Aceleração de Convergência para problemas de Otimização estócástica multiestágio

Colaboração ONS – UFRJ

Bernardo Freitas Paulo da Costa, Iago Leal de Freitas & Rafael Benchimol Klausner

UFRJ – Universidade Federal do Rio de Janeiro
Departamento de Matemática Aplicada – Instituto de Matemática
Av. Athos da Silveira Ramos, 149
Centro de Tecnologia, Bloco C - Cidade Universitária
Caixa Postal 68530 – 21941-909 – Rio de Janeiro – RJ – Brasil

Filipe Goulart Cabral, Joari Paulo da Costa & Débora Dias Jardim Penna

ONS – Operador Nacional do Sistema Elétrico
Rua Júlio do Carmo, 251 - Cidade Nova
20211-160 – Rio de Janeiro – RJ – Brasil

Dezembro 2019

Conteúdo

Sumário

Dando continuidade ao projeto de pesquisa *"Metodologias de Programação Estocástica Inteira"*, estabelecido entre o ONS e a COPPETEC/Instituto de Matemática da UFRJ, contrato GMC-CT-093/17, este projeto visa o estudo das funções de custo futuro resultantes da modelagem com variáveis inteiras. De fato, a maior flexibilidade de modelagem resultante leva, também, à perda da garantia teórica de convexidade, que é um dos ingredientes fundamentais para o sucesso dos algoritmos de otimização estocástica multi-estágio. Assim, estudaremos novas metodologias de cálculo da função de custo futuro, tanto de forma exata como aproximada, visando a redução do custo computacional e incorrendo em um impacto limitado na perda de qualidade da política de operação resultante. Esperamos obter uma melhor convergência com o algoritmo proposto sob dois pontos de vista: primeiro, menor *gap* entre a solução e a simulação, se comparado com o uso de cortes de Benders para cada cenário. Em segundo lugar, diminuição do tempo de processamento, se comparado com a formulação via SDDiP que discretiza e binariza as variáveis de estado.

Este relatório consolida os resultados principais tanto do desenvolvimento metodológico quanto dos estudos de caso realizados, sendo composto por três partes principais.

A primeira parte é dedicada às diversas contribuições matemáticas, das quais destacamos:

- o conceito de *convexificação por médias*, que sugere que, na presença de incertezas, a função de custo futuro possa ser convexa, mesmo que suas componentes, referentes a cenários específicos, sejam não-convexas;
- a desigualdade $\mathbb{E}\left[\text{conv}(Q(x,\Xi))\right] \leqslant \text{conv}(\mathbb{E}\left[Q(x,\Xi)\right]) \leqslant \mathbb{E}\left[Q(x,\Xi)\right] = \bar{\mathcal{Q}}(x)$, que explicita a qualidade da aproximação da função de custo futuro $\bar{\mathcal{Q}}(x)$ em função da *ordem* entre o cálculo da relaxação Lagrangiana, representada pelo operador $\text{conv}(\cdot)$, e o cálculo da média, representada pelo operador $\mathbb{E}\left[\cdot\right]$;
- a noção de *medidas de não-convexidade*, que permitem deduzir ao mesmo tempo resultados mais precisos ou mais gerais referentes à convexificação, tanto de um ponto de vista quantitativo quanto do ponto de vista qualitativo;
- a organização dos resultados teóricos desenvolvidos enfatizando sua relação com algoritmos de planos cortantes, que justifica e dá suporte aos algoritmos desenvolvidos.

Desta parte, emergem modificações do algoritmo de programação dual dinâmica estocástica, dentre as quais o cálculo de cortes de Benders *reforçados* e a *formulação conjunta* dos problemas de cada estágio.

A segunda parte deste relatório consiste na análise e comparação dos diferentes algoritmos propostos utilizando tanto um problema estilizado, para o qual é possível calcular a função de custo futuro exata para cada estágio, quanto um modelo reduzido do SIN, composto por 2 subsistemas. A política é calculada considerando os casos tanto de função objetivo neutra a risco como o caso onde esta é avessa a risco. Desta análise, pudemos observar que:

- a decomposição em cenários independentes, no caso não convexo, nem sempre é capaz de fornecer cortes exatos para a função de custo futuro, ao contrário de uma formulação conjunta;
- a construção de cortes para a formulação conjunta dos problemas de um estágio requer maior tempo computacional, em particular quanto maior for a quantidade de variáveis de decisão descontínuas;
- este maior tempo computacional, se usado em algoritmos que abordam a formulação decomposta, é todavia insuficiente para obter uma redução comparável do *gap* de dualidade, o que sugere uma limitação intrínseca da abordagem por decomposição;

- diversos destes resultados ainda são válidos para o caso avesso a risco.

Os diversos exemplos apresentados sugerem que esta abordagem é promissora para obter uma melhor representação das funções de custo futuro.

Para implementar as metodologias propostas e avaliar seu efeito na operação, continuamos o desenvolvimento do protótipo computacional, na linguagem Julia, usando o pacote `SDDP.jl`, tornando-o mais flexível e atualizando para a nova versão da linguagem e bibliotecas. Além disso, desenvolvemos três bibliotecas, a saber: `MIPManip`, `Relink` e `SDDP_SB`; para implementar os algoritmos propostos neste trabalho. O protótipo e as bibliotecas auxiliares estão disponibilizados em código aberto para o ONS, e sua documentação constitui a terceira parte deste relatório.

1 Introdução

Até o surgimento do algoritmo SDDiP em 2016 [ZAS16], os únicos problemas de otimização estocástica multi-estágio que possuíam um método eficiente de solução eram aqueles cuja formulação era convexa, e para os quais se podia assegurar que as funções de custo futuro eram convexas. De fato, nestes casos é possível representar as funções de custo futuro através de planos cortantes e, no caso particular, mas importante, em que a estocasticidade é *independente* ao longo do tempo, o algoritmo SDDP [PP91] permite resolver problemas com muitos estágios, já que sua complexidade cresce apenas linearmente com o número de estágios de decisão.

A necessidade de representar restrições operacionais de forma mais detalhada no planejamento da operação ampliou o interesse em modelos não-convexos, que são suficientemente versáteis para incorporar tais restrições. Entretanto, dada a necessidade de representar funções de custo futuro mais gerais, estes algoritmos são também computacionalmente mais custosos, tanto pelo aumento da complexidade de cada subproblema, que deixa de ser um problema linear e passa a ser inteiro-misto, como pelo aumento do custo computacional para a obtenção dos cortes.

Este projeto se dedica ao estudo de novas metodologias de cálculo da função de custo futuro, tanto de forma exata como aproximada, visando a redução do custo computacional e incorrendo em um impacto limitado na perda de qualidade da política de operação resultante. Esperamos obter uma melhor convergência sob dois pontos de vista: primeiro, o algoritmo proposto deveria resultar em um menor *gap* entre a solução e a simulação, se comparado com o uso de cortes de Benders para cada cenário. Em segundo lugar, pela diminuição do tempo de processamento, se comparado com a formulação via SDDiP que discretiza e binariza as variáveis de estado.

Para tal, iremos explorar como a presença de incerteza estocástica pode resultar em funções de custo futuro convexas, mesmo quando os problemas de otimização formulados não o são. Uma fonte particular desta "convexificação", que está no cerne de nossas investigações, é como a média de diversas funções não convexas pode resultar em uma função convexa, ou, pelo menos, com não-convexidades menos pronunciadas.

Iniciamos este relatório apresentando uma classe específica de problemas de otimização estocástica, onde a incerteza afeta apenas o estado inicial, que é típico de problemas de controle ótimo, como no caso do planejamento da operação. Como veremos nos exemplos da seção 2, este tipo de estocasticidade resulta em uma suavização e convexificação da função de custo futuro, o que sugere novas abordagens para a aproximá-la. Em seguida, a seção 3 irá estudar em detalhes o *gap* de dualidade das funções de custo imediato e de custo futuro, e a partir daí deduzimos os primeiros resultados de suavização, justificando e generalizando os exemplos vistos na seção anterior em duas direções: primeiro, para uma estrutura geral de incerteza, e depois para o caso de otimização estocástica com aversão a risco.

Em seguida, introduzimos na seção 4 a noção de *medida de não convexidade*, que nos permite estabelecer outros critérios, além do *gap* de dualidade, para avaliar o quão não-convexa é uma dada função. Esta seção generaliza diversos resultados obtidos na seção 3 em um arcabouço mais abstrato, e que mostra que a presença de incerteza reduz a não-convexidade da função de custo futuro, para praticamente todos os critérios naturais que se escolheria para quantificar se uma função é mais ou menos convexa do que outra. Esta seção se encerra com alguns exemplos de medidas de não-convexidade, que surgem naturalmente seja na caracterização de funções convexas, seja em algoritmos de otimização.

Enfim, encerramos esta primeira parte teórica apresentando na seção 5 as operações de convolução e um resumo da teoria de distribuições. Com isto, poderemos estender as definições e resultados sobre derivadas da seção 4 a funções contínuas não diferenciáveis, importantes por serem típicas de problemas de otimização linear. Além disso, este ponto de vista trata diretamente variáveis

aleatórias gerais, unificando probabilidades discretas e contínuas num mesmo arcabouço lógico.

Em direção às aplicações, apresentamos na seção 6 as modificações para o procedimento de cálculo de cortes que visam descrever a função de custo futuro em programação dinâmica dual estocástica. A principal observação, decorrente dos resultados das seções anteriores, é que a decomposição em cenários independentes, no caso não convexo, nem sempre é capaz de fornecer cortes exatos para a função de custo futuro, ao contrário de uma formulação conjunta, que descreveremos. Estas modificações são um pouco mais delicadas no caso de problemas estocásticos onde a função de custo futuro é calculada com aversão a risco, pois exigem uma modelagem mais cuidadosa da formulação conjunta para que se possam obter cortes.

Em seguida apresentamos os resultados da prova de conceito computacional, ilustrando as técnicas desenvolvidas. Iremos comparar a proposta de uso de cortes de Benders reforçados para a formulação conjunta com metodologias existentes, que usam a relaxação linear das restrições, ou a binarização das variáveis de estado. Faremos na seção 7 estas comparações em um problema estilizado, que nos permite uma análise direta, porque é possível calcular a função de custo futuro exata para cada estágio.

Antes de abordar um problema mais próximo do planejamento da operação do setor elétrico, descrevemos na seção 8 o ferramental computacional desenvolvido neste projeto para incorporar estas novas metodologias à base de código existente, tanto oriunda do projeto anterior como da evolução das bibliotecas em Julia. Esta seção termina com uma descrição mais precisa da interface de linha de comando para nosso programa principal, denominado `DisjHTPlan`.

Na seção 9 apresentamos os resultados computacionais para uma versão reduzida de um problema de planejamento hidrotérmico. Aqui, iremos gradativamente tornar o modelo mais complexo: começando de uma versão convexa do problema, sem restrições operativas, avançamos para uma formulação não-convexa, e, finalmente, introduzimos um critério de aversão a risco via CVaR.

Enfim, encerramos o relatório com uma análise global dos resultados da prova de conceito, e apontamos algumas direções de pesquisa futura.

2 Exemplos de suavização e convexificação promovidos pela estocasticidade

Nesta seção, apresentamos através de alguns exemplos diversas intuições relativas à suavização de funções de custo futuro, enfatizando a relação entre os casos convexo e não-convexo. Em particular, estamos interessados no caso em que a presença da estocasticidade em um problema de otimização reduz o *gap* de dualidade, o que permitirá utilizar aproximações convexas para a função de custo futuro. No caso extremamente favorável em que o *gap* é reduzido a zero, diremos que a função foi *convexificada* pela estocasticidade.

Estas intuições servirão como apoio e motivação para as formulações mais precisas e mais gerais que veremos nas seções 3.2.1 e 4. Ilustraremos inicialmente o que ocorre no caso convexo, para só depois apresentarmos o caso não-convexo. Começamos apresentando uma breve recapitulação dos conceitos de função valor ótimo e de função de custo futuro e fixamos a notação que será utilizada neste texto para as mesmas.

2.1 Funções de valor ótimo e funções de custo futuro

Uma *função de valor ótimo* é o resultado de um problema de otimização, enquanto que uma *função de custo futuro* incorpora, também, a incerteza estocástica. Mais precisamente:

$$\begin{aligned} Q_t(x_{t-1}, \xi_t) = \min_{x_t, y_t} \quad & c^\top x_t + d^\top y_t + \bar{\mathcal{Q}}_{t+1}(x_t) \\ \text{s.a} \quad & Ax_t + By_t = b_t + Tx_{t-1} + W\xi_t, \\ & (x_t, y_t) \in D_t \end{aligned} \tag{2.1}$$

é a função de valor ótimo do problema de otimização correspondente ao estágio t, dados o estado inicial x_{t-1} e uma realização ξ_t da variável aleatória Ξ_t, e cujas variáveis são o estado final x_t e as decisões locais y_t. Ao considerarmos a distribuição de Ξ_t, obtemos a função de custo futuro

$$\bar{\mathcal{Q}}_t(x_{t-1}) = \mathbb{E}\left[Q_t(x_{t-1}, \Xi_t)\right]. \tag{2.2}$$

Vale notar que o próprio problema do estágio t também inclui uma função de custo futuro, $\bar{\mathcal{Q}}_{t+1}$, construída recursivamente a partir dos problemas dos estágios seguintes. Para nossos propósitos, podemos supor que as matrizes A, B, T e W estão fixas, e D_t designa o conjunto viável para a decisão (x_t, y_t) o qual não depende das equações de transição de estado.

Em problemas de controle ótimo, onde a nova variável de estado é dada por um balanço de "entradas e saídas", é comum que a incerteza corresponda à chegada ou saída aleatória do estoque. Assim, a equação de transição de estado fica na seguinte forma particular:

$$Ax_t + By_t = b + T(x_{t-1} + \xi_t). \tag{2.3}$$

Por exemplo, no caso hidrotérmico com um modelo aditivo para as afluências ξ, a equação de transição de estado é simplesmente

$$v_f + q = v_i + \xi,$$

onde v_f é o volume final, v_i o volume inicial e q a defluência total, que é de fato da forma (2.3), com $b = 0$.

Nestas condições, a função de valor ótimo $Q_t(x_{t-1}, \xi_t)$ pode ser escrita como uma função de uma única variável:

$$Q_t(x_{t-1}, \xi_t) = f(x_{t-1} + \xi_t), \tag{2.4}$$

e a função de custo futuro $\bar{\mathcal{Q}}(x_{t-1})$, pela média:

$$\bar{\mathcal{Q}}_t(x_{t-1}) = \int_{-\infty}^{\infty} f(x_{t-1} + s)\, p(s) ds,$$

onde $p(s)$ denota a densidade de probabilidade da variável aleatória Ξ_t.

A fim de não sobrecarregar a notação, omitiremos o subíndice t nos exemplos das seções a seguir.

2.2 Suavização no caso convexo

Para vermos um exemplo da suavização, suponha que f é a função módulo, $f(u) = |u|$, e que a incerteza seja dada por uma variável uniforme $\Xi \sim U(-1, 1)$. Então teremos:

$$\bar{\mathcal{Q}}(x) = \int_{-1}^{1} |x + s|\ ds = \begin{cases} |x| & \text{se } |x| \geqslant 1 \\ (1 + x^2)/2 & \text{se } |x| \leqslant 1 \end{cases}$$

Como podemos ver na figura 2.1, a função $\bar{\mathcal{Q}}$ é derivável em todos os pontos, enquanto que a função original f é apenas contínua, já que não possui derivada em $x = 0$. Assim, ao tomar médias, aumentamos o grau de regularidade das funções de custo futuro, o que é uma das indicações da suavização promovida pela presença da estocasticidade.

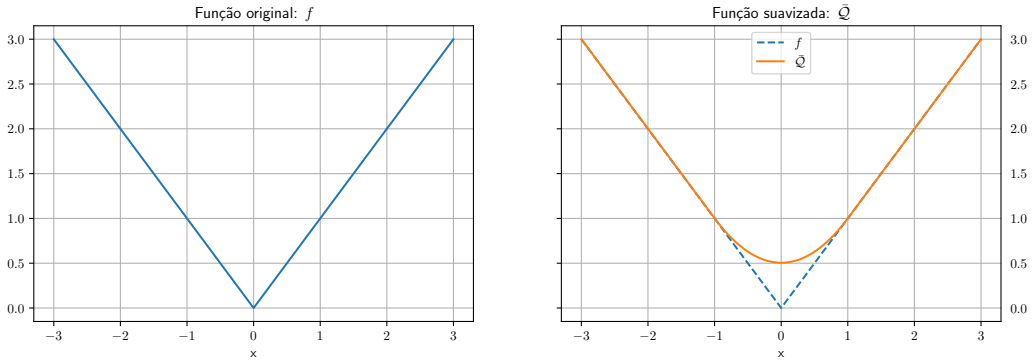


Figura 2.1: A função valor absoluto f e sua suavização por médias $\bar{\mathcal{Q}}$.

Ainda vale notar que, ao considerar a função de custo futuro $\bar{\mathcal{Q}}$, também "regularizamos" a função valor ótimo f segundo outro critério: os extremos de f são menos pronunciados, o que é bastante visível na figura 2.1 à direita. De fato, vemos que no intervalo $[-1, 1]$ a função $\bar{\mathcal{Q}}$ "descola" de f, e seu valor mínimo é 0.5, e não 0 como para f,

Do ponto de vista da convexidade, observamos na figura 2.1 que a passagem de f para $\bar{\mathcal{Q}}$ não destrói a convexidade, já que $\bar{\mathcal{Q}}$ é uma média de funções convexas. Na verdade, distribuímos a convexidade de f numa região maior. Intuitivamente, podemos considerar que a segunda derivada de f é uma função delta, infinita em $x = 0$ e nula nos outros pontos, e que ao tomar médias com translações esta massa infinita é repartida no intervalo $[-1, 1]$, onde a segunda derivada de $\bar{\mathcal{Q}}$ é igual a um. De forma mais precisa, notamos que há apenas um ponto x onde f é *estritamente convexa*, ou seja, para apenas um ponto $(x, f(x))$ existe uma reta suporte ao gráfico de f que toca o gráfico *apenas em* x: o ponto $(0, 0)$. Ao levar em conta a estocasticidade dada pela variável aleatória Ξ, de suporte $[-1, 1]$, a função $\bar{\mathcal{Q}}$ passa a ser estritamente convexa em todo o intervalo $(-1, 1)$.

2.3 Suavização no caso não-convexo

Vejamos agora como a incerteza opera sobre as propriedades da função de custo futuro $\bar{\mathcal{Q}}$ caso o problema possua elementos não-convexos. Um exemplo de não convexidade é a introdução de variáveis binárias para modelar políticas operacionais, como já descrevemos nos relatórios anteriores. Outro uso de variáveis binárias é em problemas de *unit commitment*, onde devemos decidir se um dado recurso estará disponível ou não no futuro.

Nesta seção, todos os exemplos irão explorar a função f em forma de "W", como ilustrado na figura 2.2 à esquerda. Esta função pode ser dada pelo mínimo de duas funções convexas, $W(x) = \min\{|x + 1|, |x - 1|\}$, como vemos na figura 2.2 à direita.

Funções que são mínimo de funções convexas são típicas de problemas com variáveis binárias. De fato, a estratégia da decomposição de Benders para resolver um problema de minimização com variáveis binárias z e variáveis contínuas consiste em dividi-lo em duas etapas: uma *externa* que decide a variável inteira z, e uma *interna*, que, dada uma escolha de z, toma a melhor decisão possível. Como para valores diferentes de z podemos ter problemas totalmente diferentes, para

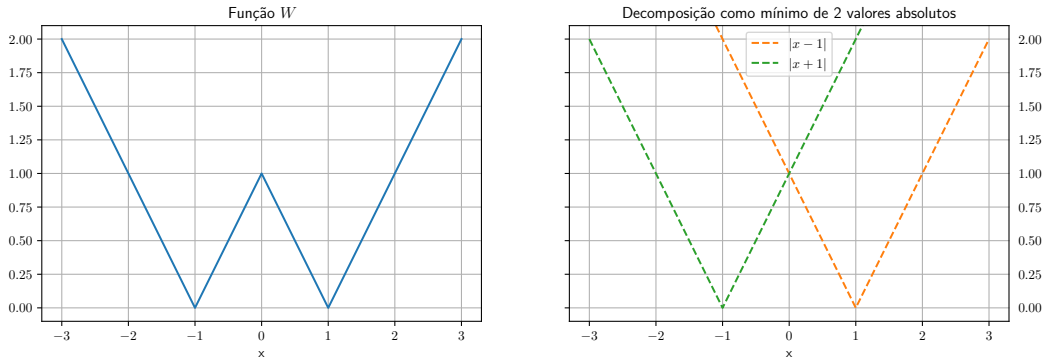


Figura 2.2: A função W e uma decomposição de Benders como mínimo de duas funções convexas.

cada z temos uma diferente função convexa $f_z(x)$ que vamos minimizar, em função do estado x, e o problema como um todo passa a ser $\min_z f_z(x)$.

Suponhamos agora que estamos no mesmo caso do problema anterior para a incerteza: $\Xi \sim U(-1, 1)$. Também podemos calcular a função de custo futuro $\bar{\mathcal{Q}}$ correspondente, e neste caso obtemos a função convexa indicada na figura 2.3 à direita. Assim, podemos observar que a média

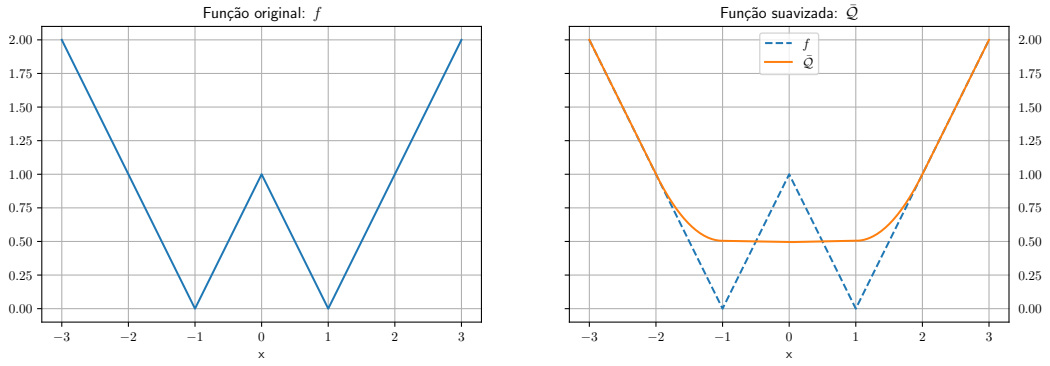


Figura 2.3: A função "W" e sua suavização por médias $\bar{\mathcal{Q}}$.

de funções não-convexas pode ser uma função convexa. De maneira geral, notamos que, como no caso convexo, o fato de tomar médias de funções "suaviza" os picos da mesma, e isso se reflete na redução de máximos locais. Neste caso, isso resultou em uma função de custo futuro $\bar{\mathcal{Q}}$ convexa, mesmo que a função valor ótimo f não seja.

Ainda de maneira informal, podemos pensar que distribuímos tanto as convexidades (presentes em $x = \pm 1$, onde f'' é uma função delta positiva) como a não convexidade (em $x = 0$, onde f'' é uma função delta negativa). Esta distribuição se dá, como no caso convexo, cada uma delas ocupando, em $\bar{\mathcal{Q}}$, um intervalo de raio um (devido à lei de Ξ) em torno do ponto onde estava presente em f. Assim, "metade" de cada convexidade (na direção de $x = 0$) cancelou a metade correspondente da concavidade proveniente de $x = 0$, enquanto que a metade restante se distribuiu nos intervalos $(-2, -1)$ e $(1, 2)$, onde $\bar{\mathcal{Q}}$ tem derivada estritamente positiva. Este intervalo também será o que contém os pontos de convexidade estrita para $\bar{\mathcal{Q}}$.

Também poderíamos usar outras distribuições para a incerteza, por exemplo a normal: $\Xi \sim \mathcal{N}(0, 1)$. Neste caso, a densidade da distribuição é dada por uma função infinitamente diferenciável,

e o suporte da variável aleatória Ξ é $\mathbb{R}$ inteiro. Assim, obtemos uma função $\bar{\mathcal{Q}}$ que também é infinitamente diferenciável, e "distribuímos" a convexidade (e a concavidade) em todos os pontos: a função resultante é agora estritamente convexa em $\mathbb{R}$, como podemos ver na figura 2.4:

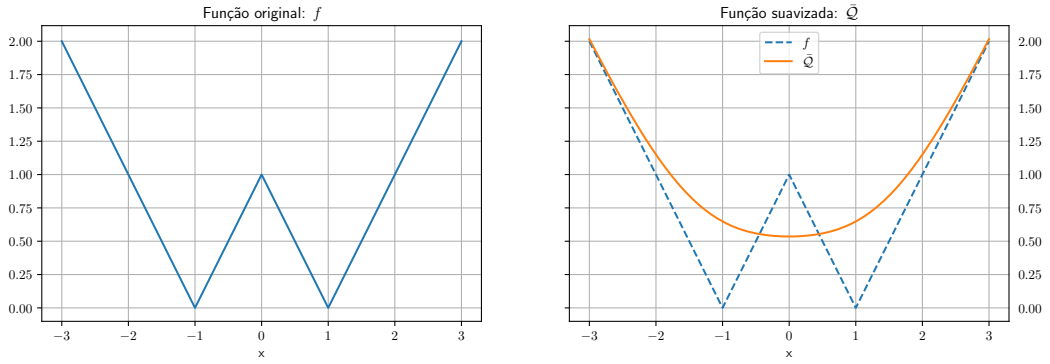


Figura 2.4: A função "W" e sua suavização por médias $\bar{\mathcal{Q}}$.

É claro que nem toda distribuição de Ξ resultará em uma função de custo futuro $\bar{\mathcal{Q}}$ convexa. Ilustramos nas figuras 2.5 e 2.6 o resultado da média da função W por três membros diferentes de cada uma das famílias de distribuições uniformes e gaussianas, respectivamente. Em cada uma das figuras, vamos aumentando progressivamente a incerteza: no caso das uniformes, ao aumentar a amplitude; no caso das gaussianas, ao aumentar a variância. Podemos observar um comportamento de certa forma regular na "amplitude" da distribuição de Ξ: quanto mais dispersa for a incerteza, maior o efeito de convexificação resultante em $\bar{\mathcal{Q}}$. Assim, para as primeiras distribuições de cada família, que ainda estão bastante concentradas em zero, o efeito de suavização não é suficiente para tornar $\bar{\mathcal{Q}}$ convexa, mesmo que o *gap de dualidade* já esteja reduzido, vide a primeira linha das figuras 2.5 e 2.6. Em seguida, ilustramos os primeiros membros de cada família que tornam $\bar{\mathcal{Q}}$ convexa: a segunda derivada se anula em zero em ambos os casos, como pode ser observado na segunda linha das figuras 2.5 e 2.6. Enfim, ao continuar aumentando a dispersão das distribuições, vemos que agora a função resultante passa a ser estritamente convexa em zero.

O conjunto de distribuições da variável aleatória Ξ que, para uma f dada, tornam $\bar{\mathcal{Q}} = \mathbb{E}\left[f(\cdot + \Xi)\right]$ convexa será objeto de estudo das próximas seções.

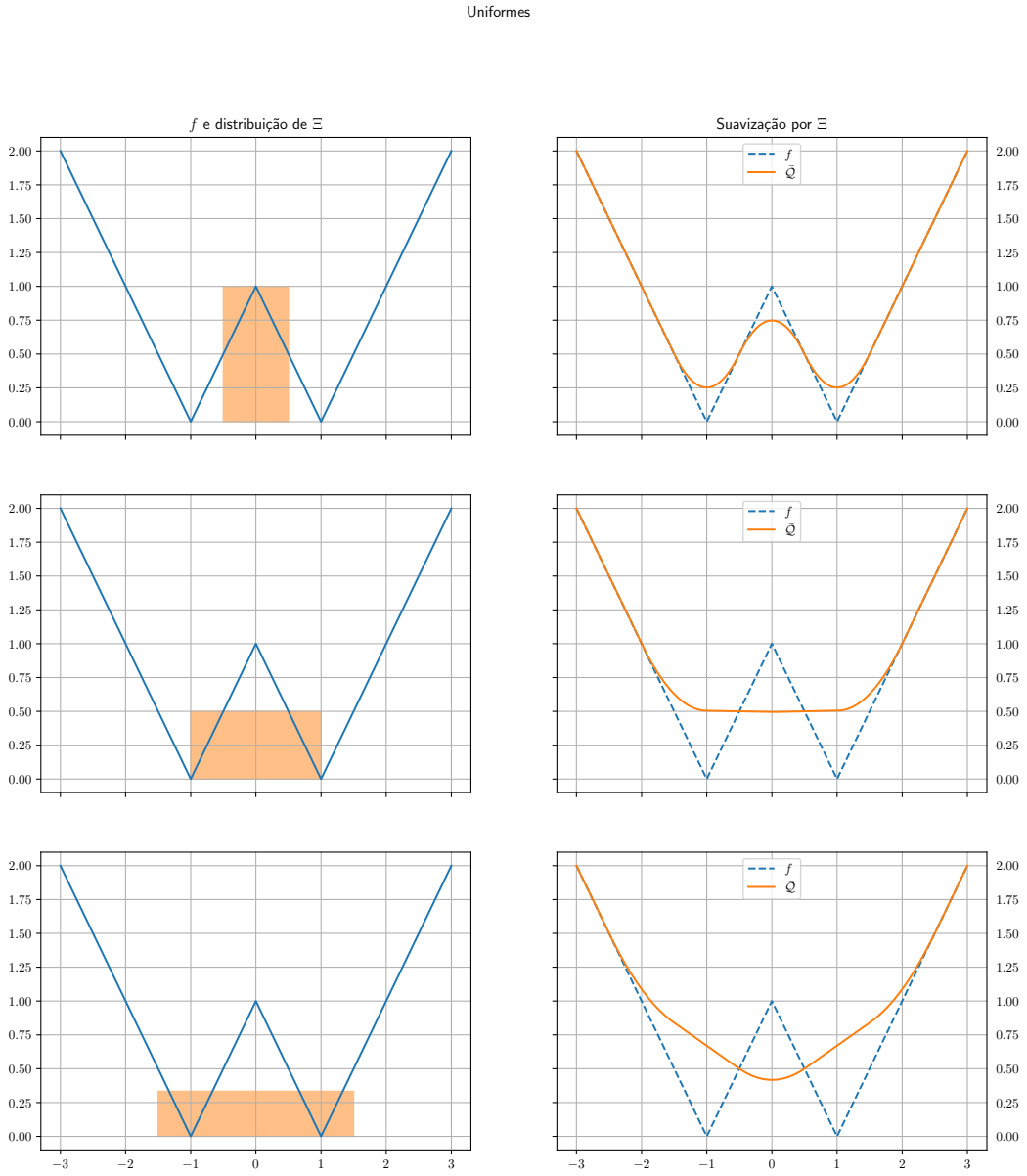


Figura 2.5: Suavização da função “W” por diferentes distribuições uniformes.

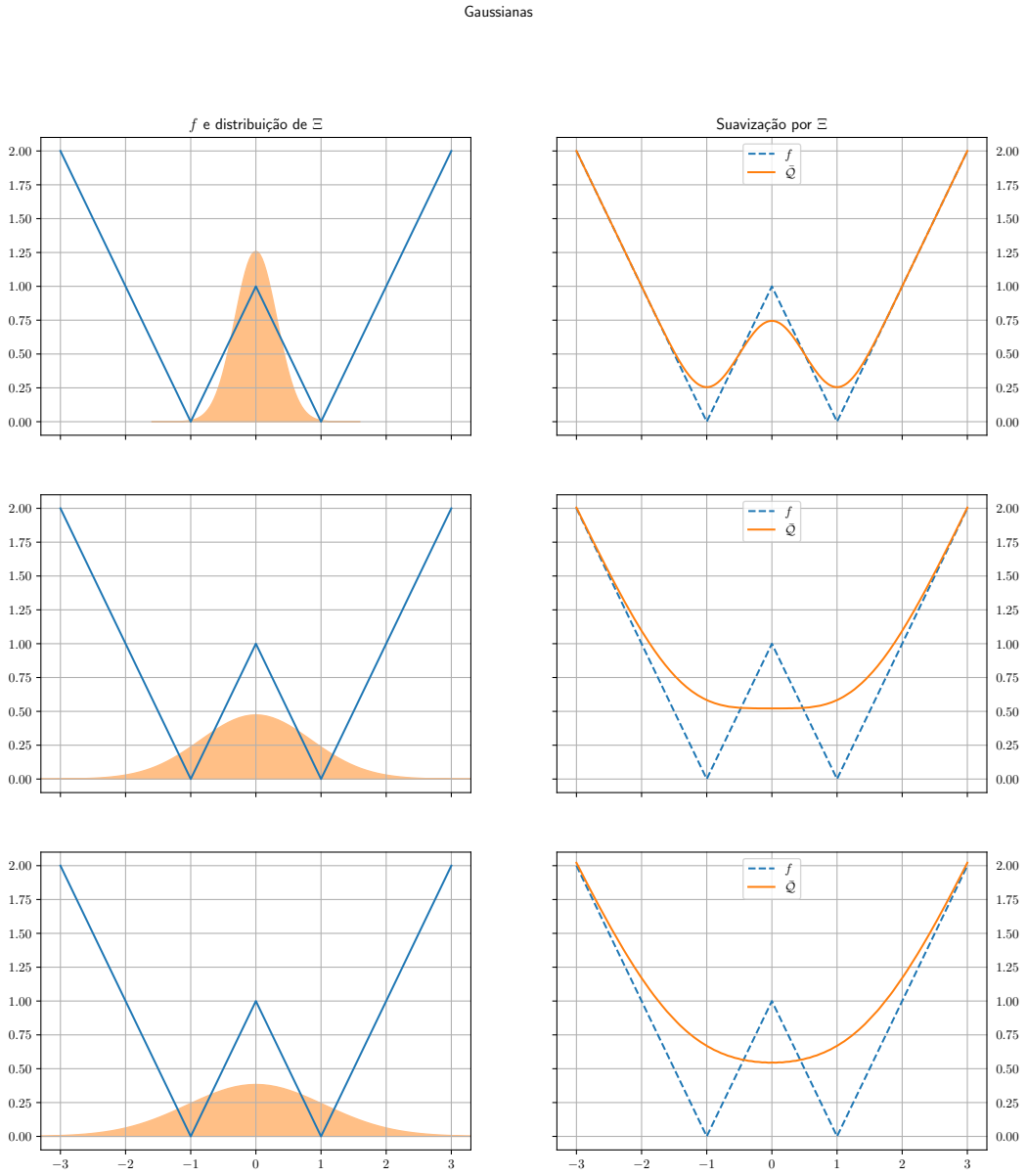


Figura 2.6: Suavização da função “W” por diferentes distribuições normais.

