

UNA PROPUESTA EVOLUTIVA PARA CONSTRUIR UN PORTAFOLIO EFICIENTE

OPTIMIZACIÓN ESTOCÁSTICA

PROYECTO FINAL

KEVIN EMMANUEL SOTO HERNÁNDEZ

Centro de Investigación en Matemáticas (CIMAT)
kevin.soto@cimat.mx

2 de junio de 2019

Resumen

Dado un conjunto de retornos obtenidos del mercado bursátil, se desea construir un portafolio de inversiones asignando a cada activo un porcentaje de nuestro capital de acuerdo al problema de optimización clásico de media-varianza. No obstante, en algunas ocasiones resolver este problema de optimización puede llevar a soluciones de "esquina" privándolo de diversificación, y por lo tanto de rentabilidad [1]. Además, a veces no es preciso invertir en el conjunto completo de activos, ya que puede haber algunos de estos activos con una alta volatilidad y bajo retorno esperado generando un portafolio con baja utilidad. En este documento se aborda una propuesta que primero filtra los activos según su correlación, para después obtener un portafolio solución al problema de media-varianza con un algoritmo novedoso que usa gradiente conjugado, aproximación por diagonalización y la estrategia de programación "divide-and-conquer". Finalmente un algoritmo genético se encarga de seleccionar el mejor sub-conjunto de activos con la más alta utilidad.

keywords: portafolio eficiente, inversiones, algoritmo genético, frontera eficiente, gradiente conjugado, quasi-diagonal, bisección, diversificación, clustering.

I. INTRODUCCIÓN

LA teoría moderna de portafolios (MPT) es una teoría sobre cómo los inversionistas aversos al riesgo pueden construir portafolios de inversiones para optimizar o maximizar el rendimiento esperado en función de un nivel dado de riesgo de mercado, enfatizando que el riesgo es una parte inherente de una mayor recompensa.

La teoría sostiene que las características de riesgo y rendimiento de una inversión no deben verse por sí solas, sino que deben evaluarse por la forma en que la inversión afecta el riesgo y el rendimiento de la cartera en general.

De acuerdo con la teoría, es posible construir una "frontera eficiente" de portafolios óptimos que ofrezcan el máximo retorno esperado posi-

ble para un nivel de riesgo dado. Esta teoría fue iniciada por Harry Markowitz en su artículo "Portfolio Selection", publicado en 1952 por el Journal of Finance. Más tarde se le otorgó un premio Nobel por el desarrollo del MPT.

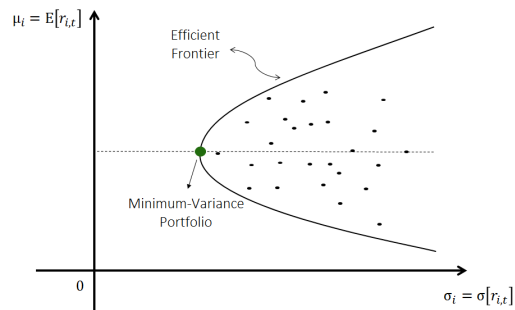


Figura 1: Frontera eficiente de selección de portafolios

En la figura 1, se puede apreciar el espacio en donde viven todas las selecciones de portafolios. Dada así, el objetivo de este documento es encontrar la frontera eficiente o frente de Pareto con el algoritmo descrito en la sección *Método*.

Teoría Moderna de Portafolios

Para las inversiones en la bolsa de valores podemos ver esto analíticamente: Supongamos que hay n activos que formarán nuestro portafolio. Sea X_i la variable aleatoria que modela el **retorno** del i -ésimo activo. Si contamos con un capital de inversión (no importa cual), se desea asignar a cada activo un porcentaje w_i para la construcción de nuestro portafolio. Así, es evidente que el retorno R de nuestro portafolio será:

$$R = w_1X_1 + w_2X_2 + \cdots + w_nX_n$$

La ley de grandes números asegurará que el rendimiento real del portafolio será casi el mismo que el rendimiento esperado.

$$E(R) = E(w_1X_1 + w_2X_2 + \cdots + w_nX_n)$$

$$E(R) = w_1E(X_1) + w_2E(X_2) + \cdots + w_nE(X_n)$$

$$E(R) = w^T \mu$$

Así mismo, definimos el **riesgo** del portafolio como la varianza del retorno. Esto tiene sentido, ya que a mayor varianza, hay más probabilidades de que nuestros activos se deprecien en la bolsa de valores.

