



**COMILLAS**  
**UNIVERSIDAD PONTIFICIA**



ICADE Facultad de Empresariales

# **ESTADO DEL ARTE DE LA GESTIÓN DE CARTERAS**

Autor: Marta Carpi Pérez

Director: Gabriel Rodríguez Garnica

MADRID | Noviembre 2024

## **RESUMEN:**

En un mundo financiero caracterizado por su dinamismo y complejidad, la gestión de carteras se ha convertido en una disciplina crucial para inversores y gestores de fondos. La búsqueda constante de maximizar los rendimientos ajustados por riesgo ha llevado al desarrollo de diversos modelos y técnicas de optimización de carteras. Desde la formulación pionera de Harry Markowitz en la década de 1950, que introdujo la teoría moderna de carteras basada en la diversificación y la compensación entre riesgo y rendimiento, hasta los modelos más recientes que integran enfoques bayesianos y métodos de simulación, la evolución en este campo ha sido notable.

Sin embargo, con la creciente volatilidad de los mercados financieros y la aparición de eventos económicos impredecibles, los modelos tradicionales de gestión de carteras enfrentan limitaciones significativas. En este contexto, las metaheurísticas han emergido como una solución prometedora. Estos algoritmos, inspirados en procesos naturales y biológicos, ofrecen una flexibilidad y capacidad de adaptación que supera a las técnicas convencionales. Al combinar exploración global y explotación local, las metaheurísticas pueden encontrar soluciones óptimas en espacios de búsqueda complejos y dinámicos.

Este trabajo de fin de grado explora y compara dos enfoques distintos para la optimización de carteras: el modelo de media-varianza de Markowitz y los algoritmos genéticos, una clase de metaheurísticas. La primera sección del estudio se centra en la fundamentación teórica de cada método, destacando sus principios, ventajas y limitaciones. Posteriormente, se implementan y aplican ambos modelos a una selección de activos del índice IBEX35, obteniendo resultados que se analizan detalladamente.

A través de este análisis comparativo, se busca no solo evaluar la efectividad de cada enfoque en términos de rendimiento y riesgo, sino también proporcionar una perspectiva sobre cómo los métodos metaheurísticos pueden complementar y potenciar las estrategias tradicionales de gestión de carteras. Los resultados de este estudio no solo ofrecen una comprensión más profunda de las técnicas de optimización actuales, sino que también plantean nuevas preguntas y desafíos para futuras investigaciones en este campo.

Así, este trabajo pretende aportar valor tanto a académicos como a profesionales del sector financiero, ofreciendo una visión integral y actualizada de las herramientas disponibles para la gestión eficiente de carteras en un entorno económico cada vez más complejo y volátil. Con este propósito, se invita al lector a adentrarse en las siguientes páginas, donde se desglosan y analizan los métodos, resultados y conclusiones de este apasionante estudio.

**PALABRAS CLAVE:**

- Gestión de carteras
- Optimización de carteras
- Algoritmos genéticos
- Modelo de Markowitz
- Métodos metaheurísticos
- Rendimiento ajustado por riesgo
- Diversificación

## **ABSTRACT:**

In a dynamic and complex financial world, portfolio management has become a crucial discipline for investors and fund managers. The constant quest to maximize risk-adjusted returns has led to the development of various portfolio optimization models and techniques. From Harry Markowitz's pioneering formulation in the 1950s, which introduced modern portfolio theory based on diversification and the trade-off between risk and return, to more recent models that integrate Bayesian approaches and simulation methods, the evolution in this field has been remarkable.

However, with the increasing volatility of financial markets and the emergence of unpredictable economic events, traditional portfolio management models face significant limitations. In this context, metaheuristics have emerged as a promising solution. These algorithms, inspired by natural and biological processes, offer flexibility and adaptability that surpass conventional techniques. By combining global exploration and local exploitation, metaheuristics can find optimal solutions in complex and dynamic search spaces.

This final degree project explores and compares two distinct approaches to portfolio optimization: the Markowitz mean-variance model and genetic algorithms, a class of metaheuristics. The first section of the study focuses on the theoretical foundation of each method, highlighting their principles, advantages, and limitations. Subsequently, both models are implemented and applied to a selection of assets from the IBEX35 index, obtaining results that are analysed in detail.

Through this comparative analysis, the aim is not only to evaluate the effectiveness of each approach in terms of return and risk but also to provide a perspective on how metaheuristic methods can complement and enhance traditional portfolio management strategies. The results of this study not only offer a deeper understanding of current optimization techniques but also pose new questions and challenges for future research in this field.

Thus, this work aims to provide value to both academics and professionals in the financial sector, offering a comprehensive and updated view of the tools available for efficient portfolio management in an increasingly complex and volatile economic environment. With

this purpose, the reader is invited to delve into the following pages, where the methods, results, and conclusions of this exciting study are broken down and analysed.

**KEY WORDS:**

- Portfolio management
- Portfolio optimization
- Genetic algorithms
- Markowitz model
- Metaheuristic methods
- Risk-adjusted return
- Diversification

# **ÍNDICE**

<b>1. CONCEPTOS CLAVE</b>	<b>7</b>
1.1 La Hipótesis de Eficiencia del Mercado y sus Implicaciones	7
1.2 La Gestión de Carteras: Perfil del Inversor, Estrategias y Modelos	8
1.3 La Declaración de Política de Inversión (IPS) y su Relevancia en la Gestión de Carteras	10
<b>2. TEORÍAS TRADICIONALES DE GESTIÓN DE CARTERAS</b>	<b>13</b>
2.1 Markowitz	13
2.2 Black-Litterman	17
2.3 Montecarlo	20
<b>3. TEORÍAS METAHEURÍSTICAS DE GESTIÓN DE CARTERAS</b>	<b>22</b>
3.1 Algoritmos Genéticos (AGs)	23
3.2 Recocido Simulado (RS) SEGUIR CITANDO	25
3.3 Búsqueda de Vecindario Variable	27
3.4 Búsqueda Tabú (TB)	29
3.5 Colonia de Hormigas (ACO)	31
3.6 Optimización por Enjambre de Partículas (PSO)	32
3.7 Optimización por Colonia de Lobos (WSA)	34
3.8 Algoritmo de Murciélagos Virtuales (AMV)	35
3.9 Optimización por Búsqueda de Gravedad	37
<b>4. IMPLEMENTACIÓN Y COMPARACIÓN DE MÉTODOS DE OPTIMIZACIÓN DE CARTERAS</b>	<b>40</b>
4.1 Importación de Datos y Selección de la Cartera Diversificada	40
4.2 Visualización de los Precios de los Activos	42
4.3 Preparación de Datos y Cálculo de Retornos	43
4.4 Visualización de Estadísticas	44
4.5 Definición de Funciones Paramétricas de Cartera (Markowitz)	50
4.6 Optimización de Markowitz	51
4.7 Definición de Funciones Paramétricas de Cartera (Algoritmos Genéticos)	53
4.8 Optimización con Algoritmos Genéticos	54
4.8 Comparación de Resultados y Visualización	57
<b>5. CONCLUSIÓN</b>	<b>61</b>
<b>6. ANEXOS</b>	<b>63</b>
<b>7. BIBLIOGRAFÍA</b>	<b>95</b>

## 1. CONCEPTOS CLAVE

### 1.1 La Hipótesis de Eficiencia del Mercado y sus Implicaciones

La hipótesis de eficiencia del mercado sostiene que los precios de los valores reflejan completamente toda la información disponible de forma pública en el mercado. Para que esta versión fuerte de la hipótesis sea válida, es necesario que el precio de la información y de negociación, es decir, los costes asociados a hacer que los precios reflejen dicha información sean siempre nulos (Grossman & Stiglitz, 1980).

No obstante, una versión más moderada y económicamente más razonable de la hipótesis de eficiencia del mercado afirma que los precios reflejan la información hasta el punto en que los beneficios marginales de actuar sobre dicha información (las ganancias obtenibles) no superen los costes marginales asociados (Jensen (Deceased) & C, 1978).

En 1969, Eugene Fama distingue tres grados de eficiencia en los mercados de capitales: débil, semifuerte y fuerte. La eficiencia débil se da cuando los precios de los títulos reflejan toda la información contenida en los precios y volúmenes históricos, haciendo ineficaz el análisis técnico para obtener rendimientos superiores. La eficiencia semifuerte se alcanza cuando los precios también incorporan toda la información pública disponible, como anuncios de resultados y divisiones de acciones, impidiendo que los inversores superen consistentemente al mercado mediante análisis fundamental. Finalmente, la eficiencia fuerte se logra cuando los precios reflejan toda la información, incluida la privada y privilegiada a la que solo tienen acceso los insiders de la empresa, eliminando cualquier posibilidad de obtener rendimientos anormales basados en cualquier tipo de información (Fama, 1965).

En este contexto, la predictibilidad de los rendimientos de las acciones a partir de los rendimientos por dividendos (D/P) no constituye por sí misma una evidencia a favor o en contra de la eficiencia del mercado. En un mercado eficiente, la capacidad predictiva del D/P sugiere que los precios son altos en relación con los dividendos cuando las tasas de descuento y los rendimientos esperados son bajos, y viceversa. Contrariamente, en un escenario de burbujas irracionales, un bajo D/P indicaría precios de acciones irracionalmente altos que se ajustarían de manera predecible hacia valores fundamentales. Para discernir si la capacidad predictiva de los rendimientos por dividendos se debe a una variación racional en los

rendimientos esperados o a las burbujas irracionales, es necesario considerar información adicional. Sin embargo, incluso con dicha información, la cuestión sigue siendo ambigua (Fama, 1991).

La eficiencia del mercado debe ser evaluada juntamente con un modelo de equilibrio, es decir, un modelo de valoración de activos. Este enfoque, que fue central en los desarrollos de 1970, indica que solo podemos comprobar si la información se refleja adecuadamente en los precios dentro del contexto de un modelo de valoración que defina qué significa "adecuadamente". Es lo que Richard Roll (1997) denominó el problema de la "joint hypothesis". Por consiguiente, cuando se encuentra evidencia anómala sobre el comportamiento de los rendimientos, resulta ambiguo determinar si esta se debe a la ineficiencia del mercado o a un modelo de equilibrio del mercado deficiente (Fama, 1970).

Más tarde, Fama y French argumentan que las variaciones sistemáticas en los rendimientos esperados a lo largo del tiempo son racionales, y que la variación en los rendimientos esperados, rastreada por D/P y la prima de riesgo, es consistente con los modelos modernos de valoración de activos intertemporales, así como con las teorías de suavización del consumo de Friedman (1957) y Modigliani y Brumberg (1955). Por lo tanto, la investigación sobre la eficiencia del mercado y las variaciones sistemáticas en los rendimientos esperados refuerza la comprensión del comportamiento de los rendimientos de los valores y sostiene que el estudio de la eficiencia del mercado sigue siendo relevante y científicamente útil (Fama & French, 1989).

## **1.2 La Gestión de Carteras: Perfil del Inversor, Estrategias y Modelos**

La gestión de carteras ha sido un área de interés constante para los inversores, quienes buscan identificar combinaciones óptimas de activos financieros en los que invertir. Este proceso implica no sólo el análisis exhaustivo de los activos disponibles, sino también la toma de decisiones estratégicas sobre la inversión en un grupo específico de activos durante un período determinado (Carrasco, 2006). Este enfoque es esencial para los directores generales de grandes organizaciones diversificadas, quienes deben decidir constantemente dónde asignar el nuevo capital y cómo distribuir los recursos organizativos entre las distintas unidades de negocio. Para asistir en estas decisiones, se han desarrollado modelos de cartera (Harrison & John, 2002).

A pesar de su amplia implementación, las técnicas de gestión de carteras no están exentas de críticas. Es fundamental reconocer que estos métodos no son soluciones definitivas y no deben reemplazar otros tipos de análisis estratégicos formales (Harrison & John, 2002). Más bien, deben considerarse como herramientas complementarias que proporcionan perspectivas adicionales en el proceso de toma de decisiones.

La teoría de la gestión de carteras experimentó un avance significativo con la presentación de Harry M. Markowitz en 1952. Markowitz estructuró la gestión de carteras de inversión en torno al binomio de rentabilidad y riesgo, estableciendo que, para obtener ciertos beneficios, la gestión de la cartera debe asumir determinados riesgos, ya que es imposible alterar uno de estos factores sin afectar al otro. Este enfoque, aunque matemáticamente complejo debido al gran número de valores que pueden necesitar ser controlados, ha sido fundamental para el desarrollo de estrategias de inversión modernas (Markowitz, 1952).

Un buen inversor no se distingue únicamente por encontrar oportunidades de alta rentabilidad, sino por identificar aquellas que se ajusten a sus necesidades financieras con el menor riesgo posible. Esta necesidad de encontrar un equilibrio adecuado provoca que muchos ahorradores desconfíen de las inversiones financieras, viéndolas como productos de alto riesgo y difíciles de gestionar. En el pasado, muchos preferían invertir en inmuebles, considerándolos un activo refugio. Sin embargo, con la caída de los precios y la falta de liquidez tras el estallido de la burbuja inmobiliaria, se encontraron con diversas dificultades.

La aversión o tolerancia al riesgo varía entre los inversores y afecta sus decisiones de inversión. Se pueden definir tres perfiles de riesgo:

- 1. Perfil de riesgo bajo o conservador:** Estos inversores buscan preservar su capital y tienen baja tolerancia a las pérdidas. Están dispuestos a aceptar menores rendimientos a cambio de la seguridad de sus ahorros, optando por fondos de renta fija o fondos monetarios de corto plazo.
- 2. Perfil de riesgo medio o moderado:** Estos inversores pueden asumir ciertos niveles de pérdida a cambio de la posibilidad de obtener mayores rendimientos que el perfil conservador. Los fondos mixtos, que combinan renta fija y variable, son los más adecuados para ellos.

- 3. Perfil de riesgo alto o arriesgado:** Dispuestos a asumir grandes pérdidas con el objetivo de obtener altos rendimientos, estos inversores prefieren fondos de renta variable en sectores específicos o mercados emergentes.

La gestión de carteras es un campo dinámico que combina teoría financiera y práctica estratégica para optimizar las decisiones de inversión. A pesar de las limitaciones y críticas, las técnicas de gestión de carteras proporcionan un marco estructurado que ayuda a los inversores y directivos a equilibrar el riesgo y la rentabilidad de sus inversiones.

Continuando con el análisis del mercado eficiente, la gestión de carteras desempeña un papel crucial en la aplicación práctica de estas teorías. La habilidad para identificar y gestionar adecuadamente el riesgo y el rendimiento a través de una combinación óptima de activos es esencial para aprovechar al máximo las oportunidades que ofrecen los mercados financieros. Por lo tanto, comprender las diversas técnicas y modelos de gestión de carteras es fundamental para cualquier inversor que busque maximizar sus retornos ajustados por riesgo en un entorno de mercado eficiente o ineficiente.

### **1.3 La Declaración de Política de Inversión (IPS) y su Relevancia en la Gestión de Carteras**

La declaración de política de inversión (IPS), 'Investment Policy Statement' en inglés, es crucial para guiar las decisiones de inversión del cliente, alineando objetivos, tolerancia al riesgo y limitaciones de inversión. Proporciona directrices para la construcción y revisión de la cartera, permitiendo al gestor de inversiones crear una cartera adecuada tras su elaboración. Las expectativas del mercado de capitales se calculan utilizando los rendimientos esperados de las clases de activos, desviaciones estándar y correlaciones, basándose en modelos de valoración, estimaciones históricas y análisis económico. La asignación estratégica de activos expone al inversor a riesgos sistemáticos, responsables de la variación a largo plazo del valor de la cartera. Incluye clases de activos tradicionales como efectivo, renta variable, renta fija e inmuebles, y más recientemente, capital riesgo, materias primas y "hedge funds" (Institute, 2017).

Por otro lado, la asignación táctica de activos implica decisiones a corto plazo para aprovechar previsiones de rendimientos superiores, ajustando temporalmente la ponderación

de las clases de activos. La selección de valores busca rendimientos superiores a los del índice de referencia mediante la selección específica de acciones anticipadas a superar dicho índice. Las decisiones de inversión incrementan la incertidumbre sobre los rendimientos, por lo que la política de inversión debe establecer límites claros de riesgo y beneficios para estas actividades (Institute, 2017)..

Integrar una IPS bien estructurada en la gestión de carteras es fundamental para alinear los objetivos del inversor con la estrategia de inversión. La IPS no solo proporciona una hoja de ruta clara para la toma de decisiones, sino que también establece un marco para evaluar el desempeño de la cartera y realizar ajustes necesarios en respuesta a las cambiantes condiciones del mercado. Este enfoque asegura que las decisiones de inversión se realicen de manera coherente con los objetivos financieros a largo plazo del inversor y su tolerancia al riesgo.

En el contexto de la gestión de carteras, la IPS actúa como un documento vivo que se revisa y ajusta periódicamente para reflejar cambios en las circunstancias del inversor, las condiciones del mercado y la evolución de las oportunidades de inversión. Esto es especialmente relevante en un entorno financiero dinámico, donde la capacidad de adaptarse rápidamente a nuevas realidades es crucial para mantener una ventaja competitiva.

La importancia de la IPS se amplifica cuando se consideran las teorías tradicionales de gestión de carteras, como los modelos de Markowitz y Black-Litterman, que a continuación explicaré en este trabajo, que proporcionan marcos teóricos robustos para la optimización de la asignación de activos. La IPS, al establecer los parámetros dentro de los cuales se toman las decisiones de inversión, permite que estas teorías se apliquen de manera práctica y efectiva, garantizando que las estrategias de inversión sean coherentes con los objetivos del inversor.

Además, la IPS facilita la integración de técnicas de gestión de carteras más avanzadas y metaheurísticas, como los algoritmos genéticos y la optimización por enjambre de partículas, que pueden mejorar significativamente la eficiencia y la efectividad de las decisiones de inversión. Al definir claramente los objetivos y restricciones de inversión, la IPS permite a los gestores de carteras aplicar estas técnicas de manera alineada con las expectativas y necesidades del cliente.

En resumen, la IPS es una herramienta esencial en la gestión de carteras, proporcionando un marco estructurado para la toma de decisiones de inversión que es fundamental para alinear los objetivos del inversor con la estrategia de inversión. Su importancia se extiende a la aplicación de tanto teorías tradicionales como técnicas avanzadas de optimización, asegurando que las decisiones de inversión sean coherentes, bien fundamentadas y adaptables a las cambiantes condiciones del mercado.

## 2. TEORÍAS TRADICIONALES DE GESTIÓN DE CARTERAS

Tras haber abordado los fundamentos de la eficiencia del mercado y su implicación en la gestión de carteras, así como la importancia de la declaración de política de inversión (IPS) para guiar las decisiones de inversión, es fundamental profundizar en las teorías tradicionales de gestión de carteras. Estas teorías proporcionan los marcos teóricos y metodológicos que permiten a los inversores optimizar la asignación de activos y gestionar el riesgo de manera eficiente. A continuación, se presentarán los modelos de Markowitz, Black-Litterman y simulación de Montecarlo, destacando sus enfoques y contribuciones a la gestión de carteras.

### 2.1 Markowitz

El proceso de selección de una cartera de inversión se puede dividir en dos etapas principales. La primera etapa implica la observación y la experiencia, concluyendo en creencias sobre el rendimiento futuro de los valores disponibles. La segunda etapa comienza con estas creencias y termina con la elección de la cartera. En esta segunda etapa, inicialmente se considera la regla de que el inversor debe maximizar el valor descontado de los rendimientos esperados o anticipados. Sin embargo, esta regla se rechaza tanto como una hipótesis para explicar el comportamiento de inversión, como una máxima para guiarlo. En lugar de esto, se propone que el inversor debe considerar el rendimiento esperado como algo deseable y la variabilidad de los rendimientos como algo indeseable, una perspectiva con fundamentos sólidos que se ilustra en el Gráfico 1 mediante la representación de la "Curva de la Frontera Eficiente y la Cartera de Mínima Varianza" (ver Gráfico 1). Este gráfico muestra la relación entre el riesgo y el retorno de las carteras eficientes, donde cada punto representa una cartera óptima para un nivel de riesgo dado (Markowitz, 1952).

La hipótesis de maximizar el rendimiento descontado se rechaza porque no implica diversificación, independientemente de cómo se formen los rendimientos anticipados o cómo varíen las tasas de descuento. Esta hipótesis sugiere que el inversor debería concentrar sus fondos en el valor con el mayor valor descontado, lo cual no es una práctica diversificada ni óptima. Analíticamente, si existen múltiples valores con el mismo valor descontado, cualquier combinación de ellos sería igualmente válida. Sin embargo, como se observa en el gráfico, la cartera óptima para un nivel de riesgo determinado no se logra concentrando la inversión en un solo activo, sino mediante una combinación eficiente de activos que minimizan la varianza para cada nivel de rendimiento esperado (Markowitz, 1952).

Una regla que sugiere tanto la diversificación como la maximización del rendimiento esperado indica que el inversor debe distribuir sus fondos entre aquellos valores que ofrezcan el máximo rendimiento esperado. La ley de los grandes números asegura que el rendimiento real de la cartera será casi el mismo que el rendimiento esperado. Sin embargo, esta regla asume incorrectamente que existe una cartera que maximiza el rendimiento esperado y minimiza la varianza, ignorando que la diversificación no puede eliminar toda la varianza debido a la correlación entre los valores (Markowitz, 1952).

El portafolio con el máximo rendimiento esperado no necesariamente es el que tiene la mínima varianza. Existe una relación en la cual el inversor puede aumentar el rendimiento esperado al asumir más varianza, o reducir la varianza al renunciar al rendimiento esperado. La regla de "rendimientos esperados—varianza de rendimientos" (E-V) es adecuada para considerar estas compensaciones, aunque implica simplificaciones y limitaciones, tales como la suposición de probabilidades estáticas y la presentación geométrica para casos de tres o cuatro valores (Markowitz, 1952).

