3
O Método de Programação Dinâmica Dual para Solução de Problemas de Programação Estocástica de Dois Estágios

3.1.
Introdução

Como será visto no capítulo 5, entre as abordagens para otimização de portfólio de contratos de energia propostas nesta tese, duas utilizam o método de programação dinâmica dual como estratégia de solução. Com isto, neste capítulo são apresentados os conceitos e algoritmos relacionados a tal método.

Na seção 3.2 é apresentada a definição de um problema de programação estocástica de dois estágios. Em seguida, na seção 3.3, são apresentados os conceitos relacionados à aplicação do método de programação dinâmica dual na solução de um problema de otimização determinístico de dois estágios. Na seção 3.4 é feita a extensão para o caso estocástico. Finalmente na seção 3.5, são apresentadas duas formas para o tratamento de inviabilidade nos problemas de segundo estágio: penalização na função objetivo e cortes de viabilidade.

3.2.
Modelo de Programação Estocástica de Dois Estágios

Um modelo de programação estocástica [40,41] de dois estágios pode ser interpretado como um processo de decisão sob condições de incertezas onde existem dois conjuntos de variáveis: aquelas que devem ser decididas antes da realização das incertezas, denominadas variáveis de primeiro estágio, e aquelas cujas decisões podem ser tomadas após a realização das incertezas, denominadas variáveis de segundo estágio.

Assumindo que as incertezas sejam representadas de forma discreta através de S cenários, e que a probabilidade de ocorrência do s-ésimo cenário seja dada por p_s ($p_s \geq 0$,

$\sum_{s=1}^{S} p_s = 1$), um problema de programação linear estocástica de dois estágios pode ser escrito como:

$$
\begin{aligned}
&\underset{x_1, x_{2s}}{\text{Maximizar}} \quad c_1^T x_1 + \sum_{s=1}^{S} p_s c_{2s}^T x_{2s} \\
&\text{s.a.} \\
&\quad A_1 x_1 \le b_1 \\
&\quad E_1 x_1 + A_{2s} x_{2s} \le b_{2s} \qquad s = 1,\dots,S \\
&\quad x_1 \ge 0 \\
&\quad x_{2s} \ge 0 \qquad s = 1,\dots,S
\end{aligned}
\tag{3.1}
$$

Em (3.1), o vetor x_1 representa as variáveis de decisão de primeiro estágio, e o vetor x_{2s} representa as variáveis de decisão de segundo estágio associadas ao s-ésimo cenário. Enquanto que as variáveis de decisão de primeiro estágio são determinísticas, pois devem ser decididas antes da realização das incertezas, as de segundo estágio variam com o cenário. Note que as variáveis de decisão de ambos os estágios se relacionam através de restrições lineares, e que os vetores e matrizes associados às variáveis de decisão de segundo estágio (c_{2s}, b_{2s} e A_{2s}) podem, no caso geral, variar de cenário para cenário. Note também que não existe nenhuma condição que relacione as decisões de segundo estágio associadas a cenários diferentes, isto é, que relacionam x_{2i} e x_{2j} para $i \neq j$.

3.3.
O Método de Programação Dinâmica Dual para Problemas de Otimização Determinísticos de Dois Estágios

Inicialmente, o método de programação dinâmica dual [42] será apresentado para o caso onde o problema de otimização a ser resolvido é determinístico, para então na próxima seção ser estendido para o caso estocástico. Considere então o seguinte problema de programação linear:

$$
\begin{aligned}
&\underset{x_1, x_2}{\text{Maximizar}} \quad c_1^T x_1 + c_2^T x_2 \\
&\text{s.a.} \\
&\qquad A_1 x_1 \le b_1 \\
&\qquad E_1 x_1 + A_2 x_2 \le b_2 \\
&\qquad x_1, x_2 \ge 0
\end{aligned}
\tag{3.2}
$$

O problema (3.2) pode ser interpretado como um processo de decisão em dois estágios. No primeiro estágio define-se o vetor x_1 que satisfaça $A_1 x_1 \le b_1$ e $x_1 \ge 0$, e dado x_1, define-se o vetor x_2 que satisfaça $A_2 x_2 \le b_2 - E_1 x_1$ e $x_2 \ge 0$. Note que quando da definição de x_2, x_1 já é conhecido, e por isso aparece no lado direito da restrição.

O objetivo é encontrar uma solução que maximize a soma das funções objetivo de primeiro e segundo estágios, conforme mostrado na Figura 3.1.

