

# **Otimização de Portfólios de Ações da B3 com Métodos de Otimização Não Linear**

**Nº USP 15111760 Gabriel Coutinho Chaves**

**Nº USP 12676615 João Pedro Pereira Balieiro**

**Nº USP 16300449 José Fábio Viana de Brito**

**2 de dezembro de 2025**

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Objetivos</b>	<b>1</b>
<b>3</b>	<b>Descrição do problema escolhido</b>	<b>1</b>
3.1	Contexto teórico . . . . .	1
3.2	Obtenção dos Dados . . . . .	3
3.3	Formulação matemática . . . . .	4
<b>4</b>	<b>Métodos de resolução</b>	<b>5</b>
4.1	Método 1: Gradiente Ascendente Projetado . . . . .	6
4.2	Método 2: Barreiras Logarítmicas com Penalidade Quadrática . . . . .	7
4.3	Método 3: Gradiente Espectral Projetado . . . . .	8
<b>5</b>	<b>Implementação</b>	<b>9</b>
<b>6</b>	<b>Resultados Numéricos</b>	<b>10</b>
<b>7</b>	<b>Conclusão</b>	<b>12</b>

# 1 Introdução

O problema de alocação de recursos em carteiras de investimento é um exemplo clássico de otimização não linear em finanças. Dado um conjunto de ativos de risco, o investidor deseja escolher pesos para cada ativo de modo a balancear retorno esperado e risco (variância ou volatilidade da carteira), respeitando limitações práticas como a necessidade de investir 100% do capital disponível e a proibição de vendas a descoberto (em inglês, *short-selling*), isto é, sem vender ativos que não possui.

Neste trabalho foi desenvolvido um painel interativo para montagem e otimização de portfólios de ações listadas na B3, a partir de dados históricos de preços. O sistema foi implementado em Python utilizando `Streamlit` e a biblioteca `yfinance` para obtenção dos dados de mercado. O conjunto de ativos disponíveis é extraído automaticamente do índice IBXL da B3 por meio de `selenium` e `pandas`, de modo que o usuário possa selecionar, de forma prática, quais ações deseja combinar na carteira.

Do ponto de vista de otimização, o problema é modelado como um problema de maximização do Índice de Sharpe da carteira sob restrições lineares (soma dos pesos igual a 1 e retorno-alvo pré-fixado) e restrições de caixa ( $w_i \geq 0$ ).

## 2 Objetivos

Os objetivos principais do trabalho são:

- (i) Formular um problema real de otimização não linear contínua a partir do contexto de montagem de carteiras de ações da B3, explicitando variáveis, parâmetros, função objetivo e restrições.
- (ii) Implementar, em Python, diferentes métodos vistos em aula de resolução para o problema de otimização.
- (iii) Integrar os métodos de otimização em um painel interativo que permita ao usuário selecionar ativos, definir um retorno-alvo e obter a carteira de mínima variância correspondente, bem como visualizar métricas e gráficos relevantes.

## 3 Descrição do problema escolhido

### 3.1 Contexto teórico

O problema de alocação de recursos em carteiras de investimento consiste em combinar diferentes ativos financeiros de modo a equilibrar retorno e risco sob condições de incerteza. O retorno de um ativo é entendido como o ganho médio esperado ao longo do tempo, enquanto o risco corresponde à variabilidade desses retornos em torno da média, usualmente mensurada pelo desvio-padrão.

A principal ferramenta prática para reduzir risco é a diversificação. Ao combinar ativos cujos retornos não são perfeitamente correlacionados, o risco total da carteira pode ser reduzido em relação ao risco individual de cada ativo [1]. O objetivo, portanto, é construir carteiras eficientes, que ofereçam o maior retorno possível para um dado nível de risco ou, inversamente, o menor risco para um retorno alvo.