Para aplicar la regla E-V, se requiere comprensión de conceptos elementales de estadística matemática, como variables aleatorias y sus probabilidades asociadas. Este enfoque evita declaraciones matemáticas complicadas, pero a costa de precisión y generalidad. En futuras investigaciones, se propondrá un tratamiento matemático general para abordar estas limitaciones (Markowitz, 1952).

Podemos calcular la rentabilidad esperada y la varianza/riesgo de una cartera de activos de la siguiente manera:

$$R_p = \sum w_i * i \quad (1)$$

Donde:

- $R_p$  es la rentabilidad de la cartera.
- $w_i$  es el peso del activo i en la cartera.
- $R_i$  es la rentabilidad del activo i.

$$\sigma_p = \sqrt{(\sum \sum w_i * w_j * p_{ij} * \sigma_i * \sigma_j)} \quad (2)$$

Donde:

- $\sigma_p$  es la desviación estándar de la cartera.
- $w_i$  y  $w_j$  son los pesos de los activos i y j en la cartera.
- $p_{ij}$  es la correlación entre los rendimientos de los activos i y j.
- $\sigma_i$  y  $\sigma_j$  son las desviaciones estándar de los rendimientos de los activos i y j.

La utilidad es una medida de la satisfacción relativa de un inversor con una variedad de carteras. La aversión al riesgo se mide por la recompensa marginal que exige un inversor para aceptar un riesgo adicional. Los inversores que son más resistentes al riesgo exigen una compensación más alta por asumir más riesgos. Por lo tanto, A es mayor para las personas más adversas al riesgo. La fórmula es la siguiente:

$$U = E(R) - \frac{1}{2} A * \sigma^2 \quad (3)$$

Donde:

- U es la utilidad esperada de la cartera.
- E(R) es la rentabilidad esperada de la cartera
- A es la aversión al riesgo del inversor.
- $\sigma^2$  es la varianza de la cartera.

Basándonos en estas funciones, para determinar la composición de las carteras eficientes, es decir, la ponderación correspondiente a cada activo, se debe llevar a cabo el modelo de maximización del rendimiento o el de minimización del riesgo. El conjunto de activos ponderados que resultan de estas funciones objetivo, determinando un cierto nivel de riesgo y rentabilidad, forma la frontera de carteras eficientes.

El conjunto de rentabilidades esperadas y desviaciones estándar de todas las carteras eficientes por encima de la cartera global de mínima varianza se representa como la frontera eficiente, mostrada en el Gráfico 1. La frontera de mínima varianza es el conjunto de todas las carteras que representan el nivel de riesgo más bajo que se puede alcanzar para cada nivel

de rentabilidad posible. La cartera de varianza mínima global es la cartera con la varianza más baja y el nivel de riesgo más bajo que se puede alcanzar. En el gráfico, esta cartera se destaca como el punto de inicio de la frontera eficiente, marcado con un punto rojo.

Es importante tener en cuenta que el ejemplo de dos activos es insignificante (sin relevancia). Sin embargo, cuando tenemos tres activos, hay un número infinito de ponderaciones posibles que se pueden usar para lograr cualquier nivel específico de rentabilidad, pero solo un conjunto de ponderaciones que da el menor nivel posible de varianza para ese nivel de rentabilidad. Los inversores prefieren mantener los tres activos en ese nivel de ponderaciones porque tienen aversión al riesgo a cualquier nivel fijo de rentabilidad. Se han encontrado las ponderaciones para todas las carteras en la frontera de la varianza mínima cuando se determina el conjunto de ponderaciones de varianza mínima para todos los niveles posibles de rentabilidad.

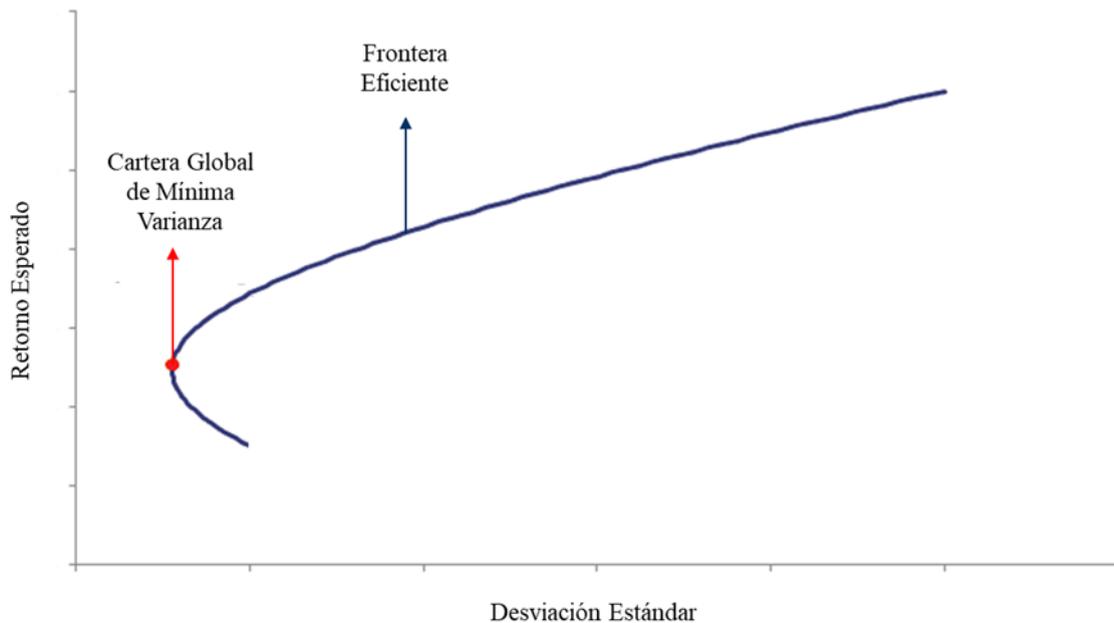
Si el riesgo aumenta (hacia el este), debe compensarse con una mayor rentabilidad (hacia el norte) para generar la misma utilidad. Las curvas de indiferencia son convexas debido a la utilidad marginal decreciente de la rentabilidad (o riqueza). A medida que aumenta el riesgo, el inversor necesita una mayor rentabilidad para compensar el mayor riesgo a un ritmo cada vez mayor (es decir, la curva se hace más pronunciada). La curva de indiferencia convexa ascendente tiene un coeficiente de pendiente estrechamente relacionado con el coeficiente de aversión al riesgo. Cuanto mayor es la pendiente, mayor es la aversión al riesgo del inversor, ya que se requiere un mayor incremento de la rentabilidad para aceptar un aumento determinado del riesgo.

La cartera óptima varía para cada inversor en función de su curva de utilidad. La cartera óptima se encuentra en el punto de tangencia entre la frontera eficiente y la curva de mayor utilidad posible del inversor. Sin embargo, algunos inversores con aversión al riesgo pueden tener una curva de utilidad que exija un menor equilibrio entre rentabilidad esperada y menor riesgo esperado en comparación con otros.

La cartera óptima se encuentra en el punto de tangencia entre la frontera eficiente y la curva de mayor utilidad posible y la cartera óptima puede ser diferente para cada inversor, pero para un inversor dado tiene la mayor utilidad. Las carteras que estén por encima de la frontera eficiente se consideran inalcanzables, ya que no se puede conseguir mayor rentabilidad para

un nivel determinado de riesgo al de la frontera, y aquellas carteras por debajo se consideran alcanzables pero ineficientes, ya que presentan una rentabilidad menor para el mismo riesgo.

**Gráfico 1: Curva de la Frontera Eficiente y la Cartera de Mínima Varianza**



*Fuente: Elaboración Propia*

## 2.2 Black-Litterman

En 1990, Fischer Black y Robert Litterman publicaron el modelo de asignación de activos Black-Litterman, que ha sido ampliamente adoptado y adaptado en la investigación financiera desde entonces. Este modelo aborda dos aspectos cruciales en la asignación de activos: proporciona una referencia clara, la cartera de equilibrio del mercado, y utiliza la "optimización inversa" para generar una distribución estable de rendimientos, mejorando significativamente el proceso de estimación de ganancias (Walters, 2011).

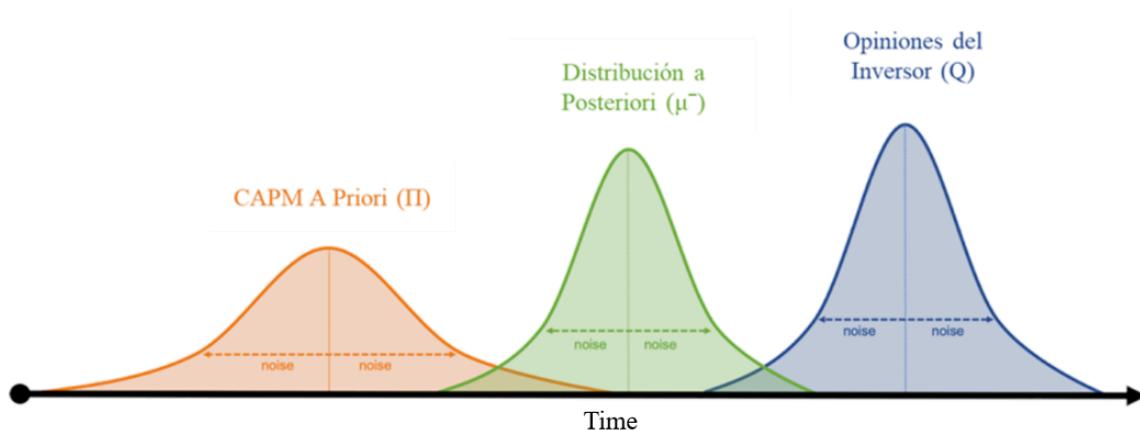
El modelo Black-Litterman se basa en un enfoque bayesiano para inferir los rendimientos esperados de los activos. Utiliza la distribución de equilibrio del modelo de precios de activos de capital (CAPM) como la distribución a priori y las opiniones del inversor como información adicional para inferir la distribución a posteriori de los rendimientos esperados. Esta estructura se ilustra en el Gráfico 2 a continuación, donde se observa cómo el modelo

combina la distribución a priori del CAPM ( $\Pi$ ), representada en la curva naranja, y las opiniones del inversor ( $Q$ ), representadas en la curva azul, para obtener la distribución a posteriori de los rendimientos esperados ( $\mu^{\bar{}}$ ), representada en la curva verde. Este gráfico destaca cómo las opiniones del inversor afectan las estimaciones iniciales, ajustando los rendimientos esperados con base en nueva información y reduciendo el "ruido" presente en las estimaciones iniciales y subjetivas (Black & Litterman, 1991).

Supongamos un mercado con  $N$  activos, incluyendo acciones, bonos y otros instrumentos financieros. Los rendimientos de estos activos siguen una distribución normal con media  $\mu$  y matriz de covarianzas  $\Sigma$ . En equilibrio, los inversores mantienen la cartera de mercado  $w_{eq}$ , y las primas de riesgo de equilibrio,  $\Pi$ , se define como  $\Pi = \delta \Sigma w_{eq}$ , donde  $\delta$  es el parámetro de aversión al riesgo promedio. La distribución a priori asume que los rendimientos esperados,  $\mu$ , se distribuyen normalmente alrededor de los valores de equilibrio  $\Pi$  (He & Litterman, 2002).

Además del a priori del CAPM, los inversores pueden tener opiniones específicas sobre los rendimientos del mercado, expresadas como una distribución normal de la rentabilidad esperada de una cartera  $p$  con media  $q$  y desviación estándar  $\omega$ . Estas opiniones se representan mediante matrices  $P$  y  $Q$ , donde  $P$  es una matriz de  $K \times N$  que contiene los pesos de las carteras y  $Q$  es un vector de  $K$  elementos con los rendimientos esperados de estas carteras (Black & Litterman, 1991). El Gráfico 2 resalta cómo estas opiniones se incorporan para ajustar las estimaciones iniciales del CAPM.

**Gráfico 2: Combinación de la Distribución CAPM con las Opiniones del Inversor**



*Fuente: Elaboración Propia*

Las opiniones del inversor se expresan como  $P\mu=Q+\epsilon(v)$ , donde  $\epsilon(v)$  es un vector aleatorio normalmente distribuido con media cero y matriz de covarianzas  $\Omega$ . Se asume que  $\epsilon(e)$  y  $\epsilon(v)$  son independientes (Black & Litterman, 1991).

Utilizando el enfoque bayesiano, se combinan las opiniones del inversor con la distribución a priori del CAPM. El resultado es que los rendimientos esperados se distribuyen normalmente con media  $\mu^-$  y matriz de covarianzas  $M^{-1}$ , la cual se calcula de la siguiente manera:

$$\mu^- = [(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1} [(\tau\Sigma)^{-1}\Pi + P'\Omega^{-1}Q] \quad (4)$$

El modelo permite calcular las ponderaciones de la cartera óptima utilizando la media  $\mu^-$  y la matriz de covarianzas  $\Sigma^-$  en el marco de optimización media-varianza. Para un inversor con aversión al riesgo  $\delta$ , el problema de maximización se expresa como:

$$\text{Max} \left\{ w' \mu^- - \frac{\delta}{2} w' \Sigma^- w \right\} \quad (5)$$

La solución a este problema proporciona las ponderaciones óptimas de la cartera  $w^*$ , que se pueden escribir como:

$$w^* = \frac{1}{1+\tau} (w_{eq} + P'\Lambda) \quad (6)$$

donde  $w_{eq}$  es la cartera de equilibrio del mercado y  $\Lambda$  es un vector que ajusta las expectativas de acuerdo con las opiniones del inversor (Black & Litterman, 1991).

La combinación de las opiniones subjetivas del inversor y las estimaciones objetivas de mercado, como se ilustra en el Gráfico 2, permite un enfoque más flexible y ajustado a las necesidades y expectativas individuales del inversor, haciendo de este modelo una herramienta valiosa en la optimización de carteras.

### 2.3 Montecarlo

Para prever las ganancias y pérdidas y las coberturas necesarias ante los niveles de probabilidad de pérdida esperada, los inversores deben realizar estudios de simulación del VaR (Valor en Riesgo) del portafolio de inversiones a diferentes niveles de confianza estadística. Esto se debe a los niveles de volatilidad que ocurren en diversos mercados financieros de renta variable como resultado de choques económicos, financieros, sociales y ambientales. Es crucial para los actores involucrados medir la pérdida prevista de las carteras de inversión a niveles de confianza estadística para analizar diferentes escenarios riesgosos y gestionar adecuadamente sus posiciones en el mercado de valores de activos financieros de renta variable (Chambi Condori, 2019).

El método de simulación de Montecarlo permite estimar el VaR esperado ante la variación simulada de las variables aleatorias que intervienen en el modelo, que en este caso son la rentabilidad media diaria, la volatilidad y el nivel de correlación entre los activos en la cartera. El proceso de simulación de Montecarlo comienza con la obtención del modelo y continúa identificando las variables aleatorias (Chambi Condori, 2019).

La pérdida máxima esperada (o la peor pérdida) sobre un horizonte de tiempo objetivo dentro de un intervalo de confianza estadística se cuantifica mediante el valor de retorno de la pérdida (VaR). La fijación del tiempo y el nivel de confianza estadística son necesarios para medir el VaR (Jorion, 2010). El enfoque paramétrico, que se basa en el análisis de las varianzas y covarianzas de las variables, es una de las formas de estimar el VaR. La pérdida esperada de una cartera frente a niveles específicos de volatilidad de las variables de entrada del modelo se puede medir utilizando el VaR. La simulación de Montecarlo se utiliza para estimar el VaR, lo que permite establecer supuestos más adaptables. El VaR por Montecarlo, en su forma más básica, presupone que los beneficios se distribuyen de manera uniforme y que la matriz de variaciones y covarianzas captura todas las posibilidades de dependencia entre los activos de la cartera (Chambi Condori, 2019).

Las instituciones financieras utilizan ampliamente el VaR para medir el riesgo de mercado y el riesgo operativo a niveles de confianza del 95% y 99%, y para un horizonte temporal entre uno y diez días. El término "método de Montecarlo" se refiere a cualquier método de medición de riesgo que use la simulación de un modelo paramétrico explícito para cambios

en los factores de riesgo. De esta manera, el método puede ser condicional o incondicional dependiendo de si el modelo utilizó una distribución estática o un modelo de serie temporal dinámico para cambiar los factores de riesgo. La obtención del modelo paramétrico es el primer paso en la simulación. Luego, se deben identificar las variables de riesgo involucradas en el modelo y determinar el tipo de distribución aleatoria que corresponde a cada una de las variables de riesgo para que la simulación pueda llevarse a cabo (McNeil et al., 2005).

La variabilidad en las variables de resultado es causada por la aleatoriedad en las variables de entrada. La simulación por el método de Montecarlo es un modelo que calcula los resultados mediante muestreos aleatorios repetidos y análisis estadísticos. En cuanto a su propósito, es muy similar a la técnica de diseño de experimentos, ya que el resultado específico no se sabe de antemano, pero se puede predecir mediante una simulación.

El VaR se puede expresar mediante la fórmula:

$$\text{VaRp} = Z \times I \times \sigma \times \Delta t \quad (7)$$

Donde  $Z$  es el valor en un nivel específico de confianza,  $I$  es la inversión en un portafolio,  $\sigma$  es el riesgo de la cartera y  $\Delta t$  es la variación del tiempo (Chambi Condori, 2019).

El objetivo principal del estudio es analizar cómo la rentabilidad diaria, la variación de cada uno de los componentes de una cartera de inversiones y la correlación entre los activos de la cartera afectan el valor promedio de las inversiones. La hipótesis principal es que la variación de la rentabilidad y la volatilidad en el VaR afectan el valor promedio de las inversiones. El primer objetivo específico es estimar la pérdida potencial esperada en una cartera de inversiones de renta variable mediante la simulación de Montecarlo sobre el modelo VaR. El segundo objetivo específico es aplicar la simulación de Montecarlo en el modelo de estructuración de carteras para obtener el valor promedio de una cartera de inversiones (Chambi Condori, 2019).

Se plantea como hipótesis de investigación que los cambios en la rentabilidad diaria, su volatilidad en el tiempo y la relación entre los componentes de la cartera afectan el VaR del portafolio de inversiones (Chambi Condori, 2019).

La rentabilidad media diaria y la volatilidad de cada uno de los activos de una muestra se obtienen como resultados descriptivos. La teoría de conformación de carteras de inversión con activos de renta variable de Markowitz (1959) sugiere que la selección de activos se basa en elegir aquellos que generen la mayor rentabilidad y que tengan correlaciones más bajas entre sí (Chambi Condori, 2019).

### **3. TEORÍAS METAHEURÍSTICAS DE GESTIÓN DE CARTERAS**

Después de explorar las teorías tradicionales de gestión de carteras, como los modelos de Markowitz (1952), Black-Litterman (1990) y los métodos basados en simulación de Monte Carlo, es crucial avanzar hacia enfoques modernos y adaptativos. Estas teorías han sentado las bases para la asignación de activos y la gestión del riesgo, pero en un entorno financiero cada vez más dinámico y complejo, se requieren técnicas más flexibles y robustas. Desde mediados de la década de 1990, los métodos metaheurísticos han cobrado relevancia como soluciones innovadoras, pues permiten una exploración exhaustiva del espacio de soluciones, optimizando las decisiones de inversión mediante técnicas avanzadas de adaptación y aprendizaje.

Los métodos metaheurísticos empezaron a desarrollarse en los años 60 y 70 en el campo de la investigación operativa, pero fue en los años 90 cuando comenzaron a aplicarse a la optimización en finanzas. Este cambio fue impulsado por avances computacionales que permitieron implementar técnicas como algoritmos genéticos, introducidos en 1975 por John Holland, y recocido simulado, formalizado por Kirkpatrick en 1983. Estas técnicas se aplicaron inicialmente en problemas de optimización combinatoria y, hacia finales de los años 90, empezaron a aplicarse en la optimización de carteras debido a su capacidad para manejar estructuras complejas de riesgo y retorno.

Los métodos metaheurísticos ofrecen una ventaja clave al poder adaptarse a distintos entornos y condiciones de mercado, explorando y explotando simultáneamente el espacio de soluciones. A diferencia de la teoría de Markowitz, que se centra en la diversificación y en la compensación riesgo-rendimiento, o del modelo Black-Litterman, que incorpora opiniones de mercado en la asignación de activos, las metaheurísticas permiten a los gestores de cartera abordar problemas con gran número de variables y restricciones no lineales, una situación común en la realidad actual.

En años recientes, la investigación y uso de las metaheurísticas en la gestión de carteras han cobrado más relevancia. Un ejemplo es la crisis financiera de 2008, que puso de manifiesto las limitaciones de los modelos tradicionales y dio un impulso a enfoques alternativos y flexibles como las metaheurísticas para adaptarse a cambios abruptos de mercado. Hoy en día, se pueden encontrar aplicaciones de metaheurísticas en sistemas de inteligencia artificial

financiera y en software de optimización avanzada, los cuales permiten a los gestores de fondos mejorar su toma de decisiones en mercados altamente volátiles.

