

# Financial Portfolio Optimization

Bernardo Gomes Kishan Rama Tomás Falcato

bernardo.n.gomes@ist.utl.pt kishan.rama@ist.utl.pt

tomas.falcato.costa@ist.utl.pt

Instituto Superior Técnico

**Abstract**—Para o problema de *Financial portfolio optimization*, dado um conjunto de *assets*, pretende-se obter uma forma de distribuir o dinheiro a investir por forma a maximizar o retorno. A cada *asset*, existe um risco associado, sendo que o risco de cada um poderá, ou não, influenciar o risco dos restantes. Se considerarmos todas as variáveis inerentes à otimização deste tipo de problemas, iremos obter funções não convexas, pelo que se considera uma aproximação facilmente computável através de otimização convexa e cujos resultados, obtidos a partir do *software CVX*, são bastante próximos dos ótimos. Posteriormente, o problema foi reformulado por forma a obter uma solução sub-ótima mas com menor tempo de computação.

**Keywords.** Portfolio Optimization •CVX •Barrier Method •Newton Method

## I. INTRODUÇÃO

Este artigo pretende obter uma solução para o problema de *Portfolio Optimization*. Para tal, a nossa abordagem maximiza o retorno esperado e minimiza o risco de investimento tendo em conta a aversão ao risco colocada pela pessoa que investe.

Na vida real, o problema discutido neste artigo tem extrema importância na medida em que praticamente todas as pessoas fazem investimentos do seu dinheiro. A formulação terá em conta não só os casos em que a pessoa pretende rápidos retornos e pretende que estes sejam bastante elevados, independentemente de o risco a ele associado ser relativamente elevado, como também casos cujo objetivo seja obter um resultado de longo prazo em que a aversão ao risco é um fator preponderante face à quantia recebida.

Por forma a resolver, e melhor compreender o problema, numa primeira abordagem foi utilizado o *software CVX*. Posteriormente, implementa-se a mesma resolução do problema, mas com este reescrito por forma a utilizar o algoritmo *Barrier Method*.

O artigo está organizado da forma enunciada de seguida. No capítulo II, é introduzida a função a otimizar e as respetivas variáveis, capítulo III descreve a abordagem para a utilização do *software CVX*, sendo esta alterada para a utilização do *Barrier Method*, descrito no capítulo IV. Os resultados obtidos nas duas fases são discutidos e comparados em V. As conclusões são apresentadas posteriormente em VI.

## II. FORMULAÇÃO DO PROBLEMA

Para a obtenção dos resultados do problema descrito anteriormente, a função utilizada foi a seguinte:

$$\begin{aligned} & \text{maximize} && \mu^\top \omega - \gamma \omega^\top \Sigma \omega \\ & \text{subject to} && 1^\top \omega = 1 \\ & && \omega \in \mathbb{R}_+^n \end{aligned} \quad (1)$$

Apresenta-se de seguida a descrição das variáveis da função de otimização:

- $\omega \rightarrow$  vetor de dimensão  $n$ , onde cada posição  $i$ , contém a fração do dinheiro a investir no *asset*  $i$  para obter o máximo de lucro. Será esta a variável que se pretende otimizar;
- $\mu \rightarrow$  vetor de dimensão  $n$ , onde cada posição  $i$ , contém o retorno esperado para o *asset*  $i$ ;
- $\Sigma \rightarrow$  matriz das covariâncias, de dimensão  $n \times n$  dos retornos dos *assets*. É através deste parâmetro que se representa a influência que cada *asset* tem nos restantes e vice-versa. Por exemplo, é de esperar que tendo um valor muito elevado na entrada  $ij$  desta matriz, no caso de haver uma quebra nos retornos do *asset*  $i$ , o *asset*  $j$  irá, também sofrer quebra nos retornos, ou um aumento dos mesmos, dependendo do valor dessa entrada. Um caso real desta formulação é a forte dependência entre as ações do petróleo e dos automóveis ou a relação de competição entre cadeias concorrentes;
- $\mu^\top \omega \rightarrow$  representa o retorno expectável do portfólio;
- $\gamma \rightarrow$  variável que controla o *tradeoff* entre o retorno e o risco. Será este parâmetro que irá controlar se se pretende investir dando preferência a maximizar o retorno, ou dando preferência à minimização do risco;
- $\gamma \omega^\top \Sigma \omega \rightarrow$  representa a variância do retorno do portfólio. É o parâmetro que mede o risco de investimento;