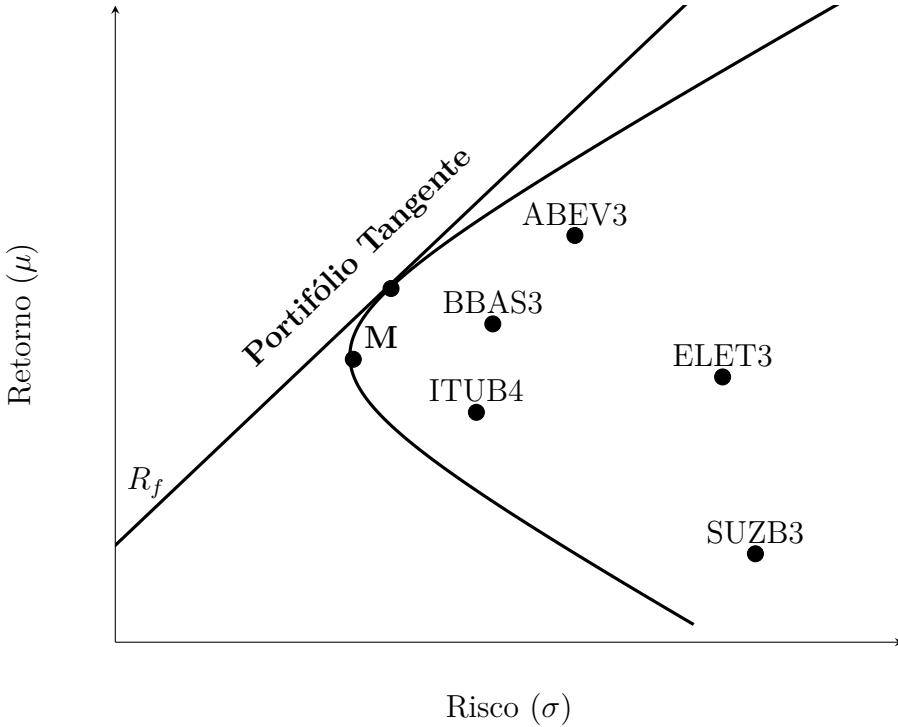


Figura 1: Gráfico de portfólios em relação à relação risco/retorno, mostrando a fronteira eficiente de Markowitz e a carteira tangente com exemplo de ativos brasileiros. O valor  $R_f$  representa o retorno da taxa livre de risco. O portfólio  $M$  é o de menor variância. O Ponto de Tangência é considerado o portifólio ótimo, cujo o Índice de Sharpe é mais elevado,

Na teoria de Markowitz [5], a fronteira eficiente representa o conjunto de portfólios que oferecem o maior retorno esperado para cada nível de risco, ou o menor risco possível para cada nível de retorno. Ela resulta da combinação ótima de ativos, levando em conta suas correlações. O desenho da curva formada por esse conjunto está exemplificado na 1. Portfólios abaixo dessa curva são considerados ineficientes, pois sempre existe uma alternativa que proporciona mais retorno para o mesmo risco ou menos risco para o mesmo retorno.

Na Figura 1, o portfólio  $M$  corresponde ao ponto de menor variância e domina todos os demais posicionados abaixo da fronteira, por apresentar menor risco para um retorno esperado superior. Esse ponto, também chamado de **carteira de risco mínimo**, marca o limite inferior da hipérbole de Markowitz e indica a composição de ativos que minimiza a volatilidade.

Na fronteira eficiente é possível identificar uma carteira de ativos que maximiza o ganho em excesso sobre a taxa livre de risco, ou seja, o melhor prêmio pelo risco. Essa carteira única domina todas as demais alternativas possíveis e permite que os investidores maximizem seus retornos ajustados ao risco[4]. O ponto tangente ilustrado na Figura 1 corresponde ao portfólio específico da fronteira eficiente que, em presença de um ativo livre de risco, gera a linha de alocação de capital (Capital Allocation Line – CAL). Essa reta conecta a taxa livre de risco ao portfólio ótimo de ativos arriscados e representa todas as combinações possíveis entre ambos.

Qualquer ponto sobre a CAL constitui uma alternativa viável de investimento, ajustada ao perfil do investidor. Sua inclinação corresponde ao índice de Sharpe máximo, indicador que será utilizado neste trabalho para orientar a escolha da alocação eficiente.

## 3.2 Obtenção dos Dados

A construção automatizada de uma carteira de investimentos no projeto foi feita a partir de ações listadas na B3 e pertencentes ao índice IBXL. O universo de ativos é atualizado a cada execução por meio de raspagem da página pública da bolsa com `selenium` e `pandas`. Após a seleção, pelo usuário, de um subconjunto desses ativos em uma interface interativa desenvolvida em `Streamlit`, o sistema utiliza `yfinance` para obter quatro anos de preços de fechamento diários. A partir dessas séries de preços, são calculados os principais indicadores estatísticos necessários para a aplicação do modelo de Markowitz, explicitados a seguir:

- **Retorno médio anual:** o vetor de retornos médios  $\mu \in \mathbb{R}^n$  é obtido como

$$\mu_i = \frac{1}{T} \sum_{t=1}^T r_{i,t} \times 252,$$

onde  $T$  é o número de dias da amostra,  $r_{i,t} = \log\left(\frac{P_{i,t}}{P_{i,t-1}}\right)$ , em que  $P_{i,t}$  é o preço de fechamento no dia  $t$ , e o fator 252 corresponde ao número médio de pregões por ano.

- **Matriz de covariância anualizada:** a matriz  $\Sigma \in \mathbb{R}^{n \times n}$  que representa a covariância entre os ativos é obtida como

$$\Sigma = \text{Cov}(r) \times 252,$$

em que  $\text{Cov}(r)$  é a matriz de covariância dos retornos diários.

- **Volatilidade anual:** para cada ativo  $i$ ,

$$\sigma_i = \sqrt{\Sigma_{ii}},$$

isto é, a raiz quadrada da variância anualizada.

A partir desses indicadores, podemos definir métricas para portfólios. Para  $n$  ativos selecionados, definimos um vetor de pesos  $w$ , em que  $w_i$  representa a fração do capital investida no ativo  $i$ , de  $i = 1, \dots, n$ . Assim, o retorno esperado é dado pela média ponderada dos retornos médios dos ativos segundo seus respectivos pesos:

$$R(w) = \mu^T w, \quad (1)$$

enquanto o risco (medido pela volatilidade) é dado por:

$$\sigma(w) = \sqrt{w^T \Sigma w}. \quad (2)$$

que incorpora não apenas a variância individual de cada ativo, mas também suas covariâncias — refletindo, portanto, os efeitos de diversificação.

Finalmente, para quantificar a relação entre retorno e risco de uma carteira, utiliza-se o índice de Sharpe,

$$S(w) = \frac{R(w) - r_f}{\sigma(w)}, \quad (3)$$

que compara o retorno excedente da carteira em relação à taxa livre de risco  $r_f$  com a volatilidade total assumida.