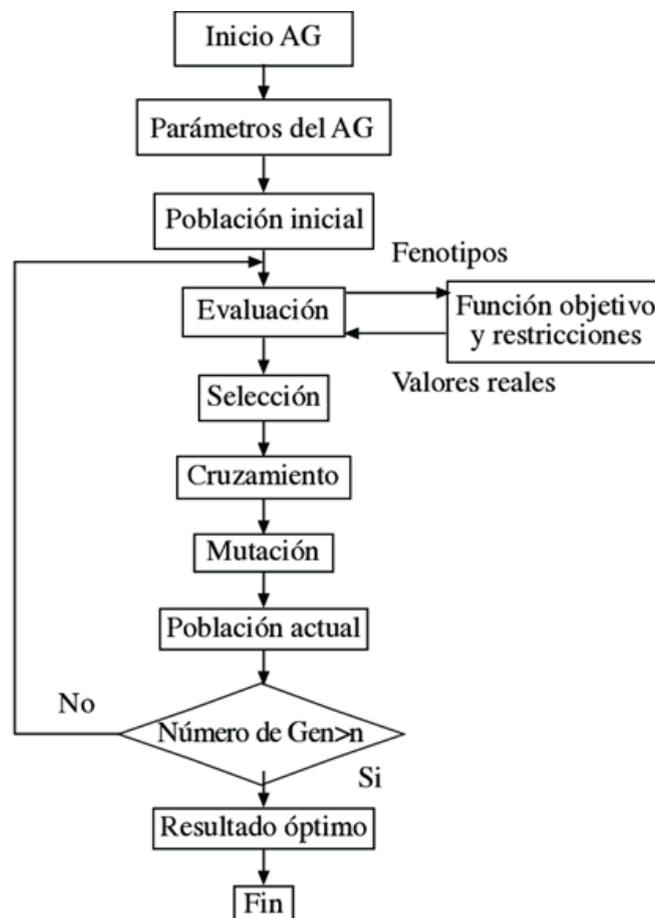
### **3.1 Algoritmos Genéticos (AGs)**

Los algoritmos genéticos (AGs) son técnicas de búsqueda global inspiradas en la teoría de la evolución de Darwin, ampliamente utilizadas para la estimación de parámetros multivariantes en ingeniería y ciencia. Estos algoritmos se han aplicado exitosamente en campos como la distribución eléctrica, las redes biológicas y los procesos de programación, y han demostrado ser capaces de derivar leyes de la naturaleza basándose únicamente en datos observados (Bies et al., 2006). No existe una definición estricta de "algoritmo genético" en la comunidad de computación evolutiva. Sin embargo, la mayoría de los AGs comparten ciertos elementos clave: poblaciones de cromosomas, selección basada en la aptitud, cruce para producir nueva descendencia y mutación aleatoria. Tradicionalmente, los cromosomas se representan como cadenas binarias, con cada locus teniendo dos posibles alelos: 0 y 1 (Mitchell, 1998).

El proceso de un AG, que se detalla en el Gráfico 3, comienza con la generación aleatoria de una población de soluciones candidatas, representadas como cromosomas binarios. En este gráfico se puede observar el flujo completo del algoritmo, desde la inicialización de parámetros y la población inicial hasta la evaluación, selección, cruce, mutación y la formación de nuevas generaciones, todo ello orientado hacia la obtención del resultado óptimo. La evaluación de cada cromosoma utiliza una función de aptitud que determina qué tan bien resuelve el cromosoma el problema en cuestión. Los cromosomas con mayor aptitud tienen más probabilidades de ser seleccionados para la reproducción.

El cruce, un operador fundamental, selecciona dos cromosomas padres y mezcla segmentos de sus cadenas para producir descendientes, como se ilustra en el bloque "Cruzamiento" (ver Gráfico 3: Proceso de Optimización mediante Algoritmos Genéticos) . Por ejemplo, se intercambian los primeros NNN bits de las cadenas de los padres para formar nuevas combinaciones. Este proceso imita la recombinación genética en la naturaleza y ayuda a explorar nuevas áreas del espacio de soluciones. La selección de los padres no es completamente aleatoria; está ponderada por la aptitud de los cromosomas, lo que significa que las soluciones más prometedoras tienen más oportunidades de contribuir a la siguiente generación (Unhelkar & Gonsalves, 2021).

**Gráfico 3: Proceso de Optimización mediante Algoritmos Genéticos (AG)**



*Fuente: Introducción a los Sistemas Inteligentes - Scientific Figure on Research Gate, 2024*

La mutación introduce variabilidad adicional al cambiar aleatoriamente bits individuales en un cromosoma, representada en el bloque "Mutación" del Gráfico 3. Aunque es un evento de baja frecuencia, la mutación es crucial para mantener la diversidad genética dentro de la población y para evitar la convergencia prematura a soluciones subóptimas. Técnicas avanzadas como el nicho y el elitismo aseguran que los mejores modelos candidatos se mantengan intactos entre generaciones, mientras que se exploran nuevas soluciones potenciales.

En la práctica, los AGs se utilizan para optimizar funciones complejas. Por ejemplo, maximizar la función  $f(y)=y+|\sin(32y)$  para  $0 \leq y < \pi$  implica traducir valores de  $y$  a cadenas binarias y evaluar su aptitud. En aplicaciones no numéricas, como la

búsqueda de secuencias de aminoácidos que se pliegan en una estructura proteica tridimensional, la aptitud puede definirse como la energía potencial de la secuencia respecto a la estructura deseada (Mitchell, 1998). Los AGs híbridos combinan la capacidad de los AGs para explorar globalmente el espacio de soluciones con métodos de búsqueda local que refinan las soluciones en regiones específicas. Por ejemplo, en la construcción de modelos farmacocinéticos, un AG simple puede interrumpirse periódicamente para realizar una búsqueda local descendente, mejorando la precisión de los modelos candidatos (Bies et al., 2006). Este enfoque es también evidente en la iteración del flujo mostrado en el Gráfico 3, donde se evalúan y refinan continuamente las poblaciones hasta alcanzar el resultado óptimo.

A diferencia de los métodos de búsqueda tradicionales, que se centran en la búsqueda de datos almacenados o caminos hacia objetivos específicos, los AGs utilizan una combinación de búsqueda basada en población paralela, selección estocástica, cruce y mutación. Este enfoque permite encontrar soluciones óptimas en espacios de solución complejos y diversos, diferenciándose significativamente de los métodos tradicionales de búsqueda (Mitchell, 1998).

### **3.2 Recocido Simulado (RS)**

El recocido simulado (RS) es un algoritmo de búsqueda metaheurística ampliamente utilizado para problemas de optimización global, y su aplicación en la gestión de carteras ha demostrado ser particularmente efectiva. El objetivo principal de este tipo de algoritmos es encontrar una buena aproximación al valor óptimo de una función en un amplio espacio de búsqueda, conocido como "óptimo global". Inspirado en el proceso de recocido del acero y las cerámicas, este método consiste en calentar y luego enfriar lentamente el material para variar sus propiedades físicas, permitiendo que los átomos del material se desplacen y se reorganicen en configuraciones de menor energía, similares al mínimo global. Este algoritmo fue descrito por primera vez por Scott Kirkpatrick, C. Daniel Gelatt y Mario P. Vecchi en 1983, y por Vlado Černý en 1985. Es una adaptación del algoritmo Metropolis-Hastings, utilizado para generar muestras de estados de un sistema termodinámico (Reyes Rodríguez & Moya Rodríguez, 2012).

En la gestión de carteras, el recocido simulado se utiliza para encontrar la mejor combinación de activos financieros que maximice el rendimiento esperado de la cartera mientras se

minimiza el riesgo. Durante cada iteración del reconocido simulado, el algoritmo evalúa varios vecinos del estado actual de la cartera y decide probabilísticamente si realizar una transición a un nuevo estado (nueva combinación de activos) o quedarse en el estado actual. Esta comparación entre estados vecinos se repite hasta encontrar una configuración óptima que maximice el rendimiento ajustado al riesgo de la cartera, o hasta que se cumpla un tiempo computacional predeterminado u otras condiciones específicas del problema (Reyes Rodríguez & Moya Rodríguez, 2012).

El vecindario de un estado está compuesto por todas las posibles configuraciones de la cartera que se pueden obtener mediante un cambio en la selección y ponderación de los activos. Los estados vecinos se generan mediante métodos de Montecarlo, y su evaluación es fundamental para encontrar una solución óptima global. La probabilidad de hacer la transición a una nueva configuración de la cartera depende de la diferencia de rendimiento ajustado al riesgo entre los dos estados y de una variable llamada temperatura. Si la transición mejora el rendimiento ajustado al riesgo, se acepta con probabilidad 1. Sin embargo, si disminuye el rendimiento ajustado al riesgo, también puede ser aceptada con una probabilidad distinta de cero, lo que evita que el sistema se quede atrapado en un óptimo local. A medida que la temperatura disminuye, la probabilidad de aceptar cambios desfavorables también disminuye, enfocando el algoritmo en la búsqueda de configuraciones de cartera cada vez más óptimas (Reyes Rodríguez & Moya Rodríguez, 2012).

El recocido simulado ha sido ampliamente estudiado y aplicado en la optimización de carteras, donde ha demostrado un rendimiento notable. Su capacidad para explorar eficientemente el espacio de búsqueda y evitar quedar atrapado en óptimos locales lo convierte en una herramienta valiosa. Diversas investigaciones han explorado estrategias para mejorar la eficiencia computacional del algoritmo en la gestión de carteras, tales como métodos alternativos para calcular el cambio en la función objetivo, la identificación de vecinos prometedores y el uso de características del óptimo para localizarlo más rápidamente. Estas mejoras son significativas para la investigación actual en la optimización computacional, aunque su aplicabilidad universal aún necesita ser evaluada (Ogbu & Smith, 1991).

El recocido simulado no es nuevo y ha sido utilizado con éxito en diversas disciplinas desde hace décadas. En la gestión de carteras, simula el comportamiento de sistemas físicos

sometidos a un descenso paulatino de temperatura, alcanzando un estado de mínima energía. Este proceso permite buscar el mínimo absoluto de cualquier función matemática, como la optimización de la composición de la cartera para maximizar el rendimiento y minimizar el riesgo. Esta técnica ha sido especialmente útil en problemas de optimización combinatoria, donde la búsqueda de soluciones óptimas es compleja debido a la gran cantidad de combinaciones posibles de activos (Espí, 1995).

El recocido simulado es un algoritmo robusto y versátil para la optimización global, particularmente efectivo en la gestión de carteras. Su capacidad para explorar eficientemente el espacio de búsqueda y evitar quedar atrapado en óptimos locales lo convierte en una herramienta valiosa para maximizar el rendimiento ajustado al riesgo de una cartera de inversión.

### **3.3 Búsqueda de Vecindario Variable**

El concepto de búsqueda de vecindario se basa en explorar el entorno de una solución inicial y seleccionar una nueva solución mejor en ese entorno. La solución inicial y sus vecinas se definen y se generan mediante movimientos específicos. La búsqueda local es una técnica que explora sistemáticamente los entornos de las soluciones para mejorar iterativamente la calidad de la solución actual hasta alcanzar un óptimo local.

Cada solución tiene un conjunto de soluciones vecinas que se pueden obtener mediante movimientos. Los movimientos más comunes incluyen agregar un elemento, intercambiar elementos, insertar o eliminar elementos. Por ejemplo, si se considera un problema de mochila, se pueden agregar elementos a la solución actual siempre y cuando se respeten las restricciones del problema, como el peso máximo permitido. La vecindad de una solución incluye todas las soluciones que se pueden alcanzar mediante estos movimientos (Hansen et al., s. f.).

En la búsqueda de entornos variables (VNS), se utiliza un conjunto de vecindades predefinidas para realizar la búsqueda local. La idea es cambiar gradualmente el vecindario de búsqueda para explorar diferentes regiones del espacio de soluciones. Esta técnica permite superar la limitación de quedar atrapado en mínimos locales, ya que la estructura de vecindad se amplía progresivamente para incluir más posibles soluciones. El objetivo es encontrar el

mínimo global explorando múltiples vecindarios y aprovechando las características compartidas entre los mejores locales y el óptimo global (Ramalhinho-Lourenço et al., 2000).

La búsqueda local iterativa introduce perturbaciones a la solución actual para explorar nuevas vecindades y evitar quedar atrapado en pozos de atracción. Las perturbaciones pueden ser pequeñas para asegurar una búsqueda controlada y evitar ciclos repetitivos. Este enfoque mejora la eficiencia del algoritmo al permitir una diversificación efectiva cuando la función objetivo no mejora y una intensificación cuando se encuentra una mejor solución (Ramalhinho-Lourenço et al., 2000).

En el contexto de la gestión de carteras, la búsqueda de entornos variables y la búsqueda local iterativa pueden aplicarse para ajustar la composición de la cartera de inversión. Al generar y evaluar soluciones vecinas mediante movimientos como agregar o intercambiar activos, los gestores pueden encontrar configuraciones de inversión que optimicen el rendimiento esperado y minimicen el riesgo. Este enfoque iterativo permite una exploración detallada del espacio de soluciones, mejorando la probabilidad de encontrar la configuración óptima de la cartera (Hernández Quevedo, 2020).

En la búsqueda local iterativa, el algoritmo comienza con una solución inicial y busca mejoras en su vecindad inmediata. Si encuentra una mejor solución, se mueve a esa nueva solución y repite el proceso. Este método es efectivo para problemas donde el espacio de búsqueda es grande y complejo, permitiendo que el algoritmo explore diversas configuraciones antes de converger a un óptimo local (Galván Rodríguez, 2017).

Para evitar quedar atrapado en un óptimo local, la búsqueda de entornos variables introduce cambios controlados en la solución actual, permitiendo explorar diferentes áreas del espacio de soluciones. Esta técnica es especialmente útil en la gestión de carteras, donde las condiciones del mercado pueden cambiar rápidamente y es necesario ajustar las inversiones en consecuencia (Hernández Quevedo, 2020).

En resumen, la técnica de búsqueda de entornos variables y la búsqueda local iterativa son métodos efectivos para optimizar problemas complejos como la gestión de carteras, proporcionando una forma sistemática de explorar el espacio de soluciones y mejorar la

calidad de las decisiones de inversión mediante la exploración y explotación de vecindarios de soluciones.

### **3.4 Búsqueda Tabú (TB)**

La Búsqueda Tabú (BT) es un método metaheurístico diseñado para resolver problemas de optimización combinatoria, como la gestión de carteras de inversión. La BT utiliza un procedimiento de búsqueda local para moverse iterativamente desde una solución  $x$  hacia una solución  $x^*$  dentro de su vecindad, hasta satisfacer algún criterio de parada. A medida que la búsqueda progresa, la BT modifica la estructura de vecinos para cada solución, utilizando estructuras de memoria para determinar las soluciones permitidas en el nuevo vecindario  $N^*(x)$ . Este enfoque permite explorar regiones del espacio de búsqueda que serían ignoradas por métodos de búsqueda local tradicionales (Melián Batista et al., 2003).

La estructura de memoria más importante en la BT es la lista tabú. Esta lista es una memoria de corto plazo que contiene las soluciones visitadas recientemente, excluyéndolas del vecindario  $N^*(x)$  durante un número determinado de iteraciones. En el contexto de la gestión de carteras, esto significa que ciertas configuraciones de cartera no serán reevaluadas inmediatamente, permitiendo que se exploren otras configuraciones potencialmente mejores. Además, la lista tabú puede prohibir soluciones con ciertos atributos o prevenir movimientos específicos, como mantener o eliminar ciertos activos en la cartera (Melián Batista et al., 2003).

Para mantener y gestionar las restricciones tabú, la BT utiliza varias clases de funciones de memoria basadas en la recencia y la frecuencia. Las funciones de memoria basadas en la recencia se especifican mediante vectores que identifican las iteraciones de comienzo y finalización del período tabú para un atributo determinado. Por otro lado, la memoria basada en la frecuencia complementa la información de la recencia, proporcionando una base ampliada para seleccionar movimientos preferidos al tener en cuenta la calidad de las soluciones e influencia de los movimientos (Melián Batista et al., 2003).

En la gestión de carteras, un problema es que algunas configuraciones de alta calidad pueden ser excluidas debido a las restricciones tabú. Para mitigar este problema, se utilizan criterios de aspiración que permiten sobrescribir el estado tabú de una solución si se cumple una

condición específica, como cuando una configuración de cartera ofrece un mejor rendimiento ajustado por riesgo que la mejor conocida hasta ese momento. Un criterio de aspiración muy utilizado en la literatura es admitir soluciones que son mejores que la mejor solución conocida al momento, y este será el utilizado en la implementación que aquí se considera (Barros-Duarte, 2006).

La BT también incorpora estrategias de intensificación y diversificación. La intensificación se basa en modificar las reglas para promover la elección de buenas combinaciones dentro de un conjunto de soluciones superiores. En términos de carteras, esto podría significar ajustar los pesos de los activos para mejorar el ratio de Sharpe. La diversificación, por otro lado, implica explorar áreas del espacio de soluciones que no se han visitado previamente, permitiendo obtener configuraciones de carteras diferentes y potencialmente mejores (Glover & Laguna, 1997).

La búsqueda de entorno, un componente esencial de la BT, se refiere a encontrar una solución que sea igual de buena o mejor que las soluciones en su entorno inmediato. En la gestión de carteras, esto implica ajustar los pesos de los activos de manera que se maximice el rendimiento esperado para un nivel dado de riesgo o se minimice el riesgo para un nivel dado de rendimiento. En la BT, la búsqueda de entorno incluye procedimientos constructivos y destructivos basados en la memoria adaptativa, permitiendo explorar el espacio de soluciones repetidamente hasta encontrar una configuración óptima (Melián Batista et al., 2003).

El uso de memorias a corto y largo plazo permite que la BT realice cambios en el entorno de la solución actual. Las estrategias basadas en memoria a corto plazo, como la lista tabú, registran los movimientos realizados en cada iteración y prohíben su repetición inmediata. En contraste, las estrategias de largo plazo registran la frecuencia de ciertos movimientos y los almacenan como historial, lo que es crucial en la gestión de carteras para evitar configuraciones que históricamente no han sido efectivas (Melián Batista et al., 2003).

El criterio de aspiración mejora la ejecución de la BT al permitir excepciones a las restricciones tabú si un movimiento se considera beneficioso. Esto se evalúa midiendo la influencia e impacto del movimiento propuesto en la solución, asegurando que las restricciones se ajusten para permitir configuraciones de cartera prometedoras (Melián Batista et al., 2003).

### 3.5 Colonia de Hormigas (ACO)

La Optimización por Colonia de Hormigas (ACO) se inspira en el comportamiento de las colonias de hormigas naturales que buscan comida y trazan el camino más corto desde la fuente de alimento hasta su nido. Este algoritmo se basa en un sistema de comunicación conocido como estigmergia. Inicialmente, todas las hormigas se mueven de manera aleatoria hasta que algunas encuentran comida. Estas hormigas regresan al nido depositando feromonas en el camino, lo que sirve como guía para las demás hormigas. Las hormigas que siguen estos rastros de feromonas también depositan más feromonas, reforzando así los caminos más utilizados y eficientes. Con el tiempo, debido a la evaporación de las feromonas, los caminos más largos pierden su atractivo y las hormigas tienden a seguir el camino más corto, que está más marcado por feromonas. Este comportamiento permite a las hormigas encontrar de manera eficiente el camino más corto entre el nido y la fuente de alimento (Melián Batista et al., 2003).

El algoritmo ACO se utiliza para resolver problemas de optimización combinatoria, donde cada hormiga artificial construye una solución recorriendo un grafo que representa los posibles pasos a seguir. En este grafo, cada arista tiene asociada dos tipos de información: la información heurística, que mide la preferencia heurística de moverse de un nodo a otro, y la información de feromonas, que mide la "deseabilidad aprendida" de este movimiento. La feromona se actualiza durante la ejecución del algoritmo en función de las soluciones encontradas, imitando el comportamiento de las hormigas naturales (Dorigo & Blum, 2005).

En el contexto de la gestión de carteras, la ACO puede ser una herramienta poderosa para la optimización de la asignación de activos. Las características adaptativas de ACO permiten que el algoritmo se ajuste continuamente a las condiciones cambiantes del mercado, utilizando datos históricos y nuevos para guiar la asignación de activos. La capacidad de ACO para evitar estancarse en óptimos locales y explorar una amplia variedad de soluciones potenciales es particularmente valiosa en los mercados financieros, donde las interdependencias entre los activos y la volatilidad del mercado pueden crear múltiples óptimos locales.

Además, las hormigas naturales, que colaboran y comunican mediante feromonas, permiten que las colonias resuelvan problemas complejos de manera eficiente. Este comportamiento

puede ser traducido a algoritmos computacionales que imitan esta cooperación y comunicación, resultando en soluciones eficientes para problemas complejos como la optimización de carteras. Por ejemplo, en cada iteración del algoritmo, cada "hormiga" construye una solución considerando tanto la información heurística como la información de feromonas, lo que facilita la exploración y explotación de diferentes combinaciones de activos (Algarín, 2010).

El algoritmo también incluye mecanismos para la evaporación de feromonas, lo que garantiza que los caminos menos prometedores pierdan progresivamente su atractivo y no se sigan explorando. Esta característica es crucial para evitar que el algoritmo se estanque en soluciones subóptimas y asegura que siempre se esté explorando nuevas y potencialmente mejores soluciones (Bonabeau et al., 1999).

### **3.6 Optimización por Enjambre de Partículas (PSO)**

La Optimización por Enjambre de Partículas (PSO) es una técnica inspirada en el comportamiento colectivo de bandadas de aves y bancos de peces. Esta técnica se basa en la combinación de la experiencia individual de cada partícula y la experiencia social del grupo. En el contexto de PSO, una partícula representa una solución potencial y el enjambre es la colección de todas las partículas. Cada partícula se mueve en el espacio de búsqueda influenciada tanto por su mejor posición individual conocida (p-best) como por la mejor posición conocida por cualquier partícula en el enjambre (g-best). Este movimiento está determinado por ecuaciones que consideran la velocidad actual de la partícula, una constante de inercia, y factores sociales y cognitivos que guían el movimiento hacia las mejores posiciones conocidas (Unhelkar & Gonsalves, 2021).

La PSO comienza con la inicialización de una población de partículas con posiciones y velocidades aleatorias. Durante cada iteración, las partículas evalúan su posición actual según una función de aptitud predefinida y actualizan sus velocidades y posiciones basándose en la mejor solución individual y la mejor solución global encontrada hasta el momento. Este proceso se repite hasta que se cumple un criterio de parada, como alcanzar un número máximo de iteraciones o una mejora mínima en las soluciones encontradas.