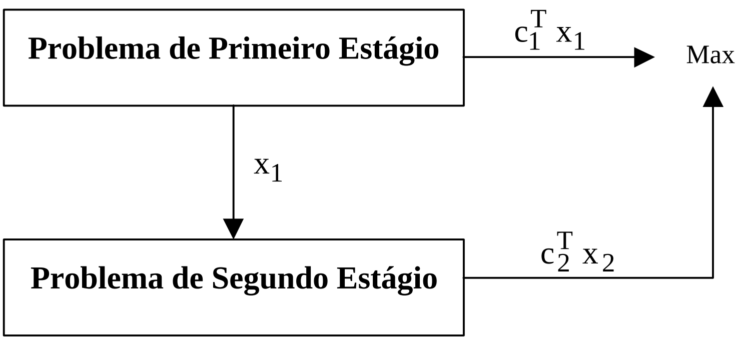

Figura 3.1 – Processo de Decisão em Dois Estágios

O método de programação dinâmica dual pode ser utilizado para resolver problemas de decisão em dois estágios, tal como (3.2). Definindo-se formalmente o problema de primeiro estágio:

$$
\begin{aligned}
&\underset{x_1}{\text{Maximizar}} \quad c_1^T x_1 + \upsilon(x_1) \\
&\text{s.a.} \\
&\qquad A_1 x_1 \le b_1 \\
&\qquad x_1 \ge 0
\end{aligned}
\tag{3.3}
$$

Em (3.3), o termo $\upsilon(x_1)$ é uma função que depende da decisão de primeiro estágio x_1. Tal função é dada pelo problema de otimização abaixo, formalmente denominado de problema de segundo estágio:

$$\begin{aligned} \upsilon(x_1) = \underset{x_2}{\text{Maximizar}}\ & c_2^T x_2 \\ \text{s.a.}\ & \\ & A_2 x_2 \leq b_2 - E_1 x_1 \\ & x_2 \geq 0 \end{aligned} \tag{3.4}$$

A função $\upsilon(x_1)$ fornece o valor ótimo do problema de segundo estágio em função da decisão de primeiro estágio. Se a representação explícita da função $\upsilon(x_1)$ for conhecida, tal representação poderia ser utilizada diretamente em (3.3), resolvendo-se assim o problema de dois estágios como um problema de apenas um estágio. Entretanto, a princípio a representação explícita de tal função não é conhecida. Como será mostrado a seguir, o método de programação dinâmica dual constrói iterativamente uma aproximação linear por partes para representar $\upsilon(x_1)$ em (3.3).

Sabe-se que para todo problema de programação linear existe um problema dual associado. O valor ótimo do problema original (denominado problema primal) e do problema dual associado coincidem. Com isso, o problema de segundo estágio também pode ser representado pelo dual de (3.4):

$$\begin{aligned} \upsilon(x_1) = \underset{\eta}{\text{Minimizar}}\ & \eta^T (b_2 - E_1 x_1) \\ \text{s.a.}\ & \\ & A_2^T \eta \geq c_2 \\ & \eta \geq 0 \end{aligned} \tag{3.5}$$

onde η é o vetor das variáveis duais associadas às restrições do problema (3.4).

Como os valores das funções objetivo de (3.4) e de (3.5) coincidem em suas soluções ótimas, ambos podem ser utilizados para representar a função $\upsilon(x_1)$. Entretanto, a região viável do primal depende de x_1, enquanto que a região viável do dual não depende. Sendo assim, o conjunto de soluções possíveis de (3.5), que correspondem aos vértices da região viável do dual (dada pelas restrições $A_2^T \eta \geq c_2$, $\eta \geq 0$), podem ser definidos independentemente do valor de x_1.

Seja $N = \{\eta^1, \eta^2, ..., \eta^v\}$ o conjunto de vetores que representam os vértices da região viável do dual. Como a solução ótima de (3.5) está em um destes vértices, o problema (3.5) poderia a princípio ser resolvido por enumeração:

$$\upsilon(x_1) = \underset{\eta}{\text{Minimizar}} \ \{\eta^{i^T} (b_2 - E_1 x_1) \quad i = 1,2,..., v\} \tag{3.6}$$

Entretanto, o problema (3.6) pode ser rescrito como um problema de programação linear da seguinte forma:

$$\begin{aligned} \upsilon(x_1) = \underset{\upsilon}{\text{Maximizar}} \quad & \upsilon \\ \text{s.a.} \quad & \\ & \upsilon \leq \eta^{1^T} (b_2 - E_1 x_1) \\ & \upsilon \leq \eta^{2^T} (b_2 - E_1 x_1) \\ & .... \\ & \upsilon \leq \eta^{v^T} (b_2 - E_1 x_1) \end{aligned} \tag{3.7}$$

A formulação apresentada em (3.7) possui uma interpretação geométrica interessante. Ela indica que $\upsilon(x_1)$ é uma função linear por partes, onde cada hiperplano $\eta^{i^T} (b_2 - E_1 x_1)$ representa uma parte de $\upsilon(x_1)$.

