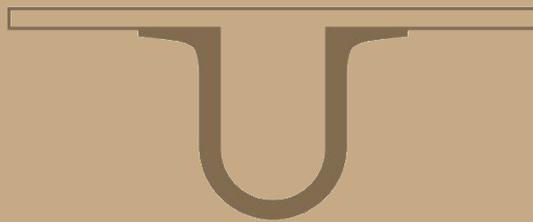




UNIVERSIDADE D
COIMBRA



Paulo Tomás Barbosa Correia Virtudes

OTIMIZAÇÃO ROBUSTA EM FINANÇAS

VOLUME 1

Dissertação no âmbito do Mestrado em Métodos Quantitativos em Finanças, orientada pelo Professor Doutor José Luis Esteves dos Santos e apresentada ao Departamento de Matemática da Faculdade de Ciências e Tecnologia e à Faculdade de Economia.

Março de 2019

Otimização robusta em finanças

Paulo Tomás Barbosa Correia Virtudes



UNIVERSIDADE D
COIMBRA



Mestrado em Métodos Quantitativos em Finanças

Master in Quantitative Methods in Finance

Dissertação de Mestrado | MSc Dissertation

April 2019

Agradecimentos

É muito importante começar esta Dissertação de Mestrado agradecendo a todos quantos, directa ou indirectamente, estiveram comigo ao longo deste percurso, com enorme incentivo e incondicional apoio, sem os quais, este trabalho, jamais se tornaria uma realidade.

Em primeiro lugar, uma palavra especial ao professor José Luis Esteves dos Santos, meu professor e orientador, expresso o meu profundo agradecimento, pela orientação prestada, total apoio, opiniões e críticas, incondicionalmente determinantes à solução de todos os problemas que foram surgindo ao longo deste projeto. Foi um orgulho ser orientado por um excelente professor e ao mesmo tempo espetacular cidadão.

Aos meus pais, pelas valiosas oportunidades que me deram durante o meu percurso académico, bem como todo este pequeno percurso de 23 anos. Pelas inúmeras manifestações de encorajamento, companheirismo e amor, mesmo tendo de abdicar muitas semanas da minha presença.

À minha namorada Catarina Coelho, pelo seu carinho, força e apoio incondicional, uma luz na minha vida e em todos os momentos de maiores dúvidas e incertezas. Um agradecimento especial por toda a paciência, compreensão e ajuda durante o percurso académico, e não só, é pouco. Obrigado, por tudo, do fundo do coração.

A todos os colegas de casa, em especial aos últimos, sem esquecer os primeiros e todos os outros e ao mesmo tempo grandes amigos que a Universidade me deu. Um agradecimento particular por me terem acolhido de uma maneira formidável, fazendo-me sentir em casa. Obrigado a Coimbra por todas as amizades realizadas e por todos os momentos proporcionados.

A todos e a cada um, o meu sincero Muito Obrigado!

Resumo

O propósito deste trabalho é consolidar o tema da otimização robusta. Como já sabemos, a otimização é uma área da matemática que visa encontrar o mínimo ou o máximo de uma função, quando sujeita a restrições. Posto isto, o intuito é realmente encontrar a solução ótima de entre várias a que chamamos de admissíveis. Esta, por sua vez, tenta resolver de forma eficiente problemas do mundo real. Contudo, muitos dos problemas reais possuem incertezas nos diversos parâmetros de entrada, tais como preços, custos, produção, etc. Daí a necessidade do aparecimento da otimização robusta, que é uma forma de tratamento de problemas de otimização com incerteza/dúvida nos parâmetros. Neste trabalho pretende-se fazer uma breve introdução a este tema, considerando vários conjuntos de incerteza e fazendo referência a alguns problemas de otimização em finanças. A par disto, e como já era de esperar, a robustez surge em diferentes circunstâncias que dá origem a vários tipos de robustez existentes, tendo desenvolvido alguns deles nesta tese. Por fim, apresentam-se resultados computacionais que mostram a importância da otimização robusta num contexto prático de seleção robusta de carteiras em múltiplos períodos.

Conteúdo

Lista de Figuras	ix
Lista de Tabelas	xi
1 Introdução	1
2 Otimização Robusta	3
2.1 Conjuntos de incerteza	5
2.2 Tipos de robustez	6
2.2.1 Robustez de restrições	6
2.2.2 Robustez objetiva	9
2.2.3 Robustez relativa	11
2.2.4 Otimização robusta ajustável	12
3 Aplicações em Finanças	15
3.1 Seleção robusta de carteiras em vários períodos	15
3.2 Seleção robusta de carteiras	20
3.3 Robustez relativa na seleção de carteiras	24
4 Resultados computacionais	29
Bibliografia	35
Anexo A	37

Lista de Figuras

2.1	Região admissível (esquerda) e soluções ótimas (direita) do problema (2.1).	4
2.2	Conjunto de incerteza elipsoidal.	6
2.3	Região admissível com incerteza na restrição.	7
3.2	Região de incerteza relativamente a μ	21
3.3	Região que contém o conjunto de incerteza relativamente a Σ (fixando $a = 1$ e ignorando a restrição $\Sigma \succeq 0$).	22
3.4	Região de incerteza do conjunto de incerteza relativamente a Σ (fixando $a = 1$).	22
3.5	Fronteira de Pareto para $\lambda \in \left[\frac{3}{4}, \frac{3}{2} \right]$	23
3.6	Região admissível Ω do problema.	25
3.7	Curva de nível com as restrições do problema.	25
3.8	Carteiras com um nível de arrependimento máximo de 0.9 nos 3 cenários.	27
4.1	Comparação das várias abordagens considerando 1000, 20 000 e 50 000 simulações relativamente ao valor da função objetivo.	34
4.2	Comparação das várias abordagens considerando 1000, 20 000 e 50 000 simulações relativamente à percentagem de simulações que falharam.	34

Lista de Tabelas

4.1	Comparação entre heurísticas numa simulação com 1000 iterações.	32
4.2	Comparação das várias abordagens considerando 1000, 20 000 e 50 000 simulações.	33

Capítulo 1

Introdução

A otimização é uma área da matemática, cujo intuito é de uma maneira geral a minimização ou maximização de uma função, denominada função objetivo. Num problema de otimização, pretendemos procurar a solução ótima de entre várias soluções que chamamos de admissíveis, tendo em conta as várias condicionantes do mesmo (restrições do problema). Quando nos referimos à solução ótima, pensamos na melhor solução possível, isto é, a melhor solução para o problema em questão. Em algumas situações procuramos o máximo, como por exemplo: uma empresa que deseja determinar a quantidade de um certo produto a ser produzido, de modo a maximizar o seu lucro. Por outras pretendemos o mínimo, como no caso da redução de gastos com o transporte de alunos para um certo colégio.

Em muitos problemas de otimização assume-se que os dados que descrevem o problema são conhecidos de forma exata e precisa. No entanto, na prática não é o que acontece. Surge então o conceito de otimização robusta, pois existe a necessidade de trabalhar com incerteza nos dados de modo a encontrar soluções admissíveis. Esta visa tornar o modelo matemático válido para as distintas realizações das variáveis aleatórias. A otimização robusta é baseada na perspectiva de pior caso, isto é, pretendemos encontrar a melhor solução possível na pior das hipóteses. Atendendo a que esses mesmos dados são estabelecidos de uma forma imprecisa, uma vez que não são conhecidos com precisão, criaram-se os conjuntos de incerteza com o objetivo de limitar as variáveis, sendo por isso constituídos por uma diversidade de valores admissíveis.

Estes conjuntos de incertezas englobam todos os cenários distintos pelos quais quem toma as decisões fica, de alguma maneira, protegido contra alguns possíveis contratempos. Para além disso, o conjunto não pode ser escolhido de uma forma qualquer, ou seja, é obrigatório que num problema de otimização todos os cenários possíveis dentro do conjunto respeitem as restrições do mesmo. A título de exemplo, uma fábrica vende jarros em que o lucro da venda de cada jarro é de 3 euros. Cada jarro tem um determinado tempo de mão de obra de 2 horas. Mas, visto tratar-se de tempo e de quem os manuseia é susceptível que pode ter incertezas, isto é, há jarros que podem demorar mais tempo do que o tempo previsto, outros menos. Podemos então ter um conjunto de incerteza para o tempo de mão de obra que consiste, por exemplo, no intervalo de uma hora e meia a duas horas e meia.

Hoje em dia, a otimização robusta é aplicada em inúmeros casos da vida quotidiana, alguns deles mencionados de seguida. Basta pensar num caso em que um doente necessita de um transporte intra-hospitalar, quando o transporte (tempo de deslocação) é enorme. Um simples problema de veículos ou mesmo o problema do vendedor ambulante também são exemplos onde a otimização robusta é utilizada. Estes estão sujeitos a um elevado grau de incerteza no seu trajeto, uma vez que os tempos de cada destino são incertos por diversos motivos, seja por trânsito, acidentes, obras na estrada, etc.

No presente trabalho começamos por desenvolver o tema de otimização robusta no geral, com foco em conjuntos de incerteza e tipos de robustez, fazendo acompanhar de exemplos práticos para uma melhor percepção. Posteriormente, e com a ajuda do capítulo anterior, mencionaremos modelos de otimização robusta aplicada nas finanças, exemplificando-os também. Por último uma secção de resultados computacionais ilustra a vantagem da otimização robusta. Este trabalho foi desenvolvido com base na referência bibliográfica número [1].

Capítulo 2

Otimização Robusta

A otimização robusta modela problemas de otimização em que os dados são sujeitos a incertezas e variam num determinado conjunto incerto. O intuito desta otimização é obter a melhor solução relativa a todas aquelas que estão dentro desse mesmo conjunto incerto, e essa ao mesmo tempo é solução admissível para as distintas ocorrências intrínsecas ao conjunto.

Para motivar o seu estudo, começamos com a ilustração de um exemplo prático, apoiado em [2], onde se pode constatar o que foi mencionado anteriormente. Inicialmente, trata-se de um problema de maximização sem incertezas uma vez que todos os coeficientes são conhecidos com exatidão. Posteriormente trataremos de reformulá-lo, introduzindo incerteza em alguns dos coeficientes, dando origem a um problema de otimização robusta.

Imagine-se uma empresa que desenvolve *software* informático. Para realizar tal tarefa, os seus funcionários precisam de passar por 2 fases. Uma primeira de elaboração e uma segunda de análise de erros e sua correção. Os lucros da empresa são 10 euros caso o programa a desenvolver seja simples ou 30 euros para os mais complexos. A primeira fase demora 4 horas para os programas mais fáceis enquanto para os outros demora 10 horas. Por outro lado, a segunda fase demora 2 e 6 horas respetivamente, para os de menor e maior dificuldade. Os gerentes da empresa sabem que têm 300 horas disponíveis, 190 para a primeira fase e 110 horas para a segunda, ambas semanais.

O problema em causa é saber qual a quantidade de problemas a implementar para maximizar o lucro da empresa semanal.

Seja x_1 a quantidade de problemas mais fáceis a realizar e x_2 a quantidade de problemas com uma dificuldade mais elevada, então podemos formular o problema da seguinte maneira:

$$\begin{aligned} \max \quad & 10x_1 + 30x_2 \\ \text{s.a.} \quad & 4x_1 + 10x_2 \leq 190 \\ & 2x_1 + 6x_2 \leq 110 \\ & x_1, x_2 \geq 0. \end{aligned} \tag{2.1}$$

Este problema pode ser resolvido pelo método gráfico, cujas soluções ótimas estão sob a reta $2x + 6y = 110$, desde o ponto de coordenadas $(0, 18.33)$ até ao ponto de coordenadas $(10, 15)$, como ilustrado na figura abaixo (lado direito). Existe, portanto, um conjunto de soluções ótimas, pois o declive de ambas as retas ($2x + 6y = 110$ e $10x + 30y = 300$) coincidem. Sendo assim, se a empresa desenvolver, por exemplo, 10 problemas simples e 15 com maior grau de dificuldade (ponto de interseção de ambas as restrições), ou 7 e 16, respectivamente, obterá um lucro semanal de 550 euros.

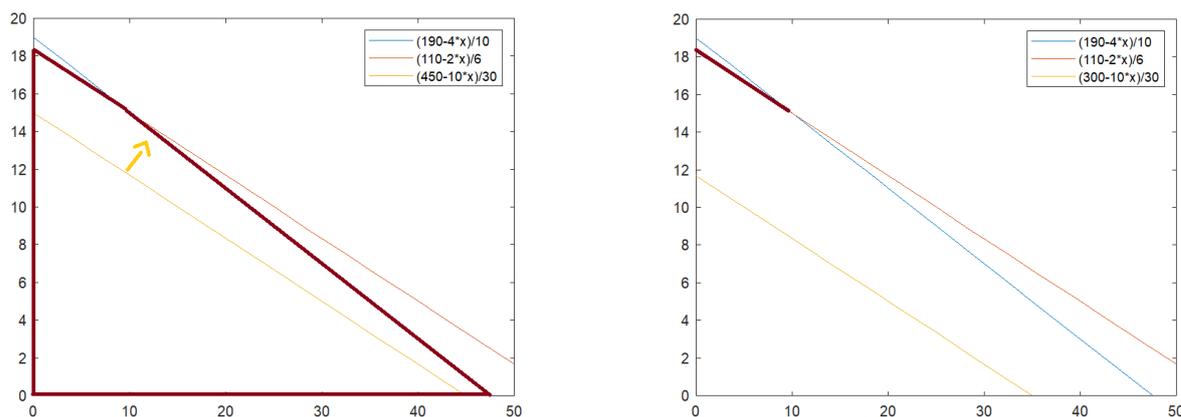


Fig. 2.1 Região admissível (esquerda) e soluções ótimas (direita) do problema (2.1).