2.4 Medidas de não-convexidade

Na seção anterior, mostramos que diversos ruídos fazem com que uma função de custo futuro com o formato de um "W" se torne mais convexa. A seguir, ilustraremos o mesmo fenômeno de um ponto de vista quantitativo, mostrando como algumas medidas de não-convexidade para a função "W" são reduzidas de acordo com o ruído considerado.

Para tal, usaremos duas medidas de não-convexidade aplicadas à função W:

1. O máximo do seu *gap*: $\sup_{x\in\mathbb{R}} \left(W(x) - \check{W}(x)\right) = 1$;
2. A área do seu *gap*: $\int \left(W(x) - \check{W}(x)\right) dx = 1$.

A figura 2.7 mostra o gráfico da relaxação convexa $\check{W}$ sobreposto ao da função W conjuntamente com essas medidas de não-convexidade. Na figura, o máximo do *gap* é o comprimento da linha pontilhada enquanto que a área do *gap* é a área da região em cinza.

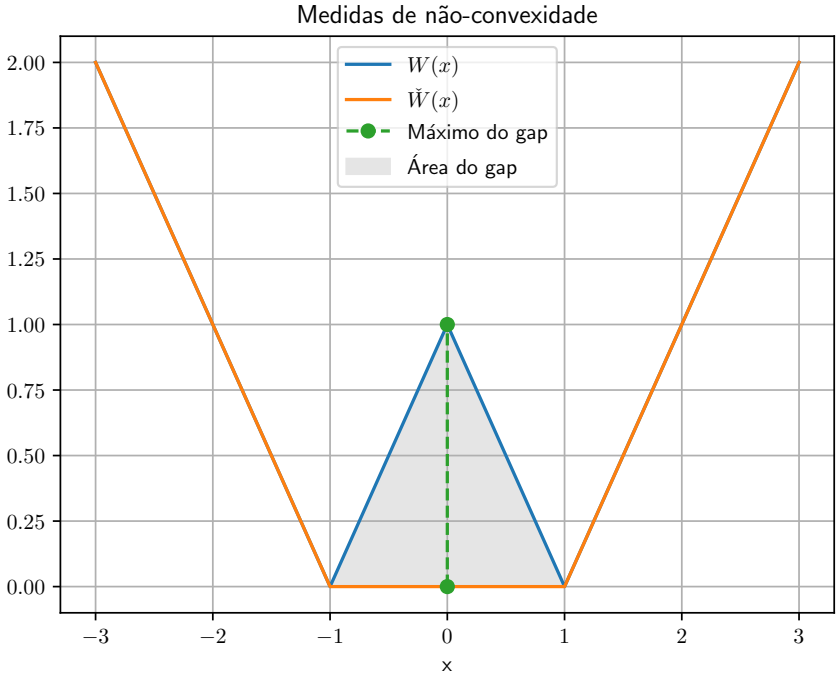


Figura 2.7: Medidas de não-convexidade para a função W.

Retornando ao exemplo de ruídos dados por uma variável aleatória uniforme, observamos este processo de "convexificação" na figura 2.8. Para ruídos da forma $\Xi \sim U(-h, h)$, a função de custo futuro $\bar{\mathcal{Q}}(x) = \mathbb{E}\left[W(x + \Xi)\right]$ terá um *gap* cada vez menor quanto maior for o valor de h. Na imagem da esquerda, são mostrados os gráficos das funções $\mathbb{E}\left[W(\cdot + \Xi)\right]$ para h pequeno e vemos que quanto maior o valor de h, mais a média $\mathbb{E}\left[W(\cdot + \Xi)\right]$ se aproxima de uma função convexa. Já na imagem da direita, são mostrados os gráficos quando a h é grande o suficiente para que ocorra convexificação total da média $\mathbb{E}\left[W(\cdot + \Xi)\right]$.

Para vermos como esta "convexificação" se traduz de forma quantitativa, indicamos na figura 2.9 como o parâmetro h da janela uniforme afeta o máximo (figura da esquerda) e a área (figura da direita) da função $\mathbb{E}\left[W(\cdot + \Xi)\right]$. Repare que, quando o tamanho da janela uniforme é maior do que

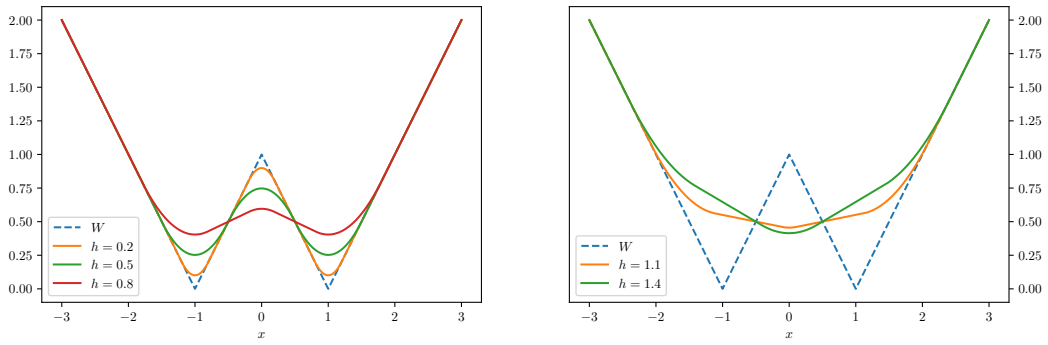


Figura 2.8: Gráfico de $\mathbb{E}[W(\cdot + \Xi)]$ para Ξ uniforme em $[-h, h]$.

1, ambas as medidas de não-convexidade são iguais a zero, o que indica que a média $\mathbb{E}[W(\cdot + \Xi)]$ passa a ser uma função convexa, o que está em total acordo com o que observamos anteriormente na figura 2.8.

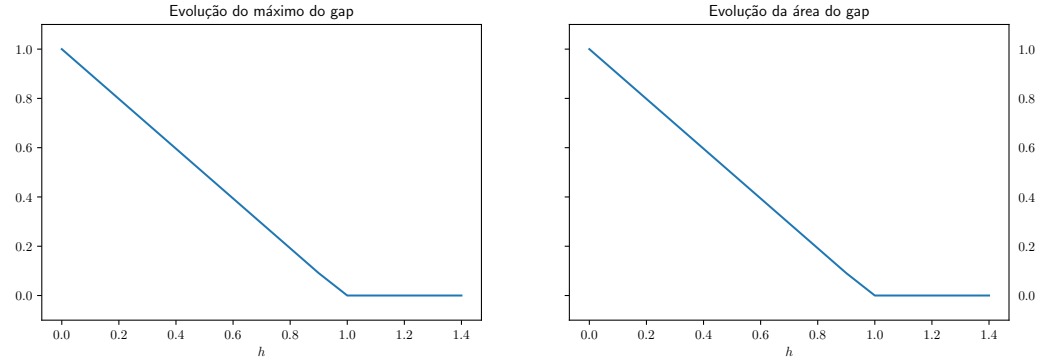


Figura 2.9: Para uma incerteza de tipo $\Xi \sim U(-h, h)$, as medidas de não-convexidade de $\mathbb{E}[W(\cdot + \Xi)]$ decrescem linearmente até que a função torne-se convexa.

Se agora assumirmos que a distribuição da variável aleatória Ξ é uma normal de média zero e variância σ^2, o mesmo estudo pode ser realizado em função do desvio padrão σ.

Na figura 2.10, são mostrados os gráficos de $\bar{\mathcal{Q}}(x) = \mathbb{E}[W(x + \Xi)]$ para diversos valores do desvio padrão. Na imagem da esquerda, os valores de σ diminuem as medidas de não-convexidade em relação às originais para a função W mas ainda não tornam a média totalmente convexa. Já na imagem da esquerda, a variância é grande o suficiente para tornar $\mathbb{E}[W(\cdot + \Xi)]$ convexa.

A evolução das medidas de não-convexidade em função da variância quando $\Xi \sim N(0, \sigma^2)$ pode ser observada na figura 2.11, que novamente mostra que há um ponto a partir do qual as medidas de não-convexidade ficam nulas, ou seja, a função $\mathbb{E}[W(\cdot + \Xi)]$ se torna convexa.

2.5 Convexificação e ruídos discretos

Agora, abordaremos exemplos onde temos uma distribuição discreta de cenários. Este caso tem, em especial, uma motivação computacional: geralmente, só é possível obter estimativas para $\bar{\mathcal{Q}}(x) = \mathbb{E}[Q(x, \Xi)]$ através de uma discretização da variável aleatória Ξ em um número finito de cenários.

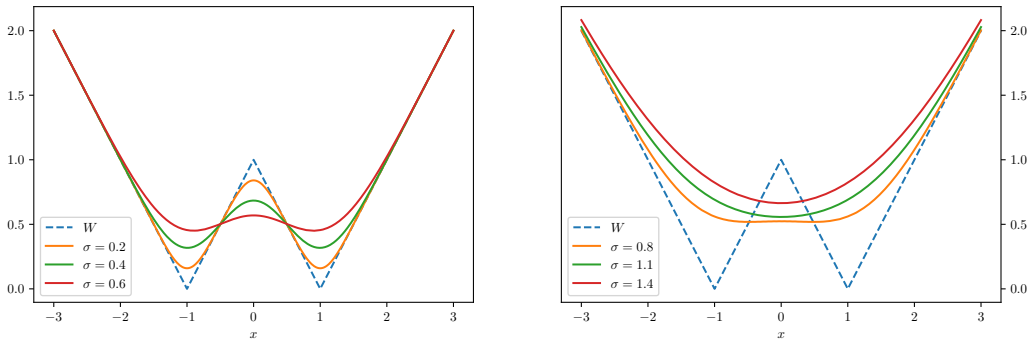


Figura 2.10: Gráfico de $\mathbb{E}\left[W(x+\Xi)\right]$ para $\Xi \sim N(0,\sigma^2)$.

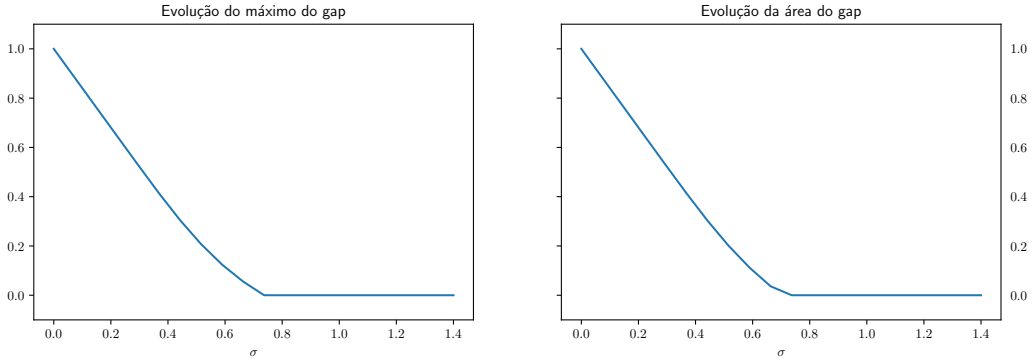


Figura 2.11: A área e o máximo do *gap* de $\mathbb{E}\left[W(\cdot+\Xi)\right]$ são decrescentes conforme a variância aumenta para uma variável aleatória $\Xi \sim N(0,\sigma^2)$.

Começamos considerando uma distribuição com apenas dois cenários equiprováveis: $\xi^1 = 0.5$ e $\xi^2 = -0.5$, para a mesma função f "em W" acima. Neste caso, as funções valor ótimo para cada cenário, Q_1 e Q_2, são translações uma da outra, como podemos ver na figura 2.12. Mais ainda, ao fazer a média de Q_1 e Q_2, obtemos a função de custo futuro, que será ainda linear por partes, mas que agora também é convexa.

Entretanto, este exemplo é muito particular. Na prática, é muito improvável que a função $\bar{\mathcal{Q}}$ resultante da média da função valor ótimo f sobre cenários discretos resulte em uma função convexa. Para vermos um exemplo deste caso, basta considerarmos, para a mesma função f acima, os ruídos $\xi^1 = 1.5$ e $\xi^2 = -1.5$, como vemos na figura 2.13

Uma outra diferença com relação ao caso contínuo ocorre ao tratarmos do fenômeno de "espalhamento". De fato, vimos nas figuras 2.8 a 2.11 que, ao aumentar a variância das distribuições, também reduzíamos a não-convexidade. Este fenômeno ocorre, mas de forma reduzida, para variáveis aleatórias discretas: em particular, mesmo que o suporte dos ruídos seja arbitrariamente grande, a função de custo futuro formada pela média não será, necessariamente convexa.

Para ilustrar o que foi dito acima, consideremos Ξ sendo uma variável aleatória equiprovável no conjunto $\{-h, 0, h\}$. Tomamos os mesmos valores que usamos no caso de variáveis uniformes em $[-h, h]$, ou seja, $h \in \{0.5, 1, 1.5\}$. Agora, as diferentes funções $\bar{\mathcal{Q}}$ produzidas estão ilustradas na figura 2.14, e podemos ver que, ao tomar $h = 1$ de fato obtemos uma função convexa, mas para um valor maior ainda de h, 1.5, a função volta a ser não-convexa. Este fenômeno é diferente

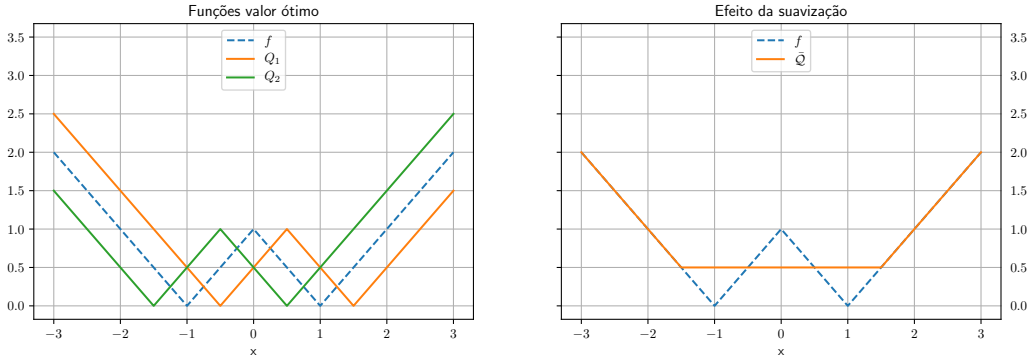


Figura 2.12: As funções Q_1 e Q_2, e sua média $\bar{\mathcal{Q}}$, convexa.

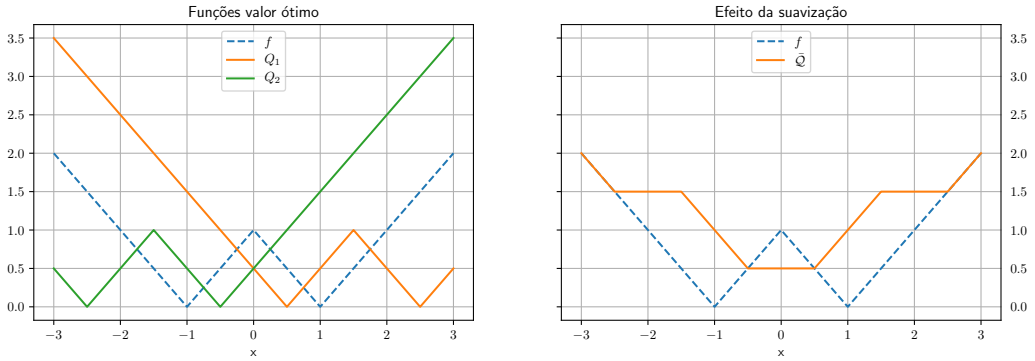


Figura 2.13: As funções Q_1 e Q_2, e sua média $\bar{\mathcal{Q}}$, não convexa.

do caso das distribuições contínuas, seja uniformes ou gaussianas, ilustrados nas figuras 2.5 e 2.6, onde a terceira distribuição, com maior "amplitude", também levava a uma função de custo futuro convexa.

Para observarmos este comportamento de "convexificação e des-convexificação" da função de custo futuro $\bar{\mathcal{Q}}$ em relação à mudança da incerteza Ξ, apresentamos na figura 2.15 uma maior quantidade de funções $\bar{\mathcal{Q}}$. Vemos na imagem da esquerda que a função se torna cada vez mais convexa conforme h se aproxima de 1. Contudo, para valores maiores de h a não-convexidade volta a aumentar até que se estabiliza para valores de h maiores do que 2, como pode ser visto na imagem da direita.

A figura 2.16 mostra a evolução das duas medidas de não-convexidade de $\mathbb{E}\left[W(\cdot + \Xi)\right]$ conforme o espaçamento entre os pontos da distribuição aumenta. Como anteriormente discutido, elas decrescem até $h = 1$, valor para o qual a função é convexa, e depois voltam a crescer até que se tornam constantes para valores de h maiores do que 2. Repare que, embora valores muito altos do espaçamento h não tornem $\mathbb{E}\left[W(\cdot + \Xi)\right]$ convexa, o seu *gap* ainda é estritamente menor do que o *gap* da função W original.

Não obstante, para distribuições discretas, há outra forma de se "aumentar o suporte": podemos refinar a discretização, utilizando mais amostras para uma mesma distribuição contínua subjacente. Assim, podemos ver na série de gráficos da figura 2.17, ilustrando discretizações cada vez mais refinadas de uma distribuição uniforme, que, apesar de não resultar em funções convexas, há uma clara redução do *gap*, principalmente quando o número de cenários aumenta. Este exemplo sugere

que, para os casos de planejamento da operação que consideram um número razoável de cenários discretos, é de se esperar que ocorra uma suavização da função de custo futuro com impactos promissores para o desempenho dos algoritmos de otimização em problemas não convexos.

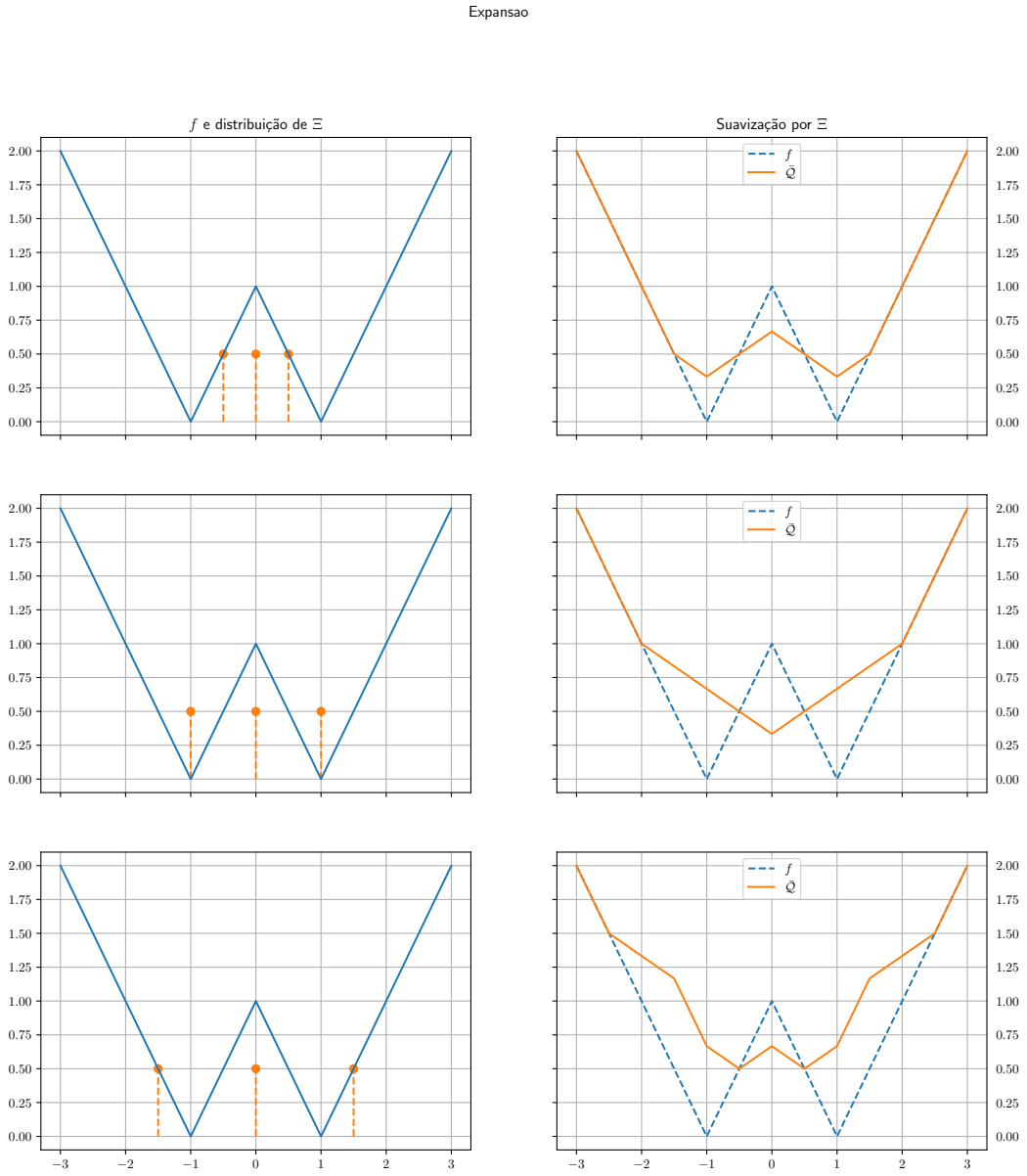


Figura 2.14: Médias da função “W” por diferentes distribuições equiprováveis discretas.

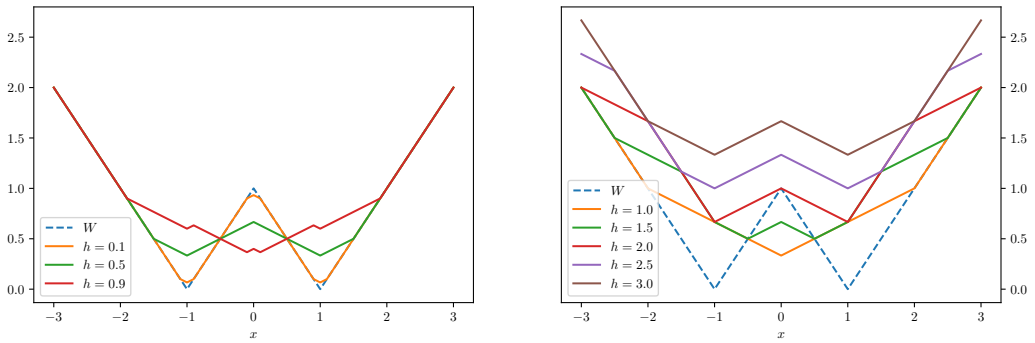


Figura 2.15: Gráfico de $\mathbb{E}\left[W(x+\Xi)\right]$ para Ξ discreta com suporte em $\{-h, 0, h\}$.

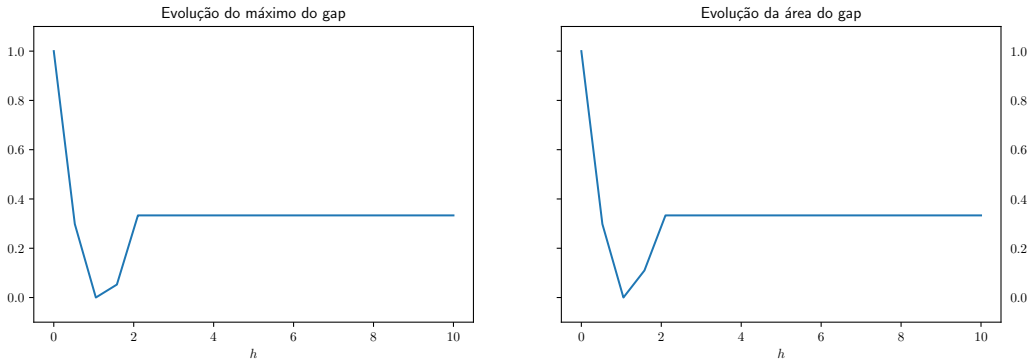


Figura 2.16: Para uma variável aleatória discreta com suporte em $\{-h, 0, h\}$, a convexificação não ocorre de maneira monótona no parâmetro h.

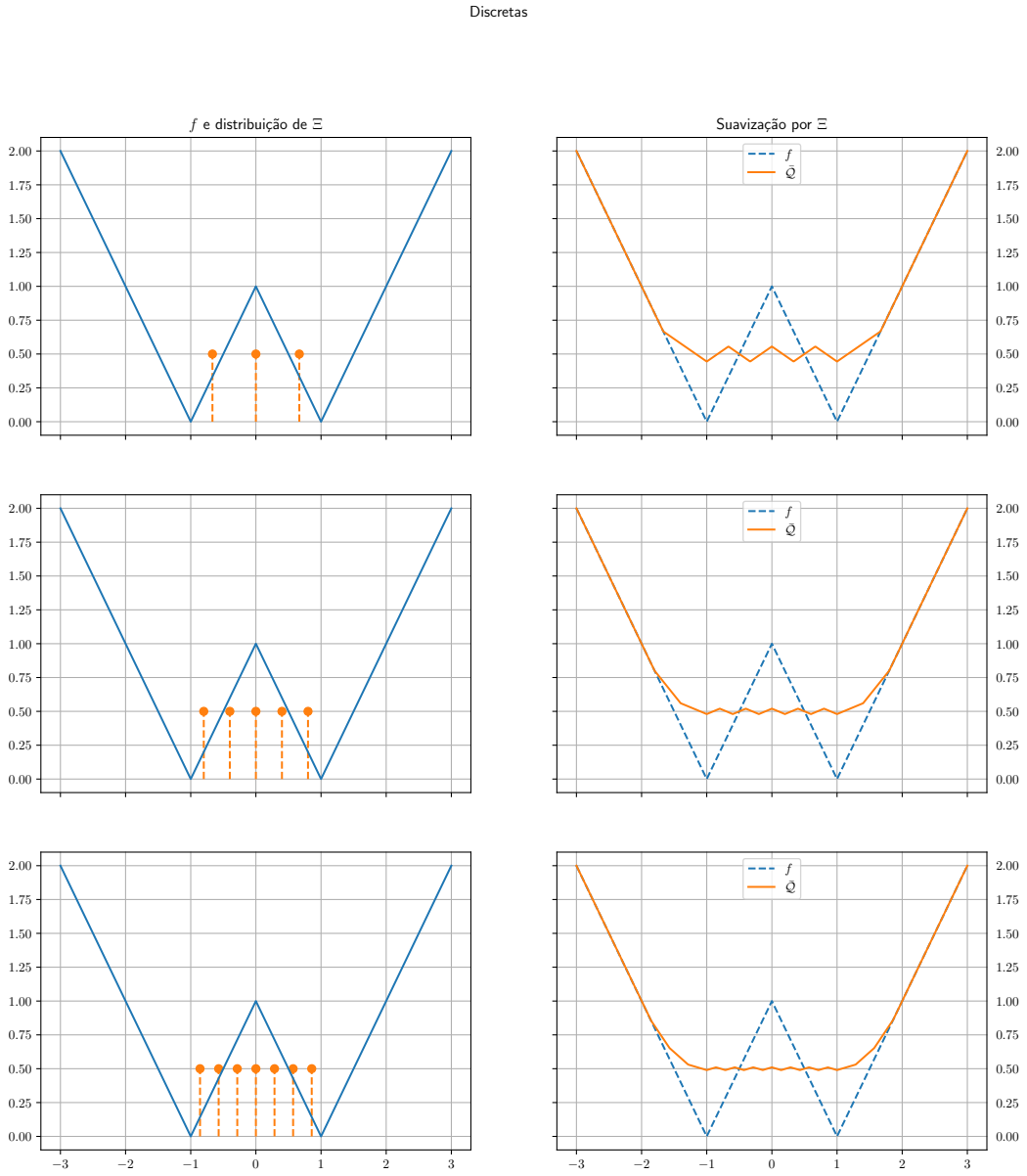


Figura 2.17: Redução da não convexidade da função “W” por diferentes distribuições discretas.

3 Redução do gap

Vimos na seção anterior alguns exemplos onde a estocasticidade do modelo agia na direção de reduzir a não-convexidade da função de custo futuro. Para tratar desta noção de forma mais precisa, precisamos definir uma forma de quantificar quão não-convexa é uma dada função. A partir destas definições, poderemos então avaliar se os efeitos estocásticos realmente tornam a função "menos" não convexa.

Como não há uma maneira canônica para medir não-convexidade, discutiremos primeiramente dois exemplos, construídos a partir da função f e sua relaxação convexa $\check{f}$. Em seguida, observaremos que estes dois exemplos satisfazem certas propriedades que, de forma geral, qualquer "medida de não-convexidade" deveria possuir. A partir daí, deduziremos alguns resultados sobre a redução da não-convexidade ao introduzir estocasticidades no modelo. Enfim, aproveitando as construções desenvolvidas, concluímos que, para outras medidas de não-convexidade que também parecem naturais, a estocasticidade também reduz a não-convexidade.

3.1 Relaxação convexa e gap

Dada uma função f qualquer, uma maneira de se medir a sua não-convexidade é olhando para o quão diferente ela é da função convexa que melhor a aproxima por baixo. Para isso, precisaremos da seguinte definição:

Definição 3.1. *A* relaxação convexa *de uma função f, denotada* $\operatorname{conv}(f)$ *e abreviada como $\check{f}$, é a maior função convexa que é menor do que ou igual a f.*

Repare que se f for convexa, então $f = \check{f}$.

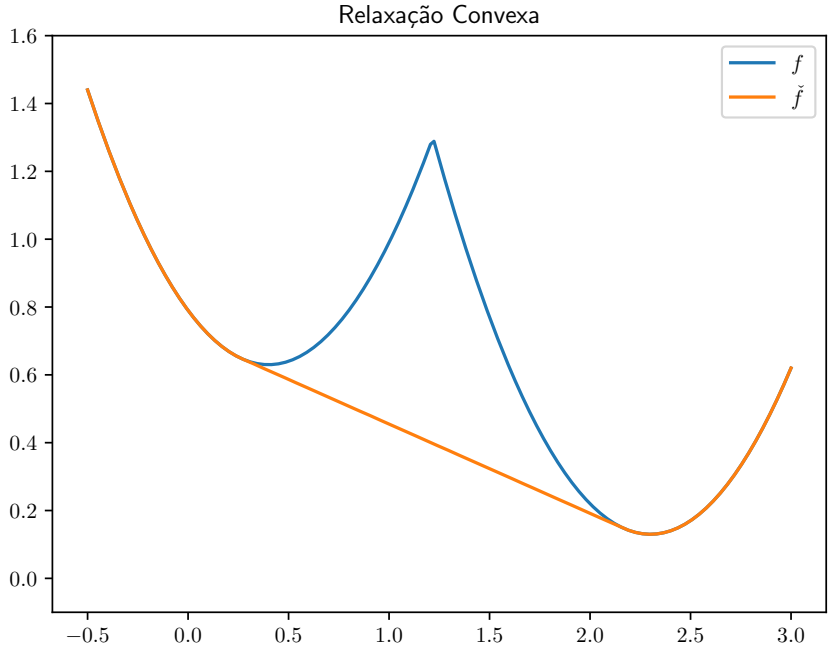


Figura 3.1: Exemplo de função não-convexa f e sua relaxação convexa $\check{f}$

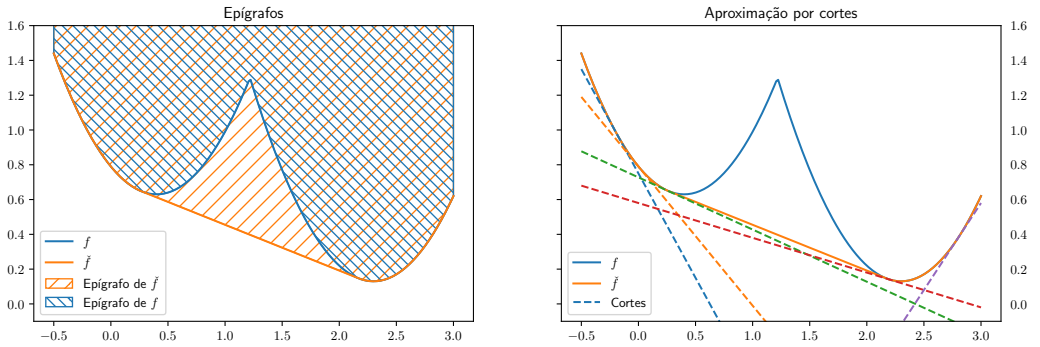


Figura 3.2: Epígrafo de $\check{f}$ e sua aproximação por cortes

Uma interpretação geométrica da relaxação convexa $\check{f}$ é que ela é a função cujo epígrafo é o fecho convexo do epígrafo de f. Esse epígrafo, sendo um conjunto convexo, pode ser calculado como a região acima de todas as funções lineares que estão abaixo de f, como ilustrado na figura 3.2. De volta ao ponto de vista das funções, isto quer dizer que, se calcularmos cortes lineares para a função f, estaremos formando uma função poliedral que aproxima por baixo a função $\check{f}$. Assim, uma maneira de se calcular $\check{f}$ é como solução do seguinte problema de otimização:

$$\begin{aligned} \check{f}(x_0) = \max_{c,b} \quad & c^\top x_0 + b. \\ \text{s.a} \quad & f(x) \geqslant c^\top x + b,\ \forall x \end{aligned}$$

Dada essa definição, podemos representar o quão não-convexa é uma função f a partir da sua diferença com sua relaxação convexa $\check{f}$. Ou seja, $f - \check{f}$ nos diz o tamanho do *gap* entre os gráficos da f e sua relaxação convexa.