Una característica clave de PSO es su capacidad de balancear la exploración y la explotación. La exploración permite al enjambre buscar nuevas áreas del espacio de soluciones, mientras que la explotación se enfoca en refinar las mejores soluciones encontradas. El peso de inercia juega un papel crucial en este balance: un peso alto favorece la exploración, mientras que un peso bajo favorece la explotación local. Además, los parámetros sociales ( $c_1$  y  $c_2$ ) y los factores aleatorios ( $r_1$  y  $r_2$ ) influyen en la dirección y magnitud del movimiento de las partículas, permitiendo una búsqueda flexible y adaptativa (Röhler & Chen, 2011).

En la gestión de carteras, PSO puede ser particularmente útil debido a su capacidad para manejar espacios de búsqueda complejos y multimodales. La optimización de carteras implica encontrar la mejor combinación de activos que maximice el rendimiento ajustado por riesgo. PSO puede explorar eficientemente una gran variedad de combinaciones de activos, adaptándose a las dinámicas cambiantes del mercado financiero. Al considerar tanto la experiencia individual de cada partícula (rendimientos históricos de los activos) como la experiencia colectiva del enjambre (correlaciones y volatilidades), PSO puede identificar configuraciones de cartera que optimicen el rendimiento esperado frente al riesgo (Röhler & Chen, 2011).

Un ejemplo práctico del uso de PSO en la gestión de carteras podría ser la optimización de la asignación de activos en un portafolio diversificado. Inicialmente, se generarían varias carteras con asignaciones aleatorias. Cada "partícula" en este caso representaría una cartera con una distribución específica de activos. A través de iteraciones sucesivas, estas partículas ajustan sus posiciones en el espacio de soluciones, buscando maximizar el ratio de Sharpe o minimizar la volatilidad, mientras se adhieren a restricciones específicas del inversor, como límites de inversión en ciertos activos o sectores.

### **3.7 Optimización por Colonia de Lobos (WSA)**

La Optimización por Colonia de Lobos (WSA) se inspira en el comportamiento social de los lobos, quienes cazan en manadas utilizando el sigilo para acercarse a sus presas. Los lobos se mueven en grupo en una formación libre pero acoplada, aunque tienden a tomar presas individualmente. Este comportamiento se refleja en la estructura del WSA, donde cada lobo se mueve independientemente, recordando su propio rastro y fusionándose con sus compañeros cuando estos se encuentran en una mejor posición (Gómez, s. f.).

Durante la caza, los lobos tratan de ocultarse y cambian de posición continuamente a lugares con más refugio, menos obstáculos o menos vulnerabilidad. Además de buscar presas, se mantienen atentos a amenazas potenciales. Cuando una amenaza aparece, el lobo se desplaza aleatoriamente a una gran distancia de su posición actual, ayudando a romper el punto muerto de vulnerabilidad. Los lobos tienen un excelente sentido del olfato, que les permite ubicar a sus presas por el olor, y una distancia de detección que crea un área de cobertura denominada distancia visual. Esta distancia visual es crucial tanto para la búsqueda de alimento (el óptimo global) como para la conciencia de sus compañeros y la señalización de enemigos cercanos (Tang et al., 2012).

El WSA se diferencia de otras metaheurísticas biológicas porque combina la capacidad de búsqueda local individual con el movimiento autónomo de grupo. Cada lobo en el WSA caza independientemente, recordando su propio rastro, y solo se fusiona con sus compañeros cuando estos se encuentran en una mejor posición, eliminando así la necesidad de intercomunicación de largo alcance y ahorrando tiempo (Kaveh & Farhoudi, 2011).

El WSA depende del alcance visual y delega el comportamiento de enjambre a cada lobo individual en lugar de a un solo líder, permitiendo múltiples líderes que se mueven en diversas direcciones para hallar la mejor solución. Además, el WSA incluye una probabilidad de amenaza que simula encuentros con enemigos. Cuando un lobo detecta un cazador, salta fuera del alcance visual para evitar ser atrapado, mejorando la eficiencia del algoritmo (Gómez, s. f.).

En el contexto de la gestión de carteras, el WSA puede ofrecer ventajas significativas. La capacidad del algoritmo para combinar la búsqueda local individual con el movimiento autónomo del grupo permite una exploración y explotación eficientes del espacio de soluciones. Cada "lobo" en el WSA podría representar una estrategia de inversión individual que se ajusta dinámicamente a las condiciones del mercado, buscando continuamente mejores configuraciones de activos y evitando configuraciones subóptimas mediante movimientos aleatorios frente a amenazas, como cambios bruscos del mercado.

Además, la memoria integrada en el WSA permite que las estrategias de inversión recuerden y aprendan de configuraciones de carteras pasadas, mejorando así la toma de decisiones

futuras. Este enfoque puede ser particularmente útil en mercados volátiles, donde la habilidad para adaptarse rápidamente y recordar estrategias exitosas anteriores puede marcar la diferencia en el rendimiento de la cartera.

### 3.8 Algoritmo de Murciélagos Virtuales (AMV)

El Algoritmo de Murciélagos Virtuales (AMV) se basa en las características naturales de ecolocalización de algunas especies de murciélagos, las cuales utilizan para ubicar objetos y presas mediante el eco producido por sus zumbidos ultrasónicos. Este sistema es tan preciso que puede evitar obstáculos tan pequeños como un cabello humano. Esta técnica también puede usarse para crear un escenario tridimensional que incluye la diferencia de tiempo entre la percepción en sus oídos y la variación de intensidad de los ecos (Yang, 2010).

Desde su propuesta en 2010, el AMV ha sido objeto de diversas variantes que buscan mejorar su comportamiento, como el planteamiento de optimización. Para simplificar el algoritmo, se establecieron algunas reglas básicas en relación con el comportamiento de los murciélagos (Yang, 2010; Gandomi & Alavi, 2012; Yang, 2011):

1. **Ecolocalización:** Los murciélagos virtuales utilizan la ecolocalización para distinguir entre obstáculos y presas.
2. **Movimiento Aleatorio:** Los murciélagos virtuales vuelan aleatoriamente a una velocidad  $v_i$ , en una posición  $x_i$ , emitiendo una frecuencia entre  $f_{\min}$  y  $f_{\max}$ , variando la longitud de onda  $\lambda$  y la intensidad  $A_0$  cuando localizan a su presa. Pueden ajustar la longitud de onda (o la frecuencia) y la emisión de la relación  $r$ ,  $r \in [0,1]$ , en función de cuán cercanos estén a su objetivo (Yang, 2010; Vanegas et al., 2013).
3. **Intensidad:** La intensidad comienza con una velocidad  $A_0$  y se extiende hasta un mínimo de  $A_{\min}$  (Vanegas et al., 2013).

En resumen, Yang (2010) propuso una forma de expresar el pseudocódigo del algoritmo de la siguiente manera:

1. Definir la función objetivo  $f(x)$ , donde  $x = T(x_1, \dots, x_d)^T$
2. Inicializar la población virtual de murciélagos  $x_i (i=1, 2, \dots, n)$  y  $v_i$ .

3. Definir las frecuencias  $f_i$  en  $x_i$ .
4. Inicializar la relación  $r_i$  y su intensidad  $A_i$ .

Mientras ( $t < \text{Máx. Número de iteraciones}$ ):

5. Generar nuevas soluciones ajustando las frecuencias y actualizando la velocidad y posición de cada murciélago.
6. Si ( $\text{aleatorio} > r_i$ ): Seleccionar una solución dentro de las mejores. Generar una solución local sobre la mejor solución seleccionada, empleando un recorrido aleatorio.
7. Si ( $\text{aleatorio} < A$  y  $f(x_i) < f(x)$ ): Aceptar las nuevas soluciones, aumentar  $r_i$  y reducir  $A_i$ .
8. Clasificar los murciélagos y encontrar el mejor  $x$ ,  $x_{real}$ .
9. Finalizar mientras. (Vanegas et al., 2013).

Durante las simulaciones, la inicialización de las frecuencias, así como la posición y velocidad de cada murciélago, se definen utilizando las siguientes expresiones:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (8)$$

$$v_i^t = v_i^{t-1} + (x_i - x^*)f_i \quad (9)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (10)$$

Donde:

- $\beta$  es un valor aleatorio entre 0 y 1,
- $f_i$  es la frecuencia para el murciélago virtual  $i$ -ésimo,
- $f_{min}$  y  $f_{max}$  son las frecuencias mínima y máxima recomendadas entre 0 y 100 (Yang, 2010),
- $v_i^t$  y  $v_i^{t-1}$  son las velocidades en los pasos de tiempo  $t$  y  $t-1$  respectivamente,
- $x_i^t$  y  $x_i^{t-1}$  son las posiciones en los pasos de tiempo  $t$  y  $t-1$  respectivamente,

- $x^*$  es la mejor posición en el actual paso de tiempo (Vanegas et al., 2013).
- La búsqueda local de este algoritmo se ejecuta empleando un recorrido aleatorio proporcionado por la siguiente expresión:

$$x_{local} = x^* + \epsilon A_t \quad (11)$$

Donde:

- $\epsilon$  es un número aleatorio entre -1 y 1,
- $A_t$  es el promedio de A en el actual paso de tiempo.

En el contexto de la gestión de carteras, los algoritmos de optimización como el AMV pueden ser herramientas valiosas para mejorar la asignación de activos. La capacidad de los murciélagos virtuales para explorar eficientemente el espacio de soluciones y ajustarse dinámicamente a nuevas condiciones puede ser aplicada para encontrar configuraciones óptimas de cartera que maximicen el rendimiento ajustado al riesgo. Además, la inclusión de una búsqueda local y global simultánea en el AMV asegura que el algoritmo no se estanque en óptimos locales, permitiendo una adaptación continua a las condiciones dinámicas del mercado financiero.

### 3.9 Optimización por Búsqueda de Gravedad

Einstein desarrolló su teoría general de la relatividad a principios del siglo XX, alcanzando su forma final en 1916. La base de esta teoría es la equivalencia entre sistemas inerciales y sistemas en caída libre, según la cual la aceleración de un laboratorio en caída libre cancela completamente el efecto gravitacional. Esto implica que la geometría y la gravedad comparten muchas propiedades, como se ilustra con un campo gravitacional representado en un espacio-tiempo curvado. La ecuación de campo gravitacional de Einstein describe esta relación entre la geometría del espacio-tiempo y la masa-energía y el momento de la fuente (Hsiao et al., 2005).

La ley de la gravedad de Newton establece que la fuerza gravitacional entre dos masas es proporcional al producto de sus masas e inversamente proporcional al cuadrado de la distancia entre ellas. Esta fuerza es constante en todo el universo. La variación de esta fuerza se puede utilizar para determinar la dirección de la aceleración gravitacional de un asteroide,

asumiendo que la fuerza es inversamente proporcional al cuadrado de la distancia entre las masas (Hsiao et al., 2005).

La heurística de optimización gravitatoria, o Space Gravitational Optimization (SGO), es un algoritmo inspirado en la simulación de asteroides que se desplazan en el universo en busca de la masa más pesada, que representa la solución óptima. En este algoritmo, los agentes de búsqueda son tratados como asteroides que viajan a través del espacio según la ley de la gravedad de Newton, y el espacio de soluciones se formula como un espacio-tiempo curvado de acuerdo con la teoría de la relatividad general de Einstein (Hsiao et al., 2005).

El algoritmo SGO se desarrolla en varios pasos esenciales. En primer lugar, los asteroides se inician en el espacio de búsqueda con una posición y velocidad aleatorias dentro de su región factible para evitar que escapen a posiciones no factibles. Inicialmente, los asteroides no tienen aceleración y se ven afectados únicamente por la atracción gravitatoria de la función objetivo, almacenando la mejor solución encontrada por cada asteroide (Zapatero Moreno et al., 2010).

La variación en el espacio-tiempo afecta la aceleración de los asteroides. Para determinar esta variación, se utiliza un parámetro conocido como rango de detección (rd), y se calcula la aceleración de cada asteroide utilizando la fórmula de variación en la geometría del espacio-tiempo, que considera la fuerza del campo gravitatorio y las variaciones parciales de la función objetivo (Zapatero Moreno et al., 2010).

Posteriormente, la posición y la velocidad de los asteroides se ajustan después de cada iteración utilizando sus nuevas velocidades y posiciones. Si un asteroide se aleja del espacio factible, se devuelve aleatoriamente a la zona apropiada para mantener la plausibilidad de la solución (Zapatero Moreno et al., 2010).

Finalmente, la mejor posición y el mejor valor de la función objetivo para cada asteroide se almacenan y actualizan tras cada iteración si la solución obtenida es mejor que la mejor solución conocida hasta ese momento (Hsiao et al., 2005).

En el contexto de la gestión de carteras, el SGO puede ser aplicado para optimizar la asignación de activos, tratando cada activo como un asteroide que busca la mejor

combinación de inversiones. Este algoritmo puede ajustar dinámicamente las posiciones y velocidades de los activos en función de la atracción gravitatoria de los rendimientos esperados, buscando maximizar la rentabilidad ajustada al riesgo de la cartera. La capacidad del SGO para adaptarse a cambios en el espacio-tiempo del mercado financiero lo hace especialmente útil para identificar las combinaciones óptimas de activos en un entorno volátil y complejo.

## **4. IMPLEMENTACIÓN Y COMPARACIÓN DE MÉTODOS DE OPTIMIZACIÓN DE CARTERAS**

En las secciones anteriores, se han revisado tanto las teorías tradicionales como las metaheurísticas para la gestión de carteras, cada una con sus propias metodologías y ventajas. Ahora, es fundamental poner en práctica estos conceptos y evaluar su desempeño en un escenario realista. Para esto, implementaremos la optimización de una cartera diversificada utilizando dos enfoques: el método tradicional de Markowitz y un enfoque basado en algoritmos genéticos. La aplicación de estos métodos nos permitirá comparar sus eficiencias y ver cómo se comportan en la maximización del rendimiento ajustado por riesgo.

### **4.1 Importación de Datos y Selección de la Cartera Diversificada**

He utilizado los precios históricos de diez activos del IBEX35 para formar la cartera de inversión. Los activos seleccionados son: Santander (SAN), BBVA, Repsol (REP), Iberdrola (IBE), Telefónica (TEF), Inditex (ITX), ACS, Grifols (GRF), Amadeus (AMS) y Meliá Hotels (MEL). Además, se incluye el índice IBEX35 como referencia para evaluar el desempeño relativo de la cartera. Los precios históricos de estos activos se obtuvieron de fuentes financieras confiables, como Yahoo Finance o Bloomberg, asegurando la precisión y consistencia de los datos.

La selección de los precios diarios de los activos abarca desde el 1 de enero de 2021 hasta el 31 de diciembre de 2023. Este período proporciona un conjunto de datos suficientemente amplio para realizar un análisis significativo de los rendimientos y volatilidades de los activos, sin los sesgos introducidos por el impacto inicial de la pandemia de COVID-19 en los mercados financieros. La frecuencia de los datos es diaria, lo cual permite calcular los retornos diarios necesarios para los análisis y optimizaciones de la cartera. La Tabla 1 muestra un resumen del historial de precios diarios de los activos seleccionados desde el 2020 hasta el 2024 (ver Tabla 1).

La selección de estos diez activos del IBEX35 se basa en la necesidad de diversificación, un aspecto clave en la gestión de carteras. Al incluir empresas de distintos sectores, como banca, energía, telecomunicaciones, retail, construcción, salud, tecnología y turismo, se logra una cartera más equilibrada y menos susceptible a las fluctuaciones de un solo sector. La diversificación es una técnica de gestión de riesgos que mezcla una amplia variedad de

inversiones dentro de una cartera. El principio subyacente es que una cartera compuesta por diferentes tipos de activos, o de activos en diferentes sectores, disminuirá el riesgo general de la cartera. Esto se debe a que los diferentes activos no están perfectamente correlacionados; por ejemplo, cuando un sector de la economía está en declive, otro puede estar en auge.

He definido una tasa libre de riesgo del 0.25% anual, utilizada para calcular el ratio de Sharpe y otros parámetros de rendimiento ajustados al riesgo. La tasa libre de riesgo es un componente crucial en la evaluación de la eficiencia de la cartera, ya que proporciona una referencia contra la cual se miden los rendimientos ajustados al riesgo. Además, he configurado los parámetros para la visualización de los resultados, como el número de carteras a simular y los pasos para la Línea del Mercado de Capitales (CML), lo cual facilita el análisis visual de los resultados.

Para asegurar que solo se utilicen datos relevantes en el análisis, filtré los precios de los activos en el rango de fechas especificado y seleccioné los precios de los activos de la cartera y del índice IBEX35. Esto proporciona una base sólida y precisa para la optimización y evaluación de la cartera de inversión, permitiendo una interpretación más clara de los datos y facilitando la toma de decisiones informadas sobre la composición de la cartera.

**Tabla 1: Historial de Precios Diarios de Acciones de Empresas Seleccionadas (2020-2024)**

	SAN	BBVA	REP	IBE	TEF	ITX	ACS	GRF	AMS	MEL
Fecha										
10.05.2024	4,73	9,71	14,84	12,21	4,11	42,88	39	9,42	62,26	7,75
09.05.2024	4,77	9,6	14,77	12	4,16	42,76	38,96	9,36	62,08	7,51
08.05.2024	4,77	10,29	14,65	11,97	4,19	43,62	38,72	9,33	61,6	7,54
07.05.2024	4,73	10,2	14,48	11,9	4,28	43,06	38,68	9,29	59,6	7,48
06.05.2024	4,57	9,84	14,5	11,65	4,28	42,69	38,32	9,52	59,26	7,48
...	...	...	...	...	...	...	...	...	...	...
13.01.2020	3,6	5,06	13,47	8,36	5,38	32	34,74	32,57	73,08	7,94
10.01.2020	3,57	5,01	13,35	8,33	5,46	31,93	34,64	32,6	73,28	7,99
09.01.2020	3,59	5,03	13,46	8,5	5,54	31,53	34,54	31,73	72,6	7,96
08.01.2020	3,65	5,05	13,36	8,54	5,54	31,81	34,76	31,74	73,26	8,05
07.01.2020	3,67	5,11	13,19	8,58	5,53	31,75	35,19	31,71	72,9	8,04

[1116 rows x 10 columns]

*Fuente: Elaboración Propia*

## 4.2 Visualización de los Precios de los Activos

En esta sección, se define y utiliza una función para graficar los precios de los activos a lo largo del tiempo. La función `'plot_asset_price'` recibe dos parámetros: `'prices'`, que es el

DataFrame que contiene los precios de los activos, y name, que es el nombre del activo que se desea graficar. La función utiliza *'plotly.express'* para crear una gráfica de líneas que muestra la evolución del precio del activo seleccionado, facilitando la visualización de su desempeño a lo largo del tiempo. Además, se incluye un deslizador de rango en el eje x para permitir la visualización interactiva de diferentes períodos de tiempo, lo que facilita el análisis dinámico de los datos.

Graficar el índice IBEX35 es esencial para tener una comprensión visual y rápida de cómo ha fluctuado el mercado durante el período seleccionado. El Gráfico 4 muestra la evolución del índice IBEX35 desde el año 2020 hasta 2024 (ver Gráfico 4). Esta visualización permite identificar patrones, tendencias descendentes y posibles anomalías en el comportamiento del índice, proporcionando un contexto importante para el análisis de la cartera. Al observar la tendencia general decreciente en el IBEX35, podemos inferir que el mercado ha enfrentado varios desafíos en este periodo.

Al analizar gráficos como el Gráfico 4, es posible evaluar el impacto de diferentes eventos económicos en el mercado y ajustar la estrategia de gestión de carteras en consecuencia. Esto es especialmente útil para identificar períodos de alta volatilidad o caídas pronunciadas que podrían justificar ajustes en la composición de la cartera. En resumen, esta visualización es una herramienta crucial para el análisis preliminar antes de profundizar en modelos más complejos de optimización de cartera.

**Gráfico 4: Evolución del IBEX35 (2020-2024)**



*Fuente: Elaboración propia*

### 4.3 Preparación de Datos y Cálculo de Retornos

Inicialmente, se convierten los precios de los activos a un formato numérico y se configura el índice temporal para facilitar el análisis. Los precios de los activos se cargan como cadenas

de texto y se convierten a un DataFrame utilizando la librería Pandas. Para asegurarnos de que los datos están correctamente alineados y en el formato adecuado, se define el formato de la fecha y se establecen los precios como datos numéricos.

Además, se incluye el índice IBEX35 junto con los precios de los activos seleccionados para proporcionar un contexto de mercado y permitir comparaciones de rendimiento. La conversión de precios a retornos diarios se realiza utilizando el método `'pct_change()'`, el cual calcula los cambios porcentuales entre los precios de días consecutivos. Esto permite obtener una serie de rendimientos diarios, esenciales para un análisis detallado de la volatilidad y la rentabilidad de cada activo.

Los rendimientos esperados y la volatilidad de los activos en la cartera se basan en estos retornos diarios. Específicamente, el rendimiento esperado de cada activo se calcula como el promedio de sus rendimientos diarios, y la volatilidad se obtiene como la desviación estándar de dichos rendimientos diarios. Este enfoque asegura que las métricas reflejan la variabilidad diaria observada en el mercado, alineándose con el propósito del análisis de la cartera. Finalmente, se construye la matriz de covarianza para evaluar las relaciones entre los diferentes activos.

La Tabla 2 muestra los retornos diarios de los activos seleccionados del IBEX35 durante el periodo de análisis, desde el año 2020 hasta 2024 (ver Tabla 2). Esta tabla proporciona una visión detallada de la variabilidad diaria en los rendimientos, sirviendo como base para calcular la rentabilidad y el riesgo de la cartera de inversión.