$$V(R) = V(w_1X_1 + w_2X_2 + \cdots + w_nX_n)$$

$$V(R) = \sum_{i=1}^n w_i^2 V(X_i) + 2 \sum_{i=1}^n \sum_{j=1, j < i}^n w_i w_j \text{Cov}(X_i, X_j)$$

$$V(R) = w^T \Sigma w$$

Esta presunción que la ley de grandes números se aplica al modelado del comportamiento de acciones, no puede ser aceptada completamente, ya que predecir el mercado bursátil es una acción que nadie a podido lograr. Por lo tanto, hay que remarcar que este modelo es solo una primera aproximación.

Diversificación: un portafolio eficiente

En finanzas, la diversificación es el proceso de asignación de capital de una manera que reduce la exposición a cualquier activo o riesgo en particular. Un camino común hacia la diversificación es reducir el riesgo o la volatilidad mediante la inversión en una variedad de activos. Si los precios de los activos no cambian en perfecta sincronía, una cartera diversificada tendrá menos varianza que la varianza promedio ponderada de sus activos constituyentes y, a menudo, menos volatilidad que la menos volátil de sus constituyentes. El ejemplo más simple de la diversificación es el proverbio "No pongas todos tus huevos en una canasta". Dejar caer la cesta romperá todos los huevos. La colocación de cada huevo en una canasta diferente es más diversificada. Hay más riesgo de perder un huevo, pero menos riesgo de perderlos todos. Por otro lado, tener muchas canastas puede aumentar los costos. En finanzas, un ejemplo de una cartera no diversificada es tener una sola acción. Esto es arriesgado; no es inusual que una sola acción baje un 50 % en un año. Es menos común que una cartera de 20 acciones baje tanto, especialmente si se seleccionan al azar. Si las acciones se seleccionan de una variedad de industrias, tamaños de empresas y tipos de activos, es incluso menos probable que experimenten una caída grande. Por lo tanto, en este documento se va a considerar una base de datos grande con muchos activos.

¿Qué hacer con tantos activos?

Cada activo tiene un riesgo y retorno individual; por lo tanto, una utilidad. En realidad cada combinación de activos tiene su propia utilidad. Suponga que ahora que se proporciona un conjunto n activos; el número de todas las combinaciones posibles de activos viene dado por:

$$\sum_{k=1}^n \binom{n}{k} = 2^n - 1$$

Está claro que una de esas combinaciones será el óptimo global. Donde el óptimo global podría considerarse como el subconjunto de utilidades que, después de ser procesadas por nuestro

algoritmo y obtener los pesos respectivos, ofrecerán la mejor utilidad (Véase Optimización del Portafolio). Pero hay un problema: si n es muy grande, las combinaciones de todos los valores crece exponencialmente. Para $n = 50$, por ejemplo, es un número enorme, y sigue aumentando cada vez que se agrega un activo al conjunto. Por tal motivo, proponemos la aplicación de una metaheurística que solventará dicho pericance con la complejidad computacional: **Un Algoritmo Genético**, cuya implementación viene dada en la siguiente sección.

II. MÉTODO

Nuestro método costa de 4 fases: clusterización de los activos, resolver el problema de optimización clásico media-varianza, bisectar los activos y aplicar “divide-and-conquer” y finalmente seleccionar los mejores subconjuntos de activos

Clusterización por Correlación

En esta 1^{ra} etapa utilizamos un algoritmo de clusterización que reorganiza las filas y columnas de la matriz de covarianza de modo que los activos con mayor correlación se encuentren más cercanos entre sí. Esta reordenación de la matriz de covarianza hace que resalte una propiedad muy útil: las inversiones similares se colocan juntas, y distintas inversiones se ubican muy alejadas de tal manera que si deseamos invertir en un subconjunto de estos, podemos tomar los activos más separados entre sí para evitar perder capital cada vez que alguno de ellos tenga una caída en la bolsa de valores.

