



UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Engenharia Elétrica e de Computação

Rodrigo Barbosa de Lima

**Abordagens alternativas por programação  
dinâmica para o planejamento da operação de  
sistemas hidroelétricos**

**Alternative dynamic programming approaches  
for hydroelectric systems operation planning**

Campinas  
2019, agosto



UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Engenharia Elétrica e de Computação

Rodrigo Barbosa de Lima

## **Abordagens alternativas por programação dinâmica para o planejamento da operação de sistemas hidroelétricos**

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na área de Energia Elétrica (AF).

Orientador: Prof. Christiano Lyra Filho

Este exemplar corresponde à versão final da dissertação defendida pelo aluno Rodrigo Barbosa de Lima e orientada pelo Prof. Christiano Lyra Filho.

Campinas  
2019, agosto

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Luciana Pietrosanto Milla - CRB 8/8129

L628a Lima, Rodrigo Barbosa de, 1992-  
Abordagens alternativas por programação dinâmica para o planejamento de sistemas hidroelétricos / Rodrigo Barbosa de Lima. – Campinas, SP : [s.n.], 2019.

Orientador: Christiano Lyra Filho.  
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Usinas hidrelétricas. 2. Política energética. 3. Processo estocástico. 4. Programação dinâmica. 5. Redes Neurais (Computação). I. Lyra Filho, Christiano, 1951-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Alternative dynamic programming approaches for hydroelectric systems operation planning

**Palavras-chave em inglês:**

Hydroelectric power plants

Energy policy

Stochastic process

Dynamic programming

Neural networks (Computer science)

**Área de concentração:** Energia Elétrica

**Titulação:** Mestre em Engenharia Elétrica

**Banca examinadora:**

Christiano Lyra Filho [Orientador]

Leonardo Tomazeli Duarte

José Federico Vizcaino González

**Data de defesa:** 30-08-2019

**Programa de Pós-Graduação:** Engenharia Elétrica

**Identificação e informações acadêmicas do(a) aluno(a)**

- ORCID do autor: <https://orcid.org/0000-0002-2273-828X>

- Currículo Lattes do autor: <http://lattes.cnpq.br/6902109323055085>

## **Comissão Julgadora - Dissertação de Mestrado**

**Candidato:** Rodrigo Barbosa de Lima (RA:186719)

**Data da Defesa:** 30 de agosto de 2019

**Título da Dissertação:** Abordagens alternativas por Programação Dinâmica para o Planejamento da Operação de Sistemas Hidroelétricos

Prof. Dr. Christiano Lyra Filho (Presidente, FEEC/UNICAMP)

Prof. Dr. Leonardo Tomazeli Duarte (FCA/UNICAMP)

Prof. Dr. José Federico Vizcaino González (FEG/UNESP - Guaratinguetá)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

# Agradecimentos

Agradeço primeiramente a Deus, por me manter forte e iluminar minha jornada, em especial nos últimos anos. Por ser desde sempre a morada de minha mente e espírito, de onde fui capaz de buscar tranquilidade e sabedoria para manter-me de pé diante de todas as adversidades.

Aos meus amados pais Lucia e Nilson, por terem me dado todo o amor, coragem e me ensinado a nunca desistir de meus sonhos. Agradeço aos dois, que mesmo com tantas dificuldades, deram seu máximo para que eu pudesse voar alto e alcançasse lugares que antes pareciam impossíveis. Agradeço por toda educação que tive e pelo privilégio de chegar a defesa de um título de mestrado.

À minha irmã Sabrina, que mesmo a distância, não deixou de acreditar em mim, de trocar conselhos e por nunca ter me deixado esquecer do caminho que percorri até aqui.

À minha amada companheira Camila, que há muito tempo tem sido meu porto seguro. Agradeço por sua paciência, mesmo quando tantas vezes deixei algumas tarefas ou compromissos de lado para que algumas dessas páginas pudessem ser preenchidas. Agradeço por seu total amor, cuidado e dedicação.

Ao prof. Christiano, o primeiro amigo que fiz quando cheguei a Campinas, o qual agradeço imensamente por ter aberto as portas do laboratório e acreditado no meu potencial. Agradeço por ter me oferecido todas as condições para crescer; por me manter sempre na direção correta e motivado. Sou grato desde os momentos de seriedade e orientação até os de descontração, sempre divertidos e muito ricos.

Aos meus queridos amigos do LABORE, os quais desejo todo sucesso. Agradeço em especial aqueles que são minhas referências de competência, dedicação e inteligência: Caio, Petra e Ellen; pela amizade, pelo ambiente favorável e estimulante de trabalho e por me ajudarem de forma ilimitada. Aos meus amigos Alan Caio, Amanda, Patrick e Isadora, que apesar da breve convivência só guardo bons momentos, os quais sou grato.

À FEEC/UNICAMP por ter sido minha segunda casa e como uma boa anfitriã ter me oferecido contato com o que há de melhor na ciência do país em termos de estrutura e conteúdo.

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil.

*“Put two ships in the open sea, without wind or tide,  
and, at last, they will come together.”  
(Jules Verne)*

*“Today is only one day in all the days that will ever be.”  
(Ernest Hemingway)*

*“I may not have gone where I intended to go,  
but I think I have ended up where I needed to be.”  
(Douglas Adams)*

*“Study hard what interests you the most in the most  
undisciplined, irreverent and original manner possible.”  
(Richard Feynman)*

# Resumo

O problema de operação de reservatórios hidrelétricos tem como objetivo a minimização de custos provenientes da operação de termelétricas no atendimento de demandas de energia elétrica. A solução do problema consiste na determinação de políticas operacionais que definam o quanto cada reservatório deve defluir dado seus estados. Uma alternativa de solução é a programação dinâmica, devido às características markovianas e à presença de variáveis aleatórias. Por outro lado, a programação dinâmica convencional sofre da “maldição da dimensionalidade”, crescimento exponencial da complexidade computacional, que tende a limitar sua aplicabilidade.

O trabalho desenvolve uma alternativa com ideias de programação dinâmica aproximada para contornar as dificuldades de complexidade computacional que levam à “maldição da dimensionalidade”. A estratégia desenvolvida adota simulações com cenários de vazões equiprováveis para concentrar a solução do problema no espaço de estados com mais probabilidade de serem visitados. Um procedimento de inferência baseado em redes neurais e interpolações lineares é usado para complementar o espaço de soluções. O estudo de caso, operação do reservatório de Ilha Solteira, procura avaliar a qualidade das aproximações e inferir os benefícios na redução da complexidade computacional.

**Palavras-chave:** Usinas hidrelétricas. Política energética. Processo estocástico. Programação dinâmica. Redes Neurais (Computação).

# Abstract

The problem of hydroelectric reservoir operation which the objective is search to minimize the time-dependent costs from the operation of thermoelectric plants that provides the complementary power to meet demands of electric energy. The solution to the problem is to determine operational policies (rules) define how much each reservoir should flow given its states. A possible approach to solve this problem is dynamic programming, due to the explicit markovian characteristics and the presence of random variables. On the other hand, conventional dynamic programming suffers from the “curse of dimensionality”, exponential growth of computational complexity, that limits its applicability.

This thesis proposes a comparison between stochastic dynamic programming and an approximated dynamic programming approach to address the stochastic optimization problem. The method proposed in this paper use scenario simulations to restrict the problem solution to the most probable configurations. A neural network and linear interpolation inference procedure complements the space of solutions. Case studies, operation of Ilha Solteira reservoir, are designed to evaluate the quality of both approaches and infer possible benefits in the reductions of computational complexity.

**Keywords:** Hydroelectric power plants. Energy policy. Stochastic process. Dynamic programming. Neural networks (Computer science).

# Lista de ilustrações

Figura 1.1 – Esquemático simplificado de um sistema de energia elétrica (SEE) . . .	18
Figura 1.2 – Ilustração do problema de planejamento da operação de reservatórios (POR). . . . .	21
Figura 1.3 – Exigências dos modelos de sistemas geradores em função do horizonte de planejamento. . . . .	23
Figura 1.4 – Custo com complementação térmica em função do volume de água defluído. . . . .	23
Figura 1.5 – Contribuição dos diferentes tipos de fontes de geração no SIN. . . . .	24
Figura 1.6 – Esquemático da ideia de reservatório equivalente. . . . .	28
Figura 1.7 – Sistema sul-sudeste parcialmente representado. . . . .	31
Figura 2.1 – Modelo de reservatório e a representação física das variáveis do sistema. . . . .	33
Figura 2.2 – Funções de custo de geração térmica (a esquerda) e de geração hidráulica em função do volume defluído (a direita). . . . .	33
Figura 2.3 – Topologias reais de reservatórios. . . . .	36
Figura 2.4 – Ilustração dos tipos de vazões afluentes. . . . .	37
Figura 2.5 – Ilustração da relação entre a densidade de distribuição log-normal e normal. . . . .	40
Figura 3.1 – Ilustração do princípio da otimalidade de Bellman. . . . .	47
Figura 3.2 – Procedimentos de discretização do espaço de estado e avaliação para cada ponto do espaço discretizado. . . . .	48
Figura 3.3 – Evolução temporal da solução da equação recursiva. . . . .	49
Figura 3.4 – Reconstrução da trajetória ótima da PD determinística. . . . .	50
Figura 3.5 – Diferenças na programação dinâmica estocástica. . . . .	55
Figura 3.6 – Problema de otimização em rede de pequeno porte. . . . .	57
Figura 3.7 – Método de enumeração para um problema com apenas duas decisões possíveis. As curvas em vermelho representam trajetórias inactíveis e a azul a trajetória ótima. . . . .	57
Figura 3.8 – Interpretação da PD como um problema de controle em malha fechada (Diagrama particular para o problema de POR). . . . .	58
Figura 3.9 – Solução da equação recursiva apenas para um ponto do espaço de estados para cada estágio. As regiões em verde indicam o intervalo de valores de $\mathbf{X}_{k+1}$ que o sistema pode vir a ter se um controle específico for adotado. As regiões em vermelho indicam estados proibidos. . . . .	60
Figura 3.10 – Primeira iteração do algoritmo, destaque para a atualização da função $F_k$ . . . . .	61
Figura 3.11 – N iterações do algoritmo aproximado. . . . .	61

Figura 3.12–Ilustração do processo da avaliação da equação recursiva de otimalidade, dado o estado do sistema, tanto para PDE quanto PDA . . . . .	63
Figura 4.1 – Conceitos introdutórios para triangulações no $\mathbb{R}^2$ . . . . .	67
Figura 4.2 – Ilustração da teorema de Thales, cujo resultado afirma que $\theta_3 < \theta_2 < \theta_1$ . . . . .	68
Figura 4.3 – Ilustração da operação de giro em algoritmos de triangulação. . . . .	69
Figura 4.4 – Ilustração da operação de interpolação usando as triangulações de Delaunay. . . . .	70
Figura 4.5 – Exemplo do uso da biblioteca na produção de malhas triangulares. Nota-se que nenhum círculo (tracejado) definido pelos triângulos da malha contém pontos (vermelho) do conjunto $R$ utilizado na triangulação. . . . .	71
Figura 4.6 – Representação das variáveis do bloco de processamento fundamental em RNAs, o nerônio. . . . .	72
Figura 4.7 – Rede neural do tipo <i>perceptron</i> de múltiplas camadas (MLP). . . . .	74
Figura 4.8 – Exemplificação do paradigma de aprendizado e representação das conexões de neurônios sequenciais. . . . .	74
Figura 4.9 – Solução do problema de aproximação de funções por MLPs. . . . .	76
Figura 5.1 – Distribuições dos dados retirados do banco de monitoramento da ANA para Ilha Solteira. . . . .	79
Figura 5.2 – Comparativo entre as políticas da PDE (verde) e da PDA (azul) de janeiro a dezembro. . . . .	80
Figura 5.3 – Distribuição de custos para a política determinada segundo a PDE para Ilha Solteira com vazões independentes. . . . .	81
Figura 5.4 – Frequência de visitação dos estados no algoritmo da PDA de janeiro a dezembro. . . . .	82
Figura 5.5 – Comparação entre as distribuições de custos variando o número de interações do método aproximado e a distribuição ótima (PDE). . . . .	83
Figura 5.6 – Comparação entre as abordagens para o problema de POR não markoviano. . . . .	84
Figura 5.7 – Política de Ilha Solteira definida pela PDE com vazões markovianas. . . . .	86
Figura 5.8 – Distribuição de custos para a política determinada segundo a PDE para Ilha Solteira com vazões markovianas. . . . .	86
Figura 5.9 – Valores esperados da energia produzida mensalmente por Ilha Solteira e sua complementação termoelétrica. . . . .	87
Figura 5.10–Dois comportamentos típicos no espaço de estados: meses com abundância em água (março), frequência de visitação estreita e próximas do reservatório cheio, em contraste com meses escassos (outubro) com distribuição central e “uniforme”. . . . .	87

Figura 5.11–Mapas de frequência de estados para o cenário markoviano: eixos $xy$ correspondem aos estados discretos de volume e vazão e o mapa de cores a frequência de visitação de cada estado, com escala normalizada pelo estado mais frequente de cada mês. . . . .	88
Figura 5.12–Política de Ilha Solteira definidas pela PDA com interpolações lineares usando triangulações de Delaunay com vazões markovianas. . . . .	89
Figura 5.13–Comparação entre as abordagens para o problema de POR markoviano.	90
Figura 5.14–Convergência das abordagens sem e com interpolações por triangulações para o POR markoviano. . . . .	90
Figura 5.15–Comparativo para as políticas para a PDE (verde) e para PDA (azul) de janeiro a dezembro. . . . .	91
Figura 5.16–Comparação entre as abordagens para o problema de POR não markoviano.	92
Figura 5.17–Comparação entre as abordagens para o problema de POR não markoviano.	92

# Lista de tabelas

Tabela 1 – Características físicas do reservatório de Ilha Solteira . . . . .	78
Tabela 2 – Demandas contratadas $d_k[MW.mes]$ e distribuições das vazões $y_k[m^3/s]$ . . . . .	78
Tabela 3 – Resultados parciais do método para o problema unidimensional. . . . .	83
Tabela 4 – Resultados finais do método para o problema bidimensional. . . . .	93

# Lista de abreviaturas e siglas

CEPEL	Centro de Pesquisas de Energia Elétrica
MLP	<i>Multilayer Perceptron</i>
ONS	Operador Nacional do Sistema Elétrico
PD	Programação dinâmica
PDE	Programação dinâmica estocástica
PDDE	Programação dinâmica estocástica dual
POR	Planejamento de operação de reservatórios
RNA	Redes Neurais Artificiais
SEE	Sistemas de energia elétrica
SIN	Sistema integrado nacional
WEC	<i>World Energy Council</i>

# Sumário

<b>1</b>	<b>OPERAÇÃO E PLANEJAMENTO ENERGÉTICO</b>	<b>18</b>
1.1	Introdução ao planejamento de sistemas elétricos	18
1.2	Sistemas de geração de energia elétrica	19
1.3	Planejamento da operação de sistemas hidrotérmicos	21
1.4	Sistema interligado nacional	24
1.5	Revisão Bibliográfica	25
1.5.1	Trabalhos pioneiros	25
1.5.2	Década de 70	27
1.5.3	Década de 80	28
1.5.4	Década de 90 e anos subsequentes	29
<b>2</b>	<b>FORMULAÇÃO DO PROBLEMA</b>	<b>32</b>
2.1	Modelagem matemática do reservatório	32
2.1.1	Generalização para múltiplos reservatórios	35
2.2	Modelagem estatística das vazões	37
2.2.1	Vazões independentes	37
2.2.2	Vazões markovianas	41
<b>3</b>	<b>PROGRAMAÇÃO DINÂMICA</b>	<b>45</b>
3.1	Programação dinâmica determinística	47
3.1.1	Solução da equação recursiva de otimalidade	48
3.1.2	Obtenção da trajetória ótima	50
3.2	Programação dinâmica estocástica	50
3.2.1	As vantagens da PD	55
3.2.2	A maldição da dimensionalidade	58
3.3	Programação dinâmica aproximada	59
3.3.1	Atualizações da função de custo	62
3.3.2	Avaliação de política	63
<b>4</b>	<b>DETALHES DE IMPLEMENTAÇÃO</b>	<b>65</b>
4.1	Exploração x Interpolação	66
4.2	Métodos interpolativos	66
4.2.1	Triangulação de Delaunay	66
4.2.2	Redes Neurais	71
4.3	Considerações sobre o POR no contexto da PD	76

<b>5</b>	<b>ENSAIOS E ESTUDO DE CASOS</b>	<b>78</b>
<b>5.1</b>	<b>Estudo de casos</b>	<b>78</b>
<b>5.2</b>	<b>Ensaio com vazões independentes</b>	<b>79</b>
<b>5.3</b>	<b>Ensaio com vazões markovianas</b>	<b>85</b>
5.3.1	PDE	85
5.3.2	PDA	85
5.3.3	PDA com triangulação de Delaunay	88
5.3.4	PDA com redes neurais artificiais	91
<b>6</b>	<b>CONCLUSÕES E PERSPECTIVAS FUTURAS</b>	<b>95</b>
	<b>REFERÊNCIAS</b>	<b>97</b>

# Apresentação e estrutura do texto

O planejamento de operação de sistemas hidrotérmicos, ou de reservatórios (POR), busca estratégias de operação que reduzam custos de geração sem comprometer a confiabilidade ou recorrer a expansão do parque gerador. Em outras palavras, consiste na determinação de políticas (funções) que indiquem qual a melhor decisão a ser tomada, quanto de água defluir por cada usina que compõe o sistema, de maneira a se atender demandas energéticas contratadas e minimizar o custo com a complementação por energia térmica.

O POR tem sido abordado com diferentes técnicas, desde modelos clássicos baseados em gradiente, métodos de fluxos em redes, programação dinâmica (PD) e até as meta-heurísticas. A aplicação de PD no POR é defendida por permitir a imediata adição de efeitos estocásticos e flexibilidade com as condições da função objetivo. No entanto, a programação dinâmica sofre da conhecida “maldição da dimensionalidade” (crescimento exponencial da complexidade computacional) que tende a limitar suas aplicações.

Desde a década de 60 existe um grande esforço pelos usuários da técnica pela busca de um método aproximado eficiente, assim como esse trabalho, onde o foco está em contribuir com uma metodologia aproximada para a otimização de sistemas de geração hidrotérmicos de médio prazo, com discretizações mensais e vazões aleatórias markovianas. A abordagem aproximada se baseia em conceitos como o de frequência de visitação dos estados do sistema, limitando a solução apenas a uma região provável. Essas ideias são associadas a métodos de interpolação como redes neurais e interpolações lineares aplicadas em um estudo de caso para o reservatório de Ilha Solteira.

Do ponto de vista de estrutura, o trabalho está dividido em cinco capítulos. No primeiro deles, é apresentado todo o contexto do POR e como ele tem sido abordado ao longo dos últimos 60 anos em especial por PD. Além disso é papel do capítulo inicial estabelecer a importância e relevância do problema estudado.

Nos capítulos intermediários, segundo e terceiro, são apresentados em detalhes toda a modelagem utilizada no problema e expõem a técnica utilizada, apresentando a metodologia tradicional de PD em contraste com a proposta, além de revisar todos os conceitos estatísticos aplicados a modelagem de vazão do problema. No quarto capítulo é dado um panorama geral sobre algumas escolhas específicas tomadas sobre o algoritmo implementado e, de forma bastante resumida, são apresentadas algumas ferramentas usadas no método aproximado.

No quinto capítulo é apresentada toda a estrutura dos ensaios realizados, em uma ordem cronológica que possa ilustrar a progressão do trabalho, e como sua evolução foi

permitindo tomar decisões e quais foram elas, justificando os conceitos apresentados no quarto capítulo. Cada justificativa é devidamente ilustrada por algum ensaio ou resultado. Finalmente o sexto capítulo, apresenta as principais conclusões e deixa sugestões de caminhos para trabalhos futuros.

# 1 Operação e planejamento energético

*Nesse capítulo inicial é realizada uma introdução ao planejamento energético, em especial o de operação de sistemas hidrotérmicos. Esse tipo de planejamento busca por estratégias de operações que reduzam custos de geração sem comprometer a confiabilidade e sem recorrer a expansão do parque gerador. Na segunda parte do capítulo, há uma revisão bibliográfica do problema, dando uma atenção especial a trabalhos com programação dinâmica. Ao final do capítulo, faz-se uma breve descrição sobre o sistema gerador brasileiro.*

## 1.1 Introdução ao planejamento de sistemas elétricos

Entende-se como planejamento a ação combinada de análise e organização de informações sobre um sistema de maneira a se justificar cada etapa de uma estratégia, ou política, e que possa sustentar a tomada de decisões em busca de alcançar um determinado objetivo (FORTUNATO et al., 1990). No caso de sistemas de energia elétrica (SEE), o objetivo, em geral, é atender demandas energéticas a custo mínimo sem comprometer fatores como confiabilidade, segurança e qualidade.

Os sistemas elétricos são divididos em três grandes componentes: geração, transmissão e distribuição. A primeira componente, a geração, é composta de elementos de conversão de energia potencial em energia elétrica. Já a transmissão é responsável pelo transporte da energia elétrica produzida na componente anterior. A distribuição lida com a entrega da energia a cada um dos consumidores. A Figura 1.1, adaptada de Conejo e Baringo (2018) ilustra um sistema elétrico. Na geração é desejável usar de maneira ótima os recursos naturais disponíveis, e maximizar os benefícios da produção de energia elétrica em cada uma das unidades produtoras.

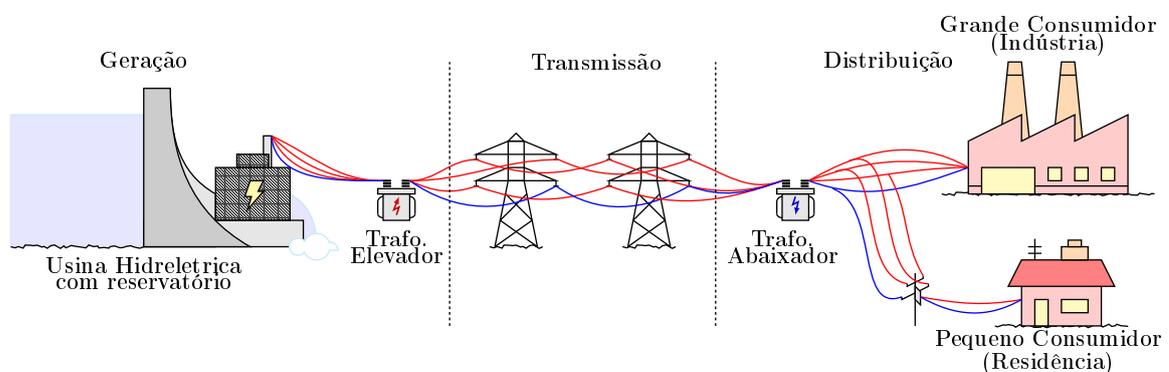


Figura 1.1 – Esquemático simplificado de um sistema de energia elétrica (SEE)

O planejamento de sistemas elétricos pode ser dividido entre a expansão e a operação

do SEE (FORTUNATO et al., 1990). O planejamento da expansão do sistema se encarrega principalmente do crescimento físico do SEE, de maneira a acompanhar as expectativas de elevações no consumo. É função desse tipo de planejamento a determinação geográfica, dimensionamento físico dos elementos do sistema (como usinas, torres de transmissão e subestações), o tipo de tecnologia de geração, e o estudo de viabilidade técnica e econômica para o atendimento de variações futuras das demandas energéticas.

Por outro lado, o planejamento da operação busca estabelecer regras, ou políticas, que guiem as decisões sobre cada elemento já existente do SEE. O objetivo desse planejamento é minimizar os custos operacionais, que incluem gastos com combustíveis na geração térmica, importação e exportação de energia de/para sistemas interligados e multas por demandas contratadas não atendidas. É importante que o planejamento operacional consiga de forma ótima controlar o sistema contando pouco, ou nada, com a expansão. Isso porque expandir o sistema é um processo caro e demorado, que deve ser usado para atender ao aumento das demandas em horizontes mais longos.

## 1.2 Sistemas de geração de energia elétrica

A geração de energia elétrica está na base do SEE e é composta principalmente de grandes projetos de engenharia para a conversão de algum tipo de fonte de energia potencial (gravitacional, química, solar, eólica, etc.) em energia elétrica, utilizando, na maioria dos casos, grandes máquinas eletro-mecânicas rotativas (CONEJO; BARINGO, 2018). Nessa descrição é possível incluir os dois elementos geradores mais importantes para o decorrer deste trabalho: as hidrelétricas e as termoelétricas.

As usinas hidráulicas que possuem reservatórios são estruturas que têm a capacidade de armazenar água, o que cria uma diferença de potencial devido às cotas a montante e jusante da barragem. A energia potencial é convertida em energia cinética (mecânica) quando a água estocada a montante é forçada a passar por turbinas, o que produz eletricidade (FORTUNATO et al., 1990). Em outros casos a hidrelétrica não possui esses reservatórios, o que faz a cota a montante ser fixa, sem capacidade de estocar água (energia). Ainda em outros casos, existem estruturas só de reservatórios (sem capacidade de geração) que funcionam apenas no sentido de regulação hídrica. No decorrer deste trabalho as duas primeiras estruturas serão frequentemente mencionadas. A primeira simplesmente por reservatório hidrelétrico e a segunda como usinas a fio d'água.

O planejamento do melhor uso de sistemas hidrelétricos requer considerar uma diversidade de fatores na sua modelagem. Por exemplo, a geração elétrica é extremamente dependente das vazões previstas no sistema hídrico no qual a usina está inserida, que em geral podem ser modeladas por variáveis aleatórias. Além disso, raramente se encontra sistemas reais onde um reservatório está isolado do funcionamento de outros, e a presença

de usinas, tanto a montante quanto a jusante, interfere na escolha de como usar a água armazenada por eles.

As usinas termoeletricas, utilizam da combustão para produzir fluidos com altas temperaturas e capazes de movimentar turbinas, transformando assim energia térmica em mecânica e, posteriormente, em energia elétrica (FORTUNATO et al., 1990). As usinas termoeletricas são extremamente diversificadas, podem ser a vapor ou a gás, com ciclos termodinâmicos diferentes (combinados ou simples), e muito flexíveis em relação aos combustíveis, podendo ser carvão, diesel, gás natural, biomassa ou combustível nucleares.

É importante destacar algumas diferenças entre os dois tipos de geração. Por exemplo, as usinas térmicas são independentes uma das outras, enquanto que as hidráulicas possuem um acoplamento que as torna dependentes. O custo de operação de usinas térmicas (fortemente associados aos custos com combustíveis) é variável, crescente e não-linear, enquanto que as usinas hidrelétricas têm custo variável de geração pouco significativo<sup>1</sup>. Além disso, o preço do combustível em geral está sujeito às variações de moedas estrangeiras (normalmente; dólar). Esses aspectos somados com o baixo rendimento das máquinas térmicas que dificilmente passam de 50% (TOLMASQUIM, 2005), tornam a geração em termoeletricas mais caras em relação à geração em hidrelétricas.

Já no aspecto físico, as termoeletricas tem menor dimensão e em geral, podem ser localizadas próximas às regiões onde a energia é consumida (ONS, 2018). Por outro lado, as hidrelétricas possuem uma inércia de geração muito baixa, ou seja, são capazes de iniciar e alterar sua geração rapidamente (principalmente em relação às térmicas a carvão), o que as qualificam para atender variações diárias de consumo (TOLMASQUIM, 2005).

Do ponto de vista ambiental, os dois tipos de geração possuem seus problemas. Enquanto que as térmicas estão diretamente ligadas à emissão de gases tóxicos, as hidroeletricas estão ligadas a mudanças nos cursos naturais de rios e formação de grandes lagos artificiais, alterando ecossistemas e até a economia local.

A diferença principal entre o planejamento da operação de termoeletricas e hidrelétricas é a fonte primária de energia. Os combustíveis de usinas térmicas estão sempre disponíveis<sup>2</sup>, enquanto que recursos hídricos possuem regimes periódicos de cheias e secas. Ou seja, usinas térmicas contribuem para a confiabilidade no abastecimento, estão sempre disponíveis, enquanto as hidrelétricas não possuem custos de geração, sendo economicamente mais atrativas (CONEJO; BARINGO, 2018),(ONS, 2018).

<sup>1</sup> Os custos existentes são os comuns a toda unidade geradora, independente do tipo (manutenção, administração, etc.). O alto investimento na construção desses sistemas é amortizado em alguns anos e após isso o reservatório produz energia sem descontos.

<sup>2</sup> Não entra-se no mérito de serem recursos esgotáveis, mas sim com disponibilidade para compra em toda época do ano.

### 1.3 Planejamento da operação de sistemas hidrotérmicos

O planejamento da operação de sistemas hidrotérmicos, ou de reservatórios (POR), consiste na determinação de políticas (funções) que digam qual a melhor decisão a ser tomada: quanto de água defluir por cada usina que compõe o sistema, de maneira a se atender demandas energéticas contratadas e minimizar o custo com a complementação por energia térmica, Figura 1.2 - adaptada de Lyra Filho (1984). Ou seja, em planejamento da operação de sistemas hidrotérmicos, o cumprimento das demandas contratadas tem como a prioridade o abastecimento com energia hidrelétrica, em consequência dos seus menores custos de operação. Existe também a alternativa de importação de energia de sistemas eletricamente interligados, caso o custo seja mais vantajoso em relação à geração térmica; esses aspectos são ilustrados na Figura 1.2.

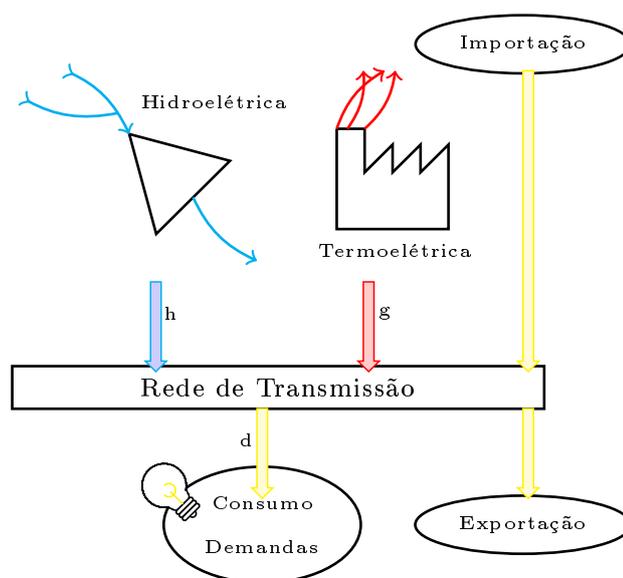


Figura 1.2 – Ilustração do problema de planejamento da operação de reservatórios (POR).

Uma estratégia normalmente adotada para levar em consideração fatores relevantes na modelagem (como vazões aleatórias, acoplamentos hídricos, não-linearidades nas funções de geração) é realizar uma sequência de planejamento acoplados e divididos temporalmente pelo intervalo de interesse e horizontes de planejamento (FORTUNATO et al., 1990). Não existe um consenso sobre essa classificação temporal nos trabalhos sobre planejamento, seja ele de expansão ou operação. Porém, uma classificação usual é dividir o problema em planejamentos de longo, médio e curto prazo além do pré-despacho. Uma breve descrição sobre cada um desses planejamentos é apresentada a seguir (LYRA FILHO, 1984).

- **Planejamento de longo prazo:** São considerados horizontes de até algumas dezenas de anos, com discretizações mensais. Suas principais funções visam as necessidades globais dos sistemas, como importação e exportação de energia, deficit de abastecimento e viabilização de mudanças na operação global do sistema. Em geral são realizados para o sistema gerador completo, definem políticas agregadas e lidam com alto nível de incertezas, principalmente de demandas.
- **Planejamento de médio prazo:** Com horizontes de até um ou dois anos, com discretizações mensais ou semanais. São aplicadas em sistemas onde se pretende individualizar a política de cada um dos elementos geradores (reservatórios, por exemplo) e tendem a propor políticas que busquem atender as metas do planejamento de longo prazo.
- **Planejamento de curto prazo:** Horizontes de até um mês com discretizações diárias. Assim como no planejamento de médio prazo, a ideia básica é fazer com que as metas dos planejamentos mais longos se tornem alcançáveis. Em geral, levam em conta restrições e aspectos não abordados nos planejamentos mais longos, como tempos de deslocamento da água entre as usinas, por exemplo. Além disso, em geral, a curto prazo podem-se supor conhecidas as vazões dos rios.
- **Pré-despacho:** Faz uma decomposição hora-hora das políticas de curto prazo. Prepara o sistema para operação em tempo real

Ou seja, a diversidade de detalhes do sistema faz com que a operação de reservatórios não tenha um modelo único que consiga abordar todos os fenômenos que permeiam o problema (ROSENTHAL, 1980),(PEREIRA; PINTO, 1985). Sendo assim, o modelo deve ser ajustado dependendo de qual o objetivo da otimização e de qual aspecto do problema deseja-se tratar. Em linhas gerais, quanto menor o intervalo de planejamento mais detalhes são necessários e menos aleatoriedades estão presentes; quanto mais longo o planejamento, menos detalhes são exigidos, porém o sistema passa a ser cada vez mais incerto. A Figura (1.3) representa esquematicamente esses aspectos. Neste trabalho os horizontes são compatíveis com o planejamento de médio prazo, com intervalos de discretização mensais.

A operação de usinas hidrelétricas precisa manter um equilíbrio entre o instante presente e o futuro do sistema (SIQUEIRA, 2009). Entende-se como presente a utilização imediata do uso da água armazenada, reduzindo os custos na complementação térmica e a confiabilidade no atendimento das demandas futuras. Já por instante futuro, se refere a manter o sistema no armazenamento máximo para diminuir o risco de deficit e aumentando a confiabilidade e uso de térmicas.

A ideia de custos presente e futuro, respectivamente, custo da decisão presente tomada sobre o sistema e a expectativa de custo que essa decisão impõe no futuro, Figura

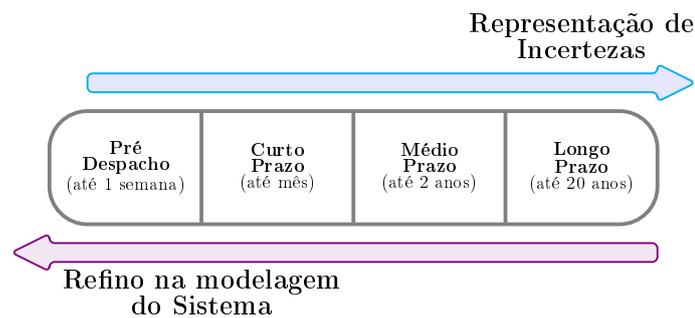


Figura 1.3 – Exigências dos modelos de sistemas geradores em função do horizonte de planejamento.