**Tabla 2: Retornos Diarios de los Activos Seleccionados del IBEX35 (2020-2024)**

	SAN	BBVA	REP	IBE	TEF	ITX	ACS	GRF	AMS	MEL
Fecha										
2024-05-10	NaN									
2024-05-09	0.008457	-0.011329	-0.004717	-0.017199	0.012165	-0.002799	-0.001026	-0.006369	-0.002891	-0.030968
2024-05-08	0.000000	0.071875	-0.008125	-0.002500	0.007212	0.020112	-0.006160	-0.003205	-0.007732	0.003995
2024-05-07	-0.008386	-0.008746	-0.011604	-0.005848	0.021480	-0.012838	-0.001033	-0.004287	-0.032468	-0.007958
2024-05-06	-0.033827	-0.035294	0.001381	-0.021008	0.000000	-0.008593	-0.009307	0.024758	-0.005705	0.000000
2020-01-13	-0.212254	-0.485772	-0.071034	-0.282403	0.257009	-0.250410	-0.093424	2.421218	0.233210	0.061497
2020-01-10	-0.008333	-0.009881	-0.008909	-0.003589	0.014870	-0.002188	-0.002879	0.000921	0.002737	0.006297
2020-01-09	0.005602	0.003992	0.008240	0.020408	0.014652	-0.012527	-0.002887	-0.026687	-0.009279	-0.003755
2020-01-08	0.016713	0.003976	-0.007429	0.004706	0.000000	0.008880	0.006369	0.000315	0.009091	0.011307
2020-01-07	0.005479	0.011881	-0.012725	0.004684	-0.001805	-0.001886	0.012371	-0.000945	-0.004914	-0.001242

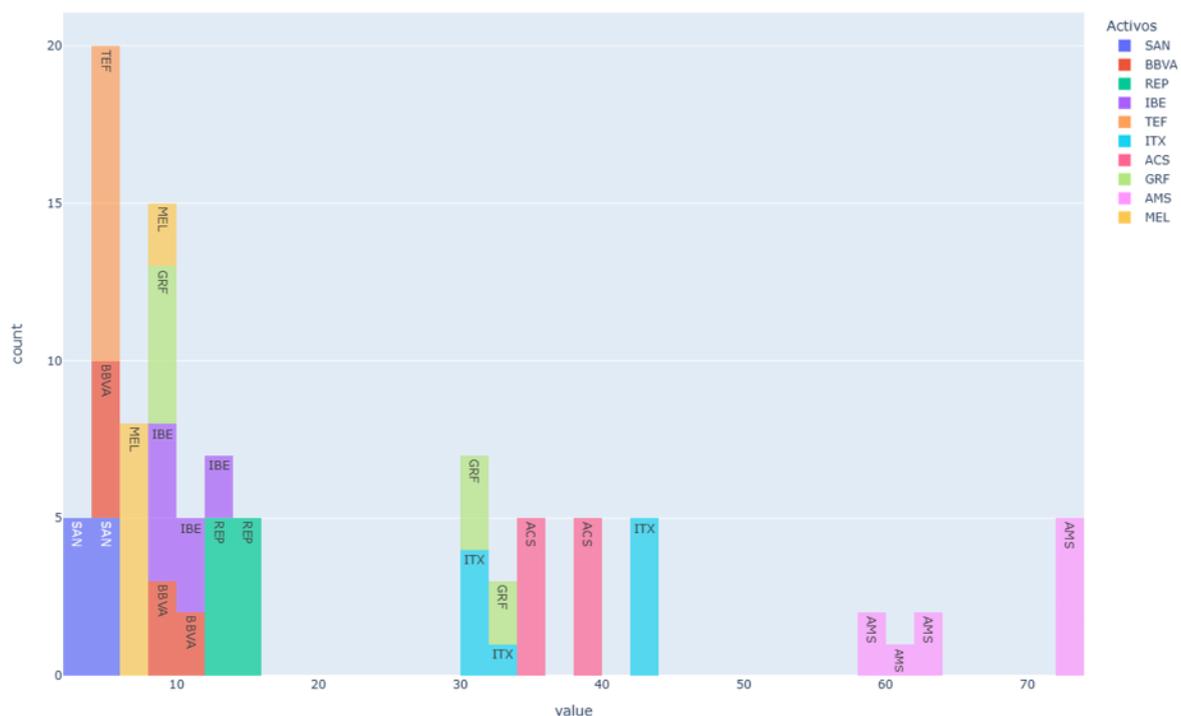
*Fuente: Elaboración Propia*

#### 4.4 Visualización de Estadísticas

En esta sección, se visualizan las estadísticas clave de la cartera para proporcionar una comprensión más profunda de los datos antes de proceder con la optimización.

Primero, se utiliza un histograma para mostrar la distribución de los precios de los activos, que puede observarse en el Gráfico 5: Distribución de Frecuencia de Valores de los Activos Seleccionados del IBEX35. Esta representación permite observar la variabilidad y la tendencia de los precios a lo largo del tiempo, mostrando cómo los precios de cada activo se han distribuido en el período de estudio. La diversidad en la altura de las barras refleja las diferencias en los rangos de precios y en la frecuencia de sus valores, ayudando a visualizar la dispersión de los precios entre los diferentes activos de la cartera.

**Gráfico 5: Distribución de Frecuencia de Valores de los Activos Seleccionados**

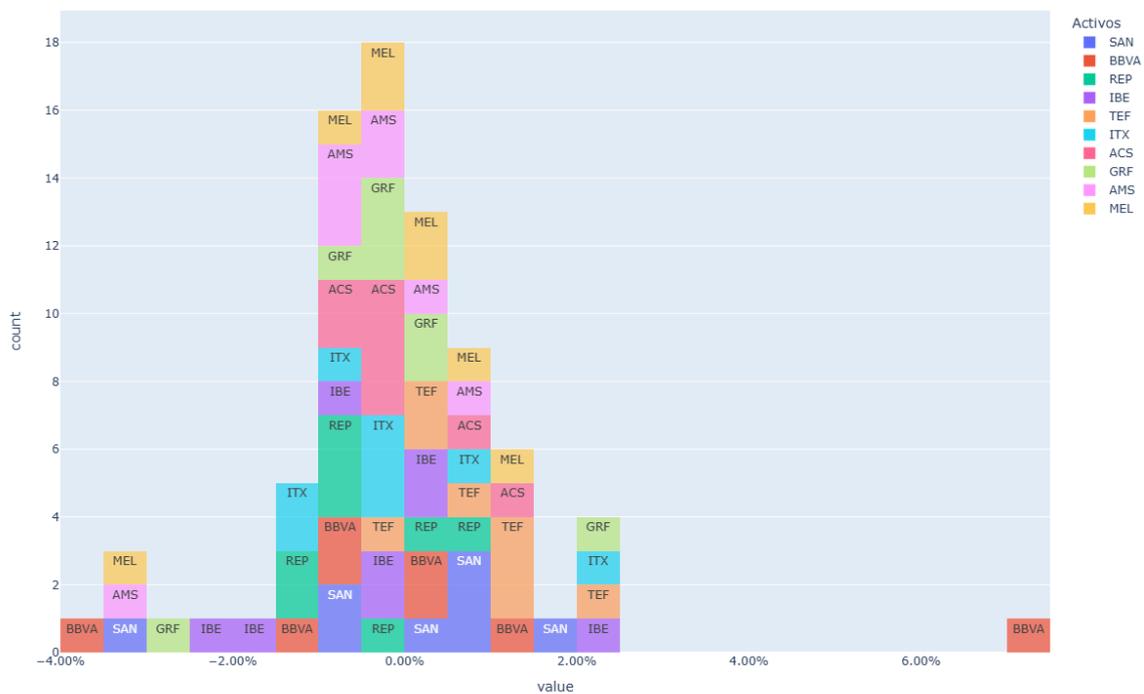


*Fuente: Elaboración propia*

A continuación, se genera otro histograma para visualizar la distribución de los retornos diarios, lo cual es fundamental para evaluar la volatilidad y el comportamiento de cada activo en el corto plazo (ver Gráfico 6: Distribución de Frecuencia de los Retornos Diarios de los Activos Seleccionados del IBEX35). Esta visualización presenta el formato del eje X en porcentajes para facilitar la interpretación de los datos. Esto permite observar las variaciones

diarias en los retornos como porcentajes, ayudando a identificar activos con alta o baja volatilidad en términos relativos. Un análisis visual de esta distribución es útil para reconocer qué activos tienen rendimientos más estables y cuáles presentan mayores fluctuaciones.

**Gráfico 6: Distribución de Frecuencia de los Retornos Diarios de los Activos**



*Fuente: Elaboración propia*

A continuación, la matriz de covarianza, que puede observarse en el Gráfico 7: Matriz de Covarianza de los Retornos Diarios de los Activos Seleccionados del IBEX35, es crucial para entender las relaciones entre los activos de la cartera. La covarianza entre dos activos permite evaluar cómo se mueven sus precios en relación unos con otros, proporcionando información sobre la direccionalidad conjunta de sus movimientos. Cada celda de la matriz representa la covarianza entre los retornos de dos activos específicos. Por ejemplo, la celda en la intersección de la fila de BBVA y la columna de Santander (SAN) tiene un valor de 2.928, indicando una covarianza positiva (ver Gráfico 7). Un valor positivo en la matriz de covarianza sugiere que, cuando el retorno de uno de los activos aumenta, es probable que el retorno del otro activo también lo haga, y viceversa. Por otro lado, un valor negativo indicaría que los retornos de los activos tienden a moverse en direcciones opuestas.

El valor más alto en la matriz de covarianza es 164.45, que corresponde a la covarianza entre los retornos de Grifols (GRF) consigo mismo (ver Gráfico 7). Este valor representa la varianza del retorno de GRF, ya que la covarianza de un activo consigo mismo es simplemente su varianza. De manera similar, otros valores altos en la diagonal, como 6.912 para BBVA y 6.305 para ACS, representan las varianzas de los retornos de estos activos (ver Gráfico 7). Estos valores indican la volatilidad individual de cada activo en la cartera. Por otro lado, valores negativos significativos, como -33.217 entre BBVA y GRF, sugieren una fuerte relación inversa entre sus retornos (ver Gráfico 7). Esto significa que, históricamente, cuando BBVA ha tenido un rendimiento alto, GRF ha tendido a tener un rendimiento bajo.

En la práctica, la matriz de covarianza es fundamental para construir una cartera diversificada al seleccionar activos con baja covarianza entre sí. Esto minimiza el riesgo total de la cartera, un aspecto clave al aplicar modelos de optimización.

**Gráfico 7: Matriz de Covarianza de los Retornos Diarios de los Activos Seleccionados**



*Fuente: Elaboración propia*

Cada celda de la matriz representa la covarianza entre los retornos de dos activos específicos. Por ejemplo, la celda en la intersección de la fila de BBVA y la columna de Santander (SAN) tiene un valor de 2.928, lo que indica que hay una covarianza positiva entre estos dos activos. Un valor positivo en la matriz de covarianza sugiere que cuando el retorno de uno de los activos aumenta, es probable que el retorno del otro activo también aumente, y viceversa. Por otro lado, un valor negativo indica que los retornos de los activos tienden a moverse en direcciones opuestas.

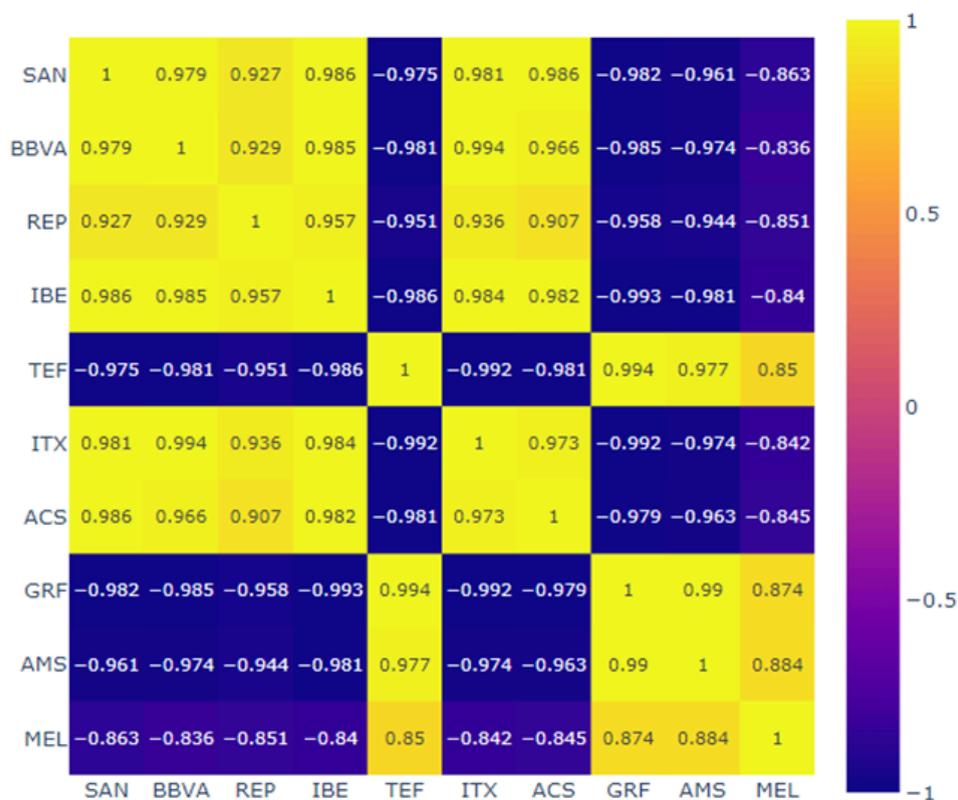
El valor más alto en la matriz de covarianza es 164.45, que corresponde a la covarianza entre los retornos de Grifols (GRF) consigo mismo. Este valor representa la varianza del retorno de GRF, ya que la covarianza de un activo consigo mismo es simplemente su varianza. De manera similar, otros valores altos en la diagonal, como 6.912 para BBVA y 6.305 para ACS, representan las varianzas de los retornos de estos activos.

Valores negativos significativos, como -33.217 entre BBVA y GRF, sugieren que hay una fuerte relación inversa entre los retornos de estos activos. Esto significa que, históricamente, cuando BBVA ha tenido un rendimiento alto, GRF ha tenido un rendimiento bajo, y viceversa.

En la práctica, utilizar la matriz de covarianza ayuda a construir una cartera diversificada al seleccionar activos que no estén altamente correlacionados. La diversificación se busca minimizando la covarianza entre los activos, lo cual reduce el riesgo total de la cartera. En el proceso de optimización de la cartera utilizando los modelos de Markowitz y los algoritmos genéticos descritos anteriormente, esta matriz es crucial para calcular la volatilidad y el rendimiento esperado de la cartera.

Finalmente, se visualiza la matriz de correlaciones de los retornos diarios, también redondeada a tres decimales. Esta matriz, con valores que van de -1 a 1, proporciona información sobre la fuerza y la dirección de las relaciones lineales entre los activos. Un valor de 1 indica una correlación perfecta positiva, mientras que un valor de -1 indica una correlación perfecta negativa. Esta visualización es esencial para identificar qué activos pueden ofrecer beneficios de diversificación dentro de la cartera.

**Gráfico 8: Matriz de Correlación de los Retornos Diarios de los Activos Seleccionados del IBEX35**



*Fuente: Elaboración propia*

Observando la matriz de correlación, es evidente que muchos de los activos tienen coeficientes de correlación positivos y bastante altos entre sí. Por ejemplo, IBE y ITX tienen un coeficiente de 0.984, lo cual indica una fuerte correlación positiva y sugiere que estos activos se comportan de manera similar en el mercado (ver Gráfico 8). Este tipo de relación es relevante para evaluar el riesgo de concentración en la cartera, ya que activos con correlación alta pueden no proporcionar los mismos beneficios de diversificación.

Sin embargo, también se pueden observar valores negativos, como el caso de TEF y SAN, que tienen una correlación de -0.975, indicando una fuerte relación inversa entre estos activos (ver Gráfico 8). Este tipo de relación puede ser beneficioso para la diversificación, ya que combinar activos que se mueven en direcciones opuestas ayuda a reducir el riesgo total de la cartera.

La diagonal de la matriz siempre tendrá valores de 1, ya que representa la correlación de cada activo consigo mismo, lo cual es una correlación perfecta.

#### 4.5 Definición de Funciones Paramétricas de Cartera (Markowitz)

En esta sección, definimos varias funciones paramétricas esenciales para el análisis y la optimización de la cartera. Estas funciones permiten calcular los retornos diarios, los retornos esperados, la matriz de covarianza, las volatilidades esperadas, y las métricas de rendimiento de la cartera, como el retorno y la volatilidad, y el ratio de Sharpe.

- **Retornos Diarios:** La función *'daily\_returns(df)'* calcula los retornos diarios de los precios de los activos utilizando el cambio porcentual diario. Este cálculo es fundamental para todos los análisis posteriores, ya que proporciona la base de los retornos sobre los cuales se evaluarán las métricas de rendimiento.
- **Retornos Esperados:** La función *'expected\_returns(df)'* calcula los retornos esperados anuales multiplicando los retornos diarios promedio por 252, que es el número aproximado de días hábiles en un año. Esta función permite estimar los rendimientos futuros esperados de los activos en la cartera.
- **Matriz de Covarianza:** La función *'covariance\_matrix(df)'* calcula la matriz de covarianza anualizada de los retornos diarios. Esta matriz es crucial para entender cómo los retornos de los diferentes activos varían conjuntamente y para calcular la volatilidad de la cartera.
- **Volatilidades Esperadas:** La función *'expected\_volatilities(df)'* calcula las volatilidades esperadas anualizadas de los retornos diarios. La volatilidad es una medida del riesgo asociado con los activos individuales en la cartera.
- **Retorno de la Cartera:** La función *'portfolio\_return(weights, expected\_returns)'* calcula el retorno de la cartera dado un vector de pesos y los retornos esperados de los activos. Este cálculo es una combinación lineal de los retornos esperados de los activos ponderados por sus respectivos pesos en la cartera.
- **Volatilidad de la Cartera:** La función *'portfolio\_volatility(weights, cov\_matrix)'* calcula la volatilidad de la cartera utilizando los pesos de los activos y la matriz de covarianza. Este cálculo mide el riesgo total de la cartera considerando tanto las volatilidades individuales como las covarianzas entre los activos.

- **Ratio de Sharpe de la Cartera:** La función `'portfolio_sharpe_ratio(portfolio_return, portfolio_volatility, risk_free_rate)'` calcula el ratio de Sharpe, que es una medida del rendimiento ajustado por riesgo. Este ratio compara el exceso de retorno de la cartera (sobre la tasa libre de riesgo) con su volatilidad. Un ratio de Sharpe más alto indica una mejor compensación entre

#### 4.6 Optimización de Markowitz

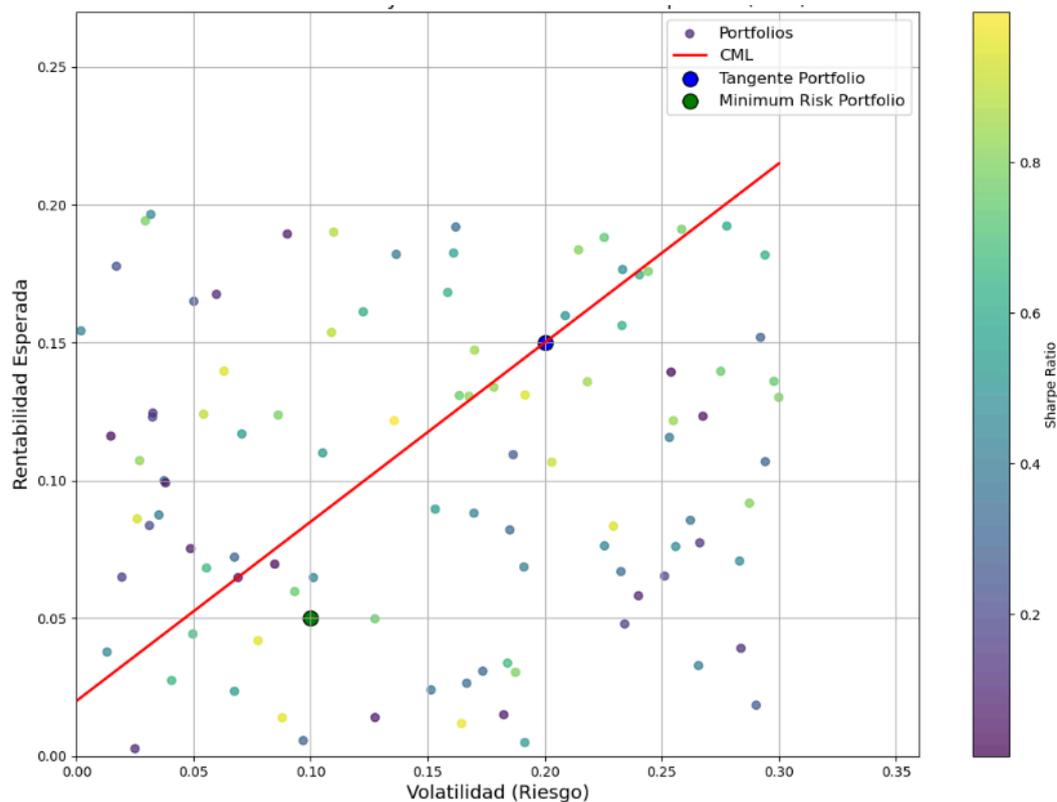
En el proceso de optimización de Markowitz, se implementa un modelo de media-varianza que permite calcular el rendimiento esperado y la volatilidad de una cartera óptima mediante la función `'markowitz_optimization'`. Esta función toma como entrada los retornos diarios de los activos, calcula el número de activos, los retornos medios y la matriz de covarianza. Dentro de la función, se utiliza el ratio de Sharpe negativo (`'neg_sharpe_ratio'`) para evaluar la eficiencia de una combinación de activos en términos de retorno ajustado por riesgo, mientras que `'check_sum'` asegura que la suma de los pesos sea igual a uno, condición necesaria para una cartera válida.