El algoritmo funciona de la siguiente manera:

- 1 Tomamos la matriz de correlación y aplicamos un mapeo $\phi(\rho_{ij}) \rightarrow [0, 1]$ a cada uno de los elementos. La función “acerca” las correlaciones altas en el espacio euclidiano y cercanas a uno en el lado derecho del intervalo $[0, 1]$, mientras que a las negativas, también las acerca pero del lado izquierdo del intervalo. La función está dada como:

$$\phi(\rho_{ij}) = \sqrt{\frac{1}{2}(1 - \rho_{ij})}$$

- 2 Todos los puntos mapeados empiezan formando un cluster cada uno.
- 3 Se calcula la distancia euclidiana por pares y se agrupan en un cluster aquellos puntos c_i, c_j que tienen una distancia mínima.

$$\min_{i,j} (d(c_1, c_2), d(c_1, c_3), \dots, d(c_i, c_j), \dots)$$

$$d(c_i, c_j) = \|c_i - c_j\|^2$$

- 4 El proceso se repite hasta que se tiene un solo cluster.
- 5 El algoritmo de clusterización da como salida la lista de los índices L reordenada de acuerdo a como se fueron agrupando en el proceso.
- 6 Con esta lista se reordena la matriz de covarianza para los siguientes procesos de la construcción del portafolio.

A continuación se muestra un ejemplo gráfico de como funciona el algoritmo de clusterización.

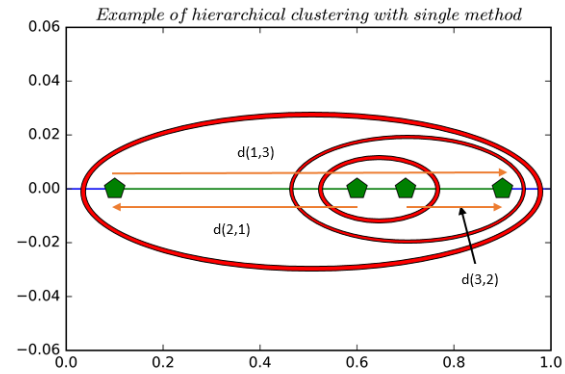


Figura 2: Ejemplo gráfico del proceso de clusterización

Optimización del Portafolio

Este algoritmo es una versión moderna de lo que Markowitz propuso en 1952, ambos se centran en la idea de que el retorno esperado es deseable y la varianza del retorno no es deseable. Sin embargo, el tratar con el problema de maximizar el retorno esperado a la vez

que minimizamos la varianza puede ser algo complicado si los tratamos de manera separada. Por eso se propone la función de utilidad $U(w) = E(R) - \frac{\rho}{2} \text{Var}(R)$ que lidia con ambos problemas a la vez. A este enfoque se le llama “el optimizador clásico de Markowitz” y se representa con el siguiente problema de optimización:

$$\begin{aligned} & \underset{w \in \mathbb{R}^n}{\text{maximize}} && U(w) = w^T \mu - \frac{\rho}{2} w^T \Sigma w \\ & \text{subject to} && 1_n^T w = 1 \\ & && w \succeq 0. \end{aligned}$$

donde ρ es un factor de penalización llamado “aversión al riesgo”. Hay muchas maneras de resolver el problema anterior, una de las más confiables y rápidas es aplicar el algoritmo de **gradiente conjugado** dado que la matriz de covarianza es simétrica y semi-definida positiva. Otra opción atractiva es la de reemplazar la matriz de covarianza con su matriz diagonal para generar soluciones aproximadas. De hecho, este enfoque es corazón de nuestro algoritmo novedoso, y procede de la siguiente manera:

Cambiamos la función cuadrática $U(w)$ por su versión lagrangiana usando solamente la primera restricción. Al aplicar las condiciones de KKT encontramos el punto de inflexión, que de hecho es un máximo, ya que el hessiano es semi-definido negativo.