A *constraint*  $1^\top \omega = 1$  indica que a soma dos investimentos no portfólio deve corresponder ao capital inicial que se pretende investir. Note-se que os investimentos são uma fração do capital inicial e não o seu valor absoluto. Ao forçar que  $\omega \in \mathbb{R}_+^n$  indica que não podem existir investimentos negativos. Na prática esta situação poderia existir, no caso de termos capital num dado *asset* e a melhor solução ser retirar o capital deste para investir noutro mais lucrativo.

Ao analisar a função que queremos otimizar, verifica-se que é quadrática, ou seja, do tipo  $x^\top A x + b^\top x + c$ . Para o contexto em questão, tem-se que  $\text{maximize } \mu^\top \omega - \gamma \omega^\top \Sigma \omega \equiv \text{minimize } \gamma \omega^\top \Sigma \omega + (-\mu^\top) \omega \equiv \text{minimize } \omega^\top \gamma \Sigma \omega + (-\mu^\top) \omega$ .

Sendo a matriz das covariâncias semi-definida positiva e simétrica, verifica-se que a nossa função é realmente quadrática sendo  $A = \gamma\Sigma$ ,  $b = -\mu$  e  $c = 0$ . Sendo a função quadrática, podemos também concluir, que é convexa. Desta forma, a otimização será feita de acordo com esta propriedade.

### III. CVX

Por forma a implementar a solução apresentada na equação 1, e sendo esta convexa, utilizou-se o *software* CVX[1]. De seguida apresenta-se a parte do código de *Matlab* onde se utiliza o CVX.

```
cvx_begin
    variable w(number_of_entries);

    maximize(miu'*w - gama*w'*cov*w);

    subject to
        (vec_ones')*w == 1; w>=0;
cvx_end;
```

### IV. BARRIER METHOD

Por forma a melhorar o tempo de otimização, utilizou-se um *Interior Point Method*, visto que o problema pertence à categoria dos *Inequality and equality-constrained smooth problems*, categoria esta que é reduzida para uma sequência de *Equality-constrained smooth problems*, a partir da utilização de *Interior Point Methods*. Iremos descrever como fizemos esta transformação ao longo do capítulo.

Para esta abordagem, pretende-se reescrever o problema sob a forma de problema de otimização da seguinte forma:

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) \leq 0 \\ &&& Ax = b \end{aligned} \quad (2)$$

As *constraints* de otimização devem portanto ser lineares ou inequações do tipo acima descrito em que os  $f_i$  são convexos e duas vezes diferenciáveis.

Por forma a traduzir a formulação de (2) numa formulação de um *Equality-constrained smooth problem*, e utilizando *Unconstrained Minimization Technique*, obtem-se (4), que é uma aproximação de (3).

$$\begin{aligned} &\text{minimize} && f_0(x) + \sum_{i=1}^m I_-(f_i(x)) \\ &\text{subject to} && Ax = b \end{aligned} \quad (3)$$

$$\begin{aligned} &\text{minimize} && f_0(x) + (1/t) \sum_{i=1}^m \log(f_i(x)) \\ &\text{subject to} && Ax = b \end{aligned} \quad (4)$$

Em (3),  $I_{-}f_i(x)$  é a função indicadora e representa uma barreira para a função de custo, ou seja, para  $f_i(x) \leq 0$ , tem valor nulo sendo infinito para os restantes. Isto traduz-se na penalização infinita caso as *constraints* se aproximem da violação das condições impostas pelas inequações. Na equação (4),  $(1/t)\log(f_i(x))$  representa a *log barrier*, ou seja, a aproximação da barreira à função indicadora. À medida que  $t$  aumenta, a *log barrier* aproxima melhor a função indicadora,

obtendo-se assim resultados cada vez mais semelhantes à solução ótima. Está ilustrado na figura 1 o aumento de  $t$  e a sua consequência na *log barrier*.