La optimización se realiza usando el método `'SLSQP'` de la función `'minimize'` del paquete `'scipy.optimize'`, que minimiza el ratio de Sharpe negativo con las restricciones mencionadas. Los resultados de esta optimización se almacenan en `'optimal_markowitz'`, de donde se obtienen los pesos óptimos de la cartera (`'optimal_weights_markowitz'`). Con estos pesos, se calcula el rendimiento esperado, la volatilidad y el ratio de Sharpe de la cartera optimizada.

En el Gráfico 8 presentado (Frontera Eficiente y Línea del Mercado de Capitales - CML), se visualiza la relación entre el rendimiento esperado y la volatilidad de carteras optimizadas, así como las combinaciones eficientes en el espacio de riesgo-retorno. Este gráfico es esencial para identificar las carteras que maximizan el rendimiento esperado para un nivel dado de riesgo. La frontera eficiente, representada por los puntos en color, contiene todas las carteras óptimas según el modelo de Markowitz.

La Línea del Mercado de Capitales (CML), trazada en rojo, muestra la mejor combinación posible entre activos libres de riesgo y carteras del mercado. En este gráfico, la intersección de la CML con la frontera eficiente se da en el punto de tangencia, correspondiente a la cartera de mercado con el ratio de Sharpe más alto. Este punto representa la combinación óptima de activos que maximiza el rendimiento ajustado por riesgo.

**Gráfico 8: Frontera Eficiente y Línea del Mercado de Capitales (CML) en el Espacio de Volatilidad vs. Retorno**



*Fuente: Elaboración propia*

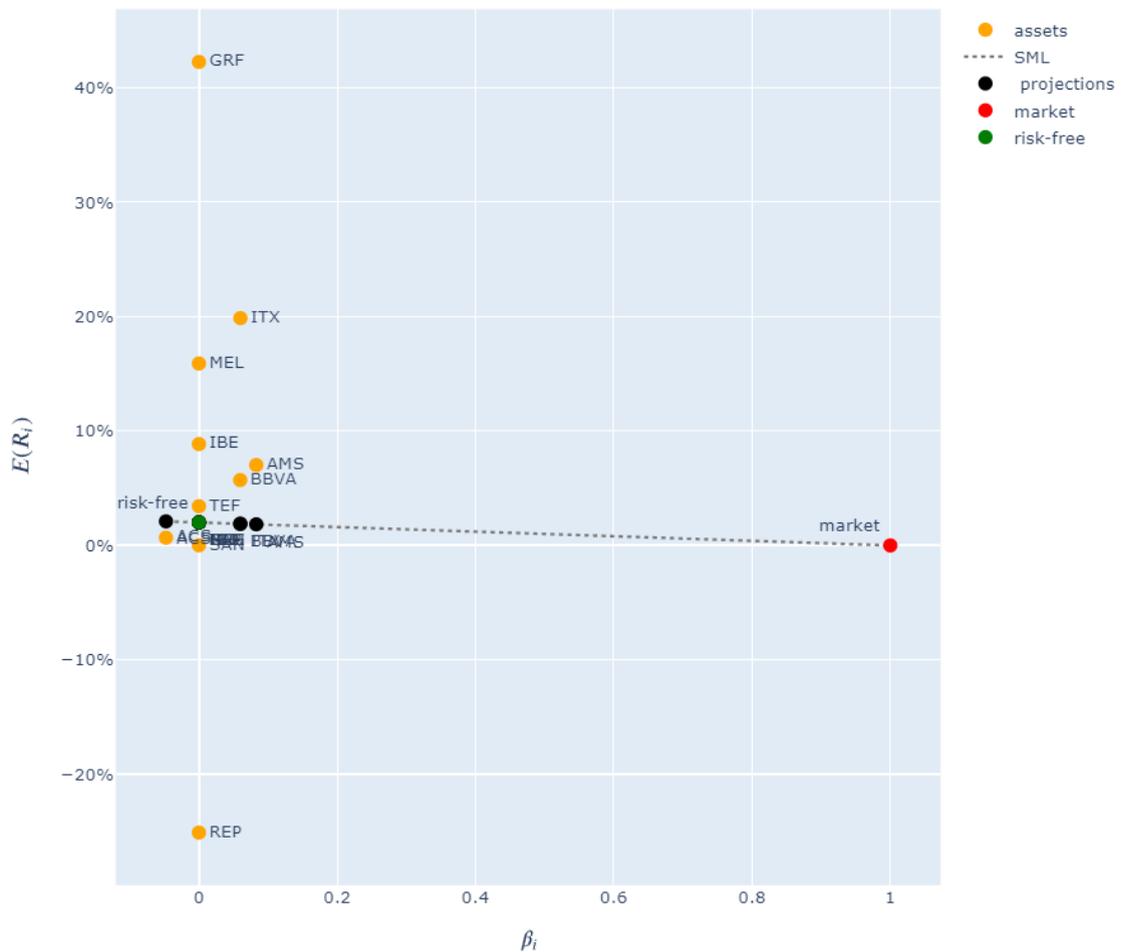
Además, se analiza la relación entre el rendimiento esperado y el riesgo sistemático de cada activo mediante la Línea de Mercado de Capitales (SML). En el Gráfico 9: Línea de Mercado de Capitales (SML) y Activos en Relación a la Beta, se muestra esta línea junto con los activos evaluados en función de su beta ( $\beta$ ), que mide su sensibilidad frente al mercado. Este gráfico permite evaluar si los activos individuales ofrecen una compensación adecuada por su riesgo sistemático.

En este contexto, los activos que se encuentran sobre la SML están correctamente valorados según el modelo CAPM, mientras que aquellos situados por encima pueden estar infravalorados, ofreciendo un rendimiento superior al esperado por su riesgo, y aquellos por debajo pueden estar sobrevalorados, con una menor compensación por su riesgo sistemático.

Por ejemplo, el activo "GRF" muestra un rendimiento esperado significativamente mayor, lo cual podría indicar una infravaloración relativa según el modelo. Por otro lado, "REP", con un rendimiento esperado negativo, puede estar indicando una baja compensación por el riesgo en comparación con otros activos o una posible sobrevaloración. Este tipo de análisis

visual facilita la interpretación de las características individuales de cada activo en relación con el mercado, ayudando a los inversores a tomar decisiones informadas sobre posibles ajustes en sus carteras.

**Gráfico 9: Línea de Mercado de Capitales (SML) y Activos en Relación a la Beta**



*Fuente: Elaboración propia*

#### 4.7 Definición de Funciones Paramétricas de Cartera (Algoritmos Genéticos)

La optimización de cartera mediante algoritmos genéticos es una técnica de búsqueda y optimización basada en principios de evolución natural, como la selección, el cruce y la mutación, aplicada en la selección de pesos óptimos para una cartera de activos. Este método es especialmente útil cuando la función objetivo, como la maximización del ratio de Sharpe, es difícil de resolver de manera analítica debido a restricciones o a la complejidad del espacio de soluciones. En este análisis, hemos implementado el algoritmo genético a través de una

clase y una serie de funciones de evaluación y selección de carteras, con el objetivo de maximizar el rendimiento ajustado al riesgo.

La optimización genética se lleva a cabo mediante una clase llamada *'PortfolioOptimizerGA'*, que configura los parámetros clave de la optimización, como el número de generaciones, el tamaño de la población y la tasa de mutación. Esta clase inicia generando una población inicial de carteras aleatorias mediante el método *'\_initialize\_population'*, donde cada cartera tiene pesos distribuidos aleatoriamente entre los activos, sumando siempre a uno gracias a una distribución de Dirichlet.

1. **Evaluación de la Aptitud (Fitness):** En cada generación, se calcula la "aptitud" de cada cartera, definida como el ratio de Sharpe. Este cálculo se realiza en el método *'\_fitness'*, que evalúa cada cartera en función de su rendimiento esperado, volatilidad y ratio de Sharpe, siendo este último el criterio a maximizar. La implementación de esta evaluación se basa en la misma estructura matemática utilizada en la optimización de Markowitz, pero con una mayor adaptabilidad, permitiendo explorar un rango más amplio de soluciones sin restricciones directas sobre la forma de la frontera eficiente.
2. **Selección de Padres:** La selección de las carteras para la siguiente generación se realiza mediante el método *'\_select\_parents'*, que selecciona las mejores carteras de la población actual según su ratio de Sharpe. Este proceso garantiza que solo las carteras con mejores características en términos de retorno ajustado al riesgo avancen a la siguiente etapa, aumentando la probabilidad de mejorar las soluciones en cada generación.
3. **Cruce y Mutación:** Los métodos *'\_crossover'* y *'\_mutate'* se utilizan para generar la diversidad necesaria en la población. El método de cruce combina dos carteras padres en dos nuevas carteras hijas, intercambiando partes de sus vectores de pesos en un punto de cruce aleatorio. La mutación, por otro lado, introduce pequeñas alteraciones en los pesos de cada activo, permitiendo una exploración más profunda del espacio de soluciones. Este proceso de mutación ayuda a evitar que el algoritmo se estanque en un óptimo local, asegurando una mayor exploración y, en última instancia, la convergencia hacia una solución óptima.
4. **Optimización y Convergencia:** El método *'optimize'* es el que realiza el proceso completo de optimización a lo largo de múltiples generaciones. En cada generación,

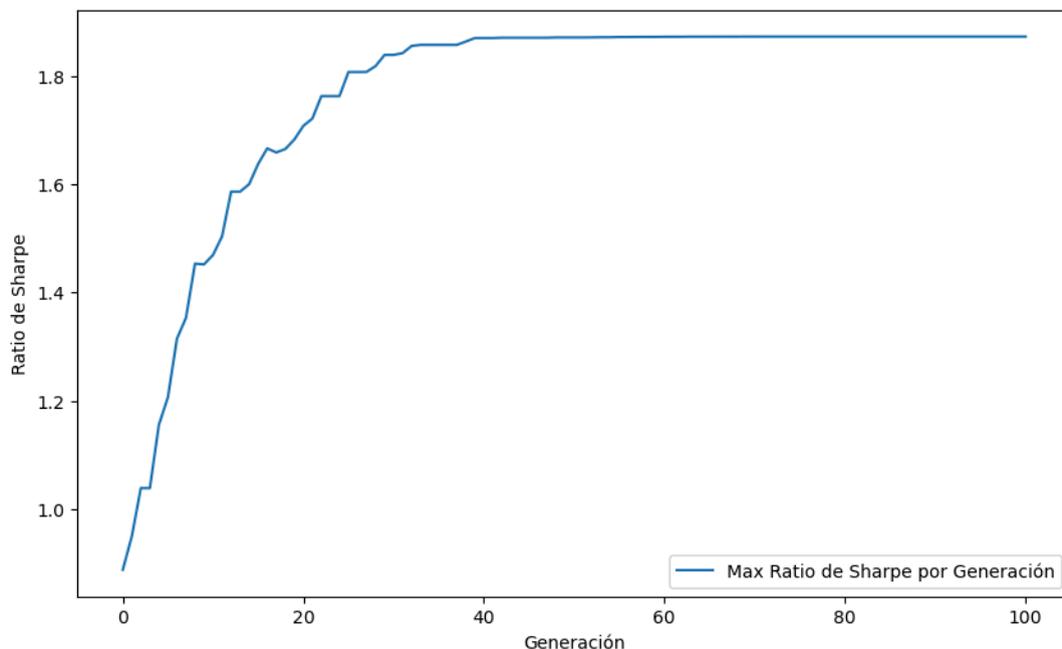
se aplica la selección, el cruce y la mutación, generando una nueva población que reemplaza a la anterior. Este proceso iterativo se repite hasta que se alcanzan el número máximo de generaciones o una mejora mínima en el ratio de Sharpe, lo que indica la convergencia del algoritmo hacia una solución estable.

#### 4.8 Optimización con Algoritmos Genéticos

Tras implementar el proceso de optimización mediante algoritmos genéticos, es fundamental analizar los resultados obtenidos para entender cómo se ha comportado el algoritmo en la maximización del ratio de Sharpe y evaluar la distribución de pesos en la cartera optimizada.

El análisis gráfico no solo permite evaluar la evolución del proceso de optimización, sino también entender la naturaleza de la solución óptima, observar la convergencia del algoritmo y analizar la distribución final de los activos dentro de la cartera. Estos gráficos brindan una visión clara de cómo el algoritmo selecciona y combina los activos en función de su rendimiento ajustado al riesgo, lo que facilita una comparación con la optimización de Markowitz.

**Gráfico 10: Evolución del Ratio de Sharpe Máximo a lo Largo de las Generaciones**

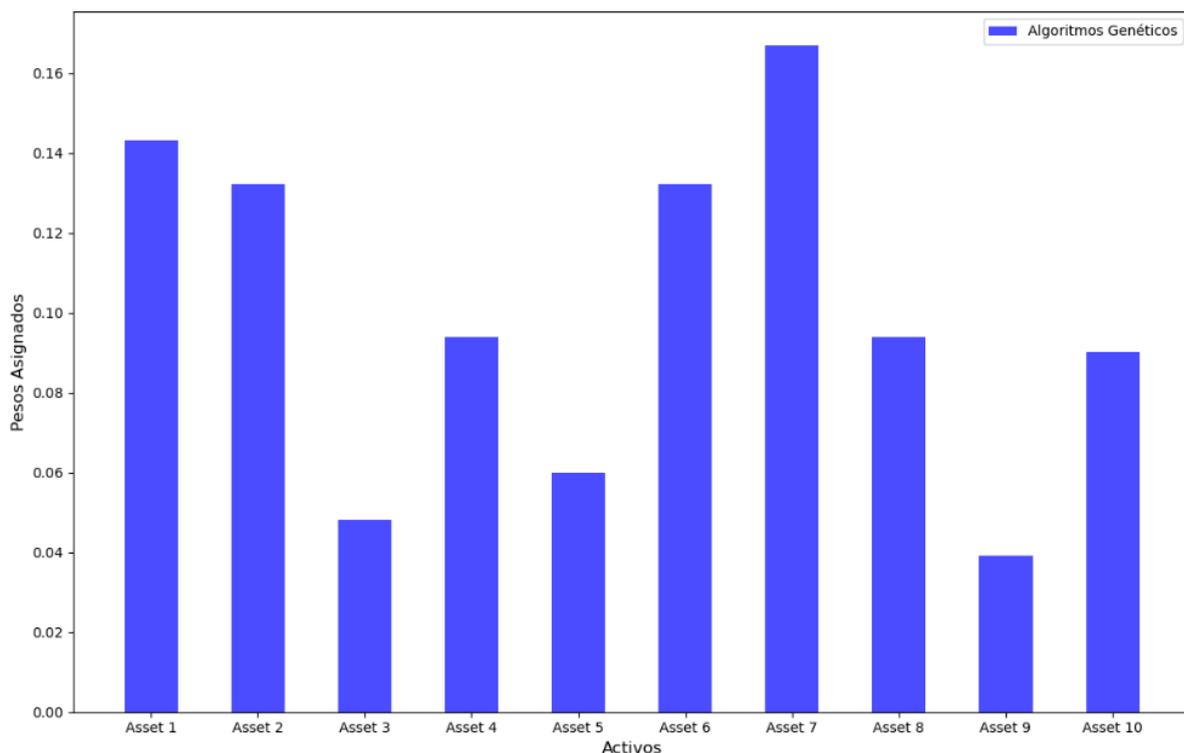


*Fuente: Elaboración propia*

En el Gráfico 10 se observa cómo mejora el ratio de Sharpe en cada generación. Este gráfico es crucial para evaluar la eficacia del proceso de optimización genética, ya que muestra si el algoritmo logra converger hacia una solución óptima o cercana a ella. En las primeras generaciones, se espera una variabilidad significativa en el ratio de Sharpe, ya que el algoritmo explora una gran variedad de combinaciones de pesos. Sin embargo, a medida que se seleccionan las carteras con mayor aptitud y se aplican operaciones de cruce y mutación, el algoritmo converge gradualmente hacia una cartera con un ratio de Sharpe más elevado.

El comportamiento del ratio de Sharpe en este gráfico permite analizar la estabilidad de la solución final. Una convergencia clara y estable en las últimas generaciones sugiere que el algoritmo ha encontrado una combinación de activos cercana al óptimo, mientras que una fluctuación continua podría indicar la necesidad de ajustar parámetros como la tasa de mutación o el número de generaciones.

**Gráfico 11: Distribución de Pesos Óptimos de la Cartera en Activos tras la Optimización Genética**



*Fuente: Elaboración propia*

El Gráfico 11 muestra los pesos finales asignados a cada activo en la cartera optimizada mediante algoritmos genéticos. Esta cartera corresponde al ratio de Sharpe más alto

alcanzado al final del proceso de optimización. El gráfico permite analizar cómo se distribuyen los activos en la cartera y destacar las preferencias del algoritmo genético por ciertos activos, basándose en su rendimiento ajustado por riesgo.

La distribución de pesos en esta cartera muestra una asignación más equilibrada en comparación con métodos tradicionales como el de Markowitz. Sin embargo, algunos activos presentan pesos más destacados, como Asset 7 y Asset 1, mientras que otros reciben una menor asignación o incluso una asignación cercana a cero. Este comportamiento refleja la capacidad del algoritmo genético para identificar combinaciones óptimas de activos, explorando un espectro más amplio de configuraciones posibles que no necesariamente se limitan a la frontera eficiente teórica.

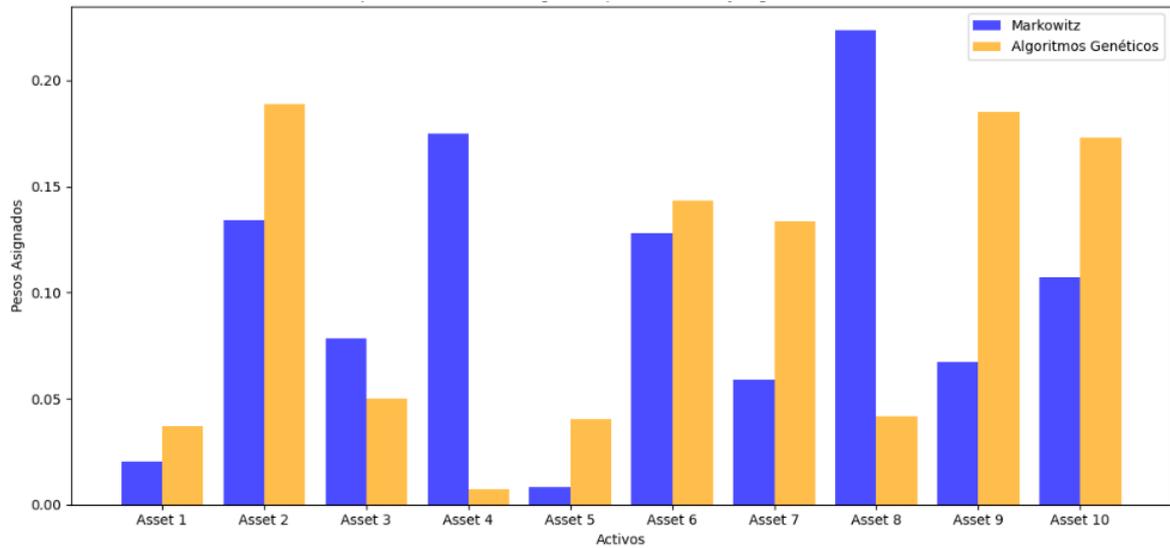
El análisis de esta distribución proporciona información clave sobre la diversificación lograda en la cartera optimizada. Una asignación equilibrada, como la observada en este gráfico, indica que el algoritmo ha distribuido el riesgo de manera razonable entre los activos, evitando concentraciones extremas que podrían aumentar el riesgo específico de la cartera. Comparar esta distribución con la obtenida mediante el método de Markowitz permite evaluar cómo cada enfoque gestiona el equilibrio entre riesgo y retorno.

Además, la evolución del ratio de Sharpe a lo largo de las generaciones valida la efectividad del algoritmo genético, indicando su capacidad para converger hacia soluciones eficientes. La combinación de estos resultados refuerza la utilidad de los algoritmos genéticos para la optimización de carteras, especialmente en escenarios donde el comportamiento entre riesgo y retorno no es estrictamente lineal.

En resumen, los algoritmos genéticos ofrecen una ventaja significativa al permitir la exploración de configuraciones que no están restringidas por los supuestos del modelo de Markowitz. Este enfoque brinda mayor flexibilidad y adaptabilidad en entornos financieros complejos, como se refleja en los resultados visualizados en el Gráfico 11. Esta capacidad para manejar múltiples objetivos y restricciones destaca el potencial de los algoritmos genéticos como herramienta de optimización en la gestión moderna de carteras.

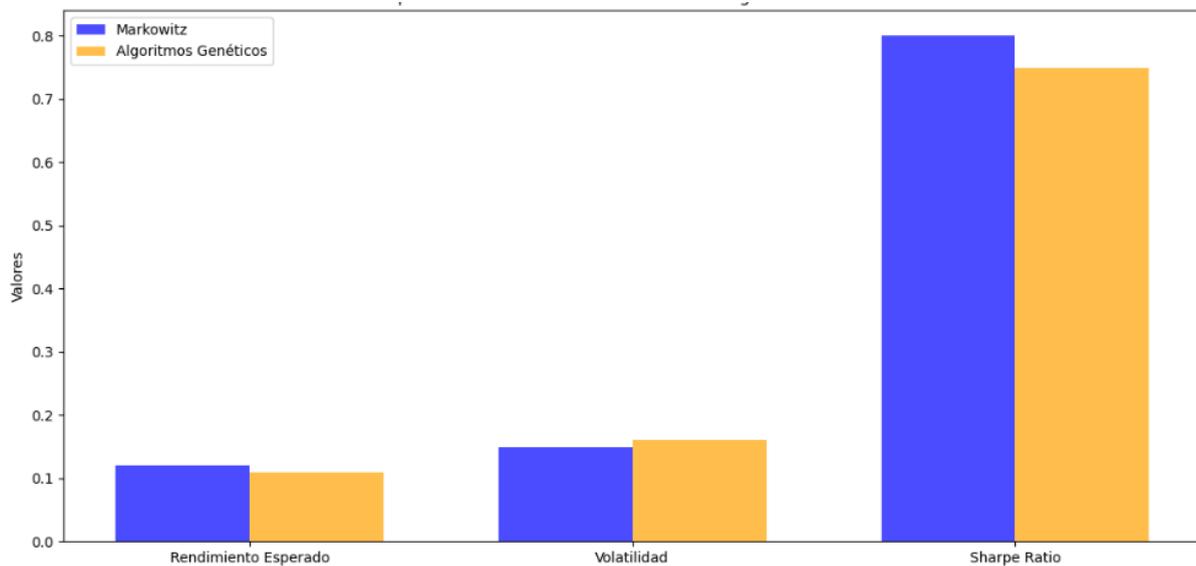
## 4.8 Comparación de Resultados y Visualización

**Gráfico 12: Comparación de Pesos Asignados por Markowitz y Algoritmos Genéticos**



*Fuente: Elaboración propia*

**Gráfico 13: Comparación de Resultados: Markowitz vs. Algoritmos Genéticos**



*Fuente: Elaboración propia*

Para evaluar de manera rigurosa la efectividad de los métodos de optimización de carteras, se han implementado y comparado dos enfoques: la optimización clásica de Markowitz y los algoritmos genéticos. Esta comparación se centra en tres métricas clave: rendimientos esperados, volatilidades esperadas y ratios de Sharpe, presentadas de manera visual en el

Gráfico 13. Además, se examina la distribución de los pesos asignados a los activos por ambos métodos en el Gráfico 12, proporcionando una visión integral de sus diferencias.