$$\mathcal{L}(w, \lambda) = w^T \mu - \frac{\rho}{2} w^T \Sigma w - \lambda (1_n^T w - 1)$$

$$\nabla_w \mathcal{L}(w, \lambda) = \mu - \rho \Sigma w - \lambda 1_n = 0 \quad (1)$$

$$\nabla_\lambda \mathcal{L}(w, \lambda) = 1_n^T w - 1 = 0 \quad (2)$$

de (1) se despeja w y se puede sustituir en (2) para obtener λ :

$$w = \frac{1}{\rho} \Sigma^{-1} (\lambda 1_n + \mu)$$

$$\frac{1}{\rho} 1_n^T \Sigma^{-1} (\lambda 1_n + \mu) = 1$$

$$\lambda = \frac{\rho - m}{v}$$

$$\text{donde } m = 1_n^T \Sigma^{-1} \mu \text{ y } v = 1_n^T \Sigma^{-1} 1_n$$

Dado que se desea evitar la inversión de la matriz de covarianza debido a su inestabilidad, elegimos diagonalizar la matriz de covarianza (es decir, eliminar factores de covarianza y dejar solo las variaciones en la diagonal principal) para invertir la matriz de covarianza de forma trivial. Finalmente el vector de pesos w se da como:

$$w = \frac{1}{\rho} \text{diag}(\Sigma)^{-1} \left[\left(\frac{\rho - m}{v} \right) 1_n + \mu \right] \quad (3)$$

de esa manera $v = \text{tr}(\text{diag}(\Sigma)^{-1})$ y $m = \text{tr}(\text{diag}(\Sigma)^{-1} \mu)$.

En el caso de tener solo dos activos ($n = 2$) el vector de pesos se puede escribir como $w = [w_1, w_2]$ y de acuerdo a (3) la solución aproximada al problema de optimización viene dado como:

$$w_1 = \frac{\sigma_2^2 \rho + r_1 - r_2}{\rho(\sigma_1^2 + \sigma_2^2)}$$

$$w_2 = \frac{\sigma_1^2 \rho + r_2 - r_1}{\rho(\sigma_1^2 + \sigma_2^2)}$$

$$\text{aquí } \mu = [r_1, r_2] \text{ y } \text{diag}(\Sigma) = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

Observe que $w_1 + w_2 = 1$ y la primera restricción se mantiene. Sin embargo, los retornos tanto r_1 como r_2 pueden ser cantidades negativas o positivas, incluso mayores que su varianza respectiva σ_i lo que puede llevar a generar pesos negativos. Esto no mantiene la segunda restricción ($w_i \geq 0$), por lo que para garantizar pesos no negativos, r_1 y r_2 se deben eliminar de la ecuación anterior. De esa manera se generan los siguientes pesos que mantienen las dos condiciones:

$$w_1 = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \quad (4)$$

$$w_2 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \quad (4)$$

Bisección Recursiva

La 3^{ra} etapa de este algoritmo lidia con el mayor problema de la construcción del portafolio: la diversificación del portafolio, dividiendo

recursivamente la lista de índices ordenados L (obtenidos en la 1^{ra} etapa de clusterización) en dos listas de igual longitud $L_1^{(t)}$ y $L_2^{(t)}$ para obtener una sub-varianza y un sub-retorno esperado dados de la siguiente manera:

$$\sigma_{i,t}^2 = \text{Var} \left(L_i^{(t)} \right) = \tilde{w}_i^{(t)T} \Sigma_{i,t} \tilde{w}_i^{(t)} \quad (5)$$

$$r_{i,t} = E \left(L_i^{(t)} \right) = \tilde{w}_i^{(t)T} \mu_{i,t} \quad (5)$$

donde $\Sigma_{i,t}$ y $\mu_{i,t}$ son la sub-matriz de covarianza y sub-vector de retorno esperado respectivamente, que únicamente contiene valores concernientes a la lista $L_i^{(t)}$ en la biseción t^{th} . Mientras que los vectores de pesos $\tilde{w}^{(t)} = [\tilde{w}_1^{(t)}, \tilde{w}_2^{(t)}]$ son obtenidos de aplicar **gradiente conjugado** al problema de optimización en la primera partición y la aproximación **quasi-diagonal** en el resto de las particiones.