1.4 - adaptada de Siqueira (2009), estabelece paralelo direto com a técnica utilizada nesse trabalho para o POR, a programação dinâmica (PD). A PD se baseia na solução da equação recursiva de otimalidade que basicamente é composta de duas parcelas, uma dependente do instante atual e outro do próximo o que dialoga com a busca do problema de operação de reservatórios por um compromisso entre atender demandas no presente e reduzir prováveis esgotamentos ou baixos níveis em reservatórios hidrelétricos no futuro.

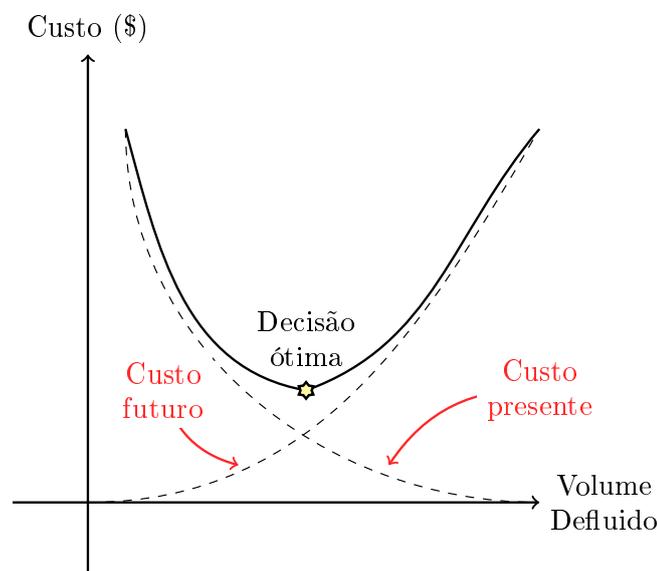


Figura 1.4 – Custo com complementação térmica em função do volume de água defluído.

Em resumo, os fatores que envolvem a operação de reservatórios o classificam como um problema de otimização de tomada de decisões de grande porte, dinâmico, estocástico e não-linear (LYRA FILHO, 1984).

## 1.4 Sistema interligado nacional

O sistema interligado nacional (SIN) é o sistema integrado gerador e transmissor brasileiro que é composto por quatro regiões: Norte, Nordeste, Sul e Sudeste/Centro-Oeste. Cada uma dessas regiões possui características distintas, como regime fluvial e tipos alternativos de fontes de geração. A interligação desses sistemas faz com que o SIN possua uma interação complexa entre cada região, o que torna o Brasil um laboratório rico em estudo de casos para o POR.

Atualmente, segundo o operador nacional do sistema elétrico (ONS), o SIN é formado principalmente por usinas hidrelétricas (mais de 150) e apenas 45% do potencial hídrico brasileiro (246 MW) foi desenvolvido (ELETROBRAS, 2017). De acordo com o último levantamento da WEC (2016), o Brasil ocupa a posição de segundo maior produtor de energia hidrelétrica, perdendo somente para a China e ligeiramente a frente do Canadá.

Mesmo com o crescimento de fontes alternativas de geração (eólica e solar, principalmente na região Norte), as hidrelétricas ainda contribuem com quase 110 GW dos 162 GW produzidos no SIN, 67,5% de contribuição. A Figura (1.5) ajuda a ilustrar a contribuição das diferentes fontes geradoras do sistema brasileiro segundo a ONS (2018).

Sobre as bacias do país duas merecem destaque. A primeira delas, a Bacia do Rio Amazonas, detém 40% do potencial hídrico do país e ao mesmo tempo possui apenas 1% desse potencial aproveitado. A segunda, a bacia do rio Paraná, é uma das principais bacias hidrográficas geradoras do mundo e atualmente a mais desenvolvida do país, localizada na região mais privilegiada economicamente do Brasil (TOLMASQUIM, 2005). Dentre as principais usinas que compõem o SIN pode-se citar as usinas de Tucuruí (8085MW/ Rio Tocantins), Xingó (3162MW/ Rio São Francisco), Ilha Solteira (3444 MW/ Rio Paraná), Itumbiara e São Simão (2124MW e 1710MW/ Rio Paranaíba).

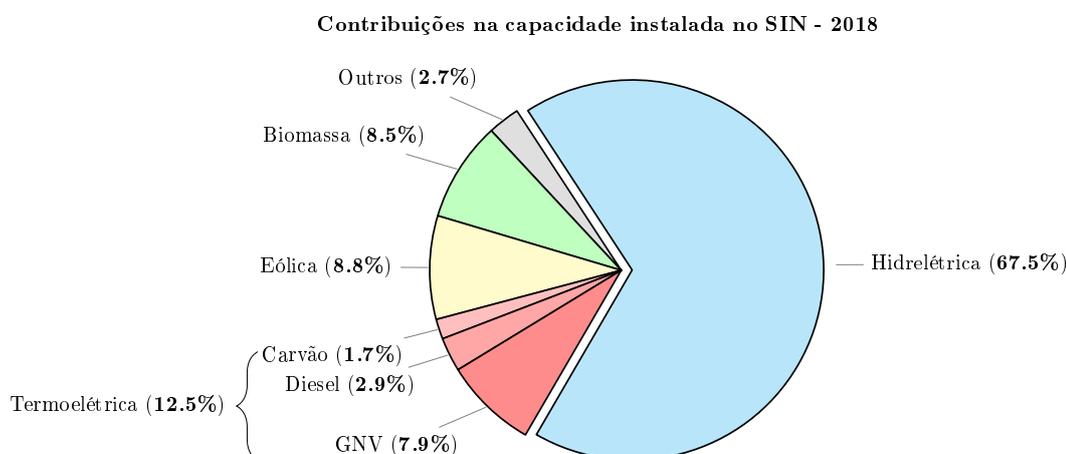


Figura 1.5 – Contribuição dos diferentes tipos de fontes de geração no SIN.

Segundo Neto et al. (1984) o SIN, até meados da década de 70, foi operado

seguindo a ideia das curvas limites e posteriormente passou a ser operado utilizando técnicas de programação dinâmica estocástica (PDE), seguindo principalmente os trabalhos desenvolvidos pela ELETROBAS/CEPEL. A partir dos anos 90, a programação dinâmica dual passou a ser a principal referência de técnica de operação de sistemas hidrotérmicos de longo e médio prazo.

De acordo com o [CEPEL \(2019\)](#), o modelo que atualmente trata do planejamento de médio e longo prazo é o NEWAVE<sup>3</sup>. Esse modelo não só é responsável pela operação mas também pela expansão e assuntos relacionados à comercialização de energia. Além disso, o CEPEL deixa claro que o NEWAVE trabalha em conjunto com outros dois modelos: DECOMP (Modelo de Planejamento da Operação de Sistemas Hidrotérmicos Interligados de Curto Prazo) e DESSEM (Modelo de Despacho Hidrotérmico de Curto Prazo). Isso para que todos os detalhes do problema sejam abordados no planejamento, como discutido na Seção 1.3.

## 1.5 Revisão Bibliográfica

O planejamento de operação de reservatórios (POR) é um problema que historicamente tem sido abordado com diferentes técnicas. Os trabalhos de [Yakowitz \(1982\)](#), [Yeh \(1985\)](#), [Rosenthal \(1980\)](#) e [Labadie \(2004\)](#) fazem uma descrição sobre a evolução dos métodos, abordagens e formulações do problema, esse último já incluindo abordagens com meta-heurísticas.

Uma das vertentes de tratamento do problema é a que defende a utilização da programação dinâmica, principalmente pela possibilidade de lidar adequadamente com os aspectos aleatórios ao problema; este trabalho segue esta linha de exploração. Dessa forma, as subseções a seguir tratam principalmente de trabalhos que utilizam programação dinâmica às vezes com alguma reformulação nos modelos do problema para lidar com a complexidade computacional. As exceções são essencialmente trabalhos com uma contribuição muito significativa para a área de planejamento da operação de reservatórios.

### 1.5.1 Trabalhos pioneiros

Por muito tempo a operação de reservatórios utilizou-se das chamadas curvas limites, que compõem regras muito simples de deságue, onde a água é liberada caso o sistema encontre-se acima dessas curvas. A determinação das curvas limites ocorre através de métodos de otimização determinísticos (seja programação linear ou não-linear, métodos de otimização de fluxos em redes, etc.) onde as vazões são conhecidas e são parte do pior

---

<sup>3</sup> O NEWAVE não só usa a programação dinâmica dual estocástica (PDDE) como também utiliza a ideia dos reservatórios equivalentes para conseguir tratar aleatoriedades e sistemas com número elevado de unidades geradoras.

cenário do histórico do sistema (ano mais seco). As curvas limites ficaram populares não só pela facilidade dos operadores a utilizarem, mas também por dar ao sistema a segurança contra o pior cenário possível.

Um dos primeiros a questionar as curvas limites foi [Maass et al. \(1962\)](#), e a justificativa para isso é que por mais que as curvas limites protejam o sistema do pior cenário essa é, em geral, a situação menos provável de acontecer. Ou seja, em troca da confiabilidade perde-se a oportunidade de explorar os cenários mais prováveis. Porém, antes disso, [Masse \(1946\)](#), já explorava a ideia de valor marginal da água, o qual ajuda na composição de políticas de operação. Masse também foi um dos primeiros autores a sugerir a ideia de que o problema seja formulado como um problema sequencial de tomada de decisão. Autores citados em ([LYRA FILHO, 1984](#)).

Com o avanço teórico proporcionado por Richard Bellman ainda na década de 50, a programação dinâmica (PD) ganhou relevância principalmente em sistemas com forte características de tomada de decisão e com presença de variáveis aleatórias, onde se encaixa a operação de reservatórios. A PD utiliza dois tipos básicos de variáveis: de estado e de decisão. A primeira descreve a situação corrente de como o sistema se encontram (volume armazenado no reservatório) e a segunda a ação que se toma sobre o sistema (volume defluído pelo reservatório). A PD busca estabelecer a relação qualquer estado e a decisão ótima.

Um dos primeiros trabalhos a tratar a gestão dos reservatórios com programação dinâmica estocástica (PDE) foi [Little \(1955\)](#), já incluindo variáveis aleatórias correlacionadas na operação do reservatório de Grand Coulee no Rio Columbia, EUA.

Os trabalhos de [Hall e Buras \(1961\)](#) e [Hall, Butcher e Esogbue \(1968\)](#) trouxeram novas releituras da operação de sistemas hidro-térmicos. Os autores abordaram problemas determinísticos de otimização de desenvolvimento e expansão de reservatórios (dimensionamento) e otimização de sistemas hidrotérmicos com múltiplos preços associados a produção de energia (aplicado a barragens de Shasta e Keswick no rio Sacramento, EUA), fazendo uma combinação linear ponderada de preços.

Outro estudo de destaque com variáveis aleatórias é o de [Hall e Howell \(1963\)](#), onde os autores sugerem substituir políticas que usem o cálculo explícito de esperanças matemáticas, por políticas que sejam determinadas seguindo algum critério de análise (como o maior retorno ou média da função objetivo) das soluções do problema determinístico para uma série de vazões realizáveis. Esse tipo de abordagem faz parte dos métodos estocásticos implícitos, por tentarem tratar o comportamento estocástico de forma indireta.

As facilidades e dificuldades da programação dinâmica são enunciadas por [Bellman \(1957\)](#) já nas primeiras páginas de seus trabalhos. Há tempos autores se dedicam a trazer formas alternativas para contornar, principalmente, a “maldição da dimensionalidade”

(crescimento exponencial da complexidade computacional).

Um exemplo clássico com técnicas aproximadas é a programação dinâmica incremental, onde o espaço de busca do problema é limitado pela política obtida na iteração anterior. Um dos primeiros a aplicar essa técnica no problema de operação de reservatórios foi [Bernholtz e Graham \(1960\)](#), em um trabalho fundamentado com critérios econômicos de otimização, e que argumenta a limitação do espaço de busca pelo fato da técnica fazer buscas incrementais de custo, que só valem para variações pequenas na política.

Posteriormente, [Larson \(1968\)](#) propôs novas práticas de limitação do espaço de estados, fazendo buscas em “blocos” e usando os resultados de forma aproximada para obter informação de estados ainda não calculados<sup>4</sup>. Nos anos seguintes trabalhos como o de [Hall et al. \(1969\)](#) aplicaram a técnica para sistemas com dois reservatórios. A técnica incremental se limita ao caso determinístico.

### 1.5.2 Década de 70

Na década seguinte, alguns trabalhos retornam as tentativas de aproximações já realizadas antes, seja para aprimorar as aproximações ou na tentativa de explorar seus limites. Os trabalhos de [Roefs e Bodin \(1970\)](#) e [McKerchar \(1975\)](#) resgatam a ideia de operar com vazões aleatórias utilizando as soluções do problema determinístico e usando vazões realizáveis. O primeiro deles usando métodos de programação linear e o segundo com programação dinâmica, ambos com métodos similares de regressão na construção das políticas e aplicados a sistemas de porte real, para três reservatórios (Shasta, Trinity e Folsom, no rio Sacramento) e dois (nos rios Tekapo e Pukaki, Nova Zelândia).

Já [Heidari et al. \(1971\)](#) utilizou a ideia da programação dinâmica em vizinhanças, como na programação dinâmica incremental, em um método que ficou conhecido como programação dinâmica diferencial discreta. Nesse variação, existe uma limitação dos estados futuros do sistema para diminuir o espaço de busca, como envolvendo os estados factíveis por um “túnel” e permitindo a busca dentro dessa região, que a cada iteração é atualizada conforme o algoritmo atinge a superfície dessa restrição.

Uma das contribuições mais significativas desse período são os trabalhos de ([ARVANITIDIS; ROSING, 1970a](#)), e seu desmembramento ([ARVANITIDIS; ROSING, 1970b](#)), e ([JACOBSON; MAYNE, 1970](#)). O primeiro destaque no âmbito da operação de reservatórios, onde é proposta uma representação de múltiplos reservatórios em um único equivalente de mesma energia potencial, que pode ser facilmente otimizado, Figura (1.6). Já no segundo é proposta uma formulação alternativa que ficou conhecida como programação dinâmica diferencial onde em cada estado a equação de otimalidade é aproximadas por uma fun-

<sup>4</sup> Ambos os trabalhos se propuseram a resolver um problema “físico” comum a época deles, que era a limitação de memória dos computadores e processamento. Inclusive [Larson \(1968\)](#) discute metodologias que usem o disco rígido para superar pequenas memórias.

ção de segunda e primeira ordem nos estados para o custo e políticas, isso permite que técnicas de programação linear e os métodos do tipo gradiente possam ser aplicados e sem discretização no estados, em troca de garantias de diferenciabilidade. O trabalho de (MURRAY; YAKOWITZ, 1979) faz aplicação do método para 4 e 10 reservatórios, ainda no problema determinístico.

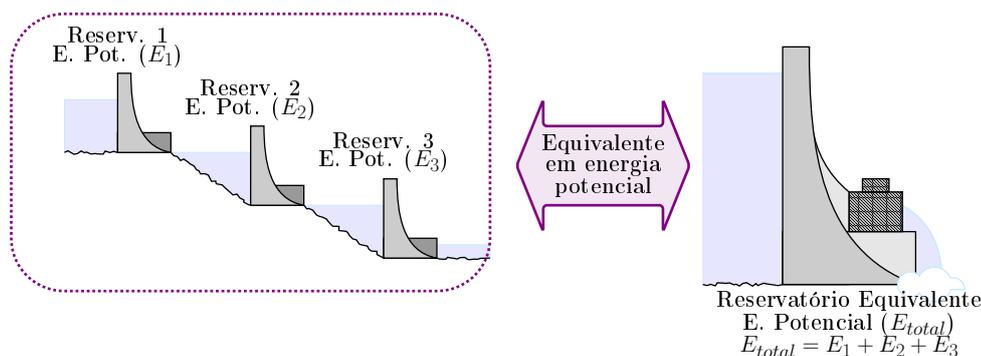


Figura 1.6 – Esquemático da ideia de reservatório equivalente.

### 1.5.3 Década de 80

Na década de 80, grande parte dos trabalhos passam a sugerir a utilização de modelos equivalentes ou soluções do tipo “um de cada vez”. Nos trabalhos Turgeon (1980) e Turgeon (1981) são discutidos os benefícios de cada formulação. A operação com modelos equivalentes, ou de agregação-desagregação<sup>5</sup>, é aplicada por Duran et al. (1985) para o sistema colombiano com 10 reservatórios, em especial usando um tipo de agregação a cada dois reservatórios. Esse tipo de agregação usa um reservatório real e outro equivalente de todos os outros a jusante (ignorando os a montante), isso para toda a cascata de reservatórios de montante à jusante. Já Sherkat et al. (1985) otimizou utilizando o que ficou conhecido como aproximações sucessivas, ou “um de cada vez”, onde cada reservatório é otimizado independentemente tendo apenas uma previsão da política dos reservatórios a jusante, recursivamente até a convergência.

O grande destaque desse período é o trabalho (PEREIRA; PINTO, 1985) que faz o uso das decomposições de Benders e métodos de programação linear para otimizar sistemas com múltiplos reservatórios com variáveis aleatórias. Além disso, o trabalho de Pereira e Pinto faz tanto aplicações pequenas de quatro reservatórios quanto para sistemas com 32 reservatórios do sistema Sul-Sudeste brasileiro<sup>6</sup>. Pode-se considerar a técnica desenvolvida pelas autores como um dos estados-da-arte da operação de sistemas hidrelétricos, além

<sup>5</sup> Agregar os reservatórios para obtenção da política ótima (otimização) e desagrega-los para aplicar a política para cada um individualmente.

<sup>6</sup> Esse sistema pode se encontra na figura (1.7). O trabalho em questão opera um subconjunto das 55 usinas ilustradas na figura.

de serem os primeiros a lidar com um número alto de reservatórios incluindo fatores estocásticos.

#### 1.5.4 Década de 90 e anos subsequentes

O trabalho de Pereira e Pinto da década anterior foi formalizado por [Pereira e Pinto \(1991\)](#) e definiu a programação dinâmica dual estocástica (PDDE) e tem sido desde então um dos alvos mais frequentes das pesquisas na operação de reservatórios. Trabalhos recentes como os de [Goor, Kelman e Tilmant \(2011\)](#), [Shapiro \(2011\)](#) e [Soares, Street e Valladao \(2017\)](#) fazem estudos sobre as propriedades, convergência e aplicações de novas aproximações para esse método, respectivamente.

Porém, não só de programação dinâmica dual a operação de reservatórios se sustenta nas últimas décadas. Com novas arquiteturas e computadores cada vez mais rápidos e complexos, trabalhos como o de [Piccardi e Soncini-Sessa \(1991\)](#) propõem novas práticas para utilização de paralelismo na solução de problemas com programação dinâmica. Além disso, esse período é marcado pela popularização das meta-heurísticas em diversas áreas e na operação de reservatórios não foi diferente.

Um exemplo de aplicação de heurística para operação com vazões determinísticas, é o trabalho de [Sharif e Wardlaw \(2000\)](#), que faz um estudo exploratório com uma discussão de trabalhos anteriores bem sucedidos com algoritmos genético e faz uma aplicação clássica do método na operação de três reservatórios na bacia do rio Brantas, Indonésia. Um outro exemplo bastante original sobre novas metodologias é o trabalho de [Cai, McKinney e Lasdon \(2001\)](#), que faz a utilização de um algoritmo genético codificando os termos não lineares da função objetivo nos *cromossomos*, tornando assim o problema linear para cada *cromossomo* e medindo seu *fitness* com técnicas de programação linear.

No âmbito das redes neurais artificiais (RNA), [Saad et al. \(1994\)](#) propõem a operação de múltiplos reservatórios, utilizando uma ideia similar de representação equivalente, otimizando deterministicamente para um conjunto de cenários relevantes, as soluções utilizadas na regressão das políticas com RNA de múltiplas camadas (MLP). Outros autores usam estruturas neurais com funções de base radial ([SAAD et al., 1996](#)) ou aprendizagem por reforço ([LEE; LABADIE, 2007](#)). Os autores [Raman e Chandramouli \(1996\)](#) e [Chandramouli e Ramam \(2001\)](#) fazem a aplicação do trabalho anterior para a bacia do rio Aliyar na Índia, otimizando o problema determinístico sem a agregação dos reservatórios e para múltiplos objetivos.

Já o trabalho de [Castelletti et al. \(2006\)](#) faz uma aplicação inspirada na programação dinâmica neural de [Bertsekas e Tsitsiklis \(1996\)](#), operando três reservatórios no rio Piave na Itália, sem modelos de equivalência e com vazões aleatórias sem correlação. No caso do trabalho proposto, as redes neurais fazem o papel de generalizar uma função de custo que

é discretizada por uma malha grosseira (poucas discretizações). No entanto o trabalho de [Bertsekas e Tsitsiklis \(1996\)](#) é mais geral e cobre inclusive assuntos sobre aprendizagem por reforço e outras práticas com redes neurais.

Por outro lado, autores como [Powell \(2007\)](#) tem se dedicado a abordagens que exploram a frequência de visitação dos estados para diminuir tempo de execução e complexidade. Em particular esse trabalho é discutido com mais detalhes nos capítulos seguintes, já que reside nele a base do método proposto.

Mesmo com um método estabelecido como estado da arte (a PDDE) e somado com a diversificação de métodos heurísticos, o POR continua sendo um problema que gera investigações comparativas entre diversas abordagens. Dentre elas, se discute os benefícios da otimização estocástica explícita (PDE, por exemplo) e métodos que usem a solução determinística com algum recurso implícito para inserir aleatoriedades (regressões, redes neurais, etc.)([MARTINEZ; SOARES, 2002](#)),([ZAMBELLI et al., 2006](#)). Outros autores comparam heurísticas, como [Darlane e Moradi \(2016\)](#), no caso a utilização de redes neurais e de aprendizagem por reforço. Até mesmo abordagens tão estabelecidas, como representação por reservatórios equivalentes, são atualmente objetos de estudo ([BRANDÃO, 2010](#)).

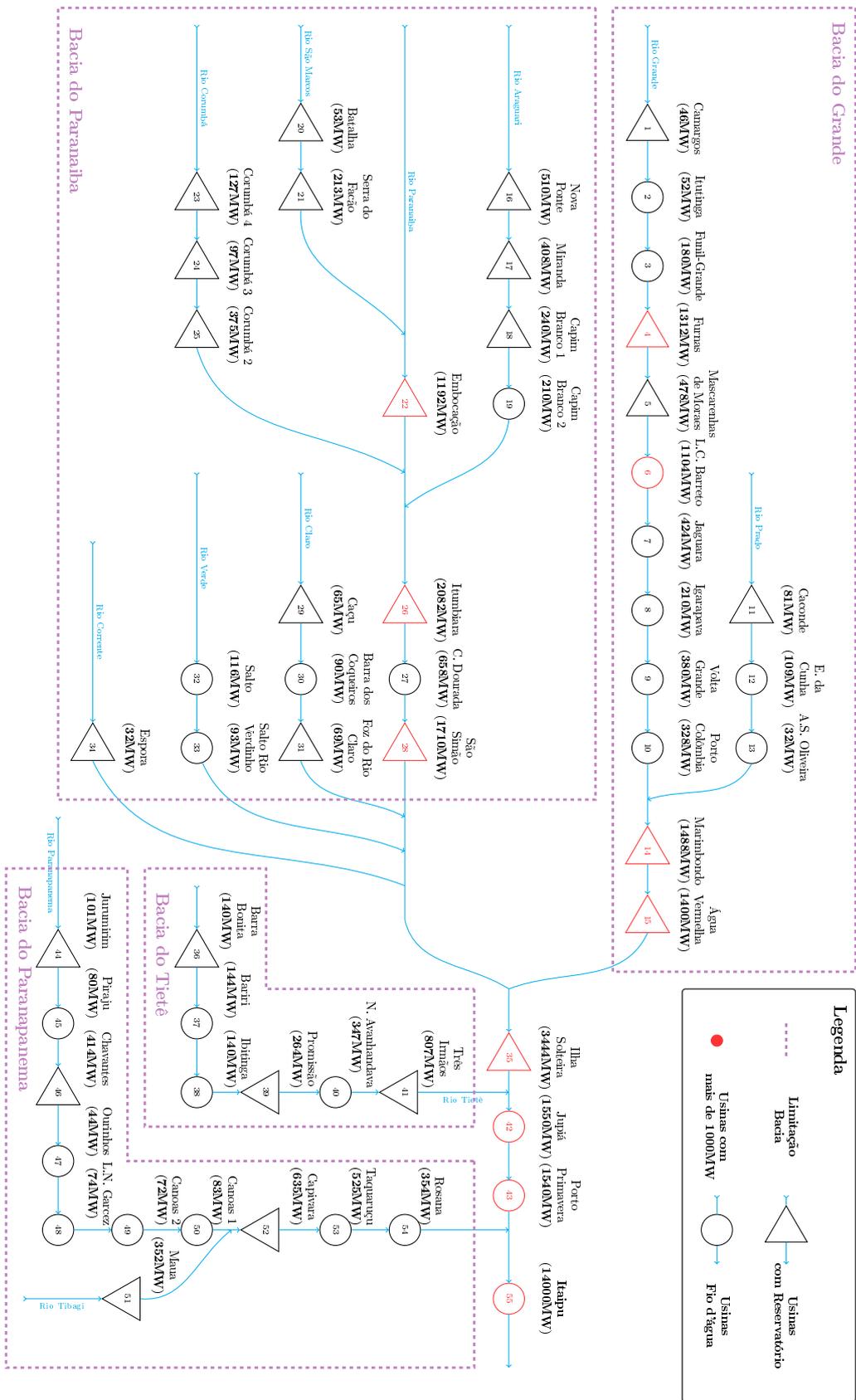


Figura 1.7 – Sistema sul-sudeste parcialmente representado.

## 2 Formulação do problema

*Neste capítulo é apresentado um modelo matemático para otimização do planejamento da operação de médio prazo de reservatórios hidrelétricos. Discute-se também os principais aspectos da estatística de vazões.*

### 2.1 Modelagem matemática do reservatório

Dentre as diversas possibilidades de formulação e objetivos a serem otimizados na operação de reservatórios, uma delas é a minimização da complementação energética para atender uma dada demanda mensal, em geral fixa e contratada durante um determinado período (um ano, por exemplo). Neste trabalho, a minimização procura reduzir gastos com o uso de energia térmica no atendimento de demandas energéticas mensais. O modelo apresentado é compatível com os objetivos do planejamento de médio prazo, procurando determinar políticas de operação individualizadas para reservatórios (LYRA FILHO, 1984).

Na operação de reservatórios as duas variáveis mais importantes para descrever o sistema são a capacidade, em volume, de armazenamento do reservatório ( $x[Mm^3]$ ) e o engolimento de sua turbina ( $u[m^3/s]$ ), vazão que atravessa a casa de máquinas da usina. Outra variável relevante no modelo é o vertimento do reservatório ( $v[m^3/s]$ ), que é a vazão que não passa pelas turbinas da unidade geradora, ou seja que não produz energia; recurso mais utilizado quando o reservatório se encontra no limite máximo de sua capacidade de armazenamento. Uma informação fundamental é a vazão afluyente que chega ao reservatório devido às características hidrográficas da área onde o reservatório está instalado, ( $y[m^3/s]$ ); em geral é uma variável tratada como aleatória. A Figura 2.1 ilustra as variáveis que descrevem o reservatório.

Devido à presença de aleatoriedades, é necessário que o objetivo do problema seja minimizar o valor esperado do custo ( $c_k$ ) com a complementação térmica ( $g_k$ ) da energia hidráulica ( $h_k$ ) em cada mês ( $k$ ) de maneira a atender as demandas mensais de energia ( $d_k$ ). Esses aspectos estão representados na Equação (2.1), onde o termo  $\psi(x_T)$  representa o valor da água armazenada no final do horizonte de planejamento.

O custo ( $c_k$ ) da geração termoelétrica ( $g_k$ ) em geral é modelado como uma função crescente e convexa (quadrática), como na Figura 2.2(a) (LYRA FILHO, 1984). Na Equação (2.2), tem-se que a geração térmica só é utilizada quando a hidrelétrica ( $h_k$ ) não consegue suprir a demanda ( $d_k$ ). Observa-se que a Equação (2.2) indica que não há benefício em gerar energia hidrelétrica a mais do que a demanda. No entanto se for possível vender o excesso para sistemas adjacentes, pode-se acrescentar um termo negativo na função objetivo, representando um lucro na geração. De forma análoga, a utilização de outras fontes de energia para cada intervalo de tempo, caso existam, ou a importação de energia,

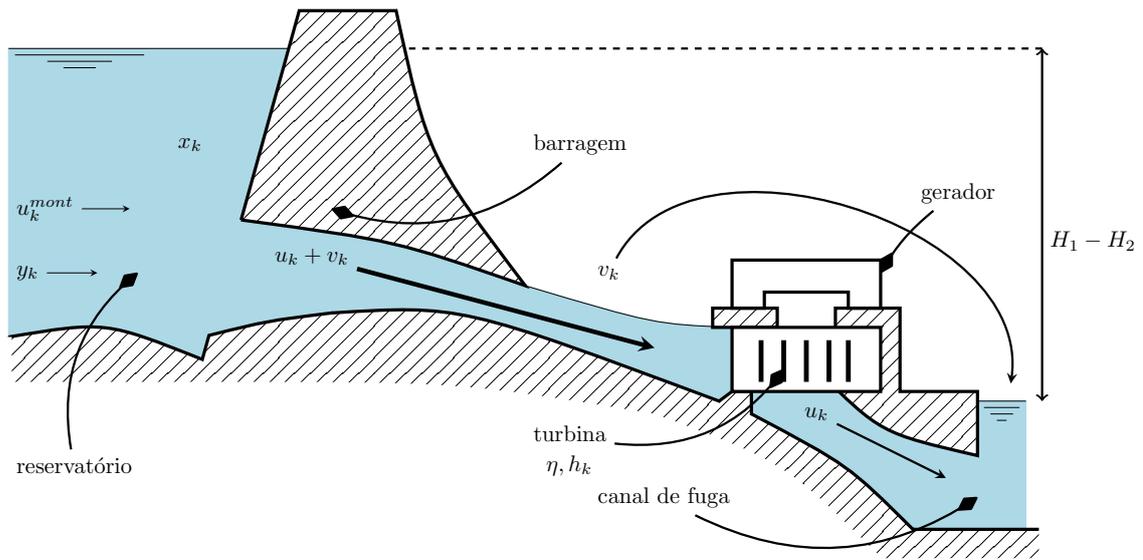
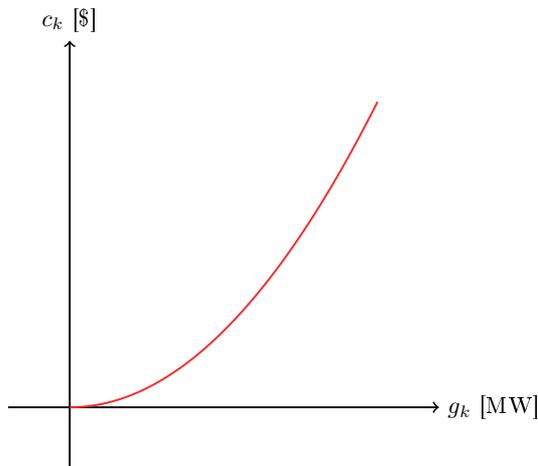
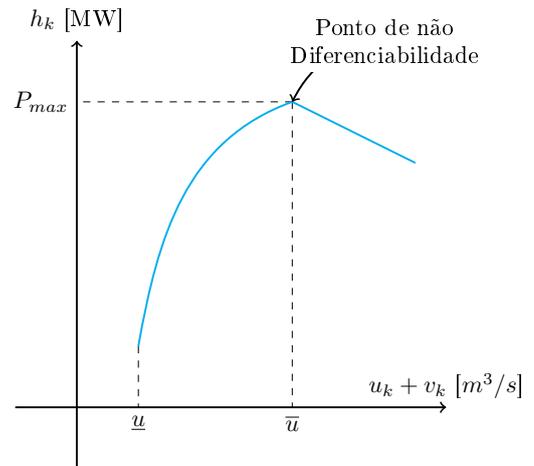


Figura 2.1 – Modelo de reservatório e a representação física das variáveis do sistema.

pode ser incluída na Equação (2.1).



(a) Função de custo de geração térmica.



(b) Geração hidráulica em função do volume defluído, com volume armazenado constante.

Figura 2.2 – Funções de custo de geração térmica (a esquerda) e de geração hidráulica em função do volume defluído (a direita).

A geração hidrelétrica no intervalo entre  $k$  e  $k + 1$  ( $h_k$ ), representada na Equação (2.3), é dependente da cota a montante ( $H_1$ ) e justante ( $H_2$ ) da barragem, que são, em geral, representadas por funções polinomiais do volume armazenado no início do intervalo entre os instantes  $k$  a  $k + 1$  ( $x_k$ ) e da vazão defluída ( $u_k + v_k$ ), respectivamente<sup>1</sup>, como mostra a Equação (2.6). Observa-se que a Equação (2.3) é consequência da definição física de trabalho e energia potencial.

<sup>1</sup> Alguns modelos preferem adotar  $H_1$  como função do valor médio entre  $x_k$  e  $x_{k+1}$

A geração hidrelétrica também depende de atributos específicos como o rendimento ( $\eta$ ) das turbinas na transformação da energia potencial para energia elétrica. Embora em algumas situações esse rendimento possa variar com a vazão turbinada, devido a atritos viscosos da água com os dutos e com as pás das turbinas, em modelos com discretizações mensais considera-se rendimentos constantes. Observa-se a Equação (2.3) também não leva em conta propriedades mecânicas da água, pois como mencionado na Seção 1.3, para discretizações mensais, o detalhamento de propriedades físicas como viscosidade, densidade variável e temperatura não é relevante.

Ainda sobre a geração hidráulica, a função de cota a jusante ( $H_2$ ) depende da defluência total do reservatório (vazão turbinada mais a vertida,  $u_k + v_k$ ), que pode ser considerada uma variável única. Porém o uso de uma única variável para representar a vazão defluída leva a pontos de não diferenciabilidade na função de geração (LYRA FILHO, 1984), Figura 2.2(b). Essas não diferenciabilidades precisam ser consideradas no processo de otimização (principalmente os baseados em derivadas).

$$\min \mathbb{E} \sum_{k=0}^{T-1} c_k = \min \mathbb{E} \sum_{k=0}^{T-1} \alpha_k g_k^2 + \psi(x_T) \quad (2.1)$$

$$g_k = \max \left( 0, d_k - h_k(x_k, u_k, y_k) \right) \quad (2.2)$$

$$h_k = \eta a_g \rho \left( H_1(x_k) - H_2(u_k + v_k(y_k)) \right) u_k \cdot 10^{-3} \quad (2.3)$$

$$\underline{x} \leq x_k \leq \bar{x} \quad (2.4)$$

$$\underline{u} \leq u_k \leq \bar{u} \quad (2.5)$$

$$H_k = \begin{cases} H_1(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 \\ H_2(u) = b_0 + b_1 u \end{cases} \quad (2.6)$$

$$\theta_k = x_k + \xi(y_k - u_k) \quad (2.7)$$

$$\xi v_k = \max(\theta_k - \bar{x}, 0) \quad (2.8)$$

$$x_{k+1} = \min(\theta_k, \bar{x}) \quad (2.9)$$

As restrições de desigualdades Equações (2.4) e (2.5) correspondem a limitações físicas para operação do sistema. As restrições no volume, por exemplo, refletem as condições de segurança da barragem contra colapsos (limite superior) e ao volume limite para que os dutos sejam capazes de aproveitar a água (limite inferior), denominado volume morto. As restrições de vazão refletem as limitações físicas da entrada de água da turbina; os limites inferiores podem garantir condições de irrigação da região, de navegabilidade a jusante e até mesmo de controle nos impactos ambientais.