Com a representação de $\upsilon(x_1)$ por uma função linear por partes, esta é caracterizada pelos coeficientes η^i dos hiperplanos (vértices do dual). Obviamente a determinação de todos os vértices η^i da região viável do dual pode ser uma tarefa extremamente árdua. Entretanto, a metodologia de programação dinâmica dual não determina todos estes vértices. Ela determina apenas um subconjunto deles, construindo uma aproximação para a função $\upsilon(x_1)$.

Considere novamente o problema de segundo estágio em sua formulação primal:

$$
\begin{aligned}
\upsilon(\hat{x}_1^i) = \underset{x_2}{\text{Maximizar}} \quad & c_2^T x_2 \\
\text{s.a.} \quad & \\
& A_2 x_2 \le b_2 - E_1 \hat{x}_1^i \qquad (\eta^i) \\
& x_2 \ge 0
\end{aligned}
\tag{3.8}
$$

Suponha que o problema acima tenha sido resolvido para um dado vetor $x_1 = \hat{x}_1^i$. Juntamente com sua solução obtém-se um vetor η^i, cujas componentes representam as variáveis duais associadas às restrições de (3.8). Da teoria de programação linear sabe-se que tal vetor η^i representa um dos vértices do conjunto N, podendo ser utilizado na construção de um dos hiperplanos que formam $\upsilon(x_1)$.

Computacionalmente, o problema (3.8) poderia ser resolvido para um conjunto de n $(n < v)$ vetores x_1 ($\hat{x}_1^1, \hat{x}_1^2, ..., \hat{x}_1^n$), obtendo-se assim n vetores de variáveis duais, utilizadas na construção de uma aproximação para $\upsilon(x_1)$.

$$
\begin{array}{ll}
\hat{x}_1^1 \rightarrow \eta^1 & \hat{\upsilon}(x_1) = \underset{\upsilon}{\text{Maximizar}} \ \upsilon \\
\hat{x}_1^2 \rightarrow \eta^2 & \quad \text{s.a.} \\
\hat{x}_1^3 \rightarrow \eta^3 & \qquad \upsilon \le {\eta^1}^T (b_2 - E_1 x_1) \\
\ldots\ldots & \qquad \upsilon \le {\eta^2}^T (b_2 - E_1 x_1) \\
& \qquad \ldots\ldots\ldots \\
\hat{x}_1^n \rightarrow \eta^n & \qquad \upsilon \le {\eta^n}^T (b_2 - E_1 x_1)
\end{array}
\tag{3.9}
$$

Note que a função aproximada $\hat{\upsilon}(x_1)$ é um *limite superior* da função real $\upsilon_1(x_1)$, pois o problema (3.9) contém apenas um subconjunto das restrições do problema (3.7).

A função $\hat{\upsilon}(x_1)$ pode ser utilizada para representar, de forma aproximada, a função $\upsilon(x_1)$ no problema de primeiro estágio. Substituindo $\upsilon(x_1)$ por $\hat{\upsilon}(x_1)$ em (3.3) obtém-se:

$$\begin{aligned}
\bar{z} = \underset{x_1,\upsilon}{\text{Maximizar}} \quad & c_1^T x_1 + \upsilon \\
\text{s.a.} \quad & \\
& A_1 x_1 \le b_1 \\
& \upsilon \le \eta^{i^T} (b_2 - E_1 x_1) \qquad i = 1,\ldots,n \\
& x_1 \ge 0
\end{aligned} \tag{3.10}$$

O problema (3.10) é uma relaxação do problema (3.3)[1], logo, $\bar{z}$, o valor ótimo de (3.10), é um *limite superior* para o valor ótimo do problema original.

Um *limite inferior* para o valor ótimo do problema original pode ser obtido a partir da solução do problema de segundo estágio, dado pela formulação (3.4), para o vetor x_1 obtido por (3.10):

$$\begin{aligned}
\underline{z} = c_1^T \hat{x}_1 + \underset{x_2}{\text{Maximizar}} \quad & c_2^T x_2 \\
\text{s.a.} \quad & \\
& A_2 x_2 \le b_2 - E_1 \hat{x}_1 \\
& x_2 \ge 0
\end{aligned} \tag{3.11}$$

Note que em (3.11) o problema de segundo estágio é resolvido utilizando-se todas as suas restrições (via formulação primal), logo pode-se garantir a viabilidade[2] do par x_1, x_2 no problema original. Entretanto, como não se pode garantir a otimalidade de x_1, também não se pode garantir a otimalidade de x_2. Por isso $\underline{z}$ é um *limite inferior* para o valor ótimo do problema original.