La razón detrás de este enfoque es convertir el problema de n activos en solo dos activos, y tras aplicar la solución (4) podemos obtener pesos no negativos que pueden ser asignados los activos de cada lista al multiplicar recursivamente por los pesos en cada partición siguiendo la famosa estrategia de programación “divide-and-conquer”. Una forma de explicar cómo funciona esta etapa, es pensar que los activos de la sub-lista izquierda ($L_1^{(t)}$) y derecha ($L_2^{(t)}$) representan solo un activo cada uno. Ahora, se crea un vector unitario w para multiplicar los pesos $w_1^{(t)}$ y $w_2^{(t)}$ obtenidos en (5) para cada una de las posiciones representadas por la lista $L_i^{(t)}$. El lector puede ver este procedimiento de la siguiente manera:

Vector unitario

$$w = [1, 1, 1, \dots, 1, 1, 1]$$

1^{ra} partición ($t = 1$)

$$w = [w_1^{(1)}, w_1^{(1)}, \dots, w_1^{(1)}, [w_2^{(1)}, w_2^{(1)}, \dots, w_2^{(1)}]]$$

2^{da} partición ($t = 2$)

$$w = [w_1^{(1)} w_1^{(2)}, \dots, w_1^{(1)} w_1^{(2)}, [\dots], [\dots], [\dots]]$$

3^{ra} partición ($t = 3$)

$$w = [[w_1^{(1)} w_1^{(2)} w_1^{(3)}, \dots], [\dots], \dots, [\dots], [\dots]]$$

k^{ma} partición ($t = k$)

$$w = [[w_1^{(1)} \dots w_1^{(k)}, [\dots], \dots, [\dots], [\dots], [\dots]]]$$

El algoritmo se detiene hasta que se tiene un solo activo en cada una de las listas $L_i^{(t)}$. De esta manera se asegura que todos los activos tengan al menos un peso diferente de cero, aumentando la diversificación del portafolio. Observe que $\|w\| = 1$ (puede hacer las cuentas por sí mismo).

Algoritmo Genético

Algunas veces la base de datos contiene demasiados activos para invertir nuestro capital (digamos n activos en la base de datos). Incluso, en la mayoría de las ocasiones, invertir en todos los activos no conduce a un portafolio eficiente a pesar de contar con el mejor optimizador. En consecuencia se desea encontrar aquel sub-conjunto $L_\delta \subset L$ con m ($\leq N$) activos que maximice el portafolio encontrado por el algoritmo anterior. Así, el problema de optimización final es:

$$\begin{aligned} \max_{L_\delta \subset L} \max_{w \in \mathbb{R}^m} \quad & U(w) = w^T \mu - \frac{\rho}{2} w^T \Sigma w \\ \text{subject to} \quad & 1_n^T w = 1 \\ & w \succeq 0. \\ & |L_\delta| \leq |L| \quad (m \leq n) \end{aligned}$$

Dado que el problema se volvió combinatorio, utilizaremos un algoritmo genético cuyo funcionamiento se resume a continuación:

1. La selección de un sub-conjunto de activos se presenta como un arreglo de bits, donde si el i -ésimo elemento es marcado como 1, está dentro de nuestro portafolio y si es marcado con 0, no lo está.

$$ind_i = [1, 0, 0, 1, 1, 0, 1, \dots, 1]$$

2. El AG comienza creando una población inicial aleatoria.

3. El AG luego crea una secuencia de nuevas poblaciones a través de un proceso de selección, cruza y mutación estándares. En cada paso, el algoritmo utiliza a los individuos de la generación actual para crear la siguiente población. Para crear la nueva población, el algoritmo realiza los siguientes pasos:

- 1 Evalúa la función de máxima utilidad para cada individuo, es decir, resuelve el problema de optimización descrito en las secciones pasadas.
- 2 Reordena los individuos por su función de aptitud
- 3 Selecciona la mejor mitad de la población

- 4 **Selección:** Esta muestra de individuos, llamados padres, serán los candidatos cuyos genes serán heredados a la siguiente generación. Toma dos individuos de forma aleatoria y los combina en un proceso de cruza generando dos hijos y agregándolos a la nueva generación. Este proceso se repite hasta que la nueva población tenga un tamaño T deseado. Si el elitismo está encendido, el mejor individuo se reproduce con otro y es preservado en la siguiente generación.

- 5 **Cruza:** Se toman los dos individuos ind_i y ind_j y se toman dos índices entre 2 y $M - 1$ para cortar la cadena de genes en dos puntos e intercambiar el segmento de genes para crear dos hijos nuevos.

$$ind_i = [1, 0, 0, 1, 1 | 0, 1, 0 | 1, 0, 0, 0, 1]$$

$$swap \updownarrow \updownarrow \updownarrow$$

$$ind_j = [0, 1, 1, 1, 1 | 1, 0, 1 | 0, 0, 1, 0, 1]$$

Este proceso es repetido con hasta completar la nueva población de tamaño T individuos.