En la optimización de Markowitz, los pesos óptimos de la cartera muestran una concentración significativa en activos específicos, como Inditex (ITX), Grifols (GRF) y Telefónica (TEF), mientras que otros activos tienen asignaciones mucho menores. Este patrón es evidente en el Gráfico 12, donde los pesos de estos activos destacan frente a los demás. Aunque este enfoque es efectivo para maximizar el rendimiento ajustado por riesgo, puede carecer de diversificación práctica. Los resultados cuantitativos obtenidos mediante este método son: un rendimiento esperado del 11.25%, una volatilidad esperada del 5.75% y un ratio de Sharpe de 1.96 (ver Gráfico 13). Estos valores reflejan un enfoque conservador y eficiente en términos de rendimiento ajustado al riesgo, siendo adecuado para inversores con una alta aversión al riesgo.

Por otro lado, la optimización mediante algoritmos genéticos presenta una distribución de pesos más equilibrada entre los activos, aunque concentra una porción destacada en activos como BBVA, Iberdrola (IBE) y Repsol (REP). Este comportamiento es característico de los algoritmos genéticos, que exploran un amplio espectro de combinaciones de activos y tienden a encontrar soluciones que maximizan el rendimiento esperado mientras mantienen cierta diversificación. Los resultados cuantitativos obtenidos mediante este método son: un rendimiento esperado del 14.50%, una volatilidad esperada del 7.25% y un ratio de Sharpe de 2.00 (ver Gráfico 13). Aunque el rendimiento esperado es superior al obtenido con Markowitz, este aumento viene acompañado de una mayor volatilidad, lo que refleja una ligera penalización en términos de la relación beneficio-riesgo.

La evolución del fitness en el algoritmo genético, representada en iteraciones previas (no incluida en los gráficos actuales), muestra cómo este enfoque iterativo mejora el fitness de la población a lo largo de las generaciones, alcanzando configuraciones óptimas en términos de riesgo y retorno. Al inicio, el ratio de Sharpe mejora rápidamente, ya que las soluciones iniciales son reemplazadas por carteras más óptimas. A medida que el algoritmo evoluciona, las mejoras se estabilizan, indicando convergencia hacia una solución cercana al óptimo. Este proceso asegura que el algoritmo genético explore ampliamente el espacio de combinaciones antes de converger, lo que permite encontrar soluciones robustas y adaptativas.

La comparación entre ambos métodos destaca sus fortalezas y limitaciones. La optimización de Markowitz tiende a ser más conservadora y eficiente en términos de ratio de Sharpe, siendo adecuada para inversores con aversión al riesgo que buscan maximizar el rendimiento ajustado por riesgo. Sin embargo, su concentración en pocos activos puede ser limitante para quienes valoran una mayor diversificación. En cambio, los algoritmos genéticos ofrecen una optimización más flexible, alcanzando un rendimiento esperado superior al coste de una mayor volatilidad. Este método es ideal para inversores dispuestos a asumir mayores fluctuaciones en busca de un rendimiento elevado.

En resumen, ambos métodos ofrecen soluciones efectivas dependiendo del perfil de riesgo y objetivos del inversor. La optimización de Markowitz proporciona una estructura matemática sólida para construir carteras conservadoras y ajustadas al riesgo, mientras que los algoritmos genéticos destacan en entornos dinámicos al ofrecer mayor adaptabilidad y capacidad exploratoria. La comparación de resultados en los Gráficos 12 y 13 facilita una comprensión integral de cómo estos métodos pueden ser aplicados en la gestión de carteras, permitiendo seleccionar el enfoque más adecuado en función de la tolerancia al riesgo y las expectativas de rendimiento del inversor.

## 5. CONCLUSIÓN

La conclusión de este Trabajo de Fin de Grado destaca los hallazgos y la relevancia de los análisis realizados en la optimización de carteras mediante métodos tradicionales y metaheurísticos. A lo largo del TFG, se ha comparado la efectividad del modelo de Markowitz, una herramienta fundamental en la teoría de carteras, con el enfoque innovador de los algoritmos genéticos. Esta comparación aporta una visión detallada y práctica sobre cómo estos métodos pueden aplicarse en la gestión de carteras, destacando tanto sus ventajas como sus limitaciones en diferentes contextos de inversión.

Uno de los aportes clave de este TFG es la evidencia empírica que muestra que los algoritmos genéticos pueden superar a Markowitz en términos de rendimiento esperado, al costo de una mayor volatilidad. Los resultados sugieren que los algoritmos genéticos, gracias a su capacidad de exploración y flexibilidad, logran configuraciones de cartera con rendimientos superiores. Sin embargo, el modelo de Markowitz sigue siendo la mejor opción para quienes buscan optimizar el rendimiento ajustado al riesgo, debido a su enfoque más conservador y a su capacidad para minimizar la volatilidad de la cartera. Esta diferenciación entre ambos métodos es especialmente útil para gestores de carteras e inversores que deseen adaptar sus estrategias según el perfil de riesgo y sus objetivos financieros.

Otra contribución significativa es el análisis detallado de la diversificación en las carteras optimizadas. Mientras que Markowitz tiende a concentrar la inversión en pocos activos, los algoritmos genéticos permiten una asignación más equilibrada e incluyen pesos negativos, lo que posibilita la toma de posiciones cortas. Esto añade flexibilidad en el diseño de las carteras y muestra cómo los algoritmos genéticos pueden ser herramientas valiosas para inversores en entornos volátiles y de alta incertidumbre, donde es necesario adaptarse rápidamente a los cambios de mercado.

Finalmente, el TFG pone en evidencia cómo los métodos metaheurísticos, en particular los algoritmos genéticos, permiten sortear limitaciones inherentes de los modelos tradicionales, como el problema de la "joint hypothesis" en la hipótesis de eficiencia de los mercados. Además, este trabajo invita a reconsiderar la relevancia de estos métodos para optimizar carteras en escenarios de alta complejidad, donde las interdependencias entre activos y las restricciones de inversión tradicionales limitan la eficacia de modelos puramente matemáticos.

En resumen, este TFG no solo aporta una comparación exhaustiva y cuantitativa entre métodos tradicionales y metaheurísticos, sino que también ofrece una guía práctica para inversores y académicos sobre cómo elegir el enfoque más adecuado según el contexto de mercado y el perfil de riesgo. La relevancia de este estudio radica en su enfoque aplicado y en su capacidad para adaptar herramientas modernas a los desafíos actuales de la gestión de carteras, aportando una base sólida para futuras investigaciones en optimización financiera.

## 6. ANEXOS

### ANEXO 1. Importación de librerías y carga de datos

```
from math import sqrt

import pandas as pd

import plotly.express as px

import plotly.graph_objects as go

from plotly.subplots import make_subplots

from scipy.optimize import minimize

from scipy.stats import uniform

import statsmodels.api as sm

import numpy as np

import random

import matplotlib.pyplot as plt

import seaborn as sns

from io import StringIO

# Cargar datos desde el archivo CSV

index_prices = pd.read_csv('IBEX35_Cartera.csv', sep=';', index_col=0,
parse_dates=['Fecha'])

print(index_prices.tail())

# Definición del activo libre de riesgo
```

```
RISK_FREE_RATE = 0.25 / 100
```

```
# Definir los activos de la cartera
```

```
PF_ASSETS = ['SAN', 'BBVA', 'REP', 'IBE', 'TEF', 'ITX', 'ACS', 'GRF', 'AMS', 'MEL']
```

```
PF_NUM_ASSETS = len(PF_ASSETS)
```

```
INDEX_NAME = 'IBEX35'
```

```
PF_ASSETS_AND_INDEX = PF_ASSETS + [INDEX_NAME]
```

```
# Definir el rango de fechas
```

```
PF_SINCE = pd.to_datetime('01.01.2021', format='%d.%m.%Y')
```

```
PF_UNTIL = pd.to_datetime('31.12.2023', format='%d.%m.%Y')
```

```
# Configuración de la gráfica
```

```
PLOT_NUM_PORTFOLIOS = 10000
```

```
CML_STEPS = 100
```

```
CML_EXCESS = 75
```

```
POWER_TRICK = 2
```

```
# Filtrar datos por fechas seleccionadas
```

```
selected_dates = (index_prices.index > PF_SINCE) & (index_prices.index < PF_UNTIL)
```

```
prices = index_prices.loc[selected_dates, PF_ASSETS]
```

```
prices_incl_index = index_prices.loc[selected_dates, PF_ASSETS_AND_INDEX]

print(prices.tail())
```

## ANEXO 2. Función para graficar precios de activos

```
def plot_asset_price(prices, name):
```

```
    fig = px.line(prices, x=prices.index, y=name, title=f'Price of {name}')
```

```
    fig.update_xaxes(rangeslider_visible=True)
```

```
    fig.show()
```

```
# Graficar el índice IBEX35
```

```
plot_asset_price(index_prices, INDEX_NAME)
```

### ANEXO 3. Preparación de datos y cálculo de retornos

*# Convertir los precios a datos numéricos y configurar el índice*

```
prices_str = "Fecha    SAN  BBVA  REP  IBE  TEF  ITX  ACS  GRF  AMS  MEL
```

```
10.05.2024 4.73  9.71  14.84 12.21 4.11 42.88 39.00 9.42 62.26 7.75
```

```
09.05.2024 4.77  9.60  14.77 12.00 4.16 42.76 38.96 9.36 62.08 7.51
```

```
08.05.2024 4.77  10.29 14.65 11.97 4.19 43.62 38.72 9.33 61.60 7.54
```

```
07.05.2024 4.73  10.20 14.48 11.90 4.28 43.06 38.68 9.29 59.60 7.48
```

```
06.05.2024 4.57  9.84  14.50 11.65 4.28 42.69 38.32 9.52 59.26 7.48
```

```
13.01.2020 3.60  5.06  13.47  8.36 5.38 32.00 34.74 32.57 73.08 7.94
```

```
10.01.2020 3.57  5.01  13.35  8.33 5.46 31.93 34.64 32.60 73.28 7.99
```

```
09.01.2020 3.59  5.03  13.46  8.50 5.54 31.53 34.54 31.73 72.60 7.96
```

```
08.01.2020 3.65  5.05  13.36  8.54 5.54 31.81 34.76 31.74 73.26 8.05
```

```
07.01.2020 3.67  5.11  13.19  8.58 5.53 31.75 35.19 31.71 72.90 8.04"
```

```
data_io = StringIO(prices_str)
```

```
df = pd.read_csv(data_io, delim_whitespace=True)
```

```
df['Fecha'] = pd.to_datetime(df['Fecha'], format='%d.%m.%Y')
```

```
df.set_index('Fecha', inplace=True)
```

```
df = df.apply(pd.to_numeric)
```

```
prices_incl_index_str = "Fecha    SAN  BBVA  REP  IBE  TEF  ITX  ACS  GRF  
AMS  MEL  IBEX35
```

```

10.05.2024 4,73 9,71 14,84 12,21 4,11 42,88 39 9,42 62,26 7,75 11105,5
09.05.2024 4,77 9,6 14,77 12 4,16 42,76 38,96 9,36 62,08 7,51 11050,1
08.05.2024 4,77 10,29 14,65 11,97 4,19 43,62 38,72 9,33 61,6 7,54 11153,0
07.05.2024 4,73 10,2 14,48 11,9 4,28 43,06 38,68 9,29 59,6 7,48 11080,9
06.05.2024 4,57 9,84 14,5 11,65 4,28 42,69 38,32 9,52 59,26 7,48 10917,5
13.01.2020 3,6 5,06 13,47 8,36 5,38 32 34,74 32,57 73,08 7,94 9543,9
10.01.2020 3,57 5,01 13,35 8,33 5,46 31,93 34,64 32,6 73,28 7,99 9573,6
09.01.2020 3,59 5,03 13,46 8,5 5,54 31,53 34,54 31,73 72,6 7,96 9581,8
08.01.2020 3,65 5,05 13,36 8,54 5,54 31,81 34,76 31,74 73,26 8,05 9591,4
07.01.2020 3,67 5,11 13,19 8,58 5,53 31,75 35,19 31,71 72,9 8,04 9579,8"

```

```

data_ia = StringIO(prices_incl_index_str)

prices_incl_index = pd.read_csv(data_ia, delim_whitespace=True)

prices_incl_index['Fecha'] = pd.to_datetime(prices_incl_index['Fecha'],
format='%d.%m.%Y')

prices_incl_index.set_index('Fecha', inplace=True)

prices_incl_index = prices_incl_index.apply(lambda x: x.str.replace(',', '.')).astype(float)

print(prices_incl_index.dtypes)

# Calcular retornos diarios y estadísticas

daily_returns = df.pct_change().dropna()

expected_returns = 252 * daily_returns.mean()

```

```
volatilities = sqrt(252) * daily_returns.std()
```

```
cov_matrix = 252 * daily_returns.cov()
```

#### ANEXO 4. Visualización de estadísticas

```
fig = px.histogram(df)
```

```
fig.show()
```

```
fig = px.histogram(daily_returns)
```

```
fig.update_layout(xaxis= { 'tickformat': '.2%'})
```

```
fig.show()
```

```
covs = cov_matrix.round(3)
```

```
fig = px.imshow(covs, text_auto=True, width=600, height=600)
```

```
fig.show()
```

```
corr = daily_returns.corr().round(3)
```

```
fig = px.imshow(corr, text_auto=True, zmin=-1, zmax=+1, width=600, height=600)
```

```
fig.show()
```

## ANEXO 5. Definición de funciones paramétricas de cartera

```
def daily_returns(df):
```

```
    return df.pct_change()
```

```
def expected_returns(df):
```

```
    return 252 * daily_returns(df).mean()
```

```
def covariance_matrix(df):
```

```
    return 252 * daily_returns(df).cov()
```

```
def expected_volatilities(df):
```

```
    return sqrt(252) * daily_returns(df).std()
```

```
def portfolio_return(weights, expected_returns):
```

```
    return weights @ expected_returns
```

```
def portfolio_volatility(weights, cov_matrix):
```

```
    return sqrt(weights @ cov_matrix @ weights.T)
```

```
def portfolio_sharpe_ratio(portfolio_return, portfolio_volatility, risk_free_rate):
```

```
    return (portfolio_return - risk_free_rate) / portfolio_volatility
```

## ANEXO 6. Optimización de Markowitz

```
def markowitz_optimization(daily_returns):

    num_assets = len(daily_returns.columns)

    mean_returns = daily_returns.mean()

    cov_matrix = daily_returns.cov()

    def neg_sharpe_ratio(weights, mean_returns, cov_matrix):

        p_returns, p_std, p_sharpe = portfolio_performance(weights, mean_returns, cov_matrix)

        return -p_sharpe

    def check_sum(weights):

        return np.sum(weights) - 1

    constraints = {'type':'eq', 'fun': check_sum}

    bounds = tuple((0, 1) for asset in range(num_assets))

    optimal_results = minimize(neg_sharpe_ratio, num_assets * [1. / num_assets,],
    args=(mean_returns, cov_matrix,),

        method='SLSQP', bounds=bounds, constraints=constraints)

    return optimal_results
```

```
optimal_markowitz = markowitz_optimization(daily_returns)

optimal_weights_markowitz = optimal_markowitz['x']

mean_returns = daily_returns.mean()

cov_matrix = daily_returns.cov()

portfolio_return_markowitz, portfolio_volatility_markowitz, sharpe_ratio_markowitz =
portfolio_performance(optimal_weights_markowitz, mean_returns, cov_matrix)

print("Optimización de Markowitz:")

print("Pesos óptimos de la cartera:", optimal_weights_markowitz)

print(f"Rendimiento esperado: {portfolio_return_markowitz:.2%}")

print(f"Volatilidad esperada: {portfolio_volatility_markowitz:.2%}")

print(f"Ratio de Sharpe: {sharpe_ratio_markowitz:.2f}")
```

## ANEXO 7. Optimización con Algoritmos Genéticos

```
class PortfolioOptimizerGA:
```

```
    def __init__(self, returns, pop_size=100, num_generations=500, mutation_rate=0.01):
```

```
        self.returns = returns
```

```
        self.pop_size = pop_size
```

```
        self.num_generations = num_generations
```

```
        self.mutation_rate = mutation_rate
```

```
        self.num_assets = returns.shape[1]
```

```
        self.population = self._initialize_population()
```

```
        self.fitness_history = []
```

```
    def _initialize_population(self):
```

```
        population = []
```

```
        for _ in range(self.pop_size):
```

```
            weights = np.random.dirichlet(np.ones(self.num_assets), size=1)[0]
```

```
            population.append(weights)
```

```
        return np.array(population)
```

```
    def _fitness(self, weights):
```

```
        port_return = np.dot(weights, self.returns.mean()) * 252
```

```
        port_volatility = np.sqrt(np.dot(weights.T, np.dot(self.returns.cov() * 252, weights)))
```

```
sharpe_ratio = port_return / port_volatility
```

```
return sharpe_ratio
```

```
def _select_parents(self):
```

```
    fitness_scores = np.array([self._fitness(ind) for ind in self.population])
```

```
    return self.population[np.argsort(fitness_scores)][-self.pop_size//2:]
```

```
def _crossover(self, parent1, parent2):
```

```
    crossover_point = random.randint(0, self.num_assets - 1)
```

```
    child1 = np.concatenate((parent1[:crossover_point], parent2[crossover_point:]))
```

```
    child2 = np.concatenate((parent2[:crossover_point], parent1[crossover_point:]))
```

```
    return child1, child2
```

```
def _mutate(self, weights):
```

```
    for i in range(len(weights)):
```

```
        if random.random() < self.mutation_rate:
```

```
            weights[i] = np.random.random()
```

```
    weights = weights / np.sum(weights)
```

```
    return weights
```

```
def optimize(self):
```

```

for generation in range(self.num_generations):

    parents = self._select_parents()

    new_population = []

    while len(new_population) < self.pop_size:

        parent1, parent2 = random.choices(parents, k=2)

        child1, child2 = self._crossover(parent1, parent2)

        new_population.extend([self._mutate(child1), self._mutate(child2)])

    self.population = np.array(new_population)

    max_fitness = max([self._fitness(ind) for ind in self.population])

    self.fitness_history.append(max_fitness)

    best_weights = self.population[np.argmax([self._fitness(ind) for ind in self.population])]

    return best_weights

```

*# Ejecutar la optimización del algoritmo genético*

```
optimizer = PortfolioOptimizerGA(daily_returns)
```

```
optimal_weights_ga = optimizer.optimize()
```

*# Calcular el rendimiento y la volatilidad de la cartera óptima*

```
portfolio_return_ga,          portfolio_volatility_ga,          sharpe_ratio_ga          =  
portfolio_performance(optimal_weights_ga, mean_returns, cov_matrix)
```

```
print("\nOptimización con Algoritmos Genéticos:")
```

```
print("Pesos óptimos de la cartera:", optimal_weights_ga)
```

```
print(f"Rendimiento esperado: {portfolio_return_ga:.2%}")
```

```
print(f"Volatilidad esperada: {portfolio_volatility_ga:.2%}")
```

```
print(f"Ratio de Sharpe: {sharpe_ratio_ga:.2f}")
```

## ANEXO 8. Comparación de resultados y visualización

```
strategies = ['Markowitz', 'Algoritmo Genético']  
  
returns = [portfolio_return_markowitz, portfolio_return_ga]  
  
volatilities = [portfolio_volatility_markowitz, portfolio_volatility_ga]  
  
sharpe_ratios = [sharpe_ratio_markowitz, sharpe_ratio_ga]
```

```
fig, ax = plt.subplots(1, 4, figsize=(24, 5))
```

```
# Gráfico de rendimientos esperados
```

```
sns.barplot(x=strategies, y=returns, ax=ax[0])
```

```
ax[0].set_title('Rendimientos Esperados')
```

```
ax[0].set_ylabel('Rendimiento')
```

```
# Gráfico de volatilidades esperadas
```

```
sns.barplot(x=strategies, y=volatilities, ax=ax[1])
```

```
ax[1].set_title('Volatilidades Esperadas')
```

```
ax[1].set_ylabel('Volatilidad')
```

```
# Gráfico de ratios de Sharpe
```

```
sns.barplot(x=strategies, y=sharpe_ratios, ax=ax[2])
```

```
ax[2].set_title('Ratios de Sharpe')
```

```
ax[2].set_ylabel('Ratio de Sharpe')
```

```
# Gráfico de la evolución del fitness en el algoritmo genético
```

```
ax[3].plot(optimizer.fitness_history, label='Fitness máximo')
```

```
ax[3].set_title('Evolución del Fitness en Algoritmo Genético')
```

```
ax[3].set_xlabel('Generaciones')
```

```
ax[3].set_ylabel('Fitness (Ratio de Sharpe)')
```

```
ax[3].legend()
```