Com o mesmo problema, podemos pensar que nem tudo é tão certo quanto isso. Assumir que as restrições estão sujeitas a algumas incertezas não é de todo errado, bem pelo contrário, isto porque não é nenhuma máquina programada que faz o trabalho, mas sim mão humana. Desde logo, e por falarmos em tempos de execução, podemos pensar em atrasos na sua execução, precalços do dia a dia e até o tempo, de pessoa para pessoa, variar a fazer o mesmo tipo de atividade. Por este mesmo motivo sente-se a utilidade e necessidade da construção de incerteza para, por exemplo, a elaboração dos programas mais exigentes, e essa incerteza assentar entre 8 horas e meia e 11 horas ou na fase de correção de erros dos problemas mais acessíveis entre 1 hora e quarenta e cinco minutos e 2 horas e meia.

Fica então dado o mote do problema genérico. Com o problema anterior com incertezas, pretende-se encontrar uma solução que seja robusta, isto é, a melhor solução possível de modo a que a admissibilidade das soluções seja preservada para qualquer realização futura do cenário.

Nas próximas duas secções trataremos de aprofundar dois temas que na otimização robusta são essenciais. O primeiro refere-se aos conjuntos de incerteza, que permitem alguma permeabilidade nos dados uma vez que não há a exatidão no seu conhecimento e um segundo tema referente aos tipos de robustez. Em ambos os casos desenvolvemos alguns dos existentes e, em cada um destes, aprofundamos e mencionamos exemplos práticos.

2.1 Conjuntos de incerteza

Os conjuntos de incerteza são originados pela falta de precisão que os dados acarretam. Urge, realmente, a necessidade de trabalhar com incerteza nos dados na perspectiva de obter melhores soluções, mais próximas da realidade. Esta incerteza, vem do facto de se lidar com problemas que se baseiam, por exemplo, com tempo, custos, peso, etc. Pertence-nos então, ver e analisar, até que ponto a solução primordial (sem incertezas) é útil e boa o suficiente na realidade comparada com a que tem incertezas.

Os conjuntos de incerteza formam-se na grande maioria dos casos por haver opiniões distintas sobre os valores futuros de certos parâmetros, como os que foram mencionados acima. Em aplicações do mundo real é quase impossível ter dados precisos para os coeficientes de entrada dos problemas. Portanto, e para atingir resultados mais confiáveis e credíveis, é conveniente considerar as oscilações que podem existir nos dados do problema. Os conjuntos de incerteza que abordamos neste trabalho são: discretos, convexos, intervalar e elipsoidais.

O primeiro a ser discutido é o caso em que o conjunto de incerteza é de forma discreta, representado por: $U = \{p_1, p_2, \dots, p_k\}$. Este, possui um número finito de possibilidades para o valor do parâmetro em questão. Para exemplificar este caso, voltamos ao exemplo base do programa linear (2.1). Ora, visto que maioria dos programas difíceis conseguem ser desenvolvidos em 10 horas, mas outros podem demorar mais (por exemplo, 11 horas) ou menos (por exemplo, 9 horas) tempo, construímos um conjunto de incerteza discreto da seguinte forma: $U = \{9, 10, 11\}$. Posto isto, já existe uma margem de manobra para o tempo necessário a realizar os programas com maior complexidade.

Um outro tipo de conjunto de incerteza é o caso convexo, $U = \text{conv}\{p_1, p_2, \dots, p_k\}$. Por sua vez, e como o nome indica, simbolizam o caso em que existe uma combinação convexa de um número finito de condições para os possíveis valores dos parâmetros. Este tipo de conjunto pode ser, por exemplo: $U = \text{conv}\{9, 10, 11\} = [9, 11]$, que vem no seguimento do caso anterior. Neste exemplo, o conjunto U reduziu-se a um intervalo mas numa situação mais geral, tal pode não acontecer. A título de exemplo, considere-se que cada cenário p é um par de valores representando, respectivamente o tempo de desenvolvimento de um programa fácil e outro difícil. Neste caso, se p_1 , p_2 e p_3 não são colineares, então consequentemente não obteremos um intervalo.

Existem também os conjuntos de incerteza sob a forma de intervalo. Por exemplo, os intervalos de confiança das estatísticas podem ser considerados deste tipo. Desta vez, exibem-se do seguinte modo: $U = \{p : l \leq p \leq u\}$. Por exemplo, e mais uma vez por tratar-se de uma duração, o tempo que demora a corrigir um problema simples do exemplo inicial, é de 2 horas. Se optarmos por uma otimização robusta e quisermos um conjunto de incerteza desta forma, poderíamos preferir este tempo entre 1 hora e quarenta e cinco minutos e duas horas e meia. Assim, $U = \{p : 1.75 \leq p \leq 2.5\}$.

Por último, apresenta-se o tipo de conjunto de incerteza elipsoidal, $U = \{p : p = p_0 + Mu, \|u\| \leq 1\}$. Note-se que neste conjunto, as entradas da matriz M que possuem incertezas são não nulas. Este

tipo de intervalos podem surgir no contexto dos intervalos de confiança da estatística.

$$\text{Seja então } p = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 10 \\ 2 \end{bmatrix}, M = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \text{ e } u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

$$\text{Assim, } \begin{cases} p_1 = 10 + 2u_1 \\ p_2 = 2 + u_2 \end{cases} = \begin{cases} u_1 = \frac{p_1 - 10}{2} \\ u_2 = p_2 - 2 \end{cases}. \text{ Por fim, substituindo na condição da norma,}$$

$\|u\| \leq 1$, obtemos o conjunto elipsoidal da seguinte forma: $\left(\frac{p_1 - 10}{2}\right)^2 + \left(\frac{p_2 - 2}{1}\right)^2 \leq 1$, representado na figura seguinte.

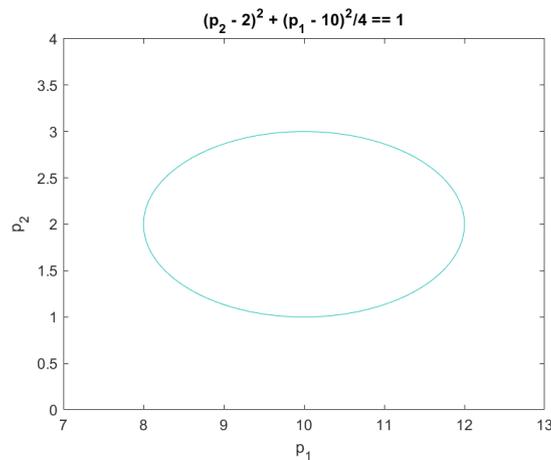


Fig. 2.2 Conjunto de incerteza elipsoidal.

Como referido, estes são alguns tipos dos conjuntos de incerteza existentes. Note-se que perante a sua diversidade, é difícil escolher qual o melhor que se ajusta ao modelo em questão. Esta seleção depende não só das fontes de incerteza, bem como da sensibilidade das soluções para essa mesma incerteza. Posto isto, aparece ainda outra adversidade, a definição do tamanho do conjunto incerto. Este, por sua vez, é maior ou menor, consoante o nível de robustez (proteção) desejado.

2.2 Tipos de robustez

Nesta secção, o principal objetivo é dar a conhecer diversos tipos de robustez existentes, nomeadamente a de restrições, objetiva, relativa e a ajustável, bem como estudar cada um deles com algum detalhe. Para os perceber melhor serão sempre acompanhados com exemplos práticos que permitem uma compreensão mais facilitada dos problemas.

2.2.1 Robustez de restrições

A robustez de restrições é um conceito importante na otimização de problemas robustos. Esta, e como o nome nos indica, refere-se a situações em que as incertezas estão presentes nas restrições do

problema a ser tratado.

Voltamos ao exemplo que foi dado (problema de *software*) para exemplificar o conjunto de incertezas discreto, em que era formado pelo seguinte conjunto de incertezas: $U = \{9, 10, 11\}$. Este é um caso em que a incerteza está presente nas restrições e, portanto, um exemplo da robustez que estamos a abordar. Neste caso, pretendemos que a solução obtida seja admissível para todas as possíveis realizações dos cenários, pelo que a primeira restrição do problema (2.1) será substituída por restrições correspondendo a cada uma das concretizações do parâmetro p . O problema primordial transforma-se e passa então a ser um problema com quatro restrições em vez de duas (exceptuando as de não-negatividade), como era inicialmente (2.1), ou seja:

$$\begin{aligned} \max \quad & 10x_1 + 30x_2 \\ \text{s.a.} \quad & 4x_1 + px_2 \leq 190, p \in \{9, 10, 11\} \\ & 2x_1 + 6x_2 \leq 110 \\ & x_1, x_2 \geq 0, \end{aligned}$$

que por sua vez dá origem ao seguinte problema:

$$\begin{aligned} \max \quad & 10x_1 + 30x_2 \\ \text{s.a.} \quad & 4x_1 + 9x_2 \leq 190 \\ & 4x_1 + 10x_2 \leq 190 \\ & 4x_1 + 11x_2 \leq 190 \\ & 2x_1 + 6x_2 \leq 110 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Neste exemplo é fácil constatar que a restrição com $p = 11$ torna as duas primeiras restrições redundantes, como se pode observar no seguinte gráfico, onde podemos ver a sua região admissível.

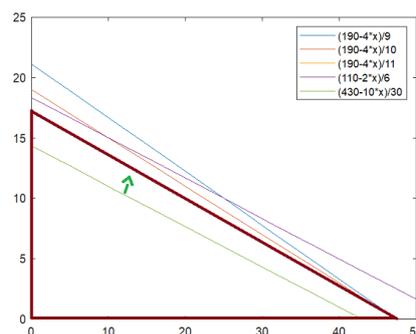


Fig. 2.3 Região admissível com incerteza na restrição.

A solução ótima deste problema é a interseção da reta $4x + 11y = 190$ com a reta $x = 0$, ou seja, o ponto ótimo tem de coordenadas $(0, 17.2727)$. Concluimos assim que o lucro semanal será de 518.181 euros, em detrimento dos 550 euros iniciais. Esta redução no lucro semanal já era de esperar, uma vez que foram introduzidos mais parâmetros que limitam o problema a nível de maximização. Isto quer dizer que, na pior das hipóteses, ou seja, no mínimo a empresa tem um lucro semanal de 518.181

euros. Caso não houvesse a incerteza colocada no cenário $p = 11$, a solução já seria sob a reta roxa ($2x + 6y = 110$), igual à solução inicial de 550 euros.

Generalizando, consideremos um problema de otimização da seguinte forma:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.a.} \quad & G(x, p) \in K, \end{aligned}$$

onde $f(x)$ é a função objetivo, x são as variáveis de decisão, G e K os elementos estruturais das decisões e p os parâmetros incertos do problema. Consideramos agora um conjunto de incerteza U , que contém todos os valores possíveis para o parâmetros incertos p . Para encontrarmos a solução ótima quando a incerteza surge nas restrições, resolvemos o seguinte problema:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.a.} \quad & G(x, p) \in K, \forall p \in U. \end{aligned}$$

Reparamos que no problema genérico anterior a incerteza apenas constava nas restrições. Porém, pode acontecer que surja ao mesmo tempo nas restrições e na função objetivo e, caso aconteça, o problema é também resolúvel. Neste caso transformamos o problema que temos num outro em que a incerteza está apenas nas restrições. A partir daí já é conhecida a sua forma de resolução.

Posto isto, consideremos o seguinte problema de otimização em que as incertezas estão presentes tanto na função objetivo como nas restrições:

$$\begin{aligned} \min_x \quad & f(x, p) \\ \text{s.a.} \quad & G(x, p) \in K. \end{aligned}$$

Podemos reformular este problema acrescentando uma nova variável. Assim a incerteza está apenas nas restrições e, deste modo, ambos os problemas são equivalentes.

$$\begin{aligned} \min_{t,x} \quad & t \\ \text{s.a.} \quad & G(x, p) \in K \\ & t - f(x, p) \geq 0. \end{aligned}$$

Vejamos com um exemplo concreto o que acontece na prática. Retomando novamente o exemplo dos programas computacionais. Admitamos que o lucro de cada programa mais complexo, p_1 , pode variar no conjunto de incerteza $\{30, 31, 32\}$. Ao invés de termos o problema linear com a dúvida somente nas restrições, agora passamos a ter também na função objetiva, ou seja, o seguinte:

$$\begin{aligned} \max \quad & 10x_1 + p_1x_2, \quad p_1 \in \{30, 31, 32\} \\ \text{s.a.} \quad & 4x_1 + px_2 \leq 190, \quad p \in \{9, 10, 11\} \\ & 2x_1 + 6x_2 \leq 110 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Posteriormente a esta análise, reformulamos este problema num equivalente com a adição da nova variável do seguinte modo:

$$\begin{aligned}
& \max \quad t \\
& \text{s.a.} \quad 4x_1 + px_2 \leq 190, p \in \{9, 10, 11\} \\
& \quad \quad 2x_1 + 6x_2 \leq 110 \\
& \quad \quad t - 10x_1 - p_1x_2 \leq 0, p_1 \in \{30, 31, 32\} \\
& \quad \quad x_1, x_2 \geq 0.
\end{aligned}$$

Com este último problema temos o que desejamos, isto é, com todas as incertezas presentes apenas e só nas restrições. Assim, fica facilitada a resolução do seguinte problema linear,

$$\begin{aligned}
& \max \quad t \\
& \text{s.a.} \quad 4x_1 + 9x_2 \leq 190 \\
& \quad \quad 4x_1 + 10x_2 \leq 190 \\
& \quad \quad 4x_1 + 11x_2 \leq 190 \\
& \quad \quad 2x_1 + 6x_2 \leq 110 \\
& \quad \quad t - 10x_1 - 30x_2 \leq 0 \\
& \quad \quad t - 10x_1 - 31x_2 \leq 0 \\
& \quad \quad t - 10x_1 - 32x_2 \leq 0 \\
& \quad \quad x_1, x_2 \geq 0.
\end{aligned}$$

A sua solução final é obtida através do *Matlab* e é dada da seguinte forma: $(0, 17.2727, 518.1818)$, com valor na função objetivo de 518.1818 euros semanais de lucro. De facto, note-se que as duas primeiras restrições são redundantes assim como as duas últimas.