A diferença entre o *limite superior* obtido por (3.10) e o *limite inferior* obtido por (3.11), $\bar{z} - \underline{z}$, pode ser utilizado para verificar a precisão com que a função $\upsilon(x_1)$ é

---

[1] Pois $\hat{\upsilon}(x_1)$, dada por (3.9), é uma relaxação para $\upsilon(x_1)$, dada por (3.7).

[2] Nesta seção assume-se, por simplicidade, que o problema de segundo estágio é viável para qualquer valor de x_1 obtido no problema de primeiro estágio. O tratamento de possíveis inviabilidades será objeto de discussão na seção 3.5 deste capítulo.

aproximada por $\hat{\upsilon}(x_1)$. Se esta diferença for menor que uma dada tolerância, o problema estará resolvido. Caso contrário, um novo conjunto de vetores x_1 deve ser utilizado para se determinar outros vértices η^i da região viável do problema (dual) de segundo estágio, que serão utilizados na construção de novos hiperplanos a serem incorporados à função $\hat{\upsilon}(x_1)$, melhorando assim a aproximação de $\upsilon(x_1)$.

A seguir é apresentado um algoritmo simplificado que descreve os passos até aqui descritos:

<u>Algoritmo I</u>

1. Faça k = 1
2. Selecione n vetores x_1 ($\hat{x}_1^i$, i = 1,...,n)
3. Para cada $\hat{x}_1^i$ resolva o problema abaixo:

$$\begin{aligned} \upsilon(\hat{x}_1^i) = \underset{x_2}{\text{Maximizar}}\ & c_2^T x_2 \\ \text{s.a.}\ & \\ & A_2 x_2 \le b_2 - E_1 \hat{x}_1^i \\ & x_2 \ge 0 \end{aligned}$$

4. Utilize as variáveis duais associadas às restrições dos problemas resolvidos no passo (3) para atualizar a aproximação de $\upsilon(x_1)$, e resolva o problema abaixo (*limite superior* do problema original):

$$\begin{aligned} \bar{z} = \underset{x_1,\upsilon}{\text{Maximizar}}\ & c_1^T x_1 + \upsilon \\ \text{s.a.}\ & \\ & A_1 x_1 \le b_1 \\ & \upsilon \le \eta^{i^T} (b_2 - E_1 x_1) \qquad i = 1,\dots,k \cdot n \\ & x_1 \ge 0 \end{aligned}$$

5. Utilize o vetor $\hat{x}_1$ obtido no passo (4) para calcular o *limite inferior* do problema original:

$$\underline{z} = c_1^T \hat{x}_1 + \underset{x_2}{\text{Maximizar}} \ c_2^T x_2$$
$$\text{s.a.}$$
$$A_2 x_2 \leq b_2 - E_1 \hat{x}_1$$
$$x_2 \geq 0$$

6. Se $\bar{z} - \underline{z} \leq \varepsilon$ pare, caso contrário faça k = k + 1 e volte ao passo (2)

Um ponto importante a ser definido diz respeito a escolha dos vetores x_1 no passo (2) do Algoritmo I. Uma escolha inteligente seria utilizar o vetor x_1 obtido no passo (4) da iteração anterior, pois a aproximação de $\upsilon(x_1)$ estará sendo feita nas vizinhanças de pontos candidatos a solução ótima. Desta forma, em cada iteração adiciona-se apenas um hiperplano para melhorar a aproximação de $\upsilon(x_1)$.

Com o procedimento acima, o algoritmo do método de programação dinâmica dual fica:

<u>Algoritmo II</u>

1. Faça n = 0
2. Resolva o problema abaixo (*limite superior* do problema original):

$$\bar{z} = \underset{x_1, \upsilon}{\text{Maximizar}} \ c_1^T x_1 + \upsilon$$
$$\text{s.a.}$$
$$A_1 x_1 \leq b_1$$
$$\upsilon \leq \eta^{i^T} (b_2 - E_1 x_1) \qquad i = 1, \ldots, n$$
$$x_1 \geq 0$$

3. Resolva o problema abaixo em função do vetor $x_1 = \hat{x}_1$ obtido no passo (2):

$$\upsilon(\hat{x}_1) = \underset{x_2}{\text{Maximizar}} \ c_2^T x_2$$
$$\text{s.a.}$$
$$A_2 x_2 \leq b_2 - E_1 \hat{x}_1$$
$$x_2 \geq 0$$

4. Faça $\underline{z} = c_1^T \hat{x}_1 + \upsilon(\hat{x}_1)$
5. Faça n = n + 1. Utilize o vetor de variáveis duais associado às restrições do problema resolvido no passo (3) para atualizar a aproximação de $\upsilon(x_1)$
6. Se $\bar{z} - \underline{z} \leq \varepsilon$ pare, caso contrário volte ao passo (2)

Vale ressaltar que o Algoritmo II acima pode ser visto como uma decomposição de Benders [43] onde tanto o problema de primeiro estágio quanto o problema de segundo estágio são de programação linear.