No projeto, essa taxa livre de risco é definida como parâmetro ou obtida diretamente da API pública do Banco Central do Brasil, por meio da função `obter_selic_atual`, que consulta o valor mais recente da meta da taxa Selic e o converte para base anualizada; em caso de falha na requisição, adota-se um valor padrão (10,75% ao ano).

### 3.3 Formulação matemática

Seja  $n$  o número de ativos selecionados e seja  $w = (w_1, \dots, w_n)^T$  o vetor de decisão, em que  $w_i$  representa a fração do capital investida no ativo  $i$ . As grandezas utilizadas na modelagem são:

- $\mu \in \mathbb{R}^n$ : vetor de retornos médios anuais estimados;
- $\Sigma \in \mathbb{R}^{n \times n}$ : matriz de covariância anualizada dos retornos;
- $r_f \in \mathbb{R}$ : taxa livre de risco anual;
- $\mathbf{1} \in \mathbb{R}^n$ : vetor coluna de uns.

O desempenho ajustado ao risco de uma carteira é medido pelo índice de Sharpe, calculado como em equação (3). Nesse caso, queremos maximizar o índice de Sharpe, mas podemos colocar na forma mais padrão de minimização buscando minimizar o oposto do índice. Assim, o problema de interesse passa a ser:

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & f(w) = -\frac{\mu^T w - r_f}{\sqrt{w^T \Sigma w}} \\ \text{sujeito a} \quad & \mathbf{1}^T w = 1, \\ & w_i \geq 0, \quad i = 1, \dots, n. \end{aligned} \tag{4}$$

A função objetivo representa o negativo do índice de Sharpe, de modo que sua minimização equivale a maximizar o retorno excedente por unidade de risco. As restrições garantem, respectivamente, que todo o capital seja alocado e que não haja posições vendidas. O domínio viável é, portanto, um *simplexo*<sup>[10]</sup>.

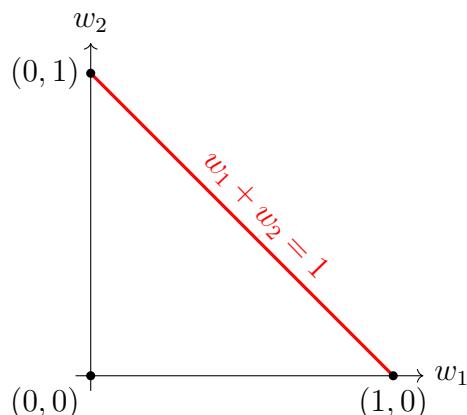


Figura 2: Região viável do problema com duas ações.

A seguir, a Figura 3 mostra um mapa de calor do índice de Sharpe para combinações de três ativos (UGPA3, MULT3 e WEGE3) e taxa livre de risco 0.05, onde cada ponto representa uma alocação possível entre os dois primeiros ativos e a terceira é ajustada para que a soma dos pesos seja igual a 1. As cores indicam o valor do Sharpe, permitindo visualizar quais proporções de investimento geram maior relação retorno-risco dentro da região válida (todos pesos não negativos).

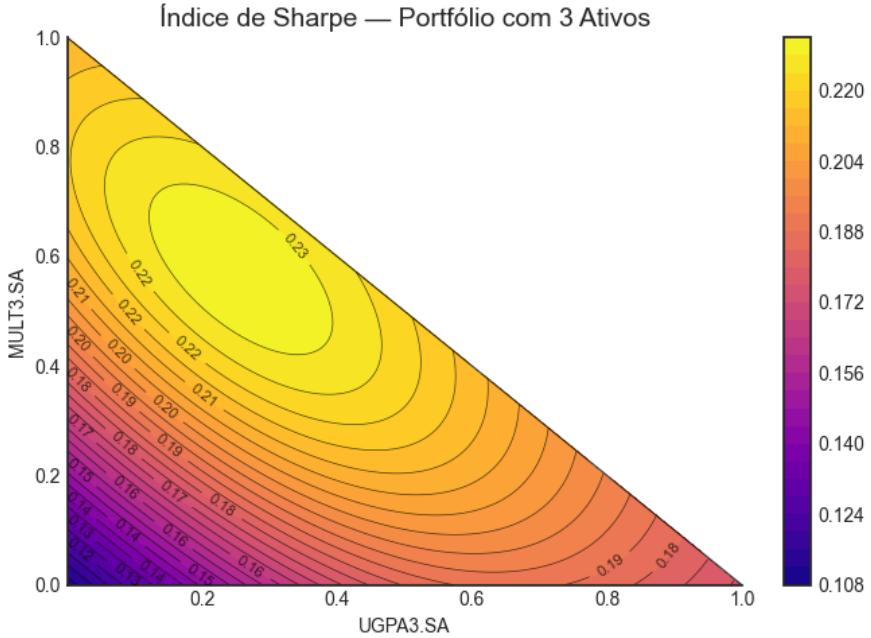


Figura 3: Gráfico do Índice de Sharpe para o problema (4) com três ações.

## 4 Métodos de resolução

Nesta seção descrevemos os métodos de resolução utilizados. Primeiramente tratamos algumas ideias comuns aos métodos e de consideração importante.