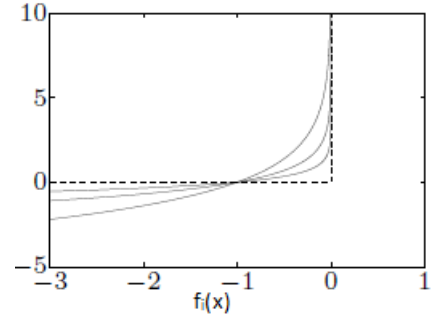


Fig. 1: Aproximação da função indicadora por vários valores de  $t$

Não sendo possível a implementação de funções não diferenciáveis, faz-se uma aproximação logarítmica, representada em (4), da barreira onde é possível aplicar o método de *Newton*. Desta forma, escolhendo uma tolerância e um valor para  $t$ , será possível obter uma solução sub-ótima para o problema em apenas uma iteração. Utiliza-se o *Barrier Method* que permite efetuar várias otimizações de  $f(z)$  em função de  $t$ . Quando  $t$  tende para infinito espera-se que  $x^*(t)$  tenda para  $x^*$ , e que este valor seja a solução do nosso problema. As soluções de cada iteração estão situadas ao longo do *central path* que corresponde ao conjunto de soluções  $x^*(t)$  e são caracterizadas pelas *KKT Conditions*. A cada iteração do *Barrier Method*, *outer iteration*, é computada uma solução sub ótima, que é depois utilizada como ponto inicial da próxima iteração. Estende-se esta computação até que o *stopping criterion* seja verificado. A cada iteração o valor de  $t$  é aumentado por um fator  $\mu$ , que controla o valor que a *log barrier* toma. O *stopping criterion* é obtido através de  $f(x^*(t)) - f^* \leq m/t$ , em que  $m$  é o número de *constraints* existentes no problema.

No *Barrier Method* a escolha dos parâmetros  $t$  e  $\mu$  é de grande importância. Caso  $\mu$  seja muito pequeno irão ser necessárias muitas *outer iterations*, visto que o aumento de  $t$  será reduzido, por outro lado, caso  $\mu$  seja muito elevado, irão ser necessárias muitas iterações do método de *Newton*, já que as soluções sub ótimas de iterações consecutivas irão estar muito afastadas entre si. Quanto ao valor de  $t$  inicial, se este for muito pequeno, irão ser também necessárias muitas *outer iterations*, a adaptação à função indicadora será feita lentamente, caso  $t$  inicial seja muito elevado, a primeira iteração do método de *Newton* (*first centering step*) irá requerir muitas iterações para calcular  $x^*$  inicial. Por último, o *Barrier Method* converge num número finito de iterações.

Pela impossibilidade de aplicação deste método com a formulação anterior devido à *constraint*  $1^\top \omega = 1$ , foi necessário recorrer a uma mudança de variável.

Deste modo, define-se  $\omega$  pela equação (5).

$$\omega = Dz + b \quad (5)$$

As matrizes serão as seguintes:

$$D_{n,n-1} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -1 & -1 & -1 & \cdots & -1 \end{pmatrix}$$

$$z_{n-1,1} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-1} \end{pmatrix} \quad b_{n,1} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

Com esta mudança de variável, a nova variável de otimização  $z$ , tem menos um grau de liberdade que a anterior,  $\omega$ . O último investimento será assim a percentagem em falta do capital inicial.