2.2.2 Robustez objetiva

A robustez objetiva é outro tipo de robustez que vai ser abordado no presente trabalho. Este tipo de robustez é caracterizada pela incerteza na sua função objetivo que leva à perturbação das soluções ótimas, já que as restrições são fixas. Quando nos referimos a robustez objetiva pensamos nas soluções que estão próximas da solução ótima para todas as possíveis realizações dos parâmetros incertos do problema. Porém, por vezes o conjunto de incerteza pode ser muito grande e a tarefa para obter a solução torna-se mais difícil.

Posto isto, surge a necessidade de arranjar uma alternativa para este caso. É então implementada uma procura de soluções de pior caso, ou seja, onde o pior caso é otimizado. Pretendemos encontrar o valor da função objetivo para a pior realização possível dos parâmetros incertos.

Consideremos o seguinte problema de otimização:

$$\begin{aligned}
& \min_x \quad f(x, p) \\
& \text{s.a.} \quad x \in S.
\end{aligned}$$

Neste problema, S é o conjunto admissível e f a função objetivo que depende dos parâmetros incertos p . Novamente designamos por U o conjunto de incertezas que contém todos os valores possíveis dos parâmetros incertos p .

Assim, o problema de robustez objetiva passa por maximizar todas as funções que contém incertezas e posteriormente obter o seu mínimo, isto é:

$$\begin{aligned} \min_x \quad & h(x) \\ \text{s.a.} \quad & x \in \mathcal{S}, \end{aligned}$$

onde $h(x) = \max_{p \in U} f(x, p)$.

Embora o desenvolvimento teórico tenha sido feito para um problema de minimização, o mesmo raciocínio pode ser aplicado a um problema de maximização, com as devidas alterações. Nesse caso, em vez de um problema do tipo mínimo-máximo, iremos obter um problema do tipo máximo-mínimo.

Por sua vez, a robustez objetiva pode ser vista como um caso especial da robustez vista anteriormente (2.2.1), por meio de uma reformulação. Contudo, importa diferenciar duas classes distintas, tendo por base as formulações iniciais do problema: problemas de otimização semi-infinito e problemas de mínimo-máximo. De acordo com o explicado anteriormente, os problemas de otimização mínimo-máximo são conhecidos por procurar maximizar o mínimo de todas as opções disponíveis, ou o inverso. Por outro lado, os problemas de otimização semi-infinito caracterizam-se por serem indexados por um parâmetro m pertencente a um conjunto contínuo B . Por este motivo, a formulação robusta pode ter infinitas restrições, como o caso do problema que se segue:

$$\begin{aligned} \min \quad & c^m x \\ \text{s.a.} \quad & a(m)^m x \geq b(m), \forall m \in B, \end{aligned}$$

onde B é um conjunto compacto (fechado e limitado).

Regressando ao problema primordial do corrente trabalho, seja agora a incerteza instalada no coeficiente associada ao custo de desenvolvimento de um programa complexo, em que os possíveis valores são 23, 27 e 32, isto é:

$$\begin{aligned} \max \quad & 10x_1 + px_2, p \in \{23, 27, 32\} \\ \text{s.a.} \quad & 4x_1 + 10x_2 \leq 190 \\ & 2x_1 + 6x_2 \leq 110 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Assim sendo, o problema de robustez objetiva é dado por:

$$\begin{aligned} \max \quad & \min_{p \in \{23, 27, 32\}} 10x_1 + px_2 \\ \text{s.a.} \quad & 4x_1 + 10x_2 \leq 190 \\ & 2x_1 + 6x_2 \leq 110 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Para resolvermos este problema, voltamos à região admissível do problema (2.1). Nesta, vemos que os vértices dessa mesma região vão ser os pontos ótimos dos diversos problemas. Esses pontos são: $x_1 = (0, 18.3)$, $x_2 = (10, 15)$ e $x_3 = (47.5, 0)$. x_1 , x_2 e x_3 são, respectivamente as soluções ótimas

dos problemas com $p = 32$, $p = 27$ e $p = 23$. Portanto, os seus valores objetivos são, mais uma vez, respectivamente, 586.66, 505 e 475, sendo que o mínimo é atingido quando $p = 23$. Concluimos assim que o máximo robusto deste problema é então o ponto de coordenadas $(47.5, 0)$ com valor objetivo 475 euros.

2.2.3 Robustez relativa

Para motivar o tema referente à robustez relativa consideremos o seguinte caso que ocorre no dia-a-dia. Em muitas situações as pessoas concorrem entre si para ver qual o melhor entre determinado grupo, quer seja na faculdade, trabalho, ou até mesmo para ganhar a corrida por um subida na carreira. Pensemos agora num grupo de gestores de carteiras. O objetivo destes, por sua vez, é desempenhar um bom papel na sua função. Por outro lado vão sempre evitar fazer um pior trabalho do que os seus concorrentes diretos. Mas, em condições adversas, algo pode não correr como os gestores esperavam. Então, têm de dar o melhor de si mesmo para, que mesmo assim, não sejam os piores. É esta a ideia da robustez relativa, optar pelo pior caso de maneira relativa, em relação à melhor solução possível de cada cenário.

Consideremos o seguinte problema, onde p é um parâmetro incerto dentro do conjunto de incertezas:

$$\begin{aligned} \min_x \quad & f(x, p) \\ \text{s.a.} \quad & x \in \mathcal{S}. \end{aligned}$$

Dado um p fixo, denotamos o valor ótimo do problema anterior por $z^*(p)$ e o respectivo conjunto de soluções por $x^*(p)$, que é a solução do seguinte problema:

$$\begin{aligned} \operatorname{argmin}_x \quad & f(x, p) \\ \text{s.a.} \quad & x \in \mathcal{S}. \end{aligned}$$

Deste modo, é possível interligar ambas as soluções dos dois problemas apresentados anteriormente concluindo que $z^*(p) = f(x^*(p), p)$.

De seguida vamos estabelecer uma medida de arrependimento associada a uma decisão após a incerteza ser resolvida. Caso x seja a escolha para o vetor e p o valor realizado do parâmetro incerto, a escolha de x em vez de $x^*(p)$ é definida como uma função de arrependimento. Essa é considerada como uma medida de benefício de retrospectiva (visão *aposteriori*) e é dada por:

$$r(x, p) = f(x, p) - z^*(p) = f(x, p) - f(x^*(p), p) \geq 0.$$

Posteriormente a conhecermos a função de arrependimento, precisamos de analisar a sua função máxima, conhecida por a função máxima de arrependimento, que nos é fornecida por:

$$R(x) = \max_{p \in U} r(x, p) = \max_{p \in U} f(x, p) - z^*(p).$$

Assim sendo, uma solução robusta relativa do problema inicial, no qual queríamos minimizar $f(x, p)$ com a única restrição $x \in S$, é o vetor x que minimiza o arrependimento máximo:

$$\min_{x \in S} \max_{p \in U} f(x, p) - z^*(p). \quad (2.2)$$

Reparemos que $z^*(p)$ é o valor ótimo de uma função, pelo que já envolve um problema de otimização. Então o problema (2.2) é um problema de otimização de 3 níveis ao invés de 2 níveis como os problemas apresentados na secção (2.2.2). Para além de ser de 3 níveis, $z^*(p)$ muito dificilmente está disponível na forma analítica e é comum ser não suave e difícil de analisar. Outra característica das formulações robustas relativas é o facto de que se f for linear em p , e acontece como muita regularidade, então $z^*(p)$ é côncava. Daqui surge que, o problema de maximização interno é um problema de maximização convexo, que é difícil para grande parte dos U . Concluímos então que, embora atraentes, as formulações robustas relativas podem ser difíceis.

Por esse mesmo motivo, podemos construir uma alternativa mais simples para estes casos, limitando o nível máximo de arrependimento, por exemplo a um dado R . Ficamos com o seguinte problema: pretendemos encontrar um x que satisfaça $x \in S$, tal que

$$r(x, p) \leq R, \forall p \in U \Leftrightarrow f(x, p) - z^*(p) \leq R, \forall p \in U.$$

Podemos ter uma segunda alternativa ao primeiro modelo. Esta, pode ser útil para, por exemplo, modelos de períodos múltiplos em que as revisões de decisões entre períodos podem sair caras. Problemas de rebalanceamento de carteira com custos de transação são exemplos destas configurações. Considere-se então $d(x, p)$ a medida de arrependimento em termos de aproximação da nossa solução escolhida para o conjunto de soluções ótimas em vez de $r(x, p)$ que mede o arrependimento em termos dos valores objetivos ótimos. Logo,

$$d(x, p) = \inf_{x^* \in x^*(p)} \|x - x^*\|.$$

Como anteriormente feito, podemos considerar a função de distância máxima:

$$D(x) := \max_{p \in U} d(x, p) = \max_{p \in U} \inf_{x^* \in x^*(p)} \|x - x^*\|.$$

Neste novo sentido, para a robustez relativa, procuramos x que

$$\min_{x \in S} \max_{p \in U} d(x, p).$$

2.2.4 Otimização robusta ajustável

Todas as distintas formulações de otimização abordadas anteriormente pressupõe que os parâmetros incertos não são observados até que todas as variáveis de decisão sejam determinadas. Apesar disso, o mesmo não acontece para todos os cenários. Para o caso dos modelos de decisão de períodos múltiplos, alguns dos parâmetros incertos são descobertos ao longo do processo de decisão. As formulações de otimização robustas ajustáveis (ARO) modificam o ambiente de decisão e permitem ação de recurso, sendo que nos períodos múltiplos, posteriormente ao conhecimento dos parâmetros, as decisões no

período seguinte podem responder a novas informações.

Os modelos ARO foram recentemente introduzidos para problemas de programação linear incertos. Nesta secção vamos introduzir este conceito considerando um problema de otimização linear de dois períodos, onde x^1 são as variáveis de decisão de primeiro período e precisam de ser determinadas agora, enquanto que x^2 são as variáveis de segundo período e podem ser escolhidas após os parâmetros incertos do problema A^1 , A^2 e b serem realizados:

$$\begin{aligned} \min_{x^1, x^2} \quad & c^T x^1 \\ \text{s.a.} \quad & A^1 x^1 + A^2 x^2 \leq b. \end{aligned}$$

Notamos de imediato que as variáveis do segundo período, x^2 , não aparecem na função objetivo. A esses problemas, chamamos a forma normalizada do problema. Ainda assim, como visto anteriormente, pode acontecer que x^2 apareça na função objetivo. Nesse caso a introdução de uma variável artificial, de modo análogo ao realizado na secção 2.2.1, permite reformulá-lo na forma padrão. Sem perda de generalidade, concentrar-nos-emos nesta forma mais simples e conveniente.

Vamos denotar novamente por U o conjunto de incertezas, agora para os parâmetros A^1 , A^2 e b . A restrição da formulação padrão do problema de otimização robusta para este problema procura encontrar os vetores x^1 e x^2 que otimizam a função objetivo e satisfazem as restrições do problema para todas as possíveis realizações dos coeficientes da restrição. Nessa formulação os dois conjuntos de variáveis devem ser escolhidos antes que os parâmetros possam ser observados, e portanto, não podem depender deles. Consequentemente, a contraparte robusta desse problema pode ser escrita da seguinte maneira:

$$\min_{x^1} \{c^T x^1 : \exists x^2 \forall (A^1, A^2, b) \in U : A^1 x^1 + A^2 x^2 \leq b\}.$$

Note-se que a formulação seguinte é equivalente à anterior, mas preferimos a anterior, uma vez que é mais notória a diferença entre esta formulação e a versão ajustável.

$$\min_{x^1, x^2} \{c^T x^1 : A^1 x^1 + A^2 x^2 \leq b, \forall (A^1, A^2, b) \in U\}.$$

Em contraste, a formulação de otimização robusta ajustável permite a escolha das variáveis x^2 (segundo período) para depender dos valores realizados dos parâmetros incertos. O problema da contraparte robusto ajustável é dado da seguinte forma:

$$\min_{x^1} \{c^T x^1 : \forall (A^1, A^2, b) \in U, \exists x^2 \equiv (A^1, A^2, b) : A^1 x^1 + A^2 x^2 \leq b\}.$$

O conjunto admissível do segundo problema é maior do que o do primeiro em geral, e, portanto, o modelo é mais flexível. A maior dificuldade das formulações ARO resultantes parece ser o "preço a pagar" pela flexibilidade de modelagem adicional. Mesmo nos problemas em que a contraparte robusta é tratável, pode acontecer que a formulação ARO nos leve a um problema NP-difícil. Um problema NP-difícil enquadra-se na teoria da complexidade computacional, tratando-se de uma classe de problemas que são "pelo menos tão difíceis" quanto os problemas mais difíceis em NP, que é a

sigla para tempo polinomial não determinístico.