3.4. Extensão do Algoritmo para Problemas Estocásticos de Dois Estágios

Seja o problema (3.1) escrito em sua forma estendida:

$$\begin{aligned}
&\underset{x_1, x_{2s}}{\text{Maximizar}} \quad c_1^T x_1 + p_1 c_{21}^T x_{21} + p_2 c_{22}^T x_{22} + ... + p_S c_{2S}^T x_{2S} \\
&\text{s.a.} \\
&\quad A_1 x_1 \leq b_1 \\
&\quad E_1 x_1 + A_{21} x_{21} \leq b_{21} \\
&\quad E_1 x_1 + A_{22} x_{22} \leq b_{22} \\
&\quad \cdots\cdots\cdots\cdots\cdots \\
&\quad E_1 x_1 + A_{2S} x_{2S} \leq b_{2S} \\
&\quad x_1, x_{21}, ..., x_{2S} \geq 0
\end{aligned} \qquad (3.12)$$

O problema (3.12) pode ser decomposto em $S+1$ problemas: um problema de primeiro estágio e S problemas de segundo estágio, um associado a cada cenário s. No problema de primeiro estágio decide-se x_1, e dada esta decisão, decide-se x_{2s} através da solução do problema de segundo estágio associado ao s-ésimo cenário. Formalmente:

<u>Problema de Primeiro Estágio</u>

$$
\begin{aligned}
&\underset{x_1}{\text{Maximizar}} \quad c_1^T x_1 + \sum_{s=1}^{S} p_s \upsilon_s(x_1) \\
&\text{s.a.} \\
&\qquad A_1 x_1 \le b_1 \\
&\qquad x_1 \ge 0
\end{aligned}
\tag{3.13}
$$

<u>Problema de Segundo Estágio Associado ao Cenário s (s = 1,…,S)</u>

$$
\begin{aligned}
&\upsilon_s(x_1) = \underset{x_{2s}}{\text{Maximizar}} \quad c_{2s}^T x_{2s} \\
&\qquad \text{s.a.} \\
&\qquad\quad A_{2s} x_{2s} \le b_{2s} - E_1 x_1 \\
&\qquad\quad x_{2s} \ge 0
\end{aligned}
\tag{3.14}
$$

O objetivo é obter uma solução que maximize a soma do valor ótimo de (3.13) com o valor esperado dos valores ótimos de (3.14), conforme ilustrado na Figura 3.2.

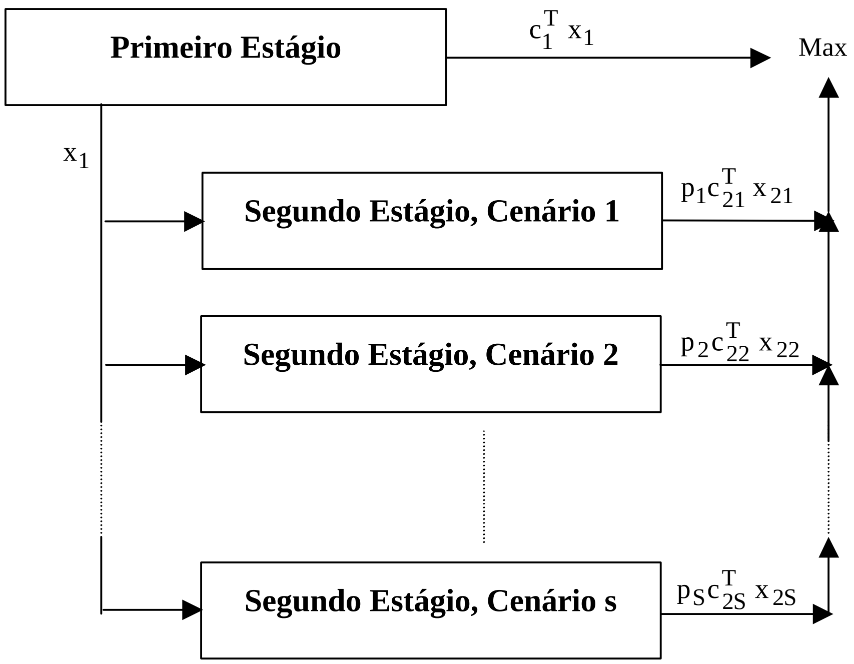

Figura 3.2 – Processo de Decisão em Dois Estágios (Caso Estocástico)