- 6 **Mutación:** De la cantidad total de hijos, se seleccionan aleatoriamente un porcentaje cr que no serán modificados. El resto de los nuevos individuos será sometido a un proceso de mutación, en donde cada uno de

los bits será cambiado a su complemento con una probabilidad mr .

- 7 El algoritmo se detiene cuando se cumple uno de los criterios de detención.

III. RESULTADOS

Estas pruebas se realizaron sobre el historial financiero de $n = 54$ activos reales ponderados en el mercado bursátil. La base de datos incluye firmas conocidas como Google, Amazon, Facebook, Intel, etc. Al obtener la valoración de las acciones, se calculó el retorno en base al precio de cierre. El historial de mercado incluye todos los precios de las acciones en un periodo de tiempo de un año exactamente (251 observaciones).

Aversión al Riesgo

Se encontró que cuando el valor de la aversión al riesgo es alto, nuestro algoritmo obtenía valores de utilidad bastante pobres. Incluso, el mejor de las evaluaciones en el algoritmo genético obtenía retornos esperados negativos, lo cual significa que perderíamos dinero si apostáramos al los portafolios construidos. De hecho, cuando imprimíamos el histograma de pesos para los activos, resultaba que la asignación solo había sido en unos pocos (Véase figura 3). Pero cuando permitíamos mas riesgo en nuestro portafolio, los resultados mejoraban considerablemente y la diversificación era la esperada (Véase figura 4).

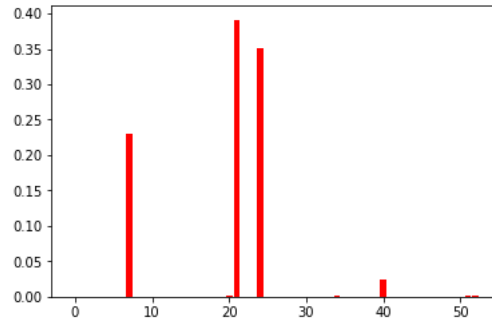


Figura 3: Histograma de pesos asignados a cada activo. Alta aversión al riesgo

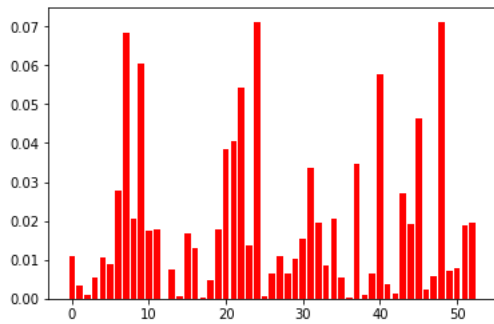


Figura 4: Histograma de pesos asignados a cada activo.
Baja aversión al riesgo

Algoritmo Genético

Se experimentó con diferente número de puntos de cruce y el que obtuvo mejores resultados fue la cruce de dos puntos. La probabilidad de mutación que mejores resultados obtuvo fue de $mt = 1/n$ para cada variable. La probabilidad de cruce se fijó en el valor de $cr = 0,8$. El tamaño de población fue de 80 individuos. Se experimentaron con 5 valores de máximo número de generaciones (5, 10 y 15) para que pudiera observar el comportamiento de los individuos al encontrar el frente de Pareto. La evolución del GA se presenta a continuación:

| GA (cruza: dos puntos) | | | | |
|------------------------|--------|--------|--------|--------|
| Gen. | sharpe | $U(w)$ | $E(R)$ | $V(R)$ |
| 1 | 0.1038 | 0.1387 | 0.0968 | 0.8897 |
| 2 | 0.1548 | 0.1524 | 0.1127 | 0.9012 |
| 3 | 0.1548 | 0.1524 | 0.1222 | 0.9712 |
| 4 | 0.1548 | 0.1863 | 0.1260 | 0.9966 |
| 5 | 0.1548 | 0.1863 | 0.1563 | 1.1983 |
| 6 | 0.1548 | 0.1863 | 0.1651 | 1.2042 |
| 7 | 0.1552 | 0.1863 | 0.1718 | 1.2078 |
| 9 | 0.1670 | 0.1863 | 0.1738 | 1.2578 |
| 10 | 0.1670 | 0.1911 | 0.1815 | 1.3024 |
| 11 | 0.1670 | 0.1911 | 0.1984 | 1.3024 |
| 12 | 0.1670 | 0.2170 | 0.2024 | 1.3134 |
| 13 | 0.1721 | 0.2212 | 0.2333 | 1.3265 |
| 14 | 0.1897 | 0.2212 | 0.2742 | 1.3342 |
| 15 | 0.2045 | 0.2312 | 0.2742 | 1.3434 |