A grande dificuldade dos modelos ARO é o facto do conjunto viável das ações de recurso (decisões do segundo período) depender não só da realização dos parâmetros incertos, mas também das decisões do primeiro período. Podemos ultrapassar essa dificuldade considerando a simplificação de premissas seja no conjunto de incertezas e/ou na estrutura de dependência de ações de recurso para parâmetros incertos. Por exemplo, as ações de recurso serem restritas a funções afins dos parâmetros incertos. Esta pode ser a única estratégia para formulações tratáveis, apesar desta restrição provavelmente nos dar soluções abaixo do ideal.

Capítulo 3

Aplicações em Finanças

Como já se constatou, a grande maioria dos problemas do quotidiano possuem incertezas nos seus coeficientes, e as formulações de otimização robustas são uma forma de abordar estes problemas. É de notar que outra forma de abordagem destes problemas é a otimização estocástica que não se considera no presente trabalho.

Nesta secção vamos dar exemplos de formulações de otimização robustas para problemas de otimização no âmbito financeiro, que envolvem, por exemplo, valores futuros de preços de título, taxas de juro, taxas de câmbio, que de uma maneira não são conhecidos mas podem ser previstos e estimados. Portanto, estes problemas encaixam na perfeição na otimização robusta.

Inicialmente daremos exemplos de formulações robustas para vários problemas de otimização financeira. Vamos iniciar com um problema de seleção robusta de carteiras em vários períodos.

3.1 Seleção robusta de carteiras em vários períodos

Consideremos um investidor que pretende investir o seu dinheiro num conjunto de n ativos em L períodos de tempo. O intuito dele é maximizar a sua riqueza no final do último período L , sendo que, para isso, necessita de ajustar a sua carteira nos períodos que se avizinham.

Para modelar um problema deste tipo usamos as variáveis de decisão b_i^l e s_i^l para denotar, respectivamente, o número de ações do ativo i compradas no início do período l e o número de ações do ativo i vendidas no início do período l , $i = 1, \dots, n$, $l = 1, \dots, L$. Assumimos que as unidades dos ativos são divisíveis (importa realçar a ordem de grandeza envolvida), caso contrário fica com um problema de programação inteira difícil de resolver.

Desde já, conseguimos saber o número de ações do ativo i que o investidor possui na carteira no início do período l (x_i^l). Este é dado pela diferença entre o número de ações do ativo i compradas e vendidas, sem esquecer de adicionar a quantidade que transita do período anterior. Portanto, vamos denotar este cálculo por x_i^l e é dado pela seguinte equação:

$$x_i^l = x_i^{l-1} - s_i^l + b_i^l, \quad i = 1, \dots, n, \quad l = 1, \dots, L.$$

Sabemos, porém, que estas compras e vendas estão sujeitas a custos de transação. Apesar disso, vamos assumir que estas estão integradas nas compras, β_i^l , e nas vendas, α_i^l , para o ativo i , no período l e vamos assumir que estes dois valores são conhecidos no princípio do período 0 (embora possam variar de período para período e de ativo para ativo). Seja P_i^l o preço do ativo i no período l e x_0^0 o capital monetário do investidor. Vamos admitir ainda que $P_0^l = 1 \forall l$, uma vez que esse dinheiro não vai ser investido. Sendo assim, surge a seguinte equação para o dinheiro que o investidor tem disponível no início do período l :

$$x_0^l = x_0^{l-1} + \sum_{i=1}^n (1 - \alpha_i) P_i^l s_i^l - \sum_{i=1}^n (1 + \beta_i) P_i^l b_i^l, \quad l = 1, \dots, L. \quad (3.1)$$

Esta última equação traduz a soma do que vem do período anterior e as vendas feitas em l , descontados pelo custo de transação, com a diferença do que foi comprado, acrescentando ao custo de transação, isto é, o saldo monetário no início do período l .

Com o intuito de facilitar a procura de uma solução inicial, para posteriormente convergir até à ótima, através de métodos iterativos, podemos aumentar a região admissível que verifica a desigualdade estrita. Assim, podemos reescrever (3.1) como:

$$x_0^l \leq x_0^{l-1} + \sum_{i=1}^n (1 - \alpha_i) P_i^l s_i^l - \sum_{i=1}^n (1 + \beta_i) P_i^l b_i^l, \quad l = 1, \dots, L.$$

Vejamos que esta alteração no sinal da restrição não afeta realmente o problema, dado que pretende-se maximizar o montante disponível no último nível (L) e portanto a solução ótima deverá verificar a restrição como igualdade. Notamos que as restrições com o saldo monetário do investidor vão ser satisfeitas com igualdade em qualquer solução ótima, pois ele pretende maximizar o seu lucro, e por isso, não quer perder parte do seu dinheiro. Traduzindo pela notação que foi mencionada temos:

$$\max \sum_{i=0}^n P_i^L x_i^L.$$

Ao assumirmos que todos os preços futuros do P_i^l são conhecidos no momento em que esse problema de investimento deve ser resolvido, obtemos o seguinte problema de otimização:

$$\begin{aligned} \max_{x,b,s} \quad & \sum_{i=0}^n P_i^L x_i^L \\ \text{s.a.} \quad & x_0^l \leq x_0^{l-1} + \sum_{i=1}^n (1 - \alpha_i) P_i^l s_i^l - \sum_{i=1}^n (1 + \beta_i) P_i^l b_i^l, \quad l = 1, \dots, L \\ & x_i^l = x_i^{l-1} - s_i^l + b_i^l, \quad i = 1, \dots, n, \quad l = 1, \dots, L \\ & s_i^l \geq 0, \quad i = 1, \dots, n, \quad l = 1, \dots, L \\ & b_i^l \geq 0, \quad i = 1, \dots, n, \quad l = 1, \dots, L \\ & x_i^l \geq 0, \quad i = 0, \dots, n, \quad l = 1, \dots, L \end{aligned} \quad (3.2)$$

Este problema é facilmente resolvido com, por exemplo, o método simplex. As restrições de não negatividade servem para não fazer empréstimos (x_0^l positivo, para ter sempre capital disponível) nem proceder com vendas a descoberto ($x_i^l, i = 1, \dots, n$, evita vender o que não é do próprio). Estas, não são essenciais para o modelo e podem ser retiradas (excepto $l = L$) para permitir as vendas a descoberto ou o empréstimo. Apesar de tudo isto, o problema (3.2) não se enquadra de todo num cenário realista, dado que normalmente os valores dos preços futuros P_i^l não são conhecidos *a priori*.

Num cenário realista, não conhecemos os P_i^l com antecedência, e por esse mesmo motivo não podemos resolver o problema de alocação de carteira ideal da maneira que desenvolvemos anteriormente. Temos, portanto de criar um novo problema de otimização robusto com incerteza nesse mesmo parâmetro.

Como visto na secção 2.2.1, e mais uma vez usando a mesma técnica, introduzimos uma nova variável, de modo a que toda a incerteza esteja somente presente nas restrições, isto é:

$$\begin{aligned}
 & \max_{x,b,s,t} \quad t \\
 \text{s.a.} \quad & t \leq \sum_{i=0}^n P_i^L x_i^L, & l = 1, \dots, L \\
 & x_i^l \leq x_0^{l-1} + \sum_{i=1}^n (1 - \alpha_i) P_i^l s_i^l - \sum_{i=1}^n (1 + \beta_i) P_i^l b_i^l, & l = 1, \dots, L \\
 & x_i^l = x_i^{l-1} - s_i^l + b_i^l, i = 1, \dots, n & l = 1, \dots, L \\
 & s_i^l \geq 0, i = 1, \dots, n, & l = 1, \dots, L \\
 & b_i^l \geq 0, i = 1, \dots, n, & l = 1, \dots, L \\
 & x_i^l \geq 0, i = 1, \dots, n, & l = 1, \dots, L
 \end{aligned} \tag{3.3}$$

O problema (3.3) tem todas as incertezas nas suas restrições. Aliás, as duas primeiras restrições são as que possuem os P_i^l que desconhecemos e, desta forma gostaríamos de encontrar uma solução que satisfaça estas restrições para as diversas realizações dos parâmetros incertos.

Vamos seguir a abordagem $3 - \sigma$ (onde σ representa o desvio padrão) para a escolha de um conjunto de incertezas apropriado aos parâmetros incertos para a posterior determinação da versão robusta destas restrições. Esta técnica admite que os P_i seguem uma distribuição normal e considera intervalos para os conjuntos de incerteza centrados na média e uma amplitude igual a 3σ .

Podemos considerar que os preços vão evoluir de uma forma que não conhecemos e, por esse mesmo motivo, serem quantidades aleatórias. Sejam $P^l = [P_1^l \dots P_n^l]^T$ os preços e o seu valor esperado $\mu^l = [\mu_1^l \dots \mu_n^l]^T$. Notemos que $\mu_0^l = 1$, pois admitimos que o capital monetário não é investido. Por sua vez, vamos denotar a matriz de covariâncias por V^l .

Vamos focar-nos na primeira restrição do problema (3.3): $t \leq \sum_{i=0}^n P_i^L x_i^L$. Fixando $x^L = (x_1^L, \dots, x_n^L)$, vamos calcular a esperança de $\sum_{i=0}^n P_i^L x_i^L$ e respectivo desvio padrão.

$$E\left(\sum_{i=0}^n P_i^L x_i^L\right) = \sum_{i=0}^n E(P_i^L x_i^L) = \sum_{i=0}^n x_i^L E(P_i^L) = \sum_{i=0}^n x_i^L \mu_i^L = x_0^L + \sum_{i=1}^n x_i^L \mu_i^L;$$

$$\begin{aligned} V\left(\sum_{i=0}^n P_i^L x_i^L\right) &= \text{cov}\left(\sum_{i=0}^n P_i^L x_i^L, \sum_{j=0}^n P_j^L x_j^L\right) = \sum_{i=0}^n \sum_{j=0}^n \text{cov}(P_i^L x_i^L, P_j^L x_j^L) = \sum_{i=0}^n \sum_{j=0}^n x_i^L x_j^L \text{cov}(P_i^L, P_j^L) \\ &= \sum_{i=0}^n \sum_{j=0}^n (x_i^L)^T V_{i,j}^L x_j^L = (x^L)^T V^L x^L. \end{aligned}$$

Notemos que na última igualdade, o caso $i, j = 0$ desaparece pois P_0^L é uma constante e ao calcular a covariância anula-se com a sua média (o mesmo valor).

$$\text{Pretendendo o desvio padrão, este é dado por } STD\left(\sum_{i=0}^n P_i^L x_i^L\right) = \sqrt{V\left(\sum_{i=0}^n P_i^L x_i^L\right)} = \sqrt{(x^L)^T V^L x^L}.$$

A versão robusta da primeira restrição, satisfeita por cerca de 99,87% é então dada por:

$$t \leq E\left(\sum_{i=0}^n P_i^L x_i^L\right) - 3STD\left(\sum_{i=0}^n P_i^L x_i^L\right) = x_0^L + (\mu^L)^T x^L - 3\sqrt{(x^L)^T V^L x^L}.$$

Vamos proceder de forma análoga para a segunda restrição, última com incertezas associadas. Primeiramente, isolaremos os termos incertos no segundo membro da desigualdade e posteriormente calculamos a média e desvio padrão dos mesmos.

$$x_0^L - x_0^{L-1} \leq \sum_{i=1}^n (1 - \alpha_i) P_i^L s_i^L - \sum_{i=1}^n (1 + \beta_i) P_i^L b_i^L. \quad (3.4)$$

$$\begin{aligned} E\left(\sum_{i=1}^n (1 - \alpha_i) P_i^L s_i^L - \sum_{i=1}^n (1 + \beta_i) P_i^L b_i^L\right) &= \sum_{i=1}^n (1 - \alpha_i) s_i^L E(P_i^L) - \sum_{i=1}^n (1 + \beta_i) b_i^L E(P_i^L) \\ &= \sum_{i=1}^n (1 - \alpha_i) s_i^L \mu_i^L - \sum_{i=1}^n (1 + \beta_i) b_i^L \mu_i^L \\ &= (\mu^L)^T D_\alpha s^L - (\mu^L)^T D_\beta b^L \\ &= (\mu^L)^T [D_\alpha \quad -D_\beta] [s^L \quad b^L]^T. \end{aligned}$$

Para ser mais fácil a notação, no cálculo da variância vamos denotar $\sum_{i=1}^n (1 - \alpha_i) P_i^L s_i^L$ pela letra a , $\sum_{i=1}^n (1 + \beta_i) P_i^L b_i^L$ pela letra b , $\sum_{j=1}^n (1 - \alpha_j) P_j^L s_j^L$ por c e $\sum_{j=1}^n (1 + \beta_j) P_j^L b_j^L$ por d . Então vejamos:

$$\begin{aligned}
V\left(\sum_{i=1}^n (1 - \alpha_i) P_i^l s_i^l - \sum_{i=1}^n (1 + \beta_i) P_i^l b_i^l\right) &= \text{cov}\left(\sum_{i=1}^n (1 - \alpha_i) P_i^l s_i^l - \sum_{i=1}^n (1 + \beta_i) P_i^l b_i^l, \sum_{j=1}^n (1 - \alpha_j) P_j^l s_j^l - \sum_{j=1}^n (1 + \beta_j) P_j^l b_j^l\right) \\
&= \text{cov}(a - b, c - d) = \text{cov}(a - b, c) - \text{cov}(a - b, d) \\
&= \text{cov}(a, c) - \text{cov}(b, c) - \text{cov}(a, d) + \text{cov}(b, d).
\end{aligned}$$