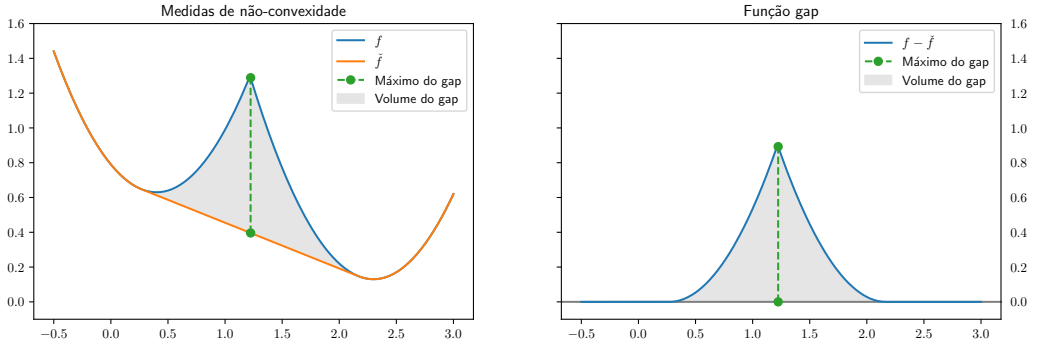


Figura 3.3: Medidas de não-convexidade e gráfico da função *gap* para a função da figura 3.1.

Definição 3.2. *Chamamos de* função gap *de f, a função definida por*

$$\text{gap}(f) = f - \check{f}.$$

Notemos que, como $\check{f}$ aproxima f por baixo, a função $\text{gap}(f)$ é sempre não-negativa e será igual a zero se e somente se f for convexa. O *gap* pode ser utilizado como uma maneira de medir pontualmente quão não-convexa é a função f. A partir disso, temos uma maneira de comparar a não-convexidade de duas funções.

Definição 3.3. *Uma função f é dita* menos não-convexa *que g se*

$$gap(f) \leqslant gap(g).$$

As comparações de não-convexidade da definição 3.3 dão apenas uma ordem parcial entre as funções, pois é possível que haja pontos nos quais f é menos não-convexa que g e pontos em que g seja menos não-convexa que f. Nesse caso as não-convexidades das funções não são comparáveis através apenas do *gap*. Uma solução para esse problema consiste em "projetar" o *gap*nos números reais atráves de algum método que preserve estas desigualdades. No que segue, daremos alguns exemplos de medidas de não-convexidade definadas a partir do *gap*:

1. A maior "altura" do *gap*, definida por $\left\|f - \check{f}\right\|_\infty = \sup_x \left|f(x) - \check{f}(x)\right|$;
2. O volume do *gap*, definido por $\left\|f - \check{f}\right\|_1 = \int \left|f(x) - \check{f}(x)\right| \, dx$.

Comentário 3.1. *Repare que o volume do gap é dado pelo volume da diferença entre os epígrafos de $\check{f}$ e f.*

Essas medidas de não-convexidade têm algumas características em comum. Primeiro, são *invariantes por translação*, o que é bem razoável se pensarmos no gráfico da função. Se transladarmos tanto horizontalmente quanto verticalmente uma função f, a forma da sua região de não-convexidade não muda. Adicionalmente, são *normas monótonas*, ou seja, a medida é calculada utilizando uma norma para qual dadas as funções g e h não-negativas, sempre vale que se g estiver uniformemente acima de h então a norma de g é maior do que a de h. De maneira mais formal:

$$\forall x,\, g(x) \geqslant h(x) \geqslant 0 \implies \|g\| \geqslant \|h\|$$

Tanto a maior altura quanto o volume do *gap* são casos específicos da chamada norma p da diferença $f - \check{f}$. A norma $\left\|f - \check{f}\right\|_p = \left(\int \left|f - \check{f}\right|^p \, dx\right)^{1/p}$ é uma norma monótona e invariante por translação para qualquer $p \in [1, \infty]$ e também pode ser utilizada nos cálculos de medida de não-convexidade. Contudo, com exceção dos casos em que $p = 1$ ou $p = \infty$, não há uma intuição geométrica tão clara do que significa $\left\|f - \check{f}\right\|_p$ diminuir.

3.2 Caso risco-neutro

Começamos recapitulando a notação para problemas de otimização estocásticos multi-estágio, sob a forma das equações de recorrência da programação dinâmica. Num dado estágio t, temos as funções de custo imediato $f_t(x_t, y_t)$, dependendo das decisões (x_t, y_t) correspondentes às variáveis de estado e variáveis locais, e a função de custo futuro $\bar{Q}_t(x_t)$, sujeitos às restrições de operação, dadas por $(x_{t-1}, x_t, y_t, \xi_t) \in X_t$, onde x_{t-1} representa o estado inicial, e ξ_t a incerteza observada no cenário correspondente. Assim, o problema de otimização em cada estágio é dado por:

$$\begin{aligned} Q_t(x_{t-1}, \xi_t) = \min_{x_t, y_t} \quad & f_t(x_t, y_t) + \bar{Q}_t(x_t) \\ \text{s.a} \quad & (x_{t-1}, x_t, y_t, \xi_t) \in X_t \end{aligned} \tag{3.1}$$

e a função de custo futuro satisfaz a recorrência:

$$\bar{Q}_{t-1}(x_{t-1}) = \mathbb{E}\big[Q_t(x_{t-1}, \Xi_t)\big] \tag{3.2}$$

onde $Q_t(x_{t-1}, \xi_t)$ é a função valor ótimo dados o estado inicial x_{t-1} e uma realização ξ_t da variável aleatória Ξ_t.

Se a função de valor ótimo Q_t for não-convexa, ao calcular cortes o melhor que podemos encontrar é uma aproximação inferior para a sua relaxação convexa. Relembramos que a relaxação convexa de uma função g é definida como a maior função convexa que a aproxima por baixo e é denotada por $\operatorname{conv}(g)$. Para resolver um problema de otimização estocástica como (3.1), a convexificação pode ser realizada de duas maneiras:

1. considerar a *formulação decomposta* do problema, que consiste em primeiro calcular a relaxação convexa da função Q_t para cada valor da incerteza e depois tomar a média, o que dá
$$\overline{\operatorname{conv}(Q)}_{t-1}(x_{t-1}) = \mathbb{E}\left[\operatorname{conv}(Q_t)(x_{t-1}, \Xi_t)\right];$$
2. considerar a *formulação conjunta*, na qual primeiro toma-se a média de Q_t entre todas as incertezas para obter a função de custo futuro $\bar{Q}_{t-1}$ e só depois calcular a sua relaxação convexa:
$$\operatorname{conv}(\bar{Q}_{t-1}) = \operatorname{conv}\mathbb{E}\left[Q_t(x_{t-1}, \Xi_t)\right].$$

A *formulação decomposta* é a mais comum de se encontrar na literatura por se encaixar melhor nos paradigmas de programação dinâmica normalmente utilizados para resolver esse tipo de problema. Contudo, demonstraremos no teorema 3.1 que a formulação conjunto sempre é uma aproximação convexa mais fiel da função de custo futuro. Isto é, sempre vale que

$$\mathbb{E}\left[\operatorname{conv} Q_t(x_{t-1}, \Xi_t)\right] \leqslant \operatorname{conv}\mathbb{E}\left[Q_t(x_{t-1}, \Xi_t)\right] \leqslant \mathbb{E}\left[Q_t(x_{t-1}, \Xi_t)\right] \tag{3.3}$$

para qualquer valor de x_{t-1}.

Para simplificar a notação, no restante dessa seção omitiremos os índices temporais tanto da função de custo futuro quanto das variáveis.

Teorema 3.1. *Seja $Q(x, \Xi)$ uma função aleatória com incerteza Ξ, então*

$$\mathbb{E}\left[\operatorname{conv} Q(x, \Xi)\right] \leqslant \operatorname{conv}\mathbb{E}\left[Q(x, \Xi)\right] \leqslant \mathbb{E}\left[Q(x, \Xi)\right].$$

Demonstração. Vale para cada realização ξ da variável aleatória Ξ que

$$\operatorname{conv}(Q)(x, \xi) \leqslant Q(x, \xi).$$

Essa desigualdade é preservada pela média o que dá,

$$\mathbb{E}\left[\operatorname{conv}(Q)(x, \Xi)\right] \leqslant \mathbb{E}\left[Q(x, \Xi)\right].$$

Como cada função $\operatorname{conv}(Q)$ é convexa, a média $\mathbb{E}\left[\operatorname{conv} Q\right]$ é uma função convexa que está abaixo de $\mathbb{E}\left[Q\right]$, e, portanto, também está abaixo de sua relaxação convexa $\operatorname{conv}\mathbb{E}\left[Q\right]$. Com isso, obtemos que

$$\mathbb{E}\left[\operatorname{conv} Q(x, \Xi)\right] \leqslant \operatorname{conv}\mathbb{E}\left[Q(x, \Xi)\right] \leqslant \mathbb{E}\left[Q(x, \Xi)\right]. \qquad \square$$

Note que como o teorema 3.1 vale pontualmente para cada valor da variável x, ele também pode ser interpretado como uma desigualdade funcional que diz que o gráfico de $\mathbb{E}\left[\operatorname{conv} Q\right]$ está abaixo do gráfico de $\operatorname{conv}\mathbb{E}\left[Q\right]$.

O teorema 3.1 diz que não importa o ruído ao qual um problema da forma (3.1) esteja sujeito, a relaxação convexa da média de Q está sempre acima da média da relaxação convexa da Q. Isso

permite levar em conta ruídos que mudem as matrizes tecnológicas ou qualquer outro parâmetro do modelo e também permite que consideremos árvores markovianas de decisão durante o processo de programação dinâmica. Para construir a formulação conjunta do nó de um problema de programação estocástica multiestágio, desenvolvemos a biblioteca `Relink`, cuja documentação se encontra em [...] e uma breve descrição pode ser lida na seção 8.2. Por conta da generalidade do teorema 3.1, essa biblioteca foi escrita de maneira a suportar uma grande gama de programas estocásticos, não se restringindo apenas àqueles considerados neste relatório.

Um corolário direto do teorema 3.1 diz que o *gap* da média de uma função aleatória é sempre menor que a média dos *gaps* para cada realização.

Corolário 3.1. *Seja $Q(x, \Xi)$ uma função aleatória, então*

$$\mathbb{E}\left[Q(x,\Xi)\right] - \operatorname{conv}\mathbb{E}\left[Q(x,\Xi)\right] \leqslant \mathbb{E}\left[Q(x,\Xi)\right] - \mathbb{E}\left[\operatorname{conv} Q(x,\Xi)\right]$$

ou, de maneira equivalente,

$$gap(\mathbb{E}\left[Q\right]) \leqslant \mathbb{E}\left[gap(Q)\right].$$

Demonstração. Do teorema 3.1, temos que para todo valor da variável x,

$$\mathbb{E}\left[\operatorname{conv} Q(x,\Xi)\right] \leqslant \operatorname{conv}\mathbb{E}\left[Q(x,\Xi)\right].$$

Subtraindo ambos esses termos de $\mathbb{E}\left[Q(x,\Xi)\right]$, chegamos ao resultado desejado.

Para a desigualdade com o *gap*, repare que por definição,

$$gap(\mathbb{E}\left[Q\right]) = \mathbb{E}\left[Q(x,\Xi)\right] - \operatorname{conv}\mathbb{E}\left[Q(x,\Xi)\right]$$

e que segue da linearidade do valor esperado que

$$\mathbb{E}\left[gap(Q)\right] = \mathbb{E}\left[Q(x,\Xi) - \operatorname{conv} Q(x,\Xi)\right] = \mathbb{E}\left[Q(x,\Xi)\right] - \mathbb{E}\left[\operatorname{conv} Q(x,\Xi)\right]. \qquad \square$$

Repare que a desigualdade para a função *gap*no corolário 3.1 é uma desigualdade funcional, valendo para cada valor da variável de estado x. Portanto, podemos aplicar qualquer norma monótona, como aquelas apresentadas anteriormente como exemplos de medidas de não-convexidade, e esses resultados ainda se manterão.

Corolário 3.2. *Seja $Q(x, \Xi)$ uma função aleatória e $\|\cdot\|$ uma norma monótona, então*

$$\left\|\mathbb{E}\left[Q(x,\Xi)\right] - \operatorname{conv}\mathbb{E}\left[Q(x,\Xi)\right]\right\| \leqslant \left\|\mathbb{E}\left[Q(x,\Xi)\right] - \mathbb{E}\left[\operatorname{conv} Q(x,\Xi)\right]\right\|.$$

Demonstração. O resultado segue do corolário 3.1 e do fato de que normas monótonas preservam desigualdades. $\square$

3.2.1 Ruído aditivo e redução do gap

Do que vimos nos exemplos e considerações acima, é bastante natural exigir que qualquer medida de não convexidade satisfaça pelo menos duas condições: a invariância por translação e a monotonicidade. Além disso, nos exemplos que construímos, o cálculo desta medida era efetuado através de uma norma, que tem um apelo intuitivo como uma forma de medir objetos. Assim, podemos obter diversas medidas de não convexidade para f, calculando $\left\|f - \check{f}\right\|$ a partir de uma *norma monótona e invariante por translação* aplicada à função *gap* de f. Como provaremos no teorema 3.2, para qualquer medida de não convexidade construída desta forma, tomar médias por translação sempre reduz a medida de não-convexidade, o que dá uma primeira explicação para a redução de não convexidade apresentada na seção 2. Antes de enunciá-lo, e com o intuito de simplificar a notação nos resultados e demonstrações mais adiante, definiremos uma notação para translação de funções por um parâmetro.

Definição 3.4. *Dada uma função f e um valor $\xi \in \mathbb{R}^n$, definimos a função f_ξ, que é a* translação de f por ξ. *Ou seja*

$$f_\xi(x) = f(x + \xi).$$

No caso em que Ξ não é um valor mas sim uma variável aleatória, definimos analogamente f_Ξ, que será uma função aleatória.

O objeto que irá aparecer no restante dessa seção é sempre a função $\mathbb{E}[f_\Xi]$ que para cada valor de x retorna o valor esperado de f ao redor de x, isto é, $\mathbb{E}[f(x + \Xi)]$. Em relação ao que foi apresentado na seção 2, $\mathbb{E}[f_\Xi]$ corresponde à função de custo futuro $\bar{Q}$.

Teorema 3.2. *Se $\|\cdot\|$ é uma norma monótona e invariante por translação, então*

$$\left\|f - \check{f}\right\| \geqslant \left\|\mathbb{E}[f_\Xi] - \mathbb{E}\left[\check{f}_\Xi\right]\right\| \geqslant \left\|\mathbb{E}[f_\Xi] - \widetilde{\mathbb{E}[f_\Xi]}\right\|. \tag{3.4}$$

Demonstração. Como a norma é invariante por translação,

$$\left\|f - \check{f}\right\| = \left\|f_\xi - \check{f}_\xi\right\|$$

para qualquer realização ξ de Ξ. Portanto possuem o mesmo valor esperado:

$$\mathbb{E}\left[\left\|f - \check{f}\right\|\right] = \mathbb{E}\left[\left\|f_\Xi - \check{f}_\Xi\right\|\right].$$

Se aplicarmos a desigualdade de Jensen em conjunto com esse fato, obtemos a primeira desigualdade do teorema:

$$\left\|f - \check{f}\right\| = \mathbb{E}\left[\left\|f - \check{f}\right\|\right] = \mathbb{E}\left[\left\|f_\Xi - \check{f}_\Xi\right\|\right] \geqslant \left\|\mathbb{E}\left[f_\Xi - \check{f}_\Xi\right]\right\|.$$

Para a segunda desigualdade, como $\check{f}$ é convexa, então a média de suas translações também o será. Além disso, a relaxação lagrangiana é definida como a *maior* função convexa sempre abaixo da função. A partir daí obtemos que

$$\check{f}_\xi \leqslant f_\xi \implies \mathbb{E}\left[\check{f}_\Xi\right] \leqslant \mathbb{E}[f_\Xi] \implies \mathbb{E}\left[\check{f}_\Xi\right] \leqslant \widetilde{\mathbb{E}[f_\Xi]}.$$

Unindo esse resultado à monotonicidade de $\|\cdot\|$, obtemos a segunda parte do teorema:

$$f - \mathbb{E}\left[\check{f}_\Xi\right] \geqslant f - \widetilde{\mathbb{E}[f_\Xi]} \implies \left\|f - \mathbb{E}\left[\check{f}_\Xi\right]\right\| \geqslant \left\|f - \widetilde{\mathbb{E}[f_\Xi]}\right\|. \qquad \square$$

Comentário 3.2. *No caso de a medida de não convexidade ser $\left\|f - \check{f}\right\|_1 = \int f(x) - \check{f}(x)\,dx$, a primeira desigualdade em* (3.4) *na verdade é uma igualdade*

$$\left\|f - \check{f}\right\|_1 = \left\|\mathbb{E}[f_\Xi] - \mathbb{E}\left[\check{f}_\Xi\right]\right\|_1. \tag{3.5}$$

Isso segue de podermos trocar a ordem do valor esperado e da integral (teorema de Fubini) e da invariância por translação da norma $\|\cdot\|_1$:

$$\int \mathbb{E}\left[f_\Xi - \check{f}_\Xi\right] dx = \mathbb{E}\left[\int f_\Xi - \check{f}_\Xi\, dx\right] = \mathbb{E}\left[\int f - \check{f}\, dx\right] = \int f - \check{f}\, dx.$$

Portanto a convexificação, medida pela norma 1, depende apenas da folga na desigualdade

$$\int \mathbb{E}[f_\Xi] - \mathbb{E}\left[\check{f}_\Xi\right] dx \geqslant \int \mathbb{E}[f_\Xi] - \widetilde{\mathbb{E}[f_\Xi]}\, dx$$

e do quanto $\mathbb{E}\left[\check{f}_\Xi\right]$ é menor do que $\widetilde{\mathbb{E}[f_\Xi]}$.

Em contraste com a igualdade do comentário 3.2, um resultado totalmente diferente é obtido quando trabalhamos com o máximo do *gap* como medida de não-convexidade. Nesse caso, a desigualdade $\left\|\mathbb{E}\left[f_\Xi\right] - \mathbb{E}\left[\check{f}_\Xi\right]\right\|_\infty \leqslant \left\|f - \check{f}\right\|_\infty$ não apenas é estrita como também podemos garantir uma estimativa na folga em função da variável aleatória Ξ e do suporte da função *gap*. Portanto, no caso da norma do supremo, deduziremos um resultado ainda melhor para a convexificação. Com esse intuito, introduziremos a noção de domínio de não-convexidade de uma função, que corresponde à região onde a função f difere do sua relaxação convexa.

Definição 3.5. *O domínio de* não-convexidade *de uma função f é o conjunto dos pontos em que f e sua relaxação convexa diferem. Isso pode ser escrito como o conjunto dos pontos em que $f(x) - \check{f}(x)$ é diferente de zero:*

$$DNC(f) = \left\{x \mid f(x) - \check{f}(x) \neq 0\right\}.$$

Comentário 3.3. *Observe que o domínio de não-convexidade é o suporte da função gap, como pode ser visto na figura 3.4.*

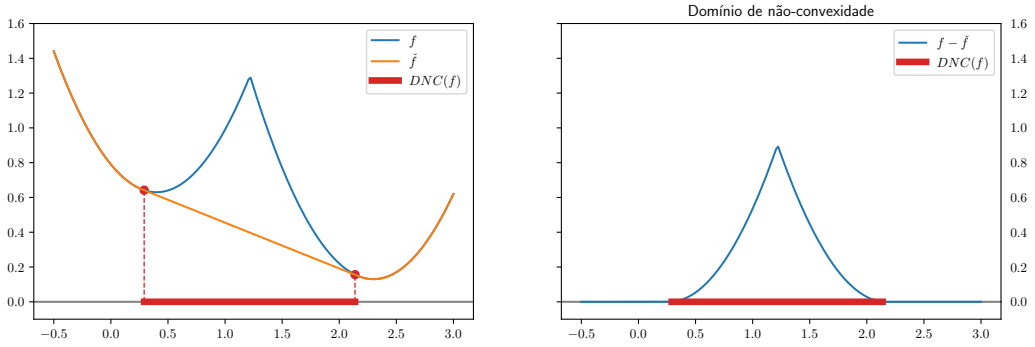


Figura 3.4: Domínio de não convexidade para a função da figura 3.1

Teorema 3.3. *Sejam f uma função, K o seu domínio de não-convexidade e chamemos*

$$\rho = \max_x \mathsf{Pr}\left(x + \Xi \in K\right) \leqslant 1.$$

Então

$$\left\|\mathbb{E}\left[f_\Xi\right] - \widetilde{\mathbb{E}\left[f_\Xi\right]}\right\|_\infty \leqslant \left\|\mathbb{E}\left[f_\Xi\right] - \mathbb{E}\left[\check{f}_\Xi\right]\right\|_\infty \leqslant \rho \left\|f - \check{f}\right\|_\infty.$$

Demonstração. Como $f - \check{f}$ é suportada em K, para cada ponto $x \in \mathbb{R}^n$ fixado a função $f(x+\Xi) - \check{f}(x+\Xi)$ será não-nula apenas quando $(x+\Xi)$ pertencer a K. Mais formalmente, se denotarmos por $\mathbb{1}_K$ a função indicadora do conjunto K, ou seja a função que vale um em K e zero fora dele, escrevemos

$$\mathbb{E}\left[f(x+\Xi) - \check{f}(x+\Xi)\right] = \mathbb{E}\left[\left(f(x+\Xi) - \check{f}(x+\Xi)\right) \cdot \mathbb{1}_K(x+\Xi)\right].$$

Podemos majorar $f - \check{f}$ em qualquer ponto pelo seu máximo em todo o $\mathbb{R}^n$. Ou seja, substituímos a desigualdade

$$\mathbb{E}\left[f_\Xi - \check{f}_\Xi\right] \leqslant \mathbb{E}\left[\left\|f - \check{f}\right\|_\infty \cdot \mathbb{1}_K(x+\Xi)\right] = \left\|f - \check{f}\right\|_\infty \mathbb{E}\left[\mathbb{1}_{K-x}\right] = \left\|f - \check{f}\right\|_\infty \mathsf{Pr}(x + \Xi \in K).$$

Como as funções nessas desigualdades são todas não-negativas, a norma $\|\cdot\|_\infty$ será igual ao máximo na variável x. Dessa forma, concluímos o teorema:

$$\left\|\mathbb{E}\left[f_\Xi\right] - \mathbb{E}\left[\check{f}_\Xi\right]\right\|_\infty = \max_x \mathbb{E}\left[f_\Xi - \check{f}_\Xi\right] \leqslant \left\|f - \check{f}\right\|_\infty \max_x \mathsf{Pr}(x + \Xi \in K) = \rho\left\|f - \check{f}\right\|_\infty. \qquad \square$$

Uma aplicação importante desse resultado é o caso em que o suporte de $f - \check{f}$ é compacto e a variável aleatória ξ tem uma densidade bastante "espalhada", sem dar probabilidade alta para nenhuma translação de K. Isso implica que a constante ρ, do teorema 3.3 será pequena e portanto o máximo do *gap* diminuirá bastante.

Para o caso específico em que Ξ é uma variável aleatória uniforme, o teorema 3.4 fornece um resultado assintótico que diz que conforme o suporte da Ξ cresce, o *gap* tende uniformemente a zero.

Teorema 3.4. *Seja f uma função cuja função gap associada é integrável, isto é $\left\|f - \check{f}\right\|_1$ é finita, e seja $\Xi_h \sim U(-h, h)$ uma variável aleatória uniforme. Então, no limite quando h tende a infinito, a diferença $\mathbb{E}\left[f_{\Xi_h}\right] - \mathbb{E}\left[\check{f}_{\Xi_h}\right]$ tende uniformemente a zero. De maneira mais formal:*

$$\left\|\mathbb{E}\left[f_{\Xi_h}\right] - \widetilde{\mathbb{E}\left[f_{\Xi_h}\right]}\right\|_\infty \leqslant \left\|\mathbb{E}\left[f_{\Xi_h}\right] - \mathbb{E}\left[\check{f}_{\Xi_h}\right]\right\|_\infty \overset{h\to\infty}{\longrightarrow} 0.$$

Demonstração. Dado qualquer $x \in \mathbb{R}$, escrevemos a expressão para a média como uma integral em $[-h, h]$

$$\left(\mathbb{E}\left[f_{\Xi_h} - \check{f}_{\Xi_h}\right]\right)(x) = \frac{1}{2h}\int_{[-h,h]} f(x+\xi) - \check{f}(x+\xi)\, d\xi$$

Fazendo a mudança de variáveis $\eta = x + \xi$, o valor desta função no ponto x será a média da função gap de f no intervalo de raio h ao redor de x:

$$\begin{aligned}\left(\mathbb{E}\left[f_{\Xi_h} - \check{f}_{\Xi_h}\right]\right)(x) &= \frac{1}{2h}\int_{[x-h,x+h]} f(\eta) - \check{f}(\eta)\, d\eta \\ &\leqslant \frac{1}{2h}\int_{\mathbb{R}} f(\eta) - \check{f}(\eta)\, d\eta \\ &= \frac{1}{2h}\left\|f - \check{f}\right\|_1\end{aligned}$$

Como o lado direito independe do ponto escolhido, essa desigualdade vale para o supremo entre todos os pontos.

$$\sup_{x\in\mathbb{R}}\left(\mathbb{E}\left[f_{\Xi_h} - \check{f}_{\Xi_h}\right]\right)(x) \leqslant \frac{1}{2h}\left\|f - \check{f}\right\|_1 \overset{h\to\infty}{\longrightarrow} 0 \qquad \square$$

3.3 Caso risco-avesso

Na equação (3.2), a função de custo futuro foi definida como a média do custo de cada cenário possível no próximo estágio. Para modelar aversão ao risco, substituímos o valor esperado nessa definição por alguma outra medida de risco ρ,

$$\bar{Q}_{t-1}(x_{t-1}) = \rho\big(Q_t(x_{t-1}, \Xi_t)\big). \tag{3.6}$$

No que segue, consideraremos apenas uma classe especial de medidas de risco ditas *coerentes*. Como exemplos destas, podemos citar o próprio valor esperado, o máximo entre todos os cenários e o CVaR.

Definição 3.6. *Uma* medida de risco coerente *é uma função ρ definida sobre as variáveis aleatórias satisfazendo*

- *Monotonicidade: Se $X \leqslant Y$ então $\rho(X) \leqslant \rho(Y)$;*
- *Equivariância por translação: Dada qualquer constante a, $\rho(X + a) = \rho(X) + a$;*
- *Convexidade: Dado qualquer $\lambda \in [0, 1]$,*

$$\rho\big(\lambda X + (1 - \lambda)Y\big) \leqslant \lambda\rho(X) + (1 - \lambda)\rho(Y);$$

- *Homogeneidade: Dada qualquer constante $t \geqslant 0$,*

$$\rho(tX) = t\rho(X).$$

As propriedades necessárias para uma medida de risco ser coerente as tornam muito parecidas com o valor esperado quando utilizadas em problemas de otimização. De fato, essas medidas de risco podem ser representadas como o supremo do valor esperado em relação a alguma família de probabilidades

$$\rho(X) = \sup_{\mu \in \mathcal{P}} \mathbb{E}_\mu [X] . \tag{3.7}$$

A equação acima é conhecida como a *representação dual* para medidas de risco coerentes e a família de probabilidades $\mathcal{P}$ define unicamente a medida de risco ρ.

A seguir, demonstramos que todas esses medidas de risco satisfazem um resultado similar ao teorema 3.1 para convexificação da função de custo futuro.

Teorema 3.5. *Seja $Q(x, \Xi)$ uma função aleatória com incerteza Ξ e ρ uma medida de de risco coerente, então*

$$\rho(\operatorname{conv} Q(x, \Xi)) \leqslant \operatorname{conv} \rho(Q(x, \Xi)) \leqslant \rho(Q(x, \Xi)).$$

Demonstração. Vale para cada realização ξ da variável aleatória Ξ que

$$\operatorname{conv}(Q)(x, \xi) \leqslant Q(x, \xi).$$

Essa desigualdade é preservada pela aplicação da medida de risco ρ:

$$\rho(\operatorname{conv}(Q)(x, \Xi)) \leqslant \rho(Q(x, \Xi)).$$

Como cada função $\operatorname{conv}(Q)$ é convexa e ρ é convexa e monótona, a sua composição $\rho(\operatorname{conv} Q)$ é uma função convexa que está abaixo de $\rho(Q)$, e, portanto, também está abaixo de sua relaxação convexa $\operatorname{conv} \rho(Q)$. Com isso, obtemos que

$$\rho(\operatorname{conv} Q(x, \Xi)) \leqslant \operatorname{conv} \rho(Q(x, \Xi)) \leqslant \rho(Q(x, \Xi)). \qquad \square$$

Esse teorema também pode ser interpretado em termos do *gap* da medida de risco, como demonstrado no corolário abaixo. Nesse caso, como a medida de risco não é linear, um termo a mais aparece na desigualdade.

Corolário 3.3. *Seja $Q(x, \Xi)$ uma função aleatória com incerteza Ξ e ρ uma medida de de risco coerente, então*

$$\rho(Q(x, \Xi)) - \rho(\operatorname{conv} Q(x, \Xi)) \leqslant \rho(Q(x, \Xi)) - \operatorname{conv} \rho(Q(x, \Xi)) \leqslant \rho\big(Q(x, \Xi) - \operatorname{conv} Q(x, \Xi)\big).$$

Demonstração. Segue do teorema 3.5 que

$$\rho(\operatorname{conv} Q(x,\Xi)) \leqslant \operatorname{conv} \rho(Q(x,\Xi)).$$

A primeira desigualdade do enunciado é obtida ao subtrairmos ambos esses termos de $\rho(Q)$. Para a segunda desigualdade usamos o fato de que ρ é subaditiva. Daí temos que

$$\rho(Q) = \rho(Q - \check{Q} + \check{Q}) \leqslant \rho(Q - \check{Q}) + \rho(\check{Q}).$$

Reorganizando os termos,

$$\rho(Q) - \rho(\check{Q}) \leqslant \rho(Q - \check{Q}).$$

□

3.3.1 Ruído aditivo e medidas de risco

Da mesma maneiro que na seção 3.2.1, restringiremos nossa atenção a ruídos aditivos, isto é funções aleatórias com a forma

$$Q(x,\Xi) = f_\Xi(x) = f(x + \Xi).$$

No caso avesso ao risco, não existe um resultado análogo ao teorema 3.2 que seja válido para qualquer norma monótona e invariante por translação. Contudo, se restringirmos a nossa atenção à norma uniforme $\|\cdot\|_\infty$, a limitação superior pelo *gap* da função transladada permanece válida.

Teorema 3.6. *Dada uma medida de risco coerente ρ e uma função f, vale que*

$$\left\|\rho(f_\Xi) - \widetilde{\rho(f_\Xi)}\right\|_\infty \leqslant \left\|\rho(f_\Xi) - \rho(\check{f}_\Xi)\right\|_\infty \leqslant \left\|\rho(f_\Xi - \check{f}_\Xi)\right\|_\infty \leqslant \left\|f - \check{f}\right\|_\infty.$$

Demonstração. As duas primeiras desigualdades seguem do corolário 3.3 e do fato de que $\|\cdot\|_\infty$ é uma norma monótona.