De forma similar ao caso determinístico, para o caso estocástico o método de programação dinâmica dual constrói iterativamente uma aproximação linear por partes para cada função $\upsilon_s(x_1)$, s = 1,…,S, e utiliza estas aproximações em (3.13). A cada iteração S novos hiperplanos são acrescentados em (3.13), um associado a cada função $\upsilon_s(x_1)$, de modo a aprimorar a aproximação destas funções.

O algoritmo para solução de problemas de programação estocástica de dois estágios via método de programação dinâmica dual [42,44] fica[1]:

<u>Algoritmo III</u>

1. Faça n = 0
2. Resolva o problema abaixo (*limite superior* do problema original)

$$\bar{z} = \underset{x_1, \upsilon_s}{\text{Maximizar}} \quad c_1^T x_1 + \sum_{s=1}^{S} p_s \upsilon_s$$
$$\text{s.a.}$$
$$A_1 x_1 \le b_1$$
$$\upsilon_s \le \eta_s^{i\,T} (b_{2s} - E_1 x_1) \qquad i = 1,\dots,n \quad s = 1,\dots,S$$
$$x_1 \ge 0$$

3. Resolva os problemas abaixo em função do vetor $x_1 = \hat{x}_1$ obtido no passo (2):

   Para s = 1,….,S resolva

$$\upsilon_s(\hat{x}_1) = \underset{x_{2s}}{\text{Maximizar}} \; c_{2s}^T x_{2s}$$
$$\text{s.a.}$$
$$A_{2s} x_{2s} \le b_{2s} - E_1 \hat{x}_1$$
$$x_{2s} \ge 0$$

4. Faça $\underline{z} = c_1^T \hat{x}_1 + \sum_{s=1}^{S} p_s \upsilon_s(\hat{x}_1)$
5. Faça n = n + 1. Utilize os vetores de variáveis duais associados às restrições dos problemas resolvidos no passo (3) para atualizar as aproximações de $\upsilon_s(x_1)$, s = 1,…,S

---

[1] Em [44], o método de programação dinâmica dual aparece com o nome de planos cortantes, denominação conhecida na área matemática.

6. Se $\bar{z} - \underline{z} \leq \varepsilon$ pare, caso contrário volte ao passo (2)

Uma implementação alternativa do Algoritmo III consiste na modificação do problema de otimização do passo (2) de modo a se trabalhar com uma função $\bar{\upsilon}(x_1)$ que forneça diretamente o valor esperado dos valores ótimos dos problemas de segundo estágio, isto é:

$$\bar{\upsilon}(x_1) = \sum_{s=1}^{S} p_s \, \upsilon_s(x_1)$$

Trabalhando-se desta forma, em cada iteração atualiza-se as aproximações das funções $\upsilon_s(x_1)$, s = 1,…,S, porém adiciona-se no problema de primeiro estágio apenas um novo hiperplano, dado por:

$$\bar{\upsilon} \leq \sum_{s=1}^{S} p_s \, \eta_s^{i^T} \, b_{2s} - \sum_{s=1}^{S} p_s \, \eta_s^{i^T} \, E_1 \, x_1$$

Com isto, o problema a ser resolvido no passo (2) do Algoritmo III fica:

$$\begin{aligned} \bar{z} = \underset{x_1, \bar{\upsilon}}{\text{Maximizar}} \quad & c_1^T x_1 + \bar{\upsilon} \\ \text{s.a.} \quad & \\ & A_1 x_1 \leq b_1 \\ & \bar{\upsilon} \leq \sum_{s=1}^{S} p_s \, \eta_s^{i^T} \, b_{2s} - \sum_{s=1}^{S} p_s \, \eta_s^{i^T} \, E_1 \, x_1 \qquad i = 1, \ldots, n \\ & x_1 \geq 0 \end{aligned}$$

A representação da função $\bar{\upsilon}(x_1)$ no problema de primeiro estágio faz com que a dimensão deste problema não cresça tão rapidamente, pois em cada iteração apenas um novo hiperplano é acrescentado a ele. Entretanto, a representação explícita de todas as funções $\upsilon_s(x_1)$, s = 1,…,S no problema de primeiro estágio é considerada mais eficiente[1], pois

[1] E esta será a abordagem adotada nos algoritmos implementados neste trabalho de tese.

permite a obtenção do valor esperado através da combinação de hiperplanos definidos em iterações diferentes [44].