Ora, vamos desenvolver o primeiro termo, sendo os seguintes análogos.

$$\begin{aligned}
\text{cov}(a, c) &= \text{cov}\left(\sum_{i=1}^n (1 - \alpha_i) P_i^l s_i^l, \sum_{j=1}^n (1 - \alpha_j) P_j^l s_j^l\right) = \sum_{i=1}^n \sum_{j=1}^n (1 - \alpha_i) s_i^l (1 - \alpha_j) s_j^l \text{cov}(P_i^l, P_j^l) \\
&= (s^L)^T (D_\alpha) V^L (D_\alpha) (s^L);
\end{aligned}$$

$$\text{cov}(b, c) = (b^L)^T (D_\beta) V^L (D_\alpha) (s^L);$$

$$\text{cov}(a, d) = (s^L)^T (D_\alpha) V^L (D_\beta) (b^L);$$

$$\text{cov}(b, d) = (b^L)^T (D_\beta) V^L (D_\beta) (b^L).$$

Referente ainda ao que em cima foi desenvolvido, $s^l = (s_1^l, \dots, s_n^l)^T$ e $b^l = (b_1^l, \dots, b_n^l)^T$ e D_α e D_β são matrizes diagonais da seguinte forma:

$$D_\alpha = \begin{bmatrix} (1 - \alpha_1) & & \\ & \ddots & \\ & & (1 - \alpha_n) \end{bmatrix} \quad D_\beta = \begin{bmatrix} (1 + \beta_1) & & \\ & \ddots & \\ & & (1 + \beta_n) \end{bmatrix}.$$

Substituindo na restrição inicial (3.4), temos:

$$\begin{aligned}
x_0^l - x_0^{l-1} &\leq E\left(\sum_{i=1}^n (1 - \alpha_i) P_i^l s_i^l - \sum_{i=1}^n (1 + \beta_i) P_i^l b_i^l\right) - 3STD\left(\sum_{i=1}^n (1 - \alpha_i) P_i^l s_i^l - \sum_{i=1}^n (1 + \beta_i) P_i^l b_i^l\right) \\
&\leq (\mu^l)^T [D_\alpha - D_\beta] [s^l \ b^l]^T - 3\sqrt{[(s^l)^T \ (b^l)^T] [D_\alpha - D_\beta] V^L [D_\alpha - D_\beta]^T [(s^l)^T \ (b^l)^T]^T}.
\end{aligned}$$

Obtemos assim uma versão robusta da segunda restrição do problema (3.3). Assumindo normalidade nos parâmetros incertos, ao satisfazer esta restrição podemos garantir que a restrição inicial é satisfeita para 99,87% das realizações dos parâmetros.

Como constatado nesta secção, o objetivo até então passa por maximizar o lucro sem ter em conta o risco da carteira associado. Na secção seguinte, pretendemos considerar estes dois critérios em simultâneo, ou seja, não só ter em atenção o lucro, bem como o risco da carteira.

3.2 Seleção robusta de carteiras

O problema de otimização de média variância de Markowitz estabelece quais as decisões a ser tomadas no investimento com base na relação do risco com o retorno. Este problema tem o intuito de diversificar a carteira e consequência disso é a redução do risco da mesma. A função objetivo, neste problema, é:

$$\max \mu^T x - \lambda x^T \Sigma x,$$

onde μ_i é a média do retorno esperado no título i , σ_{ii} a variância desse retorno, σ_{ij} a covariância dos títulos i e j e λ uma constante não negativa de aversão ao risco usada para ponderar o retorno e o risco.

Resolvendo este problema para vários valores de λ obtemos a fronteira eficiente do conjunto de carteiras admissíveis. Há, portanto, um conjunto infinito de carteiras que pode ser escolhido, dependendo do perfil de risco do investidor. Este, por ser avesso ao risco, prefere carteiras com menor risco e, sendo assim, com menor retorno ou, por outro lado, um investidor atraído pelo risco prefere carteiras com um maior risco, podendo dar origem a maiores retornos.

Este modelo tem a limitação da necessidade de estimação dos retornos e covariâncias. Se estes forem usados com valores incertos a carteira eficiente escolhida pode não ter o comportamento previsto/desejado. Além disso, a solução ótima é sensível a perturbações nesses parâmetros, ou seja, pequenas alterações na estimativa do retorno leva, por vezes, a grandes mudanças na solução. Mais, os resultados podem ser pouco intuitivos e levar a carteiras ineficientes. Para rebalancear a carteira, com base em novos dados, poderá ter de pagar custos de transação significativos, já que pequenas alterações nos dados podem impôr elevadas mudanças nas posições. Para além do mais, o uso de estimativas pontuais, de retorno e covariância, podem não corresponder a necessidades de investidores conservadores que não confiem nessas mesmas estimativas e prefiram carteiras que tenham um melhor desempenho perante cenários distintos. Neste caso, terão menor lucro em todos os cenários, mas certamente protege-se contra carteiras de um elevado risco.

Todos estes argumentos apontam para a necessidade de uma formulação robusta para este tipo de problemas de otimização, com o mesmo objetivo, tentar encontrar uma solução que seja pouco sensível a imprecisões nos dados de entrada.

Na otimização robusta de carteiras, consideramos um modelo em que o conjunto de incerteza para o retorno e covariância são fornecidos em forma de intervalo. Por exemplo, esta informação pode assumir a forma: "o retorno esperado de j é entre 8% e 10%" ao invés de afirmar que é 9%.

Matematicamente, representamos essas informações com o seguinte conjunto:

$$U = \{(\mu, \Sigma) : \mu^L \leq \mu \leq \mu^U, \Sigma^L \leq \Sigma \leq \Sigma^U, \Sigma \succeq 0\}, \quad (3.5)$$

onde $\mu^L, \mu^U, \Sigma^L, \Sigma^U$ são os valores extremos dos intervalos mencionados acima. A última restrição indica que Σ tem de ser uma matriz simétrica e semi-definida positiva, pois trata-se de uma matriz de covariância.

Por fim, pretendemos encontrar uma carteira que maximize o modelo de Markowitz na pior das hipóteses dos parâmetros μ e Σ , tendo em conta o conjunto de incerteza. Perante estas situações, o problema de otimização robusto assume a seguinte forma:

$$\max_{x \in X} \{ \min_{(\mu, \Sigma) \in U} \mu^T x - \lambda x^T \Sigma x \}.$$

Quando o conjunto de incerteza é dado pela equação 3.5, consegue-se mostrar o seguinte resultado:

Proposição 1 *No caso especial em que Σ^U é semi-definida positiva, $\mu^* = \mu^L$ e $\Sigma^* = \Sigma^U$ constitui uma solução ótima do problema $\{ \min_{(\mu, \Sigma) \in U} \mu^T x - \lambda x^T \Sigma x \}$. Neste caso, a solução ótima do problema robusto é dada pela solução de $\max (\mu^L)^T x - \lambda x^T \Sigma^U x$.*

Ilustramos de seguida com um exemplo prático. Sejam $\mu^L = [2 \ 5]^T$ e $\mu^U = [3 \ 8]^T$ o limite superior e inferior para o vetor μ e $\Sigma^L = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$ e $\Sigma^U = \begin{bmatrix} 1 & 3 \\ 3 & 4 \end{bmatrix}$ o limite inferior e superior para a matriz de covariâncias (ambos dados incertos do problema). Notemos que ambas as matrizes Σ definidas anteriormente são simétricas mas Σ^U não é semi-definida positiva. Deste modo, não podemos utilizar a proposição anterior sem redefinir a matriz Σ^U de forma a que verifique $\Sigma^U \succeq 0$ sem alterar o conjunto de incerteza U .

Uma vez que ambos os parâmetros atuam de forma independente no conjunto U , começamos por retratar ambos os conjuntos de incerteza associados a este mesmo exemplo. O primeiro, verifica-se para $2 \leq u_1 \leq 3, 5 \leq u_2 \leq 8$ como podemos verificar na figura abaixo.

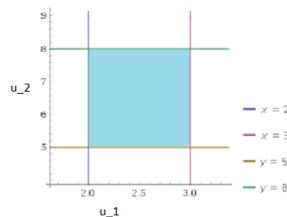


Fig. 3.2 Região de incerteza relativamente a μ .

Por sua vez, a matriz Σ como já vimos tem de ser numa primeira instância simétrica. Portanto, vai ser do seguinte modo:

$$\Sigma = \begin{bmatrix} a & b \\ b & c \end{bmatrix},$$

$$\text{em que } \begin{cases} 1 \leq a \leq 1; \\ 1 \leq b \leq 3; \\ 2 \leq c \leq 4. \end{cases}$$

Deste modo, podemos fixar $a = 1$ e os restantes parâmetros produzem uma região da mesma forma que o anterior em que varia o parâmetro b e c , ou seja,

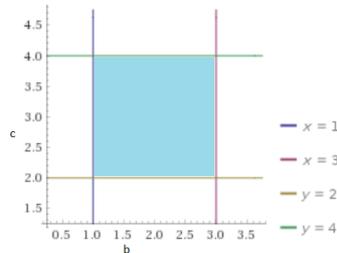


Fig. 3.3 Região que contém o conjunto de incerteza relativamente a Σ (fixando $a = 1$ e ignorando a restrição $\Sigma \succeq 0$).

Além disso, temos de garantir que a matriz também é semi-definida positiva, ou seja, $\begin{cases} a \geq 0; \\ c \geq 0; \\ ac - b^2 \geq 0. \end{cases}$

A primeira e segunda condições são verificadas visto que $a = 1$ e $2 \leq c \leq 4$. A última condição, $c \geq b^2$ tem de ser adicionada ao conjunto de incerteza para a matriz Σ . Deste modo surge a região apresentada abaixo em que podemos verificar que o máximo valor de b e c é 2 e 4, respectivamente.

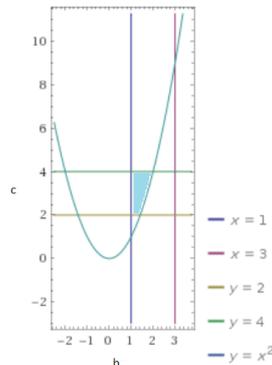


Fig. 3.4 Região de incerteza do conjunto de incerteza relativamente a Σ (fixando $a = 1$).

A matriz Σ^U pode ser substituída por $\Sigma^{U'} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$ sem alterar o conjunto de incerteza.

Aplicando a Proposição 1, temos o seguinte problema para resolver:

$$\begin{aligned} \max \quad & (\mu^L)^T x - \lambda x^T \Sigma^{U'} x \\ \text{s.a.} \quad & x_1 + x_2 = 1. \\ & x_1, x_2 \geq 0 \end{aligned}$$

A restrição de igualdade permite reduzir o problema a apenas uma variável fazendo $x_2 = 1 - x_1$. Assim, substituindo pelos dados do exemplo obtemos:

$$\begin{aligned} \max \quad & 2x_1 + 5x_2 - \lambda(x_1^2 + 4x_1x_2 + 4x_2^2) \\ \text{s.a.} \quad & x_1 + x_2 = 1. \\ & x_1, x_2 \geq 0 \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \max \quad & 5 - 3x_1 - \lambda(x_1^2 - 4x_1 + 4) \\ \text{s.a.} \quad & 0 \leq x_1 \leq 1. \end{aligned}$$

Ao derivar a função a maximizar ($f(x_1) = 5 - 3x_1 - \lambda(x_1^2 - 4x_1 + 4)$) e igualando-a a zero, temos a expressão geral do vértice da parábola, isto é,

$$f'(x_1) = 0 \Leftrightarrow -3 - \lambda(2x_1 - 4) \Leftrightarrow x_1 = -\frac{3}{2\lambda} + 2, \lambda \neq 0.$$

Este ponto corresponde ao maximizante de f uma vez que a função é côncava. Ora, como vimos anteriormente x_1 varia entre zero e um. Então, para λ compreendido entre $\frac{3}{4}$ e $\frac{3}{2}$ a solução é dada por $\left(-\frac{3}{2\lambda} + 2, \frac{3}{2\lambda} - 1\right)$.

Caso $\lambda < \frac{3}{4}$, o vértice da parábola tem abscissa negativa pelo que o maximizante de f no intervalo $[0, 1]$ é $x_1 = 0$. Por outro lado, se $\lambda > \frac{3}{2}$, o vértice da parábola será seguramente com um maximizante maior que 1 ($x_1 > 1$). Neste caso, o maximizante no intervalo $[0, 1]$ será $x_1 = 1$.

Para exemplificar um caso, escolhemos $\lambda = 1$. A solução é então dada por $(x_1, x_2) = \left(\frac{1}{2}, \frac{1}{2}\right)$, o seu retorno é $\frac{7}{2}$ e o seu risco $\frac{9}{4}$.

Na figura abaixo apresenta-se a fronteira de Pareto obtida com $\lambda \in \left[\frac{3}{4}, \frac{3}{2}\right]$, apresentando no eixo das abscissas o risco e no eixo das ordenadas o retorno da carteira ótima.