Esta condição é conseguida através da equação matricial (5), onde a matriz  $D$  é uma matriz identidade nas primeiras  $n-1$  linhas e a última linha desta matriz conter apenas "-1"s; a matriz  $b$  é uma matriz coluna de "0"s com apenas a ultima posição "1", sendo a *constraint* referida anteriormente assim respeitada.

Relativamente à segunda *constraint*,  $\omega \in \mathbb{R}_+^n$ , com a mudança de variável efetuada, tem-se  $Dz + b \geq 0$ . Como tal, o problema descrito pela equação (2) adaptado à situação em causa ficará escrito da seguinte forma:

$$\begin{aligned} \text{minimize} \quad & -\mu^\top (Dz + b) + \gamma (Dz + b)^\top \Sigma (Dz + b) \\ \text{subject to} \quad & -Dz - b \leq 0 \end{aligned} \quad (6)$$

Os  $f_i(x)$  são assim  $f_i(x) = -\delta_i^\top z - b_i$  onde  $\delta_i^\top$  representa a linha  $i$  da matriz  $D$ .

Neste algoritmo, faz-se num primeiro passo a minimização com recurso ao método de *Newton*, sendo a função utilizada a apresentada na equação (7),

$$f(z) = t f_0(z) + \phi(z) \quad (7)$$

onde

$$f_0(z) = -\mu^\top (Dz + b) + \gamma (Dz + b)^\top \Sigma (Dz + b) \quad (8)$$

e

$$\phi = -\sum_{i=1}^m \log(-f_i(x)) \quad (9)$$

O termo  $\phi$  evita que a minimização com o método de *Newton* vá para soluções não permitidas, uma vez que a função de otimização assume o valor infinito (*penalty*), obrigando o método de *Newton* a calcular outro *step* para encontrar uma solução viável.

Para a aplicação do método de *Newton*, é ainda necessário o cálculo do gradiente e da hessiana da função  $f(z)$ . Como tal, o gradiente é obtido da seguinte forma:

$$\nabla f(z) = t(\nabla f_0(z))^\top + \nabla \phi(z) \quad (10)$$

onde

$$\nabla f_0(z) = -\mu^\top D + \gamma (Dz + b)^\top 2\Sigma D \quad (11)$$

e

$$\nabla \phi(z) = \sum_{i=1}^m \frac{1}{-f_i(z)} \nabla f_i(z) \quad (12)$$

A expressão da hessiana será obtida de forma idêntica pelas equações (13), (14) e (15).

$$\nabla^2 f(z) = t(\nabla^2 f_0(z)) + \nabla^2 \phi(z) \quad (13)$$

$$\nabla^2 f_0(z) = \gamma 2(D^\top \Sigma D) \quad (14)$$

$$\nabla^2 \phi(z) = \sum_{i=1}^m \frac{1}{-f_i^2(z)} \nabla f_i(z) (\nabla f_i(z))^\top \quad (15)$$

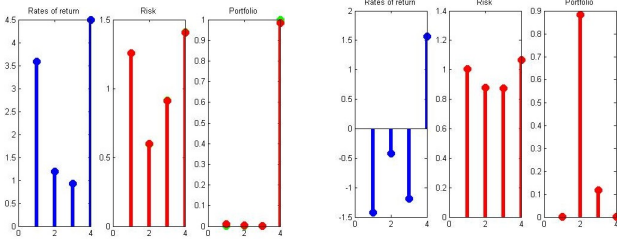
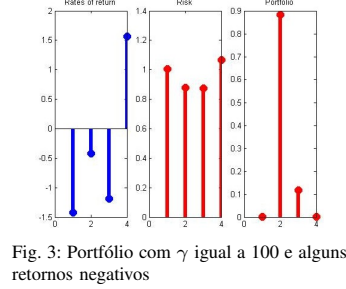
Com esta reformulação é possível aplicar o *Barrier Method* ao problema de *Portfolio Optimization*.