| GA (cruza: uniforme) | | | | |
|----------------------|--------|--------|--------|--------|
| Gen. | sharpe | $U(w)$ | $E(R)$ | $V(R)$ |
| 1 | 0.0983 | 0.1087 | 0.1150 | 0.9052 |
| 2 | 0.1056 | 0.1124 | 0.1624 | 1.273 |
| 3 | 0.1056 | 0.1124 | 0.1624 | 1.273 |
| 4 | 0.1122 | 0.1263 | 0.1721 | 1.1944 |
| 5 | 0.1122 | 0.1263 | 0.1721 | 1.1944 |
| 6 | 0.1167 | 0.1411 | 0.1996 | 1.3832 |
| 7 | 0.1181 | 0.1590 | 0.1996 | 1.2635 |
| 8 | 0.1324 | 0.1617 | 0.2096 | 1.2665 |
| 9 | 0.1392 | 0.1670 | 0.2165 | 1.3833 |
| 10 | 0.1458 | 0.1670 | 0.1973 | 1.3471 |
| 11 | 0.1698 | 0.1943 | 0.1996 | 1.2635 |
| 12 | 0.1698 | 0.1943 | 0.1996 | 1.2635 |
| 13 | 0.1731 | 0.2096 | 0.2125 | 1.2181 |
| 14 | 0.1795 | 0.2112 | 0.2350 | 1.3236 |
| 15 | 0.1888 | 0.2212 | 0.2318 | 1.2911 |

| GA (cruza: un punto) | | | | |
|----------------------|--------|--------|--------|--------|
| Gen. | sharpe | $U(w)$ | $E(R)$ | $V(R)$ |
| 1 | 0.0917 | 0.1087 | 0.1481 | 1.4430 |
| 2 | 0.1117 | 0.1124 | 0.1499 | 1.3223 |
| 3 | 0.1156 | 0.1224 | 0.1511 | 1.001 |
| 4 | 0.1256 | 0.1690 | 0.1296 | 0.9746 |
| 5 | 0.1256 | 0.1712 | 0.1364 | 0.8903 |
| 6 | 0.1256 | 0.1863 | 0.1578 | 0.8801 |
| 7 | 0.1612 | 0.1943 | 0.1612 | 0.8790 |
| 8 | 0.1612 | 0.1943 | 0.1612 | 0.8790 |
| 9 | 0.1612 | 0.1943 | 0.1612 | 0.8790 |
| 10 | 0.1612 | 0.1943 | 0.1612 | 0.8790 |
| 11 | 0.1612 | 0.1943 | 0.1612 | 0.8790 |
| 12 | 0.1612 | 0.1943 | 0.1612 | 0.8790 |
| 13 | 0.1612 | 0.1943 | 0.1612 | 0.8790 |
| 14 | 0.1612 | 0.1943 | 0.1612 | 0.8790 |
| 15 | 0.1612 | 0.1943 | 0.1612 | 0.8790 |