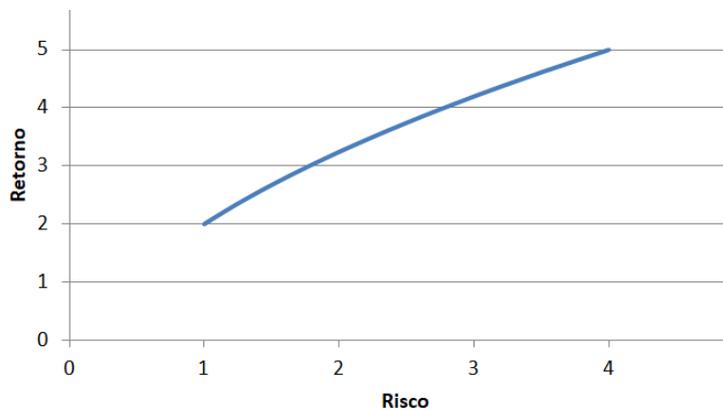


Fig. 3.5 Fronteira de Pareto para $\lambda \in \left[\frac{3}{4}, \frac{3}{2}\right]$.

3.3 Robustez relativa na seleção de carteiras

Consideremos de seguida um exemplo com um problema de otimização de carteira de 3 ativos, em que o terceiro, x_3 , representa a proporção dos fundos que não são investidos nos dois primeiros. Por sua vez, os ativos x_1 e x_2 têm um desvio padrão de 35% e 28%, respectivamente, sendo que o seu coeficiente de correlação é 0.65. A carteira de referência, x_{ref} , em que se investe metade em cada um dos dois primeiros ativos, isto é 50%, em cada. Pretende-se analisar carteiras que não se afastem muito de x_{ref} , pelo que é necessário considerar uma função $TE(x)$ que mede o erro de acompanhamento como a medida de variabilidade do excesso de retorno, ou seja, como o desvio padrão do excesso de retorno: $TE(x) = \sqrt{(x - x_{ref})^T Q (x - x_{ref})}$. Neste caso, Q é uma matriz simétrica e semi-definida positiva, ou seja, satisfaz $Q = Q^T$ e $x^T Q x \geq 0$, para todo o $x \in \mathbb{R}^n$. Posto isto, o objetivo é determinar uma carteira cujo retorno seja o maior possível em relação ao retorno de referência e cujo o erro de acompanhamento seja inferior a uma quantidade $TE > 0$ (ou seja, $TE(x) \leq TE$), que no caso se assume $TE = 0.22\%$. O problema explicitado anteriormente passa então por resolver:

$$\begin{aligned} \max \quad & \mu_1 x_1 + \mu_2 x_2 + \mu_3 x_3 \\ \text{s.a.} \quad & TE(x_1, x_2, x_3) \leq 0.0022 \\ & x_1 + x_2 + x_3 = 1 \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

Para a sua resolução vamos assumir que a estimativa da matriz de covariâncias, Q , é certa e dada por

$$Q = \begin{bmatrix} 0.1225 & 0.0637 & 0 \\ 0.0637 & 0.0784 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

onde

$$Q_{11} = Cov(x_1, x_1) = V(x_1) = 0.35^2 = 0.1225;$$

$$Q_{12} = Q_{21} = Cov(x_1, x_2) = 0.65 \times 0.35 \times 0.28 = 0.0637;$$

$$Q_{22} = Cov(x_2, x_2) = V(x_2) = 0.28^2 = 0.0784.$$

A entrada da matriz Q_{33} é nula visto que a variância do terceiro ativo também o é. Por outro lado, as restantes posições da matriz Q são nulas, uma vez que a correlação entre o ativo 3 e os restantes é nula.

Por outro lado, $(x - x_{ref})^T$ é dado por $(x - x_{ref})^T = [x_1 - 0.5 \quad x_2 - 0.5 \quad x_3]$.

O problema em causa é, portanto:

$$\begin{aligned} \max \quad & \mu_1 x_1 + \mu_2 x_2 + \mu_3 x_3 \\ \text{s.a.} \quad & \sqrt{\begin{bmatrix} x_1 - 0.5 & x_2 - 0.5 & x_3 \end{bmatrix} \begin{bmatrix} 0.1225 & 0.0637 & 0 \\ 0.0637 & 0.0784 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 - 0.5 \\ x_2 - 0.5 \\ x_3 \end{bmatrix}} \leq 0.0022 \\ & x_1 + x_2 + x_3 = 1 \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

O problema pode ser reduzido apenas a duas variáveis, x_1 e x_2 . Deste modo a restrição que contém $x_3 \geq 0$, passa a ser $x_1 + x_2 \leq 1$. A região admissível do problema encontra-se dentro da elipse delimitada pela equação $TE(x) \leq 0.0022$ e abaixo da reta $x_1 + x_2 = 1$, isto é, o conjunto Ω que pode ser visualizado na seguinte figura:

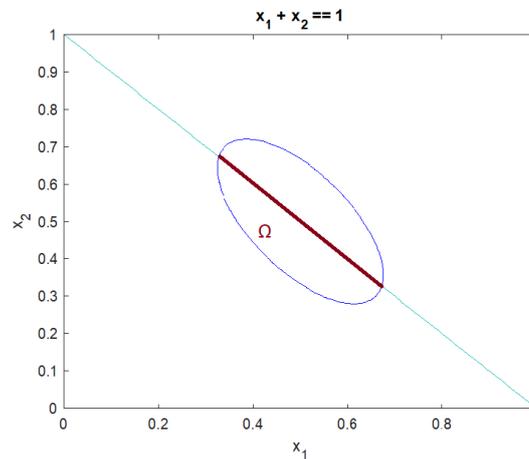


Fig. 3.6 Região admissível Ω do problema.

Para determinar a solução ótima deste problema, precisamos de indicar o valor do retorno para cada um dos ativos. Deste modo, consideremos 3 cenários distintos: $\mu^1 = (7, 3, 0)$, $\mu^2 = (3, 7, 0)$ e $\mu^3 = (5, 5, 0)$.

Vamos construir um modelo de robustez relativa para resolver este problema, sabendo que Q é certa, mas com incerteza na estimativa do retorno esperado com os 3 cenários distintos já apresentados.

Para o primeiro caso, isto é $\mu^1 = (7, 3, 0)$, temos a seguinte figura, onde se pode ver uma curva de nível da função objetivo e ao mesmo tempo as restrições do problema.

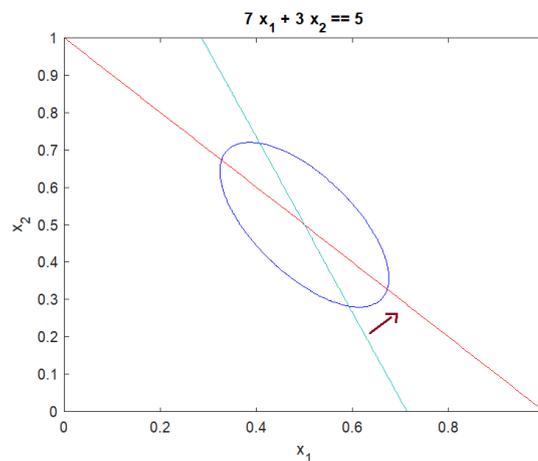


Fig. 3.7 Curva de nível com as restrições do problema.

Verificamos que a solução do problema é na interseção mais à direita da elipse com a reta $x_1 + x_2 = 1$ (a vermelho), quando x_1 é aproximadamente $x_1 = 0.7$. Portanto, a solução ótima é $(x_1, x_2) = (0.673, 0.327)$ e, sendo assim, tem um valor objetivo de 5.692.

Do mesmo modo, para $\mu^2 = (3, 7, 0)$, a solução ótima é $(x, y) = (0.327, 0.673)$, com valor objetivo novamente de 5.692.

Para o último cenário, $\mu^3 = (5, 5, 0)$, todas as soluções admissíveis são ótimas sendo o valor da função objetivo igual a 5.

Deste modo, a solução ótima (e o seu valor) são muito influenciados pela concretização dos valores do retorno numa situação concreta. Portanto, é preferível utilizar um modelo robusto e para esse efeito iremos considerar o modelo de robustez relativa. Neste caso, pretende-se minimizar o arrependimento (diferença entre o valor ótimo e o valor obtido numa dada solução, para cada um dos cenários), sendo necessário resolver o seguinte problema:

$$\begin{aligned}
 & \min_{x,t} \quad t \\
 \text{s.a.} \quad & 5.692 - (7x_1 + 3x_2) \leq t \\
 & 5.692 - (3x_1 + 7x_2) \leq t \\
 & 5 - (5x_1 + 5x_2) \leq t \\
 & TE(x_1, x_2, x_3) \leq 0.0022 \\
 & x_1 + x_2 + x_3 = 1 \\
 & x_1, x_2, x_3 \geq 0.
 \end{aligned}$$

Recorrendo ao *Matlab*, encontramos a seguinte solução ótima deste problema $x = (0.5, 0.5, 0)$ e $t = 0.692$.

Por vezes, pode não ser necessário determinar a carteira que produz o arrependimento mínimo, sendo apenas suficiente encontrar uma solução que esteja abaixo de um certo nível de arrependimento. Nesse caso, o problema torna-se mais simples uma vez que a variável t fica fixa. A título de exemplo, se considerarmos o nível máximo de arrependimento igual a 0.9, obtemos o seguinte problema:

$$\begin{aligned}
 & \text{Encontrar } x \\
 \text{s.a.} \quad & 5.9336 - (7x_1 + 3x_2) \leq 0.9 \\
 & 5.9336 - (3x_1 + 7x_2) \leq 0.9 \\
 & 5 - (5x_1 + 5x_2) \leq 0.9 \\
 & TE(x_1, x_2, x_3) \leq 0.0022 \\
 & x_1 + x_2 + x_3 = 1 \\
 & x_1, x_2, x_3 \geq 0.
 \end{aligned}$$

A figura que se segue mostra as carteiras que têm um nível de arrependimento de 0.9 (ou menos) em todos os 3 cenários considerados, que é o triângulo sombreado a verde.

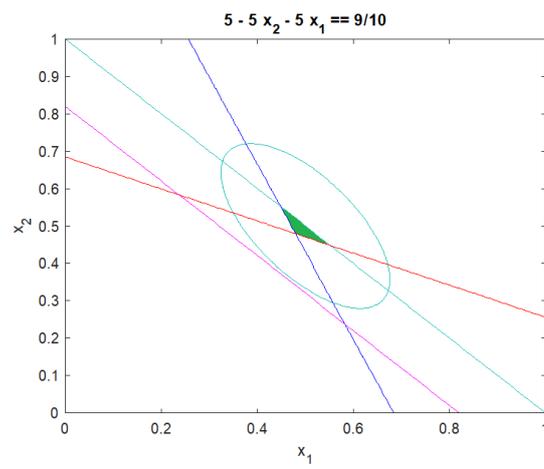


Fig. 3.8 Carteiras com um nível de arrependimento máximo de 0.9 nos 3 cenários.

Capítulo 4

Resultados computacionais

Nesta secção foi posto em prática a aplicação do capítulo 3.1, cujo tema é a seleção robusta de carteiras em vários períodos.

Numa primeira abordagem foi realizado um programa em *Matlab* para a resolução da versão determinística (problema (3.2)) considerando como *input* os valores médios dos preços dos ativos. Para tal usamos a função *linprog*, uma vez que o problema em questão é linear. Notemos que a função *linprog* é predefinida para a minimização, sendo necessário implementar o simétrico da função objetivo apresentada no problema (3.2). De um modo geral, pretendemos maximizar a função objetivo sujeito a um sistema de desigualdade $Ay \leq c$, bem como um sistema de igualdade $Aeq y = beq$. O intuito deste programa é encontrar a solução ótima y^* .

A par disto, este programa necessita ainda de vários parâmetros de entrada para ser executado, sendo eles:

- *saldo0*, o saldo inicial que o investidor tem em mão;
- x_0 , a quantidade de ações de cada ativo que o investidor possui no instante inicial;
- P , o preço de cada ativo em cada instante temporal;
- α , o custo de transação das vendas de cada ativo;
- β , o custo de transação das compras de cada ativo.

Neste caso, o vetor y é um vetor de 4 blocos, isto é

$$y = [\textit{saldo}^T \quad x^T \quad b^T \quad s^T]^T,$$

onde

- *saldo* é um vetor que indica o saldo que o investidor possui nos diversos instantes temporais;
- x é um vetor que representa a quantidade de ações de cada ativo em cada instante temporal;

- b e s são vetores que estão associados, respectivamente, à quantidade de ações compradas ou vendidas de cada ativo nos diferentes instantes temporais.

Portanto, a matriz A vai ser também constituída pelos mesmos 4 blocos da seguinte forma:

$$A = [A_{saldo} \ A_x \ A_b \ -A_s].$$

Assim sendo, sabendo que o vetor c é composto pelo $saldo0$ no primeiro elemento e todas as restantes entradas do vetor são nulas, o sistema de desigualdade $Ay \leq c$ fica concluído.

Por outro lado, a matriz Aeq vai ser também constituída por 4 blocos, isto é

$$Aeq = [Aeq_{saldo} \ Aeq_x \ -Aeq_b \ Aeq_s]$$

e beq é um vetor em que as primeiras n componentes (número de ativos) correspondem ao $x0$ e as restantes são nulas.

Ora, o programa *Aplicacao.m* (ver anexo), implementado em *Matlab*, visa retornar a melhor solução y^* que podemos obter, devolvendo o saldo que o investidor fica após realizar as transações, o número de unidades que são compradas ou vendidas de cada ativo para os diferentes instantes temporais, bem como o número de ações de cada ativo atualizado.