### Gradiente e Hessiana

Adotando as notações (1) e (2), temos para a função objetivo  $f(w) = -S(w)$  as seguintes expressões das derivadas:

$$\frac{\partial f}{\partial \mathbf{w}} = -\frac{\mu}{\text{cov}(\mathbf{w})} + \frac{R(\mathbf{w}) - r_f}{\text{cov}(\mathbf{w})^{3/2}} \Sigma \mathbf{w} \quad (5)$$

$$\frac{\partial^2 f}{\partial \mathbf{w}^2} = \frac{\mu (\Sigma \mathbf{w})^\top + (\Sigma \mathbf{w}) \mu^\top}{\text{cov}(\mathbf{w})^{3/2}} + \frac{(R(\mathbf{w}) - r_f) \Sigma}{\text{cov}(\mathbf{w})^{3/2}} - \frac{3(R(\mathbf{w}) - r_f)}{\text{cov}(\mathbf{w})^{5/2}} (\Sigma \mathbf{w})(\Sigma \mathbf{w})^\top \quad (6)$$

que podem ser obtidos facilmente mediante derivação simbólica usando sympy<sup>[6]</sup>.

A Hessiana *não* é, em geral, positiva (nem negativa) definida, pode ser indefinida. Então é preciso cautela na escolha dos métodos de passo linear.

## Projeção no simplexo<sup>[3]</sup>

A projeção de um vetor  $v \in \mathbb{R}^n$  no simplexo padrão

$$\Delta = \{w \in \mathbb{R}^n : w_i \geq 0, \sum_{i=1}^n w_i = 1\}$$

corresponde a resolver

$$\min_{w \in \Delta} \|w - v\|_2^2.$$

A solução ótima possui necessariamente a forma

$$w_i = \max(v_i - \theta, 0), \quad (7)$$

onde  $\theta$  é um parâmetro que ajusta o vetor para satisfazer a restrição  $\sum_i w_i = 1$ . O problema reduz-se, portanto, a determinar esse único escalar.

O método consiste em ordenar  $v$  em ordem decrescente, obtendo  $u$ , e computar as somas acumuladas  $S_k = \sum_{i=1}^k u_i$ . Busca-se então o maior índice  $\zeta$  que satisfaz

$$u_\zeta > \frac{S_\zeta - 1}{\zeta}.$$

Esse índice identifica exatamente quantos elementos permanecem positivos após a projeção. Define-se então

$$\theta = \frac{S_\zeta - 1}{\zeta}.$$

Substituindo este valor em (7), obtém-se o vetor projetado no simplexo. O algoritmo é  $O(n \log n)$  devido à ordenação e fornece a projeção exata.

### 4.1 Método 1: Gradiente Ascendente Projetado

O código adota uma abordagem dividida em duas fases: uma tentativa de solução analítica fechada e, caso esta viole as restrições, uma solução numérica via Gradiente Ascendente com projeção no simplexo.

#### Solução Analítica (Irrestrita)

Inicialmente, o algoritmo relaxa as restrições de desigualdade ( $0 \leq w_i \leq 1$ ). A solução para o portfólio tangente sem limites de pesos é dada por:

$$\mathbf{w}^* = \frac{\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - r_f \cdot \mathbf{1})}{\mathbf{1}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - r_f \cdot \mathbf{1})} \quad (8)$$

Na implementação:

- **Inversão Matricial:** Calcula-se a inversa da covariância. Se a matriz for singular, utiliza-se a pseudoinversa para robustez numérica.
- **Verificação de Viabilidade:** Se o vetor resultante respeitar as restrições de caixa (dentro de uma tolerância  $\epsilon = 10^{-6}$ ), a solução é aceita imediatamente, economizando custo computacional.

## Solução Numérica (Gradiente Ascendente)

Se a solução analítica violar as restrições (exigindo venda a descoberto), o algoritmo utiliza um método iterativo de primeira ordem, fazendo uso do gradiente do Índice de Sharpe dado em (5). O passo é calculado por:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \nabla f(\mathbf{w}_k)$$

## Backtracking

Além disso, em vez de abordagens com taxa de aprendizado fixa ou decaimento simples, o método implementa uma busca linear com **backtracking** (condição de Armijo). A cada iteração, procura-se um passo  $\alpha_k$  que garanta um aumento suficiente na função objetivo. Isso assegura a convergência global e evita oscilações excessivas próximas ao ótimo.

## Projeção no Simplexo

A inovação crítica para o tratamento das restrições reside no operador de projeção  $\mathcal{P}_\Omega$ . Após o passo de gradiente, o vetor resultante pode sair da região viável. Em vez de utilizar heurísticas mais simples como *clipping*, aplica-se o método de projeção discutido anteriormente.

## 4.2 Método 2: Barreiras Logarítmicas com Penalidade Quadrática

### Restrições e Redefinição do Problema

A implementação computacional adota uma abordagem mista para tratar as restrições. As restrições de desigualdade ( $w_i > 0$ ) continuam sendo tratadas pelo método de barreira logarítmica, enquanto a restrição de igualdade ( $\sum w_i = 1$ ) é incorporada à função objetivo através de uma penalidade quadrática.