Como mencionado anteriormente, a vazão turbinada ( $u_k$ ) é uma variável de controle, decisão tomada sobre o sistema. Da mesma forma, a vazão vertida ( $v_k$ ) também está

associada com ideia de escolha, no caso de quanto do volume é desperdiçado (não produz energia). No entanto, a Equação (2.8) define uma condição otimizante pelo qual o vertimento passa a não ser mais uma decisão; mas determinado pela vazão turbinada ( $u_k$ ) e o volume armazenado ( $x_k$ ). Ou seja a vazão vertida só é não-nula caso a capacidade de armazenamento máxima do reservatório seja alcançada<sup>2</sup>.

A Equação (2.7), conhecida como equação de transição de estado, é definida pela equação de continuidade do reservatório; ou seja, balanço da água que chega e sai do reservatório. É importante notar que o estado futuro,  $x_{k+1}$  é definido por essa equação caso a capacidade máxima do reservatório não seja alcançada, (2.9), caso contrário, o estado futuro é definido por  $\bar{x}$ . A variável  $\theta_k$  funciona como um estado “virtual”, usado somente para determinar os valores de vertimento, Equação (2.8), e o estado futuro, Equação (2.9).

As constantes do modelos são:  $a_g$ , a aceleração da gravidade local [ $m/s^2$ ];  $\rho$ , a densidade da água [ $kg/m^3$ ];  $\alpha_k$ , valor monetário da energia [ $\$/ (MW.mes)^2$ ];  $\xi$ , fator de conversão entre  $m^3/s$  para  $Mm^3$  (milhões de  $m^3$ ) para um intervalo de tempo;  $a_i$  e  $b_i$ , são constantes específicas para cada reservatório, que caracterizam os polinômios de cotas.

### 2.1.1 Generalização para múltiplos reservatórios

Para a modelagem com múltiplos reservatórios poucos detalhes adicionais são necessários. As topologias ilustradas na Figura 2.3 ajudam visualizar as modificações necessárias no modelo.

A inclusão de outros reservatórios implica na adição de novas variáveis de estado. Neste caso, o  $i$ -ésimo reservatório possui volume  $x_k^{(i)}$  e vazão de turbinagem  $u_k^{(i)}$ , ambos no início do intervalo entre os instantes (meses)  $k$  e  $k + 1$ . De maneira análoga as vazões incrementais,  $y_k^{(i)}$ , são associadas a cada um dos reservatórios.

Seja  $\mathcal{R}$  o conjunto de índices de todos os reservatórios do sistema e  $\mathcal{I}$  conjunto de índices de todos os reservatórios que deságuam na  $i$ -ésima usina. As Equações do modelo (2.2) e (2.7) são generalizadas nas Equações (2.10-2.11).

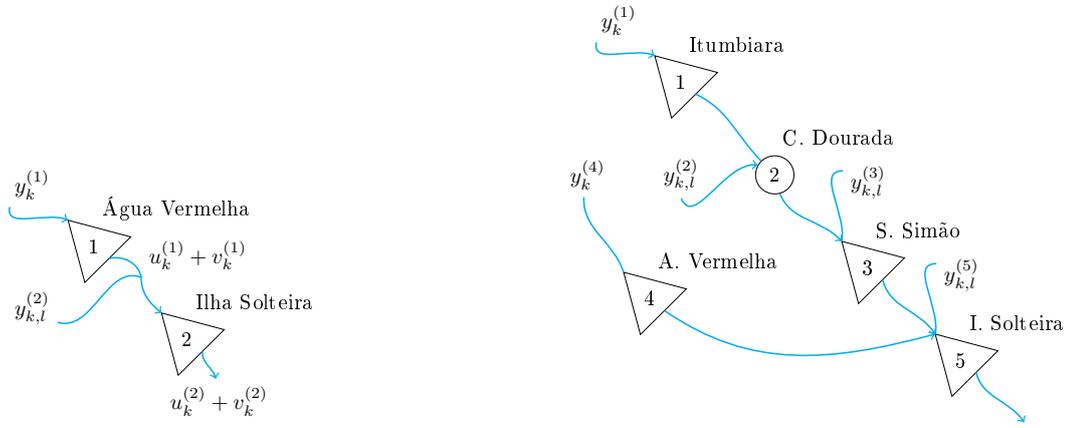
Na Equação (2.11) a variável  $y_{l,k}$  representa a vazão lateral do reservatório, definida na Seção 2.2. O estado do sistema completo<sup>3</sup> é  $\mathbf{x}_k = \left( x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(N)} \right)^T$ ; o vetor controles do sistema é  $\mathbf{u}_k = \left( u_k^{(1)}, u_k^{(2)}, \dots, u_k^{(N)} \right)^T$ . Em sistemas com múltiplas usinas é necessário percorrer a topologia de montante até a última usina a jusante para se definir cada geração ( $h_k^{(j)}$ ) e cada vazão defluída ( $v_k^{(i)} + u_k^{(i)}$ ), informação relevante para usinas rio abaixo.

<sup>2</sup> A afirmação pode não ser verdadeira em situações particulares de sistemas com múltiplos usinas acopladas.

<sup>3</sup> Utilizado a notação matricial em que T denota transposto.

$$g_k = \max \left( 0, d_k - \sum_{j \in \mathcal{R}} h_k^{(j)}(x_k^{(j)}, u_k^{(j)}, y_k^{(j)}) \right) \quad (2.10)$$

$$\theta_k^{(i)} = x_k^{(i)} + \xi \left( \underbrace{y_{l,k}^{(i)} + \sum_{j \in \mathcal{I}} (u_k^{(j)} + v_k^{(j)})}_{\text{Vazão afluyente na } i\text{-ésima usina}} - \underbrace{u_k^{(i)} - v_k^{(i)}}_{\text{Defluência da } i\text{-ésima usina}} \right) \quad (2.11)$$



(a) Topologia para 2 reservatórios.

(b) Topologia para cinco usinas hidrelétricas (quatro reservatórios e uma fio d'água.).

Figura 2.3 – Topologias reais de reservatórios.

Além disso, para múltiplos reservatórios é necessário indicar se a usina é do tipo com reservatório ou a fio d'água. Para as usinas a fio d'água não existe volume armazenado. Nesses casos a transição de estados é representada pela Equação (2.12). A Equação (2.12) implica que a Equação (2.11) define a defluência da usina fio d'água, equação (2.13), e ela ainda sinaliza que esse tipo de usina deflui tudo o que chega até ela.

$$x_{k+1}^{(i)} = x_k^{(i)} = 0 \quad (2.12)$$

$$0 = 0 + \xi \left( y_k^{(i)} - u_k^{(i)} - v_k^{(i)} + \sum_{j \in \mathcal{I}} (u_k^{(j)} + v_k^{(j)}) \right)$$

$$u_k^{(i)} + v_k^{(i)} = y_{l,k}^{(i)} + \sum_{j \in \mathcal{I}} (u_k^{(j)} + v_k^{(j)}) \quad (2.13)$$

## 2.2 Modelagem estatística das vazões

As vazões afluentes dos reservatórios são classificadas em dois tipos: vazões controláveis e vazões incrementais. A primeira delas corresponde a vazão total correspondente a turbinagem e vertimentos das usinas imediatamente à montante. A segunda vazão incremental, inclui apenas o termo de contribuição da área de drenagem da bacia, também conhecida como contribuição lateral (FORTUNATO et al., 1990).

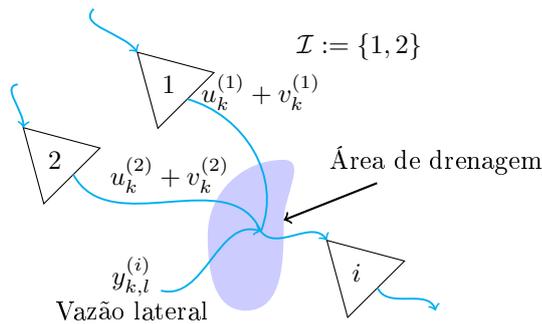


Figura 2.4 – Ilustração dos tipos de vazões afluentes.

Do ponto de vista técnico, apenas vazões totais são medidas; com esta informação, podem ser obtidas pelo balanço hídrico no sistema. A Equação (2.14) indica a contribuição da vazão afluente da usina  $i$ . Reforçando que a notação  $y_k^{(i)}$  indica a vazão afluente que chega a  $i$ -ésima usina entre os intervalo de tempo  $k$  e  $k + 1$ .

$$y_k^{(i)} = \underbrace{y_{k,l}^{(i)} + \sum_{j \in \mathcal{I}} (v_k^{(j)} + u_k^{(j)})}_{\text{vazão total}} \quad (2.14)$$

Naturalmente, o conhecimento da distribuição estatística das vazões é fundamental para o POR. Neste trabalho, impõe-se uma distribuição estatística bem definida de uma curva log-normal, que apresenta uma boa adesão aos dados numéricos do banco de dados da ANA (2018). Esta hipótese descreve os dados de vazões de forma adequada e permite o uso de um grande repertório de conhecimentos para lidar com o aspecto estocástico do problema.

### 2.2.1 Vazões independentes

Antes de iniciar a discussão sobre a modelagem das vazões independentes, é interessante discutir algumas propriedades úteis sobre duas distribuições de probabilidades

contínuas bastante conhecidas: a log-normal e a normal.

Uma variável aleatória contínua<sup>4</sup>  $Y$  é definida como uma função do espaço amostral (espaço de todos os resultados possíveis de um experimento) em um número real,  $Y : \Omega \rightarrow \mathbb{R}$ . Um caso especial de variável aleatória é aquele em que uma variável contínua  $Y$  possui uma função de densidade de probabilidade que pode ser parametrizada. Ou seja, pode-se determinar a probabilidade da variável  $Y$  estar contida em um conjunto  $\mathcal{B}$  através de uma função  $l$ , equação (2.15).

$$\mathbb{P}(Y \in \mathcal{B}) = \int_{\mathcal{B}} l(y) dz \quad (2.15)$$

A função de densidade permite que seja definida a função densidade acumulada de probabilidade<sup>5</sup>, Equação (2.16) (ROSS, 1998). A função  $L : \mathbb{R} \rightarrow [0; 1]$  define a probabilidade da variável aleatória contínua  $Y$  estar em um intervalo  $]-\infty, y]$ . É frequente também denotar  $(\omega \in \Omega | Y(\omega) < y)$  como o domínio de  $\mathbb{P}(\cdot)$ .

$$L(y) = \mathbb{P}(Y < y) = \int_{-\infty}^y l(y') dy' \quad (2.16)$$

A função de densidade de probabilidade log-normal é caracterizada pela Equação (2.17) (CROW; SHIMIZU, 1988). A distribuição log-normal possui parâmetros  $\mu$  e  $\sigma$ ; denota-se que  $Y$  segue essa distribuição com parâmetros  $\mu$  e  $\sigma$  como  $Y \sim \ln N(\mu, \sigma)$ .

$$l(y) := \frac{1}{y} \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{\ln(y) - \mu}{\sigma}\right)^2\right) \quad (2.17)$$

O teorema 2.1 estabelece a relação entre duas distribuições distintas (CROW; SHIMIZU, 1988).

**Teorema 2.1.** *Seja  $Y$  uma variável aleatória com distribuição contínua em um intervalo  $S \subseteq \mathbb{R}$  com função densidade de probabilidade  $l$  e função probabilidade acumulada  $L$ . Supondo uma transformação  $\phi$  tal que seja possível definir uma nova variável aleatória  $Z = \phi(Y)$ , exigindo que  $\phi : S \rightarrow T$  seja diferenciável, inversível e estritamente crescente. Então vale para as funções  $G(z)$  e  $g(z)$ :*

1.  $G(z) = L[\phi^{-1}(z)]$
2.  $g(z) = l[\phi^{-1}(z)] \frac{d}{dz} \phi^{-1}(z)$

*Demonstração.*

<sup>4</sup> Será adotada a notação comum em estatística: letras maiúscula para denotar a variável aleatória e minúscula para uma realização (valor) dessa variável.

<sup>5</sup> Mais uma vez, a notação é a convencional de estatística: para função densidade de probabilidade é utilizada letras minúsculas e para função densidade acumulada, letras maiúsculas.

$$1. G(z) \stackrel{\text{eq. (2.16)}}{=} \mathbb{P}(Z < z) = \mathbb{P}(\phi(Y) < z) \stackrel{\text{crescente}}{=} \mathbb{P}(Y < \phi^{-1}(z)) = L[\phi^{-1}(z)]$$

Para  $y \in S$  e  $z \in T$ .

2. Da equação (2.15) é possível afirmar que  $G'=g$ . Assim, derivando o resultado do item anterior.

$$g(z) = \frac{d}{dz}G(z) = \frac{d}{dz} \left\{ L[\phi^{-1}(z)] \right\} \stackrel{\text{regra da cadeia}}{=} l[\phi^{-1}(z)] \frac{d}{dz} \phi^{-1}(z)$$

□

No caso da distribuição log-normal,  $y \in S = ]0, \infty[$  e definindo  $z = \phi(y) = \ln(y)$ , que é diferenciável, inversível e estritamente crescente  $\phi : S \rightarrow \mathbb{R}$ . Então aplicando o teorema (2.1), segue a equação (2.18), que é a definição da distribuição de probabilidade normal.

$$g(z) = l(\exp(z)) \frac{d}{dz} \left\{ \exp(z) \right\} = \frac{1}{\exp(z)} \frac{1}{\sigma\sqrt{2\pi}} \exp \left( -\frac{1}{2} \left( \frac{\ln(\exp(z)) - \mu}{\sigma} \right)^2 \right) \exp(z)$$

$$g(z) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left( -\frac{1}{2} \left( \frac{z - \mu}{\sigma} \right)^2 \right) \quad (2.18)$$

Se  $Y$  for a variável aleatória que corresponde as vazões afluentes aos reservatórios caracterizada por uma distribuição log-normal, Equação (2.17), então  $Z = \ln(Y)$  define uma variável aleatória normal,  $Z \sim N(\mu, \sigma^2)$ , ilustrada na Figura 2.5.

Os dois parâmetros dessas distribuições,  $\mu$  e  $\sigma$ , podem ser definidos e relacionados pelas Propriedades (2.1) e (2.2) (CROW; SHIMIZU, 1988).

**Propriedade 2.1. Valor esperado:** O valor esperado  $\mathbb{E}(\cdot)$ , ou esperança matemática, corresponde a um indicador do valor médio obtido se um experimento (ou variável) for repetido um número suficientemente grande de vezes, segundo a Lei dos grandes números. No caso da distribuição normal:

$$\mathbb{E}(Z) = \mu \quad (2.19)$$

O parâmetro  $\mu$  é denominado de média gaussiana.

Para a distribuição log-normal:

$$\mathbb{E}(Y) = \exp\left(\mu + \frac{1}{2}\sigma^2\right) \quad (2.20)$$

**Propriedade 2.2. Variância:** A variância  $Var(\cdot)$  sinaliza o quanto a variável aleatória, ou realizações de um experimento, se distribuem entorno de seu valor esperado. Para a distribuição normal:

$$Var(Z) = \sigma^2 \quad (2.21)$$

O parâmetro  $\sigma^2$  é denominado de variância gaussiana. Para a distribuição log-normal:

$$\text{Var}(Y) = \exp(2\mu + \sigma^2)(\exp(\sigma^2) - 1) \quad (2.22)$$

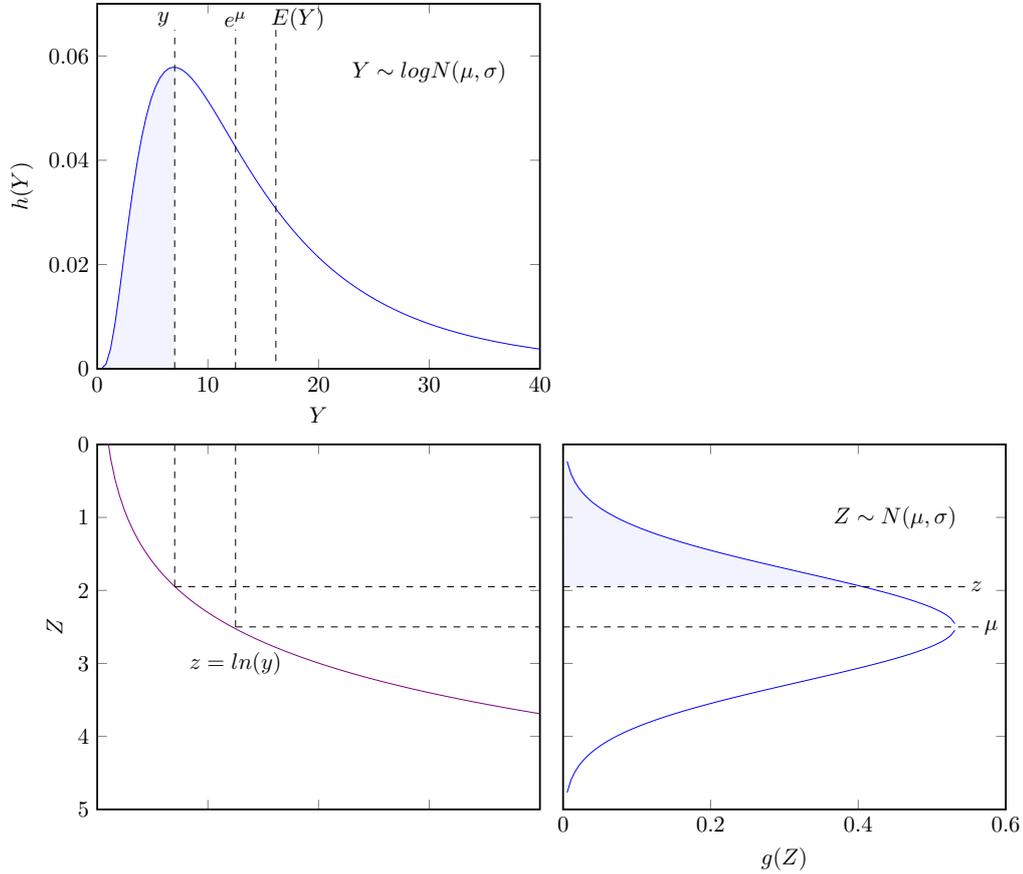


Figura 2.5 – Ilustração da relação entre a densidade de distribuição log-normal e normal.

Sobre as vazões afluentes, seja  $Y_k$  a variável aleatória da vazão afluyente (em um sistema de apenas um reservatório), então a amostragem de um cenário possível, para um horizonte de planejamento de  $T$  intervalos de tempos (meses, por exemplo) é denotado como  $\tilde{\Omega} := \{\omega_1, \omega_2, \dots, \omega_N\}$ , onde  $\omega_k$  é uma amostragem de vazão.

**Definição 2.1. Variáveis aleatórias independentes:** Seja um  $Y_k : \Omega_k \rightarrow \mathbb{R}$  uma variável aleatória, então as variáveis  $(Y_1, \dots, Y_N)$  são ditas independentes se para quaisquer conjuntos reais  $\mathcal{B}_1, \dots, \mathcal{B}_T$  com  $Y_k \in \mathcal{B}_k, \forall k = 1, 2, \dots, N$  tem-se que:

$$\mathbb{P}(Y_1 \in \mathcal{B}_1, \dots, Y_N \in \mathcal{B}_N) = \prod_{k=1}^N \mathbb{P}(Y_k \in \mathcal{B}_k) \quad (2.23)$$

Em outras palavras, vazões independentes implicam que a realização da vazão  $\omega_k$  não altera a densidade de probabilidade de nenhuma outra vazão do cenário  $\tilde{\Omega}$ , o que é evidenciado na Equação (2.23) pelo produto de probabilidades não condicionais.

### 2.2.2 Vazões markovianas

Para o escopo desse trabalho será necessário utilizar um conjunto de propriedades e definições de cadeia de Markov. A apresentação dessas propriedades é baseada no capítulo introdutório de (STROOCK, 2000).

**Definição 2.2. Cadeias de Markov em tempo discreto de ordem  $m$ :** *Seja as variáveis aleatórias  $Y_k : \Omega_k \rightarrow \mathbb{R}$ , então a sequência  $\{Y_1, Y_2, \dots, Y_N\}$  de variáveis aleatórias, é dito que formam uma Cadeia de Markov em tempo discreto de memória  $m$  se a probabilidade de um evento futuro só depender dos  $m$  instantes anteriores. Em outras palavras, a probabilidade de um evento futuro no instante  $k + 1$  só depende dos  $k - m$  instantes anteriores, com  $m < k$ .*

$$\mathbb{P}(Y_{k+1} = y | Y_1 = y_1, Y_2 = y_2, \dots, Y_k = y_k) = \mathbb{P}(Y_{k+1} = y | Y_k = y_k, \dots, Y_{k-m+1} = y_{k-m+1})$$

No caso particular de ordem um,  $m = 1$ , ou seja a probabilidade do evento futuro ( $k + 1$ ) só depende do presente ( $k$ ), o que é comumente chamado de um processo “sem memória”.

$$\mathbb{P}(Y_{k+1} = y | Y_1 = y_1, Y_2 = y_2, \dots, Y_k = y_k) = \mathbb{P}(Y_{k+1} = y | Y_k = y_k) \quad (2.24)$$

Lê-se  $\mathbb{P}(Y_{k+1} = y | Y_k = y_k)$  como a probabilidade da ocorrência de  $Y_{k+1}$  no intervalo  $] - \infty, y[$  dado a realização de  $Y_k = y_k$ .

Do ponto de vista das vazões afluentes dos reservatórios, a Equação (2.24) diz que a vazão no próximo mês ( $k + 1$ ) só depende da vazão medida no mês atual ( $k$ ). É interessante notar a diferença dessa modelagem para a representação por vazões independentes. Para as vazões da subseção 2.2.1 é possível encontrar cenários onde dois meses consecutivos possuem vazões completamente diferentes (um mês muito “seco” seguido de outro muita água). Assim do ponto de vista de um processo próximo do real, as cadeias de Markov parecem sugerir um caminho correto para a modelagem das vazões (um mês seco tende a ser seguido por outro seco, compondo um ano de escassez). Ambos os modelos são aplicados no Capítulo 5, cada um demonstrando sua relevância para os objetivos deste trabalho.

Uma das maneiras se estabelecer uma cadeia de Markov, é através a distribuição normal multivariada (generalização da Equação (2.18))(MARTIN; BRENNER, 1999). Por isso a importância das discussões da Subseção 2.2.1, dando fundamento para utilização dessa distribuição.

**Definição 2.3. Densidade de probabilidade multivariada normal:** *Seja  $\mathbf{Z} = (Z_1, Z_2, \dots, Z_N)^T$  um vetor de variáveis aleatórias, então é possível definir:*

1. Vetor de médias  $\mu$ :

$$\mu = \mathbb{E}(\mathbf{Z}) = \left( \mathbb{E}(Z_1), \dots, \mathbb{E}(Z_N) \right)^T \in \mathbb{R}^N$$

2. Matriz de covariância  $\Sigma$ :

$$\Sigma = (\sigma_{ij}) \in \mathbb{R}^{N \times N} \implies \sigma_{ij} := \mathbb{E}\left((Z_i - \mathbb{E}(Z_i))(Z_j - \mathbb{E}(Z_j))\right) = \text{Cov}(Z_i, Z_j)$$

Se a matriz  $\Sigma$  é semi-positiva definida<sup>6</sup>, então:

$$g(\mathbf{Z}) = \frac{1}{\sqrt{(2\pi)^T |\Sigma|}} \exp\left\{-\frac{1}{2}(\mathbf{Z} - \mu)^T \Sigma^{-1}(\mathbf{Z} - \mu)\right\} \quad (2.25)$$

Onde  $|\Sigma|$  é o determinante e  $\Sigma^{-1}$  a matriz inversa de  $\Sigma$ , respectivamente.

O caso particular, e de principal interesse aqui é quando  $n = 2$ , definindo a distribuição bivariada. Para o caso de duas variáveis é interessante definir o coeficiente de correlação, Definição 2.4 (ROSS, 1998)

**Definição 2.4. Coeficiente de correlação:** O coeficiente de correlação entre duas variáveis  $Z_1$  e  $Z_2$ , com variâncias  $\text{Var}(Z_1)$  e  $\text{Var}(Z_2)$ , e com covariância  $\text{Cov}(Z_1, Z_2)$ , é definido por:

$$\rho := \frac{\text{Cov}(Z_1, Z_2)}{\sqrt{\text{Var}(Z_1)\text{Var}(Z_2)}}. \quad (2.26)$$

O coeficiente de correlação  $\rho \in [-1, 1]$  e indica o quão próximo duas variáveis se relacionam de forma linear, quanto mais próximo de 1 e -1 mais forte é essa relação.

Usando a definição de matriz de covariância com a definição de coeficiente de correlação, é possível escrever as propriedades expressas nas Equações (2.27-2.28).

$$\begin{aligned} \Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} &\implies |\Sigma| = \sigma_{11}\sigma_{22} - \sigma_{12}\sigma_{21} \stackrel{\sigma_{12} \equiv \sigma_{21}}{=} \underbrace{\sigma_{11}\sigma_{22}}_{\text{Var}(Z_1)\text{Var}(Z_2)} - \underbrace{\sigma_{12}^2}_{\text{Cov}(z_1, Z_2)^2} \\ &\stackrel{\text{eq. (2.26)}}{=} \sigma_{11}\sigma_{22}(1 - \rho^2) \end{aligned} \quad (2.27)$$

$$\implies \Sigma^{-1} = \frac{1}{\sigma_{11}\sigma_{22}(1 - \rho^2)} \begin{bmatrix} \sigma_{22} & -\sigma_{21} \\ -\sigma_{12} & \sigma_{11} \end{bmatrix} \quad (2.28)$$

Com as Equações (2.27-2.28) é possível explicitar a expressão  $(\mathbf{Z} - \mu)^T \Sigma^{-1}(\mathbf{Z} - \mu)$ .

$$(\mathbf{Z} - \mu)^T \Sigma^{-1}(\mathbf{Z} - \mu) = (\mathbf{Z} - \mu)^T \frac{1}{\sigma_{11}\sigma_{22}(1 - \rho^2)} \begin{bmatrix} \sigma_{22} & -\sigma_{21} \\ -\sigma_{12} & \sigma_{11} \end{bmatrix} \begin{bmatrix} z_1 - \mu_1 \\ z_2 - \mu_2 \end{bmatrix}$$

<sup>6</sup>  $\Sigma$  é uma matriz  $N \times N$  real por definição. E uma matriz é dita semi-positiva definida se  $\forall \mathbf{x} \in \mathbb{R}^N$  valer que  $\mathbf{x}^T \Sigma \mathbf{x} > 0$ . No caso de matrizes Hermitianas, se todos os autovalores forem não negativos é suficiente para que a matriz seja semi-positiva definida. No caso de matrizes reais, basta que sejam simétricas para serem Hermitianas, como é  $\Sigma$ , por definição.

$$\begin{aligned}
&= \frac{1}{\sigma_{11}\sigma_{22}(1-\rho^2)} \begin{bmatrix} z_1 - \mu_1 & z_2 - \mu_2 \end{bmatrix} \begin{bmatrix} \sigma_{22}(z_1 - \mu_1) - \sigma_{21}(z_2 - \mu_2) \\ \sigma_{11}(z_2 - \mu_2) - \sigma_{12}(z_1 - \mu_1) \end{bmatrix} \\
&\stackrel{\sigma_{12}=\sigma_{21}}{=} \frac{1}{(1-\rho^2)} \left( \frac{(z_1 - \mu_1)^2}{\sigma_{11}} + \frac{(z_2 - \mu_2)^2}{\sigma_{22}} - 2\frac{\sigma_{12}}{\sigma_{11}\sigma_{22}}(z_1 - \mu_1)(z_2 - \mu_2) \right) \\
&\stackrel{\sigma_{ii}=\sigma_i^2}{=} \frac{1}{(1-\rho^2)} \left( \left( \frac{z_1 - \mu_1}{\sigma_1} \right)^2 + \left( \frac{z_2 - \mu_2}{\sigma_2} \right)^2 - 2\rho \left( \frac{z_1 - \mu_1}{\sigma_1} \right) \left( \frac{z_2 - \mu_2}{\sigma_2} \right) \right) \quad (2.29)
\end{aligned}$$

Usando a Equação (2.29), é possível escrever a densidade de probabilidade bivariada, Definição (2.5) (MARTIN; BRENNER, 1999).

**Definição 2.5. Densidade de probabilidade bivariada:** Dadas duas variáveis aleatórias  $\mathbf{Z} = (Z_k, Z_{k+1})^T$  com distribuições normais  $N(\mu_k, \sigma_k^2)$  e  $N(\mu_{k+1}, \sigma_{k+1}^2)$  (distribuições marginais), então o vetor de variáveis aleatórias  $\mathbf{Z}$  segue uma distribuição normal bivariada com densidade de probabilidade dada por:

$$\begin{aligned}
g(z_k, z_{k+1}) &= \frac{1}{2\pi\sigma_k\sigma_{k+1}} \frac{1}{\sqrt{1-\rho_{k,k+1}^2}} \exp -\frac{1}{2} \frac{1}{(1-\rho_{k,k+1}^2)} \left\{ \left( \frac{z_k - \mu_k}{\sigma_k} \right)^2 + \left( \frac{z_{k+1} - \mu_{k+1}}{\sigma_{k+1}} \right)^2 \right. \\
&\quad \left. - 2\rho_{k,k+1} \left( \frac{z_k - \mu_k}{\sigma_k} \right) \left( \frac{z_{k+1} - \mu_{k+1}}{\sigma_{k+1}} \right) \right\} \quad (2.30)
\end{aligned}$$

Da Equação (2.30), pode-se determinar a densidade de probabilidade condicional.

**Definição 2.6. Probabilidade condicional:** Sejam duas variáveis aleatórias  $Z_{k+1}$  e  $Z_k$ . Define-se a probabilidade condicional, ou seja a probabilidade de ocorrência de um evento de  $Z_{k+1}$ , dado a realização de  $Z_k = z_k$ , como:

$$g(Z_{k+1} = z | Z_k = z_k) = \frac{g(z, z_k)}{g_k(z_k)} \quad (2.31)$$

Onde  $g(z, z_k)$  é a densidade conjunta de probabilidade e  $g_k(z_k)$  a distribuição marginal de  $Z_k$ .

Usando a Definição 2.6 e a distribuição bivariada:

$$\begin{aligned}
g(z_{k+1} | Z_k = z_k) &= \frac{g(z_{k+1}, z_k)}{g_Z(z_k)} = \frac{1}{\sqrt{2\pi}\sigma_{k+1}} \frac{1}{\sqrt{1-\rho_{k,k+1}^2}} \exp -\frac{1}{2} \left\{ \right. \\
&\quad \left. \frac{1}{(1-\rho_{k,k+1}^2)} \left\{ \rho_{k,k+1}^2 \left( \frac{z_k - \mu_k}{\sigma_k} \right)^2 + \left( \frac{z_{k+1} - \mu_{k+1}}{\sigma_{k+1}} \right)^2 - 2\rho_{k,k+1} \left( \frac{z_k - \mu_k}{\sigma_k} \right) \left( \frac{z_{k+1} - \mu_{k+1}}{\sigma_{k+1}} \right) \right\} \right\} \quad (*)
\end{aligned}$$

$$\begin{aligned}
(*) &= \left\{ \rho_{k,k+1}^2 \left( \frac{z_k - \mu_k}{\sigma_k} \right)^2 + \left( \frac{z_{k+1} - \mu_{k+1}}{\sigma_{k+1}} \right)^2 - 2\rho_{k,k+1} \left( \frac{z_k - \mu_k}{\sigma_k} \right) \left( \frac{z_{k+1} - \mu_{k+1}}{\sigma_{k+1}} \right) \right\} \\
&= \left\{ \frac{z_{k+1} - \mu_{k+1}}{\sigma_{k+1}} - \rho_{k,k+1} \frac{z_k - \mu_k}{\sigma_k} \right\}^2 = \frac{1}{\sigma_{k+1}^2} \left\{ z_{k+1} - \left( \mu_{k+1} + \rho_{k,k+1} \frac{\sigma_{k+1}}{\sigma_k} (z_k - \mu_k) \right) \right\}^2
\end{aligned}$$

$$\begin{aligned}
g(z_{k+1}|Z_k = z_k) &= \\
&= \frac{1}{\sqrt{2\pi}\sigma_{k+1}} \frac{1}{\sqrt{1 - \rho_{k,k+1}^2}} \exp - \frac{1}{2} \underbrace{\frac{1}{\sigma_{k+1}^2(1 - \rho_{k,k+1}^2)}}_{\text{Desvio padrão}} \left\{ z_{k+1} - \underbrace{\left( \mu_{k+1} + \rho_{k,k+1} \frac{\sigma_{k+1}}{\sigma_k} (z_k - \mu_k) \right)}_{\text{Média}} \right\}^2
\end{aligned}$$

Ou seja,  $(Z_{k+1}|Z_k = z_k) \sim N(\mu_{k+1|k}, \sigma_{k+1|k}^2)$ , onde o desvio padrão e a média condicional são dadas pelas Equações (2.32) e (2.33).

$$\mu_{k+1,k} = \mu_{k+1} + \rho_{k,k+1} \frac{\sigma_{k+1}}{\sigma_k} (z_k - \mu_k) \quad (2.32)$$

$$\sigma_{k+1|k}^2 = \sigma_{k+1}^2 (1 - \rho_{k,k+1}^2) \quad (2.33)$$

É interessante notar que a média condicional  $\mu_{k+1,k}$  é linear com a realização da variável  $z_k$ , ou seja essa última altera as probabilidades da realização futura de  $Z_{k+1}$ .

Voltando para o contexto das vazões, as duas últimas equações mostram que a densidade de probabilidade da vazão futura ( $k+1$ ) tem sua forma modificada em função da vazão presente ( $k$ ), estabelecendo uma forma de se gerar cenários de vazões markovianas de ordem 1. Nota-se que as definições de cadeias de Markov, de densidade de probabilidade multivariada e de probabilidade condicional, Definições 2.2, 2.3 e 2.6, permitem descrever qualquer ordem de processo markoviano.