Para a última desigualdade, utilizaremos a representação dual de ρ. Isto é, escreveremos

$$\rho(X) = \sup_{\mu\in\mathcal{P}} \mathbb{E}_\mu[X]$$

onde $\mathcal{P}$ é um conjunto de probabilidades que determina a medida de risco ρ. Também repare que como $f - \check{f}$ é não-negativa, segue da monotonicidade de ρ que

$$f_\Xi - \check{f}_\Xi \geqslant 0 \implies \rho\big(f_\Xi - \check{f}_\Xi\big) \geqslant \rho(0) = 0.$$

Portanto podemos usar que supremos comutam para escrever

$$\begin{aligned}
\left\|\rho\big(f_\Xi - \check{f}_\Xi\big)\right\|_\infty &= \sup_x \left|\rho\big(f_\Xi - \check{f}_\Xi\big)\right| \\
&= \sup_x \rho\big(f_\Xi - \check{f}_\Xi\big) \\
&= \sup_x \sup_{\mu\in\mathcal{P}} \mathbb{E}_\mu\left[f_\Xi - \check{f}_\Xi\right] \\
&= \sup_{\mu\in\mathcal{P}} \sup_x \mathbb{E}_\mu\left[f_\Xi - \check{f}_\Xi\right] \\
&= \sup_{\mu\in\mathcal{P}} \left\|\mathbb{E}_\mu\left[f_\Xi - \check{f}_\Xi\right]\right\|_\infty
\end{aligned}$$

Usando o fato de que $\|\cdot\|$ é convexa, podemos aplicar a desigualdade de Jensen à equação anterior. Além disso, a invariância por translação implica que

$$\begin{aligned}\left\|\rho\left(f_\Xi - \check{f}_\Xi\right)\right\|_\infty &\leqslant \sup_{\mu\in\mathcal{P}} \mathbb{E}_\mu\left[\left\|f_\Xi - \check{f}_\Xi\right\|_\infty\right] \\ &= \sup_{\mu\in\mathcal{P}} \mathbb{E}_\mu\left[\left\|f - \check{f}\right\|_\infty\right]\end{aligned}$$

Como o valor esperado de um valor determinístico é ele mesmo, concluímos que

$$\left\|\rho\left(f_\Xi - \check{f}_\Xi\right)\right\|_\infty \leqslant \left\|f - \check{f}\right\|_\infty. \qquad \square$$

4 Medidas de não-convexidade

4.1 Medidas de não-convexidade para funções diferenciáveis

Suponhamos, por ora, que f é uma função duas vezes diferenciável. Nesse caso, existe uma noção *pontual* de convexidade da qual podemos nos aproveitar. Uma função f duas vezes diferenciável é dita convexa no ponto x se

$$f''(x) \geqslant 0.$$

Assim, a *parte negativa* de f'' também dá informação sobre a não-convexidade da f. Para facilitar a notação, definiremos as partes positiva e negativa de uma função f como

$$f_+ = \max(f, 0), \quad f_- = -\min(f, 0),$$

que são duas funções não-negativas que satisfazem $f = f_+ - f_-$.

A partir daí, as mesmas normas da seção 3.2.1 podem ser usadas novamente, porém com $[f'']_-$ ao invés da diferença $f - \check{f}$. Alguns exemplos são:

1. O menor valor atingido pela segunda derivada da f: $\inf_x f''(x) = \sup_x [f'']_-(x)$.
2. O volume da região onde f'' é negativa: $\int [f'']_- \, dx$;

Os dois teoremas abaixo são resultados análogos aos do teorema 3.2 para o volume e valor mínimo da parte negativa da segunda derivada.

Teorema 4.1. *Seja $Q(x, \Xi)$ uma função aleatória duas vezes diferenciável, então*

$$\left[\mathbb{E}\left[Q\right]''\right]_- \leqslant \mathbb{E}\left[\left[Q''\right]_-\right].$$

Demonstração. Trocando a ordem de diferenciação com o valor esperado, temos que

$$\left[\mathbb{E}\left[Q(x,\Xi)\right]''\right]_- = \left[\mathbb{E}\left[Q''(x,\Xi)\right]\right]_-.$$

Agora reparemos que a parte negative é definida como o máximo de funções lineares, $[x]_- = \max\{-x, 0\}$, portanto é uma função convexa. Isso diz que podemos aplicar a desigualdade de Jensen à equação anterior e obtemos

$$\left[\mathbb{E}\left[Q(x,\Xi)\right]''\right]_- = \left[\mathbb{E}\left[Q''(x,\Xi)\right]\right]_- \leqslant \mathbb{E}\left[\left[Q''(x,\Xi)\right]_-\right].$$

Com isso, concluímos o teorema. $\square$

No caso de ruído aditivo, $Q(x, \Xi) = \mathbb{E}[f_\Xi]$, esse teorema pode ser modificado para dar uma estimativa em termos da função transladada f.

Teorema 4.2. *Seja f uma função duas vezes diferenciável e $\|\cdot\|$ uma norma monótona e invariante por translação. Então*

$$\left\|[\mathbb{E}[f_\Xi]'']_-\right\| \leqslant \|[f'']_-\|.$$

Demonstração. Da monotonicidade da norma $\|\cdot\|$, temos que esta última desigualdade é preservada. Usando isto e a convexidade da norma obtemos que

$$\left\|[\mathbb{E}[f_\Xi]'']_-\right\| \leqslant \|\mathbb{E}\left[[f''_\Xi]_-\right]\| \leqslant \mathbb{E}\left[\|[f''_\Xi]_-\|\right].$$

A invariância por translação e o fato de que a esperança de uma função determinística é ela mesma implicam que

$$\mathbb{E}\left[\|[f''_\Xi]_-\|\right] = \mathbb{E}\left[\|[f'']_-\|\right] = \|[f'']_-\|.$$

Com isso, concluímos o teorema. □

Um comentário importante a se fazer é que por mais que $\mathbb{E}[f_\Xi]$ tenda a ser uma função suave pelas propriedades discutidas em 2.2 e 2.3, não há nada que garanta que a função original f tenha alguma regularidade. Muito pelo contrário, as funções de valor ótimo que aparecem nas aplicações do setor elétrico normalmente são funções lineares por partes ou mínimos de funções convexas. A regularidade que estas funções apresentam é apenas serem contínuas.

Todavia, nesses casos menos regulares podemos estender a noção de segunda derivada de forma que ela ainda contenha informação sobre a não-convexidade da função f. Essa definição será dada na seção 5.2, onde construiremos o formalismo da teoria das distribuições. Veremos que a segunda derivada de uma função contínua é sempre uma *medida* e que, assim como funções, podemos decompô-la em uma parte positiva e outra negativa. Por mais que essa parte negativa não esteja necessariamente definida ponto a ponto, ela pode ser aplicada a conjuntos e nos dizer o quão não-convexa é a função em uma dada região do domínio.

A partir daí algumas noções se generalizam, como a ideia de volume e o resultado do teorema 4.2, que ainda é válido para funções apenas contínuas.

4.2 Definições e propriedades

Nas seções anteriores encontramos duas maneiras diferentes de se medir a não-convexidade de uma função f: o *gap* dado pela diferença entre f e a sua relaxação convexa $\check{f}$, válida para qualquer função, e a parte negativa da segunda derivada de f, medida válida apenas quando f é duas vezes diferenciável. Vimos também que essas duas medidas de não-convexidade são reduzidas da mesma forma quando tomamos médias. No que segue, estudamos as características em comum desses operadores e chegamos a uma definição geral de *medida de não-convexidade* para a qual as propriedades de convexificação são sempre válidas.

Veremos que esses dois operadores satisfazem um caso generalizado da desigualdade de Jensen, além de outras propriedades que os tornam bons candidatos para medir a não-convexidade de funções. Todavia, primeiramente precisamos passar por algumas definições.

Definição 4.1. *Um subconjunto K de um espaço vetorial V é dito um* cone convexo pontuado *se*

- *K é convexo;*
- *K contém todos os seus raios: se $x \in K$ então para todo $t \geqslant 0$, $tx \in K$;*

- *K não contém nenhuma reta. Isto é, se $x \in K$ e $-x \in K$ então $x = 0$.*

A importância destes conjuntos no estudo das medidas de não-convexidade está no fato de que todo cone convexo pontuado K induz uma ordem parcial nos elementos de V definida por

$$x \leqslant_K y \iff x - y \in K. \tag{4.1}$$

Toda ordem induzida por um cone é compatível com soma e multiplicação por escalar,

$$\begin{aligned} x \leqslant_K y &\implies x + v \leqslant_K y + v,\ \forall v \in V, \\ x \leqslant_K y &\implies tx \leqslant_K ty,\ \forall t \geqslant 0. \end{aligned}$$

Utilizando essas desigualdades cônicas, podemos estender a definição de convexidade para funções cujo codomínio são um espaço vetorial e não apenas a reta $\mathbb{R}$.

Definição 4.2. *Dado um cone $K \subset Y$, uma função $f\colon X \to Y$ é dita* convexa em relação a K *ou* K-convexa *se satisfaz, para todo $\lambda \in [0, 1]$,*

$$f(\lambda x + (1 - \lambda)y) \leqslant_K \lambda f(x) + (1 - \lambda)f(y).$$

Usando ordens cônicas, podemos chegar a uma definição geral das propriedades que uma medida de não-convexidade deve ter para que seja bem comportada em relação a problemas de otimização estocástica.

Definição 4.3. *Uma* medida de não convexidade *em um conjunto X de funções é um operador $\mathcal{M}\colon X \to Y$ cujo codomínio é um espaço munido de uma ordem cônica $\leqslant_K$ e que satisfaz as seguintes propriedades*

- *$\mathcal{M}(f)$ é zero se e somente se f é convexa;*
- *$\mathcal{M}(f) \geqslant_K 0$ para qualquer $f \in X$;*
- *$\mathcal{M}$ é convexa em relação a K.*

Teorema 4.3. *Seja Q uma função aleatória. Dada qualquer medida de não convexidade $\mathcal{M}$,*

$$\mathcal{M}(\mathbb{E}\left[Q\right]) \leqslant_K \mathbb{E}\left[\mathcal{M}(Q)\right].$$

Demonstração. Como $\mathcal{M}$ é K-convexa, ela satisfaz a desigualdade de Jensen para qualquer probabilidade. □

Teorema 4.4. *Seja $\mathcal{M}\colon X \to Y$ uma medida de não-convexidade e $\|\cdot\|$ uma norma monótona em relação a K. Então a função $\|\cdot\| \circ \mathcal{M}\colon X \to [0, \infty]$ também é uma medida de não-convexidade.*

Na seção 3.2.1, estudamos o caso particular de problemas com ruído aditivo e vimos que alguns resultados mais finos podiam ser obtidos ao consideramos medidas de não-convexidade invariantes por translação. No contexto da definição 4.3, isso equivale a nos restringirmos a medidas de não-convexidade que satisfaçam

$$\mathcal{M}(f_\xi) = \mathcal{M}(f) \tag{4.2}$$

para qualquer função f e translação ξ. Para essas medidas de não-convexidade, temos o seguinte teorema que diz que a não convexidade das médias de translações de uma função f nunca é maior que a não-convexidade da própria f.

Teorema 4.5. *Seja f uma função sujeita a um ruído aditivo Ξ e $\mathcal{M}$ uma medida de não-convexidade. Então*

$$\mathcal{M}(\mathbb{E}\left[f_\Xi\right]) \leqslant_K \mathbb{E}\left[\mathcal{M}(f_\Xi)\right] \leqslant_K \mathcal{M}(f).$$

Demonstração. Da desigualdade de Jensen segue que

$$\mathcal{M}(\mathbb{E}\left[f_\Xi\right]) \leqslant_K \mathbb{E}\left[\mathcal{M}(f_\Xi)\right]$$

e do fato de $\mathcal{M}$ ser invariante por translação, concluímos que

$$\mathbb{E}\left[\mathcal{M}(f_\Xi)\right] = \mathbb{E}\left[\mathcal{M}(f)\right] = \mathcal{M}(f). \qquad \square$$

Teorema 4.6. *Seja $\mathcal{M}$ uma medida de não convexidade cujo codomínio está contido num conjunto de funções. Então*

$$\left\|\mathcal{M}(\mathbb{E}\left[f_\Xi\right])\right\|_\infty \leqslant_K \left\|\mathbb{E}\left[\mathcal{M}(f_\Xi)\right]\right\|_\infty \leqslant_K \gamma \left\|\mathcal{M}(f)\right\|_\infty$$

onde γ é definida por

$$\gamma = \sup_x \mathsf{Pr}\big(x + \Xi \in \operatorname{supp}\mathcal{M}(f)\big) \leqslant 1.$$

Demonstração. Denotemos por $K = \operatorname{supp}\mathcal{M}(f)$ e por $\mathbb{1}_K$ a função indicadora deste conjunto. Podemos escrever

$$\left|\mathbb{E}\left[\mathcal{M}(f_\Xi)\right](x)\right| = \left|\mathbb{E}\left[\mathcal{M}(f_\Xi)(x) \cdot \mathbb{1}_K(x + \Xi)\right]\right|.$$

Majorando a $\mathcal{M}(f_\Xi)$ pela sua norma uniforme, ficamos com

$$\left|\mathbb{E}\left[\mathcal{M}(f_\Xi)\right](x)\right| \leqslant \left\|\mathcal{M}(f)\right\|_\infty \mathbb{E}\left[\mathbb{1}_K(x + \Xi)\right] = \left\|\mathcal{M}(f)\right\|_\infty \mathsf{Pr}\big(x + \Xi \in K\big).$$

Tomando-se o supremo de ambos os lados da desigualdade acima, obtemos que

$$\left\|\mathcal{M}(f)\right\|)\infty = \sup_x \left|\mathbb{E}\left[\mathcal{M}(f_\Xi)\right](x)\right| \leqslant \left\|\mathcal{M}(f)\right\|_\infty \sup_x \mathsf{Pr}\big(x + \Xi \in K\big).$$

Com isso provamos o teorema. $\square$

Teorema 4.7. *Seja $\mathcal{M}$ uma medida de não convexidade cujo codomínio está contido num conjunto de funções integráveis e seja μ a densidade da variável aleatória ξ. Se μ for uma função limitada,*

$$\left\|\mathcal{M}(\mathbb{E}\left[f_\Xi\right])\right\|_\infty \leqslant_K \left\|\mathbb{E}\left[\mathcal{M}(f_\Xi)\right]\right\|_\infty \leqslant_K \left\|\mu\right\|_\infty \left\|\mathcal{M}(f)\right\|_1.$$

Demonstração. Podemos escrever

$$\mathbb{E}\left[\mathcal{M}(f)(x + \Xi)\right] = \int \mathcal{M}(f)(x + \xi)\mu(\xi)\, d\xi.$$

Aplicando o valor absoluto de ambos os lados desta equação,

$$\left|\mathbb{E}\left[\mathcal{M}(f)(x + \Xi)\right]\right| = \left|\int \mathcal{M}(f)(x + \xi)\mu(\xi)\, d\xi\right| \leqslant \int \left|\mathcal{M}(f)(x + \xi)\right| \mu(\xi)\, d\xi.$$

Majorando μ pela sua norma uniforme,

$$\left|\mathbb{E}\left[\mathcal{M}(f)(x + \Xi)\right]\right| \leqslant \left\|\mu\right\|_\infty \int \left|\mathcal{M}(f)(x + \xi)\right|\, d\xi. = \left\|\mu\right\|_\infty \left\|\mathcal{M}(f)\right\|_1$$

Como o termo direita é uma constante, ele também majora o supremo do termo do lado esquerdo. Portanto,

$$\left\|\mathbb{E}\left[\mathcal{M}(f)\right]\right\|_\infty \leqslant \left\|\mu\right\|_\infty \left\|\mathcal{M}(f)\right\|_1. \qquad \square$$

5 Convexificação de funções poliedrais e ruídos discretos

Dada uma função f e um erro aditivo Ξ, temos sempre trabalhado diretamente com a variável aleatória Ξ e suas realizações. Contudo, o objeto pelo qual nos interessamos são as médias $\mathbb{E}\left[f_\Xi\right](x)$, que só dependem da probabilidade com que cada realização ξ ocorre e não da variável aleatória em si.

Suponhamos por um momento que Ξ tem uma distribuição contínua. Então ela induz uma densidade de probabilidade μ no $\mathbb{R}^n$. Nesse caso, podemos escrever a expressão para a média como dependendo apenas de μ:

$$\mathbb{E}\left[f_\Xi\right](x) = \int f(x+\xi)\mu(x)\,d\xi. \tag{5.1}$$

A princípio, essa fórmula não parece nos dar nenhuma informação adicional sobre $\mathbb{E}\left[f_\Xi\right]$, contudo vamos manipulá-la para que fique mais simétrica entre f e μ. Chamando $\tilde{\mu}(x) = \mu(-x)$ e fazendo a mudança de variáveis $\eta = x + \xi$ obtemos

$$\mathbb{E}\left[f_\Xi\right](x) = \int f(\eta)\mu(\eta - x)\,d\eta = \int f(\eta)\tilde{\mu}(x-\eta)\,d\eta.$$

Por outro lado, se a mudança de variáveis escolhida for $\eta = -\xi$, a fórmula obtida é

$$\mathbb{E}\left[f_\Xi\right](x) = \int f(x-\eta)\tilde{\mu}(\eta)\,d\eta.$$

As fórmulas acima dão os primeiros indícios da simetria que existe em $\mathbb{E}\left[f_\Xi\right](x)$ em relação a f e $\tilde{\mu}$. Veremos nessa seção que o fato de podermos escolher em qual das duas funções dentro da integral o argumento x de $\mathbb{E}\left[f_\Xi\right]$ aparece nos permitirá deduzir que $\mathbb{E}\left[f_\Xi\right]$ mantém diversas características boas de *ambas* as funções, como diferenciabilidade e convexidade.

Estudaremos abaixo as propriedades da fórmula $\int f(\eta)\tilde{\mu}(x-\eta)\,d\eta$, para a qual damos o nome de *convolução*. Nessa seção, também escreveremos sempre $f * \tilde{\mu}$ ao invés de $\mathbb{E}\left[f_\Xi\right]$ para indicar que só dependemos da densidade μ em tudo que virá.

5.1 Convoluções de funções

Definição 5.1. *Dadas duas funções f e g, damos o nome de* convolução *entre f e g à função $f * g$ definida por*

$$(f * g)(x) = \int f(x-\eta)g(\eta)\,d\eta.$$

A partir dessa definição, podemos enumerar algumas propriedades da convolução que a tornam um objeto muito similar a um produto entre funções.

1. Comutatividade: $f * g = g * f$;
2. Associatividade: $f * (g * h) = (f * g) * h$;
3. Linearidade: Dado $\lambda \in \mathbb{R}$, sempre vale $(\lambda f + g) * h = \lambda(f * h) + g * h$.

Demonstração. Para provarmos essas propriedades, primeiro reparemos que para qualquer ponto x podemos fazer a mudança de variáveis $\tau = x - \eta$. Com isso obtemos que a convolução $f * g$ é simétrica entre f e g:

$$(f * g)(x) = \int f(x-\eta)g(\eta)\,d\eta = \int f(\tau)g(x-\tau)\,d\tau = (g * f)(x).$$

A associatividade vem do teorema de Fubini, que nos permite trocar a ordem entre as integrais abaixo:

$$\begin{aligned}
[f * (g * h)](x) &= \int f(x-y) \left[\int g(y-\eta) h(\eta)\, d\eta \right] dy \\
&= \int f(x-y) g(y-\eta) h(\eta)\, d\eta dy \\
&= \int \left[\int f(x-y) g(y-\eta) dy \right] h(\eta)\, d\eta \\
&= \int \left[\int f((x-\eta)-z) g(z) dz \right] h(\eta)\, d\eta \\
&= \int (f * g)(x-\eta) h(\eta)\, d\eta \\
&= [(f * g) * h](x).
\end{aligned}$$

Por fim, como a integral é linear, a convolução herda essa característica:

$$\begin{aligned}
(\lambda f + g) * h(x) &= \int [\lambda f(x-\eta) + g(x-\eta)]\, h(\eta)\, d\eta \\
&= \lambda \int f(x-\eta) h(\eta)\, d\eta + \int g(x-\eta) h(\eta)\, d\eta \\
&= \lambda f * h(x) + g * h(x)
\end{aligned}$$

□

Comentário 5.1. *Por conta da associatividade, escreveremos $f * g * h$ sem nos preocupar com parênteses.*

Retornado à motivação das variáveis aleatórias, se tivermos a aplicação sucessiva de duas incertezas com densidades μ e ρ, a densidade de sua soma será a convolução entre as duas densidades: $\mu * \rho$. Nesse contexto, a comutatividade e associatividade representam o fato de que, para o resultado final, não importa a ordem em que as incertezas são somadas. Abaixo formalizamos o significado disso.

Teorema 5.1. *Sejam f uma função, Υ e Ξ duas variáveis aleatórias com distribuições contínuas cujas densidades são respectivamente ρ e μ. Então*

$$\mathbb{E}\left[f_{\Upsilon+\Xi}\right] = f * \tilde{\rho} * \tilde{\mu}.$$

Demonstração. Primeiro reparemos que, pelo teorema de Fubini, podemos iterar a média entre diferentes variáveis aleatórias:

$$\mathbb{E}\left[f_{\Upsilon+\Xi}\right] = \mathbb{E}_{\Upsilon}\left[\mathbb{E}_{\Xi}\left[f_{\Upsilon+\xi} \mid \Xi = \xi\right]\right].$$

Fixando uma realização ξ de Ξ, temos

$$\mathbb{E}\left[f(x+\Upsilon+\xi)\right] = f * \tilde{\rho}(x+\xi) = \left[f * \tilde{\mu}\right]_{\xi}(x).$$

Se agora tirarmos a média entre todas as realizações ξ, conseguimos o resultado desejado:

$$\mathbb{E}\left[f_{\Xi+\Upsilon}\right] = \mathbb{E}\left[\left[f * \tilde{\rho}\right]_{\xi}\right] = (f * \tilde{\rho}) * \tilde{\mu}.$$

□

No nosso caso, outro resultado importante é que a convolução com uma função não-negativa preserva convexidade. Para o problema de convexificação de funções de custo futuro, no qual estamos interessados, a convolução sempre ocorre entre uma função f e a densidade μ da variável aleatória Ξ, que sempre é não-negativa. Com o teorema 5.2, veremos que se a função f já for convexa, então a média $\mathbb{E}[f_\Xi]$ também o será.

Teorema 5.2 (Convoluções preservam convexidade)**.** *Seja f uma função convexa e ϕ não-negativa. Então $f * \phi$ também é convexa.*

Demonstração. Dados dois pontos x, y e um número $\lambda \in [0, 1]$ segue da convexidade da f que para qualquer outro ponto z:

$$f(\lambda x + (1-\lambda)y - z) = f(\lambda(x-z) + (1-\lambda)(y-z)) \leqslant \lambda f(x-z) + (1-\lambda)f(y-z).$$

Como a ϕ é não-negativa, multiplicar por ela e integrar em z não altera o sinal da desigualdade:

$$\int f(\lambda x + (1-\lambda)y - z)\phi(z)dz \leqslant \int \lambda f(x-z) + (1-\lambda)f(y-z)\phi(z)dz.$$

Na notação: $f * \phi(\lambda x + (1-\lambda)y) \leqslant \lambda f * \phi(x) + (1-\lambda) f * \phi(y)$. Ou seja $f * \phi$ é convexa. □

Corolário 5.1. *Seja f uma função qualquer e ϕ tal que $f * \phi$ é convexa. Ou seja, ϕ convexifica f. Então para qualquer outra $\psi \geqslant 0$, $\phi * \psi$ também convexifica f.*

Demonstração. Pela associatividade da convolução $f * (\phi * \psi) = (f * \phi) * \psi$. Como $f * \phi$ é convexa e $\psi \geqslant 0$, o teorema 5.2 diz que $f * (\phi * \psi)$ também é convexa. □

Esse corolário é interessante pelo seguinte fato: digamos que temos uma função não-convexa $f(x)$ cujo argumento x sofre um erro aditivo Ξ. Se soubermos que $\mathbb{E}[f_\Xi]$ é convexa, então para qualquer erro adicional que somarmos a x, a função continuará convexa. Ou seja, qualquer erro $\Psi = \Xi + \Upsilon$ também convexificará f.

Por conta desse resultado, sabemos que o conjunto das variáveis aleatórias que convexificam uma dada função f nunca pode ser "pequeno". De fato, desde que exista um ruído Ξ que torne f convexa, todos os elementos do conjunto $\mathcal{P}_\Xi = \{\Xi + \Upsilon \mid \Upsilon \text{ é uma variável aleatória}\}$ também a convexificarão.

Teorema 5.3 (Derivada da convolução)**.** *Seja f uma função diferenciável. Então, para qualquer outra função g, $f * g$ também é diferenciável e sua derivada pode ser calculada como:*

$$\frac{\partial}{\partial x_i}(f * g) = \frac{\partial f}{\partial x_i} * g$$

Demonstração. A partir da definição de convolução e da regra de Leibniz para trocar a ordem entre integração e diferenciação segue que

$$\frac{\partial}{\partial x_i}(f * g) = \frac{\partial}{\partial x_i}\int f(x-\eta)g(\eta)\,d\eta = \int \left(\frac{\partial}{\partial x_i} f(x-\eta)\right) g(\eta)\,d\eta.$$ □

Corolário 5.2. *Como a convolução é simétrica, o mesmo resultado se aplica quando g é diferenciável e escrevemos*

$$\frac{\partial f}{\partial x_i} * g = \frac{\partial}{\partial x_i}(f * g) = f * \frac{\partial g}{\partial x_i}.$$

Corolário 5.3 (Convoluções suavizam)**.** *Se f é k vezes diferenciável e g é m vezes diferenciável, então $f * g$ será $k + m$ vezes diferenciável.*

Olhemos agora para a convolução como a média das translações de uma função f por um ruído Ξ. Se o ruído tiver uma densidade μ suave, então a função $\mathbb{E}[f_\Xi] = f * \tilde{\mu}$ não apenas manterá a regularidade da f receberá tantas derivadas adicionais quanto a μ possuir.

5.2 Teoria das Distribuições

Os resultados da seção 4.1 apenas fazem sentido para funções diferenciáveis e os resultados da seção 5.1 requerem que a variável aleatória Ξ tenha uma distribuição contínua. Essas restrições são muito fortes e dificilmente são atendidas por problemas reais, já que estes contam muitas vezes com variáveis aleatórias discretas ou mistas e com funções de custo imediato que são apenas contínuas. Contudo, todos os resultados lá apresentados podem ser generalizados no contexto da teoria das distribuições. Essa teoria foi desenvolvida inicialmente por Laurent Schwartz em [Sch66] com o intuito de estudar as soluções de equações diferenciais parciais no caso de funções não diferenciáveis. Para isso, é necessário generalizar a ideia de derivada estudando o caso em que a derivada de uma função é um objeto mais geral que uma função contínua.

Nesta seção enunciaremos alguns dos resultados da teoria que são mais importantes para o estudo da convexificação de funções de custo futuro. Os teoremas 5.6, 5.4, 5.5 e 5.7 falam sobre a regularidade da derivada generalizada de funções. Com eles, veremos que se f é uma função contínua, então a sua derivada nunca pode ser um objeto menos regular que uma medida. Além disso, a condição de que uma função duas vezes diferenciável seja convexa se e somente se f'' é sempre não-negativa pode ser generaliada nessa contexto. Aqui veremos que dada *qualquer* função f, ela será convexa se e somente se f'' for uma medida não-negativa.

Além do livro [Sch66], outras referências para a construção completa da teoria das distribuições, assim como as demonstrações dos teoremas desta seção, são os livros [Hö03] e [Fol99].

5.2.1 Distribuições e derivadas

A ideia central da teoria das distribuições é que podemos trocar a noção usual de derivada pela de integração por partes em conjunto com um certo tipo de função teste. Para os objetivos dessa seção, as funções teste serão sempre funções suaves e de suporte compacto, como descrito na definição 5.2.

Definição 5.2. *Chamamos de* função teste *toda função ϕ infinitamente diferenciável e de suporte compacto.*

Como elas têm suporte compacto, as funções teste sempre são iguais a zero para x distante o suficiente da origem. Portanto, se f for uma função diferenciável, a fórmula de integração por partes sempre pode ser escrita para qualquer função teste ϕ como

$$\int f'\phi \, dx = -\int f\phi' \, dx. \tag{5.2}$$

Repare que embora o lado esquerdo de (5.2) dependa da diferenciabilidade de f, o lado direito ainda está bem definido mesmo que *apenas* ϕ seja diferenciável. Um exemplo é quando $f(x) = |x|$, uma função poliedral e conhecidamente não diferenciável. Nesse caso, o lado direito da equação (5.2) fica

$$\begin{aligned} -\int_{-\infty}^{\infty} |x|\, \phi'(x)\, dx &= -\left[\int_{-\infty}^{0} (-x)\phi'(x)\, dx + \int_{0}^{\infty} x\phi'(x)\, dx\right] = -\int_{-\infty}^{0} \phi(x)\, dx + \int_{0}^{\infty} \phi(x)\, dx \\ &= \int_{-\infty}^{\infty} \operatorname{sgn}(x)\phi(x)\, dx. \end{aligned}$$

Onde $\operatorname{sgn}(x)$ é chamada de função sinal e é uma função não contínua definida por

$$\operatorname{sgn}(x) = \begin{cases} -1, & x < 0 \\ 1, & x \geqslant 0. \end{cases} \tag{5.3}$$

Nesse caso, a função sinal atua em relação à fórmula de integração por partes como se fosse a derivada da função valor absoluto. Isso motiva a definição 5.3, que estende a noção de derivada para uma quantidade muito maior de funções.

Definição 5.3 (Derivada fraca)**.** *Dada uma função f, se g satisfaz*

$$\int g\phi\,dx = -\int f\phi'\,dx$$

para toda função função teste ϕ, então g é dita derivada fraca *de f e denotada por f'.*

Um fator importante a se reparar é que, por conta da fórmula de integração por partes, a definição de derivada fraca 5.3 coincide com a usual no caso em que f é uma função diferenciável. Por conta disto, neste texto a noção de derivada fraca será referida como *derivada*. A derivada (fraca) também está bem definida para outras classes importantes de funções, como as poliedrais ou contínuas em geral. O teorema 5.4 é um resultado que garante que a derivada de uma função contínua sempre define uma função.

Teorema 5.4. *f é uma função contínua se e somente se f' existe e é uma função integrável em compactos.*

Contudo, podemos enfraquecer essas definições de forma que tenha-se uma noção de derivada que não apenas dê informação útil sobre a função f como também esteja definida para qualquer f. Para isso, é preciso reparar que, dada uma função f fixa, multiplicá-la por uma função teste e depois integrar é um mapa linear das funções testes em $\mathbb{R}$. Ou seja, se tivermos T_f definida por

$$T_f(\phi) = \int f\phi\,dx, \tag{5.4}$$

então T_f é linear, pois

$$T_f(\phi + \lambda\psi) = \int f(\phi + \lambda\psi)\,dx = \int f\phi\,dx + \lambda\int f\psi\,dx = T_f(\phi) + \lambda T_f(\psi).$$

Além dos mapas lineares com a forma da equação (5.4), podemos considerar *todos* os funcionais lineares que levam uma função teste em um número real. A isso, damos o nome de distribuição.

Definição 5.4 (Distribuições)**.** *Uma* distribuição *é um funcional linear contínuo no espaço das funções teste.*

Além de todas as funções integráveis em compactos, o espaço das distribuições contém diversos outros objetos, como todas as probabilidades e medidas, por exemplo.

Comentário 5.2. *Embora a função f e a distribuição T_f definida na equação* (5.4) *sejam objetos diferentes, para as aplicações aqui apresentadas não precisaremos nos preocupar com essa distinção. Portanto, sempre que dissermos que uma distribuição é uma função f, o verdadeiro significado é que existe uma distribuição T_f que representa f.*

A partir disso, definimos uma noção de derivada que vale para *qualquer* distribuição e que condiz com a definição 5.3 no caso de estarmos trabalhando com funções.

Definição 5.5 (Derivadas de distribuições)**.** *Dada uma distribuição T, chamamos de derivada de T e denotamos por T', a distribuição que satisfaz*

$$T'(\phi) = -T(\phi')$$

para toda função teste ϕ.

Repare que, para qualquer T, T' sempre existe e define um funcional linear, portanto também é uma distribuição. Por conta disso, esse processo pode ser iterado e toda distribuição tem infinitas derivadas. Nesse contexto, o teorema 5.5, analogamente ao teorema 5.4, garante que se f for uma função então sua derivada será necessariamente uma medida.

Teorema 5.5. *f é uma função integrável em compactos se e somente se f' é uma medida.*

Corolário 5.4. *f é uma função contínua se e somente se f'' é uma medida.*

Um exemplo de função cuja derivada é uma medida e não uma função é a função sinal apresentada como a derivada do módulo e definida na equação (5.3). Nesse caso, a derivada será dada por

$$\begin{aligned}
-\int \operatorname{sgn}(x)\phi'(x)\,dx &= -\int_{-\infty}^{0} -\phi'(x)\,dx - \int_{0}^{\infty} \phi'(x)\,dx \\
&= \int_{-\infty}^{0} \phi'(x)\,dx - \int_{0}^{\infty} \phi'(x)\,dx \\
&= [\phi(x)]_{-\infty}^{0} - [\phi(x)]0^{\infty} = \phi(0) + \phi(0) \\
&= 2\phi(0).
\end{aligned}$$

Nesse caso, a derivada da função sinal é dada pela medida $2 \cdot \delta_0$, igual a uma massa pontual de valor 2 na origem.