Definimos a função de barreira para a não-negatividade:

$$\phi(\mathbf{w}) = - \sum_{i=1}^n \log(w_i).$$

Para a restrição de orçamento (soma dos pesos), introduzimos o termo de penalidade quadrática:

$$P(\mathbf{w}) = \frac{\rho}{2} \left( \sum_{i=1}^n w_i - 1 \right)^2.$$

No algoritmo implementado, o parâmetro de penalidade  $\rho$  é acoplado ao parâmetro de barreira  $t$ , de modo que  $\rho = t$ . Isso implica que, conforme a barreira se torna mais "fina" (permitindo aproximação da fronteira  $w_i = 0$ ), a penalidade pela violação da soma torna-se mais severa.

A função de custo global  $f_t(\mathbf{w})$ , parametrizada por  $t$ , torna-se:

$$f_t(\mathbf{w}) = -S(\mathbf{w}) - \frac{1}{t} \sum_{i=1}^n \log(w_i) + \frac{t}{2} \left( \mathbf{1}^\top \mathbf{w} - 1 \right)^2.$$

Desta forma, transformamos o problema original em uma sequência de problemas de minimização irrestrita que, no limite  $t \rightarrow \infty$ , forçam a solução a respeitar ambas as restrições.

## Minimização e Gradientes

O problema irrestrito foi resolvido por meio do algoritmo BFGS, pertencente à família dos métodos quase-Newton discutidos na disciplina, que aproxima a Hessiana inversa. A implementação utilizou a função `minimize` da biblioteca *SciPy*<sup>[9]</sup>, que requer, para maior eficiência e precisão, o fornecimento explícito do gradiente analítico da função objetivo combinada.

As derivadas dos componentes da função de custo são dadas por:

$$\nabla_{\mathbf{w}} \left( -\frac{1}{t} \sum \log(w_i) \right) = -\frac{1}{t} \mathbf{w}^{-1} \quad \text{e} \quad \nabla_{\mathbf{w}} \left( \frac{t}{2} (\mathbf{1}^\top \mathbf{w} - 1)^2 \right) = t (\mathbf{1}^\top \mathbf{w} - 1) \mathbf{1}$$

O gradiente completo utilizado na otimização é a soma dos gradientes do Sharpe negativo, da barreira e da penalidade:

$$\nabla_{\mathbf{w}} f_t(\mathbf{w}) = \nabla(-S(\mathbf{w})) = \underbrace{-\frac{1}{t} \mathbf{w}^{-1}}_{\text{Gradiente Barreira}} + \underbrace{t \left( \sum_{i=1}^n w_i - 1 \right) \mathbf{1}}_{\text{Gradiente Penalidade}}.$$

## Estratégia de Atualização

O algoritmo segue uma estratégia de incremento iterativo do parâmetro  $t$ . Inicializando com  $t_{init}$  e um fator de crescimento  $\eta$ , o processo resolve o subproblema  $\min_{\mathbf{w}} f_{t_k}(\mathbf{w})$  usando BFGS até convergência ou número máximo de iterações internas, atualiza o parâmetro de rigidez:  $t_{k+1} = \eta t_k$  e, por fim, utiliza a solução ótima da iteração anterior  $\mathbf{w}^{(k)}$  como ponto de partida (*warm start*) para a iteração  $k + 1$ . O loop externo encerra quando  $t$  atinge um valor máximo ( $10^6$ ) ou quando a variação entre soluções consecutivas é desprezível, indicando convergência para o ponto ótimo viável.

## 4.3 Método 3: Gradiente Espectral Projetado

### Restrições e Tratamento do Espaço Viável

Na implementação computacional do Método do Gradiente Espectral Projetado (em inglês *Spectral Projected Gradient*, ou SPG) as restrições, tanto de igualdade quanto de desigualdade, são satisfeitas mantendo-se a solução iterativa estritamente dentro do conjunto viável  $\Omega$  (o simplex unitário), o que é feito modificando a direção de busca. Definimos o operador de projeção  $P_\Omega(v)$ , que mapeia qualquer vetor  $v \in \mathbb{R}^n$  para o ponto mais próximo dentro do conjunto viável:

$$P_\Omega(v) = \arg \min_{z \in \Omega} \|z - v\|_2 \tag{9}$$

Desta forma, o problema original de minimização do Sharpe Negativo  $-S(w)$  é tratado diretamente na sua forma restrita, garantindo que todo candidato  $w_k$  gerado pelo algoritmo seja um portfólio válido, eliminando a necessidade de ajustes de hiperparâmetros de penalidade ( $\rho$  ou  $t$ ).

## Minimização e Passo Espectral

O problema é resolvido utilizando uma estratégia de descida que combina o gradiente projetado com o passo espectral de Barzilai-Borwein (BB). Este método

aproxima a curvatura da função (informação de segunda ordem) através de um escalar  $\lambda_k$ , sem a necessidade de calcular a matriz Hessiana.

Para tal, definem-se as variações de posição e gradiente entre iterações consecutivas:

$$s_{k-1} = w_k - w_{k-1} \quad \text{e} \quad y_{k-1} = \nabla(-S(w_k)) - \nabla(-S(w_{k-1})) \quad (10)$$

O passo espectral  $\lambda_k$  é calculado para satisfazer a equação secante no sentido de mínimos quadrados:

$$\lambda_k = \frac{s_{k-1}^\top s_{k-1}}{s_{k-1}^\top y_{k-1}} \quad (11)$$

A direção de busca  $d_k$  não é simplesmente o gradiente negativo, mas sim o resíduo da projeção do passo espectral:

$$d_k = P_\Omega(w_k - \lambda_k \nabla(-S(w_k))) - w_k \quad (12)$$

O gradiente  $\nabla(-S(w))$  utilizado é derivado analiticamente da função de Sharpe, conforme apresentado nos métodos anteriores, mas aqui ele é escalado por  $\lambda_k$  antes da projeção, permitindo passos mais longos e convergência acelerada em vales estreitos.