### 3 Programação Dinâmica

*O método de otimização para problemas de operação de reservatórios hidrelétricos utilizado nesse trabalho é a programação dinâmica, que é apresentada nesse capítulo em sua versão de referência e na versão denominada “programação aproximada”. São discutidos os principais pontos do método; incluindo algoritmos, vantagens e limitações, procurando destacar sua aplicação ao POR. De forma mais específica, mostra-se como a programação dinâmica aproximada (PDA), alicerce desse trabalho, permite contornar as limitações do método convencional de programação dinâmica estocástica (PDE).*

A programação dinâmica (PD) é uma técnica de otimização para problemas de tomada de decisões sequenciais, entre os quais se destacam os problemas dinâmicos de otimização. Um dos principais nomes do desenvolvimento e formulação da técnica, Richard Bellman, atribuiu a denominação “dinâmica” por se tratar de um método de otimização em que o tempo tem grande importância, além de se tratar de problemas em que as decisões são tomadas sequencialmente (BELLMAN, 1957).

Os principais componentes que caracterizam os problemas de otimização dinâmicos abordados por PD são: estágios, estados, controles, informação aleatória, equação de transição de estados e função objetivo. Esses conceitos são resumidos a seguir.

- **Estágio** ( $k$ ) representam instantes discretos de tempo ou etapas onde uma decisão é escolhida ( $k \in \mathbb{N}$ ).
- **Variáveis de estados** ( $\mathbf{x}_k \in \mathbb{R}^n$ ) sintetizam as informações sobre a configuração do sistema. Por exemplo, no POR podem representar os volumes d’água armazenados nos reservatórios.
- **Variáveis de controle** ( $\mathbf{u}_k \in \mathbb{R}^m$ ) são ações, ou decisões tomadas sobre o sistema e que modificam seus estados.
- **Informações/variáveis externas aleatórias** ( $\mathbf{Y}_k \in \mathbb{R}^q$ ), são variáveis que afetam a configuração do sistema de forma estocástica.

A transição de estado é caracterizada pela **função de transição de estado**, que descreve como o sistema evolui no tempo a partir de um determinado estado  $\mathbf{x}_k$ , no

estágio  $k$ , quando for tomada uma decisão  $\mathbf{u}_k$ , sob uma determinada realização da variável aleatória  $\mathbf{Y}_k$ , Equação (3.1)<sup>1</sup>.

$$\mathbf{X}_{k+1} := \varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{Y}_k) \quad (3.1)$$

O objetivo da programação dinâmica é, dadas as limitações do sistema; as restrições de estado ( $\mathbf{x}_k \in \mathcal{X}_k$ ) e as restrições de controle ( $\mathbf{u}_k \in \mathcal{U}_k$ ) em cada estágio  $k$  ( $k = 1, 2, \dots, T - 1$ )<sup>2</sup>, as condições iniciais ( $\mathbf{x}_0$ ) e finais ( $\mathbf{x}_T$ ), encontrar funções (**políticas de operação**)  $\pi_k(\mathbf{x}_k) : \mathbb{R}^n \rightarrow \mathbb{R}^m$  que minimizem uma função objetivo  $C_k^{(\pi)}$ . A notação com índice  $\pi$  tem o sentido de reforçar a ideia de que a função objetivo é dependente da regra/política de operação. Ou seja, tem-se o problema:

$$C_k^* = \min_{\pi \in \Pi} C_k^{(\pi)} \quad (3.2)$$

A notação  $*$  indica valor ótimo e  $\pi := \left\{ \pi_k(\mathbf{x}_k) \right\} = \left\{ \pi_0(\mathbf{x}_0), \dots, \pi_{T-1}(\mathbf{x}_{T-1}) \right\}$  uma política de operação, onde  $\Pi$  representa o conjunto de todas as políticas factíveis considerando as restrições do problema.

Antes de discutir as diferentes formulações em PD é necessário definir o princípio fundamental por trás desses algoritmos, que pode ser segundo Bellman definido como:

**Definição 3.1. Princípio da otimalidade de Bellman:** “Uma política dita ótima tem a propriedade que, não importe qual o estado inicial e a primeira decisão escolhida, as decisões restantes até o fim do horizonte devem constituir uma política ótima em relação ao estado resultante da decisão inicial.”, (BELLMAN, 1957).

Os autores Larson (1968) e Bertsekas (1976) fazem uma interpretação similar desse princípio. Seja dada uma trajetória gerada por uma política ótima (AC), então para um ponto intermediário (B), a trajetória até o ponto final (C) deve ser o seguimento da trajetória ótima BC, como mostra a Figura 3.1 - adaptada de Larson (1968). O argumento dos autores é simples e por contradição: Seja a trajetória (I)-(II) ótima. Por hipótese, para o ponto intermediário B, seja (III), a trajetória ótima até o final C. Ou seja, com isso a trajetória (I)-(III) tem um custo (função objetivo) menor que (I)-(II), já que (I) é comum a ambos, o que contradiz a hipótese de que (I)-(II) é ótima.

<sup>1</sup> O caso geral, é que  $\mathbf{Y}_k$  seja aleatória e  $\mathbf{x}_k$  e  $\mathbf{u}_k$  determinísticos, o primeiro conhecido e o segundo escolhido. Nesse caso,  $\mathbf{X}_{k+1}$  é uma variável aleatória, por isso a notação maiúscula para variáveis aleatórias e minúscula para uma realização dessa variável.

<sup>2</sup> Em algumas situações  $k$  não é limitado, isso ocorre em geral em problemas aperiódicos ou problemas chamados de horizonte infinito. Aqui será assumido sempre problemas de otimização de horizonte finito.

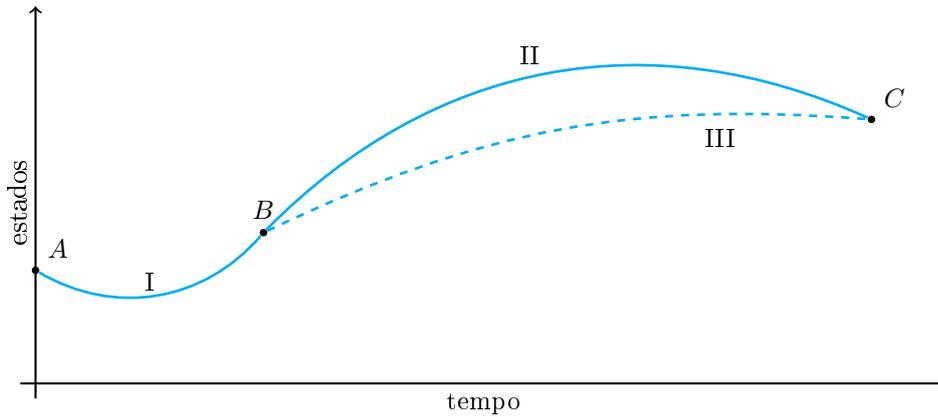


Figura 3.1 – Ilustração do princípio da otimalidade de Bellman.

### 3.1 Programação dinâmica determinística

Como o nome sugere, a programação dinâmica determinística (PDD), é um caso particular onde o sistema não possui variáveis aleatórias ( $\mathbf{Y}_k$ ). Nesse caso, pode-se definir a função objetivo, para um problema em tempo discreto com início em  $k$  e término no instante  $T$ , como na Equação (3.3). Atribuindo um significado monetário, pode-se pensar que o problema é minimizar o custo total para levar o sistema de um estado inicial até o final ao longo de um período de tempo. Nesse caso, os termos  $c_k$  são custos incrementais entre dois instantes de tempo consecutivos do sistema. A função  $\psi_T$  é dada e serve como um indicador de custo final para o problema<sup>3</sup>.

$$C_k^{(\pi)}(\mathbf{x}_k) = \sum_{k'=k}^{T-1} c_{k'}(\mathbf{x}_{k'}, \pi_{k'}(\mathbf{x}_{k'}), \mathbf{x}_{k'+1}) + \psi(\mathbf{x}_T) \quad (3.3)$$

Segundo a programação dinâmica, a minimização do problema anterior se dá resolvendo a equação recursiva de otimalidade de Bellman, Equação (3.4), o que é provado de forma rigorosa na Seção 3.2. Por enquanto, a versão determinística ajuda a esclarecer o procedimento de solução da equação recursiva e alguns detalhes da PD.

**Definição 3.2. Equação recursiva de otimalidade de Bellman:** O custo  $F_k$  restante até o fim do horizonte de otimização, a partir de um estado qualquer no instante  $k$ , depende de um termo no instante presente (custo incremental) e um termo que indique o custo futuro para o término da trajetória (custo futuro).

$$F_k(\mathbf{x}_k) = \min_{\mathbf{u}_k \in \mathcal{U}_k} \left( \underbrace{c_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1})}_{\text{custo imediato}} + \underbrace{F_{k+1}(\mathbf{x}_{k+1})}_{\text{Custo futuro}} \right) \quad (3.4)$$

<sup>3</sup> Em problemas como o POR a função  $\psi_T$  serve como um indicador monetário para o próximo horizonte de planejamento, como o próximo ano de operação, por exemplo. Em outros casos, pode servir como uma função de penalidade para impedir que o sistema acabe em um estado pouco desejado do sistema.

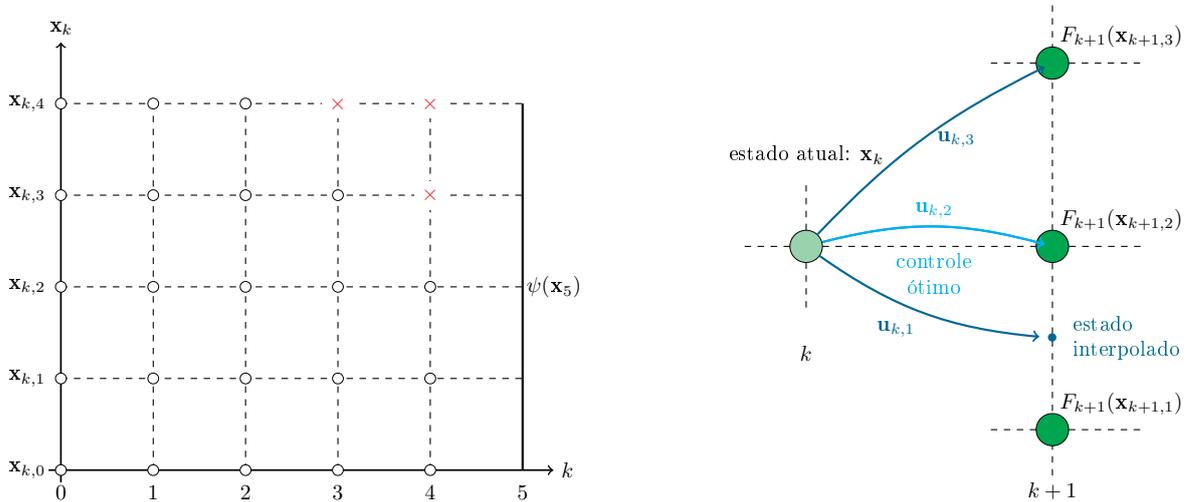
**Definição 3.3. Política ótima de operação:** A política ótima do instante  $k$ ,  $\pi_k$  é definida para cada estado como o argumento que minimiza a Equação recursiva (3.4).

$$\pi_k(\mathbf{x}_k) = \mathbf{u}_k^* = \arg \min_{\mathbf{u}_k \in \mathcal{U}_k} \left( c_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1}) + F_{k+1}(\mathbf{x}_{k+1}) \right) \quad (3.5)$$

### 3.1.1 Solução da equação recursiva de otimalidade

A abordagem tradicional para a solução da Equação (3.4) envolve a discretização do espaço de estados (estados factíveis,  $\mathcal{X}_k$ ) e de controle (controles realizáveis,  $\mathcal{U}_k$ ). Para exemplificar a solução da equação recursiva, suponhamos um problema unidimensional ( $\mathbf{x}_k = x_k \in \mathbb{R}$ ,  $\mathbf{u}_k \in \mathbb{R}$ ), com restrições de desigualdades:  $\underline{x} \leq x_k \leq \bar{x}$  e  $\underline{u} \leq u_k \leq \bar{u}$ .

Denotemos como a  $j$ -ésima discretização da variável  $x$ , no instante (ou estágio)  $k$ , como  $x_{k,j}$ , e analogamente para o controle. Suponhamos que o estado possa ser discretizado em 5 valores, Figura 3.2(a) e o controle em 3, ou seja  $\mathcal{U}_k := \{u_{k,1}, u_{k,2}, u_{k,3}\}$ , para  $T=5$  estágios. Cada ponto da discretização do espaço de estados precisa ter a Equação (3.4) solucionada, isso significa que dado um estado fixo  $x_{k,j}$ , os três possíveis valores de  $u_k$  precisam ser usados para avaliar a equação recursiva de otimalidade, Figura 3.2(b). É importante notar que cada par  $(x_{k,j}, u_{k,i})$  produz uma realização de  $F_k(x_k)$ , em que o primeiro termo  $c_k$  é função do par estado-controle e o segundo depende de  $x_{k+1} = \varphi(x_k, u_k)$ .



(a) Ilustração da discretização do espaço de estados. Os pontos em vermelhos são estados proibidos devido as restrições do problema.

(b) Cada estado precisa avaliar todos os controles disponíveis, e dentre eles adotar como decisão aquele que produz o menor valor da equação recursiva (em destaque na figura).

Figura 3.2 – Procedimentos de discretização do espaço de estado e avaliação para cada ponto do espaço discretizado.

Ao final da avaliação, o estado  $x_{k,j}$  dado terá associado a ele dois valores: o valor de  $u_{k,i}$  (controle ótimo), Equação (3.5), e o valor ótimo de  $F_k(x_{k,j})$ , Equação (3.4). Além disso, cada controle precisa produzir um estado futuro  $x_{k+1} = \varphi(x_k, u_k)$  factível.

O par  $(x_{k,j}, u_{k,i})$  nem sempre produz um  $x_{k+1} = \varphi(x_k, u_k)$  que corresponde exatamente a um valor da discretização de  $\mathcal{X}_k$ , como mostra o controle  $u_{k,1}$  do exemplo da Figura 3.2(b). Nesses casos, o indicado, em geral, é interpolar com os valores das duas discretizações (superior e inferior) do  $x_{k+1}$  para obter um valor estimado de  $F_{k+1}(x_{k+1})$ .

Podem existir alguns estados para os quais não existe qualquer controle factível como o estado  $x_{3,3}$  da Figura 3.3(b). Esses estados vão surgindo ao decorrer da recursão e precisam ser sinalizados como estados proibidos (atribuindo um custo “infinito”, por exemplo). Outros estados, podem ter um número menor de controles do que o número de discretizações de  $u_k$ ; por exemplo, alguns controles podem levar a estados fora da região de factibilidade dos estados. No caso ilustrado na Figura 3.3(b), o estado  $x_{2,2}$  só teria duas opções para decidir pela ótima, por um dos controles levar a um estado proibido,  $x_{3,3}$ .

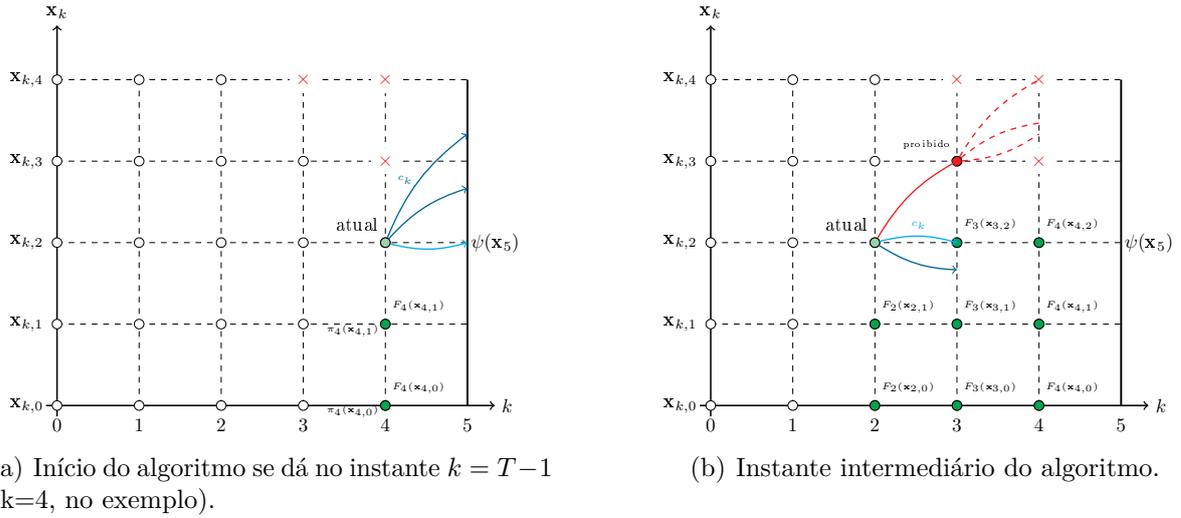


Figura 3.3 – Evolução temporal da solução da equação recursiva.

Como a Equação (3.4) é recursiva, é necessário que seja fornecido um valor inicial de  $F_k(x_k)$  para que a recursão comece. Em geral, isso corresponde a função  $\psi_T$  da Equação (3.3), ou seja  $F_T(x_T) = \psi(x_T)$ . No exemplo em questão o início do algoritmo se dá em  $k = T - 1 = 4$ , Figura 3.3(a).

$$F_4(x_{4,j}) = \min_{i \in \{1,2,3\}} \left\{ c(x_{4,j}, u_{4,i}, x_5) + \underbrace{\psi(x_5)}_{F_5(x_5)} \right\} \quad \forall j \in \{0, 1, 2, 3, 4\}$$

A recursão continua no sentido decrescente do tempo  $k$  (*backward*), Figura 3.3(b).

### 3.1.2 Obtenção da trajetória ótima

Ao fim do algoritmo, têm-se a função  $F$  e  $\pi$  (política) para todo  $k$ . A função  $\pi$  produz os custos de  $F$  e ela é a solução do problema. Além disso, dado o ponto de partida  $x_0$ , pode-se utilizar essa função para gerar uma sequência de controles ótimos.

Por exemplo, usa-se  $\pi_0(x_0) = u_0^*$ , o controle inicial, para avaliar o custo  $c_0$  e para fazer a transição de estado,  $x_1 = \varphi(x_0, u_0)$ . Se for repetido isso para  $\pi_1(x_1) = u_1$ , o segundo controle permite calcular  $c_1$  e  $x_2$ . Ou seja, o problema foi solucionado no sentido contrário do tempo, mas agora pode-se criar a trajetória, uma sequência de  $\mathbf{x}^* = x^* := \{x_0^*, x_1^*, \dots, x_T^*\}$ , produzida por uma sequência ótima de decisões:  $\mathbf{u}^* = u^* := \{u_0^*, u_1^*, \dots, u_{T-1}^*\}$  no sentido crescente do tempo. O procedimento é chamado de reconstrução da trajetória ótima, Figura 3.4.

Pode ocorrer um problema similar ao encontrado na avaliação da função  $F_{k+1}$  em  $x_{k+1}$  discutido na subseção anterior. A política  $\pi$  é obtida para pontos discretos de  $x_k$  e, portanto, o  $x_{k+1}$  obtido da decisão tomada pode não cair exatamente sobre um valor discreto. Por isso, utiliza-se, por exemplo, os dois valores mais próximos,  $\pi_k(x_{k+1,j})$  e  $\pi_k(x_{k+1,j+1})$ , para produzir uma interpolação do controle.

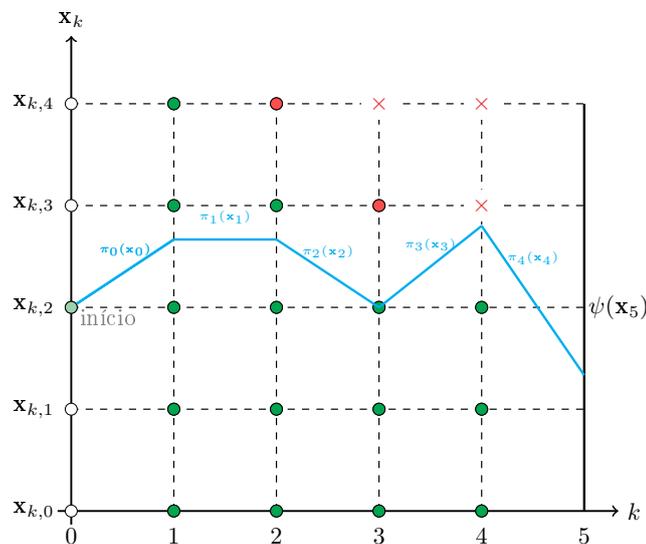


Figura 3.4 – Reconstrução da trajetória ótima da PD determinística.

## 3.2 Programação dinâmica estocástica

A programação dinâmica estocástica, o caso mais interessante de se tratar de processos dinâmicos de tomada de decisão, é caracterizada pela existência de variáveis aleatórias que influenciam as transições de estado do sistema, Equação (3.1). Do ponto de vista da função objetivo, normalmente procura-se minimizar a esperança matemática

(definida na Subseção 2.2.1) de uma função de custo. Ou seja deseja-se minimizar a função de custo expressa na Equação (3.6), onde  $\psi$  sinaliza o custo além do horizonte finito de otimização de  $T$  estágios.

$$C_k^{(\pi)}(\mathbf{x}_k) := \mathbb{E} \left\{ \sum_{k'=k}^{T-1} c_{k'}(\mathbf{x}_{k'}, \pi_{k'}(\mathbf{x}_{k'}), \mathbf{X}_{k'+1}) + \psi(\mathbf{X}_T) \middle| \mathbf{x}_k \right\} \quad (3.6)$$

A função  $C_k^{(\pi)}$  define o valor esperado do custo total partindo de um estado qualquer  $\mathbf{x}_k$  de um instante  $k$  até o instante final do horizonte  $T$ . A notação com índice  $\pi$  reforça a ideia de que essa função depende da política de operação.

Para resolver o problema programação dinâmica, a equação recursiva de otimalidade descrita na Equação (3.4) pode ser generalizada pela Equação (3.7). De forma análoga, a definição de política, Equação (3.8).

$$F_k(\mathbf{x}_k) = \min_{\mathbf{u}_k \in \mathcal{U}_k} \mathbb{E} \left( c_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{X}_{k+1}) + F_{k+1}(\mathbf{X}_{k+1}) \middle| \mathbf{x}_k \right) \quad (3.7)$$

$$\pi_k(\mathbf{x}_k) = \mathbf{u}_k^* = \arg \min_{\mathbf{u}_k \in \mathcal{U}_k} \mathbb{E} \left( c_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{X}_{k+1}) + F_{k+1}(\mathbf{X}_{k+1}) \middle| \mathbf{x}_k \right) \quad (3.8)$$

Para que o método tenha a garantia de otimalidade da função  $C_k^{(\pi)}$  justificada devidamente é necessário iniciar-se com a demonstração da propriedade estatística da esperança total, Propriedade 3.1 (ROSS, 1998).

**Propriedade 3.1. Lei da esperança total:** *Seja um conjunto de três variáveis aleatórias  $C$ ,  $\mathbf{X}_{k+1}$  e  $\mathbf{X}_k$ . Então:*

$$\mathbb{E} \left( \mathbb{E}(C | \mathbf{x}_k, \mathbf{x}_{k+1}) \middle| \mathbf{x}_k \right) = \mathbb{E}(C | \mathbf{x}_k) \quad (3.9)$$

*Demonstração.* Sem perda de generalidade, impõem-se todas as variáveis discretas e finitas. Denota-se  $\mathbb{P}(\cdot)$  como a probabilidade da realização de seu argumento. Assim,

$$\begin{aligned} \mathbb{E}(C | \mathbf{x}_k) &= \sum_{c \in \mathcal{C}} c \underbrace{\frac{\mathbb{P}(c | \mathbf{x}_k)}{\mathbb{P}(\mathbf{x}_k)}}_{\substack{\text{Def. 2.6} \\ \text{versão discreta}}} = \sum_{c \in \mathcal{C}} c \left( \frac{\mathbb{P}(c, \mathbf{x}_k)}{\mathbb{P}(\mathbf{x}_k)} \right) = \sum_{c \in \mathcal{C}} \frac{c}{\mathbb{P}(\mathbf{x}_k)} \sum_{\mathbf{x}_{k+1} \in \mathcal{X}_{k+1}} \left( \underbrace{\mathbb{P}(c, \mathbf{x}_k, \mathbf{x}_{k+1})}_{\substack{\text{Def. 2.6} \\ \text{versão discreta}}} \right) \\ &= \sum_{c \in \mathcal{C}} \frac{c}{\mathbb{P}(\mathbf{x}_k)} \sum_{\mathbf{x}_{k+1} \in \mathcal{X}_{k+1}} \left( \mathbb{P}(f | \mathbf{x}_k, \mathbf{x}_{k+1}) \mathbb{P}(\mathbf{x}_k, \mathbf{x}_{k+1}) \right) \end{aligned}$$

$$\begin{aligned}
 &= \sum_{c \in \mathcal{C}} \sum_{\mathbf{x}_{k+1} \in \mathcal{X}_{k+1}} \frac{c}{\mathbb{P}(\mathbf{x}_k)} \mathbb{P}(f | \mathbf{x}_k, \mathbf{x}_{k+1}) \mathbb{P}(\mathbf{x}_k, \mathbf{x}_{k+1}) \\
 &= \left( \sum_{\mathbf{x}_{k+1}} \underbrace{\frac{\mathbb{P}(\mathbf{x}_k, \mathbf{x}_{k+1})}{\mathbb{P}(\mathbf{x}_k)}}_{\mathbb{P}(\mathbf{x}_{k+1} | \mathbf{x}_k)} \underbrace{\sum_{c \in \mathcal{C}} c \mathbb{P}(c | \mathbf{x}_k, \mathbf{x}_{k+1})}_{\mathbb{E}(C | \mathbf{x}_k, \mathbf{x}_{k+1})} \right) \\
 &= \sum_{\mathbf{x}_{k+1} \in \mathcal{X}_{k+1}} \mathbb{P}(\mathbf{x}_{k+1} | \mathbf{x}_k) \mathbb{E}(C | \mathbf{x}_k, \mathbf{x}_{k+1}) = \mathbb{E} \left( \mathbb{E}(C | \mathbf{x}_k, \mathbf{x}_{k+1}) \middle| \mathbf{x}_k \right)
 \end{aligned}$$

□

No caso da programação dinâmica, o índice  $k$  tem função temporal. Portanto, uma função  $C$  de uma variável aleatória  $\mathbf{X}_{k+1}$  que representa o estado do sistema tem a propriedade de condiciona-lá a  $\mathbf{X}_{k+1}$ ; isto faz com que toda informação anterior a  $k+1$  seja irrelevante (POWELL, 2007). Pode-se pensar nisso como uma consequência do princípio de otimalidade, informações de estados “recentes”, que carregam as informações de instantes anteriores, são suficientes para descrever as probabilidades do sistema. A consequência disso é que se pode escrever  $\mathbb{E}(C | \mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k, \mathbf{x}_{k+1}) = \mathbb{E}(C | \mathbf{x}_{k+1})$ . No caso da PD pode-se escrever a Equação (3.10).

$$\mathbb{E} \left( \mathbb{E}(C | \mathbf{x}_{k+1}) \middle| \mathbf{x}_k \right) = \mathbb{E}(C | \mathbf{x}_k) \quad (3.10)$$

É também possível provar, como mostra Powell (2007), que a equação recursiva de otimalidade é equivalente ao problema de otimização definido pela Equação (3.11).

$$C_k^* = \min_{\pi \in \Pi} C_k^{(\pi)} \quad k = 0, 1, 2, \dots, T-1 \quad (3.11)$$

**Teorema 3.1.**  $F_k = C_k^*$ . Com  $F_k$  definida pela equação recursiva de otimalidade de Bellman, Equação (3.7)

*Demonstração.* A demonstração desse resultado é feita em duas partes:

Parte 1)

$$\begin{aligned}
 F_k(\mathbf{x}_k) &= \min_{\mathbf{u}_k \in \mathcal{U}_k} \mathbb{E} \left( c_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{X}_{k+1}) + F_{k+1}(\mathbf{X}_{k+1}) \middle| \mathbf{x}_k \right) \\
 &\leq \mathbb{E} \left( c_k(\mathbf{x}_k, \pi_k(\mathbf{x}_k), \mathbf{X}_{k+1}) + F_{k+1}(\mathbf{X}_{k+1}) \middle| \mathbf{x}_k \right) \\
 &= \mathbb{E} \left( c_k(\mathbf{x}_k, \pi_k(\mathbf{x}_k), \mathbf{X}_{k+1}) \right. \\
 &\quad \left. + \min_{\mathbf{u}_{k+1} \in \mathcal{U}_{k+1}} \mathbb{E} \left( c_{k+1}(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}, \mathbf{X}_{k+2}) + F_{k+2}(\mathbf{X}_{k+1}) \middle| \mathbf{x}_{k+1} \right) \middle| \mathbf{x}_k \right) \\
 &\leq \mathbb{E} \left( c_k(\mathbf{x}_k, \pi_k(\mathbf{x}_k), \mathbf{X}_{k+1}) \right)
 \end{aligned}$$

$$\begin{aligned}
& + \underbrace{\mathbb{E}\left(c_{k+1}(\mathbf{x}_{k+1}, \pi_{k+1}(\mathbf{x}_{k+1}), \mathbf{X}_{k+2}) + F_{k+2}(\mathbf{X}_{k+2}) \middle| \mathbf{x}_{k+1}\right)}_{\text{usando a Eq. (3.10)}} \middle| \mathbf{x}_k \\
& = \mathbb{E}\left(c_k(\mathbf{x}_k, \pi_k(\mathbf{x}_k), \mathbf{X}_{k+1}) + c_{k+1}(\mathbf{x}_{k+1}, \pi_{k+1}(\mathbf{x}_{k+1}), \mathbf{X}_{k+2}) + F_{k+2}(\mathbf{X}_{k+2}) \middle| \mathbf{x}_k\right)
\end{aligned}$$

Repetindo  $T - 1 - k$  vezes o procedimento anterior

$$\begin{aligned}
F_k(\mathbf{x}_k) & \leq \mathbb{E}\left(c_k(\mathbf{x}_k, \pi_k(\mathbf{x}_k), \mathbf{X}_{k+1}) + \dots + c_{T-1}(\mathbf{x}_{T-1}, \pi_k(\mathbf{x}_{T-1}), \mathbf{X}_T) + F_T(\mathbf{X}_T) \middle| \mathbf{x}_k\right) \\
& = \mathbb{E}\left(c_k(\mathbf{x}_k, \pi_k(\mathbf{x}_k), \mathbf{X}_{k+1}) + \dots + c_{T-1}(\mathbf{x}_{T-1}, \pi_k(\mathbf{x}_{T-1}), \mathbf{X}_{k+1}) + \psi_T(\mathbf{X}_T) \middle| \mathbf{x}_k\right) \\
& = \mathbb{E}\left(\sum_{k'=k}^{T-1} c_{k'}(\mathbf{x}_k, \pi_k(\mathbf{x}_k), \mathbf{X}_{k'+1}) + \psi_T(\mathbf{X}_T) \middle| \mathbf{x}_k\right) \stackrel{\text{Eq. 3.6}}{=} C_k^{(\pi)}(\mathbf{x}_k) \\
\therefore F_k(\mathbf{x}_k) & \leq C^{(\pi)}(\mathbf{x}_k) \quad \forall \pi \in \Pi
\end{aligned}$$

Parte 2)

É fácil notar que qualquer que seja a política  $\pi \in \Pi$  existe um  $\epsilon_k^{(\pi)} > 0$  tal que possa ser estabelecida a seguinte desigualdade:

$$F_k(\mathbf{x}_k) \geq \mathbb{E}\left(c_k(\mathbf{x}_k, \pi_k(\mathbf{x}_k), \mathbf{X}_{k+1}) + F_{k+1}(\mathbf{X}_{k+1}) \middle| \mathbf{x}_k\right) - \epsilon_k^{(\pi)}$$

Aplicando o mesmo processo para parte 1) com a recursão de  $F_{k+1}$ ,  $T - k - 1$  vezes:

$$\begin{aligned}
F_k(\mathbf{x}_k) & \geq \mathbb{E}\left(\sum_{k'=k}^{T-1} c_{k'}(\mathbf{x}_{k'}, \pi_k(\mathbf{x}_{k'}), \mathbf{X}_{k'+1}) + \psi_T(\mathbf{X}_k) \middle| \mathbf{x}_k\right) - \sum_{k'=k}^{T-1} \epsilon_{k'}^{(\pi)} \\
& = C_k^{(\pi)} - \sum_{k'=k}^{T-1} \epsilon_{k'}^{(\pi)}
\end{aligned}$$

Fazendo  $\epsilon := \max\left(\epsilon_{k'}^{(\pi)}\right) \forall k' = k, k+1, k+2, \dots, T-1$ :

$$\therefore F_k(\mathbf{x}_k) \geq C_k^{(\pi)} - (T - k - 1)\epsilon \quad \pi \in \Pi$$

Considerando-se as desigualdades das duas partes, que se mantêm para qualquer política, então para a política ótima:

$$C_k^* \geq F_k(\mathbf{x}_k) \geq C_k^* - (T - k - 1)\epsilon$$

No caso da política ótima, por definição, pode-se fazer  $\epsilon$  tão pequeno quanto se queira ou seja,  $\epsilon \rightarrow 0$ , o que prova  $C_k^* = F_k$ .  $\square$

Provada a garantia de otimalidade da equação recursiva, as interpretações de cada parcela da Equação (3.7) são as mesmas da seção anterior, uma relacionada com a transição

para o próximo instante (custo presente) e a segunda com o custo restante até o fim do horizonte, porém agora em termos de um valor esperado dessas parcelas.

Além disso, é interessante notar que a Equação (3.7) otimiza a função  $C_k^{(\pi)}$  que não inclui termos anteriores ao instante  $k$ , ou seja, a equação recursiva de otimalidade é a transcrição matemática do princípio de otimalidade, onde a otimização partindo do instante  $k$  só precisa olhar para os instantes futuros (e não para o passado).