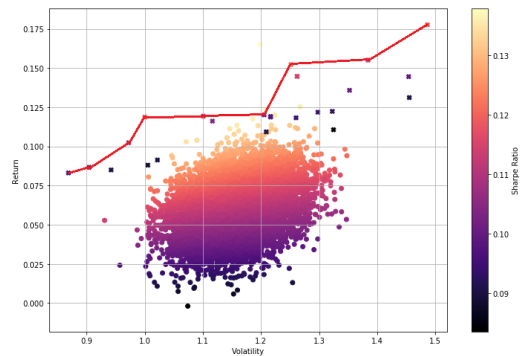


Figura 5: frente de Pareto para 5 generaciones

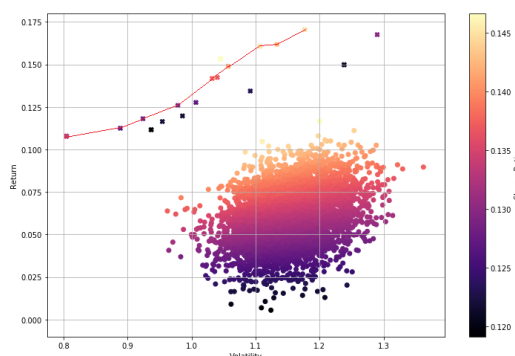


Figura 6: frente de pareto para 10 generaciones

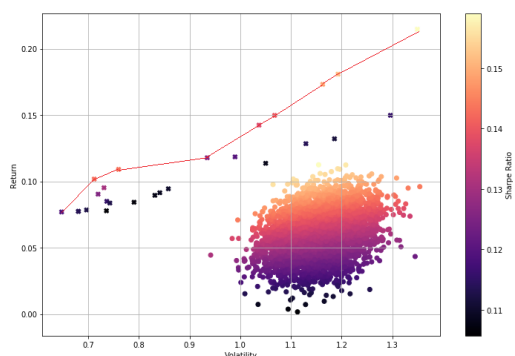


Figura 7: frente de pareto para 15 generaciones

IV. DISCUSIÓN

Este comportamiento tan peculiar al modificar el parámetro del aversión al riesgo da bastante de que hablar. Al principio que creamos nuestro modelo de optimización con la base de datos; no tomábamos en cuenta el factor de aversión al riesgo (es decir, lo fijábamos en 1) y los resultados arrojados nos molestaban debido a que los portafolios construidos tenían un retorno esperado negativo prediciendo, posiblemente, una pérdida de nuestro capital a lo largo del tiempo. Otro motivo era la baja diversificación de activos seleccionados como

lo puede apreciar en la figura 3. Sin embargo, cuando modificamos la aversión a un valor más bajo, los resultados mejoraban considerablemente. Esto fue algo revelador, ya que nos parecía que el invertir dinero en acciones de la bolsa de valores obedecía al famoso refrán: “El que no arriesga no gana”. Con el mismo motivo, fuimos disminuyendo la aversión y notábamos que los resultados no cambiaban en lo absoluto; también pasaba lo mismo cuando lo aumentábamos a valores exageradamente grandes. Es decir, nuestro modelo convergía a un portafolio estacionario de perfil de inversión de alto riesgo cuando $\rho \rightarrow 0$, y a un perfil de inversión seguro cuando $\rho \rightarrow \infty$. No sabemos explicar este comportamiento a ciencia cierta, pero remarcamos que sería interesante demostrarlo analíticamente. En cuanto al GA, no hay mucho que decir. Los resultados obtenidos son asombrosos a pesar de las pocas generaciones permitidas. Los valores de utilidad, retorno esperado y radio de sharpe incrementan en cada generación, pero, como era de esperarse, el riesgo aumenta. Esto último no es más que lo que explicábamos al principio. De los tres métodos de cruce, el que obtuvo mejores resultados fue la cruce de dos puntos, aunque perdía diversidad muy rápidamente. La cruce uniforme obtenía resultados casi igual de buenos, pero conservaba la diversidad. La cruce de un punto obtuvo un sub-rendimiento y alcanzó un óptimo local en muy pocas iteraciones; por lo que no recomendamos su uso en este problema. Nuestro objetivo de encontrar el frente de pareto fue cumplido y para mostrar la distancia entre la creación de portafolio de la manera más ingenua (portafolios con pesos aleatorios) y nuestro modelo, se imprimieron en la gráfica una nube de puntos circulares que representan portafolios generados de forma aleatoria con pesos con distribución uniforme. Note la convergencia hacia el frente de pareto en las figuras 5, 6 y 7.

REFERENCIAS

- [1] Marcos Lopez de Prado. Advances in Financial Machine Learning. Wiley Editorial,

2016.

- [2] Mohammed J. Zaki & Wagner Meira Data Mining and Analysis: Fundamental Concepts and Algorithms 2014, Cambridge University Press.
- [3] Harry Markowitz. The Journal of Finance. Portfolio Selection, 1952.