## V. RESULTADOS NUMÉRICOS

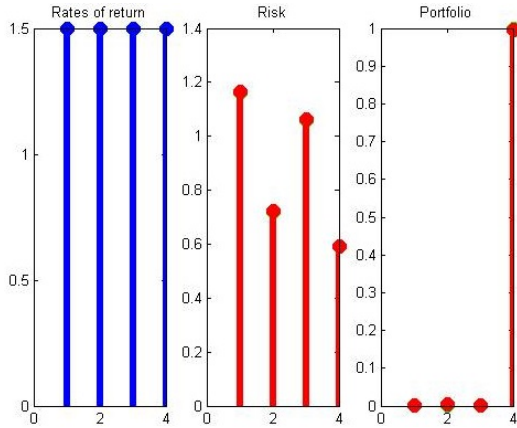
Nas figuras que irão ser expostas tem-se a vermelho os resultados do *Barrier Method* e a verde do *CVX*. Os retornos e os riscos dos *assets* são iguais para ambos os programas. Para verificar o correto funcionamento de ambos os programas realizaram-se testes em que os resultados podem ser facilmente previstos, tais como:

- Igualar  $\gamma$  a zero, ou seja, caso o utilizador não tenha quaisquer preocupações com o risco associado a cada investimento. Espera-se, portanto, que o portfólio obtido seja investir todo o capital no *asset* com maior retorno, independentemente do risco.
- Na figura 2 observa-se que, de facto, o capital foi investido no *asset* de maior retorno. Existem pequenas diferenças entre o portfólio apresentado pelos dois programas, pelo que se pode observar que no resultado do *CVX* o investimento é total no *asset* de maior retorno, enquanto que no resultado do *Barrier Method* existem pequenas quantias investidas pelos outros *assets*.
- Igualar  $\gamma$  a 100 (100 é um valor arbitrário), pretende-se ilustrar uma aversão ao risco muito elevada por parte do utilizador. Espera-se um portfólio bastante conservador, ou seja, que seja investido capital apenas no(s) *asset(s)* com menor risco associado. Quando  $\gamma$  tende para infinito, o portfólio será investido somente nos *assets* com menor risco associado, ao contrário do caso em que  $\gamma$  é igual a 100, onde existem pequenos investimentos nos outros *assets*.

Na figura 3 a solução dada por ambos os programas é em tudo semelhante, note-se que, apesar das *rates of return* dos *assets* de menor risco serem negativas, a aversão ao risco do utilizador irá gerar um portfólio em que a importância dada ao risco é bastante superior à dos retornos. A divisão do portfólio pelos dois *assets* de menor risco é a esperada, ambos os programas escolheram investir mais capital no *asset* com maior retorno.

Fig. 2: Portfólio com  $\gamma$  igual a 0Fig. 3: Portfólio com  $\gamma$  igual a 100 e alguns retornos negativos

- Igualar apenas os *returns* de todos os *assets*. O portfólio deverá investir a grande parte do seu capital no *asset* de menor risco. Quando se aumenta  $\gamma$  nesta situação é esperado que a quantidade investida no *asset* de menor risco aumente.

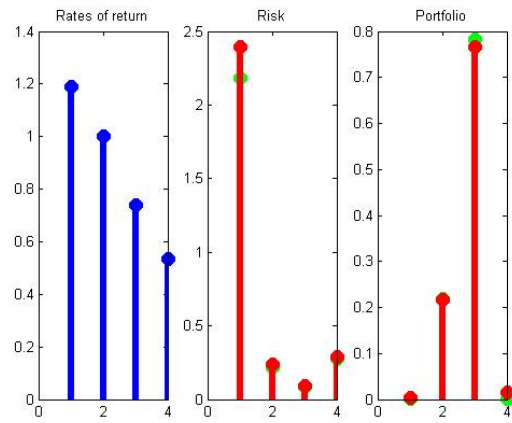
Fig. 4: Portfólio com retornos iguais para todos os *assets*

No caso enunciado acima, a solução de portfólio ótima é intuitiva, os programas deverão investir todo o capital no *asset* de menor risco, visto que todos retornam o mesmo valor do investimento inicial. É isto que se comprova na figura 4.