Definição 5.6. *Denota-se por δ_p a medida de probabilidade que é uma massa unitária no ponto p, definida por*

$$\delta_p(A) = \begin{cases} 1, & p \in A \\ 0, & p \in A. \end{cases}$$

Comentário 5.3. *Como distribuição, δ_p atua como*

$$\delta_p(\phi) = \phi(p).$$

Comentário 5.4. *A distribuição de probabilidade μ de toda variável aleatória discreta pode ser escrita como combinação convexa de funções δ_{x_i}. Ou seja, se $\sum p_i = 1$, então*

$$\mu = \sum p_i \delta_{x_i}.$$

Essas distribuições são especialmente importantes pois permitem generalizar diversos resultados válidos para variáveis aleatórias contínuas para o contexto de variáveis aleatórias discretas. Assim, todos os resultados da seção 5.1 também são válidos para variáveis aleatórias discretas, por exemplo. Logo, se Ξ tiver uma distribuição discreta, podemos escrever $\mathbb{E}[f_\Xi] = f * (\sum p_i \delta_{\xi_i})$. Mais geralmente, no contexto de distribuições de suporte compacto, todas as propriedades de convolução se generalizam e continuam válidos.

Um resultado importante no caso diferenciável é que uma função é convexa se e somente se a sua segunda derivada é uma função não-negativa. Analogamente, no contexto de distribuições, uma distribuição será uma função convexa se e somente se a sua segunda derivada como distribuição for uma medida positiva. Com isso, ganhamos uma nova ferramenta para analisar a convexidade de uma função que nos permitirá generalizar os resultados da seção 4.1 para o caso de funções apenas contínuas. Para isso, precisamos da definição 5.7 sobre o que significa uma distribuição ser não-negativa e do teorema 5.6 garante que distribuições não-negativas são sempre medidas.

Definição 5.7. *Uma distribuição T é dita não-negativa se, para qualquer função teste ϕ não-negativa,*

$$T(\phi) \geqslant 0.$$

Teorema 5.6. *T é uma distribuição não-negativa se e somente se T é uma medida não-negativa.*

A partir do teorema 5.6 e do resultado 5.4, conseguimos uma caracterização de funções convexas a partir de sua derivada como distribuição, descrita no teorema 5.7. Isso induz uma medida de não-convexidade análoga à da seção 4.1 e será uma parte importante nas demonstrações dos teoremas sobre redução de não-convexidade da seção 5.3.

Teorema 5.7. *f é uma função convexa se e somente se f'' é uma distribuição não-negativa.*

5.2.2 Fórmula dos saltos

Usando o ferramental de distribuições, pode-se deduzir uma fórmula para calcular a derivada de funções que são diferenciáveis a menos de pontos isolados. Esse resultado é conhecido como fórmula dos saltos e diz que podemos calcular a derivada normalmente nos pontos de diferenciabilidade mas que podem aparecer massas pontuais (os "saltos") nos pontos de não continuidade da função. Com o intuito de facilitar a notação, definiremos a função de Heaviside H e o teorema 5.8 se escreve em termos de suas derivadas.

Definição 5.8 (Função de Heaviside)**.** *Chamamos de* função de Heaviside*, a função definida por*

$$H(x) = \begin{cases} 1, & x \geqslant 0 \\ 0, & x < 0. \end{cases}$$

Comentário 5.5. *A derivada de H é a função delta no ponto zero. Ou seja, $H' = \delta_0$.*

Teorema 5.8 (Fórmula dos saltos)**.** *Seja f uma função diferenciável por partes. Ou seja, tal que existem f_1, f_2 diferenciáveis e um $a \in \mathbb{R}$ pros quais:*

$$f(x) = f_1(x) \cdot H(x-a) + f_2(x) \cdot H(a-x) = \begin{cases} f_1(x), & x \leqslant a \\ f_2(x), & x > a \end{cases}.$$

Então a derivada de f será

$$f' = f_1' \cdot H(x-a) + f_2' \cdot H(a-x) + \left[f_2(a^+) - f_1(a^-)\right] \delta_a.$$

Corolário 5.5. *Seja $f = \sum_i f_i \cdot \mathbb{1}_{(a_{i-1}, a_i)}$, onde cada f_i é diferenciável no intervalo (a_{i-1}, a_i). Então a sua derivada é dada pela fórmula*

$$f' = \sum_i f_i' \cdot \mathbb{1}_{(a_{i-1}, a_i)} + \sum_i \left[f_{i+1}(a_i^+) - f_i(a_i^-)\right] \delta_{a_i}.$$

Corolário 5.6 (Derivada do mínimo de funções convexas)**.** *Seja $f = \min_i f_i$, onde cada f_i é uma função convexa duas vezes diferenciável. Então f é uma função contínua e definida por partes como*

$$f = \sum_i f_i \cdot \mathbb{1}_{(a_{i-1}, a_i)}$$

cujas primeira e segunda derivada são

$$f' = \sum_i f_i' \cdot \mathbb{1}_{(a_{i-1},a_i)}, \tag{5.5}$$

$$f'' = \sum_i f_i'' \cdot \mathbb{1}_{(a_{i-1},a_i)} - \sum_i \left[f_i'(a_i^-) - f_{i+1}'(a_i^+)\right] \delta_{a_i}. \tag{5.6}$$

onde cada $f_i'' \cdot \mathbb{1}_{(a_{i-1},a_i)}$ define, pelo teorema 5.7, uma medida positiva com suporte em (a_{i-1}, a_i) e todos os $-\left[f_i'(a_i^-) - f_{i+1}'(a_i^+)\right] \delta_{a_i}$ são medidas pontuais negativas.

A partir do corolário 5.6 temos uma fórmula para calcular a derivada de funções de valor ótimo de problemas inteiros mistos. Repare que embora a parte que depende das f_i'' seja sempre não-negativa, f'' também tem massas pontuais negativas nos pontos em que o mínimo "troca" de uma f_i para outra. Essa parte negativa é o que causa a não-convexidade vista nos exemplos da seção 2.

No caso de funções poliedrais, a fórmula (5.6) toma uma forma particularmente simples. Como cada f_i é uma função linear por partes, suas segundas derivadas serão massas pontuais *positivas* nos pontos de não diferenciabilidade da f_i. Portanto f'' será uma diferença entre as δ_{x_i} positivas vindas da convexidade das f_i e as δ_{a_i} negativas da fórmula.

5.3 Redução da parte negativa de f''

As funções de valor ótimo que desejamos convexificar normalmente são contínuas mas não diferenciáveis. Isso impede que resultados como o teorema 4.2 sejam utilizados para essas funções. Contudo, a partir do ferramental desenvolvido, é possível generalizar essa resultado para funções que são apenas contínuas, obtendo que convoluções com medidas de probabilidade sempre diminuem a integral da segunda derivada de uma função contínua. Esse resultado está enunciado no teorema 5.10, cujo demonstração é análoga à do teorema 4.2, mudando apenas que as derivadas são tomadas no sentido das distribuições.

Tomando f como uma função contínua, os teoremas 5.5 e 5.4 garantem que a sua segunda derivada, f'', será uma medida. Assim como separávamos a segunda derivada de uma função diferenciável em partes negativa e positiva, existe um resultado, chamado Teorema de Hahn-Jordan, que diz que qualquer medida pode ser decomposta na diferença entre duas medidas positivas cujos suportes não se interceptam. Aqui apenas enunciaremos o teorema, cuja demonstração e outras aplicações podem ser encontradas na literatura de teoria da medida como em [Fol99, Fis12].

Teorema 5.9 (Decomposição de Hahn-Jordan)**.** *Seja μ uma medida. Existe uma única decomposição*

$$\mu = \mu_+ - \mu_- \tag{5.7}$$

onde μ_+, μ_- são medidas positivas cujos suportes não se interceptam.

Corolário 5.7 (Minimalidade da decomposição de Hahn-Jordan)**.** *Qualquer outra decomposição de uma medida μ como diferença de medidas positivas terá componentes maiores que a de Hahn-Jordan. Ou seja, se podemos escrever $\mu = \lambda - \nu$ para λ, ν positivas, então*

$$\lambda \geqslant \mu_+, \quad \nu \geqslant \mu_-.$$

O corolário 5.7 diz que a parte negativa de uma medida, definida na equação (5.7), é menor que a parte negativa de qualquer outra decomposição desta medida. Para o teorema a seguir, aplicaremos essa decomposição à segunda derivada da $(f * \mu)''$ e a majoraremos pela parte negativa de f'', de maneira análoga ao que foi feito na demonstração do teorema 4.2.

Teorema 5.10. *Seja f uma função contínua. Dada qualquer medida de probabilidade μ, vale que*

$$\int (f'' * \mu)_- \leqslant \int (f'')_-.$$

Demonstração. Como f'' é uma medida, ela pode ser decomposta como a diferença entre uma parte positiva e outra negativa. Assim, existe uma decomposição minimal garantida pelo teorema 5.9 e cujas componentes estão abaixo de qualquer outra decomposição de f''. Denotamos essa decomposição e a de $(f * \mu)''$ (que também é uma medida) por

$$f'' = \lambda_+ - \lambda_-,$$
$$(f * \mu)'' = \nu_+ - \nu_-.$$

Onde $\lambda_+, \lambda_-, \nu_+, \nu_-$ são medidas não-negativas. Usando a fórmula para derivadas de convoluções, a medida $(f * \mu)''$ também aceita outra decomposição alternativa dada por

$$(f * \mu)'' = f'' * \mu = (\lambda_+ - \lambda_-) * \mu = \lambda_+ * \mu - \lambda_- * \mu.$$

Da propriedade de minimalidade da decomposição de Hahn-Jordan segue que ν_+ e ν_- são menores que as componentes de qualquer outra decomposição de $(f * \mu)''$. Ou seja,

$$\lambda_- * \mu \geqslant \nu_-.$$

Por fim, como μ é uma medida de probabilidade, uma aplicação do teorema de Fubini e a monotonicidade da integral nos dizem que

$$\int \lambda_- = \int \lambda_- * \mu \geqslant \int v_-. \qquad \square$$

Comentário 5.6. *Repare que embora o teorema 4.2 sobre normas da segunda derivada possa ser generalizado para o caso não diferenciável ao usar a norma 1, não é possível encontrar nenhum análogo ao teorema para a convexificação na norma do supremo. Isso se dá pelo fato de que podemos apenas garantir que f'' será uma medida para a qual embora a noção de integral seja sempre bem definida, não há análogo claro da definição de supremo no caso em que f'' não é uma função.*

Em aplicações computacionais, nem sempre é possível trabalhar diretamente com a variável aleatória Ξ. Ao invés disso, é necessário considerar uma variável aleatória discreta $\bar{\Xi}$ que aproxime Ξ em um número finito de pontos. Ou seja, $\bar{\Xi}$ terá uma distribuição de probabilidade da forma $u = \sum_i p_i \delta_{\xi_i}$. A partir daí, surge a pergunta se as propriedades de convexificação de Ξ também serão válidas, mesmo que parcialmente, para $\bar{\Xi}$. Os teorema 3.2 e 5.10 garantem que a convolução de uma função f com a distribuição de $\bar{\Xi}$ sempre diminui a não-convexidade de f. Contudo, o resultado do teorema 5.11 diz que se f for mínimo de funções diferenciáveis, a função $f * u$ nunca pode se tornar totalmente convexa se u for uma probabilidade discreta.

Teorema 5.11 (Funções suaves não são convexificadas por distribuições discretas)**.** *Seja $f = \min_i f_i$, onde cada f_i é uma função convexa cuja segunda derivada é uma função. Então não existe nenhuma probabilidade discreta $u = \sum c_j \delta_{\xi_j}$ para a qual $(f \star u)'' \geqslant 0$. Em outras palavras, f não pode ser convexificada por distribuições discretas.*

Demonstração. Como f é mínimo de funções convexas, ela é contínua e existem intervalos disjuntos (a_i, a_{i+1}) nos quais ela é igual a alguma função convexa f_i. Ou seja,

$$f = \sum f_i \cdot \mathbb{1}_{[a_{i-1}, a_i]}.$$

Usando a fórmula dos saltos, a continuidade da f dá as seguintes fórmulas para suas derivadas:

$$\begin{aligned} f' &= \sum f_i' \cdot \mathbb{1}_{(a_{i-1},a_i)}, \\ f'' &= \sum f_i'' \cdot \mathbb{1}_{(a_{i-1},a_i)} + \sum \underbrace{-\left[f_i'(a_i^-) - f_{i+1}'(a_i^+)\right]}_{\leqslant 0} \cdot \delta_{a_i}. \end{aligned}$$

Como f é mínimo de funções convexas, cada um dos saltos na derivada é negativo. Convoluindo com u obtemos por linearidade que:

$$\begin{aligned} (f \star u)'' = f'' \star u &= \sum f_i'' \cdot \mathbb{1}_{(a_{i-1},a_i)} \star u + \left(\sum -\left[f_i'(a_i^-) - f_{i+1}'(a_i^+)\right] \cdot \delta_{a_i}\right) \star u \\ &= \sum f_i'' \cdot \mathbb{1}_{(a_{i-1},a_i)} \star \left(\sum c_j \delta_{\xi_j}\right) + \left(\sum -\left[f_i'(a_i^-) - f_{i+1}'(a_i^+)\right] \cdot \delta_{a_i}\right) \star \left(\sum c_j \delta_{\xi_j}\right) \\ &= \underbrace{\sum f_i''(\cdot - \xi_j) \cdot \mathbb{1}_{(a_{i-1}+x_j, a_i+\xi_j)}}_{\text{parte integravel e positiva}} - \underbrace{\sum c_j \left[f_i'(a_i^-) - f_{i+1}'(a_i^+)\right] \cdot \delta_{a_i+\xi_j}}_{\text{parte discreta e negativa}} \end{aligned}$$

Pela fórmula acima, $(f \star u)''$ é uma medida. Portanto, podemos aplicá-la ao conjunto $\{a_1 + \xi_1\}$ e apenas a parte negativa será não-nula:

$$(f \star u)''(\{a_1 + \xi_1\}) = -c_1 \left[f_1'(a_1^-) - f_2'(a_1^+)\right] < 0.$$

Logo $f \star u$ não é convexa. □

Uma consideração a se fazer é que as funções de valor ótimo de problemas inteiros mistos são sempre da forma $f = \min f_i$, onde cada f_i é solução de um problema de otimização com apenas variáveis contínuas. Portanto, se as f_i forem diferenciáveis, não há nenhuma aproximação discreta de Ξ que torne f convexa. Repare que no caso de programas lineares, as f_i serão funções poliedrais e f ainda pode ser convexificada por $\bar{\Xi}$, como pode ser visto no exemplo da figura 2.12. Além disso, como vimos no exemplo da figura 2.17 é razoável esperar que a presença de um número significativo de cenários seja capaz de reduzir o *gap* de dualidade mesmo que a convexificação complete da função não seja possível.

6 Algoritmos de planos cortantes para funções não-convexas

Vimos, nas seções 3, 4 e 5, diversos resultados relacionando a relaxação Lagrangiana da média $\bar{\mathcal{Q}}(x)$ de funções $Q(x, \xi_i)$, correspondentes a diversos cenários de um problema de otimização estocástico, e a média das respectivas relaxações Lagrangianas. Em especial, destacamos o teorema 3.1, que mostra que a relaxação Lagrangiana de $\bar{\mathcal{Q}}$ está mais próxima de $\bar{\mathcal{Q}}$ do que a média das relaxações Lagrangianas de cada uma das $Q(x, \xi_i)$. Para o caso de ruídos aditivos, em que $Q(x, \xi_i) = f(x + \xi_i)$ para uma função f dada, vimos que o *gap* entre a função média $\bar{\mathcal{Q}}$ e sua relaxação Lagrangiana $\operatorname{conv} \bar{\mathcal{Q}}$ pode ser bastante menor do que o *gap* entre f e $\operatorname{conv} f$, como no caso do teorema 3.4, onde a média é tomada sobre conjuntos de incerteza suficientemente grandes.

De forma geral, observamos que sempre temos a seguinte desigualdade, que explicita a qualidade da aproximação de $\bar{\mathcal{Q}} = \mathbb{E}[Q]$ em função da *ordem* com a qual se calcula a média e a relaxação Lagrangiana:

$$\mathbb{E}\left[\operatorname{conv}(Q(x, \Xi))\right] \leqslant \operatorname{conv}(\mathbb{E}\left[Q(x, \Xi)\right]) \leqslant \mathbb{E}\left[Q(x, \Xi)\right] = \bar{\mathcal{Q}}(x), \tag{6.1}$$

onde denotamos por $\operatorname{conv}(\phi)$ a relaxação Lagrangiana de uma função ϕ, visto que a relaxação pode ser obtida ao tomar o fecho convexo do epígrafo de ϕ. Observe que a relaxação Lagrangiana

$\text{conv}(Q(x,\Xi))$ deve ser efetuada apenas em relação à variável x. A "mesma" desigualdade vale se, em vez da média, tomamos uma medida de risco, como demonstrado no teorema 3.5.

Estes resultados corroboram os diversos exemplos da seção 2, onde a média $\bar{\mathcal{Q}}(x)$ podia ser convexa mesmo quando as funções $Q(x,\xi)$ eram não-convexas. Adicionalmente, nos casos onde $\bar{\mathcal{Q}}(x)$ resultava não-convexa, que esta possuía aproximações convexas muito melhores do que a média das aproximações convexas das correspondentes $Q(x,\xi)$.

Nesta seção, motivados pelos resultados destacados acima sobre a convexificação da função de custo futuro pela incerteza Ξ, e pela desigualdade fundamental (6.1) apresentamos algumas modificações para o procedimento de cálculo de cortes em programação dinâmica dual estocástica. Estudaremos 3 tipos de cortes, a saber os cortes de Benders, cortes Lagrangianos e cortes de Benders reforçados, e observaremos como o procedimento de cálculo destes cortes deve ser alterado segundo calculemos o termo intermediário $\text{conv}(\mathbb{E}[Q(x,\Xi)])$, correspondentes à *formulação conjunta*, ou o termo inferior $\mathbb{E}[\text{conv}(Q(x,\Xi)]$, correspondente à *formulação decomposta*.

6.1 A relação entre cortes e relaxação Lagrangiana

O algoritmo de programação dual dinâmica estocástica pode ser resumido, de forma bastante genérica, na iteração alternada entre duas etapas, conhecidas como *forward* e *backward*:

Forward Seleciona-se, para cada estágio de 1 a T, uma realização da incerteza, gerando assim um *cenário*: um caminho do primeiro ao último estágio na árvore de incertezas;

Backward Em seguida, para cada estágio t desde $T-1$ até 1, obtêm-se um corte para a função $\mathfrak{Q}_t$ a partir da estimativa da função de custo futuro avaliada no estado final do estágio t.

Relembremos a equação (2.2) que dá a função de custo futuro $\bar{\mathcal{Q}}_t(x_{t-1})$ em função das várias funções valor ótimo $Q_t(x_{t-1},\xi_t)$:

$$\bar{\mathcal{Q}}_t(x_{t-1}) = \mathbb{E}\left[Q_t(x_{t-1},\Xi_t)\right].$$

Observe que, para uma dada realização ξ_t da incerteza, $Q_t(x_{t-1},\xi_t)$ é uma função de x_{t-1} apenas, que poderia ser indicada por $Q_{t,\xi_t}(x_{t-1})$. Para simplificar a notação, iremos denotar x_{t-1} por x_{ini}, e consideramos um conjunto de índices I para os cenários $\{\xi_t^i\}_{i\in I}$, de forma que podemos escrever

$$\bar{\mathcal{Q}}(x_{\text{ini}}) = \mathbb{E}\left[Q_i(x_{\text{ini}})\right],$$

também omitindo os índices designando o estágio.

Comentário 6.1. *Esta notação para Q_i pode induzir uma pequena confusão, pois habitualmente o índice em Q indica o estágio t, e não o cenário ξ^i escolhido. Entretanto, o foco desta seção, como também nas seções anteriores, é a transformação das diversas Q_i em $\bar{\mathcal{Q}}$ através da média (ou medida de risco), e não a dinâmica através de todos os estágios. Além disso, iremos comparar estas funções e suas respectivas relaxações convexas,* $\text{conv}\, Q_i$ *e* $\text{conv}\, \bar{\mathcal{Q}}$*, todas elas funções apenas da variável de estado inicial x_{ini}. Se mantivéssemos a notação $Q(x_{ini},\xi)$, teríamos que sempre enfatizar que se trataria de uma "relaxação convexa parcial", apenas na primeira variável, e a notação* $\text{conv}\, Q(x_{ini},\xi)$ *poderia ser ambígua.*

Tradicionalmente, os algoritmos de planos cortantes utilizados para resolver problemas convexos de programação estocástica multi-estágio se baseiam em aproximar a função de custo futuro $\bar{\mathcal{Q}}$ por meio do cálculo de cortes para as funções Q_i correspondentes a cada cenário ξ_i. Ou seja, decompõe-se o problema "do estágio seguinte" em cada um dos cenários ξ_i, calcula-se um corte para cada Q_i,

e depois encontra-se um corte médio, válido para $\bar{\mathcal{Q}}$. Mais ainda, devido à convexidade das funções em questão, é possível mostrar que este corte é *exato*: o valor em x_{ini} do corte coincide com a média do valor calculado em cada cenário.

Ora, no caso não-convexo, esta decomposição não resulta, necessariamente, em cortes exatos. De fato, um corte para a função Q_i está certamente abaixo de $\operatorname{conv} Q_i$, e portanto sua média estará abaixo de $\mathbb{E}\left[\operatorname{conv}(Q_i)\right]$, que pode ser diferente de $\bar{\mathcal{Q}} = \mathbb{E}\left[Q_i\right]$. Assim, a equação (3.3) indica que calcular cortes diretamente para a função de custo futuro, sem decompô-la em cenários, fornece melhores aproximações, pois estas estão limitadas apenas por $\operatorname{conv}\mathbb{E}\left[Q_i\right]$, que pode ser maior do que $\mathbb{E}\left[\operatorname{conv} Q_i\right]$. O processo de calcular cortes para a formulação conjunta requer um maior esforço computacional mas garante uma maior redução no *gap*.

Se aproximamos cada uma das $Q_i(x_{\text{ini}})$ por planos cortantes, o melhor que podemos obter é a sua regularização convexa $\check{Q}_i(x_{\text{ini}})$, (cf. Definição 3.1). Daí, ao tomarmos a média destas aproximações, esta dará, novamente na melhor das hipóteses, um *gap* médio:

$$\bar{\mathcal{Q}}(x_{\text{ini}}) - \mathbb{E}\left[\operatorname{conv} Q_i(x_{\text{ini}})\right] = \mathbb{E}\left[Q_i(x_{\text{ini}}) - \operatorname{conv} Q_i(x_{\text{ini}})\right]. \tag{6.2}$$

Entretanto, podemos tentar aproximar diretamente a função $\bar{\mathcal{Q}}(x_{\text{ini}})$ por planos cortantes, isto é, sem decompor o "estágio seguinte" em um problema para cada cenário ξ_t^i e resolvê-los separadamente, mas sim tentar resolver uma formulação com todas as realizações simultaneamente. Isso nos permitirá calcular cortes para $\bar{\mathcal{Q}}$. No caso em que esta última é convexa, apesar de as Q_i não o serem, estes cortes seriam capazes de representar $\bar{\mathcal{Q}}$ com *gap* zero. No caso geral, o erro de aproximação será

$$\bar{\mathcal{Q}}(x_{\text{ini}}) - \operatorname{conv} \bar{\mathcal{Q}}(x_{\text{ini}}),$$

que, como vimos nas seções 2 e 3, pode ser bem menor do que o *gap* médio dado pela equação (6.2) acima, de acordo com diferentes normas usadas para quantificá-lo, como explorado na seção 3.2.1.

6.2 Formulação dos problemas para geração de cortes

Uma forma típica de se obter um corte para uma função de custo futuro é através da solução dos problemas do estágio seguinte, cada um deles fornecendo um corte por dualidade, e em seguida calculando o corte médio. Vejamos esta parte em mais detalhes, a começar pela equação (2.1) que descreve cada uma das Q_i. Em nossa notação, temos:

$$\begin{aligned} Q_i(x_{\text{ini}}) = \min_{x_i, y_i} \quad & c^\top x_i + d^\top y_i \\ \text{s.a} \quad & Ax_i + By_i = b + Tx_{\text{ini}} + w_i, \\ & (x, y) \in D \end{aligned} \tag{6.3}$$

onde retiramos a dependência em t e efetuamos o produto $W\xi_t^i = w_i$ para a incerteza.

Também poderíamos considerar todos os problemas Q_i simultaneamente. Neste caso, obtemos diretamente uma expressão para $\bar{\mathcal{Q}}$:

$$\begin{aligned} \bar{\mathcal{Q}}(x_{\text{ini}}) = \min_{x_i, y_i} \quad & \textstyle\sum_i p_i(c^\top x_i + d^\top y_i) \\ \text{s.a} \quad & Ax_i + By_i = b + Tx_{\text{ini}} + w_i, \qquad \forall i, \\ & (x_i, y_i) \in D, \qquad \forall i \end{aligned} \tag{6.4}$$

Esta formulação contém todos os cenários ao mesmo tempo, e por isso contém todas as variáveis de decisão (x_i, y_i) correspondentes simultaneamente.

6.2.1 O caso da programação linear

Suponhamos que não temos restrições de integralidade, de forma que, para cada i, o problema de otimização (6.3) é um PL apenas com variáveis contínuas. Suponhamos agora que resolvemos cada um deles para um valor de x^*_{ini} dado. Por dualidade forte, válida no caso em que o valor ótimo é finito, temos que

$$Q_i(x_{\text{ini}}) - Q_i(x^*_{\text{ini}}) \geqslant \pi_i^\top T(x_{\text{ini}} - x^*_{\text{ini}})$$

onde π_i é o multiplicador de Lagrange correspondente à restrição

$$Ax + By = b + Tx_{\text{ini}} + w_i.$$

Ao efetuar a média de todas estas desigualdades, multiplicando pelas probabilidades p_i de cada cenário, obtemos:

$$\bar{\mathcal{Q}}(x_{\text{ini}}) - \bar{\mathcal{Q}}(x^*_{\text{ini}}) \geqslant \bar{\pi}^\top T(x_{\text{ini}} - x^*_{\text{ini}}),$$

onde $\bar{\pi}$ é a média $\sum_i p_i\pi_i$ dos multiplicadores de Lagrange.

Ao considerarmos a formulação conjunta para $\bar{\mathcal{Q}}$ como visto em (6.4), obtemos diretamente uma desigualdade para $\bar{\mathcal{Q}}$. Note que a variável x_{ini} aparece em cada uma das restrições para os cenários, dando origem a um multiplicador de Lagrange ν_i para cada uma delas. Assim, obtemos:

$$\bar{\mathcal{Q}}(x_{\text{ini}}) - \bar{\mathcal{Q}}(x^*_{\text{ini}}) \geqslant [\nu_1^\top, \nu_2^\top, \ldots] \begin{bmatrix} T \\ T \\ \vdots \end{bmatrix} (x_{\text{ini}} - x^*_{\text{ini}}) = \left(\sum_i \nu_i\right)^\top T(x_{\text{ini}} - x^*_{\text{ini}}) = \nu^\top T(x_{\text{ini}} - x^*_{\text{ini}}).$$

Dado $\bar{\pi} = \sum_i p_i\pi_i$, é sempre possível escolher multiplicadores ν_i no problema (6.4) de forma que $\nu_i = p_i\pi_i$, e portanto $\bar{\pi} = \nu$. Mas isto ocorre apenas no caso de PL. A seguir, veremos como as formulações decompostas e conjuntas diferem no caso de problemas inteiros-mistos, onde analisamos três métodos para obter cortes válidos em problemas não-convexos.

6.2.2 Geração de cortes de Benders via relaxação linear

Uma primeira técnica para obter cortes válidos para problemas inteiros-mistos é através da relaxação PL do problema (6.3). Nestes casos, o domínio D é dado pela interseção de um poliedro P e de algumas restrições inteiras, que designamos por X. A relaxação PL é obtida ignorando as restrições em X, e portanto aumentando o conjunto viável. O problema resultante é designado por $Q_{i,LP}(x_{\text{ini}})$, e sempre temos a desigualdade $Q_i(x_{\text{ini}}) \geqslant Q_{i,LP}(x_{\text{ini}})$.

Podemos usar dualidade para o problema relaxado, como descrito anteriormente. Como esta relaxação dá uma função de valor ótimo $Q_{i,LP}(x_{\text{ini}})$ convexa, teremos então as desigualdades

$$Q_i(x_{\text{ini}}) \geqslant Q_{i,LP}(x_{\text{ini}}) \geqslant Q_{i,LP}(x^*_{\text{ini}}) + \pi_{i,LP}^\top T(x_{\text{ini}} - x^*_{\text{ini}}),$$

que fornecem um corte válido para Q_i, mas que pode ter um grande *gap* em x_{ini}, que depende de quão boa é a aproximação $Q_{i,LP}(x_{\text{ini}})$ para $Q_i(x_{\text{ini}})$. Novamente, tomamos o corte médio para obter um corte para $\bar{\mathcal{Q}}$, e o *gap* resultante será

$$\sum_i p_i\big(Q_i(x_{\text{ini}}) - Q_{i,LP}(x_{\text{ini}})\big).$$

Também poderíamos usar a formulação conjunta e obter apenas um corte, diretamente, para $\bar{\mathcal{Q}}$, também relaxando as restrições de integralidade. Como a relaxação irá afetar todas as decisões (x_i, y_i), o problema conjunto relaxado será equivalente à média dos problemas de cada cenário, e ambos serão PLs. Assim, por um argumento similar ao caso convexo, o corte obtido será o mesmo.

6.2.3 Geração de cortes via Lagrangianas

Os cortes de Benders são de obtenção relativamente simples, mas o *gap* $Q_i(x_{\text{ini}}) - Q_{i,LP}(x_{\text{ini}})$ pode ser bastante significativo. Ora, $Q_{i,LP}$ é uma função convexa, que é menor do que ou igual a Q_i, logo ela também é menor do que ou igual a $\check{Q}_i$. Como esta última tem, por definição, o menor *gap* possível para uma aproximação convexa, poderíamos tentar aproximar Q_i por $\check{Q}_i$. Isto é possível usando cortes Lagrangianos, que já foram tratados nos relatórios anteriores devido ao seu uso no SDDiP. Recapitulamos aqui sua formulação.

Definição 6.1 (Lagrangianas e convexificação)**.** *A* Lagrangiana *da função $Q_i(x_{ini})$, descrita pelo problema de otimização* (6.3)*, é a função*

$$L_i(x_{ini}, u, x_0, x, y) = c^\top x + d^\top y + u^\top (x_0 - x_{ini}),$$

e a Função dual de Lagrange *será*

$$\begin{aligned} g_i(x_{ini}, u) = \min_{x_0,x,y} \quad & L_i(x_{ini}, u, x_0, x, y) \\ s.a \quad & Ax + By = b + Tx_0 + w_i, \\ & (x, y) \in D \end{aligned} \tag{6.5}$$

Esta construção corresponde à relaxação Lagrangiana da restrição $x_0 = x_{\text{ini}}$ do seguinte problema de otimização, equivalente a (6.3):

$$\begin{aligned} Q_i(x_{\text{ini}}) = \min_{x_0,x,y} \quad & c^\top x + d^\top y \\ \text{s.a} \quad & Ax + By = b + Tx_0 + w_i, \\ & x_0 = x_{\text{ini}}, \\ & (x, y) \in D \end{aligned} \tag{6.6}$$

Observe-se que, ao contrário da relaxação PL, não ignoramos as restrições de integralidade presentes em D.

Proposição 6.1. *A* relaxação Lagrangiana *de Q_i pode ser calculada pelo problema de otimização dual*

$$\check{Q}_i(x_{ini}) = \max_u \; g_i(x_{ini}, u). \tag{6.7}$$

A parte importante da construção da relaxação Lagrangiana é que, para x_{ini} fixo, as funções $g_i(x_{\text{ini}}, \cdot)$ são côncavas em u, como mínimo de funções lineares em u. Portanto, o problema de otimização em (6.7) é um problema bem-posto de otimização convexa.

Mais ainda, como podemos separar o termo $-u^\top x_{\text{ini}}$ de L_i, que é independente da minimização em (6.5), as funções $g_i(x_{\text{ini}}, u)$ são lineares em x_{ini}, e portanto $\check{Q}_i(x_{\text{ini}})$ é uma função convexa de x_{ini}, como máximo de uma família de funções convexas.

Assim, temos um problema de otimização que dá $\check{Q}_i$, a relaxação Lagrangiana de Q_i, e portanto sua melhor aproximação convexa. Esta melhoria em comparação com a relaxação PL, $Q_{i,LP}$, vem de considerarmos, ainda que de forma indireta, devido à sobreposição de dois problemas de otimização, as restrições de integralidade do domínio. Um exemplo de função Q_i, sua relaxação PL e sua relaxação convexa pode ser visto na figura 6.1.