Uma forma conveniente de reescrever essa equação é através da discretização do espaço de estado e controle. Seja  $\mathcal{I}_{k+1}$  um conjunto finito de índices para o conjunto discreto  $\mathcal{X}_{k+1}$ . Assim, com o conjunto de índices  $\mathcal{I}_{k+1}$  e  $\mathcal{X}_{k+1}$  conjuntos finitos, pode-se aproximar a esperança matemática por uma soma finita, como mostra a Equação (3.12).

$$F_k(\mathbf{x}_k) = \min_{\mathbf{u}_k \in \mathcal{U}_k} \sum_{i \in \mathcal{I}_{k+1}} \mathbb{P}(\mathbf{x}_{k+1,i} | \mathbf{x}_k) \left( c_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{x}_{k+1,i}) + F_{k+1}(\mathbf{x}_{k+1,i}) \right) \quad (3.12)$$

$$F_k(\mathbf{x}_{k,j}) = \min_{\mathbf{u}_k \in \mathcal{U}_k} \sum_{i \in \mathcal{I}_{k+1}} p_{ij}^{(\pi)} \left( c_k(\mathbf{x}_{k,j}, \mathbf{u}_k, \mathbf{x}_{k+1,i}) + F_{k+1}(\mathbf{x}_{k+1,i}) \right) \quad (3.13)$$

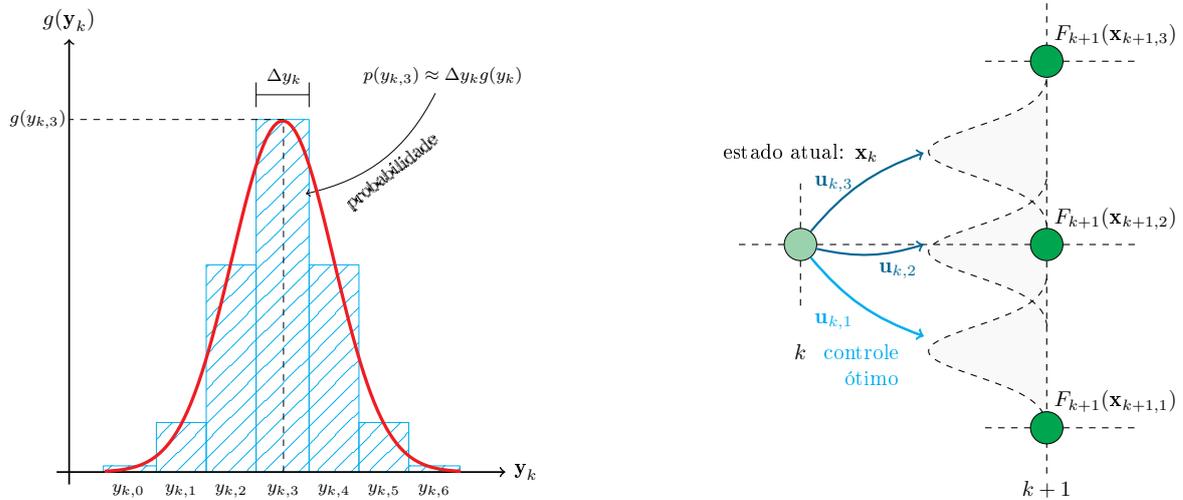
Uma forma mais compacta de escrever a equação recursiva é através da notação  $p_{ij} = \mathbb{P}(\mathbf{x}_{k+1,i} | \mathbf{x}_{k,j})$ , ou seja, a probabilidade de transitar do  $j$ -ésimo estado indexado do conjunto discreto  $\mathcal{X}_k$  para o  $i$ -ésimo estado do espaço  $\mathcal{X}_{k+1}$ . Como essa transição depende de uma decisão  $\mathbf{u}_k$ , é conveniente usar a notação  $p_{ij}^{(\pi)}$ , para enfatizar que essa probabilidade depende da política de decisões do sistema, como mostra a Equação (3.13). De uma forma geral, é a função de densidade de probabilidade de  $\mathbf{Y}_k$  que define essas probabilidades, já que em  $\mathbf{X}_{k+1} = \varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{Y}_k)$ . Como  $p_{ij}^{(\pi)}$  corresponde a transição de estados discretos, então é necessária também a discretização do espaço  $\mathcal{Y}_k$  e de sua densidade de probabilidade, como mostra a Figura 3.5(a).

Do ponto de vista de implementação, há pouca diferença em relação à programação dinâmica determinística. Em uma versão estocástica do exemplo dado na Subseção 3.1.1, cada controle leva o sistema a um intervalo de prováveis valores de estados futuros ( $\mathbf{X}_{k+1}$ ), como mostra a Figura 3.5(b). Ou seja, cada valor da discretização de  $\mathcal{Y}_k$  junto com o par de estado-controle dado  $(\mathbf{x}_k, \mathbf{u}_k)$  define um conjunto finito de probabilidades  $\{p_{i,j}^{(\pi)}\}$ , onde cada elemento é associado a um valor realizável de  $\mathbf{X}_{k+1}$ , e, conseqüentemente, a um valor de  $F_{k+1}(\mathbf{X}_{k+1})$  (calculado em um passo anterior do algoritmo) e a um custo  $c_k$ .

É importante lembrar que os estados futuros prováveis que não correspondam aos da discretização precisam ser interpolados, como na Subseção 3.1.1. O Algoritmo 3.1 exemplifica uma implementação genérica da PDE.

A diferença mais significativa da PDD em relação a PDE é que na versão estocástica não há mais sentido em falar de reconstrução de trajetória; ou seja não há mais o conceito de uma trajetória ótima. Na versão determinística o sistema sempre pode caminhar sobre

uma trajetória desejada; na versão estocástica isso não é possível devido às aleatoriedades.



(a) Discretização da função de densidade de probabilidade da variável  $Y_k$

(b) Cada estado precisa avaliar todos os controles disponíveis. Porém dessa vez, uma faixa de valores de  $X_{k+1}$  pode ser alcançada.

Figura 3.5 – Diferenças na programação dinâmica estocástica.

### 3.2.1 As vantagens da PD

Dentre as principais vantagens da programação dinâmica está a possibilidade de tratar formalmente as variáveis aleatórias no método, como mostra a Equação (3.13). Se essa vantagem não fosse o bastante para dar destaque a PD, outras vantagens ainda poderiam ser listadas.

Uma dessas outras vantagens é a decomposição temporal do problema, propriedade na qual a Equação (3.13) deixa em evidência. Por exemplo, considerando um problema de pequeno porte de otimização de fluxos em redes, com 3 nós<sup>4</sup>, com uma topologia simples radial, no qual deseja-se otimizar o fluxo nessa rede no tempo, Figura 3.6. Em métodos clássicos de otimização de fluxos em redes, são necessárias 36 variáveis no modelo, uma para cada arco da rede. Já para a programação dinâmica, o problema é decomposto temporalmente, sendo necessário tratar 5 problemas de dimensão 6 (3 estados e 3 controle).

Além disso a adição de mais um nó faz a dimensão dos métodos de fluxo aumentar para 46, enquanto para a PD faz crescer para 8. A adição de um instante de tempo faz o primeiro método crescer para 54, enquanto com programação dinâmica, apenas mais um problema de dimensão 6 precisa ser resolvido, o que faz o tempo de execução crescer linearmente para a PD, (BELLMAN, 1957).

<sup>4</sup> Os nós são representados por triângulos para evitar confusões com a notação circular dos estados discretos de  $\mathcal{X}_k$  fazer um paralelo com o POR.

**Algoritmo 3.1:** Algoritmo de Programação Dinâmica Estocástica.

---

```

1  Dados os conjuntos de índices  $\mathcal{I}_k$  e  $\mathcal{J}_k$ , de  $\mathbf{x}_k$  e  $\mathbf{u}_k$ , respectivamente, e respeitando as
   restrições. E  $\forall k \in \{0, 1, \dots, T-1\}$ .
2  Faça  $k \leftarrow T-1$ 
3  Dado a condição de contorno  $\mathbf{x}_0$  e a função  $\psi(\mathbf{x}_T)$ .
4  enquanto  $k \geq 0$  faça
5      para  $i \in \mathcal{I}_k$  faça
6          minimo  $\leftarrow \infty$ 
7           $\mathbf{u}^* \leftarrow 0$ 
8          para  $j \in \mathcal{J}_k$  faça
9              Dados  $\mathbf{x}_{k,i}$  e  $\mathbf{u}_{k,j}$ ,  $\forall$  discretizações de  $\mathbf{y}_k$  calcule todos  $\mathbf{x}_{k+1}$  possíveis.
10             Encontre a probabilidade de transição  $\mathbf{x}_{k,i} \rightarrow \mathbf{x}_{k+1,l}$ :  $p_{il}^{(\pi)}$ 
11             (usando a densidade de probabilidade de  $\mathbf{y}_k$ , discretizada).
12              $f \leftarrow \sum_{l \in \mathcal{I}_{k+1}} p_{il}^{(\pi)} (c_k(\mathbf{x}_{k,i}, \mathbf{u}_{k,j}, \mathbf{x}_{k+1,l}) + F_{k+1}(\mathbf{x}_{k+1,l}))$ 
13             se  $f < \text{minimo}$  então
14                 minimo  $\leftarrow f$ 
15                  $\mathbf{u}^* \leftarrow \mathbf{u}_{k,j}$ 
16             fim
17         fim
18          $F(\mathbf{x}_{k,i}) \leftarrow \text{minimo}$ 
19          $\pi(\mathbf{x}_{k,i}) \leftarrow \mathbf{u}^*$ 
20     fim
21      $k \leftarrow k - 1$ 
22 fim
23 Avaliação final de  $\pi(\mathbf{x})$  para diversos cenários de variáveis aleatórias.

```

---

Outro benefício da PD é relacionado as exigências sobre a função objetivo. Existem poucos requisitos sobre o comportamento da função objetivo ( $C_k^{(\pi)}$ ) ou sobre os termos dos custos incrementais ( $c_k$ ). Esta generalidade é útil na solução do POR apresentado no Capítulo 2, onde há pontos onde a função objetivo é não diferenciável.

Também é usual comparar a PD com o método de enumeração, como mostra a Figura 3.7, adaptada de Larson (1968). Recuperando o exemplo da Subseção 3.1.1, onde o sistema unidimensional possui três opções de controle, a forma imediata seria a enumeração de todos os caminhos possíveis, ou seja, tomar todas as decisões e calcular todos os custos de transições possíveis - o custo é sempre a soma do custo para realizar a transição mais o custo acumulado no estado de origem da transição. No caso do exemplo simples da seção sobre PDD, deveriam ser avaliados  $3^5 = 243$  caminhos no espaço de estados. Já para a programação dinâmica, a avaliação da equação recursiva de otimalidade se daria apenas  $5 \cdot 5 \cdot 3 = 75$  vezes<sup>5</sup>.

<sup>5</sup> Nota-se claramente uma vantagem numérica sobre o número de avaliações. Porém em problemas pequenos (poucos controles e estágio) isso pode não ser verdade se o número de discretizações dos estados for muito grande. Na Figura 3.7 a situação ilustrada corresponde a um problema com apenas dois controles, diferente dos três do problema da seção anterior. Nesse caso haveria 32 caminhos

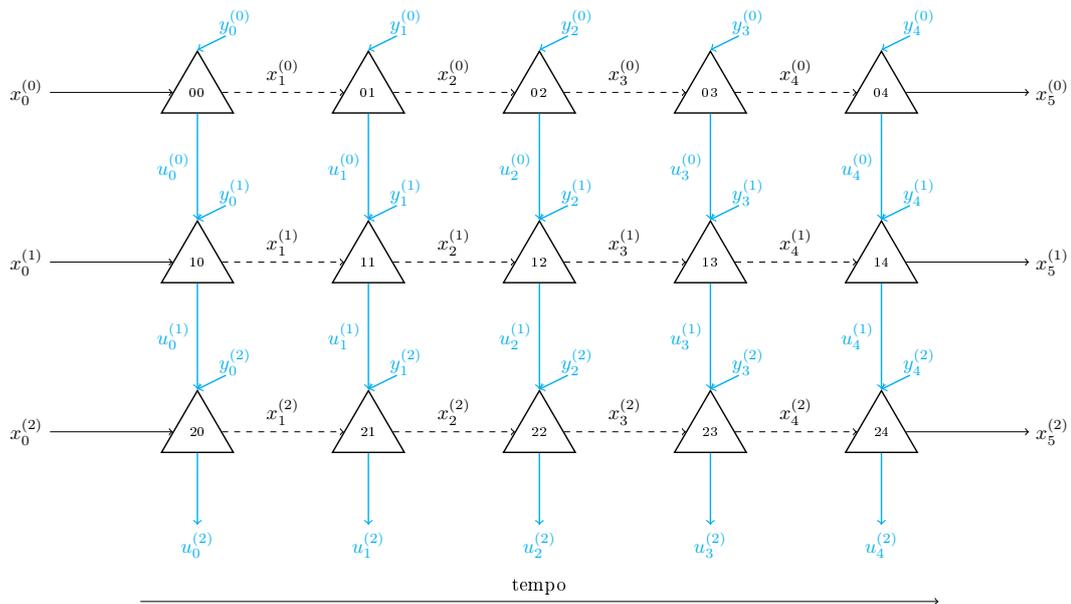


Figura 3.6 – Problema de otimização em rede de pequeno porte.

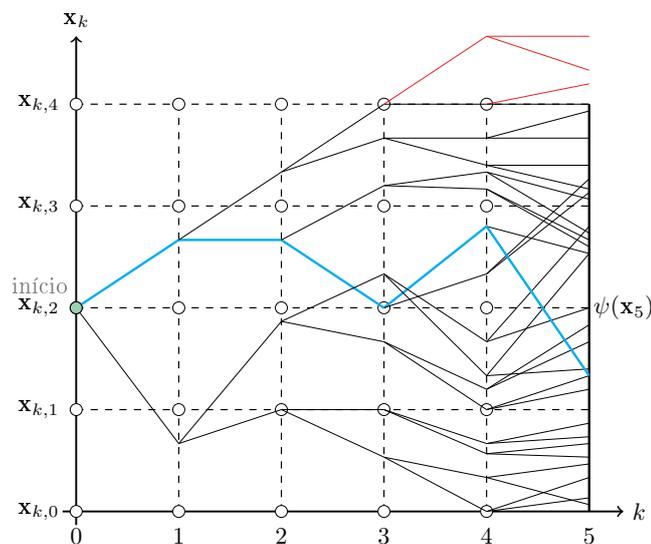


Figura 3.7 – Método de enumeração para um problema com apenas duas decisões possíveis. As curvas em vermelho representam trajetórias infactíveis e a azul a trajetória ótima.

Bellman (1957) chama a atenção para o fato de que, mesmo no caso determinístico, a PD fornece a decisão ótima para qualquer estado do sistema, e não somente para cada estágio. Ou seja, a política final determinada pelo método soluciona um vasto número de cenários possíveis de uma única vez; Bellman chamou essa propriedade de “invariant

---

possíveis e a PD avaliaria 50 vezes a equação recursiva, para 5 estados e 5 estágios, nesses casos é necessário avaliar o tamanho da discretização dos estados (o que interfere de forma direta no número de vezes que a equação recursiva é solucionada).

*imbedding*”. De forma mais precisa, essa característica descreve um problema de malha fechada (BERTSEKAS, 1976), onde o sistema é capaz de responder de forma ótima a perturbações arbitrárias. A Figura 3.8 ilustra esta característica.

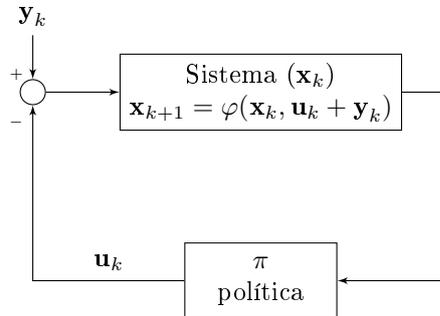


Figura 3.8 – Interpretação da PD como um problema de controle em malha fechada (Diagrama particular para o problema de POR).

### 3.2.2 A maldição da dimensionalidade

Das subseções anteriores pode-se notar que as soluções com PD são extremamente dependentes das discretizações que se escolhe: uma escolha grosseira (poucos valores discretos) faz com que o problema tenha uma rápida, porém imprecisa, solução (apesar de ótima para a discretização adotada); por outro lado, uma discretização muito granular faz a solução mais precisa, em troca de um aumento da complexidade computacional, que tende a um comportamento exponencial, efeito conhecido como “maldição da dimensionalidade” (BELLMAN, 1957). Os principais fatores que influenciam na maldição da dimensionalidade são os três seguintes:

- É necessária a solução de (3.13) para todos os valores discretos dos estados. Tanto uma grande cardinalidade de  $\mathcal{X}_k$  quanto o tamanho da discretização de  $\mathcal{U}_k$  tendem a tornar a complexidade computacional intratável, para a abordagem convencional.
- Aleatoriedades fazem do cálculo da esperança matemática um fator adicional para o crescimento da complexidade computacional. Se  $\mathbf{Y}_k$  possuir uma dimensão elevada, e cada uma de suas componentes seguir uma distribuição contínua, que deve ser discretizada, então cada valor discreto de cada densidade de probabilidade deverá ser combinado um a um, para gerar todos os valores possíveis de  $\mathbf{X}_{k+1}$ , e também  $F_{k+1}(\mathbf{X}_{k+1})$ , novamente para cada valor discreto de controle.
- A solução de (3.13) exige que os valores da função  $F_k$  sejam armazenados. Em termos práticos, é necessária uma malha (com o número de posições igual a cardinalidade de  $\mathcal{X}_k$ ) para armazená-los. Sendo assim, uma discretização muito granular implica em um maior volume de dados sendo armazenados. Essa estratégia é chamada de

tabular (*lookup table*), pois é sempre necessário realizar uma consulta em uma tabela, função  $F_k$ , para resolver a equação de otimalidade.

Para exemplificar o impacto do problema de crescimento exponencial da complexidade computacional, pode-se estimar o tempo necessário na solução de um problema “não muito grande”. Seja um problema determinístico de 10 estágios com um espaço de estado com dimensão 10 e o mesmo para o controle, e ainda com discretizações de 100 valores. Nesse caso, a equação recursiva de otimalidade precisaria ser resolvida para  $100^{10}$  estados avaliando  $100^{10}$  controles factíveis. Para 10 estágios, teríamos um total de  $10^{41}$  operações.

Se cada operação for aproximada como uma operação em ponto flutuante, e realizada com o supercomputador mais rápido<sup>6</sup> em 2018, com uma capacidade de aproximadamente  $200 \cdot 10^{15}$  FLOPS (operações em ponto flutuante por segundo) a solução do problema requer aproximadamente  $5 \cdot 10^{23}$  s. Este tempo é muito maior que ordem de grandeza da idade aproximada do universo<sup>7</sup>,  $4,3 \cdot 10^{17}$  s.

Desde a formulação da programação dinâmica Bellman já alertava para esta dificuldade, sintetizada na expressão “maldição da dimensionalidade”. Como observado na Seção 1.5, desde a década de 60 já se busca desenvolver métodos alternativos que tenham as vantagens da PD, mas que atenuem o efeito agressivo do crescimento da complexidade computacional.

### 3.3 Programação dinâmica aproximada

A programação dinâmica aproximada (POWELL, 2007) não altera a formulação e nem mesmo a equação fundamental da PD, Equação (3.13). Em termos intuitivos, a programação dinâmica aproximada<sup>8</sup> (PDA) altera apenas a forma como a equação recursiva é resolvida.

Na Seção 3.2, mencionou-se que a a *PDE* clássica necessita que a Equação (3.13) seja avaliada para todos os pontos discretos do espaço de estados  $\mathcal{X}_k$ . Porém nem todos esses estados tem uma probabilidade representativa de serem visitados em algum instante. A PDA busca uma alternativa para que a equação recursiva só precise ser solucionada nos estados mais prováveis.

<sup>6</sup> IBM Summit - Oak Ridge National Laboratory,

<sup>7</sup> Ou seja, em um problema como a de operação de reservatórios, o SIN seria impossível de ser solucionado pela abordagem convencional de PD. Um problema determinístico de 6 reservatórios com planejamento para um ano, determinístico, levaria um ano para ser solucionado por um supercomputador como o do exemplo.

<sup>8</sup> Essa nomenclatura é um pouco ambígua, já que qualquer método que não siga o Algoritmo 3.1 provavelmente utiliza uma aproximação; portanto, também poderiam ser chamados de programação dinâmica aproximada

Uma das maneiras de se processar apenas estados prováveis na otimização é usando simulação de cenários. Se  $\tilde{\mathbf{y}} = \{\tilde{\mathbf{y}}_0, \tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_T\}$  corresponde a um cenário específico de variáveis aleatórias, então dado o estado inicial do sistema  $\mathbf{x}_0$  a equação recursiva é solucionada apenas para esse ponto e a primeira informação aleatória utilizada para fazer a transição  $\mathbf{x}_1 = \varphi(\mathbf{x}_0, \mathbf{u}_0, \tilde{\mathbf{y}}_0)$ , que é o único estado que a equação recursiva será tratada em  $k = 1$ . Na Figura 3.9 é ilustrado um problema bidimensional ( $\mathbf{x}_k \in \mathbb{R}^2$ ); cada grade representa uma tabela que armazena os valores de  $F_k$  (abordagem tabular), e cada célula corresponde a um valor dessa função para um estado discreto de  $\mathcal{X}_k$ .

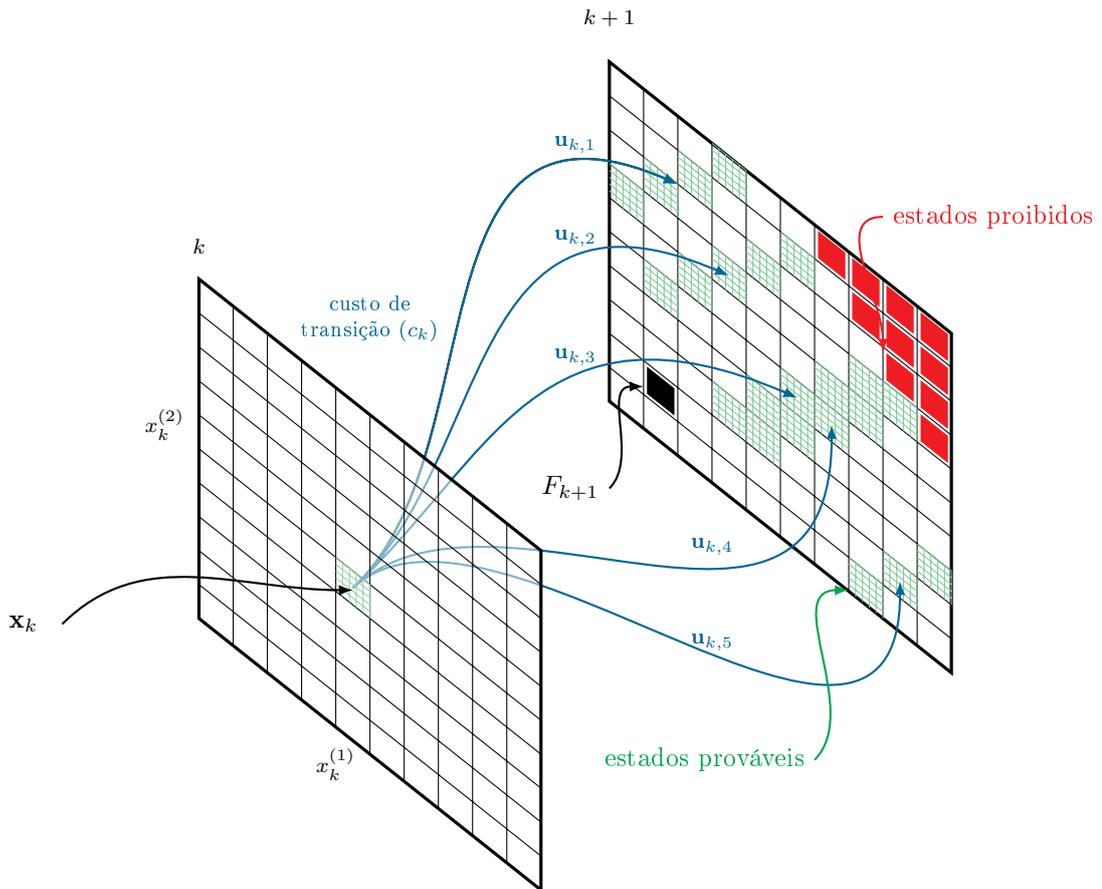


Figura 3.9 – Solução da equação recursiva apenas para um ponto do espaço de estados para cada estágio. As regiões em verde indicam o intervalo de valores de  $\mathbf{X}_{k+1}$  que o sistema pode vir a ter se um controle específico for adotado. As regiões em vermelho indicam estados proibidos.

A Equação (3.13) necessita no instante  $k$  de informações da função  $F_{k+1}(\mathbf{x}_{k+1})$ , portanto essa função precisa ser inicializada para o algoritmo. A inicialização pode ser qualquer, porém uma boa inicialização pode ajudar na convergência do algoritmo. Pensando em termo de uma abordagem tabular, cada  $F_k$  forma uma tabela de consulta (assim como  $\pi_k$ ); se a equação recursiva for solucionada para um determinado ponto (estado) específico, o valor dessa função e da política nesse ponto é atualizado.

Após o algoritmo chegar ao fim do horizonte, um ponto de cada instante  $k$  terá sido atualizado, tanto para  $F_k$ , quanto para  $\pi_k$ , como mostra a Figura 3.10. Em seguida, volta-se para o instante inicial e repete-se o procedimento. Nas iterações seguintes a função  $F_k$  vai sendo atualizada e o algoritmo passa a ter acesso às informações das iterações anteriores, como mostra a Figura 3.11.

Nota-se aqui a diferença básica entre a PDE e a PDA: na primeira a equação recursiva é solucionada para todos os estados no sentido *backward* no tempo, enquanto que no método aproximado a solução é no sentido *forward* no tempo e apenas para um estado em cada instante de tempo, porém repetindo o procedimento em diversas iterações. Deixa-se de resolver a equação recursiva para todos os estados em troca da garantia de otimalidade, pois não é mais possível realizar a demonstração da Seção 3.2 (POWELL, 2007).

Comparando os Algoritmos 3.1 e 3.2, esse último para a PDA, observa-se que no Algoritmo da PDA não há avaliações da equação recursiva em todos os estados; isto alivia a complexidade computacional. Porém, mantém-se ainda aspectos de discretização associados a abordagens tabulares, pois facilitam a consulta a valores de  $F_k$ .

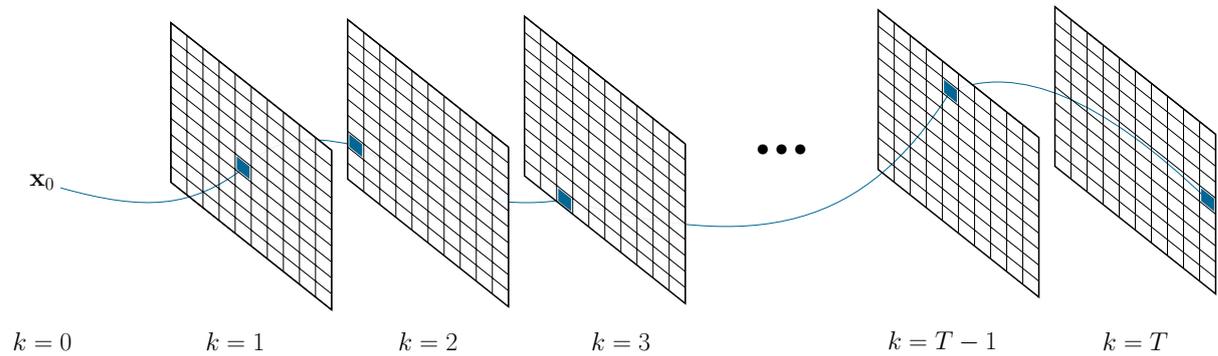


Figura 3.10 – Primeira iteração do algoritmo, destaque para a atualização da função  $F_k$ .

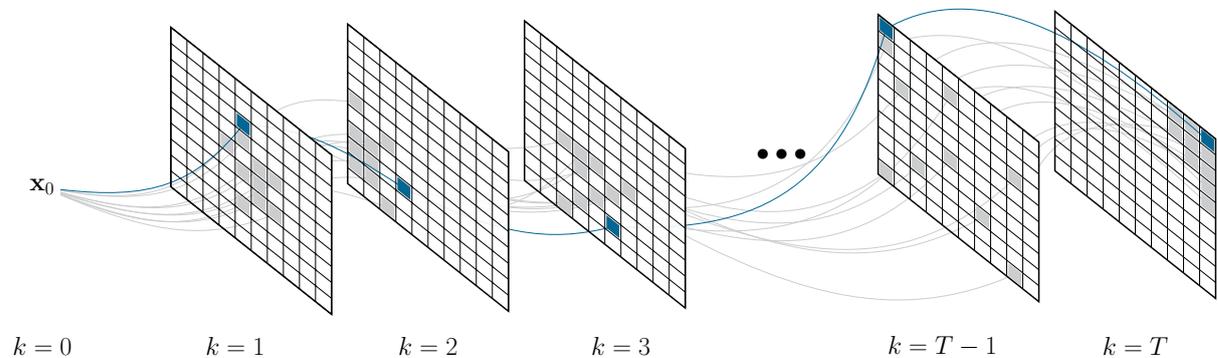


Figura 3.11 –  $N$  iterações do algoritmo aproximado.

### 3.3.1 Atualizações da função de custo

Uma iteração da programação dinâmica aproximada produz uma sequência de variáveis aleatórias, pois  $\mathbf{Y}_k$  é aleatória e  $\mathbf{X}_{k+1} = \varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{Y}_k)$  forma uma variável aleatória; logo  $F_{k+1}(\mathbf{X}_{k+1})$  também. A cada iteração, o algoritmo passa por pontos do espaço de estados e faz amostragens da função  $F_k$  com as informações disponíveis para isso. No início do algoritmo essas informações são inicializadas; ao longo das iterações atualizadas e corrigidas.

Uma forma conveniente de reescrever a equação recursiva é em termos da probabilidade  $p_k(\mathbf{y}_k|\mathbf{y}_{k-1})$ ; ou seja, a probabilidade é obtida diretamente sobre a distribuição da informação externa  $\mathbf{Y}_k$ , Equação (3.14). Do ponto de vista do POR, a relação entre  $\mathbf{Y}_k$  e  $\mathbf{X}_{k+1}$  é linear, e ambos tendem a seguir a mesma distribuição de probabilidade<sup>9</sup>.

$$f_k(\mathbf{x}_k) = \min_{\mathbf{u}_k \in \mathcal{U}_k} \sum_{\mathbf{y}_k \in \mathcal{Y}_k} p_k(\mathbf{y}_k|\mathbf{y}_{k-1}) \left( c_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{y}_k) + F_{k+1}(\varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{y}_k)) \right) \quad (3.14)$$

No contexto da programação dinâmica aproximada, pode-se pensar na Equação (3.14) como uma realização de uma variável aleatória (por isso o lado esquerdo é escrito em letras minúsculas). Como os valores exatos das funções  $F_k$  não estão disponíveis, principalmente nas primeiras iterações, Powell (2007) sugere que a estimativa de  $F_k$  seja dada pela Equação (3.15)<sup>10</sup>, onde  $n$  corresponde ao índice da iteração e  $\alpha_n$  corresponde a um passo de atualização.

$$F_k^{(n)} = (1 - \alpha_{n-1})F_k^{(n-1)} + \alpha_{n-1}f_k \quad (3.15)$$

Assim, a Equação (3.15) define uma regra para que os valores armazenados nas tabelas de  $F_k$  sejam atualizados caso os estados associados a eles sejam visitados. Em geral o passo  $\alpha$  depende do número da iteração - volta-se a discutir esse parâmetro no capítulo seguinte.

A amostragem de  $f$  é feita exatamente como no caso da PDE. Retomamos o contexto de um problema bidimensional com função da densidade de probabilidades de  $\mathbf{Y}_k$ : Neste caso, dada a distribuição de  $\mathbf{Y}_k$ , um conjunto de valores discretos e suas probabilidades de ocorrência, todos os valores de  $\mathbf{X}_{k+1}$  são determinados e com eles, o conjunto de  $c_k$  também é calculado. Além disso, com o conjunto de estados futuros, são consultados nas tabelas os valores de  $F_{k+1}$ , produzindo mais um conjunto de valores. Com  $c_k$  e  $F_k$  determinados, soma-se um a um, multiplicados pelas respectivas probabilidades de

<sup>9</sup> Com exceção quando  $\mathbf{X}_{k+1}$  supera o valor máximo do reservatório, quando há um pico de probabilidade nesses estados. Porém, isto não traz problemas adicionais na solução da equação recursiva

<sup>10</sup> Nota-se que a equação é similar a um algoritmo de gradiente estocástico, muito comum em problemas de otimização dinâmicos estocásticos, como o POR.

$\mathbf{Y}_k$ , somados para todos os valores realizáveis dessa variável. A Figura 3.12 ilustra este processo.

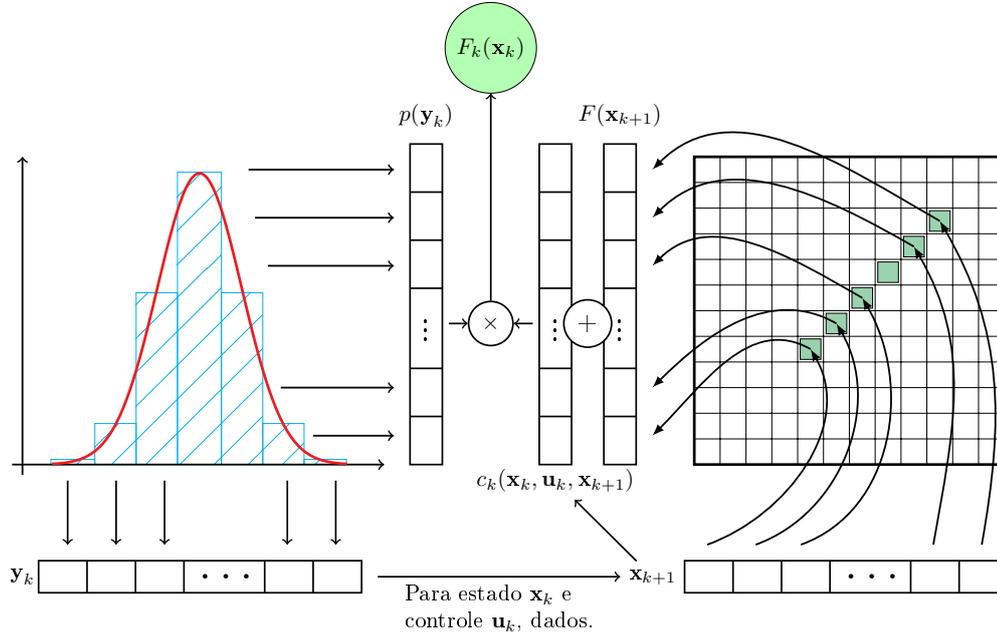


Figura 3.12 – Ilustração do processo da avaliação da equação recursiva de otimalidade, dado o estado do sistema, tanto para PDE quanto PDA

### 3.3.2 Avaliação de política

A cada iteração  $n$  a política  $\pi$  também é atualizada. A atualização da política pode servir como um parâmetro de parada do algoritmo. Ou seja, cada política produz um valor esperado de custo, como representado na Equação (3.6).