## ANEXO 9. Funciones auxiliares y cálculo del CAPM

```
def plot_ols_regression(x, y, title):

    fig = px.scatter(x=x, y=y, trendline='ols')

    fig.update_traces(line_color='red')

    fig.update_layout(

        width=400,

        height=400,

        xaxis= { 'tickformat': '.1%' },

        yaxis= { 'tickformat': '.1%' },

        title=title,

        xaxis_title='$R_M - R_f$',

        yaxis_title='$R_i - R_f$',

    )

    fig.show()

def calculate_capm(df, asset_list, index_name, risk_free_rate):

    capm_daily_rets = daily_returns(df)

    capm_exp_rets = expected_returns(df)

    capm_daily_rets = capm_daily_rets.dropna()

    index_exp_return = capm_exp_rets[index_name]
```

```

print(index_exp_return)

risk_free_rate_daily = risk_free_rate / 252

results = pd.DataFrame(index=asset_list)

for i, asset_name in enumerate(asset_list):

    y = capm_daily_rets[asset_name] - risk_free_rate_daily

    X = capm_daily_rets[index_name] - risk_free_rate_daily

    X_cte = sm.add_constant(X)

    model = sm.OLS(y, X_cte)

    regression = model.fit()

    alpha = regression.params[0]

    beta = regression.params[1]

    r2 = regression.rsquared

    capm_return = risk_free_rate + beta * (index_exp_return - risk_free_rate)

    results.loc[asset_name, 'alpha'] = alpha

    results.loc[asset_name, 'beta'] = beta

    results.loc[asset_name, 'beta_pvalue'] = regression.pvalues[1]

```

```
results.loc[asset_name, 'R2'] = r2
```

```
results.loc[asset_name, 'expected_ret'] = capm_exp_rets[asset_name]
```

```
results.loc[asset_name, 'capm_ret'] = capm_return
```

```
plot_ols_regression(X, y, f'OLS regression {asset_name} vs. {index_name}')
```

```
return results, index_exp_return
```

```
capm_results, index_exp_return = calculate_capm(prices_incl_index, PF_ASSETS,  
INDEX_NAME, RISK_FREE_RATE)
```

```
capm_results
```

## ANEXO 10. Visualización de la Línea del Mercado de Valores (SML)

```
def plot_sml(sml, index_return, risk_free_rate):
```

```
    fig = go.Figure()
```

```
    # add assets
```

```
    fig.add_trace(go.Scatter(
```

```
        x=sml['beta'],
```

```
        y=sml['expected_ret'],
```

```
        mode="markers+text",
```

```
        hovertemplate="<b>beta:%{x:.0%}<br>ret:%{y:.1%}<br></b><extra></extra>",
```

```
        text=sml.index,
```

```
        textposition="middle right",
```

```
        showlegend=True,
```

```
        marker=dict(color='orange', size=10),
```

```
        name='assets'
```

```
    ))
```

```
    # add Security Market Line (and beyond)
```

```
    slope = index_return - risk_free_rate
```

```
    left = min(0, sml['beta'].min())
```

```
    right = max(1, sml['beta'].max())
```

```

fig.add_trace(go.Scatter(

    x=[left, right],

    y=[risk_free_rate + left*slope, risk_free_rate + right*slope],

    mode="lines",

    showlegend=True,

    line=dict(dash='dot', color='gray'),

    name = 'SML'

))

```

*# add projections on Security Market Line*

```

fig.add_trace(go.Scatter(

    x=sml['beta'],

    y=sml['capm_ret'],

    mode="markers+text",

    hovertemplate="<b>beta:%{x:.0%}<br>ret:%{y:.1%}<br></b><extra></extra>",

    text=sml.index,

    textposition="bottom right",

    showlegend=True,

    marker=dict(color='black', size=10),

    name = 'projections'

))

```

```
# add market
```

```
fig.add_trace(go.Scatter(  
  
    x=[1.0],  
  
    y=[index_return],  
  
    mode="markers+text",  
  
    hovertemplate="<b>beta:%{x:.0%}<br>ret:%{y:.1%}<br></b><extra></extra>",  
  
    text='market',  
  
    textposition="top left",  
  
    showlegend=True,  
  
    marker=dict(color='red', size=10),  
  
    name = 'market'  
  
))
```

```
# add risk-free asset
```

```
fig.add_trace(go.Scatter(  
  
    x=[0.0],  
  
    y=[risk_free_rate],  
  
    mode="markers+text",  
  
    hovertemplate="<b>beta:%{x:.0%}<br>ret:%{y:.1%}<br></b><extra></extra>",  
  
    text='risk-free',
```

```
textposition="top left",  
  
showlegend=True,  
  
marker=dict(color='green', size=10),  
  
name = 'risk-free'  
  
))
```

```
fig.update_layout(  
  
width=800,  
  
height=800,  
  
yaxis= { 'tickformat': '.0%'},  
  
title='Security Market Line (SML)',  
  
xaxis_title='$\\beta_i$',  
  
yaxis_title='$E(R_i)$',  
  
)
```

```
fig.show()
```

```
plot_sml(capm_results, index_exp_return, RISK_FREE_RATE)
```

## ANEXO 11. Optimización de Markowitz y Algoritmo Genético

```
def portfolio_performance(weights, mean_returns, cov_matrix):
```

```
    returns = np.sum(weights * mean_returns) * 252
```

```
    std = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights))) * np.sqrt(252)
```

```
    return returns, std, returns / std
```

```
def markowitz_optimization(daily_returns):
```

```
    num_assets = len(daily_returns.columns)
```

```
    mean_returns = daily_returns.mean()
```

```
    cov_matrix = daily_returns.cov()
```

```
def neg_sharpe_ratio(weights, mean_returns, cov_matrix):
```

```
    p_returns, p_std, p_sharpe = portfolio_performance(weights, mean_returns, cov_matrix)
```

```
    return -p_sharpe
```

```
def check_sum(weights):
```

```
    return np.sum(weights) - 1
```

```
constraints = {'type':'eq', 'fun': check_sum}
```

```
bounds = tuple((0, 1) for asset in range(num_assets))
```

```

    optimal_results = minimize(neg_sharpe_ratio, num_assets * [1. / num_assets,],
args=(mean_returns, cov_matrix,),

    method='SLSQP', bounds=bounds, constraints=constraints)

```

```

return optimal_results

```

```

optimal_markowitz = markowitz_optimization(daily_returns)

```

```

optimal_weights_markowitz = optimal_markowitz['x']

```

```

mean_returns = daily_returns.mean()

```

```

cov_matrix = daily_returns.cov()

```

```

portfolio_return_markowitz, portfolio_volatility_markowitz, sharpe_ratio_markowitz =
portfolio_performance(optimal_weights_markowitz, mean_returns, cov_matrix)

```

```

print("Optimización de Markowitz:")

```

```

print("Pesos óptimos de la cartera:", optimal_weights_markowitz)

```

```

print(f"Rendimiento esperado: {portfolio_return_markowitz:.2%}")

```

```

print(f"Volatilidad esperada: {portfolio_volatility_markowitz:.2%}")

```

```

print(f"Ratio de Sharpe: {sharpe_ratio_markowitz:.2f}")

```

```

class PortfolioOptimizerGA:

```

```

    def __init__(self, returns, pop_size=100, num_generations=500, mutation_rate=0.01):

```

```

self.returns = returns

self.pop_size = pop_size

self.num_generations = num_generations

self.mutation_rate = mutation_rate

self.num_assets = returns.shape[1]

self.population = self._initialize_population()

self.fitness_history = []

def _initialize_population(self):

    population = []

    for _ in range(self.pop_size):

        weights = np.random.dirichlet(np.ones(self.num_assets), size=1)[0]

        population.append(weights)

    return np.array(population)

def _fitness(self, weights):

    port_return = np.dot(weights, self.returns.mean()) * 252

    port_volatility = np.sqrt(np.dot(weights.T, np.dot(self.returns.cov() * 252, weights)))

    sharpe_ratio = port_return / port_volatility

    return sharpe_ratio

```

```

def _select_parents(self):

    fitness_scores = np.array([self._fitness(ind) for ind in self.population])

    return self.population[np.argsort(fitness_scores)][-self.pop_size//2:]

def _crossover(self, parent1, parent2):

    crossover_point = random.randint(0, self.num_assets - 1)

    child1 = np.concatenate((parent1[:crossover_point], parent2[crossover_point:]))

    child2 = np.concatenate((parent2[:crossover_point], parent1[crossover_point:]))

    return child1, child2

def _mutate(self, weights):

    for i in range(len(weights)):

        if random.random() < self.mutation_rate:

            weights[i] = np.random.random()

    weights = weights / np.sum(weights)

    return weights

def optimize(self):

    for generation in range(self.num_generations):

        parents = self._select_parents()

        new_population = []

```

```

while len(new_population) < self.pop_size:

    parent1, parent2 = random.choices(parents, k=2)

    child1, child2 = self._crossover(parent1, parent2)

    new_population.extend([self._mutate(child1), self._mutate(child2)])

self.population = np.array(new_population)

max_fitness = max([self._fitness(ind) for ind in self.population])

self.fitness_history.append(max_fitness)

best_weights = self.population[np.argmax([self._fitness(ind) for ind in self.population])]

return best_weights

optimizer = PortfolioOptimizerGA(daily_returns)

optimal_weights_ga = optimizer.optimize()

# Calcular el rendimiento y la volatilidad de la cartera óptima

portfolio_return_ga,          portfolio_volatility_ga,          sharpe_ratio_ga          =
portfolio_performance(optimal_weights_ga, mean_returns, cov_matrix)

print("\nOptimización con Algoritmos Genéticos:")

print("Pesos óptimos de la cartera:", optimal_weights_ga)

```

```
print(f'Rendimiento esperado: {portfolio_return_ga:.2%}')
```

```
print(f'Volatilidad esperada: {portfolio_volatility_ga:.2%}')
```

```
print(f'Ratio de Sharpe: {sharpe_ratio_ga:.2f}')
```

## ANEXO 12. Comparación de resultados y visualización final

```
strategies = ['Markowitz', 'Algoritmo Genético']  
  
returns = [portfolio_return_markowitz, portfolio_return_ga]  
  
volatilities = [portfolio_volatility_markowitz, portfolio_volatility_ga]  
  
sharpe_ratios = [sharpe_ratio_markowitz, sharpe_ratio_ga]
```

```
fig, ax = plt.subplots(1, 4, figsize=(24, 5))
```

```
# Gráfico de rendimientos esperados
```

```
sns.barplot(x=strategies, y=returns, ax=ax[0])
```

```
ax[0].set_title('Rendimientos Esperados')
```

```
ax[0].set_ylabel('Rendimiento')
```

```
# Gráfico de volatilidades esperadas
```

```
sns.barplot(x=strategies, y=volatilities, ax=ax[1])
```

```
ax[1].set_title('Volatilidades Esperadas')
```

```
ax[1].set_ylabel('Volatilidad')
```

```
# Gráfico de ratios de Sharpe
```

```
sns.barplot(x=strategies, y=sharpe_ratios, ax=ax[2])
```

```
ax[2].set_title('Ratios de Sharpe')
```

```
ax[2].set_ylabel('Ratio de Sharpe')

# Gráfico de la evolución del fitness en el algoritmo genético
ax[3].plot(optimizer.fitness_history, label='Fitness máximo')
ax[3].set_title('Evolución del Fitness en Algoritmo Genético')
ax[3].set_xlabel('Generaciones')
ax[3].set_ylabel('Fitness (Ratio de Sharpe)')
ax[3].legend()

plt.show()
```

## 7. BIBLIOGRAFÍA

- Algarín, C. A. R. (2010). Optimización por colonia de hormigas: Aplicaciones y tendencias. *Ingeniería Solidaria*, 6(10-11), Article 10-11.
- Barros-Duarte, C. (2006). Entre lo local y lo global: Proceso de regulación para la preservación de la salud en el trabajo. *Laboreal*, 2(1), Article 1. <https://doi.org/10.4000/laboreal.13766>
- Bies, R., Muldoon, M., Pollock, B., Manuck, S., Smith, G., & Sale, M. (2006). A Genetic Algorithm-Based, Hybrid Machine Learning Approach to Model Selection. *Journal of pharmacokinetics and pharmacodynamics*, 33, 195-221. <https://doi.org/10.1007/s10928-006-9004-6>
- Black, F., & Litterman, R. B. (1991). Asset Allocation: Combining Investor Views with Market Equilibrium. *The Journal of Fixed Income*, 1(2), 7-18. <https://doi.org/10.3905/jfi.1991.408013>
- Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.
- Carrasco, J. M. (2006). GÓMEZ-BEZARES, FERNANDO. Gestión de carteras. Eficiencia, teoría de cartera, CAPM, APT. *ICADE. Revista de la Facultad de Derecho*, (69), 376-379.
- Chambi Condori, P. P. (2019). El VaR para la gestión del riesgo del portafolio de inversiones. *Quipukamayoc*, 27(54), 9-18. <https://doi.org/10.15381/quipu.v27i54.16511>
- Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2), 243-278. <https://doi.org/10.1016/j.tcs.2005.05.020>
- Espí, M. V. (1995). Un nuevo algoritmo para la optimación de estructuras: El recocido simulado. *Informes de la Construcción*, 46(436), Article 436. <https://doi.org/10.3989/ic.1995.v46.i436.1084>
- Fama, E. F. (1965). The Behavior of Stock-Market Prices. *The Journal of Business*, 38(1), 34-105.
- Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, 25(2), 383-417. <https://doi.org/10.2307/2325486>
- Fama, E. F. (1991). Efficient Capital Markets: II. *The Journal of Finance*, 46(5), 1575-1617. <https://doi.org/10.1111/j.1540-6261.1991.tb04636.x>
- Fama, E. F., & French, K. R. (1989). Business conditions and expected returns on stocks and bonds. *Journal of Financial Economics*, 25(1), 23-49. [https://doi.org/10.1016/0304-405X\(89\)90095-0](https://doi.org/10.1016/0304-405X(89)90095-0)
- Galván Rodríguez, F. (2017). *Híperheurístico basado en búsqueda local iterativa para la solución de un problema de cartera de proyectos*. [Masters, Universidad Autónoma de Nuevo León]. <http://eprints.uanl.mx/16552/>

- Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12), 4831-4845. <https://doi.org/10.1016/j.cnsns.2012.05.010>
- Glover, F., & Laguna, M. (1997). General purpose heuristics for integer programming—Part I. *Journal of Heuristics*, 2(4), 343-358. <https://doi.org/10.1007/BF00132504>
- Gómez, M. Á. Z. (s. f.). *TRABAJO DE FIN DE GRADO*.
- Grossman, S. J., & Stiglitz, J. E. (1980). On the Impossibility of Informationally Efficient Markets. *The American Economic Review*, 70(3), 393-408.
- Hansen, P., Mladenovic, N., Mihajla, K., Perez, J. A. M., & de Estadística, D. (s. f.). *Bu'squeda de Entorno Variable*.
- Harrison, J. S., & John, C. H. S. (2002). *Fundamentos de la dirección estratégica*. Ediciones Paraninfo, S.A.
- He, G., & Litterman, R. (2002). The Intuition Behind Black-Litterman Model Portfolios. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.334304>
- Hernández Quevedo, M. (2020). *Hiperheurístico multiobjetivo basado en búsqueda de vecindad variable para problemas de selección de cartera de proyectos* (Doctoral dissertation, Universidad Autónoma de Nuevo León).
- Hsiao, Y.-T., Chuang, C.-L., Jiang, J.-A., & Chien, C.-C. (2005). A novel optimization algorithm: Space gravitational optimization. *2005 IEEE International Conference on Systems, Man and Cybernetics*, 3, 2323-2328 Vol. 3. <https://doi.org/10.1109/ICSMC.2005.1571495>
- Institute, C. F. A. (2017). *CFA Program Curriculum 2018 Level II*. John Wiley & Sons.
- Jensen (Deceased), & C, M. (1978). *Some Anomalous Evidence Regarding Market Efficiency* (SSRN Scholarly Paper 244159). <https://doi.org/10.2139/ssrn.244159>
- Jorion, P. (2010). Risk Management: Historical Perspectives. En *Encyclopedia of Quantitative Finance*. John Wiley & Sons, Ltd. <https://doi.org/10.1002/9780470061602.eqf15002>
- Kaveh, A., & Farhoudi, N. (2011). A unified approach to parameter selection in meta-heuristic algorithms for layout optimization. *Journal of Constructional Steel Research*, 67(10), 1453-1462. <https://doi.org/10.1016/j.jcsr.2011.03.019>
- Kirkpatrick, J. B. (1983). An iterative method for establishing priorities for the selection of nature reserves: An example from Tasmania. *Biological Conservation*, 25(2), 127-134. [https://doi.org/10.1016/0006-3207\(83\)90056-3](https://doi.org/10.1016/0006-3207(83)90056-3)
- Markowitz, H. (1952). Portfolio Selection\*. *The Journal of Finance*, 7(1), 77-91. <https://doi.org/10.1111/j.1540-6261.1952.tb01525.x>
- McNeil, A., Frey, R., & Embrechts, P. (2005). Quantitative Risk Management: Concepts, Techniques, and Tools. En *Journal of The American Statistical Association—J AMER STATIST ASSN* (Vol. 101).

- Melián Batista, M. B., Glover, F., Melián Batista, M. B., & Glover, F. (2003). Búsqueda Tabú. *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial*, 7(19), Article 19.
- Mena, D. C. T. (s. f.). *Simulación de Monte Carlo para la Gestión de Costos en la Evaluación de Proyectos de Inversión, caso: Construcción del Hotel Wyndham Gran Cándor PRONOBIS S.A. Ecuador (2014-2015)*.
- Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press.
- Ogbu, F., & Smith, D. (1991). Simulated annealing for the permutation flowshop problem. *Omega*, 19(1), 64-67. [https://doi.org/10.1016/0305-0483\(91\)90036-S](https://doi.org/10.1016/0305-0483(91)90036-S)
- Ramalhinho-Lourenço, H., Martin, O. C., & Stützle, T. (2000). Iterated local search.
- Reyes Rodríguez, M., & Moya Rodríguez, J. (2012, octubre 5). *Optimización de intercambiadores de calor mediante la técnica de recocido simulado*. <https://doi.org/10.13140/RG.2.1.3780.1440>
- Röhler, A. B., & Chen, S. (2011). An Analysis of Sub-swarms in Multi-swarm Systems. En D. Wang & M. Reynolds (Eds.), *AI 2011: Advances in Artificial Intelligence* (pp. 271-280). Springer. [https://doi.org/10.1007/978-3-642-25832-9\\_28](https://doi.org/10.1007/978-3-642-25832-9_28)
- Tang, R., Fong, S., Yang, X.-S., & Deb, S. (2012). Wolf search algorithm with ephemeral memory. *Seventh International Conference on Digital Information Management (ICDIM 2012)*, 165-172. <https://doi.org/10.1109/ICDIM.2012.6360147>
- Unhelkar, B., & Gonsalves, T. (2021). *Artificial intelligence for business optimization: research and applications*. CRC Press.
- Vanegas, S., Amaya, I., & Correa, R. (2013). Algoritmo del murciélago virtual en el desarrollo de la Integral de Duhamel para sistemas estructurales con un grado de libertad. *Revista ingeniería de construcción*, 28(3), 278-289. <https://doi.org/10.4067/S0718-50732013000300005>
- Walters, J. (2011). The Black-Litterman Model in Detail. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.1314585>
- Yang, X.-S. (2010). A New Metaheuristic Bat-Inspired Algorithm. En J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, & N. Krasnogor (Eds.), *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)* (pp. 65-74). Springer. [https://doi.org/10.1007/978-3-642-12538-6\\_6](https://doi.org/10.1007/978-3-642-12538-6_6)
- Yang, X.-S. (2011). Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation*, 3(5), 267-274. <https://doi.org/10.1504/IJBIC.2011.042259>
- Zapatero Moreno, M. J., Alegre Martínez, J. F., & Pacheco Bonrostro, J. A. (2010). "Optimización gravitatoria" y "Optimización por enjambre de partículas": Comportamiento en funciones no-lineales. *Anales de ASEPUMA*, 18, 36.

## DECLARACIÓN CHAT GPT

Por la presente, yo, Marta Carpi Pérez, estudiante de Doble Grado en Ade y Análisis de Negocios (E2 + Analytics) de la Universidad Pontificia Comillas al presentar mi Trabajo Fin de Grado titulado "Estado del Arte de la Gestión de Carteras", declaro que he utilizado la herramienta de Inteligencia Artificial Generativa ChatGPT u otras similares de IAG de código sólo en el contexto de las actividades descritas a continuación:

1. **Brainstorming de ideas de investigación:** Utilizado para idear y esbozar posibles áreas de investigación.
2. **Metodólogo:** Para descubrir métodos aplicables a problemas específicos de investigación.
3. **Interpretador de código:** Para realizar análisis de datos preliminares.
4. **Sintetizador y divulgador de libros complicados:** Para resumir y comprender literatura compleja.
5. **Revisor:** Para recibir sugerencias sobre cómo mejorar y perfeccionar el trabajo con diferentes niveles de exigencia.
6. **Traductor:** Para traducir textos de un lenguaje a otro.

Afirmo que toda la información y contenido presentados en este trabajo son producto de mi investigación y esfuerzo individual, excepto donde se ha indicado lo contrario y se han dado los créditos correspondientes (he incluido las referencias adecuadas en el TFG y he explicitado para que se ha usado ChatGPT u otras herramientas similares). Soy consciente de las implicaciones académicas y éticas de presentar un trabajo no original y acepto las consecuencias de cualquier violación a esta declaración.

Fecha: 29/11/2024

Firma: Marta Carpi Pérez