Posto isto, apresentamos de seguida um exemplo prático, correndo o programa explicitado anteriormente para 2 ativos distintos e 3 instantes temporais, com os seguintes parâmetros de entrada:

$$saldo0 = 1000 \text{ u.m.}, x0 = [50; 50], P = \begin{bmatrix} 10 & 20 & 30 \\ 15 & 10 & 20 \end{bmatrix}, \alpha = [0.1 \ 0.2] \text{ e } \beta = [0.2 \ 0.05].$$

Assim, a solução ótima deste exemplo prático é a seguinte:

$$\begin{aligned} saldo &= [0 \ 0 \ 0]; \\ x &= \begin{bmatrix} 183.3333 & 0 & 0 \\ 0 & 314.2857 & 314.2857 \end{bmatrix}; \\ s &= \begin{bmatrix} 133.3333 & 0 & 0 \\ 0 & 314.2857 & 0 \end{bmatrix}; \\ s &= \begin{bmatrix} 0 & 183.3333 & 0 \\ 50 & 0 & 0 \end{bmatrix}. \end{aligned}$$

Em análise a este *output*, podemos verificar que, para maximizar o lucro no último instante temporal ($L = 3$), devemos investir tudo no primeiro ativo dado que este vai valorizar mais no segundo instante temporal, em detrimento do que se sucede com o segundo ativo (desvalorização). Deste modo, são vendidas todas as unidades iniciais do segundo ativo (isto é 50 ações) e compradas $\frac{400}{3} = 133.3333$ unidades do primeiro ativo, tendo esgotado o saldo existente. Atendendo ao valor baixo do segundo ativo no segundo instante temporal e a sua elevada valorização no último instante

(duplicação do seu valor), o investimento ótimo consistirá em vender todas as unidades do primeiro ativo e investi-las no segundo. No último instante temporal não é feita nenhuma mudança na carteira, visto que existem custos de transação.

Sendo assim, dada a função objetivo $\sum_{i=0}^2 P_i^3 x_i^3$, o valor ótimo deste problema prático é de 6285.7 u.m. . Este número é o capital do investidor, que é referente apenas ao valor do ativo 2 que este possui, visto que o seu saldo é nulo no último instante temporal. Ora, iniciando com um capital de 1000 u.m. , podemos verificar que houve um aumento bastante significativo de 5285.7 u.m. , sendo esse o seu lucro.

Após a implementação da versão determinística, implementamos também o programa *solucaoRobusta.m* (ver anexo) em *Matlab* para a solução robusta. A estrutura deste programa é semelhante à da versão determinística, porém utilizamos agora a função *fmincon*. Recorremos a esta função uma vez que na versão robusta passamos a ter restrições de desigualdade não lineares, pela envolvimento do desvio padrão nas mesmas.

O objetivo deste capítulo é tentar mostrar numericamente a vantagem das soluções robustas. Desse modo, a solução ótima da versão determinística foi comparada com 3 soluções robustas utilizando, respectivamente a estratégia $1 - \sigma$, $2 - \sigma$ e $3 - \sigma$ as quais irão considerar, respectivamente 68%, 95% e 99.7% dos valores possíveis para os preços dos ativos. Finalmente, estas soluções serão comparadas com soluções heurísticas clássicas. Assim, as versões comparadas são as seguintes:

optDeterminística: Solução ótima do problema linear determinístico, isto é, aplicação do problema (3.2);

optRobusto1: Solução robusta segundo o problema (3.3) com coeficiente $\sigma = 1$;

optRobusto2: Solução robusta segundo o problema (3.3) com coeficiente $\sigma = 2$;

optRobusto3: Solução robusta segundo o problema (3.3) com coeficiente $\sigma = 3$;

Heurística 1: Solução do instante inicial mantida ao longo do tempo;

Heurística 2: Solução onde o saldo inicial é utilizado para comprar o mesmo número de unidades de cada um dos ativos, mantendo-a posteriormente ao longo do tempo;

Heurística 3: Solução onde o saldo inicial é utilizado para investir a mesma quantidade em cada um dos ativos, mantendo-a posteriormente ao longo do tempo.

Tendo obtidas as soluções de cada uma destas abordagens, pretendemos fazer uma avaliação do seu desempenho *ex-post* obtida em 1000 simulações. Para o efeito, utilizamos o programa *resultadosAbordagens.m* (ver anexo) que simula cada uma das abordagens para cada concretização dos preços gerada aleatoriamente. A avaliação do desempenho é feita tendo em conta as seguintes métricas:

- a) Valor da função objetivo, isto é, capital obtido no último instante temporal;

- b) Percentagem de simulações em que a solução não verifica todas as restrições do problema;
- c) Número médio de restrições violadas nas simulações que falharam;
- d) Valor médio que falta no saldo nas simulações que falharam;
- e) Percentagem do valor que falta no saldo relativamente ao saldo inicial nas simulações que falharam.

	a) u.m.	b) %	c)	d) u.m.	e) %
optDeterminística	6249.3	75.3	1.3267	336.4617	33.65
optRobusto1	4955.8	26	1.0462	70.6803	7.07
optRobusto2	4564.6	8.6	1.0233	3.2083	0.32
optRobusto3	4230.2	0	NaN	NaN	NaN
Heurística 1	3485.3	0	NaN	NaN	NaN
Heurística 2	4276.5	50.9	1	57.8727	5.79
Heurística 3	4359.1	51.6	1	57.2699	5.73

Tabela 4.1 Comparação entre heurísticas numa simulação com 1000 iterações.

Por análise à tabela, podemos verificar que a optDeterminística é aquela que apresenta o melhor valor da função objetivo (primeira coluna), sendo a sua diferença bastante considerável face às restantes estratégias. No entanto, a sua percentagem de simulações falhadas (segunda coluna), é bastante elevada, sendo de aproximadamente 75%. Deste modo, apesar de ser a estratégia que mais valoriza o capital no último instante, em geral não pode ser executada por falta de saldo. Reparemos ainda que, em média, nas simulações que falharam, foram violadas 1.3267 restrições (das três possíveis), pelo que em 32.67% das simulações que falharam, houve mais do que uma restrição a ser violada. Deste modo, foi originado uma falta de saldo médio de 336.4617 u.m. (33.65% do saldo inicial).

No que diz respeito às abordagens robustas verificamos que o valor da função objetivo diminui à medida que o σ aumenta. Esta é uma situação espectável, visto que por exemplo a estratégia optRobusto3 vai considerar 99.7% dos valores possíveis para os preços dos ativos, enquanto que essa percentagem decresce nas restantes estratégias robustas. Logo, a possibilidade de variação de valores dos preços face aos estipulados é tanto maior quanto menor for o valor de σ , o que permite alcançar um melhor valor da função objetivo na estratégia optRobusto1. A par disto, quanto maior é o valor de σ menor é a probabilidade de ocorrência de falhas, sendo que na optRobusto3 essa percentagem é nula (segunda coluna). Para as primeiras duas estratégias robustas, em geral, apenas uma das restrições falha e somente em 4.62% e 2.33% das simulações que falharam envolvem mais do que uma restrição violada. Também o défice no saldo (e respetiva percentagem face ao saldo inicial) vai diminuindo com o aumento do σ , sendo nulo na estratégia optRobusto3.

Por fim, resta fazer uma análise acerca das heurísticas clássicas apresentadas. Através da tabela, constatamos que a primeira heurística pode ser sempre executada pois não foi feita nenhuma alteração na carteira. No entanto, é a que apresenta pior resultado em termos do valor da função objetivo. Nas outras duas heurísticas, apesar de existir uma melhoria no valor da função objetivo, é de realçar que aproximadamente 50% das simulações falharam, originando um défice médio no saldo inferior a 6% face ao saldo inicial. Notemos que nestas duas estratégias, apenas falha uma restrição, uma vez que a carteira só é alterada no primeiro instante temporal.

Repetimos a análise realizada anteriormente considerando agora 20 000 e 50 000 simulações. A análise de desempenho será apenas focada nos campos a) e b) da tabela anterior.

	Nº de simulações	a) u.m.	b) %
optDeterminística	1000	6249.3	75.3
	20 000	6290.6	75.25
	50 000	6286.9	75.44
optRobusto1	1000	4955.8	26
	20 000	4986.5	24.29
	50 000	4985	24.1
optRobusto2	1000	4564.6	8.6
	20 000	4591.8	8.78
	50 000	4591.2	8.77
optRobusto3	1000	4230.2	0
	20 000	4256	0
	50 000	4255.1	0
Heurística 1	1000	3485.3	0
	20 000	3500.8	0
	50 000	3500	0
Heurística 2	1000	4276.5	50.9
	20 000	4303.1	50.42
	50 000	4301.8	50.26
Heurística 3	1000	4359.1	51.26
	20 000	4386.2	50.22
	50 000	4384.9	50.34

Tabela 4.2 Comparação das várias abordagens considerando 1000, 20 000 e 50 000 simulações.

Como podemos averiguar na tabela anterior o aumento significativo de iterações não provocou alterações relevantes nos resultados. Este facto pode também ser visualizado nos gráficos que se seguem.

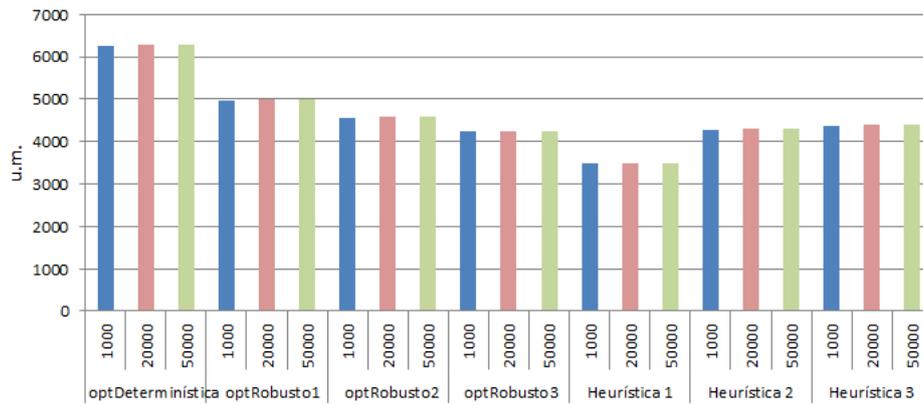


Fig. 4.1 Comparação das várias abordagens considerando 1000, 20 000 e 50 000 simulações relativamente ao valor da função objetivo.

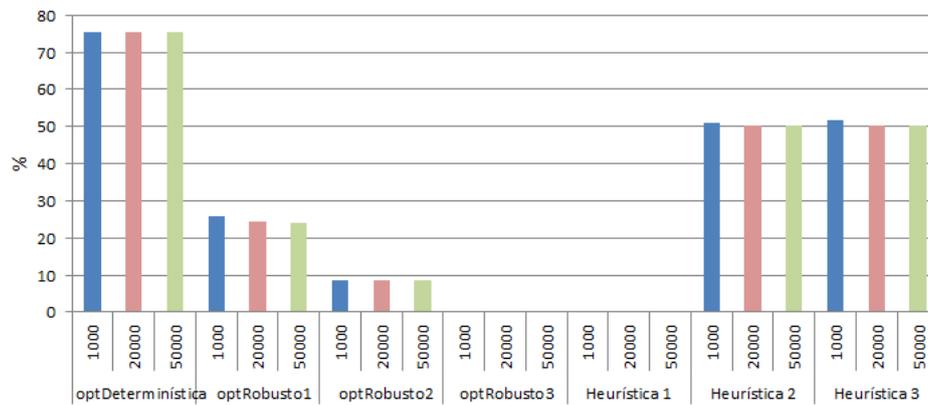


Fig. 4.2 Comparação das várias abordagens considerando 1000, 20 000 e 50 000 simulações relativamente à percentagem de simulações que falharam.

Em conclusão da análise dos gráficos e das tabelas relativamente à comparação entre as 7 estratégias utilizadas podemos verificar que a estratégia com $\sigma = 2$ destaca-se na relação valor objetivo - percentagem de simulações falhada tendo um bom valor de capital final e uma baixa percentagem de falha de cerca de 8.6% (comparando por exemplo com a Heurística 2 e 3, estas apresentam menor valor objetivo e bastante maior % de falha, levando a afirmar que são piores estratégias).

Bibliografia

- [1] Gerard Cornuejols and Reha Tütüncü. *Optimization Methods in Finance*.
- [2] Vinícius Aparecido Salatta et al. Abordagem robusta de Bertsimas e Sim aplicada a problemas de programação linear sujeitos a incertezas. *II Encontro Anual de Iniciação Científica*, 2016. URL <http://www.fecilcam.br/eventos/index.php/eaic/ieaic/paper/view/4223/1104>.