3.5. Tratamento de Inviabilidade nos Problemas de Segundo Estágio

Durante a execução do passo (3) do Algoritmo III pode não existir um vetor $x_{2s} \geq 0$ que satisfaça a restrição $A_{2s} x_{2s} \leq b_{2s} - E_1 \hat{x}_1$. Em outras palavras, pode ser detectada inviabilidade do problema de segundo estágio associado a um ou mais cenários. Quando isto ocorre, não é possível construir um novo hiperplano para atualizar a aproximação da função $\upsilon_s(x_1)$ correspondente. Entretanto, é importante fornecer algum tipo de informação para o problema de primeiro estágio de que o vetor x_1 nele definido não representa uma boa escolha, pois resulta na inviabilidade do problema como um todo[1].

Nesta tese são consideradas duas formas para o tratamento de inviabilidade nos problemas de segundo estágio: penalização na função objetivo, descrita na seção 3.5.1, e corte de viabilidade, descrita na seção 3.5.2.

3.5.1. Penalização na Função Objetivo

O tratamento de inviabilidade nos problemas de segundo estágio via penalização na função objetivo consiste na adição de variáveis artificiais nas restrições onde estas possam ocorrer. Tais variáveis são incluídas também na função objetivo, multiplicadas por um coeficiente muito grande ($Pen \rightarrow +\infty$), de modo a haver uma penalização no valor assumido pela função objetivo caso estas variáveis assumam valor não nulo.

Com tal tratamento, os problemas a serem resolvidos no passo (3) do Algoritmo III tornam-se:

---

[1] Ou seja, tal vetor $\hat{x}_1$ não é viável para o problema (3.12).

$$\upsilon_s(\hat{x}_1) = \underset{x_{2s},\theta_s}{\text{Maximizar}}\ c_{2s}^T x_{2s} - m^T \theta_s$$
$$\text{s.a.}$$
$$A_{2s} x_{2s} - \theta_s \leq b_{2s} - E_1 \hat{x}_1 \qquad (3.15)$$
$$x_{2s} \geq 0$$
$$\theta_s \geq 0$$

onde $m^T = [Pen, ..., Pen]$ com $Pen \to +\infty$.

Note que sempre que θ_s assume valor não nulo, a função objetivo de (3.15) sofre uma grande redução. Porém, (3.15) nunca é inviável, independente do valor de $\hat{x}_1$ por ele recebido. Sendo assim, sempre é possível construir um novo hiperplano que melhore a aproximação de $\upsilon_s(x_1)$ no problema de primeiro estágio.

Vale ressaltar que sempre que a penalização na função objetivo de (3.15) é ativada, tal penalização é refletida no novo hiperplano a ser adicionado no problema de primeiro estágio. Com isso, o tratamento via penalização na função objetivo não informa diretamente para o problema de primeiro estágio que o vetor $\hat{x}_1$ por ele escolhido conduz a inviabilidade do problema, e sim indica que tal vetor $\hat{x}_1$ causa uma enorme redução no valor da função objetivo (a qual deseja-se maximizar). Desta forma, o algoritmo buscará uma solução na qual o valor assumido por θ_s seja o menor possível. Se com a convergência do algoritmo tal variável assumir valor não nulo, provavelmente o problema (3.12) é inviável.

Um cuidado a ser tomado quando trabalha-se com penalização na função objetivo está na escolha do coeficiente muito grande ($Pen \to \infty$). Se este coeficiente for demasiadamente grande, pode haver instabilidade no processo de solução numérica do problema de otimização, resultando na não convergência do processo[1,2].

---

[1] Tal fato foi verificado nas simulações efetuadas com os modelos propostos nesta tese.

[2] Vale ressaltar que em alguns casos utiliza-se variáveis naturais do problema para penalizar a função objetivo. Por exemplo, nos modelos de planejamento da operação energética adotados no Brasil utiliza-se a variável de déficit de energia na penalização da função objetivo. Tal modelagem tem sido bastante satisfatória, não ocasionando instabilidades numéricas no processo de otimização.

3.5.2. Corte de Viabilidade

O tratamento via corte de viabilidade consiste em gerar uma restrição a ser adicionada ao problema de primeiro estágio, que elimine de sua região viável o ponto $x_1 = \hat{x}_1$ que resultou na inviabilidade de um ou mais problemas de segundo estágio. Tal restrição é denominada corte de viabilidade.

O corte de viabilidade é construído a partir da solução de um problema de otimização auxiliar, cujo objetivo é minimizar o somatório das inviabilidades do problema no qual estas foram detectadas.