A partir da solução do problema de otimização em (6.7), para x^*_{ini} dado, é possível determinar um corte para $\check{Q}_i$, e consequentemente um corte válido para Q_i:

$$Q_i(x_{\text{ini}}) \geqslant \check{Q}_i(x_{\text{ini}}) \geqslant \check{Q}_i(x^*_{\text{ini}}) + \pi_i^\top T(x_{\text{ini}} - x^*_{\text{ini}}).$$

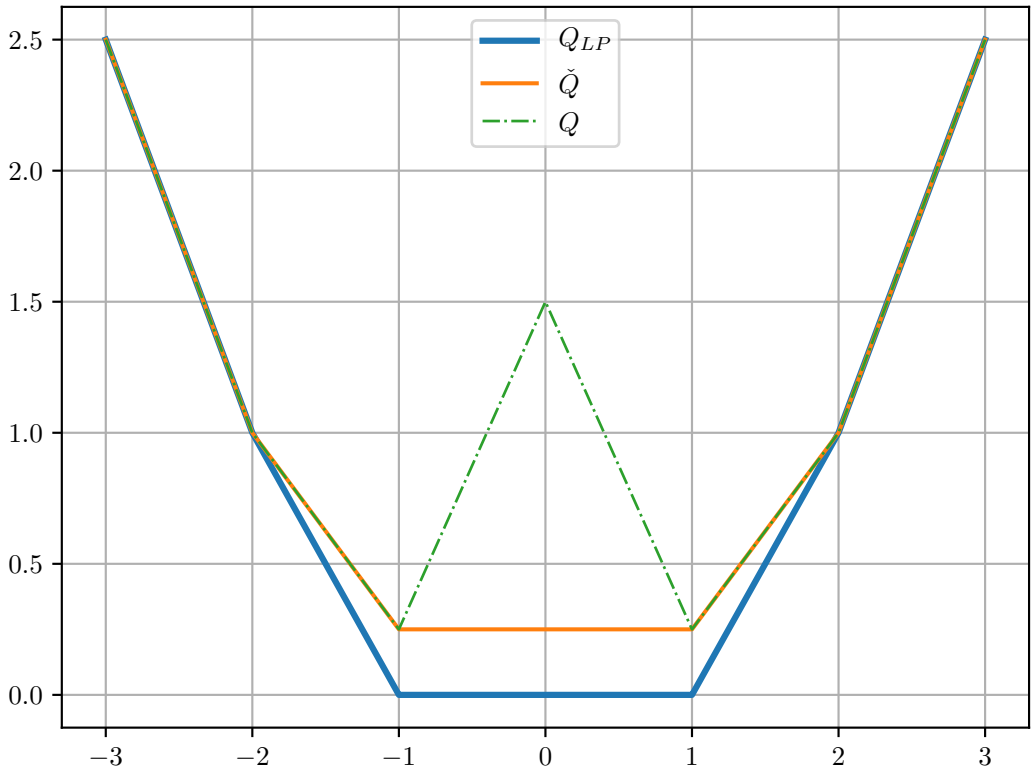


Figura 6.1: Uma função valor ótimo Q, sua relaxação convexa $\check{Q}$ e sua relaxação PL Q_{PL}.

Novamente, ao tomarmos a média destes cortes, obtemos um corte válido para $\bar{\mathcal{Q}}$. Entretanto, ainda temos um *gap*, que será o *gap* médio entre Q_i e $\check{Q}_i$.

Vejamos agora o que aconteceria numa formulação conjunta. A construção do problema dual segue os mesmos passos do caso onde os cenários estão separados, mas inclui apenas uma vez a restrição $x_{\text{ini}} = x_0$ na reformulação do problema visto em (6.4):

$$\begin{aligned} \bar{\mathcal{Q}}(x_{\text{ini}}) = \min_{x_0, x_i, y_i} \quad & \textstyle\sum_i p_i (c^\top x_i + d^\top y_i) & \\ \text{s.a} \quad & Ax_i + By_i = b + Tx_0 + w_i, & \forall i \\ & (x_i, y_i) \in D, & \forall i \\ & x_0 = x_{\text{ini}} & \end{aligned} \tag{6.8}$$

Agora, a Lagrangiana é dada por

$$\bar{L}(x_{\text{ini}}, u, x_0, x_i, y_i) = u^\top (x_0 - x_{\text{ini}}) + \sum_i p_i (c^\top x_i + d^\top y_i)$$

e a função dual, por

$$\begin{aligned} \bar{g}(x_{\text{ini}}, u) = \min_{x_0, x_i, y_i} \quad & \bar{L}(x_{\text{ini}}, u, x_0, x_i, y_i) & \\ \text{s.a} \quad & Ax_i + By_i = b + Tx_0 + w_i, & \forall i \\ & (x_i, y_i) \in D & \forall i \end{aligned} \tag{6.9}$$

Observe que, ao contrário de (6.4), que é um problema separável nas variáveis (x_i, y_i), a presença da variável de decisão x_0 em todas as restrições de (6.8) torna este último não-separável. É claro que podemos eliminar esta variável, que está fixa em x_{ini}, e obter $\bar{\mathcal{Q}}(x_{\text{ini}}) = \sum p_i Q_i(x_{\text{ini}})$, separando o problema. Entretanto, ao relaxarmos esta restrição, para obter a função dual, perdemos esta possibilidade, e apenas podemos garantir que

$$\bar{g}(x_{\text{ini}}, u) \geqslant \sum p_i g_i(x_{\text{ini}}, u).$$

Portanto, para calcular a função dual $\bar{g}$ exatamente, teremos que resolver um problema com N vezes mais variáveis e restrições, o que implica um custo computacional maior. Entretanto, como a relaxação Lagrangiana permite calcular a relaxação convexa, o corte que obtivermos ao resolver

$$\check{\bar{\mathcal{Q}}}(x_{\text{ini}}) = \max_u \ \bar{g}(x_{\text{ini}}, u). \tag{6.10}$$

descreverá, de fato, a envoltória convexa de $\bar{\mathcal{Q}}$.

O ganho ao se usar a formulação conjunta para a Lagrangiana será a diferença entre realizar a média das convexificações $\check{Q}_i$ e a convexificação da média, $\check{\bar{\mathcal{Q}}}$. Como vimos nos capítulos anteriores, esta diferença pode ser significativa. Por exemplo, no caso em que as funções Q_i eram não-convexas, como quando elas eram translações da função W, mas sua média $\bar{\mathcal{Q}}$ era convexa, devido à suavização, o *gap* entre $\bar{\mathcal{Q}}$ e $\check{\bar{\mathcal{Q}}}$ é zero, enquanto que o *gap* médio não será. Podemos observar esta diferença na figura a seguir:

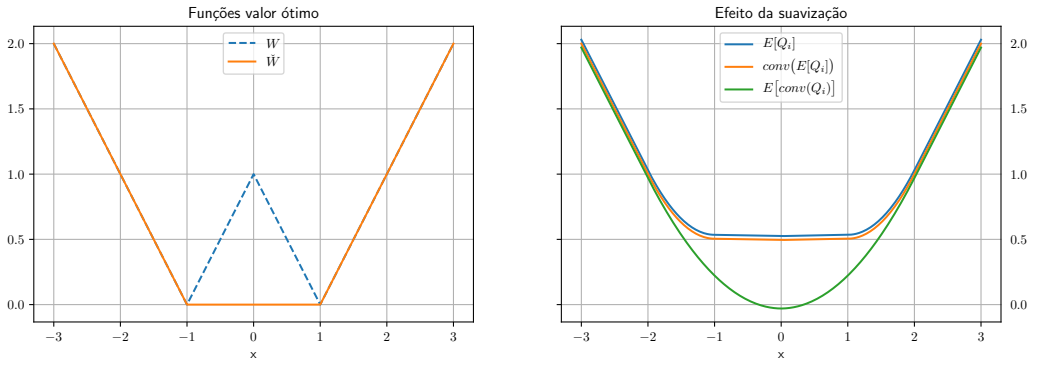


Figura 6.2: Diferença entre efetuar a relaxação lagrangiana antes ou depois de calcular a média.

6.2.4 Geração de cortes de Benders reforçados

Vimos que os cortes Lagrangianos são mais precisos do que os cortes de Benders, mas o seu cálculo é significativamente mais custoso. De fato, para determinar o multiplicador de Lagrange ótimo, devemos resolver um problema de otimização onde, a cada iteração, resolvemos o problema inteiro-misto de cada estágio, em sua formulação conjunta. Por outro lado, para obter cortes de Benders, basta resolver a relaxação linear de cada estágio, que é um problema convexo e separável, o que é muito mais rápido do que uma única iteração do cálculo do corte Lagrangiano.

Um compromisso entre estes cortes é o corte de Benders *reforçado*. Sua motivação é o fato de a *função relaxada* ser o máximo de funções lineares. Assim, podemos calcular um plano suporte à relaxação convexa de $\bar{\mathcal{Q}}$ calculando $\bar{g}$ a partir da equação (6.9), para um valor de u que não seja necessariamente o ótimo para x_{ini}. Assim, não precisamos calcular u^* tal que $\bar{g}(x_{\text{ini}}, u^*) = \bar{\mathcal{Q}}(x_{\text{ini}})$,

o que exige diversas iterações do algoritmo de subgradiente, e obtemos um plano suporte de $\bar{\mathcal{Q}}$, que será exato em um ponto x'_{ini}, provavelmente distinto de x_{ini}. Este ponto estará tão mais distante de x_{ini} quanto for diferente o multiplicador u escolhido do multiplicador ótimo u^*.

Resta apenas obter uma heurística para escolher o multiplicador u a ser utilizado. Uma possibilidade natural é utilizar o multiplicador de Benders, cujo cálculo é, como vimos, simples. Assim, podemos em uma etapa preliminar resolver a relaxação linear, separável, e calcular o multiplicador médio u para o problema conjunto. Em seguida, utilizando a formulação conjunta, calculamos $\bar{g}(x_{\text{ini}}, u)$ resolvendo o problema de otimização inteiro-misto de (6.9). Este procedimento é conhecido como "cortes de Benders reforçados", exatamente por calcular, primeiro, a inclinação do corte via Benders e, em seguida, "reforçar" este corte, calculando a função dual para o problema inteiro-misto.

Mesmo no caso em que este procedimento é feito para cada uma das funções duais, g_i, há um ganho pois, como vimos, a relaxação linear não é, necessariamente, ótima, como a relaxação convexa. Além disso, no caso em que calculamos $\bar{g}$, estamos, de fato, calculando um plano suporte para $\bar{\mathcal{Q}}$, o que também não é necessariamente verdade no caso da média das funções duais, $\sum_i p_i g_i(x_{\text{ini}}, u_i)$.

6.3 Geração de cortes com aversão a risco

No caso de problemas com aversão a risco, ainda vale $\rho(\operatorname{conv} Q(x,\Xi)) \leqslant \operatorname{conv} \rho(Q(x,\Xi)) \leqslant \rho(Q(x,\Xi))$, como mostrado no teorema 3.3. Isso novamente sugere que vale a pena efetuar cortes para $\rho(Q(x,\Xi))$, a função "conjunta", em vez de cada uma das funções $Q(x,\xi)$. Para calcular um corte para $\bar{\mathcal{Q}}(x) = \rho(Q(x,\Xi))$, é preciso formular esta função como um único problema de otimização, da mesma forma que fizemos na seção 6.2.3, para em seguida poder construir sua relaxação Lagrangeana, e então obter um corte, seja Lagrangiano, seja de Benders reforçado. Entretanto, a soma das funções objetivo não fornece o resultado correto, já que temos que aplicar uma medida de risco aos custos de cada cenário.

Nesta seção, iremos apresentar como construir estes problemas conjuntos, primeiro de forma genérica, na seção de representações poliedrais de funções, e em seguida no caso particular do CVaR, que será importante para nossas aplicações. Vale ressaltar que estas representações poliedrais são bastante flexíveis, e por isso são bastante fáceis de combinar para formular funções de custo futuro para medidas de risco construídas a partir de medidas simples.

6.3.1 Representações poliedrais e geração de cortes

Para motivar a construção das representações poliedrais, iremos apresentar, primeiramente, uma nova formulação conjunta, análoga à da equação (6.8), que recapitulamos abaixo:

$$\begin{aligned} \bar{\mathcal{Q}}(x_{\text{ini}}) = \min_{x_0, x_i, y_i} \quad & \textstyle\sum_i p_i (c^\top x_i + d^\top y_i) \\ \text{s.a} \quad & Ax_i + By_i = b + Tx_0 + w_i, \qquad \forall i \\ & (x_i, y_i) \in D, \qquad \forall i \\ & x_0 = x_{\text{ini}} \end{aligned}$$

É possível isolar as funções objetivo de cada estágio, $p_i(c^\top x_i + d^\top y_i)$, em variáveis t_i para cada cenário. Como x_i e y_i são variáveis de decisão, é possível formular o problema acima substituindo

cada termo na função objetivo por t_i, e adicionando a restrição de igualdade correspondente:

$$\begin{aligned} \bar{\mathcal{Q}}(x_{\text{ini}}) = \min_{x_0, x_i, y_i, t_i} \quad & \textstyle\sum_i p_i t_i \\ \text{s.a} \quad & Ax_i + By_i = b + Tx_0 + w_i, \qquad \forall i \\ & (x_i, y_i) \in D \qquad \forall i \\ & c^\top x_i + d^\top y_i = t_i, \qquad \forall i \\ & x_0 = x_{\text{ini}} \end{aligned} \tag{6.11}$$

Também seria possível incluir as restrições que determinam t_i como restrições de desigualdade, ou seja,

$$c^\top x_i + d^\top y_i \leqslant t_i,$$

já que, ao minimizarmos em t_i, e como as probabilidades p_i são não-negativas, iremos necessariamente atingir a igualdade.

De forma um pouco mais abstrata, podemos dizer que re-escrevemos a função de custo futuro $\bar{\mathcal{Q}}$ através do seguinte problema de otimização:

$$\begin{aligned} \bar{\mathcal{Q}}(x_{\text{ini}}) = \min_{t_i} \quad & \textstyle\sum_i p_i t_i \\ \text{s.a} \quad & Q(x_{\text{ini}}, \xi_i) \leqslant t_i, \qquad \forall i \end{aligned} \tag{6.12}$$

onde as funções de valor ótimo $Q(x_{\text{ini}}, \xi_i)$ correspondem a cada um dos cenários. A formulação conjunta pode ser vista como a expansão de $Q(x_{\text{ini}}, \xi_i)$ "dentro" do problema 6.12, incluindo suas variáveis e restrições.

Chamamos esta construção de *formulação poliedral* porque usamos uma representação da média ponderada dos custos $Q(x_{\text{ini}}, \xi_i)$ por um problema de otimização linear, que fornece, portanto, uma função cujo epígrafo é um poliedro.

Para o caso de medidas de risco mais gerais, iremos construir a formulação do problema de otimização que calcula a função de custo futuro através de representações poliedrais destas medidas de risco. A ideia é construir um problema de otimização, similar ao que foi apresentado em (6.12), mas cuja solução, em vez da média dos valores $Q(x_{\text{ini}}, \xi_i)$, corresponda à aplicação da medida de risco desejada.

De forma geral, uma função $f : \mathbb{R}^N \to \mathbb{R}$ admite uma representação poliedral quando existe um poliedro $P \subset \mathbb{R}^N \times \mathbb{R}$ que é o epígrafo de f. Neste caso, é possível calcular $f(x)$ através da solução do problema de otimização

$$\begin{aligned} f(x) = \min_{t} \quad & t \\ \text{s.a} \quad & (x, t) \in P \end{aligned} \tag{6.13}$$

Com tal representação, podemos construir um problema de otimização "conjunto"

$$\begin{aligned} \bar{\mathcal{Q}}(x_{\text{ini}}) = \min_{t, t_i} \quad & t \\ \text{s.a} \quad & Q(x_{\text{ini}}, \xi_i) \leqslant t_i, \qquad \forall i \\ & (T, t) \in P \end{aligned} \tag{6.14}$$

onde o vetor T contém todas as realizações das funções de custo futuro Q_i: $T = (t_1, t_2, \ldots, t_N)$.

Um exemplo de medida de risco com representação poliedral é o máximo: dados N valores t_i, o máximo entre eles pode ser calculado pelo problema de otimização linear

$$\begin{aligned} \max_i(t_i) = \min_{t} \quad & t \\ \text{s.a} \quad & t \geqslant t_i \ \forall i \end{aligned} \tag{6.15}$$

e assim, naturalmente, se quiséssemos calcular $\max_i[Q(x_{\text{ini}}, \xi_i)]$ bastaria resolver

$$\begin{array}{ll} \min\limits_{t} & t \\ \text{s.a} & Q(x_{\text{ini}}, \xi_i) \leqslant t_i, \qquad \forall i \\ & t \geqslant t_i \qquad\qquad\quad \forall i \end{array}$$

Esta forma pode parecer simples, mas de fato a *descrição* do poliedro P poderia ser bastante complexa. Um exemplo típico é dado pela função $f(x) = \max_i\{\, c_i^\top x + d_i \,\}$, que corresponde ao poliedro

$$P = \{\, (x,t) \mid c_i^\top x + d_i \leqslant t \ \forall i \,\}.$$

Se a quantidade de "cortes" $c_i^\top x + d_i$ for muito grande, a complexidade de representação pode também ser muito grande.

Por isso, estende-se a noção de representação poliedral para se permitirem variáveis auxiliares $u \in \mathbb{R}^M$:

$$\begin{array}{rll} f(x) = & \min\limits_{t,u} & t \\ & \text{s.a} & (x,u,t) \in \tilde{P} \end{array} \tag{6.16}$$

onde $\tilde{P}$ agora é um poliedro em $\mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R}$. Em diversas ocasiões (e, em especial, para o CVaR, como veremos na seção a seguir) a descrição de $\tilde{P}$ pode ser muito menos complexa do que a de P. Além disso, como a projeção de um poliedro ainda é um poliedro, e como P é a projeção de $\tilde{P}$ em $\mathbb{R}^N \times \mathbb{R}$, vemos que esta extensão para variáveis auxiliares não adiciona novas funções à classe das que admitem uma representação poliedral: ela apenas permite avaliar f com menor complexidade computacional, se Q possuir uma descrição mais simples do que P.

Da mesma forma que fizemos na passagem de (6.12) para (6.14), podemos usar esta formulação mais geral para obter um problema de otimização único que calcule $f(T)$ onde os valores t_i em T correspondem aos custos de cada um dos cenários, $Q(x_{\text{ini}}, \xi_i)$:

$$\begin{array}{rll} f\big(Q(x_{\text{ini}}, \xi_i)\big) = & \min\limits_{t,t_i,u} & t \\ & \text{s.a} & Q(x_{\text{ini}}, \xi_i) \leqslant t_i, \qquad \forall i \\ & & (T,u,t) \in \tilde{P} \end{array} \tag{6.17}$$

Um exemplo interessante deste fenômeno é dado pela função $s_k(x)$ que dá a soma das k maiores entradas do vetor $x \in \mathbb{R}^n$. Esta função é convexa, e sua representação mais simples é como o máximo de todas as $\binom{N}{k}$ formas de construir somas de k coordenadas de x: $s_k(x) \geqslant x_{i_1} + x_{i_2} + \ldots + x_{i_k}$ para todas as possíveis escolhas de índices $1 \leqslant i_1 < i_2 < \ldots < i_k \leqslant N$. Por exemplo, se $N = 5$ e $k = 2$, teremos as 10 restrições

$$\begin{array}{llll} s_2(x) \geqslant x_1 + x_2 & s_2(x) \geqslant x_2 + x_3 & s_2(x) \geqslant x_3 + x_4 & s_2(x) \geqslant x_4 + x_5 \\ \qquad\geqslant x_1 + x_3 & \qquad\geqslant x_2 + x_4 & \qquad\geqslant x_3 + x_5 & \\ \qquad\geqslant x_1 + x_4 & \qquad\geqslant x_2 + x_5 & & \\ \qquad\geqslant x_1 + x_5 & & & \end{array} \tag{6.18}$$

Entretanto, se os parâmetros N e k forem um pouco maiores, por exemplo se tivéssemos $N = 85$ e $k = 8$, esta representação direta demandaria um número gigantesco de desigualdades: 48 124 511 370, quase 50 bilhões.

Por outro lado, é possível construir uma representação muito mais compacta de $s_k(x)$ se utilizarmos variáveis auxiliares. Seja v uma variável auxiliar (cuja função será representar a k-ésima maior

entrada de x), e sejam u_i variáveis que representam o quanto x_i está acima de v: $u_i = [x_i - v]_+$. Em desigualdades, temos

$$u_i \geqslant 0 \quad \text{e} \quad u_i \geqslant x_i - v.$$

Observe que, se v de fato for a k-ésima maior entrada de x, a expressão $kv + \sum u_i$ corresponde à soma das k maiores entradas: apenas $k-1$ valores de u_i serão maiores do que zero, justamente aqueles correspondentes às maiores entradas do vetor x, então para cada um deles temos um termo $v + u_i = x_i$, e enfim o termo $v + 0$ para a k-ésima maior entrada.

Se v fosse maior do que a k-ésima entrada, teríamos menos u_i's positivos, mas então "sobrariam" também termos v maiores, que, aparecendo k vezes na expressão $kv + \sum u_i$, a tornarão maior. Por outro lado, se v fosse menor do que a $k+1$-ésima maior entrada de x, então teríamos mais u_i's positivos, e estes também fariam que a soma ficasse maior, pois apenas k deles, pareados com v, já corresponderiam à soma das k maiores entradas de x. Isto mostra que o seguinte problema de otimização

$$\begin{aligned} \min_{u,v} \quad & kv + \textstyle\sum u_i \\ \text{s.a} \quad & u_i \geqslant 0 \\ & u_i \geqslant x_i - v \end{aligned}$$

de fato corresponde à soma das k maiores coordenadas de x. Observe que este problema necessita de apenas $2N+1$ variáveis e $2N$ desigualdades, independentemente do valor de k. Assim, para $N = 85$ e $k = 8$ do exemplo anterior, basta construir um problema com 171 variáveis e 170 restrições.

6.3.2 Representação poliedral do CVaR

O CVaR-α de uma variável aleatória X é a média condicional dos seus valores que estão acima de seu quantil $1-\alpha$: estes representam, de certa forma, os "α piores casos" da variável X. Quando X é uma variável de distribuição contínua, esta definição não é ambígua, já que o valor v correspondente ao quantil $1-\alpha$ tem probabilidade zero. Assim, podemos escrever

$$\mathsf{CVaR}_\alpha[X] = \mathbb{E}\left[X | X \geqslant v\right], \quad \text{onde } \mathbb{P}[X \geqslant v] = \alpha.$$

Entretanto, se X for uma variável aleatória com distribuição discreta, esta definição não pode ser aplicada, já que, em função do valor de α, mesmo que o valor v correspondente ao quantil $1-\alpha$ esteja bem determinado, não haverá um número inteiro de realizações a serem somadas. Por exemplo, se quiséssemos o CVaR-10% de uma variável X com 85 realizações, o que significaria a média dos 8.5 maiores valores do vetor $x \in \mathbb{R}^{85}$?

Vimos, na seção anterior, que a *soma* das 8 maiores entradas de um vetor x pode ser representada como

$$\begin{aligned} s_8(x) = \min_{u,v} \quad & 8v + \textstyle\sum u_i \\ \text{s.a} \quad & u_i \geqslant 0 \\ & u_i \geqslant x_i - v \end{aligned} \tag{6.19}$$

e que, se em vez das 8 maiores entradas quiséssemos as 9 maiores entradas, bastaria utilizar $9v + \sum u_i$ na função objetivo. Ora, a média das k maiores entradas pode ser simplesmente calculada dividindo $s_k(x)$ por k, e portanto o problema de otimização

$$\begin{aligned} \min_{u,v} \quad & v + \tfrac{1}{k}\textstyle\sum u_i \\ \text{s.a} \quad & u_i \geqslant 0 \\ & u_i \geqslant x_i - v \end{aligned} \tag{6.20}$$

irá calcular a média das k maiores entradas do vetor x, e portanto o $\mathsf{CVaR}\text{-}\frac{k}{N}$ da variável aleatória X.

Assim, é natural tomar esta última forma como a *definição* do CVaR para variáveis com distribuição discreta, bastando para isso admitir que k possa tomar qualquer valor real entre 0 e N, correspondendo à proporção exata $\alpha = \frac{k}{N}$. Esta forma dá uma interpolação simples para os valores do CVaR que correspondem a um número inteiro de realizações de X, e ainda permite explicar porque, no caso de "termos fracionários", é exatamente o menor que entra com a "probabilidade restante para somar α": Neste caso, o valor de v no problema (6.20) estará determinado, e igual à $\lceil \alpha N \rceil$, e a variável u_i correspondente será zero.

Por fim, observamos que a construção que fizemos pode ser generalizada, se admitirmos uma formulação mais abstrata para a "média condicional" $\frac{1}{k}\sum u_i$, levando à definição habitual para variáveis aleatórias quaisquer que pode ser encontrada em [RU00, thm. 1, pg. 5]:

$$\mathsf{CVaR}_\alpha[X] = \inf_{z \in \mathbb{R}} \; z + \frac{1}{\alpha}\,\mathbb{E}\left[[X - z]_+\right].$$

6.3.3 Representação poliedral de combinações de medidas de risco

Uma técnica comum em programação estocástica com aversão a risco é criar medidas de risco a partir de combinações convexas de outras medidas. Um exemplo bastante comum, em uso no setor elétrico, é o *mean-cvar*, que realiza uma ponderação entre o CVaR e o valor esperado para a variável aleatória X:

$$\rho_{\lambda,\alpha}(X) = (1-\lambda)\,\mathbb{E}\left[X\right] + \lambda \mathsf{CVaR}_\alpha[X].$$

Nesta seção, omitiremos o parâmetro α do CVaR para não sobrecarregar a notação, já que estaremos interessados, principalmente, na combinação entre eles.

Para representar $\rho\big(Q(x_{\text{ini}}, \xi_i)\big)$, podemos simplesmente "empilhar" as representações poliedrais de cada uma das medidas de risco envolvidas, em um único problema de otimização:

$$\begin{aligned}
f\big(Q(x_{\text{ini}}, \xi_i)\big) = \min_{t, t_i, u} \quad & t & & & (6.21)\\
\text{s.a} \quad & Q(x_{\text{ini}}, \xi_i) \leqslant t_i, & \forall i \quad & \text{Cenários} \\
& \textstyle\sum_i p_i t_i \leqslant t_{mean} & & \text{Média} \\
& v + \tfrac{1}{k}\textstyle\sum u_i \leqslant t_{cvar} & & \;\; | \\
& u_i \geqslant 0 & & \mathsf{CVaR} \\
& u_i \geqslant x_i - v & \forall i \quad & \;\; | \\
& (1-\lambda) t_{mean} + \lambda t_{cvar} \leqslant t & & \textit{mean-avar}
\end{aligned}$$

Assim, podemos ver que trabalhar com representações poliedrais permite uma grande flexibilidade de modelagem: cada uma delas é responsável pelo "cálculo" de uma função convexa f no interior do problema de otimização, e podemos compor estas funções ao utilizar as variáveis epigráficas t de uma representação como argumentos x para uma outra função (também convexa) g.

6.4 Modificação do algoritmo de PDDE

As considerações acima sugerem que a parte de construção de cortes na PDDE pode ser alterada para usar cortes para a função $\bar{\mathcal{Q}}$ diretamente, com o objetivo de aproximá-la com menor *gap*. Assim, durante a etapa *backwards*, em vez de resolver um problema para cada cenário ξ^i, iremos

1. Montar o problema conjunto para $\bar{\mathcal{Q}}$, com todos os cenários;
2. Calcular um multiplicador de Lagrange inicial u_0;
3. Construir um corte a partir da relaxação Lagrangiana e da função dual $\bar{g}$.

O multiplicador inicial u_0 pode ser escolhido de diversas formas, mas a mais natural será resolver a relaxação linear e calcular o multiplicador médio, pois este pode ser utilizado tanto para iniciar a busca no caso de cortes Lagrangianos como para os cortes de Benders reforçados. Em seguida, o corte, Lagrangiano ou de Benders reforçado, será calculado através da função dual $\bar{g}$, obtida através da relaxação lagrangiana do problema conjunto.

7 Prova de conceito

Nesta seção, apresentamos os resultados da aplicação da metodologia de cortes conjuntos em diversos problemas de otimização estocástica. Após uma breve discussão dos objetos matemáticos envolvidos, trataremos de um modelo convexo simples, para familiarizar o leitor com alguns comportamentos que já podem ser observados neste caso. Em seguida, apresentaremos dois modelos não convexos, ainda simples, para observar o efeito dos diferentes tipos de cortes na aproximação da função de custo futuro.

Enfim, também apresentaremos alguns resultados preliminares quanto à aplicação dos cortes conjuntos em um modelo reduzido de operação hidrotérmica.

7.1 Aproximações da função de custo futuro

Relembramos, conforme as equações (3.1) e (3.2), que a função de custo futuro é dada pela recorrência da programação dinâmica

$$\bar{\mathcal{Q}}^k_{t-1}(x_{t-1}) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{x_t} & f_t(x_t, y_t) + \bar{\mathcal{Q}}^k_t(x_t) \\ \text{s.a} & (x_{t-1}, x_t, y_t, \xi_t) \in X_t \end{array}\right]$$

As funções $\mathfrak{Q}^k$, aproximações de $\bar{\mathcal{Q}}$ construídas com k cortes da etapa *backward* na PDDE, são representadas como o máximo de funções lineares:

$$\mathfrak{Q}^k_{t-1}(x_{t-1}) = \max_{1 \leqslant j \leqslant k} \{c_j^\top x_{t-1} + d_j\}$$

Uma função auxiliar, importante para a análise do algoritmo de PDDE, é a média dos problemas de otimização do estágio seguinte, onde usamos uma aproximação $\mathfrak{Q}^k_t$ para a função de custo futuro $\bar{\mathcal{Q}}_t$. De fato, para obter o valor do corte na etapa *backward*, resolvemos os problemas do estágio seguinte, com a aproximação corrente da função de custo futuro, e efetuamos a média para todos os cenários. Esta operação corresponde a avaliar a função $\tilde{Q}$, que é dada exatamente pela mesma fórmula de $\bar{\mathcal{Q}}$, apenas substituindo $\bar{\mathcal{Q}}_t$ por $\mathfrak{Q}_t$ como função de custo futuro:

$$\tilde{Q}^k_{t-1}(x_{t-1}) = \mathbb{E}\left[\begin{array}{ll} \min\limits_{x_t} & f_t(x_t, y_t) + \mathfrak{Q}^k_t(x_t) \\ \text{s.a} & (x_{t-1}, x_t, y_t, \xi_t) \in X_t^{\xi_t} \end{array}\right] \tag{7.1}$$

Vale a pena notar que, em uma dada iteração do algoritmo de PDDE, os únicos cortes possíveis de serem calculados para a função de custo futuro $\bar{\mathcal{Q}}_{t-1}$ são aqueles obtidos através de $\tilde{Q}_{t-1}$, pois apenas esta última é possível de ser avaliada.

O algoritmo de PDDE possui duas propriedades muito importantes de monotonicidade. Primeiramente, como todos os cortes são estimativas inferiores para $\bar{\mathcal{Q}}_t$, temos que $\mathfrak{Q}_t^k$, com k cortes, também é menor do que $\bar{\mathcal{Q}}_t$. Assim, ao incluir esta desigualdade no problema (7.1), observamos também que $\tilde{Q}_{t-1}^k \leqslant \bar{\mathcal{Q}}_{t-1}$. Enfim, como os cortes são efetuados apenas calculando $\tilde{Q}_t$, sem nunca de fato poder acessar $\bar{\mathcal{Q}}_t$, na verdade, a cada iteração k, temos:

$$\mathfrak{Q}_t^k \leqslant \tilde{Q}_t^k \leqslant \bar{\mathcal{Q}}_t.$$

A segunda propriedade é que as estimativas serão sempre crescentes com o número de iterações k: ao acrescentar cortes, tanto $\mathfrak{Q}_t$ como $\tilde{Q}_{t-1}$ crescem em direção a $\bar{\mathcal{Q}}_t$ e $\bar{\mathcal{Q}}_{t-1}$, respectivamente:

$$\mathfrak{Q}_t^k \leqslant \mathfrak{Q}_t^{k+1} \qquad e \qquad \tilde{Q}_{t-1}^k \leqslant \tilde{Q}_{t-1}^{k+1}$$

7.2 Um problema de controle multi-estágio

Consideramos para efeito de ilustração o seguinte modelo de otimização, com variável de estado x_t unidimensional:

$$\begin{aligned} \min \quad & \mathbb{E}\left[\sum_{t=1}^{T} |x_t|\right] \\ \text{s.a} \quad & x_t = x_{t-1} + y_t + \xi_t \\ & y_t \in [-1, 1] \end{aligned} \tag{7.2}$$

A decisão local (ou controle) y_t está limitada em $[-1, 1]$, e temos uma incerteza ξ_t independente em cada estágio. O objetivo é minimizar o valor esperado da distância à origem, somado ao longo de todos os T estágios.

Há duas razões para a escolha deste problema. Primeiramente, sendo um problema unidimensional, é relativamente simples de observar graficamente as aproximações das funções de custo futuro produzidas pelo algoritmo de PDDE ou suas modificações. Veremos duas destas aproximações: tanto $\mathfrak{Q}_t$, representada pelos cortes; como $\tilde{Q}_t$, os problemas "de estágio seguinte". Em segundo lugar, por ser um problema relativamente simples, é possível calcular as funções de custo futuro $\bar{\mathcal{Q}}_t$ analiticamente, o que permite também verificar a qualidade das aproximações $\mathfrak{Q}_t$ e $\tilde{Q}_t$.

Para os exemplos de que trataremos, fixamos $T = 8$ e em cada estágio a incerteza ξ_t é dada por quatro cenários $\{-1.5, -0.5, 0.5, 1.5\}$, com probabilidades respectivamente 1/6, 1/3, 1/3 e 1/6. Nesse caso, a função de custo total para o último estágio, $Q_8(x_7, 0)$, está ilustrada na figura 7.1 a seguir, bem como a resultante função de custo futuro $\bar{\mathcal{Q}}_7(x_7)$.