Anexo A

Todos os programas computacionais implementados em *Matlab* foram realizados com a colaboração do orientador.

```
function [solOtima] = Aplicacao(x0,saldo0, P, alfa ,beta)

%dimensoes
% n: numero de ativos, L: numero de instantes temporais
[n, L] = size(P);
beta_novo=1+beta;
alfa_novo=1-alfa;

%Definir coeficientes da f.o. (f)
f = zeros(3*n*L+L,1);
f(L) = -1;
f((L+(L-1)*n)+(1:n)) = -P(:,end);

%Definir A e b em "Ax <= b"
A_saldo = eye(L) - [zeros(1, L); eye(L-1) zeros(L-1, 1)];
A_x = zeros(L, n*L);
A_b = zeros(L, n*L);
A_s = zeros(L, n*L);
for l = 1:L
    A_b(l, (n*(l-1))+1:n) = beta_novo.*P(:,l)';
    A_s(l, (n*(l-1))+1:n) = alfa_novo.*P(:,l)';
end
A = [A_saldo A_x A_b -A_s];
b = zeros(L, 1);
b(1) = saldo0;

%Definir Aeq e beq em "Aeq x = beq"
Aeq_saldo = zeros(n*L, L);
Aeq_x = eye(n*L) - [zeros(n, n*L); eye((L-1)*n) zeros((L-1)*n, n)];
Aeq_b = eye(n*L);
Aeq_s = eye(n*L);
Aeq = [Aeq_saldo Aeq_x -Aeq_b Aeq_s];
beq = zeros(n*L, 1);
beq(1:n) = x0;

%Definir "LB" e "UB"
LB = zeros(3*n*L+L,1);
UB = ones(3*n*L+L,1)*Inf;

solOtima = linprog(f,A,b,Aeq,beq,LB,[]);
solOtima'
end

function [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo, x, P, alfa, beta, b, s)
% saldo0 e x0: saldo monetario e unidades dos ativos disponiveis no instante l=0

%dimensoes
% n: numero de ativos, L: numero de instantes temporais
[n, L] = size(P);

beta_novo=1+beta;
alfa_novo=1-alfa;

soma_b = (beta_novo)*(P.*b); %fazer o somatorio
soma_s = (alfa_novo)*(P.*s); %fazer o somatorio

%diferencas na primeira restricao de (3.2)
diferencaSaldo = zeros(1, L); %criar um vetor de zeros com l instantes temporais
```

```

diferencaSaldo(1) = saldo(1) - saldo0 - soma_s(1) + soma_b(1); %l=1
for l=2:3
diferencaSaldo(1) = saldo(1) - saldo(l-1) - soma_s(1) + soma_b(1); %l=2,3
end

diferencaX = zeros(n, L); % criar uma matriz n por l de zeros
diferencaX(:, 1) = x(:, 1) - x0 + s(:, 1) - b(:, 1); %l=1
diferencaX(:, 2:end) = x(:, 2:end) - x(:, 1:end-1) + s(:, 2:end) - b(:, 2:end); %l=2 e 3

valorFO = saldo(end) + (x(:, end))*P(:, end); % calculo da funcao objetivo
end

function [saldo, x, b, s] = dividirSolucao(solOtima, n, L)

saldo = solOtima(1:L)';
indicesSeparados = L;
x = reshape(solOtima(indicesSeparados+(1:n*L)),n, L);
indicesSeparados = indicesSeparados+n*L;
b = reshape(solOtima(indicesSeparados+(1:n*L)),n, L);
indicesSeparados = indicesSeparados+n*L;
s = reshape(solOtima(indicesSeparados+(1:n*L)),n, L);
indicesSeparados = indicesSeparados+n*L;
end

function [solOtima] = solucaoRobusta(x_0, saldo_0, mu_P, cov_P, alfa, beta, coefSigma)

%dimensoes
%an: numero de ativos, L: numero de instantes temporais
[n, L] = size(mu_P);
beta_novo=1+beta;
alfa_novo=1-alfa;

%Aproximacao inicial
x_inicial = zeros(3*n*L+L+1,1);

%Definir coeficientes da f.o. (f)
f = zeros(3*n*L+L+1,1);
f(end) = -1;

%Definir Aeq e beq em "Aeq x = beq"
Aeq_saldo = zeros(n*L, L);
Aeq_x = eye(n*L) - [zeros(n, n*L); eye((L-1)*n) zeros((L-1)*n, n)];
Aeq_b = eye(n*L);
Aeq_s = eye(n*L);
A_t= zeros(n*L,1);
Aeq = [Aeq_saldo Aeq_x -Aeq_b Aeq_s A_t];
beq = zeros(n*L, 1);
beq(1:n) = x_0;

%Definir "LB" e "UB"
LB = zeros(3*n*L+L+1,1);
UB = [];

solOtima = fmincon(@fRobusta, x_inicial, [], [], Aeq, beq, LB, [], @nonLinCon);
% solOtima';

%Funcoes auxiliares
function fValue = fRobusta(x)
fValue = f'*x;
end

%Restricoes no lineares
function [C, Ceq] = nonLinCon(x)
Ceq = [];
C = zeros(L+1, 1);
%l = 1
ult_indice_ant = (2*n+1)*L;
alfa_s = alfa_novo.*(x(ult_indice_ant+(1:n)))';
ult_indice_ant = (n+1)*L;
beta_b = beta_novo.*(x(ult_indice_ant+(1:n)))';
media = ( alfa_s - beta_b )*mu_P(:, 1);
variancia = alfa_s*cov_P*(alfa_s') - 2*alfa_s*cov_P*(beta_b') + beta_b*cov_P*(beta_b');
C(1) = x(1) - saldo_0 - media + coefSigma*sqrt(variancia);
%l >= 2
for l = 2:L
ult_indice_ant = (2*n+1)*L + n*(l-1);
alfa_s = alfa_novo.*(x(ult_indice_ant+(1:n)))';
ult_indice_ant = (n+1)*L + n*(l-1);
beta_b = beta_novo.*(x(ult_indice_ant+(1:n)))';
media = ( alfa_s - beta_b )*mu_P(:, 1);
variancia = alfa_s*cov_P*(alfa_s') - 2*alfa_s*cov_P*(beta_b') + beta_b*cov_P*(beta_b');

```

```

        C(1) = x(1) - x(1-1) - media + coefSigma*sqrt(variancia);
    end
    %t <= ...
    ult_indice_ant = L+n*(L-1);
    xL = x(ult_indice_ant+(1:n));
    C(end) = x(end) - x(L) - mu_P(:, L)'*xL + coefSigma*sqrt(xL'*cov_P*xL);
end
end

function [resultados] = resultadosAbordagens(saldo0, x0, P, alfa, beta, K)
%K : numero de simulacoes a realizar

    eps = 10^-6;
    desvPadrao_Coef = 0.1;
    resultados = zeros(7, 5);

    %dimensoes
    %n: numero de ativos, L: numero de instantes temporais
    [n, L] = size(P);

    %solucao otima programa linear
    xOpt = Aplicacao(x0, saldo0, P, alfa, beta);
    [saldo_opt, x_opt, b_opt, s_opt] = dividirSolucao(xOpt, n, L);
    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_opt, x_opt, P, alfa, beta, b_opt, s_opt);
    fprintf(' solucao_otima_programa_linear Valor_FO=%f, restricoes_<=nao_verificadas:%d, restricoes_==nao_verificadas%d\n',
        valorFO, sum(diferencaSaldo >eps), sum(sum(abs(diferencaX) > eps)));
x_opt

    cov_P = diag(desvPadrao_Coef*mean(P,2));
    %solucao otima robusta programa linear
    xOpt = solucaoRobusta(x0, saldo0, P, cov_P, alfa, beta, 1);
    [saldo_robust1, x_robust1, b_robust1, s_robust1] = dividirSolucao(xOpt(1:end-1), n, L);
    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_robust1, x_robust1, P, alfa, beta, b_robust1, s_robust1);
    fprintf(' solucao_otima_robusta Valor_FO=%f, restricoes_<=nao_verificadas:%d, restricoes_==nao_verificadas%d\n', valorFO,
        sum(diferencaSaldo >eps), sum(sum(abs(diferencaX) > eps)));
x_robust1

    %solucao otima robusta programa linear
    xOpt = solucaoRobusta(x0, saldo0, P, cov_P, alfa, beta, 2);
    [saldo_robust2, x_robust2, b_robust2, s_robust2] = dividirSolucao(xOpt(1:end-1), n, L);
    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_robust2, x_robust2, P, alfa, beta, b_robust2, s_robust2);
    fprintf(' solucao_otima_robusta Valor_FO=%f, restricoes_<=nao_verificadas:%d, restricoes_==nao_verificadas%d\n', valorFO,
        sum(diferencaSaldo >eps), sum(sum(abs(diferencaX) > eps)));
x_robust2

    %solucao otima robusta programa linear
    xOpt = solucaoRobusta(x0, saldo0, P, cov_P, alfa, beta, 3);
    [saldo_robust3, x_robust3, b_robust3, s_robust3] = dividirSolucao(xOpt(1:end-1), n, L);
    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_robust3, x_robust3, P, alfa, beta, b_robust3, s_robust3);
    fprintf(' solucao_otima_robusta Valor_FO=%f, restricoes_<=nao_verificadas:%d, restricoes_==nao_verificadas%d\n', valorFO,
        sum(diferencaSaldo >eps), sum(sum(abs(diferencaX) > eps)));
x_robust3

    %Solucao inicial mantida ao longo do tempo
    saldo_x0 = saldo0*ones(1:L);
    x_x0 = repmat(x0, 1, L);
    b_x0 = zeros(n, L);
    s_x0 = zeros(n, L);
    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_x0, x_x0, P, alfa, beta, b_x0, s_x0);
    fprintf(' solucao_x0 Valor_FO=%f, restricoes_<=nao_verificadas:%d, restricoes_==nao_verificadas%d\n', valorFO,
        sum(diferencaSaldo >eps), sum(sum(abs(diferencaX) > eps)));
x_x0

    %Solucao homogenea x
    xAux = ones(n, 1)*saldo0/sum((1+beta')*P(:,1));
    saldo_xFixo = zeros(1:L);
    x_xFixo = repmat(x0+xAux, 1, L);
    b_xFixo = zeros(n, L);
    b_xFixo(:,1) = xAux;
    s_xFixo = zeros(n, L);
    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_xFixo, x_xFixo, P, alfa, beta, b_xFixo, s_xFixo);
    fprintf(' solucao_homogenea_x Valor_FO=%f, restricoes_<=nao_verificadas:%d, restricoes_==nao_verificadas%d\n', valorFO,
        sum(diferencaSaldo >eps), sum(sum(abs(diferencaX) > eps)));
x_xFixo

    %Solucao homogenea xP
    xAux = saldo0./(n*(1+beta')*P(:,1));
    saldo_xPFixo = zeros(1:L);
    x_xPFixo = repmat(x0+xAux, 1, L);
    b_xPFixo = zeros(n, L);
    b_xPFixo(:,1) = xAux;

```

```

s_xPFixo = zeros(n, L);
[diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_xPFixo, x_xPFixo, P, alfa, beta, b_xPFixo, s_xPFixo);
fprintf('solucao_homogenea_xP_valor_FO=%f, restricoes_<=, nao_verificadas: %d, restricoes_>=, nao_verificadas: %d\n', valorFO,
sum(diferencaSaldo > eps), sum(sum(abs(diferencaX) > eps)));
x_xPFixo

%simulacao sobre incerteza
for k=1:K
    Paux = P + desvPadrao_Coef*P.*randn(n, L);

    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_opt, x_opt, Paux, alfa, beta, b_opt, s_opt);
    resultados(1, :) = resultados(1, :) + [valorFO, sum(diferencaSaldo > eps), sum(sum(abs(diferencaX) > eps)),
    sum(max(diferencaSaldo, 0), min(sum(diferencaSaldo > eps), 1))];
    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_robust1, x_robust1, Paux, alfa, beta, b_robust1, s_robust1);
    resultados(2, :) = resultados(2, :) + [valorFO, sum(diferencaSaldo > eps), sum(sum(abs(diferencaX) > eps)),
    sum(max(diferencaSaldo, 0), min(sum(diferencaSaldo > eps), 1))];
    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_robust2, x_robust2, Paux, alfa, beta, b_robust2, s_robust2);
    resultados(3, :) = resultados(3, :) + [valorFO, sum(diferencaSaldo > eps), sum(sum(abs(diferencaX) > eps)),
    sum(max(diferencaSaldo, 0), min(sum(diferencaSaldo > eps), 1))];
    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_robust3, x_robust3, Paux, alfa, beta, b_robust3, s_robust3);
    resultados(4, :) = resultados(4, :) + [valorFO, sum(diferencaSaldo > eps), sum(sum(abs(diferencaX) > eps)),
    sum(max(diferencaSaldo, 0), min(sum(diferencaSaldo > eps), 1))];
    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_x0, x_x0, Paux, alfa, beta, b_x0, s_x0);
    resultados(5, :) = resultados(5, :) + [valorFO, sum(diferencaSaldo > eps), sum(sum(abs(diferencaX) > eps)),
    sum(max(diferencaSaldo, 0), min(sum(diferencaSaldo > eps), 1))];
    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_xFixo, x_xFixo, Paux, alfa, beta, b_xFixo, s_xFixo);
    resultados(6, :) = resultados(6, :) + [valorFO, sum(diferencaSaldo > eps), sum(sum(abs(diferencaX) > eps)),
    sum(max(diferencaSaldo, 0), min(sum(diferencaSaldo > eps), 1))];
    [diferencaSaldo, diferencaX, valorFO] = simulacao(saldo0, x0, saldo_xPFixo, x_xPFixo, Paux, alfa, beta, b_xPFixo, s_xPFixo);
    resultados(7, :) = resultados(7, :) + [valorFO, sum(diferencaSaldo > eps), sum(sum(abs(diferencaX) > eps)),
    sum(max(diferencaSaldo, 0), min(sum(diferencaSaldo > eps), 1))];
end
resultados = resultados/K;

fprintf('Valor_da_funcao_objetivo\n');
disp(resultados(:,1)');

fprintf('Proporcao_de_problemas_que_falharam\n');
disp(resultados(:,5)');

fprintf('Numero_medio_de_restricoes_violadas_nos_problemas_que_falharam\n');
disp((resultados(:,2)./resultados(:,5))');

fprintf('Valor_medio_que_falta_no_saldo_nos_problemas_que_falharam\n');
disp((resultados(:,4)./resultados(:,5))');

fprintf('Proporcao_do_valor_que_falta_no_saldo_relativamente_ao_saldo_inicial_nos_problemas_que_falharam\n');
disp(((resultados(:,4)./resultados(:,5)))/saldo0);
end

```