### Estratégia de Atualização

Em cada iteração externa  $k$ , calcula-se o passo  $\lambda_k$  (limitado entre  $[\lambda_{\min}, \lambda_{\max}]$ ) e a direção  $d_k$ . Em seguida, realiza-se uma busca linear (*backtracking*) para encontrar um tamanho de passo  $\alpha \in (0, 1]$  que satisfaça a condição de decrescimento suficiente:

$$f(w_k + \alpha d_k) \leq f(w_k) + c \cdot \alpha \nabla f(w_k)^\top d_k \quad (13)$$

Onde  $f(w) = -S(w)$ . O ponto é atualizado como  $w_{k+1} = w_k + \alpha d_k$ . O ciclo encerra quando a norma da direção projetada (medida de estacionariedade) é inferior a uma tolerância  $\epsilon$  ( $10^{-6}$ ), indicando que nenhum passo de descida viável adicional pode ser tomado, ou seja,  $P_\Omega(w_k - \nabla f(w_k)) \approx w_k$ .

## 5 Implementação

Os métodos numéricos foram implementados em Python utilizando herança e polimorfismo. Todas as estratégias de solução derivam da classe base abstrata `Otimizacao`. O construtor desta classe é responsável pela inicialização dos atributos de estado (retorno, covariância e taxa livre de risco), enquanto seus métodos concretos fornecem as ferramentas matemáticas compartilhadas, como a função de Sharpe, derivadas analíticas e operadores de projeção. O processo de otimização é padronizado através do método abstrato `fit`, cuja implementação é obrigatória e exclusiva para cada classe filha, garantindo uma interface consistente.

As classes concretas foram implementadas. Com exceção da utilização do método Quase-Newton BFSG que foi utilizado para realizar as minimizações intermediárias no Método com **Barreiras Logaritmicas e Penalização Quadrática**, o restante teve implementação própria.

## Parâmetros Iniciais

Os métodos implementados compartilham parâmetros básicos no procedimento `fit`, comuns a todas as classes derivadas: a tolerância de parada, definida como  $tol = 10^{-6}$ , e o número máximo de iterações, fixado em  $maxiter = 3000$ .

Para o método de *Barreira Logarítmica com Penalização Quadrática*, adotaram-se como parâmetros do caminho central um valor inicial  $t = 1.0$  e um fator de crescimento  $t_{\text{factor}} = 4.0$ , responsável por intensificar progressivamente tanto a barreira quanto a penalização da restrição de soma.

Nos métodos *Gradiente Ascendente Projetado* e *Gradiente Espectral Projetado (SPG)*, a busca linear por backtracking utiliza um passo inicial  $\alpha_0 = 1.0$ , fator de redução  $\rho = 0.5$  e constante de Armijo  $c = 10^{-4}$ , garantindo estabilidade e convergência global. Para o SPG, adicionalmente, o passo espectral é limitado pelos parâmetros  $\lambda_{\min} = 10^{-3}$  e  $\lambda_{\max} = 10^3$ , assegurando controle numérico adequado sobre a curvatura aproximada.

## 6 Resultados Numéricos

Realizamos uma análise comparativa entre os diferentes métodos numéricos de otimização implementados. O objetivo principal foi avaliar o desempenho relativo de cada técnica quando aplicada ao problema de maximização do índice de Sharpe. Foram considerados três tamanhos de universo de ativos —  $n = 3$ ,  $n = 10$  e  $n = 25$  — com o intuito de avaliar o comportamento das heurísticas tanto em cenários de baixa dimensionalidade quanto em dimensões moderadas, onde a complexidade do problema se torna mais evidente.

Para cada dimensão considerada,  $n \in \{3, 10, 25\}$ , foi realizada uma amostragem aleatória sem reposição a partir do conjunto completo de ativos, de modo a obter um subconjunto distinto de *ações* em cada experimento. Essa escolha aleatória assegura que os resultados não sofram viés decorrente de seleção manual. Cada tempo de otimização é calculado 5 vezes e tirado a média para mitigar variações de processamento. Adicionalmente, como referência de comparação, empregamos um minimizador de *benchmark* disponível na biblioteca *scipy* — especificamente o método *SLSQP* — utilizado apenas como gabarito para avaliar o desempenho relativo das heurísticas.