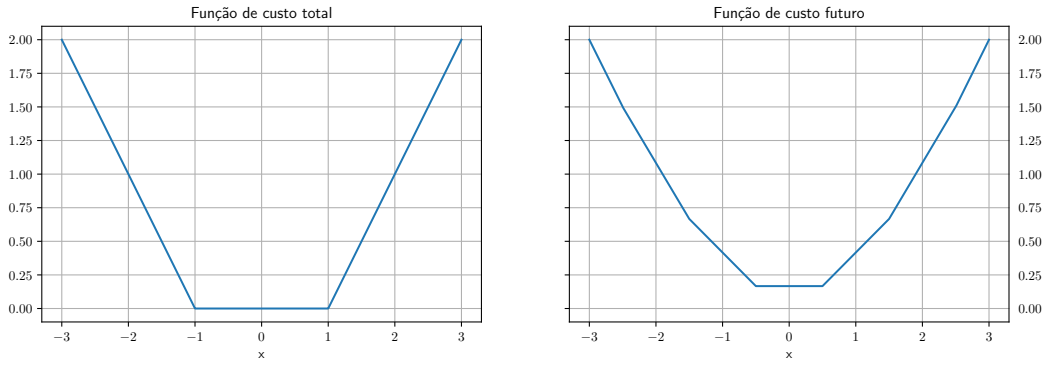


Figura 7.1: Funções de custo para o problema de controle

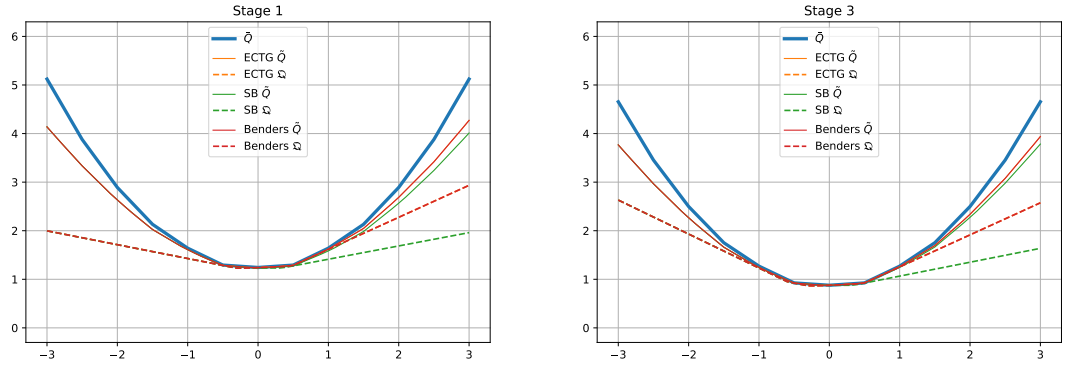
7.2.1 O caso convexo

Como o problema em (7.2) é convexo, teremos cortes exatos: nos pontos x_t^k onde calculamos os cortes na k-ésima iteração, vale

$$\mathfrak{Q}_t^k(x_t^k) = \tilde{Q}_t^k(x_t^k).$$

Vale notar que os cortes são exatos para $\tilde{Q}_t$, e não necessariamente para $\bar{\mathcal{Q}}_t$: com efeito, o algoritmo não tem como calcular $\bar{\mathcal{Q}}_t$ exatamente, a menos do último estágio, para o qual a função de custo futuro é suposta conhecida.

Apresentamos, na figura 7.2, as funções de custo futuro obtidas para este problema, usando cortes de Benders, de Benders reforçados, e para a formulação conjunta. Neste caso, o esperado é que todos os cortes sejam iguais, e portanto todas as $\mathfrak{Q}_t$ também. Na prática, pode haver uma certa variação entre os multiplicadores escolhidos por cada método, visto que, especialmente em programação linear, é bastante comum haver diversas soluções duais ótimas, correspondentes às faces delimitando o vértice da solução primal.



(a) Estágio 1

(b) Estágio 3

(c) Estágio 5

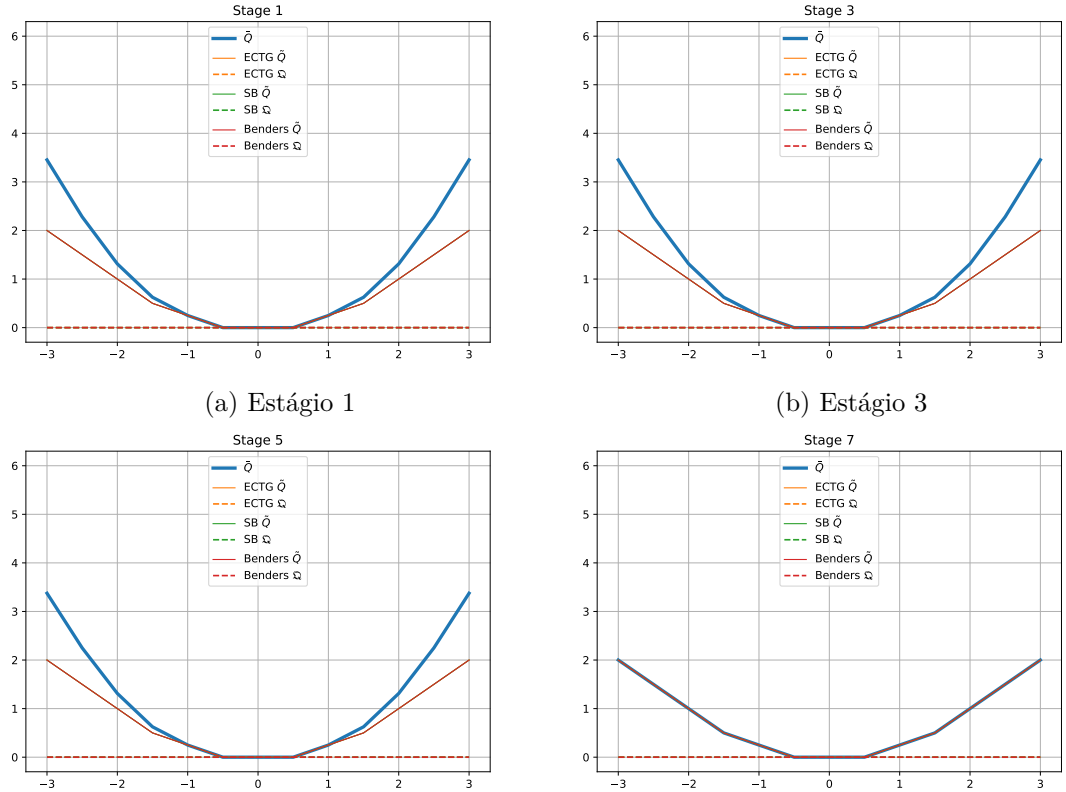
(d) Estágio 7

Figura 7.2: Funções de custo futuro para o problema convexo

Um outro fenômeno interessante a ser observado neste caso é que, conforme nos afastamos do horizonte de planejamento, e nos aproximamos do primeiro estágio, a aproximação da função de custo futuro $\bar{\mathcal{Q}}_t$ vai deteriorando. Nas figuras, isso pode ser observado como uma diferença cada vez maior entre a função analítica e a função obtida pela PDDE. Isso ocorre porque a função de custo futuro $\bar{\mathcal{Q}}_T$ é conhecida exatamente, o que permite resolver o problema de último estágio também exatamente. Mas, ao percorrer os estágios no sentido inverso do tempo, construímos

apenas aproximações, que são em seguida elas mesmas usadas para novas aproximações no estágio anterior, o que faz com que a qualidade de aproximação se degrade.

Para ilustrar a dependência do problema com relação à escolha dos cenários, apresentamos na figura 7.3 as funções de custo futuro e suas aproximações obtidas pelas diferentes metodologias no caso em que, em cada estágio, temos apenas 2 cenários equiprováveis, $\xi_t \in \{-0.5, 0.5\}$. Aqui, há menos margem para erro numérico, já que todos os cortes calculados são os cortes horizontais com altura zero. Desta forma, todas as metodologias produzem exatamente as mesmas estimativas, e não há as pequenas variações observadas no caso com 4 cenários, onde há cortes com inclinação diferente de zero.



(a) Estágio 1

(b) Estágio 3

(c) Estágio 5

(d) Estágio 7

Figura 7.3: Funções de custo futuro para o problema convexo com apenas dois cenários

7.2.2 O caso não-convexo

Agora, consideramos uma modificação do problema anterior, onde o controle y_t pode tomar apenas os valores 1 ou -1.

Apesar de o modelo não ser convexo, devido à restrição $y_t \in \{-1, 1\}$, a função de custo futuro $\bar{\mathcal{Q}}_t$ é convexa quando a incerteza é dada pelos mesmos 4 cenários escolhidos anteriormente, $\{-1.5, -0.5, 0.5, 1.5\}$, com probabilidades respectivamente 1/6, 1/3, 1/3 e 1/6, como pode ser visto na figura 7.4, análoga à figura 7.1 que foi apresentada no caso convexo. Este é o caso favorável, onde a incerteza do modelo é suficiente para suavizar as não convexidades de cada estágio. Apresentamos na figura 7.5, análoga à figura 7.2, as funções de custo futuro e suas aproximações para os estágios 1, 3, 5 e 7.

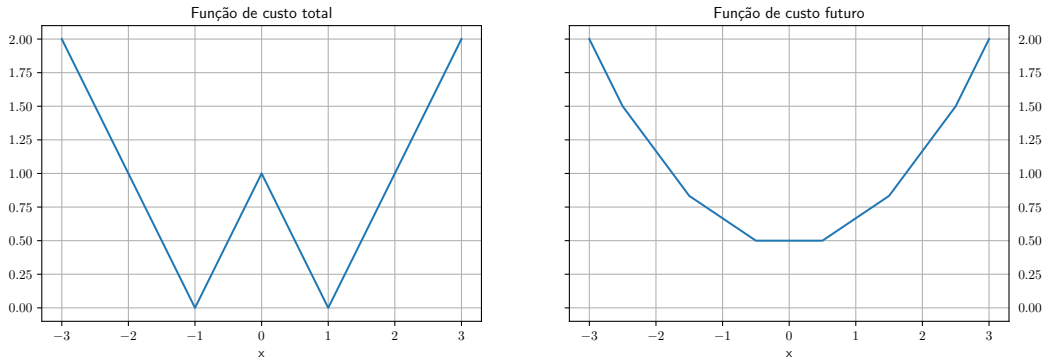
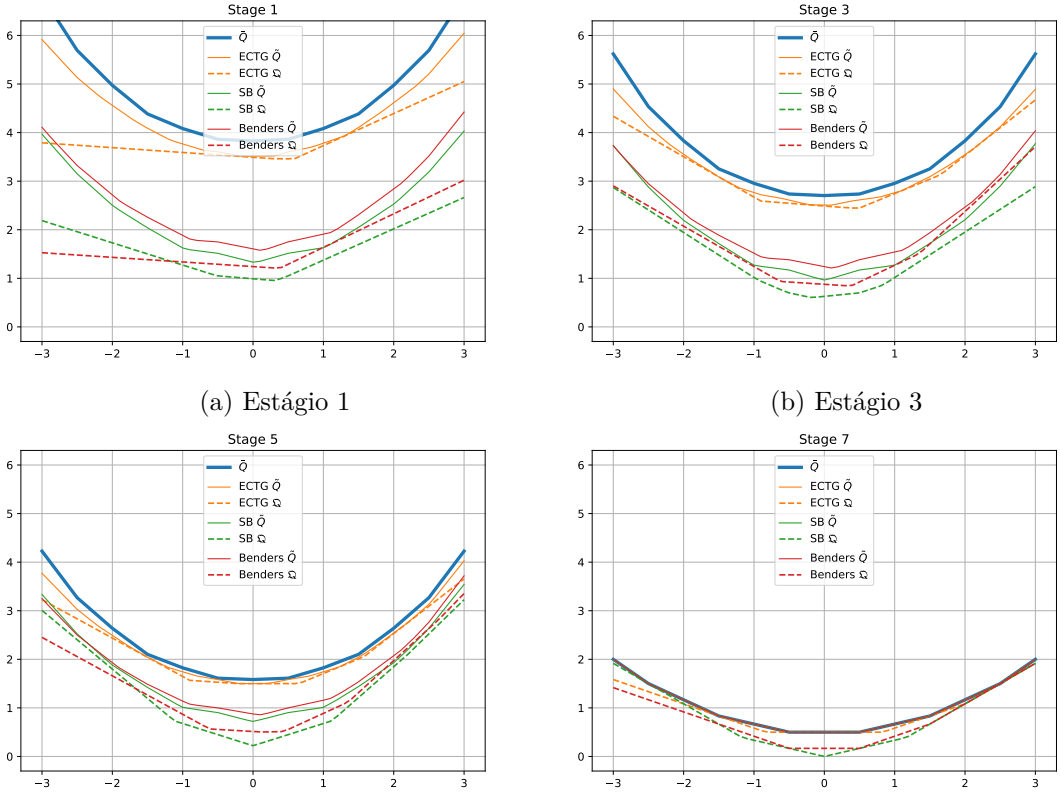


Figura 7.4: Funções de custo para o problema de controle não-convexo

E, novamente, com apenas 2 cenários equiprováveis, $\{-0.5, 0.5\}$, temos os resultados da figura 7.6.

Observe que as funções $\tilde{Q}_t$ estão "congeladas" do quinto estágio para trás, tanto para os cortes de Benders como para os cortes de Benders reforçados. Mais marcante ainda é que as $\tilde{Q}_t$ são não-convexas para estes cortes, enquanto que as $\bar{\mathcal{Q}}_t$ são, de fato, convexas. Ao contrário, os cortes para a formulação conjunta ao mesmo tempo (i) acompanham mais de perto a função $\bar{\mathcal{Q}}_t$, como (ii) resultam em funções $\tilde{Q}_t$ convexas.



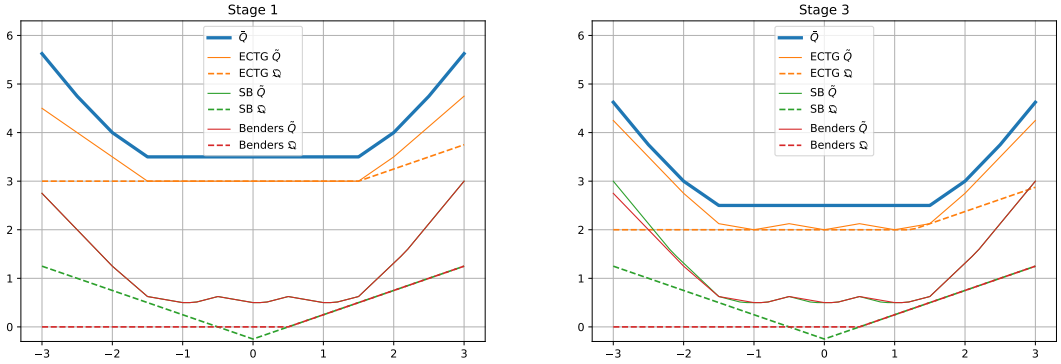
(a) Estágio 1

(b) Estágio 3

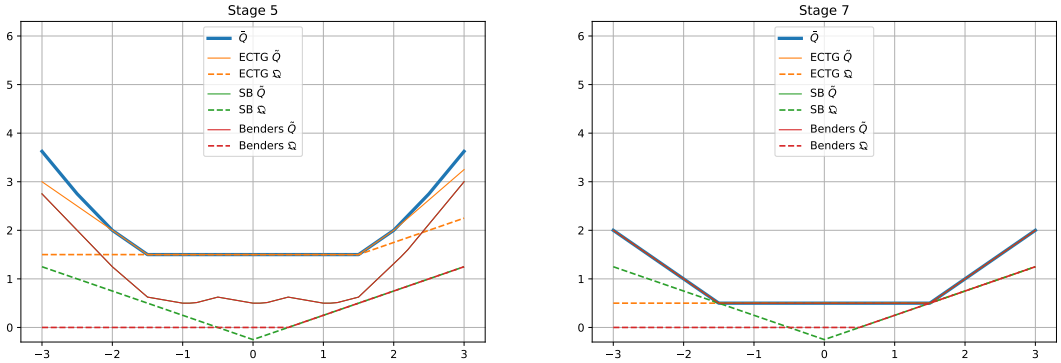
(c) Estágio 5

(d) Estágio 7

Figura 7.5: Funções de custo futuro para o problema não convexo



(a) Estágio 1 (b) Estágio 3



(c) Estágio 5 (d) Estágio 7

Figura 7.6: Funções de custo futuro para o problema não convexo com apenas dois cenários

8 Implementação

8.1 Desenvolvimento do protótipo

A implementação do protótipo computacional desenvolvido para a primeira fase do projeto de pesquisa *"Metodologias de Programação Estocástica Inteira"*, estabelecido entre o ONS e a COPPETEC/Instituto de Matemática da UFRJ, contrato GMC-CT-093/17, foi atualizada de forma a prover maior compartimentalização do programa, separando as etapas de entrada de dados, execução do algoritmo e exploração dos resultados obtidos por meio de tabelas e gráficos.

Uma vantagem dessa abordagem é permitir fazer alterações de dados e parâmetros de forma estruturada e eficiente. Desta forma, um usuário, mesmo sem conhecimento aprofundado do código ou da linguagem `Julia`, pode realizar diversos estudos utilizando o programa sem risco e de forma rápida.

8.1.1 Novo formato para entrada dos parâmetros

A entrada de dados para aplicar os parâmetros do problema agora é feita por intermédio de arquivos de formato `INI`. Esses arquivos são arquivos de texto simples com uma estrutura básica composta de seções e campos. Um exemplo, o qual chamamos de `exemplo_1.ini` pode ser observado abaixo:

```
[files]
data_file = ../examples/data_2REE.jl
outflow_file = ../examples/minoutflow.jl
noise_file = ../examples/eafs.npz

[solve]
nbiterations = 50
log = out.log
```

Listing 1: exemplo_1.ini

O parser de arquivo de formato `INI` utilizado vai transformar esse arquivo em uma estrutura de dados. No caso do protótipo desenvolvido, esse arquivo é passado como argumento do programa que, utilizando a biblioteca `ConfParser`, (disponível em `https://github.com/JuliaIO/ConfParser.jl`), converterá os valores desse arquivo para uma estrutura de dados em `Julia`.

É importante perceber que outro arquivo `INI` com as mesmas seções, campos e valores funcionará de forma equivalente. Note que a ordem das seções e dos campos dentro das seções nos arquivos `INI` não importam. Um outro exemplo de arquivo `INI`, chamado de `exemplo_2.ini` que é equivalente ao `exemplo_1.ini`, pode ser visto abaixo:

```
[solve]
nbiterations = 50
log          = out.log

[files]
data_file    = ../examples/data_2REE.jl
noise_file   = ../examples/eafs.npz
outflow_file = ../examples/minoutflow.jl
```

Listing 2: exemplo_2.ini

Para o protótipo desenvolvido, definimos um arquivo de base das configurações do programa chamado `main_config.ini`, que está disponibilizado no diretório `confs/`. O arquivo `main_config.ini`

usado como referência possui seis seções: `ModelConfig, files, solve, evaluation, others` e `sddip`. Para obter mais detalhes, veja a `seção 8.5.1`.

Para executar o programa, é necessário fornecer um arquivo no formato `INI`. Caso queira usar um arquivo diferente, digamos `params_base.ini`, esse deve possuir as seções e os campos preenchidos com o mesmo formato que o arquivo de base, `main_config.ini`. Por exemplo, existe uma seção chamada `ModelConfig` no `main_config.ini`. Assim, o arquivo `params_base.ini` precisa ter a mesma seção, levando em consideração letras maiúsculas e minúsculas. É importante destacar que não importa a ordem das seções dentro do arquivo.

Os campos não funcionam se estiverem colocados em outra seção. Por exemplo, o arquivo de dados é um parâmetro da seção `files`, assim, o programa não conseguirá localizar o arquivo de dados se o campo `data_file` estiver em outra seção. Cada seção está relacionada com a estrutura de dados do código, e cada campo com um parâmetro do problema.

O protótipo também permite adicionar mais de um arquivo `INI`, com o objetivo de auxiliar os usuários que precisem executar muitos casos com pequenas variações. Assim, o programa concatena as informações de cada arquivo antes de sua inclusão na estrutura de dados efetivamente utilizada. Caso ocorra um conflito de valores, prevalece o valor informado no último arquivo incluído.Dessa forma, pode-se realizar diversos testes utilizando um arquivo-padrão, por exemplo `params_base.ini`, e em seguida adicionar arquivos curtos com mudanças pontuais, por exemplo `mais_iteracoes.ini`, `vminop.ini`, etc. Com isto, ficam evidentes, a partir da leitura de linha de comando utilizada, quais foram as mudanças de parâmetros para cada rodada do experimento.

8.1.2 Migração do código para Julia 1

Uma das mudanças significativas entre o final de 2018 e 2019 foi a consolidação da linguagem `Julia`, que teve sua primeira versão estável publicada em 8 de Agosto de 2018. Aproveitando esta transição, tanto a biblioteca `JuMP`, que é a interface utilizada para modelar problemas de otimização em `Julia`, quanto o pacote `SDDP.jl`, pacote para resolver problemas estocásticos multi estágio usando Programação Dinâmica Dual Estocástica, também lançaram novas versões, respectivamente `JuMP 0.19`, em 15 de Fevereiro de 2019, e `SDDP.jl`, em 20 de Março de 2019.

A nova interface do `JuMP`, além de oferecer representações mais eficientes dos problemas de otimização, também permite modificar os coeficientes das restrições, o que é importante para possibilitar a representação da dependência temporal das afluências, por exemplo, utilizando um modelo periódico multivariado auto-regressivo com ruído multiplicativo.

Para disponibilizar e acessar estas novas funcionalidades, o protótipo desenvolvido durante a Fase 1 foi adaptado para a versão mais atual do pacote `SDDP.jl`,que utiliza a nova versão, 0.19, da biblioteca `JuMP`, e a nova versão (estável) da linguagem `Julia`. Assim, o código teve que ser adaptado para uma dupla mudança: tanto da linguagem em si, quanto das bibliotecas utilizadas. Se as mudanças da linguagem e da maior parte das funções da biblioteca `JuMP` foram relativamente pequenas, a nova organização do pacote `SDDP.jl`, capaz de representar estruturas de decisão mais complexas, representou uma maior carga de trabalho.

Além disso, a nova versão do pacote `SDDP.jl` é menos extensível, o que reduz a capacidade de alterar o algoritmo implementado. Por esta razão, esta nova versão ainda não é compatível com a metodologia de programação binária do pacote `SDDiP.jl`, desenvolvido para a versão anterior, e que modificava tanto a declaração das variáveis de estado como a construção dos cortes na etapa *backward*. Esta redução da extensibilidade também se revelou trabalhosa para a implementação da metodologia proposta neste convênio, pois exigiu efetuar modificações diretamente no código do pacote `SDDP.jl` para alterar o comportamento durante a etapa *backward*, necessárias para calcular os cortes tanto para *Strengthened Benders* como para a formulação conjunta.

Para nos assegurarmos de que a migração foi feita corretamente foram realizados testes de consistência comparando os resultados da versão anterior com os da nova versão.

8.1.3 Transformação do protótipo em um pacote `Julia`

Outro esforço realizado durante esta fase do projeto foi a adaptação da estrutura de código do protótipo para que este esteja mais próximo de um pacote `Julia`.

Este movimento tem duas grandes motivações: primeiramente, para possibilitar que nossa base de código possa ser testada com as ferramentas da própria linguagem `Julia`, o que permitirá um desenvolvimento com maior segurança, visto que podem ser feitos testes de compatibilidade, testes de prevenção de regressão e testes de cobertura de código. A segunda é que um pacote `Julia` pode ser muito mais facilmente distribuído e instalado, o que esperamos irá torná-lo mais acessível para uso.

Para tanto, realizamos uma primeira etapa de compartimentação do código, organizando as dependências entre funções e o fluxo de dados entre as diferentes etapas de processamento de forma que a maior parte das funções não trabalhe com variáveis globais. Esta parte, a mais custosa, foi necessária para possibilitar o carregamento de diferentes *solvers* em função da escolha do usuário, que poderia nem possuir algum dos *solvers* que utilizamos para desenvolvimento. Também aproveitamos esta etapa para reorganizar os arquivos de configuração em uma pasta separada do código executável principal.

Em seguida, passamos à etapa de inclusão de testes, de forma a verificar que as diferentes de configurações para o programa, detalhadas na seção 8.5, estavam funcionando corretamente. Ainda que não tenhamos desenvolvido testes unitários para o pacote, o teste de várias configurações já nos dá uma cobertura de código bastante satisfatória, acima de 80%.

8.2 Desenvolvimento de novos pacotes

Além do protótipo para planejamento da operação, no decorrer deste projeto também foram desenvolvidas diversas funções com o intuito de facilitar e automatizar a maneira com que manipulamos os modelos de problemas de otimização escritos com as bibliotecas `JuMP` e `SDDP.jl`. Essas funções foram organizadas em relação ao seu propósito e divididas em diferentes pacotes `Julia`, de forma a facilitar reaproveitamento de código. Abaixo listamos estes pacotes:

- `MIPManip.jl`: Baseado na interface da biblioteca `MathOptInterface`, esse pacote permite manipular problemas de otimização. Entre suas funcionalidades constam a capacidade de copiar variáveis, restrições e funções objetivo de um problema inteiro misto assim como anexá-las a outros.

- `Relaxation.jl`: Esse pacote implementa funcionalidades dedicadas a receber o modelo um problema inteiro misto escrito na biblioteca `JuMP` e modificá-lo para sua relaxação linear ou Lagrangeana, assim como funções que permitem retornar ao modelo original.

- `SDDP_SB.jl`: Por padrão, o pacote `SDDP.jl` só permite o cálculo de cortes do tipo *Benders* para resolver problemas inteiros mistos. Este pacote provê a funcionalidade de calcular cortes do tipo *Strenghtened Benders* para esses problemas.

- `Relink.jl`: Esse pacote contém adições ao pacote `SDDP.jl` que permitem o cálculo da formulação conjunta da função de custo futuro. A definição dessa formulação, assim como o embasamento teórico para as modificações ao SDDP usual feitas por este pacote, pode ser encontrada na seção 5 do relatório de Metodologia.

8.3 Contribuições à comunidade `Julia`

As funções encontradas nos pacotes `Relaxation.jl` e `SDDP_SB.jl` para realizar a relaxação linear de um problema inteiro misto e sua posterior restauração são baseadas nas que já haviam sido desenvolvidas por Oscar Dowson no pacote `SDDP.jl`. Durante o processo de testes das nossas funções, percebemos que as variáveis binárias não estavam sendo relaxadas apenas para o intervalo $[0, 1]$ mas sim para qualquer valor real. Assim, editamos nossas funções para realizarem a relaxação correta de uma maneira que outros possíveis limites da variável fossem preservados. Essas funções foram readaptadas para o contexto do `SDDP.jl` e enviadas para Oscar Dowson de forma que possam ser corrigidas em seu pacote.

8.4 Ambiente de programação

Os casos rodados como exemplo na seção 7 foram executados com o seguinte ambiente de programação:

- Hardware

  CPU: Intel i7-8700 (6 cores, 12 threads e clock de 4.60 GHz)

  RAM: 32Gb RAM, DDR, 2666 Mhz

- Ubuntu Linux 18.04.2
- Linguagem de Programação: Julia 1.1.1
- Solver: Gurobi 8.1
- Versão dos pacotes Julia utilizados:

Pacote	Versão	commit (SHA-1)
Desenvolvido para o convênio:		
DisjHTPlan	0.1.0	5c30496e
SDDP_SB	0.1.0	59a45430
Adaptado para o convênio:		
SDDP	0.0.0	f4570300
Próprio do Julia:		
ConfParser	0.1.0	88353bc9
Revise	2.1.6	295af30f
Documenter	0.22.4	e30172f5
GLPK	0.10.0	60bf3e95
GZip	0.5.0	92fee26a
Gurobi	0.6.0	2e9cd046
JuMP	0.19.2	4076af6c
Libz	1.0.0	2ec943e9
MathOptFormat	0.1.1	f4570300
MathOptInterface	0.8.4	b8f27783
NPZ	0.4.0	15e1cf62
OSQP	0.5.2	ab2f91bb
PyPlot	2.8.1	d330b81b
Libdl		8f399da3
Random		9a3f8284

8.5 Documentação da implementação

8.5.1 Arquivo `main_config.ini`

A seguir, mostraremos o arquivo `main_config.ini` que é usado como base para a entrada dos parâmetros. Como vimos anteriormente, esse arquivo possui seis seções: `ModelConfig`, `files`, `solve`, `evaluation`, `others` e `sddip`.

```
name = Examples/2REE

[ModelConfig]
nstages = 10
solver = Gurobi
timeout = inf
initmonth = 1
op_constr = 0,0,0,0,0
vminop_ratio = 0.2
p_violation = 0.0
pq_violation = 0.0
binary = false
useSB  = false
ECTG_SB = false

[files]
data_file = ../examples/data_2REE.jl
outflow_file = ../examples/minoutflow.jl
noise_file = ../examples/eafs.npz

[solve]
nbiterations = 50
cut_select = 0
log = out.log
cut_file = cuts.csv

[evaluation]
nsimulations = 100
sim_out_file = simul.csv

[others]
verbose = 1
debug_file = debug.log

[sddip]
precision = 1000.0
method = KelleyMethod
initialbound = 0.0
wait = 10
tol = 1e-2
maxit = 100
level = 0.6
benders = 1
strengthened_benders = 0
integeroptimality = 0
lagrangian = 1
```

Listing 3: main_config.ini

O diretório dos arquivos de saída será determinado pelo argumento do campo `name`. Por padrão esse diretório será DisjHTPlan/confs/$`name`/results, no qual $`name` representa o argumento do

campo `name`. Porém, pode ser determinado outro caminho para o diretório de saída, além do padrão. Isso é realizado por uma função chamada `joinpath`, em que se o argumento no campo `name` for um caminho absoluto, o diretório de saída será esse caminho absoluto. Para mais detalhes, visite `https://docs.julialang.org/en/v1/base/file/index.html#Base.Filesystem.joinpath`

No exemplo `main_config.ini` da seção `seção 8.5.1`, podemos ver que o caminho do diretório dos arquivos saída será DisjHTPlan/confs/Examples/2REE/results. Mas se o argumento do campo `name` for alterado de Examples/2REE para /home/user/htplan, o diretório será /home/user/htplan/results, pois /home/user/htplan é um caminho absuluto. Os campos da seção `ModelConfig` representam os parâmetros para a formulução do problema e de configurações do código. Cada campo corresponde a respectivamente:

Campo	Descrição	Tipo do Campo
`nstages`	número de estágios do problema	Int
`solver`	software de otimização utilizado, são suportados Gurobi e GLPK	String
`timeout`	tempo máximo (s) de processamento de um corte	Int
`initmonth`	mês inicial	Int
`op_constr`	array de restrições operacionais para o problema	Int,Int,Int,Int,Int
`binary`	opção de usar técnica de binarização para calcular o corte de Strengthened Benders	Binary
`useSB`	utilizar os cortes de Strengthened Benders	Binary
`ECTG_SB`	utilizar os cortes de Strengthened Benders junto com a formulação ECTG	Binary
`vminop_ratio`	Fração do volume máximo do reservatório abaixo da qual a restrição de Volume Mínimo Operativo é ativada	Float
`p_violation`	Penalidade para a violação do Volume Mínimo Operativo	Float
`pq_violation`	Penalidade para a violação da restrição de Vazão Mínima Operativa	Float

Tabela 1: Parâmetros para configuração do modelo

O array de inteiros `op_constr` contém 5 posições, correspondentes a diferentes restrições operacionais, baseado na tabela a seguir:

Posição	Restrição operacional
1	volume mínimo operacional
2	sem déficit preventivo
3	vazão mínima operativa, dada energia armazenada final
4	vazão mínima operativa, dada energia armazenada inicial
5	vazão mínima operativa, dada energia disponível

Tabela 2: Equivalência entre a posição do array `op_constr` e restrições operacionais

A formulação matemática destas restrições foi feita através de Restrições Disjuntivas, conforme descrito na primeira fase do projeto *"Metodologias de Programação Estocástica Inteira"*, e pode ser encontrada no relatório final correspondente [FPdCLdFCdC18]. As restrições 3, 4 e 5 se referem à política de operação de reservatórios com vazão defluente mínima definidas em função do volume armazenado, e permitem implementar as decisões contidas na resolução 2081/2017 da Agência Nacional de Águas [ANA17].

Além de ativar ou desativar restrições, existe a opção de aplicar uma restrição em sua forma binária ou contínua. Assim, há 3 valores válidos para os elementos do array `op_constr`, conforme a tabela abaixo:

Opção	Tipo da restrição
0	sem restrição
1	binária
2	contínua

Tabela 3: Valor de cada elemento do array `op_constr` e o tipo da restrição aplicada

Como exemplo, se no campo `op_constr` fosse colocado o valor `1,1,0,0,0` seria ativado as restrições de volume mínimo operacional e de não ter déficit preventivo, com variáveis binárias. Se fosse colocado o valor `2,2,0,0,0` seriam ativados as mesmas restrições, só que com variáveis contínuas.