Considere então o problema de otimização associado ao s-ésimo cenário a ser resolvido no passo (3) do Algoritmo III, cuja formulação é reproduzida a seguir:

$$\begin{aligned} \upsilon_s(\hat{x}_1) = \underset{x_{2s}}{\text{Maximizar}} \ & c_{2s}^T x_{2s} \\ \text{s.a.} \ & \\ & A_{2s} x_{2s} \leq b_{2s} - E_1 \hat{x}_1 \\ & x_{2s} \geq 0 \end{aligned} \tag{3.16}$$

Suponha que para o vetor $x_1 = \hat{x}_1$ em questão, o problema acima não possua solução viável. O problema auxiliar que minimiza o somatório das inviabilidades de (3.16) é dado por:

$$\begin{aligned} U_s(\hat{x}_1) = \underset{\theta}{\text{Minimizar}} \ & e^T \theta \\ \text{s.a.} \ & \\ & A_{2s} x_{2s} - \theta \leq b_{2s} - E_1 \hat{x}_1 \\ & x_{2s} \geq 0 \\ & \theta \geq 0 \end{aligned} \tag{3.17}$$

onde $e^T = [1,....,1]$.

Seja $U_s(x_1)$ uma função que forneça o valor ótimo de (3.17) em função do valor assumido por x_1. De forma similar ao apresentado para a função $\upsilon_s(x_1)$, $U_s(x_1)$ também pode ser representada como uma função linear por partes. Um dos hiperplanos que compõem tal representação pode ser obtido utilizando-se as variáveis duais associadas às restrições de (3.17), isto é[1]:

$$U_s \geq \eta_s^{i^T}(b_{2s} - E_1 x_1) \tag{3.18}$$

Note entretanto que os vetores x_1 para os quais (3.16) é viável necessariamente satisfazem $U_s(x_1) \leq 0$. Logo, para eliminar o ponto $x_1 = \hat{x}_1$ que resultou na inviabilidade de (3.16), basta adicionar ao problema de primeiro estágio uma restrição que é uma variante de (3.18). Tal restrição é chamada de corte de viabilidade, e é dada por:

$$\eta_s^{i^T}(b_{2s} - E_1 x_1) \leq 0$$

O tratamento via corte de viabilidade elimina da região viável do problema de primeiro estágio aquelas soluções que resultaram em inviabilidade em um ou mais problemas de segundo estágio. Quando a adição de um corte de viabilidade no problema de primeiro estágio resulta na sua inviabilidade, pode-se concluir que o problema original (3.12) é inviável.

3.6. Conclusões

Neste capítulo foram apresentados os conceitos e algoritmos de programação dinâmica dual, utilizado como método de solução de duas das três abordagens para otimização de portfólio de contratos de energia propostas nesta tese. As principais conclusões do capítulo são:

- O problema de segundo estágio associado a cada cenário, quando escrito via formulação dual, pode ser interpretado como uma função linear por partes.

[1] Tal hiperplano possui sinal de maior ou igual, pois ao contrário da dedução apresentada anteriormente, o problema (3.17) é de minimização.

- O algoritmo de programação dinâmica dual pode ser visto como uma decomposição de Benders onde os problemas de primeiro e segundo estágio são de programação linear.

- O valor esperado dos valores ótimos dos problemas de segundo estágio no problema de primeiro estágio pode ser representado de duas formas: através de uma função que forneça diretamente este valor esperado, ou através do somatório das funções valor ótimo dos problemas de segundo estágio associados a todos os cenários, ponderados pelas respectivas probabilidades. A primeira opção tem a vantagem de fazer com que a dimensão do problema de primeiro estágio não cresça tão rapidamente, entretanto a segunda opção é considerada mais eficiente, pois permite a obtenção do valor esperado através da combinação de hiperplanos definidos em iterações diferentes.

- Nesta tese considerou-se duas formas para o tratamento de inviabilidades nos problemas de segundo estágio: através de penalização na função objetivo e através de cortes de viabilidade. O tratamento via penalização na função objetivo não informa diretamente para o problema de primeiro estágio que a solução por ele escolhida conduz a inviabilidade do problema, e sim indica que tal solução causa uma enorme redução no valor da função objetivo a qual deseja-se maximizar. Já o tratamento via corte de viabilidade informa explicitamente para o problema de primeiro estágio que a solução por ele escolhida resulta na inviabilidade do problema, e a elimina da região viável do problema de primeiro estágio.

- O tratamento de inviabilidades nos problemas de segundo estágio via penalização na função objetivo, dependendo do peso associado a tal penalização, pode resultar em instabilidade no processo de solução numérica do problema de otimização.