N	Método	Sharpe	Gap Sharpe	Tempo (s)	Iterações
3	Benchmark	0.15282	0.00000	-	4
3	Barreira + Penalizacao	0.15281	0.00001	0.48536	84
3	Gradiente Projetado	0.15282	0.00000	0.00119	3
3	SPG	0.15282	0.00000	0.00081	2
10	Benchmark	1.40019	0.00000	-	2
10	Barreira + Penalizacao	1.40015	0.00003	1.82535	176
10	Gradiente Projetado	1.40019	0.00000	0.00415	1
10	SPG	1.40019	0.00000	0.00105	1
25	Benchmark	0.64128	0.00000	-	11
25	Barreira + Penalizacao	0.64120	0.00008	7.80132	284
25	Gradiente Projetado	0.64128	0.00000	0.16332	22
25	SPG	0.64128	0.00000	0.09371	12

Tabela 1: Comparação de Desempenho dos Métodos de Otimização

Os experimentos computacionais realizados evidenciam divergências fundamentais

na robustez e eficiência entre as estratégias de penalidade e as abordagens baseadas em projeção direta. No que tange à precisão da convergência, os métodos de projeção, especificamente o Gradiente Espectral Projetado (SPG) e o Gradiente Projetado clássico, demonstraram consistência superior. O método de Barreira Logarítmica com Penalidade Quadrática, embora eficaz em pequenas dimensões, apresentou uma falha crítica no cenário de  $N = 25$ , convergindo para uma solução inviável com Índice de Sharpe negativo, o que denota sensibilidade ao ajuste de hiperparâmetros e problemas de mal-condicionamento à medida que o espaço de busca se expande.

Sob a ótica da eficiência computacional, a vantagem do passo espectral tornou-se clara no cenário de maior complexidade. Enquanto o Gradiente Projetado clássico exigiu 22 iterações para convergir com 25 ativos, o SPG atingiu a mesma solução ótima em apenas 12 iterações, reduzindo o tempo de execução pela metade (0.09s contra 0.18s). Este resultado corrobora a teoria de que a incorporação de informações de segunda ordem (curvatura) permite passos mais longos e assertivos, mitigando o comportamento de "ziguezague" típico dos métodos de primeira ordem. Já a abordagem de Barreira mostrou-se ordens de magnitude mais lenta (4.5s para  $N = 25$ ), penalizada pela necessidade de resolver repetidos subproblemas de otimização irrestrita a cada atualização dos parâmetros de penalidade.

Em síntese, os resultados indicam que a projeção ortogonal é a estratégia dominante para problemas com restrições de orçamento e não-negatividade. O Gradiente Espectral Projetado destacou-se como o método mais equilibrado, aliando a robustez da viabilidade estrita com uma velocidade de convergência comparável a métodos quase-Newton, sem incorrer no alto custo computacional destes.

Limitados a, no máximo, três dimensões para dispor graficamente o problema, podemos utilizar o exemplo da Figura 3 para visualizar como ocorre a execução dos três métodos.

