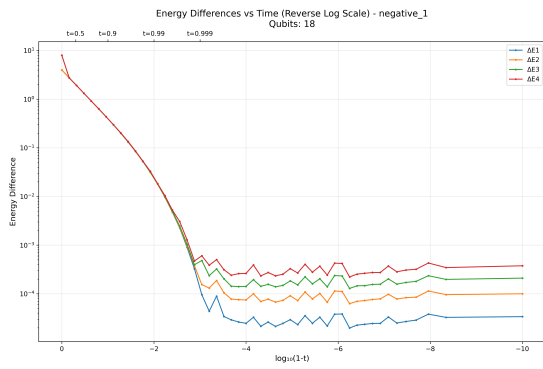


(b) 16 bits

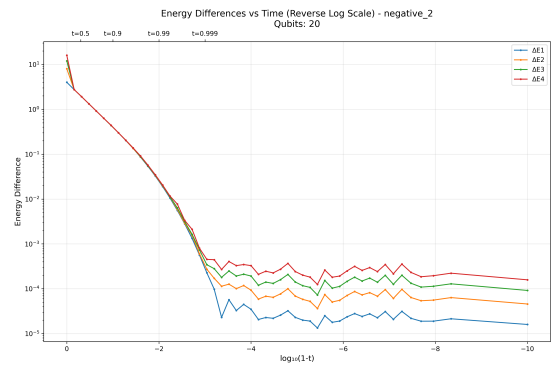


(c) 18 bits

Figura 34: Evolución del gap para carteras con covarianza nula.



(a) 18 bits



(b) 20 bits

Figura 35: Evolución del gap para carteras con covarianza negativa.