Pode-se considerar um cenário possível de variáveis aleatórias  $\tilde{\mathbf{y}} \in \tilde{\mathcal{Y}}$ . Este cenário produz uma trajetória para o sistema, que possui um custo total para levar o sistema do instante  $k = 0$  a  $k = T$ . Se uma série de cenários possíveis forem avaliados pela política  $\pi$ , uma análise estatística pode ser utilizada para avaliar propriedades paramétricas que esses custos possuem para a política específica, média e desvios por exemplo.

Assim, pode-se adotar como critério de parada a diferença produzida por duas políticas, como mostra a Equação (3.16). A cada  $N_a$  iterações, duas políticas (a anterior e a atual) podem ser avaliadas, definindo um critério para encerrar o algoritmo.

$$\Delta^{(\pi_1 \pi_2)} := \left| \frac{C^{(\pi_2)} - C^{(\pi_1)}}{C^{(\pi_1)}} \right| \quad (3.16)$$

**Algoritmo 3.2:** Algoritmo de Programação Dinâmica Aproximada.

- 
- 1 Dados os conjuntos de índices  $\mathcal{I}_k$  e  $\mathcal{J}_k$ , de  $\mathbf{x}_k$  e  $\mathbf{u}_k$ , respectivamente, e respeitando as restrições. E  $\forall k \in \{0, 1, \dots, T-1\}$  Inicializa-se  $F_k$ .
  - 2 Dado a condição de contorno  $\mathbf{x}_0$  e a função  $\psi(\mathbf{x}_T)$ .
  - 3 Dado o número de iterações  $N_{it}$  e o intervalo de avaliação das políticas  $N_a$ .
  - 4 Dado o critério de parada  $\epsilon$ .
  - 5 **enquanto**  $n < N_{it}$  e  $\Delta < \epsilon$  **faça**
  - 6     Sorteio de um cenário:  $\tilde{\mathbf{y}} = \{\tilde{\mathbf{y}}_0, \tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_T\}$
  - 7      $\mathbf{x}_k \leftarrow \mathbf{x}_0$
  - 8     Faça  $k \leftarrow 0$
  - 9     **enquanto**  $k < T$  **faça**
  - 10          $\text{minimo} \leftarrow \infty$
  - 11          $\mathbf{u}^* \leftarrow 0$
  - 12         **para**  $j \in \mathcal{J}_k$  **faça**
  - 13             Dados  $\mathbf{x}_k$  e  $\mathbf{u}_{k,j}$ ,  $\forall$  discretizações de  $\mathbf{y}_k$  calcule todos  $\mathbf{x}_{k+1}$  possíveis. Com as probabilidades das informações aleatórias externas  $\mathbf{y}_k$ :  $p_k^{(\pi)}(\mathbf{y}_k | \mathbf{y}_{k-1})$
  - 14              $f_k \leftarrow \sum_{\mathbf{y}_k \in \mathcal{Y}_k} p_k^{(\pi)}(\mathbf{y}_k | \mathbf{y}_{k-1}) \left( c_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{y}_k) + F_{k+1}(\mathbf{x}_{k+1}) \right)$
  - 15             **se**  $f_k < \text{minimo}$  **então**
  - 16                  $\text{minimo} \leftarrow f$
  - 17                  $\mathbf{u}^* \leftarrow \mathbf{u}_{k,j}$
  - 18             **fim**
  - 19         **fim**
  - 20          $F_k(\mathbf{x}_k)^{(n)} \leftarrow (1 - \alpha_{n-1})F_k(\mathbf{x}_k)^{(n)} + \alpha_{n-1}\text{minimo}$
  - 21         Atualização da política:  $\pi(\mathbf{x}_{k,i}) \leftarrow \mathbf{u}^*$
  - 22         Transição de estado:  $\mathbf{x}_k \leftarrow \varphi(\mathbf{x}_k, \mathbf{u}^*, \tilde{\mathbf{y}}_k)$
  - 23          $k \leftarrow k + 1$
  - 24     **fim**
  - 25      $n \leftarrow n + 1$
  - 26     Atualização do passo  $\alpha$
  - 27     **se**  $n \bmod(N_a) = 0$  **então**
  - 28         | Avalia duas políticas consecutivas para vários cenários:  $\Delta$ .
  - 29     **fim**
  - 30 **fim**
  - 31 Avaliação final de  $\pi(\mathbf{x})$  para diversos cenários de variáveis aleatórias.
-

## 4 Detalhes de implementação

*Este capítulo tem o objetivo de indicar as escolhas adotadas na PDA implementada e procura fazer uma apresentação de algumas ferramentas adicionais utilizadas na solução do POR.*

A proposta principal da programação dinâmica (POWELL, 2007), PDA, é reduzir o custo computacional da PD inserindo um tipo de simulação markoviana, evitando a solução recursiva em todo o espaço de estado, reduzindo então um “laço” no algoritmo. Além disso, a PDA também se propõe reduzir o esforço no cálculo da esperança matemática na avaliação de cada par estado-controle. Para tal, é necessário reescrever o problema entorno de um estado intermediário determinístico, chamado de estado pós-decisão (estado virtual).

Uma forma de exemplificar o estado pós-decisão seria no caso de uma função de transição de estado simplificada<sup>1</sup> para o POR de uma única usina, Equação (4.1).

$$x_{k+1} = x_k + \xi \left( y_k - u_k \right) = \underbrace{x_k - \xi u_k}_{\text{estado pós-decisão}} + \xi y_k = \hat{\varphi}(x_k, u_k) + \xi y_k \quad (4.1)$$

Portanto o estado  $\theta_k := \hat{\varphi}(x_k, u_k)$  é determinístico e denota o estado levado imediatamente após a escolha da decisão. A abordagem de Powell da PDA inclui reescrever a equação recursiva entorno  $\theta_k$ , ao invés de  $\mathbf{x}_k$ . A vantagem dessa abordagem é que todas as equações se tornam determinísticas, não necessitando o cálculo da soma finita da esperança matemática. Porém, o requisito para usar o artifício do pós-estado é que o custo incremental  $c_k$  seja determinístico, dependa apenas do estado atual e da decisão, o que é incompatível com o modelo do Capítulo 2. No caso do POR,  $c_k$  depende do vertimento das usinas, que é tratado como aleatório.

Dessa forma, sem a possibilidade de usar o artifício da equação recursiva determinística, é necessário que um modelo estatístico seja assumido a priori, como foi feito no Capítulo 2. A vantagem de se assumir um modelo estatístico está relacionada com o passo de atualização  $\alpha$ , definido na Equação (3.15).

A definição de um passo estocástico  $\alpha$  é por si só um problema não trivial, e em muitos casos o passo possui parâmetros que precisam ser ajustados empiricamente. Em geral, o passo é uma função no número de iterações, já que nas primeiras iterações os valores experimentados pelo algoritmo não são muito precisos, pois a função  $F_k$  ainda foi pouco amostrada. Ou seja, existe um “transiente” para que o sistema comece a estimar os custos de forma correta. No entanto, esse transiente é muito rápido para sistemas em

<sup>1</sup> Assumindo que não há vertimentos, pois caso contrário precisaria ser levado em conta os dois casos da função de transferência do modelo do Capítulo 2, equações (2.7-2.9).

que é possível calcular explicitamente o valor da esperança matemática (POWELL, 2007). Assim foi adotado passos de  $\alpha_n = 1$ , ou seja, a atualização do valor específico  $F_k(\mathbf{x}_k)$  é a substituição por  $f_k$ , Equação (3.14).

## 4.1 Exploração x Interpolação

Um dos aspectos fundamentais da PDA é estabelecer um compromisso entre explorar o espaço de estado e utilizar a informação adquirida para tomar a melhor decisão. Uma das características do método é fazer o uso de simulações de cenários para percorrer os estados mais prováveis e, portanto, experimentar os valores de custos para as decisões disponíveis. De certa forma em algumas situações pode ser interessante passar por um estado específico, para determinar alguma informação ainda não coletada, essa é a ideia de exploração.

Por outro lado, permitir que o sistema explore muito o espaço de estado pode ser custoso, já que alguns estados não tem informações úteis ou que dificilmente ocorreriam. Em alguns casos a melhor abordagem é utilizar toda a informação já adquirida para produzir uma estimativa do que ainda não foi experimentada, fazendo a interpolação de estados com os valores de custos coletados dos estado visitados.

A exploração é efeito direto do número de iterações realizadas pelo algoritmo, e a interpolação usa a informação adquirida para estimar os estados ainda não explorados. Um dos desafios da PDA é estabelecer um compromisso entre o número de iterações até que se faça uma interpolação. Portanto, um novo parâmetro precisa ser inserido ao método, o número de iterações para ocorrer uma interpolação  $N_{ip}$ . Junto com o número de iterações totais do algoritmo ( $N$ ), com o número de iterações necessárias até a avaliação da política obtida ( $N_a$ ) e com o número de cenários usados nessa avaliação ( $N_{ava}$ ),  $N_{ip}$  forma os parâmetros necessários para a aplicação do PDA na solução do POR.

## 4.2 Métodos interpolativos

Nesta seção discute-se os métodos interpolativos, tanto para as funções  $F_k$  quanto para as próprias políticas  $\pi_k$ . As ferramentas utilizadas nesse trabalho foram duas: a interpolação linear utilizando malhas triangulares (triangulação de Delaunay) e a generalização por redes neurais. Ambas as abordagens são discutidas a seguir no contexto do POR com PDA.

### 4.2.1 Triangulação de Delaunay

A necessidade da generalização das interpolações lineares surge no contexto de ensaios bidimensionais que são descritos com maior detalhes na Seção (5.3). Uma das

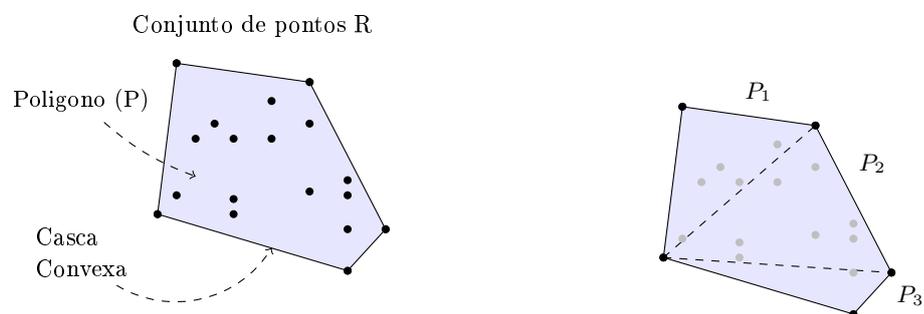
alternativas encontradas para realizar interpolações, sem assumir previamente qualquer comportamento das funções interpoladas, foi a triangulação de Delaunay, (BERG et al., 2008). O principal propósito das triangulações é a construções de malhas triangulares; ferramentas básicas em áreas como a computação geométrica, uma aplicação conhecida é no método de elementos finitos.

As definições a seguir são baseadas em argumentos geométricos bidimensionais. Porém, podem facilmente serem generalizadas para espaços maiores.

**Definição 4.1. Triangulação:** *Dado um conjunto finito de pontos  $R := \{r_1, r_2, \dots, r_n\}$  e  $\mathcal{T}(V, E)$  um grafo composto pelo conjunto de vértices e de arcos,  $V$  e  $E$ , de tal forma que seja planar e que possua o número máximo de arcos de tal forma que a adição de uma nova aresta não seja possível sem que o mesmo perca sua planaridade, ou seja que nenhuma de suas arestas se intersectem.*

*Dessa forma, define-se a triangulação do conjunto  $R$  de pontos como o grafo  $\mathcal{T}$  tal que o conjunto de vértices seja  $V = R$ .*

Uma argumentação pouco formal sobre a topologia do grafo  $\mathcal{T}$  é construída partindo da afirmação que para um conjunto  $R$ , com no mínimo três elementos, existe uma envoltória convexa que pode ser vista como um polígono  $P$ , Figura 4.1(a). Um resultado bastante importante sobre regiões convexas é que qualquer segmento de reta que as divide produz polígonos também convexas. Se forem tomadas as diagonais como segmentos divisores, Figura 4.1(b), então os polígonos resultantes terão número de arestas menores do que o original. Se esse procedimento for repetido, por indução, ele se mantém até que os polígonos resultantes sejam triângulos, completando assim a argumentação sobre a topologia de  $\mathcal{T}$ .



(a) Exemplo de envoltória convexa.

(b) Diagonais do polígono

Figura 4.1 – Conceitos introdutórios para triangulações no  $\mathbb{R}^2$ .

Portanto, baseado apenas nas argumentações anteriores, assegura-se a existência

da triangulação de uma região convexa. Outra definição bastante útil é a do chamado vetor de ângulos de uma triangulação, definição (4.2).

**Definição 4.2. Vetor de ângulos:** *Seja  $\mathcal{T}$  uma triangulação composta por  $M$  triângulos, então define-se o vetor de ângulos da triangulação como:*

$$A(\mathcal{T}) = \alpha = (\alpha_1, \alpha_2, \dots, \alpha_M)^T$$

Onde as componentes  $\alpha_i$  são ordenadas de forma crescente. Além disso, dois vetores de ângulos podem ser comparados através da relação lexicográfica. Ou seja, para duas triangulações de um mesma região convexa,  $\mathcal{T}$  e  $\mathcal{T}'$ , diz-se que  $A(\mathcal{T}) > A(\mathcal{T}')$  se existe  $i \in \{1, 2, 3, \dots, M\}$  tal que:

$$\alpha_j = \alpha'_j \quad \forall j < i \quad e \quad \alpha_i > \alpha'_i$$

É comum que alguns algoritmos de triangulação partam de uma malha arbitrária e a melhorem iterativamente. Uma das operações mais frequentes sobre o grafo  $\mathcal{T}$  é chamada operação de giro, que baseia-se no conhecido teorema de Thales, Figura 4.2, cujo resultado principal afirma que o ângulo formado por um ponto arbitrário fora da circunferência é menor do que um formado por um ponto interior à mesma. Esse resultado é bastante útil e junto com a Definição (4.3) permite construir um procedimento de melhora nas triangulações.

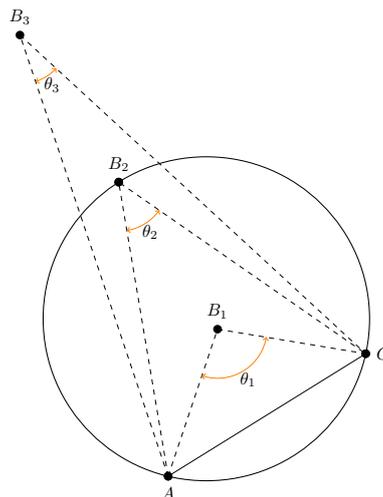


Figura 4.2 – Ilustração da teorema de Thales, cujo resultado afirma que  $\theta_3 < \theta_2 < \theta_1$ .

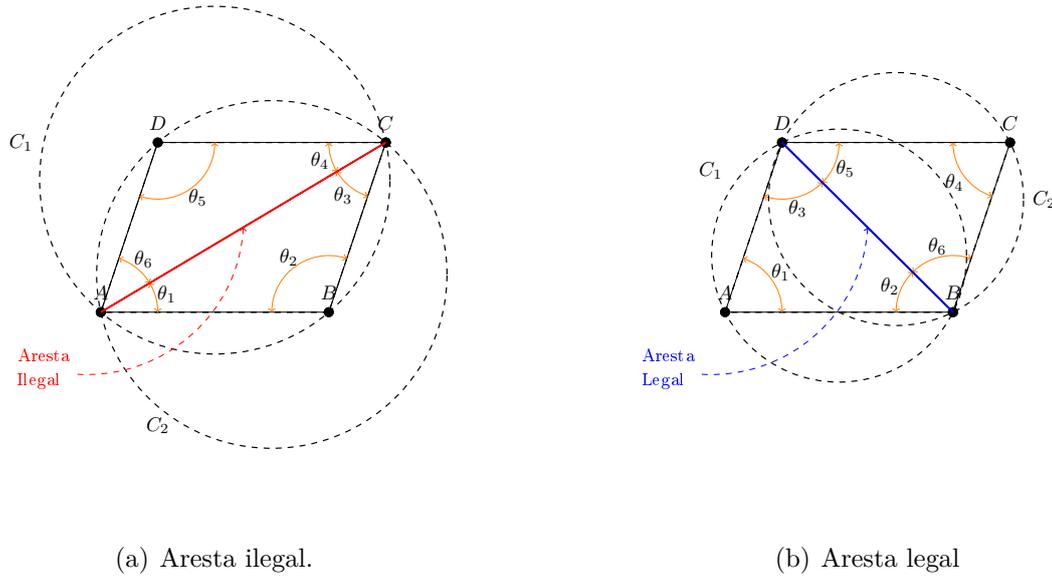


Figura 4.3 – Ilustração da operação de giro em algoritmos de triangulação.

**Definição 4.3. Aresta ilegal:** *Seja o quadrilátero  $ABCD$  e o círculo  $C_1$  que circunscreve o triângulo  $A\hat{B}C$ , então a aresta  $\overline{AC}$  é dita ser uma aresta ilegal se o vértice  $D$  está contido na circunferência  $C_1$ .*

Outro ponto bastante importante é que em um quadrilátero regular, se os seus quatro vértices não estão contidos na mesma circunferência, então das duas diagonais possíveis apenas uma é ilegal, o que garante que uma operação de giro sempre melhora uma aresta ilegal em um quadrilátero, Figura 4.3. Além disso, esse resultado, define a abordagem recursiva das triangulações (BERG et al., 2008), em que particionando um conjunto de pontos em subconjuntos de quatro pontos, constrói-se a triangulação completa.

Com os conceitos anteriores, (BERG et al., 2008) define uma triangulação de Delaunay usando o Teorema 4.1.

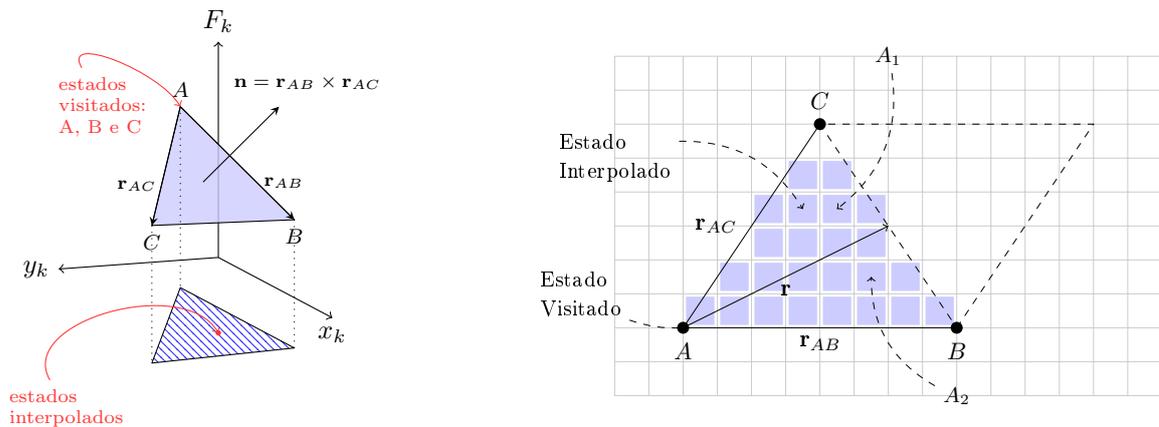
**Teorema 4.1.** *Dado um conjunto  $R$  de pontos. Uma triangulação  $\mathcal{T}$  é dita ser de Delaunay se qualquer circunferência que circunscreve qualquer triângulo desse grafo não contém nenhum outro ponto de  $R$  em seu interior. Em outras palavras, uma triangulação de Delaunay é uma triangulação dita legal, ou seja, não contém arestas ilegais.*

Uma consequência do algoritmo é que qualquer triangulação ótima em relação aos ângulos de seus triângulos, ou seja, que maximiza o menor ângulo da triangulação, é dita ser de Delaunay (BERG et al., 2008).

No contexto do POR, principalmente para algoritmos aproximados de PD, pode-se pensar no conjunto  $R$  como sendo o conjunto de estados do POR visitados ao longo do processo iterativo. Portanto adotando a notação  $R_k := \{\hat{\mathbf{x}}_k^{(i)}\}$  como o conjunto de estados

visitados no instante  $k$  na iteração  $i$ , tem-se que os estados visitados formam os vértices da triangulação e podem ser usados para interpolar os estados ainda não visitados. O intuito dessa associação fica claro na Figura 3.9 transportada para o contexto das triangulações.

A interpolação de estados no interior da região definida pelos estados visitados,  $A$ ,  $B$  e  $C$ , é feita de forma que dado os três pontos (e o valor da função  $F_k$  de custo acumulado), o produto vetorial entre os vetores definidos por  $AB$  e  $AC$  define a equação do plano que passa por esses pontos,  $(\hat{\mathbf{x}}_k, F_k)$ . A equação do plano é usada para atualizar  $F_k(\mathbf{x})$  os estados no interior da região definida por esses três pontos, Figura 4.4(a).



(a) Interpolação linear de estados não visitados.

(b) Interpolação de regiões discretas

Figura 4.4 – Ilustração da operação de interpolação usando as triangulações de Delaunay.

É importante ressaltar que a malha triangular a priori poderia ser qualquer uma. Porém as de Delaunay são úteis pelo fato de produzirem triângulos com ângulos maiores, ou seja, não tão estreitos. Essa propriedade ajuda em métodos que utilizam discretizações, dado que pode-se produzir regiões com maior número de estados a serem interpolados de uma vez, Figura 4.4(b). Além disso, essa relação estabelece um compromisso comum em PDA, já que muitas iterações irão produzir triangulações com muitas regiões, o que pode ser custoso, e poucos triângulos podem resultar em uma interpolação de baixa qualidade.

Para a implementação computacional das malhas triangulares foram utilizadas as bibliotecas na linguagem PYTHON em especial o método `scipy.spatial.Delaunay`, integrante da biblioteca `Qhull`. Segundo a documentação da biblioteca, os algoritmos de envoltória convexa são baseados no algoritmo `QuickHull` e, no caso de aplicações de triangulação, é utilizado um resultado bastante importante da área de geometria computacional, que é o fato das triangulações de Delaunay em  $\mathbb{R}^n$  poderem ser determinadas obtendo-se a envoltória convexa em  $\mathbb{R}^{n+1}$  da projeção de seus pontos em um parabolóide (BARBER; DOBKIN; HUHDANPAA, 1996).

Nas interpolações lineares o teste para pontos internos de uma região triangular é necessário; ele consiste em um teste simples baseado em geometria analítica. Da Figura

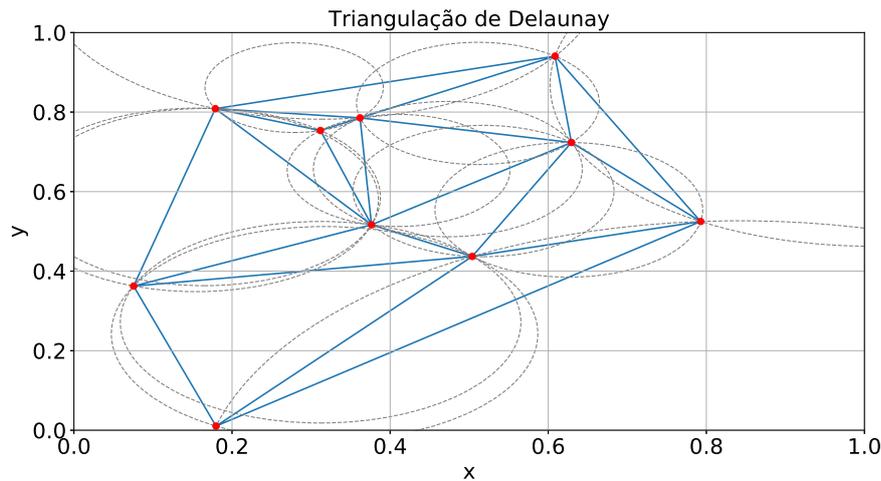


Figura 4.5 – Exemplo do uso da biblioteca na produção de malhas triangulares. Nota-se que nenhum círculo (tracejado) definido pelos triângulos da malha contém pontos (vermelho) do conjunto  $R$  utilizado na triangulação.

4.4(b), se forem tomados dois vetores  $\mathbf{r}_{AB}$  e  $\mathbf{r}_{AC}$ , linearmente independentes, então os coeficientes da combinação linear da base formada por eles para um vetor qualquer dado ( $\mathbf{r}$ , entre um ponto arbitrário  $P$  e o vértice  $A$ , origem) é facilmente obtida pela solução de um sistema linear:  $\beta = \mathbf{A}\mathbf{r}$ , onde  $\beta$  é o vetor com os coeficientes da combinação e  $\mathbf{A}$  a matriz com os vetores da base dispostos em colunas. O teste de pontos internos em  $\mathbb{R}^2$  se resume ao fato de que um ponto  $P$  na diagonal oposta do paralelogramo, formado pelos vetores  $\mathbf{r}_{AB}$  e  $\mathbf{r}_{AC}$ , divide a área total do triângulo ( $A_{\Delta}$ ) tal como na Equação (4.2).

$$\begin{aligned}
 A_{\Delta} &= \frac{1}{2} \|\mathbf{r}_{AB} \times \mathbf{r}_{AC}\| = \frac{1}{2} \|\mathbf{r}_{AB} \times \mathbf{r}\| + \frac{1}{2} \|\mathbf{r} \times \mathbf{r}_{AC}\| = A_1 + A_2 \\
 &= \frac{1}{2} \|\mathbf{r}_{AB} \times (\beta_1 \mathbf{r}_{AB} + \beta_2 \mathbf{r}_{AC})\| + \frac{1}{2} \|(\beta_1 \mathbf{r}_{AB} + \beta_2 \mathbf{r}_{AC}) \times \mathbf{r}_{AC}\| \\
 &= \frac{1}{2} \beta_2 \|\mathbf{r}_{AB} \times \mathbf{r}_{AC}\| + \frac{1}{2} \beta_1 \|\mathbf{r}_{AB} \times \mathbf{r}_{AC}\| \\
 A &= \beta_1 A + \beta_2 A \quad \beta_1 + \beta_2 = 1
 \end{aligned} \tag{4.2}$$

Portanto, se o ponto não estiver sob a diagonal oposta e sim no interior do triângulo, então a condição de desigualdade não se mantém e é substituída por  $\beta_1 + \beta_2 \leq 1$ . Também é geometricamente óbvio que  $\beta_1 > 0$  e  $\beta_2 > 0$ , pois se não fossem o ponto  $P$  estaria na direção oposta dos vetores, fora do paralelogramo definido.

## 4.2.2 Redes Neurais

As redes neurais artificiais (RNA) são estruturas de dados modeladas a partir da mimetização do sistema nervoso dos seres humanos. Essas arquiteturas são conhecidas por

buscar descrever sistemas com características complexas, não-lineares e com capacidade de computação paralela. As redes neurais, apesar de sua computação massiva de informações, são compostas de unidades de processamento bastante simples (neurônios) capazes de armazenar conhecimento, disponibilizando-o e generalizando-o (HAYKIN, 2009).

O conhecimento adquirido por essas redes é armazenado pelas conexões realizadas por cada neurônio da rede, que se comunicam dependendo do tipo de arquitetura da rede. Essas conexões possuem pesos sinápticos que refletem o conhecimento da rede. Além disso, o conhecimento é externo a rede e é apresentado a partir do sistema na qual a mesma está inserida, e que é frequentemente referenciado como ambiente. O ajuste desses pesos sinápticos é conhecido como processo de treinamento, ou processo de aprendizagem (HAYKIN, 2009). Os benefícios das redes neurais no contexto do POR proposto é a capacidade de tratar não linearidades, estabelecimento de um mapeamento de entrada-saída, e, principalmente, a capacidade de generalização dessas estruturas, ou seja, a capacidade de produzir saídas desconhecidas baseadas nas evidências apresentadas a ela no processo de aprendizagem.

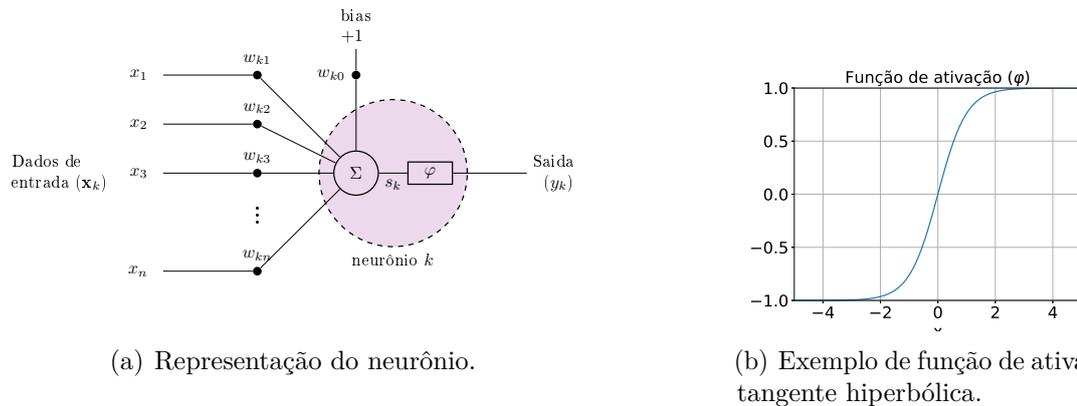


Figura 4.6 – Representação das variáveis do bloco de processamento fundamental em RNAs, o neurônio.

A informação apresentada a um neurônio, Figura 4.6(a), corresponde aos sinais de entrada (pertencente a  $\mathbb{R}^n$ ) e cada componente  $i$  é multiplicada pelo respectivo peso sináptico do neurônio  $k$ ,  $w_{ki}$ . Cada componente é adicionada pela unidade aditiva do neurônio produzindo a saída  $s_k$  (o termo  $b_k$  é conhecido como *bias*), Equação (4.3). A função  $\varphi$  concede a rede as características necessárias para uma dada aplicação.

$$s_k = \sum_{i=1}^m w_{ki}x_i + b_k \quad (4.3)$$

$$y_k = \varphi(s_k) \quad (4.4)$$

O ajuste dos pesos sinápticos, processo de treinamento da rede, afim de responder a um estímulo externo do ambiente e se adaptar ao novo meio, pode ser diverso. Porém, um dos mais conhecidos é chamado de aprendizagem por correção de erro, onde se busca pelos pesos sinápticos que produzam uma saída  $y_k$ , Equação (4.4), que se aproxime de um valor esperado (ou uma resposta) conhecido; um gabarito  $\hat{y}_k$ . Ou seja, se for definida a função de erro, equação (4.5), cujo índice  $i$  denota neurônios na camada de saída da rede e  $n$  o  $n$ ésimo par de sinais de entrada-saída. Então busca-se por uma matriz  $\mathbf{W}$  de pesos sinápticos que minimize o erro médio de uma população de amostras de entrada-saída de tamanho  $N$ , Equação (4.6).

$$E_n = \frac{1}{2} \sum_{i \in \mathcal{I}} e_{i,n}^2 = \frac{1}{2} \sum_{i \in \mathcal{I}} (\tilde{y}_{i,n} - y_{i,n})^2 \quad (4.5)$$

$$\mathbf{W} = \arg \min_{w_{ki}} E = \arg \min_{w_{ki}} \frac{1}{N} \sum_{n=1}^N E_n \quad (4.6)$$

O conhecimento do valor esperado de resposta para a RNA, estabelece um dos paradigmas de aprendizagem de rede chamado de aprendizado supervisionado (informalmente, vamos denominar o aprendizado supervisionado de aprendizado com um professor), em que a rede não possui um conhecimento prévio sobre o ambiente, porém há um professor capaz de dizer a ela qual a resposta correta para um determinado estímulo, Figura 4.8(a). Dessa forma, o professor apresenta a rede um par de entrada-saída esperada para que a os pesos possam ser ajustados e, após treinamento, a rede com base no que experimentou, possa prever comportamento de estímulos antes não apresentados, (HAYKIN, 2009).

Para cada aplicação com redes neurais, seja na associação ou reconhecimento de padrões, aproximação de funções, controle e filtragem, etc., uma estrutura conexionista diferente é recomendada. No caso de aproximação de funções (principal objetivo no trabalho), a estrutura mais comum é a chamada *perceptron* de múltiplas camadas (MLP); uma estrutura formada por camadas de neurônios onde todo neurônio se conecta a todos os outros da camada seguinte. Nessas redes as camadas se resumem a uma de entrada, saída e uma quantidade variável de camadas intermediárias, chamadas de ocultas, Figura 4.7, com número distinto de neurônios.

No caso das MLPs, no contexto do aprendizado supervisionado, tem-se um conjunto  $R$  de amostras coletadas do ambiente na qual a rede está inserida, o professor se encarrega de classifica-las, ou atribuir a elas a resposta esperada da rede na presença de cada uma delas, Figura (4.8(a)). Já no contexto do POR, o par entrada-saída corresponde ao estado visitado pela PDA no instante  $k$  ( $\hat{\mathbf{x}}_k$ ) e a decisão ótima ( $\hat{\mathbf{u}}_k$ ), ou o valor aproximado do custo  $F_k$ . Portanto a PDA funciona como o professor da rede.