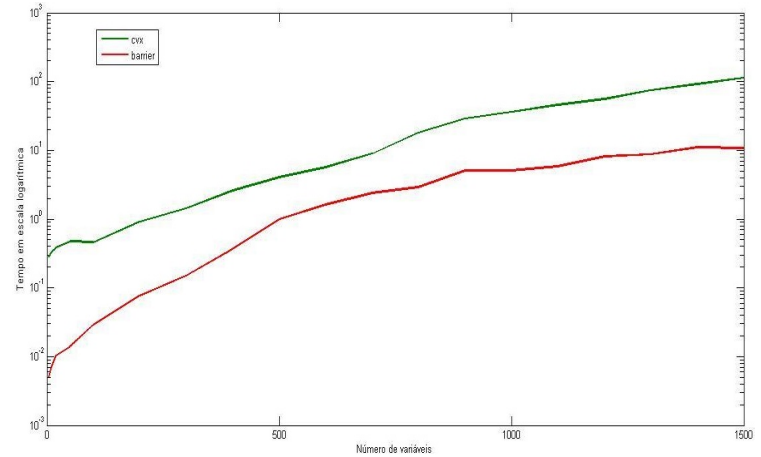
- Correlação entre *assets*. Criou-se uma matriz de covariâncias que apresenta uma grande correlação entre o *asset* um e o *asset* dois. Deste modo pretende-se verificar se o risco de investir num dado *asset* influencia a decisão de investir noutro. Para este caso foi utilizado  $\gamma = 1$ . Na figura 5 pode-se constatar que apesar dos dois primeiros *assets* serem os que têm maiores retornos, o portfólio acaba por dividir os investimentos entre os *assets* dois e três. De notar que os riscos dos *assets* dois, três e quatro não são muito diferentes, pelo que o investimento no *asset* dois seria uma boa solução. Este investimento não é ótimo pois há uma forte correlação entre o um e dois, e portanto, grande parte do investimento é feito em três.

No gráfico da figura 6, é possível observar o tempo requerido para a otimização de uma solução para o problema, consoante a abordagem utilizada, em função do número de variáveis do problema.

Quando o número de variáveis é pequeno, menor que cem, o *Barrier Method* é cerca de cem vezes mais rápido que o

Fig. 5: Influência de um *asset* noutro

CVX. Ao aumentar o seu número, a diferença entre os tempos de otimização diminui, continuando o *Barrier Method* a ser mais rápido (cerca de dez vezes). A partir de cerca de mil variáveis, a diferença entre os tempos volta a ser superior.

Fig. 6: Tempo de resolução do problema pelos dois *solvers*

## VI. CONCLUSÃO

A partir da utilização do *Barrier Method* foi possível aproximar os resultados aos obtidos pelo CVX só que com um tempo bastante inferior. Os testes realizados foram feitos com número de incógnitas variável de forma a obter uma gama de resultados com maior abrangência, sendo obtidos resultados até mil e quinhentas variáveis, número a partir do qual os resultados deixam de ser produzidos em tempo útil. Sendo possível obter pelo *Barrier Method* resultados em menor tempo, este será um método de eleição para a vida real, em que milissegundos são fulcrais na perda/ganho de capital próprio.

## REFERENCES

- [1] [CVX15] <http://cvxr.com/cvx>, Junho 2015
- [2] [BV\_CVXSLIDES] [https://web.stanford.edu/boyd/cvxbook/bv\\_cvxslides.pdf](https://web.stanford.edu/boyd/cvxbook/bv_cvxslides.pdf)