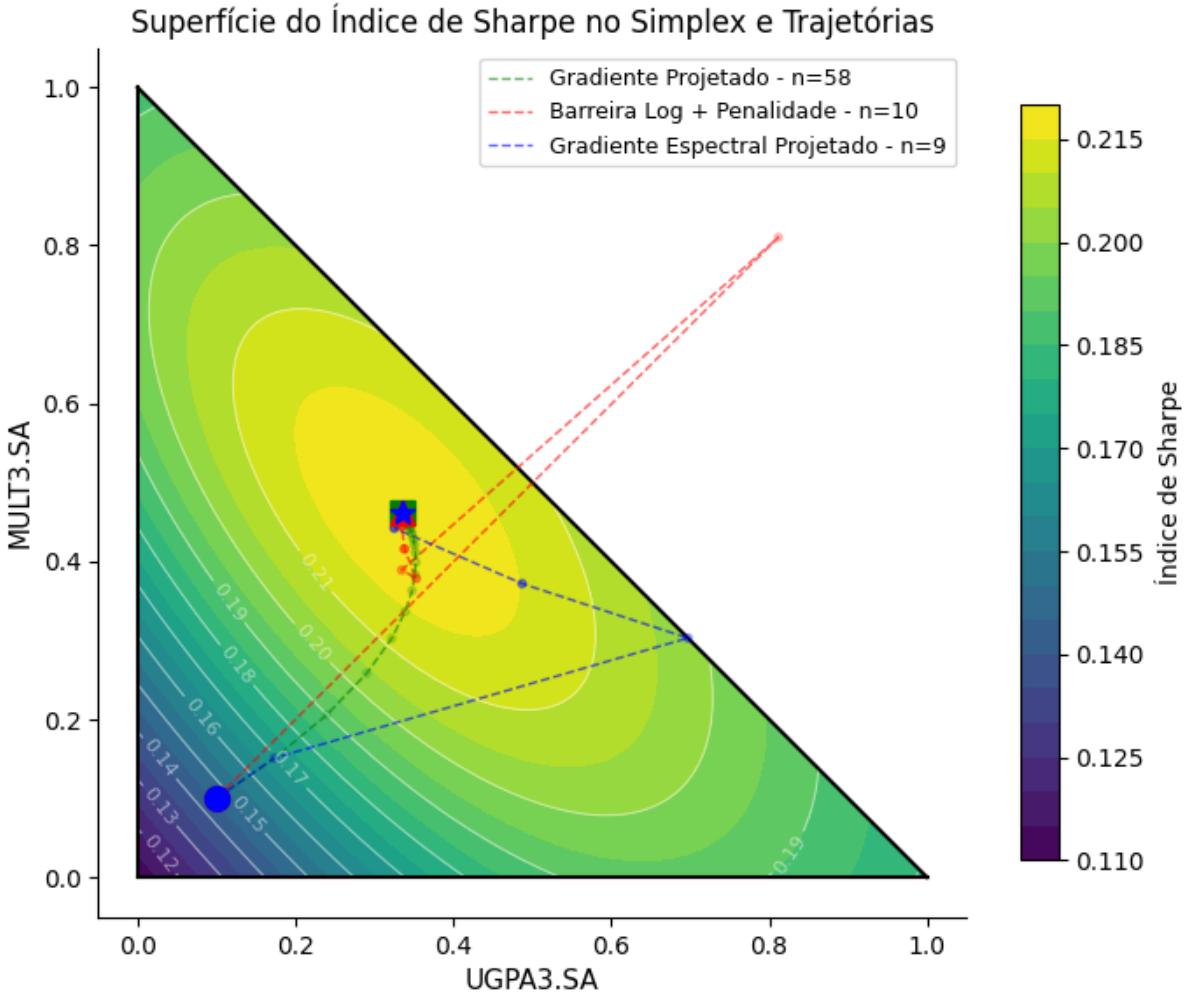


Figura 4: Trajetória dos Métodos no caso de três ativos

Embora a baixa dimensionalidade do problema (três ativos) resulte em um desempenho final semelhante entre as abordagens, na Figura 4 é possível observar diferenças marcantes na trajetória de cada método. No **Método de Barreira Logarítmica com Penalidade Quadrática**, as iterações podem violar temporariamente a região viável, sendo forçadas a retornar apenas pela ação do termo de penalidade. Em contrapartida, os métodos baseados em projeção garantem a viabilidade estrita de todos os iterandos  $x_k$ .

Entre estes, nota-se uma distinção clara: o **Gradiente Projetado** apresenta passos curtos e conservadores — comportamento coerente com a teoria sobre a convergência lenta de métodos de primeira ordem, conforme visto em aula. Já o **Gradiente Espectral Projetado (SPG)** exibe um comportamento mais agressivo, explorando as fronteiras do simplex com passos longos que frequentemente 'ricocheteiam' nas restrições, evidenciando a eficiência do passo espectral.

## 7 Conclusão

O desenvolvimento deste trabalho permitiu consolidar o entendimento prático sobre as diferentes abordagens para o tratamento de restrições em otimização não linear. Foi possível aprender que, embora a transformação de problemas restritos em irrestritos via

métodos de Barreira e Penalidade seja teoricamente sólida, sua aplicação prática impõe desafios significativos de estabilidade numérica e ajuste de parâmetros, especialmente à medida que a dimensionalidade do sistema aumenta. Em contrapartida, compreendeu-se que a utilização de operadores de projeção ortogonal, quando disponíveis para a geometria do problema (como no caso do simplex), oferece um caminho muito mais robusto e intuitivo para garantir a viabilidade da solução, eliminando a necessidade de heurísticas complexas de penalização.

Além disso, ficou evidente que métodos puramente de primeira ordem, como o Gradiente Ascendente clássico, embora simples de implementar, sofrem com a ineficiência em vales estreitos. O estudo do Método do Gradiente Espectral Projetado (SPG) proporcionou o aprendizado valioso de que é possível acelerar drasticamente a convergência utilizando o passo espectral, que aproxima a eficiência de métodos de segunda ordem (Newton) sem incorrer no alto custo computacional do cálculo de Hessianas, demonstrando ser a estratégia mais equilibrada para problemas de alocação de ativos.

## Referências

- [1] ASSAF NETO, Alexandre. *Mercado Financeiro*. 9. ed. São Paulo: Atlas, 2008.
- [2] CHOEVY, Mark; WEIGEND, Andreas S. *Nonlinear Trading Models Through Sharpe Ratio Maximization*. Leonard N. Stern School of Business, New York University, 1996.
- [3] DUCHI, J.; SHALEV-SHWARTZ, S.; SINGER, Y.; CHANDRA, T. Efficient Projections onto the L1-Ball for Learning in High Dimensions. In: Proceedings of the 25th International Conference on Machine Learning (ICML). ACM, 2008. Disponível em: <https://dl.acm.org/doi/10.1145/1390156.1390191>. Acesso em: 02 dez. 2025.
- [4] ELTON, Edwin J.; GRUBER, Martin J.; BROWN, Stephen; GOETZMANN, William. *Moderna teoria de carteira e análise de investimentos*. São Paulo: Elsevier, 2012.
- [5] MARKOWITZ, Harry. Portfolio Selection. *The Journal of Finance*, 1952.
- [6] MEURER, Aaron; SMITH, Christopher P.; PAPAGHEORGHIOS, Mateusz; et al. SymPy: symbolic computing in Python. PeerJ Computer Science, v. 3, n. e103, 2017.
- [7] NOCEDAL, Jorge; WRIGHT, Stephen J. *Numerical Optimization*. 2. ed. New York: Springer, 2006.
- [8] QUANTPEDIA. Markowitz Model. Disponível em: <https://quantpedia.com/markowitz-model/#:~:text=Tangency%20portfolio%2C%20the%20red%20point,an%20E2%80%9Coptimal%20portfolio%2E%80%9D>. Acesso em: 30 nov. 2025.
- [9] THE SCIPY COMMUNITY. `scipy.optimize.minimize`. SciPy v1.11.4 Reference Guide. Disponível em: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html>. Acesso em: 02 dez. 2025.
- [10] WIKIPEDIA. Simplex. Disponível em: <https://en.wikipedia.org/wiki/Simplex>. Acesso em: 30 nov. 2025.