A principal estratégia de otimização de redes neurais é o bastante conhecido algoritmo de retro-propagação, baseado em algoritmos do tipo gradiente descendente. Se

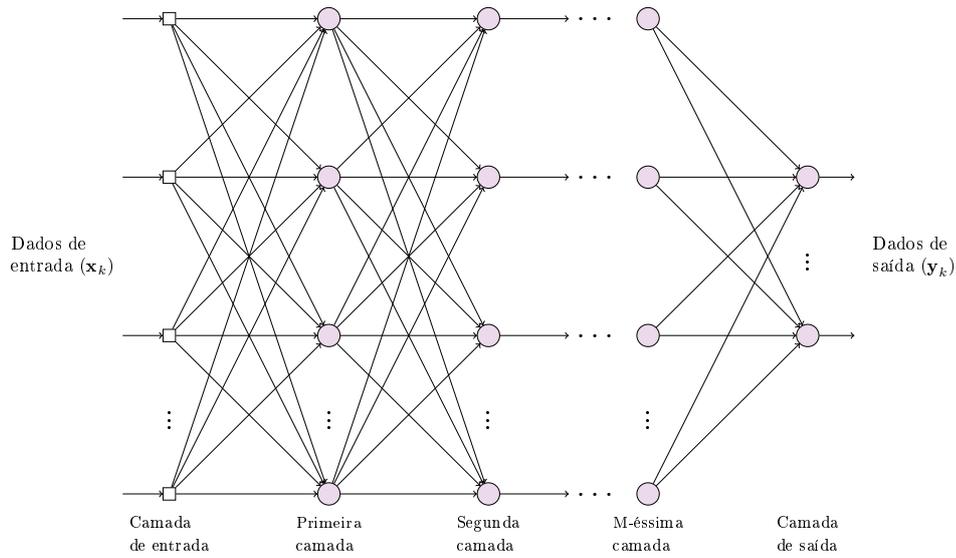
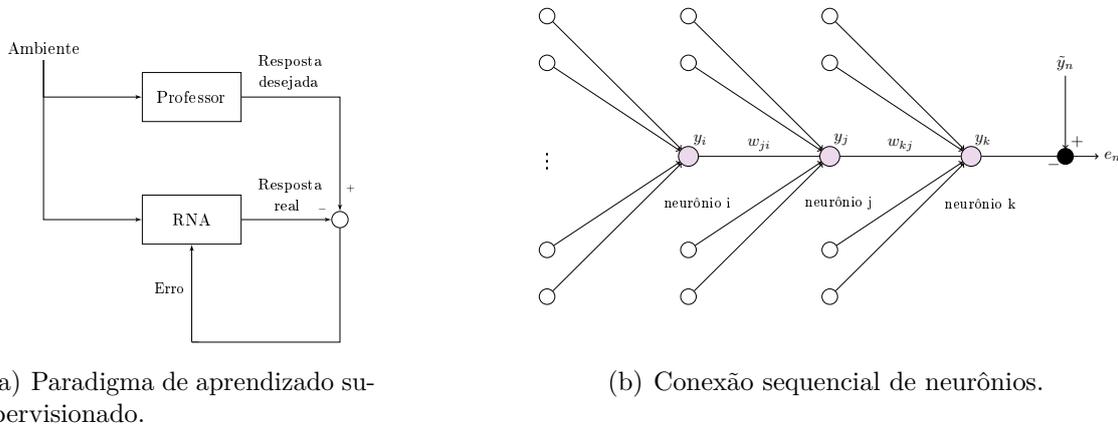


Figura 4.7 – Rede neural do tipo *perceptron* de múltiplas camadas (MLP).



(a) Paradigma de aprendizado supervisionado.

(b) Conexão sequencial de neurônios.

Figura 4.8 – Exemplificação do paradigma de aprendizado e representação das conexões de neurônios sequenciais.

$w_{ij}$ , Figura 4.8(b), corresponde ao peso sináptico que conecte a saída do neurônio  $i$  a entrada do  $j$ , então para a  $n$ ésima amostra do conjunto  $R$ , tem-se que o gradiente do erro, em relação aos pesos, se dá pela Equação (4.7).

$$\frac{\partial E_n}{\partial w_{ij}} = \underbrace{\frac{\partial E_n}{\partial e_j}}_{Eq.4.5} \underbrace{\frac{\partial e_j}{\partial y_j}}_{=-1} \underbrace{\frac{\partial y_j}{\partial s_j}}_{\varphi'} \underbrace{\frac{\partial s_j}{\partial w_{ij}}}_{Eq.4.3} = \underbrace{-e_j \varphi'}_{\delta_j} y_i \quad (4.7)$$

A variação dos pesos sinápticos pode ser definida como na Equação (4.8), o termo  $\delta_j$  é chamado de gradiente local, por representar a direção de descida parcial de um dos

pesos sinápticos do neurônio.

$$\Delta w_{ij} = \eta \frac{\partial E_n}{\partial w_{ij}} = -\eta e_j \varphi' y_i = \eta \delta_j y_i \quad (4.8)$$

A definição formal do gradiente local, pode ser dada pela Equação (4.9), que para os neurônios  $j$  da camada de saída resultam em uma variação dos pesos sináptico como na Equação (4.8).

$$\delta_j = -\frac{\partial E_n}{\partial s_j} \quad (4.9)$$

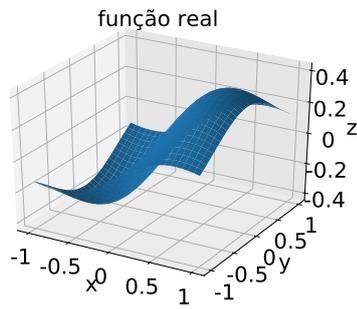
No caso de neurônios nas camadas ocultas intermediárias, não há um definição para o erro  $e_j$ , pois não se conhece o valor esperado de saída para esses neurônios. Portanto o gradiente local não possui uma dependência com a variável de erro  $e_j$ . A atualização dos pesos continua como na Equação (4.8). Porém, para neurônios intermediários a Equação (4.10), Figura 4.8(b), deve ser utilizada no cálculo do gradiente local.

$$\begin{aligned} \delta_j &= -\frac{\partial E_n}{\partial y_j} \frac{\partial y_j}{\partial s_j} = -\frac{\partial E_n}{\partial y_j} \varphi' \\ &= -\varphi' \frac{\partial}{\partial y_j} \left\{ \frac{1}{2} \sum_i e_i^2 \right\} = -\varphi' \sum_i e_i \frac{\partial e_i}{\partial y_j} = -\varphi' \sum_i e_i \underbrace{\frac{\partial e_i}{\partial s_i}}_{Eq.4.5} \underbrace{\frac{\partial s_i}{\partial y_j}}_{Eq.4.3} \\ &= -\varphi' \sum_i \underbrace{e_i \varphi'_i}_{\delta_i} w_{ij} = -\varphi'_j \sum_i \delta_i w_{ij} \end{aligned} \quad (4.10)$$

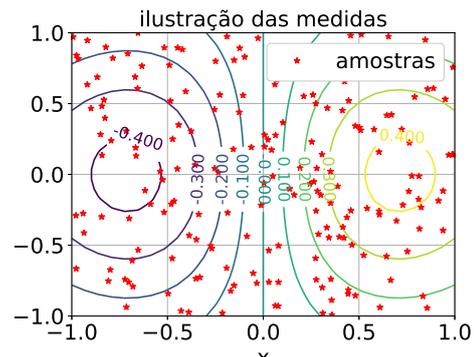
O algoritmo de retro-propagação é assim denominado justamente por ser definido em dois passos: no primeiro deles o sinal de entrada da amostra é apresentado e cada sinal funcional (saída de cada neurônio,  $s_j$ ) é calculado até a camada de saída, onde é comparado com o sinal desejado  $\tilde{y}_k$ . No segundo passo, com o erros  $e_j$  definidos, percorre-se no sentido contrário a rede, definindo cada um dos gradientes locais, podendo, assim definir o  $\delta w_{ij}$ .

A biblioteca que auxiliou na aplicação de RNAs foi `sklearn.neural_network` desenvolvida e disponibilizada pela `Sci-kit Learn`, em especial a classe `MLPRegressor`, também em linguagem `PYTHON` (PEDREGOSA et al., 2011). Uma ilustração da aplicação do pacote oferecido é apresentado na Figura 4.9, onde é aproximada uma função não linear, com uma amostragem de 200 pontos. Além disso, exibe o comportamento do erro da estimativa em um mapa de cores.

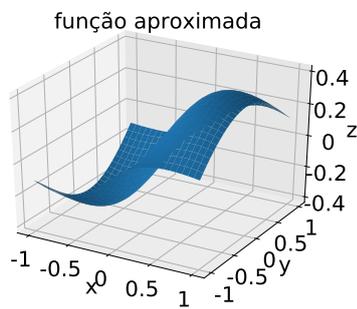
A classe `MLPRegressor` utiliza uma otimização similar anteriormente apresentada nas Equações (4.7-4.10), e diferindo apenas por usar o logaritmo da função de erro. O algoritmo de treinamento foi o do gradiente estocástico descendente, opção `sgd`. Os



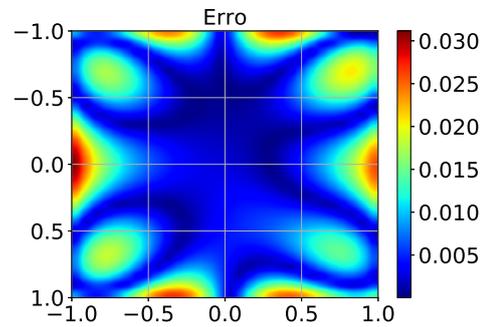
(a) Função original a ser aproximada.



(b) Amostragem da função.



(c) Função aproximada.



(d) Distribuição do erro.

Figura 4.9 – Solução do problema de aproximação de funções por MLPs.

parâmetros importantes no treinamento da rede foram deixados como os padrões pre-definidos pelo pacote: número de dados para validação mantido em 10%, função de ativação como tangente hiperbólica e taxa de aprendizado ( $\eta$ ) constante em um valor de 0,01. O critério de parada do pacote por *default* é de  $10^{-4}$  (erro em relação a fração de amostras não apresentadas a rede do conjunto de dados, validação cruzada).

### 4.3 Considerações sobre o POR no contexto da PD

Antes de dar início a seções experimentais, é interessante reforçar alguns aspectos relevantes da iteração PD e POR. A primeira delas, é que o algoritmo de programação dinâmica aproximada utiliza o cenário sorteado apenas para fazer os saltos de  $\mathbf{x}_k$  para  $\mathbf{x}_{k+1}$  e nada mais. As vazões utilizadas no cálculo da esperança matemática da função (3.14) são as vazões prováveis de ocorrer e são resultantes da discretização da densidade de probabilidade.

A interpretação da PD como um problema de malha fechada, Figura 3.8, onde  $\mathbf{u}_k + \mathbf{y}_k$  é a informação relevante para a mudança de estado, Equação (2.7): a decisão  $\mathbf{u}_k$ , volume turbinado pelo reservatório, é escolhido antes de saber a realização da vazão  $\mathbf{y}_k$  entre os meses  $k$  e  $k + 1$ , o que tira qualquer viés do algoritmo. Ou seja não se sabe qual é

a vazão (mesmo que a mesma já tenha sido sorteada) antes que se tome a decisão.

Como discutido no Capítulo 2, as vazões de cada reservatório seguem distribuições log-normais, portanto sem perda de generalidade pode-se reescrever a equação recursiva de otimalidade em função da variável aleatória  $\mathbf{Z}_k$  (logaritmo das vazões), Equação (4.11), novamente utilizando a discretização das funções densidade de probabilidade e condicionando o conjunto do logaritmo das vazões  $\mathcal{Z}_k$  a um conjunto finito. O motivo para isso é simples, já que  $\mathbf{Z}_k$  segue uma distribuição normal a discretização fica simétrica e os valores de probabilidade também o que melhora a aproximação da esperança inclusive para grandes discretizações.

$$f_k(\mathbf{x}_k) = \min_{\mathbf{u}_k \in \mathcal{U}_k} \sum_{\mathbf{z}_k \in \mathcal{Z}_k} p_k(\mathbf{z}_k | \mathbf{z}_{k-1}) \left( c_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{z}_k) + F_{k+1}(\varphi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{z}_k)) \right) \quad (4.11)$$

No caso das vazões markovianas, a densidade de probabilidade no próximo instante depende da realização no instante anterior, ou seja a densidade de  $\mathbf{Z}_k$  depende da realização  $\mathbf{z}_{k-1}$ . Portanto, a informação da vazão realizada precisa ser mantida como uma variável de estado. Ou seja para um reservatório que segue vazões markovianas de ordem um, a variável de estado é definida pela Equação (4.12). Nesse caso, mesmo para uma única usina, o estado é bidimensional o que justifica as ilustrações das Figuras (3.9-3.11).

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ z_{k-1} \end{bmatrix} \quad (4.12)$$

No Capítulo 3, foram mencionados estado proibitivos, no caso dos reservatórios isso ocorre em estados onde o volume do reservatório está próximo do limite inferior. Nestes casos, levando em consideração que há uma turbinagem mínima  $\underline{u}$ , quando  $\underline{u} > \exp(\mu_k + 3\sigma_k)$  implica que qualquer vazão sorteada fará com que mais água saia do que chegue ao reservatório e, portanto, viole o limite do reservatório, caso o mesmo se encontre abaixo do volume mínimo permitido.

## 5 Ensaios e Experimentos

*O capítulo procura apresentar as principais simulações e justificativas das escolhas sobre a PDA. O estudo de casos para as simulações foi o reservatório de Ilha Solteira, um reservatório no sistema sudeste e um dos maiores na bacia do rio Paraná.*

### 5.1 Estudo de casos

O estudo de caso do trabalho discute a aplicação da programação dinâmica para o reservatório de Ilha Solteira<sup>1</sup> em modelos de vazões independentes e markovianas.

As características físicas do reservatório podem ser encontradas no trabalho de Lyra Filho (1984). As estatísticas das vazões mensais foram calculadas utilizando o sistema de acompanhamento de reservatórios (SAR) da Agência Nacional de Águas (ANA, 2018). Os valores de cada constante e restrição do reservatório podem ser encontrados na Tabela 1. Já os valores das distribuições são apresentados na Tabela 2, assim como as demandas contratadas para cada mês. As distribuições dos dados de vazões também podem ser observadas na Figura 5.1.

Tabela 1 – Características físicas do reservatório de Ilha Solteira

$\bar{x}[Mm^3]$	$x[Mm^3]$	$\bar{u}[m^3/s]$	$\underline{u}[m^3/s]$	$P_{max}[MW]$	$\alpha_k[BI\ R\$/MW^2]$	$\eta$
21200	9743	7955	1400	3444	$2,6 \cdot 10^{-6}$	0,88
$a_0$	$a_1$	$a_2$	$a_3$	$b_0$	$b_1$	
303,04	$155,19 \cdot 10^{-5}$	$-173,77 \cdot 10^{-10}$	0	279.84	$221,305 \cdot 10^{-6}$	

Tabela 2 – Demandas contratadas  $d_k[MW.mes]$  e distribuições das vazões  $y_k[m^3/s]$ .

k	0	1	2	3	4	5	6	7	8	9	10	11	
ln $y_k$	$\mu$	8,98	9,03	9,04	8,77	8,41	8,24	8,01	7,83	7,74	7,76	8,12	8,65
	$\sigma$	0,46	0,41	0,31	0,30	0,22	0,24	0,20	0,20	0,26	0,36	0,27	0,35
	$\rho$	0,38	0,33	0,64	0,72	0,87	0,93	0,83	0,65	0,54	0,50	0,67	0,24
$d_k$	2800	2500	2300	2500	2600	2800	2800	2600	2600	2800	3000	3100	

Os ensaios foram realizados para o número de discretizações dos estados  $n_x$  e de controles  $n_u$  de 100, assim como para as vazões, tanto para o cálculo da esperança matemática quanto para as vazões do estado aumentado. A maioria dos trabalhos de POR com PD citados no Capítulo 1 usam discretizações bem baixas, da ordem de 10 valores. O estado inicial do reservatório no mês de janeiro foi de  $x_0 = 15000Mm^3$  e exigindo que o mesmo, ao fim do horizonte se encontrasse acima desse valor,  $x_T > 15000Mm^3$ .

<sup>1</sup> Todas as rotinas foram codificadas em PYTHON 3.6.0 e as simulações realizadas em um PC com Intel Core i7-4790 CPU executando em 3.60GHz, 8Gb RAM e sistema operacional Ubuntu 16.04.

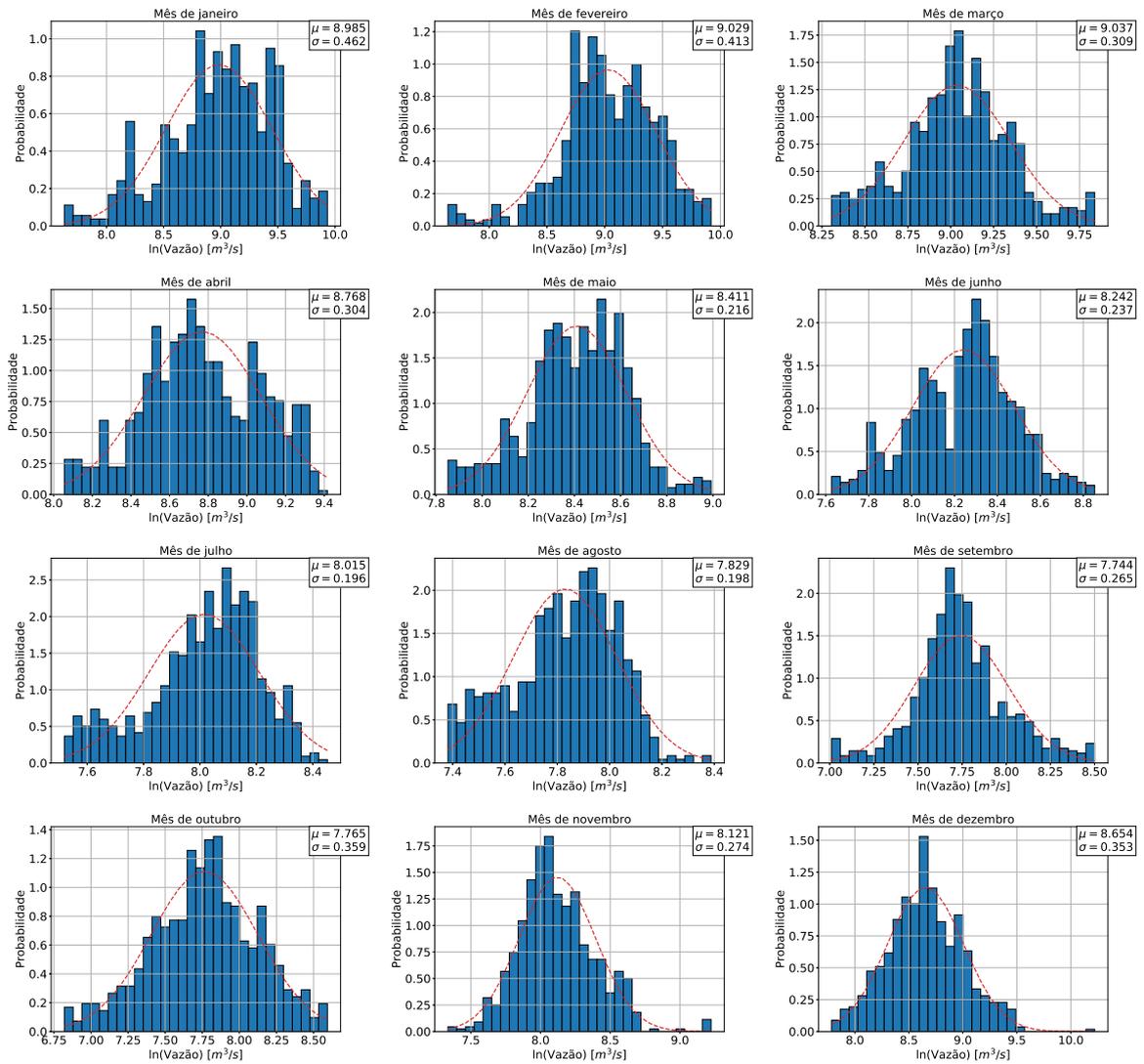


Figura 5.1 – Distribuições dos dados retirados do banco de monitoramento da ANA para Ilha Solteira.

## 5.2 Ensaio com vazões independentes

O primeiro modelo de vazões utilizado, correspondente a vazões independentes ( $\rho_k = 0$ ), implica na PDE que apenas o volume do reservatório compõe a variável de estado ( $\mathbf{x}_k = x_k$ ); portanto, a política para o reservatório é uma função  $\pi_k : \mathbb{R} \rightarrow \mathbb{R}$ . Como mencionado no Capítulo 3, para a solução pelo método tradicional, parte-se do mês de dezembro ( $k = 11$ ) até o instante inicial, janeiro ( $k = 0$ ), onde conhecesse o estado inicial  $\mathbf{x}_0$ , solucionando para cada estado discreto a equação recursiva (4.11).

A solução para Ilha Solteira, com os dados históricos de vazões e utilizando PDE, pode ser observada na Figura 5.2. Na figura em questão, nota-se doze funções, onde o eixo abscisso representa o estado do reservatório em termos do valor discretizado<sup>2</sup>. As curvas

<sup>2</sup> O valor 100 corresponde ao valor do reservatório completamente cheio ( $\bar{x}$ ) e 0 o valor mínimo armazenado ( $\underline{x}$ ).

em verde correspondem a solução seguindo o procedimento *backward* da PD (PDE).

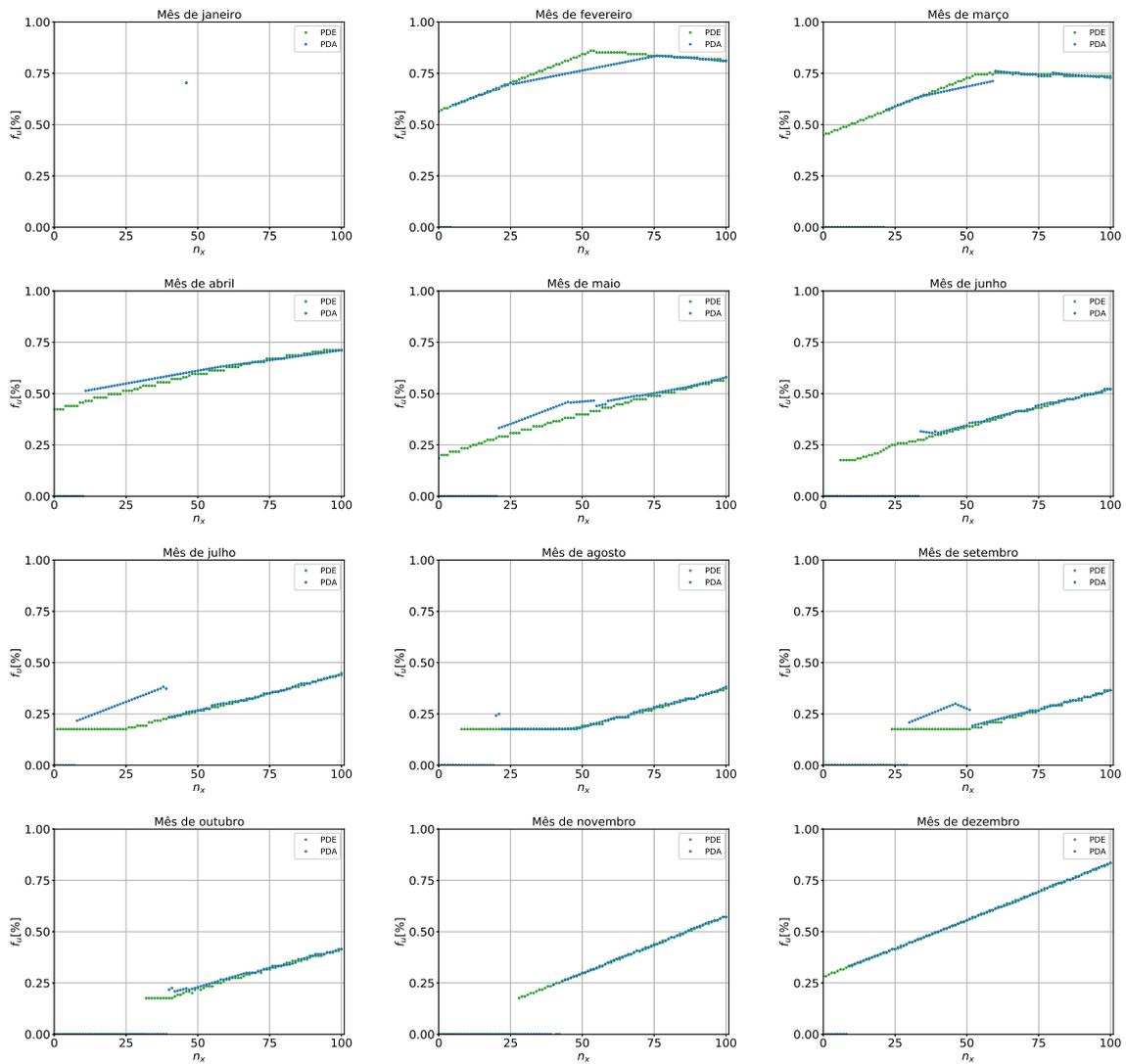


Figura 5.2 – Comparativo entre as políticas da PDE (verde) e da PDA (azul) de janeiro a dezembro.

As políticas da Figura 5.2 fornecem o principal resultado para operar o reservatório, já que fornecem as regras para comanda-lo (quanto defluir) dependendo do volume estocado de água. Porém, uma outra informação bastante relevante sobre o problema corresponde a distribuição de custos que essa política ótima produz. A Figura 5.3 exhibe a distribuição de custos com a complementação térmica para um ano seguindo as políticas da PDE. O valor esperado da distribuição é de **42,31u.m.**. O tempo computacional necessário para obter essa política foi de 13 segundos.

Por outro lado, seguindo a abordagem aproximada da PD (PDA), parte-se do estado inicial e no sentido *forward* ao longo dos meses, constrói-se iterativamente as políticas para o reservatório. Novamente na Figura 5.2, as funções, agora em azul, exibem as regras produzidas. Tanto para PDE quanto para a PDA, nota-se que para alguns meses a função

não está definida para alguns pontos. Por exemplo, para o mês de outubro os estados discretos menores que 25 não possuem decisão ótima, essas situações correspondem a estado sem decisão que respeite as restrições do problema. No mês em questão, os estados inferiores (reservatório vazio) não conseguiriam, por exemplo, respeitar a restrição do reservatório terminar o ano acima de um limiar seguro, por exemplo.

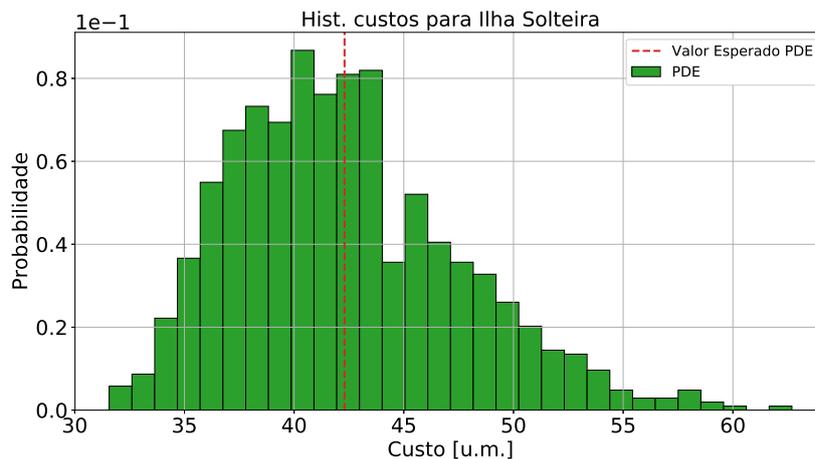


Figura 5.3 – Distribuição de custos para a política determinada segundo a PDE para Ilha Solteira com vazões independentes.

Nota-se algumas descontinuidades na política da PDA, a razão possui relação direta com a ideia básica do algoritmo aproximado de frequência de visitação dos estados do sistema. Na PDA a função de custo e política é sempre atualizada quando o reservatório experimenta algum estado em uma determinada iteração. Assim, se o reservatório não passar por algum estado ou demorar a amostrar a informação de alguma região a função pode apresentar uma informação pouco relevante (ou confiável), e tende a ter um comportamento diferente dos estados mais visitados, como descontinuidades. Portanto, essas descontinuidades, ilustram o efeito de estados pouco frequentados.

Na PDA uma informação bastante útil está relacionada a medida de quanto um estado do sistema é frequentado. A Figura 5.4 ilustra a distribuição de vezes que cada estado discreto do reservatório foi amostrado. O que se nota é um comportamento bastante esperado: os primeiros meses, que correspondem aos mais chuvosos e, portanto, com maiores vazões naturais, fazem com que independente da decisão escolhida, o reservatório seja levado a experimentar o estado de máximo volume. Nos meses intermediários, como são mais secos, possuem estados intermediários com frequência mais relevante em comparação com o início do ano.

Assim como na PDE, na PDA também pode-se buscar pela distribuição de custos produzida pela política, que pode ser encontrada na Figura 5.5, onde a função  $\pi$  dos dois algoritmos foram avaliadas para 1000 cenários de vazões possíveis. Para a PDA, obviamente que a política final depende do número de iterações; portanto na Figura 5.5(a)

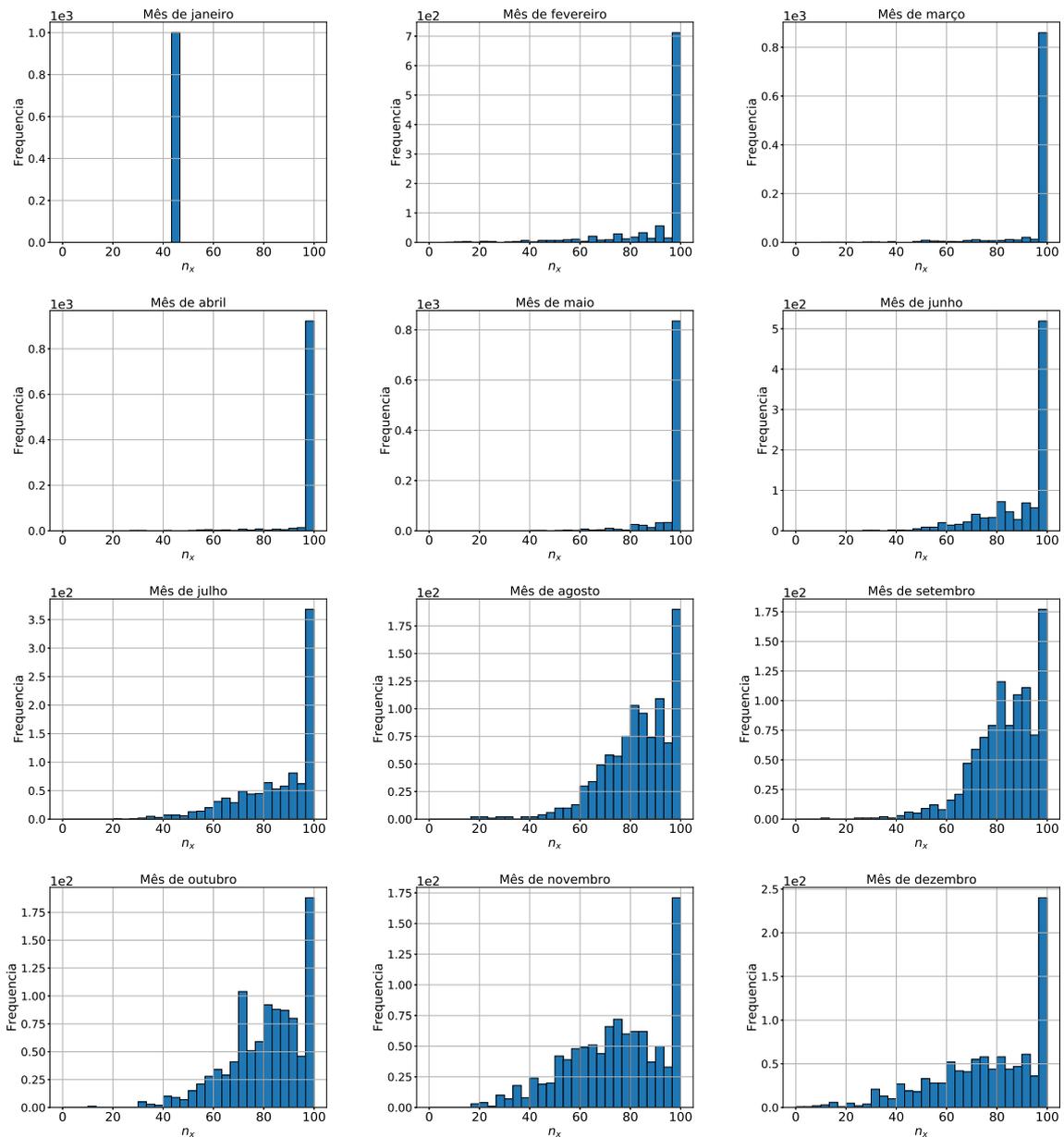


Figura 5.4 – Frequência de visitação dos estados no algoritmo da PDA de janeiro a dezembro.

a PDA foi iterada em um total de 100 vezes, enquanto que na Figura 5.5(b) em 1000, que corresponde à distribuição das políticas da Figura 5.2. Nota-se que o algoritmo para esse estudo de caso consegue produzir uma distribuição muito próxima da ótima e exata (PDE) se for permitido o número suficiente de iterações, mesmo possuindo políticas ligeiramente diferentes em alguns meses.

Apesar da convergência da solução apresentada, 1000 iterações necessitam de 180 segundos, portanto 10 vezes mais do que a versão clássica da PD. É possível justificar essa demora devido à frequência para explorar alguns estados relevantes do sistema, da Figura 5.4, observa-se que o sistema experimenta muitas vezes o reservatório cheio. Portanto,

a política se atualiza muito rápido nesses pontos, mas nos meses finais a informação de estados correspondentes a níveis mais baixos ainda é relevante para a construção da regra de defluência, e o algoritmo precisa contar com a “sorte” para amostrar esses pontos. Em outras palavras, no caso da PD, estados no instante  $k$  precisam da informação do estados  $k + 1$  para tomar a melhor decisão (nas duas versões), e portanto precisam esperar não apenas passar por estados menos frequentes em  $k$  mas precisam contar com o acaso e terem atualizado corretamente os de  $k + 1$  também, causando uma espécie de atraso na convergência devido baixas visitasões.

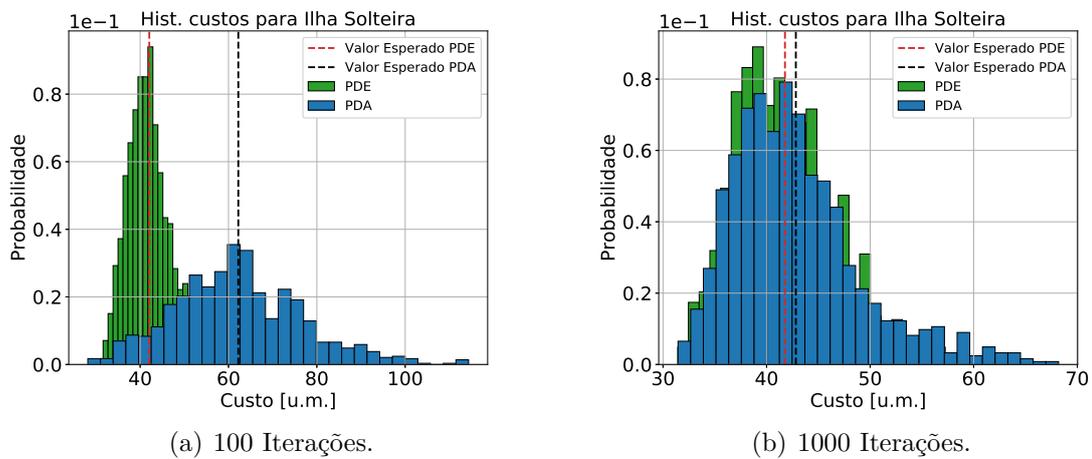


Figura 5.5 – Comparação entre as distribuições de custos variando o número de interações do método aproximado e a distribuição ótima (PDE).