Os campos da seção `files` representam, respectivamente, o caminho para acessar os arquivos:

Campo	Descrição	Tipo do Campo
`data_file`	entrada de dados	String
`outflow_file`	limites de vazão dos reservatórios	String
`noise_file`	entrada dos ruídos	String

Tabela 4: Parâmetros para configuração dos dados

Os campos da seção `solve` estão relacionados com os parâmetros do solver e representam, respectivamente:

Campo	Descrição	Tipo do Campo
`nbiterations`	quantidade de iterações realizadas	Int
`cut_select`	frequência de seleção dos cortes	Int
`log`	nome do arquivo de saída do arquivo de log	String
`cut_file`	nome do arquivo de saída dos cortes	String

Tabela 5: Parâmetros para configuração do solver

Os campos da seção `evaluation` são parâmetros da simulação da operação e representam, respectivamente:

Campo	Descrição	Tipo do Campo
`nsimulations`	quantidade de cenários forward realizados	Int
`sim_out_file`	nome do arquivo de saída da simulação	String

Tabela 6: Parâmetros para configuração da simulação

Os campos da seção `others` servem para fazer o debug do modelo e representam, respectivamente:

Campo	Descrição	Tipo do Campo
`verbose`	ativação do modo `verbose`	Int
`debug_file`	nome do arquivo de saída do debug	String

Tabela 7: Parâmetros para configuração do debug

Os campos da seção `sddip` estão relacionados com os parâmetros do método de programação inteira e representam, respectivamente:

Campo	Descrição	Tipo do Campo
`precision`	precisão da expansão binária para variáveis contínuas	Float
`method`	tipo do solver Lagrangiano utilizado	String
`initialbound`	o limite inferior inicial para o problema Lagrangiano dual	String
`wait`	parâmetro para reduzir o tamanho do passo no método Subgradiente Lagrangiano	String
`tol`	tolerância para término do método Lagrangiano	Float
`maxit`	limite de interações para o método Lagrangiano	Int
`level`	parâmetro entre 0 e 1 do nível para o método Lagrangiano de níveis	Float
`benders`	Benders	Int
`strengthened_benders`	Benders reforçado	Int
`integeroptimality`	Integralidade	Int
`lagrangian`	Lagrangiano	Int

Tabela 8: Parâmetros para configuração do sddip

Os campos `benders`, `strengthened_benders`, `integeroptimality` e `lagrangian` da seção `sddip` representam o padrão de corte para o método SDDiP. Por exemplo, se os valores forem benders = 4, strengthened_benders = 3, integeroptimality = 2 e lagrangian = 1, a cada 10 cortes o programa realizará 4 cortes de benders, 3 de strengthened benders, 2 de integralidade e 1 lagrangiano. Assim, um ciclo de corte é igual a soma dos valores desses campos e a cada ciclo será realizado a quantidade de cortes do tipo específico determinada pelo valor do campo.

8.5.2 Descrição dos arquivos de saída

O diretório dos arquivos de saída possui 4 arquivos: `allconfs.ini, out.log, runparams.txt` e `simul.csv`. O nome dos arquivos `out.log` e `simul.csv` podem ser alterados nas configurações.

O arquivo `runparams.txt` possui informações do caminho absuluto do diretório DisjHTPlan, dos arquivos `INI` utilizados e das configurações já tratadas pela biblioteca `ConfParser` e pelo

programa. Para uma vizualização mais simples das configurações, o arquivo `allconfs.ini` possui as informações sobre as configurações no formato `INI`.

O `out.log` é um arquivo de saída da biblioteca `SDDP`. Esse arquivo possui um relatório de estabilidade numérica do problema fornecido e, em seguida, uma tabela de evolução do algoritmo com 4 colunas representando, respectivamente: as iterações, a simulação estatística do limite inferior, o limite inferior e o tempo de execução total.

O `simul.csv` é o arquivo que possui todas as simulações de políticas do modelo com o número de replicações determinadas no campo `nsimulations` da seção `evaluation`. A amostragem é baseada no método de Monte Carlo, implementada na biblioteca `SDDP`.

8.5.3 Descrição das ferramentas de análise

Após a execução da rodada, é possível criar gráficos e gerar estatísticas com a própria biblioteca `DisjHTPlan`. Os gráficos são montados com auxílio do arquivo `graph_utils.jl`. Existem duas formas de usar as ferramentas: acessando as informações na memória RAM logo após a rodada no `REPL` ou pelo arquivo `CSV` salvo correspondente ao campo `sim_out_file` determinado na seção `evaluation`.

Para gerar os gráficos direto da memória basta incluir o arquivo `graph_utils.jl` no `REPL` e, em seguida, utilizar a função `do_plots`, que possui a sintaxe:

```
do_plots(results, sysvec, stage_vec, keys; save=false, hl = [],
outdir="./")
```

O argumento `results` representa o dicionário com os resultados, o próprio programa já coloca os resultados no dicionário referenciado pela variável com nome `results`, então basta escrever `results` nesse argumento.

O argumento `sysvec` representa os subsistemas equivalentes para os quais serão feitos os gráficos. Pode ser informado um vetor para referenciar mais de um subsistema. Por exemplo, caso queira montar os gráficos dos subsistemas 2 e 3, basta colocar `[2,3]` como argumento. O `stage_vec` são quais estágios serão colocados, o argumento pode ser passado como um intervalo de números, por exemplo, para os estágios de 1 a 12, podemos colocar `1:12`.

O argumento `keys` está relacionado com quais gráficos serão produzidos, quer dizer, se será produzido o gráfico de volume hídrico, custo operacional, etc. Nesse argumento, basta colocar `graphs_to_do` que é uma estrutura já definida no arquivo `graph_utils.jl`.

O argumento `save` determina se os gráficos serão salvos em alguma pasta. Caso o argumento seja `true` ou apenas mostrados no terminal, caso o argumento seja `false`. Se a opção de salvar for verdadeira, é obrigatório colocar um diretório de saída no argumento do `output`, caso contrário, o argumento é opcional. O `hl` tem como objetivo realçar alguma série específica no gráfico.

Após a execução da função `do_plots`, os gráficos serão produzidos na pasta designada pelo argumento `outdir`. Serão produzidos gráficos de `séries, quantis e violinos`, cujos parâmetros podem ser mudados no arquivo `graph_utils.jl`.

Para produzir os gráficos a partir do arquivo `CSV`, o processo tem algumas diferenças. Precisa-se abrir um terminal `Julia`, incluir os arquivos `system_definition.jl`, de dados `(data.jl)`, de volume mínimo (`minoutflow.jl`). Com isso, os dados do `CSV` são obtidos com a função `CSV.read` da biblioteca `CSV`, e em seguida, inclui-se os arquivos `assemble_data.jl,setup_plot_csv.jl, graph_utils.jl`. Depois, basta utilizar a função `do_plots` da mesma forma que no caso de produzir gráficos a partir do conteúdo da memória RAM. Um código com todos esses passos está disponível na pasta `examples` com o nome de `csvplot.jl`.

9 Aplicação a um sistema hidrotérmico

9.1 Modelo reduzido do SIN

Agora, iremos comparar estas metodologias em um modelo reduzido do sistema interligado nacional, composto por 2 subsistemas, cada um correspondendo a um reservatório equivalente, um total de 3 usinas térmicas, e uma linha de intercâmbio entre os subsistemas. Em todos os casos, a política é calculada de forma neutra a risco, usando apenas o valor esperado do custo.

Novamente, começaremos com um caso convexo, onde esperamos que tanto os cortes de Benders reforçados como os cortes para a formulação conjunta coincidam com os cortes de Benders tradicionais. Esta verificação irá reassegurar a consistência das implementações de ambos os cortes, além de servir como indicativo do acréscimo de custo computacional relativo a cada uma destas metodologias. Em seguida, implementaremos um modelo não-convexo, através das restrições de Volume Mínimo Operativo modeladas por restrições disjuntivas, conforme desenvolvido na primeira fase do convênio [FPdCLdFCdC18].

Também usaremos estes mesmos modelos para comparar a performance com o SDDiP, por enquanto apenas disponível para a versão 0.6 da linguagem Julia.

9.2 Caso risco-neutro

9.2.1 Caso convexo com 2 subsistemas

Para este caso, efetuamos 500 iterações *forward e backward*, construindo 500 cortes para a função de custo futuro, considerando um horizonte de planejamento de 12 meses.

Reportamos, na tabela 9, duas medidas de custo computacional: o tempo necessário para o cálculo dos cortes e o máximo de memória RAM utilizada. Também indicamos duas medidas para o valor ótimo do problema, o limite inferior de custo da política, calculado durante a otimização, e o custo médio da operação, estimado através da simulação de 500 novos cenários, após o cálculo da política. Enfim, deduzimos a partir do custo calculado em cada metodologia o *gap* de otimalidade da mesma, adotando o menor custo simulado como valor-alvo.

	Cortes utilizados			
Métrica	Benders	SB	Binary SB	ECTG
`Tempo (seg)`	30	142	1353	758
`Memória (GB)`	0.394	0.396	0.452	1.005
`Custo calculado (Bi R$)`	8.668	8.671	8.666	8.670
`Custo simulado (Bi R$)`	9.135	9.153	9.153	9.128
`Gap (%)`	5.04	5.01	5.06	5.02

Tabela 9: Resultados para o modelo reduzido com 2REE

Observamos que os valores do custo, tanto o limite inferior, resultado do problema de otimização, como a estimativa do limite superior, obtido por simulações, são bastante próximos qualquer que seja a forma de cálculo dos cortes, o que está de acordo com o esperado. As leves diferenças que há podem ser atribuídas a dois efeitos. Um primeiro, já comentado na seção anterior, é a diferenças entre soluções primais ou duais em função de uma ordem diferente de cálculo, induzida pela mudança do método para avaliar cortes. O segundo, relevante apenas no caso binário, é que a discretização da variável de estado torna este problema ligeiramente diferente dos demais, já que o estado final em cada estágio pode apenas assumir valores discretos. Isto pode ser visto como uma restrição suplementar, que tende a aumentar os custos. Além disso, os cortes calculados no

problema discreto também serão calculados somente para estes valores discretizados da variável de estado, o que também gera diferenças. O fato de os valores estarem todos bastante próximos é um bom indicativo de que a discretização escolhida resulta em um problema suficientemente próximo do original contínuo.

Além disso, observamos que o método de cortes de Benders é de fato o mais rápido de todos, por uma grande margem. Isto se deve a manter os problemas de otimização de cada estágio na memória, e simplesmente adicionar cortes ou mudar parâmetros de cenários, o que permite ao GUROBI recalcular soluções com bastante velocidade. Em todos os outros casos, isto não acontece, já que o uso de cortes reforçados implica a relaxação Lagrangiana das restrições de acoplamento temporal, o que modifica a função de custo do problema. Ainda assim, vale notar que os cortes de Benders reforçados na formulação decomposta, coluna `SB`, são significativamente mais rápidos do que a formulação conjunta (coluna ECTG) ou para a binarização de estado (coluna Binary SB). Isto pode ser explicado porque os cortes da formulação conjunta necessitam, também, a construção do problema conjunto a cada iteração *backward*, o que tem um custo computacional, e em seguida o problema de otimização a ser resolvido também é mais complexo. Já no caso da binarização do estado, o problema passa a incluir variáveis inteiras, e isto apresenta alguma dificuldade para o *solver*, ainda que o problema original fosse convexo.

9.2.2 Caso com 2 subsistemas e VMinOp

A diferença deste caso para o anterior é a presença de duas restrições não-convexas, correspondentes às restrições de Volume Mínimo Operativo, modeladas por restrições disjuntivas. Também efetuamos 500 iterações do algoritmo, e usamos os mesmos 500 cenários de simulação final, utilizados para avaliar as mesmas metodologias do caso anterior, que resultaram na tabela 10.

	Cortes utilizados			
Métrica	Benders	SB	Binary SB	ECTG
`Tempo (seg)`	41	759	1548	16854
`Memória (GB)`	0.422	0.418	0.446	1.089
`Custo calculado (Bi R$)`	10.222	10.410	10.417	11.170
`Custo simulado (Bi R$)`	12.231	12.227	12.166	12.209
`Gap (%)`	15.98	14.44	14.38	8.19

Tabela 10: Resultados para o modelo reduzido com 2REE e VMinOp

A primeira observação a ser feita neste caso é que os custos estão, todos, maiores do que os apresentados no caso anterior, o que é consistente com a inclusão das restrições operativas. Por exemplo, o custo resultante da simulação, de aproximadamente 12.2 bilhões, é 34 % maior do que 9.15 bilhões, valor correspondente sem as restrições de volume mínimo. Mesmo para cortes de Benders, que correspondem à relaxação linear dos problemas em cada estágio, o limite inferior é significativamente maior (18 %) do que no caso convexo. Já para o uso de cortes de Benders reforçados na formulação decomposta, seja com binarização do estado ou não, há um acréscimo da estimativa do limite inferior, de 10.2 para 10.4 bilhões. Entretanto, é com os cortes para a formulação conjunta que obtemos uma melhoria significativa deste valor, chegando a 11.17 bilhões. Ainda assim, estes valores estão, todos, aquém da estimativa por simulação, de 12.2 bilhões em qualquer um dos casos.

Além disso, podemos observar que o tempo de cálculo para os métodos também aumentou, mas de forma significativamente diferente em função da metodologia. O caso de cortes de Benders teve um leve acréscimo, devido à necessidade de incorporar as variáveis binárias durante as iterações

forward. Também para a metodologia de discretização do estado, o acréscimo relativo foi modesto (15 %), provavelmente devido à maior quantidade de variáveis binárias presentes, e também à maior dificuldade de cálculo dos cortes de Benders reforçados. Já os cortes de Benders reforçados no caso contínuo levaram significativamente mais tempo (mais de 5 vezes), sem, contudo, se tornar mais lentos do que os cortes no caso binário. Este aumento indica que há, de fato, uma maior dificuldade para resolver os problemas com estas restrições operativas, e o leve aumento do limite inferior indica que há um *gap* de dualidade nos problemas de cada estágio, que os cortes reforçados são capazes de preencher. Por fim, os cortes para a formulação conjunta agora são os mais demorados, necessitando 22 vezes mais tempo do que no caso convexo, e mais de 400 vezes mais tempo do que os cortes de Benders. Entretanto, estes são os únicos que são capazes de, efetivamente, reduzir o *gap* do problema como um todo, aproximando-se de valores típicos para o problema convexo: 8.5 % para o modelo não-convexo com restrições operativas, versus 5 % para o modelo convexo simples.

9.2.3 Teste de tempo com cortes de Benders reforçados

Ao analisar os resultados da tabela anterior, vemos que o tempo de execução do método que usa cortes de Benders reforçados para a formulação decomposta é bem menor do que no caso da formulação conjunta. Assim, para fazer uma comparação baseada em tempo de execução, rodamos o mesmo caso com 8 vezes mais cortes do que o anterior (4000 cortes), que possui um tempo de execução um pouco maior do que a formulação conjunta com 500 cortes.

Métrica	SB
`Tempo (seg)`	18154
`Memória (GB)`	0.474
`Custo calculado (Bi R$)`	10.498
`Custo simulado (Bi R$)`	12.141
`Gap (%)`	13.79

Tabela 11: Resultados para o modelo reduzido com 2REE e VMinOp

Podemos perceber que o decréscimo do *gap* ao executar 3500 cortes suplementares com a formulação decomposta foi de apenas 0.65 pontos percentuais, ao passo que o uso da formulação conjunta permitiu uma redução de 6.25 pontos percentuais. Este resultado sugere que de fato a formulação decomposta já está relativamente próxima de sua saturação, o que é provavelmente devido ao acúmulo de *gap*s de dualidade ao longo de diferentes cenários, enquanto que a formulação conjunta permitiu reduzir estes *gap*s.

9.2.4 Modelo reduzido periódico para VMinOp

Neste caso, incluímos um modelo periódico da operação, de forma a evitar o efeito de "fim de horizonte". Adotamos um fator de desconto anual de 10 %, de forma que o custo esperado será em torno de 10 vezes maior. Como podemos ver na tabela 12, o custo calculado é mais de 10 vezes maior, o que pode ser explicado porque não é mais razoável esvaziar os reservatórios no final do horizonte de planejamento, o que leva a uma operação que possui maior despacho térmico. Isso também pode ser visto no custo simulado, para o mesmo período operativo de 12 meses, que também é maior do que os custos simulados sem considerar a operação periódica.

Além do aumento esperado dos custos, observamos, também, uma redução no tempo de cálculo para todas as metodologias. Imaginamos que este efeito pode ser atribuído ao maior volume armazenado ao longo dos cenários, o que reduz a probabilidade de violação das restrições de volume

	Cortes utilizados			
Métrica	Benders	SB	Binary SB	ECTG
`Tempo (seg)`	21	420	833	2372
`Memória (GB)`	0.559	0.560	0.589	0.833
`Custo calculado (Bi R$)`	217.813	219.086	219.186	219.471
`Custo simulado (Bi R$)`	19.101	19.705	18.786	19.184

Tabela 12: Resultados para o modelo reduzido com 2REE e VMinOp, periódico

mínimo, e portanto reduz o número de vezes que as restrições de integralidade se tornam significativas nos estados visitados durante a etapa *forward* do algoritmo de programação dinâmica. Daí, com maior frequência mesmo os cortes de Benders serão exatos, o que também simplifica o cálculo dos cortes reforçados, já que o próprio problema com variáveis inteiras estará mais frequentemente sem *gap* de dualidade, sua relaxação linear já possuindo solução inteira. Este efeito é principalmente visível no caso dos cortes para a formulação conjunta, cujo tempo de simulação foi reduzido mais de 7 vezes.

9.2.5 Comparação com o Julia 0.6 para o modelo reduzido para VMinOp

Enfim, incluímos um estudo para a versão 0.6 da linguagem Julia, o que nos permite usar cortes Lagrangianos para o modelo binarizado, que converge, dado um número suficiente de cortes, para a solução exata do problema não-convexo. Apresentamos, na tabela 13 os resultados para os mesmos cortes, apenas substituindo os cortes para a formulação conjunta (que não estão disponíveis para esta versão) pelos cortes Lagrangianos.

	Cortes utilizados			
Métrica	Benders	SB	Binary SB	Binary Lagr
`Tempo (seg)`	328	612	1125	124100
`Memória (GB)`	1.816	2.518	2.944	2.444
`Custo calculado (Mi R$)`	10.222	10.400	10.399	10.175
`Custo simulado (Mi R$)`	11.874	11.842	11.850	11.885
`Gap (%)`	13.68	12.18	12.18	14.08

Tabela 13: Resultados para o modelo reduzido com 2REE e VMinOp, usando Julia 0.6

Em comparação com os casos para a nova versão da linguagem Julia e as novas versões das bibliotecas `JuMP` e `SDDP.jl`, os cortes de Benders reforçados estão mais rápidos na versão 0.6, que possui uma implementação diferente para a relaxação Lagrangiana e o cálculo de cortes reforçados. Entretanto, os cortes de Benders estão significativamente mais demorados, possivelmente porque nesta mudança a interface oferecida pela biblioteca `JuMP` com os *solvers* está mais eficiente, e pode ser melhor explorada quando resolve-se sempre o mesmo problema, sem mudar a função objetivo como necessário no caso dos cortes reforçados.

Além disso, podemos ver que os cortes Lagrangianos demandam um tempo de cálculo 202 vezes maior do que o uso de cortes de Benders reforçados, enquanto que, no caso anterior, da tabela 10, o uso dos cortes conjuntos era apenas 22 vezes mais demorado do que os mesmos cortes de Benders reforçados. Mais ainda, o uso de cortes Lagrangianos não leva a uma melhor cota inferior, o que ilustra não apenas a dificuldade de cálculo dos cortes Lagrangianos, mas também a necessidade de cálculo de um número talvez proibitivo de cortes para obter uma melhor solução do problema.

9.3 ρCTG

Nesta parte, replicamos alguns dos experimentos descritos na seção anterior considerando o planejamento da operação com aversão a risco. Relembramos que a medida de risco adotada será uma média entre o valor esperado e o CVaR, dada por:

$$\rho_{\lambda,\alpha}(X) = (1-\lambda)\,\mathbb{E}\,[X] + \lambda \mathsf{CVaR}_{\alpha}[X],$$

onde α determina o nível de aversão a risco do CVaR, e corresponde aos α cenários mais caros.

Começamos com uma escolha tradicional dos parâmetros de aversão a risco, dada por $\lambda = 0.15$ e $\alpha = 0.1$. Os resultados obtidos para cada uma das metodologias de cálculo de cortes foram: Neste caso, como incluímos uma medida de risco na função objetivo do problema de otimização, o

	Cortes utilizados		
Métrica	Benders	SB	ECTG
`Tempo (seg)`	41	771	15 190
`Memória (GB)`	0.419	0.421	1.111
`Risco calculado (Bi R$)`	14.541	14.718	15.511
`Custo simulado (Bi R$)`	12.583	12.576	12.560

Tabela 14: Resultados para o modelo reduzido com 2REE, VMinOp e aversão a risco

valor calculado pelo programa, correspondente à linha "Risco calculado", é uma cota inferior para o risco total, e não para o real custo de operação, cuja estimativa da média é dada na linha seguinte. Por conta disto, não é mais possível exibir o *gap* entre o valor obtido pela solução aproximada do problema e o custo de simulação.

Vemos, novamente, que o tempo de cálculo para os cortes na formulação conjunta (coluna ECTG) demandam maior tempo; de fato, a diferença é ainda maior com relação aos casos de cortes de Benders ou de Benders reforçados (coluna SB), o que provavelmente se explica pela maior complexidade dos problemas na formulação conjunta. Também notamos que a diferença entre o valor ótimo obtido com cortes de Benders e cortes de Benders reforçados (177 milhões, ou 1.2% do maior) é muito menor do que o ganho obtido ao usar cortes para a formulação conjunta (970 milhões e 793 milhões, respectivamente 6.2% e 5.11%).

Por fim, observamos que o custo simulado da operação está apenas levemente superior ao custo no caso neutro a risco, que era da ordem de 12.2 bilhões de reais para um ano, e passou para 12.5 bilhões de reais, que é um aumento da ordem de 2.5%. Podemos interpretar esta modificação como um sinal de que a política de operação avessa a risco não será tão alterada com relação ao caso neutro a risco.

Também realizamos um outro experimento, com uma parametrização diferente da aversão a risco. Aqui, demos um maior peso para o CVaR, utilizando $\lambda = 0.6$, mas também aumentamos a proporção de cenários a incluir, ao tomar $\alpha = 0.3$. Assim, em uma direção temos mais aversão a risco, ao aumentar λ, mas menos aversão, ao efetuar uma média sobre um maior número de cenários.

Neste experimento, obtivemos uma maior diferença entre os três casos, dos cortes de Benders para Benders reforçados para cortes efetuados na formulação conjunta. Ainda assim, os valor obtido com a formulação conjunta é maior, como esperado. Vale observar que o efeito líquido de nossa parametrização foi uma maior aversão a risco, o que pode ser visto tanto nos valores calculados, que estão aproximadamente 70% maiores, como nos custos simulados, que chegaram a 13.2 bilhões, o que corresponde a um aumento de 8% com relação ao caso neutro a risco.

	Cortes utilizados		
Métrica	Benders	SB	ECTG
`Tempo (seg)`	39	752	3778
`Memória (GB)`	0.874	0.928	1.013
`Risco calculado (Bi R$)`	24.561	25.570	26.471
`Custo simulado (Bi R$)`	13.257	13.234	13.227

Tabela 15: Resultados para o modelo reduzido com 2REE, VMinOp e uma outra parametrização para aversão a risco

10 Conclusão

Neste relatório, apresentamos os conceitos de medidas de não convexidade, assim como algumas ilustrações do comportamento das mesmas com a presença de incertezas. Em seguida, apresentamos um exemplo de aplicação destas ideias, que leva ao uso de cortes de Benders reforçados para a formulação conjunta de cada estágio em problemas de otimização estocástica multi-estágio.

A partir daí, desenvolvemos um conjunto de bibliotecas em `Julia` que fossem capazes de ser integradas às bibliotecas `JuMP` e `SDDP.jl` já existentes, para avaliar a performance deste método de cálculo da função de custo futuro. Os resultados dos problemas indicam que, de fato, o uso de cortes de Benders reforçados para a formulação conjunta é capaz de reduzir o *gap* entre o valor calculado e o valor simulado de um problema de otimização multi-estágio, tomando partido da redução do *gap* operada pela presença de incertezas, o que foi ilustrado com os gráficos das funções de custo futuro nos problemas da seção 7.2.

Esta mesma redução do *gap* foi observada para um problema de otimização que usa uma versão reduzida do problema de operação do SIN, com 2 subsistemas interligados. Embora o uso da formulação conjunta necessite um acréscimo do tempo computacional, uma maior disponibilidade de tempo de processamento para as demais estratégias de solução do problema consideradas não leva a melhores resultados. Para a formulação decomposta, um maior número de cortes resulta em uma redução muito menor do *gap*; para a binarização das variáveis de estado para o uso do SDDiP, o cálculo dos cortes Lagrangianos é mais demorado ainda, e também não resulta em uma redução do *gap* se comparado com o uso de cortes de Benders tradicionais.

Em suma, observamos que o uso de cortes para a formulação conjunta permite uma representação convexa das funções de custo futuro, que dá uma aproximação significativamente mais acurada das mesmas nas duas famílias de exemplos que exploramos.

11 Generalizações e problemas em aberto

Vimos, nas seções anteriores, que a estocasticidade em um problema de otimização pode agir no sentido de reduzir *gaps de dualidade*. Isto foi observado graficamente, através dos exemplos apresentados na seção 2, e analiticamente, como demonstrado na seção 3, para medidas concretas de redução deste *gap*, tanto em valores absolutos como em média; e também da redução da parte negativa da segunda derivada, provada nas seções 4.1 e 5.

Estes resultados têm aplicabilidade geral, e em particular para problemas de otimização estocástica, como vimos na seção 6, ao comparar métodos para construção de cortes, e apresentar as necessárias modificações aos algoritmos de programação dinâmica. Indicamos a seguir algumas linhas de pesquisa, em direção a resultados mais precisos, ou mais próximas do contexto de otimização estocástica.

11.1 Resultados quantitativos

A maior parte dos resultados que obtivemos são qualitativos, e não podem garantir que, para uma dada função valor ótimo f e uma variável aleatória Ξ com lei μ, a função de custo futuro resultante $\bar{\mathcal{Q}}(x) = \mathbb{E}\left[f(x + \Xi)\right]$ será convexa. Sabemos que sua não-convexidade não aumenta, pelos teoremas 3.2, 3.6 sobre o *gap*, e suas respectivas generalizações 4.5 e 4.6. Em alguns casos até podemos dizer que, se μ for suficientemente "dispersa", o *gap* será arbitrariamente pequeno, como visto no teorema 3.4. Além disso, obtivemos um resultado de impossibilidade de convexificação, no caso de funções que são mínimo de funções suaves, e distribuições discretas, teorema 5.11.

Em especial devido a este último, seria interessante obter resultados quantitativos mais precisos, em duas direções, que delineamos a seguir. A primeira seria uma garantia da redução do volume do *gap*, em paralelo ao teorema 3.3 sobre a redução do máximo do *gap*. Com efeito, vimos ao longo dos exemplos que é possível que o domínio de não convexidade da função de custo futuro $\bar{\mathcal{Q}}$ seja maior do que o da função valor ótimo f, e a incerteza atuaria apenas "espalhando" a não-convexidade de f. Assim, não é claro que, em média, haverá uma redução do erro de aproximação, que é o *gap* médio. A decomposição de f'' em parte positiva e negativa ("de Hahn-Jordan"), vista na seção 5.3, e também usada indiretamente no teorema 4.2, será provavelmente o ponto de partida para um resultado quantitativo mais preciso, da mesma forma que o domínio de não-convexidade foi para o resultado sobre o máximo do *gap*.

Em segundo lugar, podemos esperar resultados mais precisos quanto aos casos onde há convexificação na passagem de f para $\bar{\mathcal{Q}}$, ao restringir a família de modelos aleatórios. Por exemplo, novamente, o teorema 3.4 se concentra nas distribuições uniformes em um intervalo, e é capaz de garantir a redução do *gap*. Uma forma de reforçar estes resultados poderia ser ao também restringir a família de funções não-convexas de que tratamos. De fato, a teoria desenvolvida não especifica a origem de tais funções mas, para nossa aplicação, as funções f serão funções valor ótimo de problemas de otimização inteiros-mistos. Esta estrutura suplementar pode ser explorada para dar resultados melhores.

11.2 Generalizando a transição de estado

Para a teoria desenvolvida, utilizamos uma forma especial da incerteza, qual seja, aditiva no estado, caracterizado pela equação dinâmica (2.3):

$$Ax_t + By_t = b + T(x_{t-1} + \xi_t).$$

Entretanto, esta forma poderia ser restritiva demais, e não abranger uma gama suficientemente grande de problemas de otimização estocástica. Vamos apresentar dois exemplos disto.

O primeiro caso é quando a incerteza não atua apenas no estado. Por exemplo, a demanda poderia ser incerta, o que seria uma alteração direta em b, sem passar pela mesma transformação T que carrega a dependência de x_{t-1} em x_t. Assim, já não é mais possível escrever $Q_t(x_{t-1}, \xi_t^i) = f(x_t + \xi_t^i)$ e efetuar uma mudança de variáveis. O que pode ser feito, contudo, é usar uma forma parcial: como temos

$$Ax_t + By_t = b_t + Tx_{t-1} + W\xi_t^i,$$

ainda é possível escrever $Q_t(x_{t-1}, \xi_t^i) = g(Tx_{t-1} + W\xi_t^i)$. Observe que, devido à presença da transformação T, a dimensão do domínio de g não tem mais relação com a dimensão de x_{t-1}, e talvez só possamos observar uma *parte* de g ao variar x_{t-1}. A variabilidade em ξ^i ainda pode reduzir a não convexidade da função g, e no caso em que $\mathcal{G}(y) = \mathbb{E}\left[g(y + W\Xi)\right]$ seja convexa em y, isto será suficiente para que $\bar{\mathcal{Q}}(x_{\text{ini}}) = \mathcal{G}(Tx_{\text{ini}})$ seja convexa. Entretanto, no caso em que $\mathcal{G}$ ainda seja não-convexa, não é imediato que a não-convexidade de $\mathcal{G}(Tx_{t-1})$ seja inferior à de $g(Tx_{t-1})$.

O segundo é o caso de problemas onde a incerteza apresenta dependência temporal, em que é comum ter um modelo com um par de equações, uma para o estado original, outra modelando o processo ξ_t:

$$\begin{aligned} Ax_t + By_t &= b + Tx_{t-1} + W\xi_t, \\ \xi_t &= \Phi_1\xi_{t-1} + \Phi_0\eta_t. \end{aligned} \tag{11.1}$$

Agora, a variável aleatória do modelo passa a ser η_t, e o processo ξ_t passa a ser uma variável de estado. Escrevendo a equação acima em blocos, obtemos:

$$\begin{bmatrix} A & 0 \\ 0 & Id \end{bmatrix} \begin{bmatrix} x_t \\ \xi_t \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} y_t = b + \begin{bmatrix} T & W\Phi_1 \\ 0 & \Phi_1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ \xi_{t-1} \end{bmatrix} + \begin{bmatrix} W\Phi_0 \\ \Phi_0 \end{bmatrix} \eta_t.$$

Este modelo ainda se encaixa no modelo geral, onde a dinâmica é regida por duas matrizes independentes, uma para o estado, e outra para a incerteza. Entretanto, como esta estrutura é bastante regular, talvez seja possível obter resultados mais precisos de convexificação da função de custo futuro.

Outra direção de pesquisa futura se refere a modelos com outras formas de incerteza, por exemplo em modelos auto-regressivos multiplicativos. Com a passagem do protótipo para `JuMP-0.19` e a nova versão da biblioteca `SDDP.jl`, resta desenvolver a interface de dados e adaptar o modelo matemático para tratar este tipo de modelos.

11.3 Decomposição parcial de cenários

O custo computacional da formulação conjunta cresce conforme aumenta o número de cenários, o que pode se tornar uma desvantagem séria caso seja necessário efetuar uma discretização mais fina das distribuições de probabilidade do problema. Uma possibilidade de mitigar este efeito seria determinar, *a priori*, subgrupos de ruídos cuja média já tenha uma medida de não-convexidade significativamente menor. Assim, seria possível calcular cortes de Benders reforçados para cada um destes subgrupos, e em seguida fazer a média destes últimos, o que poderia reduzir o tamanho máximo do PL que seria calculado. Em última análise, gostaríamos de determinar o menor conjunto de ruídos a ser considerado para obter a convexificação da função de custo futuro, e se um tal "menor conjunto" de fato existe. Estas questões também levam a investigar a escala dos ruídos face às não-convexidades da função valor ótimo.

Referências

[ANA17] Resolução nº 2081, Dez 2017.

[Fis12] Tom Fischer. Existence, uniqueness, and minimality of the jordan measure decomposition. 2012.

[Fol99] Gerald B. Folland. *Real analysis : Modern Techniques And Their Applications*. Wiley, New York, 1999.

[FPdCLdFCdC18] Bernardo Freitas Paulo da Costa, Iago Leal de Freitas, Filipe Goulart Cabral, and Joari Paulo da Costa. Uso de restrições disjuntivas para modelar políticas operativas. techreport, Convênio ONS – UFRJ, 2018. Restrições Disjuntivas – Relatório Final.

[Hö03] Lars Hörmander. *The Analysis of Linear Partial Differential Operators I*. Springer Berlin Heidelberg, 2003.

[PP91] Mario VF Pereira and Leontina MVG Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, 52(1-3):359–375, 1991.

[RU00] R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[Sch66] Laurent Schwartz. *Théorie des distributions*. Hermann, Paris, 3rd edition, 1966.

[ZAS16] Jikai Zou, Shabbir Ahmed, and Andy Sun. Stochastic dual dynamic integer programming. 2016. preprint Optimization Online, `http://www.optimization-online.org/DB_FILE/2016/05/5436.pdf`.