Ainda sobre os ensaios com diferentes iterações da PDA, a Tabela 3 apresenta os tempos de execução de cada ensaio, tanto para PDE, quanto para a PDA. Nota-se que para apresentar o mesmo tempo de execução, foram necessárias 100 iterações, que corresponde a política que possuía distribuição de custos da Figura 5.5(a).

Tabela 3 – Resultados parciais do método para o problema unidimensional.

<i>metodo</i>	$N_{it}$	$\mathbb{E}(f.o.)$ [u.m.]	$t[s]$	$N_{resol}$	interpolação	gap [%]
PDE	-	42,31	13	1200	-	-
PDA	100	63,40	14	1200	não	50
PDA	300	47,84	51	3600	não	13
PDA	1000	43,17	180	12000	não	2

Na Tabela 3 a coluna *gap* corresponde a diferença entre a PDA e PDE relativa à solução ótima da PDE e  $N_{resol}$  o número de vezes que a equação recursiva foi solucionada.

Baseado nos resultados anteriores é que surge a necessidade das ideias expostas na Seção 4.1 e 4.2. No caso, se for utilizada uma simples interpolação linear entre dois estados visitados para determinar os intermediários tem-se resultados bastante expressivos.

Para cada 30 interações, realizando interpolações obtém-se uma política com distribuição com valor de esperado com *gap* de 3% em 13 segundos.

Não somente as ideias de interpolação são interessantes de serem analisadas nesse ponto da discussão. Um outro aspecto bastante relevante na PDA é sobre a inicialização das funções de custos e políticas ( $F_k$  e  $\pi_k$ ) que em algumas aplicações podem ajudar a colocar o algoritmo em um melhor ponto de partida na curva de convergência. No ensaio de vazões independentes foram utilizadas as inicializações nulas e utilizando a solução do problema determinístico com vazões iguais as médias das distribuições da Figura 5.1.

Na Figura 5.6 observa-se três versões da PD, a primeira delas (curva em azul) a versão puramente exploratória, a segunda (curva tracejada em preto) com uma inicialização das funções usando o problema determinístico e a terceira (curva cheia em preta) utilizando uma interpolação linear simples. O que se nota é que a inicialização coloca o problema em vantagem de partida, porém pouco se sustenta com o decorrer do algoritmo, colocando o recurso de interpolação em evidência.

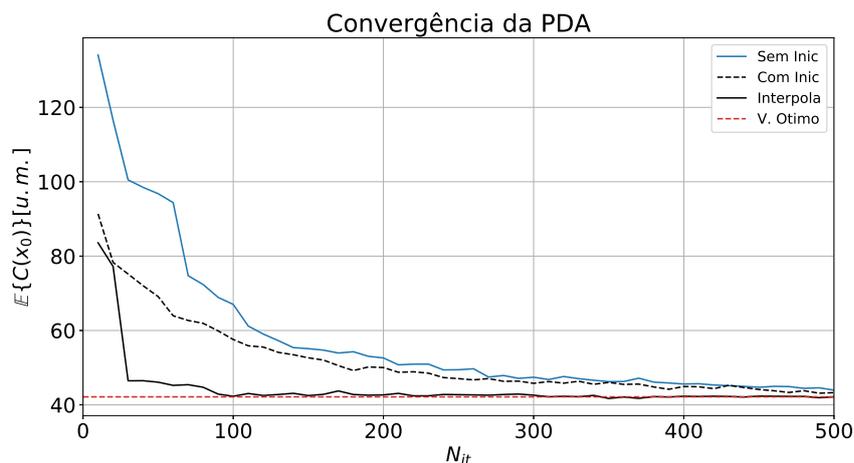


Figura 5.6 – Comparação entre as abordagens para o problema de POR não markoviano.

Apesar da inicialização da função no ensaio não ter surtido tanto efeito, não significa que é um recurso pouco relevante e sim que apenas não foram exploradas inicializações mais eficazes. Na Figura 5.6, têm-se que para menos de 100 iterações o algoritmo aproximado com interpolações consegue convergir para uma boa solução, contra 500 da versão exploratória. A Figura 5.2 ilustra políticas com recurso de interpolação, nessa versão, junto com o fato da política ser linear por partes, a visitação de duas extremidades do espaço de estados (reservatório cheio e vazio) em alguns casos já é suficiente para traçar uma boa política.

Apesar do ensaio de vazões não markovianas ser um modelo não tão próximo de uma situação real, ele ajuda a ilustrar os principais conceitos e pontos necessários para se caminhar para um algoritmo de fato eficiente de PDA, ilustrando conceitos, como frequência de visitação de estados, distribuição de custos gerados pela política, inicializações

do algoritmo e principalmente de ferramentas interpolativas. O principal resultado do ensaio é a necessidade de um método interpolativo associado a PDA no POR; o que justifica e sustenta todo o conteúdo da Seção 4.2 e o ensaio a seguir.

## 5.3 Ensaio com vazões markovianas

O ensaio com vazões markovianas, onde o estado do reservatório passa a ser bidimensional (volume armazenado e a vazão realizada no mês anterior), corresponde ao modelo mais realista do POR. Um cenário onde algum mês possui vazão acima da média tende a ter uma probabilidade alta dos outros meses também terem esse comportamento, diferentemente do ensaio anterior onde dois meses consecutivos podem ter vazões nos extremos de suas distribuições.

### 5.3.1 PDE

A solução do problema por PDE não difere em nada do ensaio anterior, com exceção do tamanho do espaço de estado. Para o POR com vazões markovianas de Ilha Solteira por PDE foram obtidas as políticas exibidas na Figura 5.7. É importante notar a similaridade entre as Figuras 5.7 e 5.2, funções lineares por partes. Assim como anteriormente, os eixos  $xy$  correspondem ao índice da discretização dos estados e o eixo  $z$  corresponde à decisão, em fração da máxima capacidade de defluência da barragem.

Sobre as políticas também é interessante notar o “degrau” em algumas funções, principalmente entre os meses de junho e dezembro, que correspondem a regiões ineficazes. Estas regiões surgem pela falta de controles factíveis, já que o reservatório possui um valor mínimo ( $u$ ) de turbinagem. A origem desses degraus em meses intermediários ocorre quando a distribuição de probabilidade de transição  $\mathbf{x}_{k+1}$  pode violar a restrição inferior mesmo para a mínima defluência da barragem.

Além da Figura 5.7, os resultados 5.8 e 5.9 completam a solução. A primeira delas ilustra a distribuição de custos produzidas pela política obtida, para 1000 cenários possíveis de vazões, e a segunda ilustra o valor esperado (em porcentagem da demanda) da energia produzida e complementada por Ilha Solteira para cada mês. A solução por PDE apresentou valor esperado de custos, Figura 5.8, de **50,64u.m.** e foi obtida em 1962 segundos.

### 5.3.2 PDA

Da mesma forma do ensaio anterior, um resultado interessante sobre o POR é a análise dos estados prováveis do sistema. A Figura 5.11 faz a representação bidimensional desse conceito. Nos mapas da figura em questão observa-se as regiões visitadas pelo

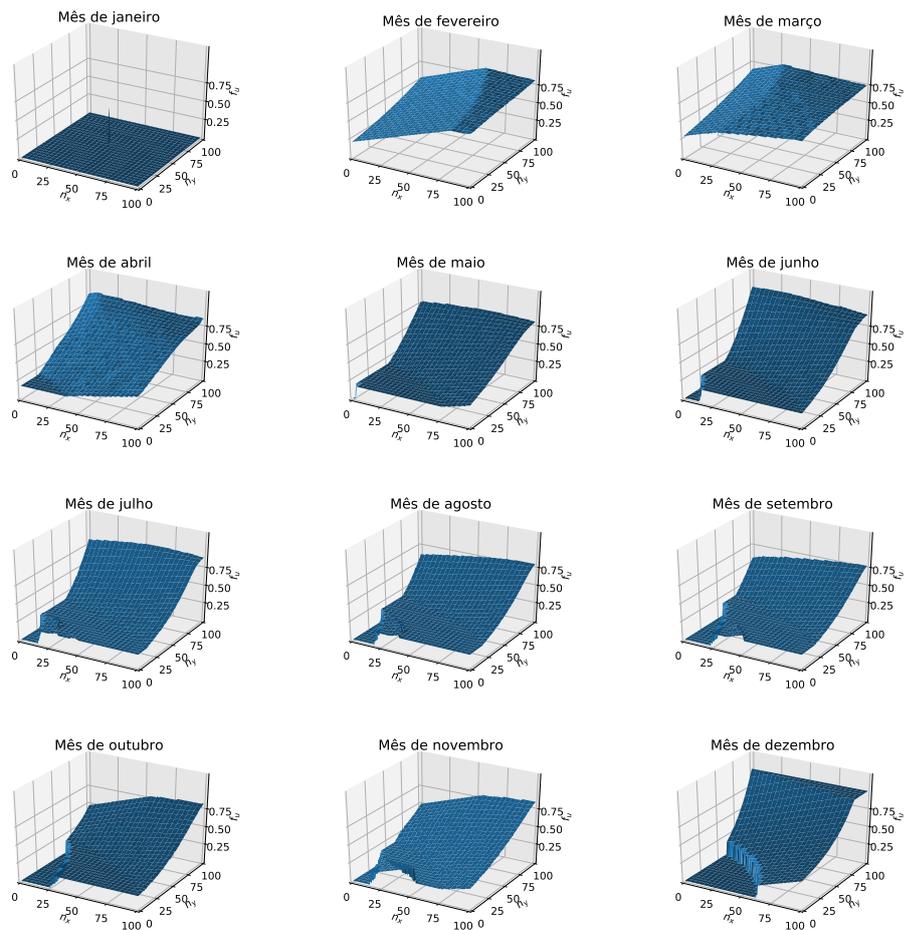


Figura 5.7 – Política de Ilha Solteira definida pela PDE com vazões markovianas.

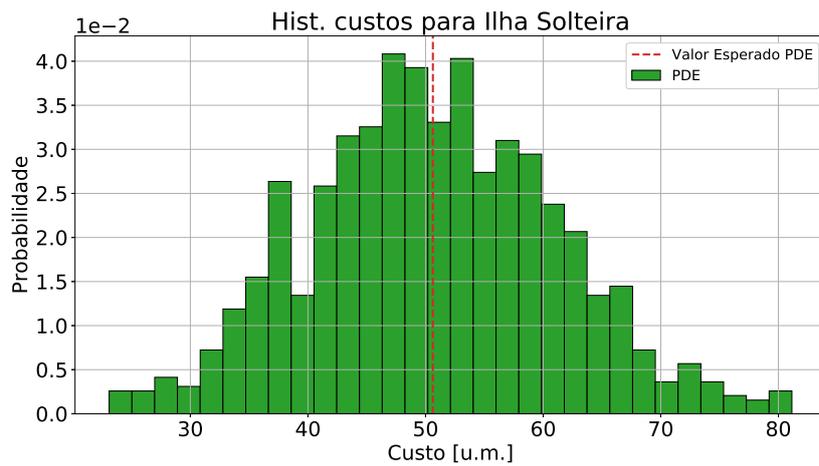


Figura 5.8 – Distribuição de custos para a política determinada segundo a PDE para Ilha Solteira com vazões markovianas.

reservatório, além disso, a figura ilustra o principal conceito da PDA, que é identificar regiões de baixa probabilidade (em tons mais fortes de azul) e evitar a solução da equação recursiva nesses pontos. Um caso fácil de identificar, e que não é evitado na PDE, são as

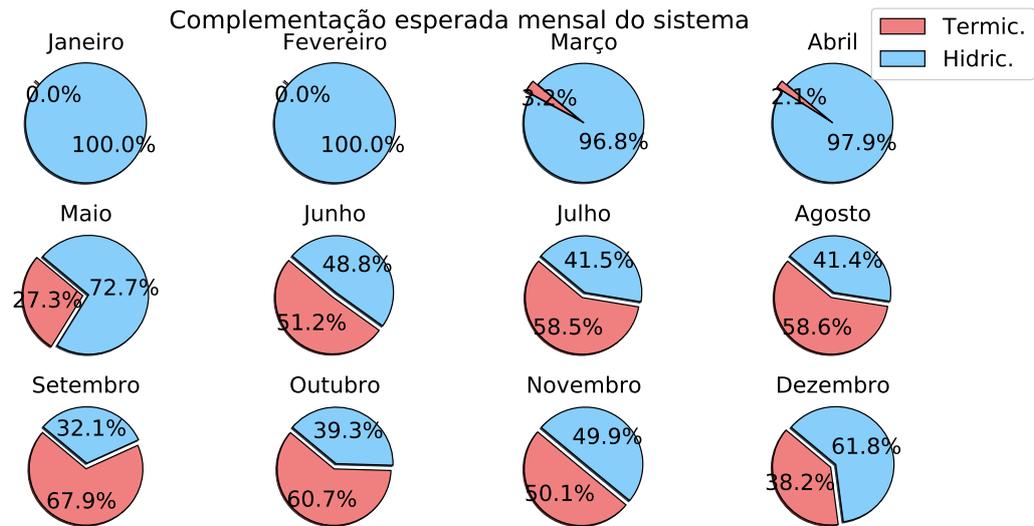


Figura 5.9 – Valores esperados da energia produzida mensalmente por Ilha Solteira e sua complementação termoelétrica.

situações em que o reservatório se encontra cheio ( $n_x = 100$ ) e com vazões naturais muito baixas ( $n_y = 0$ ), o que é uma combinação pouco provável como ilustrado na Figura 5.10.

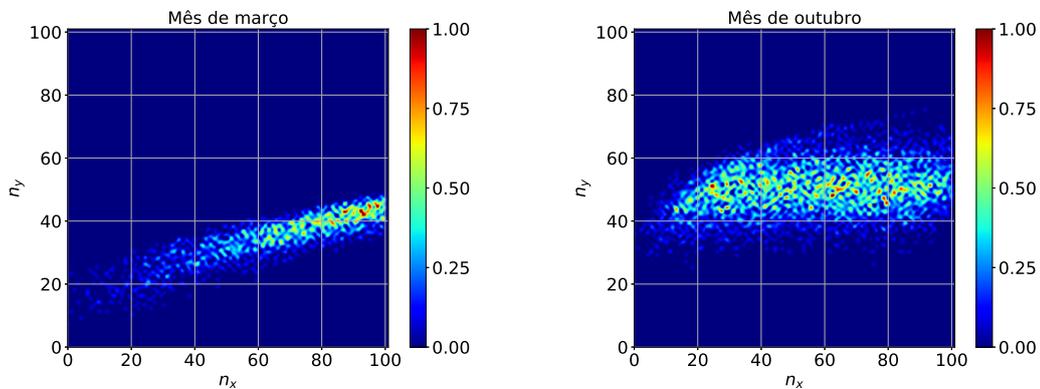


Figura 5.10 – Dois comportamentos típicos no espaço de estados: meses com abundância em água (março), frequência de visitação estreita e próximas do reservatório cheio, em contraste com meses escassos (outubro) com distribuição central e “uniforme”.

Outra observação sobre a similaridade entre as análises de frequência, é a intensidade das visitações em cada mês. Assim como no cenário unidimensional da Seção 5.2, existem três grupos de comportamento: nos primeiros meses a visitação se resume a uma distribuição estreita e com estados onde o reservatório alcança o limite superior independente do controle (meses de alta afluência, de janeiro a abril); no segundo, os meses intermediários (mais secos, maio a outubro), tendem fazer o reservatório visitar uma porção central do espaço de estado mais uniformemente, e ao fim um comportamento intermediário aos outros dois

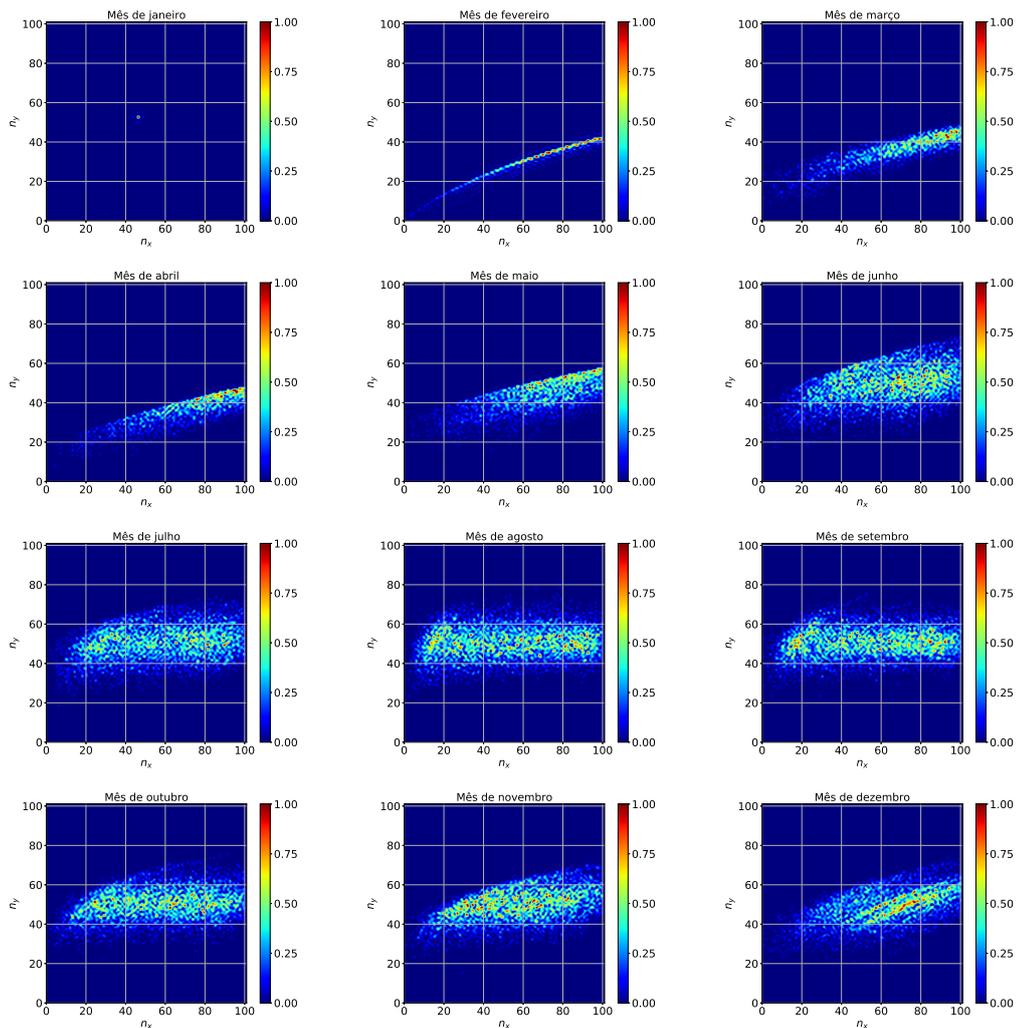


Figura 5.11 – Mapas de frequência de estados para o cenário markoviano: eixos  $xy$  correspondem aos estados discretos de volume e vazão e o mapa de cores a frequência de visitaç o de cada estado, com escala normalizada pelo estado mais frequente de cada m es.

(novembro e dezembro).

### 5.3.3 PDA com triangulaç o de Delaunay

A necessidade de um algoritmo com recurso interpolativo surge no estudo expresso na Seç o 5.2, por m mais que isso, a necessidade de um m todo que seja capaz de, se necess rio, interpretar e tratar as n o diferenciabilidade de “dobras” e “degraus” que pudessem aparecer como os das pol ticas da Figura 5.7. Por isso,   que a triangulaç o de Delaunay   sugerida, para criar uma interpolaç o linear tamb m por partes.

A Figura 5.12 ilustra a pol tica obtida com a abordagem com triangulaç es. Por m, o mais relevante dessa pol tica   a distribuiç o de custos que ela   capaz de produzir. Na Figura 5.13, onde para 1000 cen rios de vaz o s o comparadas as distribuiç es da PDE e

PDA com triangulações, visualmente nota-se uma alta similaridade. No histograma da PDA obteve-se um resultado de um valor esperado de **52,63u.m.** com 5000 iterações, a cada 200 realizando interpolações com as informações coletadas, custando um total de 1070 segundos. Ou seja, obteve-se uma solução com *gap* de 3,9% com redução de 45% no tempo de execução.

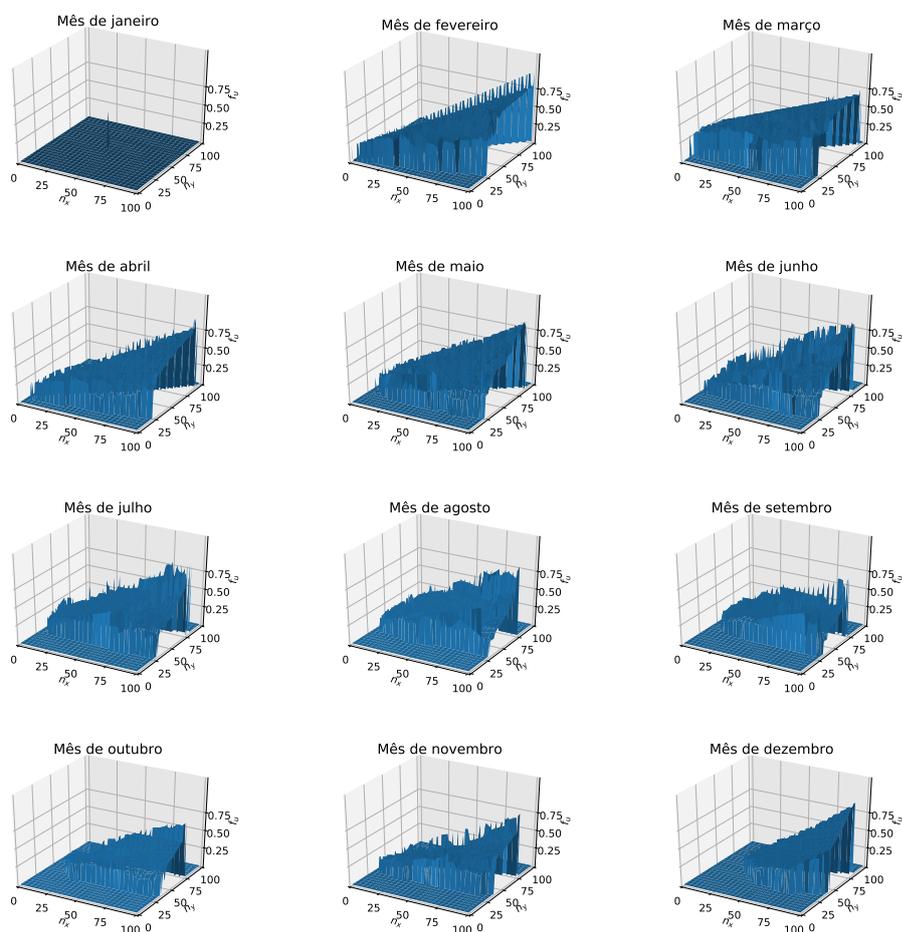


Figura 5.12 – Política de Ilha Solteira definidas pela PDA com interpolações lineares usando triangulações de Delaunay com vazões markovianas.

Pode-se notar que apesar da diferença visual entre as políticas das Figuras 5.7 e 5.12, elas produzem soluções de distribuição de custos similares. É fácil justificar a visual diferença das políticas, já que os algoritmos percebem os dados do mês seguinte  $k + 1$  de forma oposta: na PDE todo conteúdo de  $k + 1$  já está disponível em  $k$  enquanto que esse índice é iterado de forma decrescente, já a PDA não tem esse privilégio, montando sua política com o que está disponível e com  $k$  incremental crescente. Portanto, o caso mais frequente é gerar políticas distintas (o que é reforçado nos próximos resultados), sendo a similaridade da Figura 5.2 o caso especial. Outro argumento para a similaridade de políticas em ensaios de vazões independentes e markovianas, dá-se pelo fato do espaço de estados do primeiro experimento ser muito menor do que no segundo, dando maior chance

da PDA construir uma política similar à PDE.

Além disso, segundo (BELLMAN, 1957), não existe garantia de unicidade em todo problema de PD, apesar da garantia de convergência para o ótimo no método clássico. Por isso, não há porque se esperar políticas idênticas.

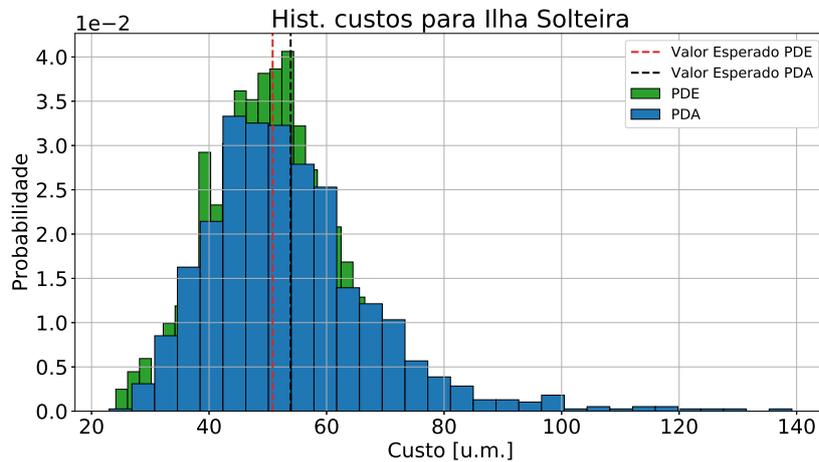


Figura 5.13 – Comparação entre as abordagens para o problema de POR markoviano.

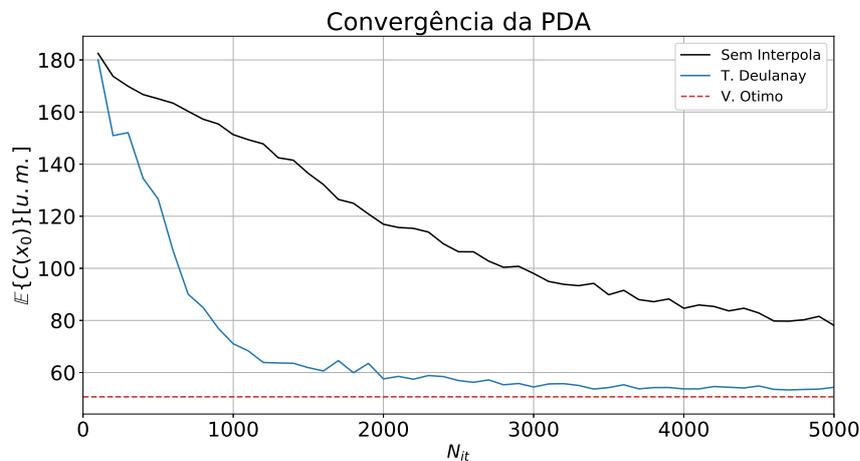


Figura 5.14 – Convergência das abordagens sem e com interpolações por triangulações para o POR markoviano.

Sobre a convergência do algoritmo, a Figura 5.14, ilustra o comportamento em função do número de iterações. A curva em vermelho representando o valor ótimo segundo a PDE, a curva cheia em preto o algoritmo da PDA sem nenhuma ferramenta interpolativa e a curva em azul usando as triangulações com interpolações a cada 200 iterações. Nota-se uma clara convergência para um valor relativamente próximo do ótimo muito mais acentuada do que o algoritmo livre de interpolações. Apesar do resultado anterior ter sido apresentado com 5000 iterações, antes o algoritmo já atinge um valor satisfatório, podendo o ganho em tempo ser superior aos 45%.

### 5.3.4 PDA com redes neurais artificiais

Assim como as triangulações, a sugestão do uso de RNA aparece como uma alternativa de interpolador. A indicação aparece principalmente pela flexibilidade de interpolação dessas estruturas e principalmente por sua generalização e aproximação universal. Além de também cuidar dos pontos de não diferenciabilidade, a sua aplicação a espaços maiores (além do  $\mathbb{R}^2$ ) é imediata.

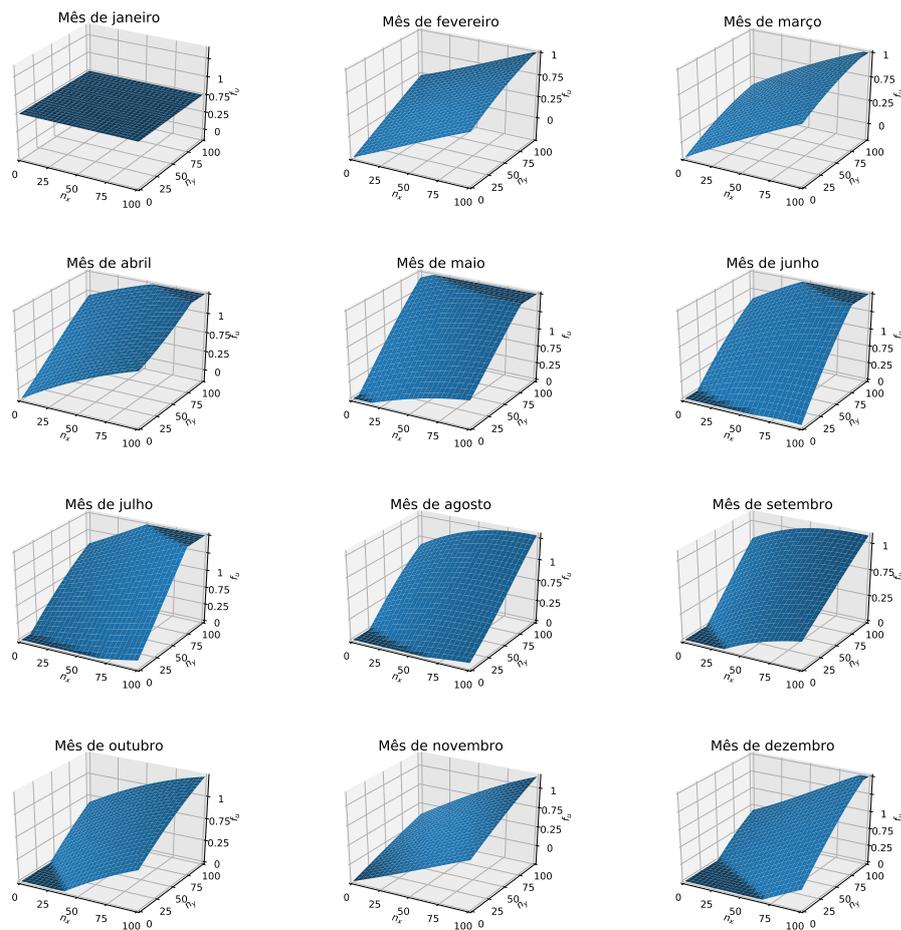


Figura 5.15 – Comparativo para as políticas para a PDE (verde) e para PDA (azul) de janeiro a dezembro.

Como nas outras aplicações, a estrutura dos resultados é idêntica. Na Figura 5.15, têm-se as políticas produzidas por uma rede neural bastante simples, apenas uma camada oculta de 5 neurônios. Assim como antes, o eixo  $z$  expressa a fração de turbinagem, limitada a um intervalo de 0 a 1, já que a rede pode produzir valores fora desse intervalo, principalmente em regiões de baixa visitação.

Os dois resultados fundamentais com as RNA são apresentados nas Figuras 5.16 e 5.17. Na Figura 5.16 ilustra-se mais uma vez a distribuição de custos da PDE e da PDA neural, e o resultado das distribuições é muito compatível com o valor esperado da versão aproximada, alcançando um valor esperado de **52,01u.m.** em um tempo computacional

baixo, 155 segundos. Ou seja, alcança-se um resultado com *gap* de 2,7% com redução de tempo de 92%. A Figura 5.17 ilustra a convergência para as três abordagens de PDA e como a versão neural, com apenas uma interpolação após 300 iterações, é suficiente para colocar o algoritmo muito próximo do ótimo rapidamente.

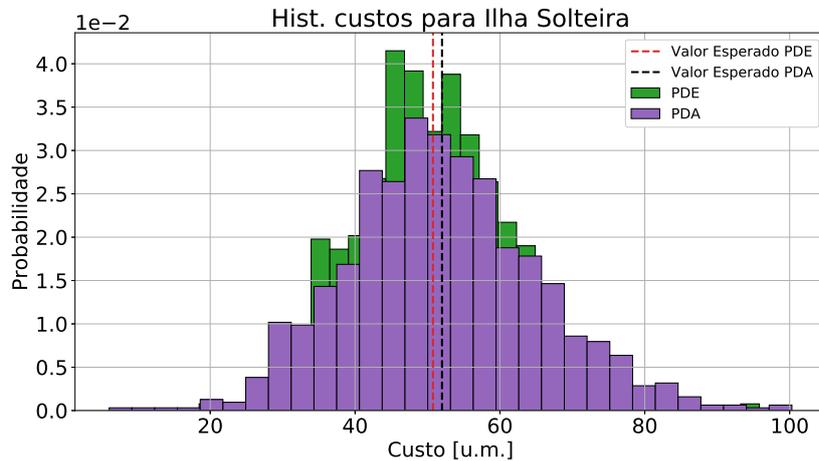


Figura 5.16 – Comparação entre as abordagens para o problema de POR não markoviano.

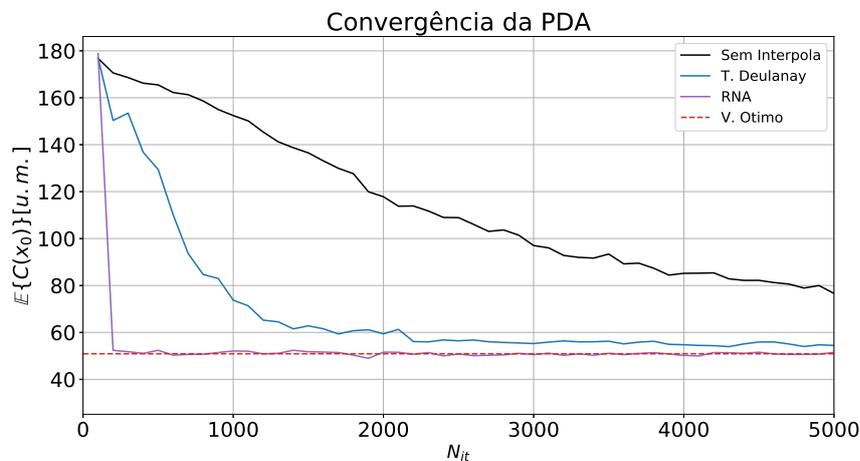


Figura 5.17 – Comparação entre as abordagens para o problema de POR não markoviano.

A superioridade da PDA com RNAs em relação à triangulação se dá principalmente pelo fato de que nas triangulações os estados precisam estar em uma região definida por três outros visitados. Enquanto isso na abordagem neural, desde que os estados visitados sejam capazes de indicar o comportamento padrão da política, a generalização dos outros ocorre com bastante qualidade (mesmo para estados relativamente distantes). Em outras palavras, nas triangulações um estado de baixa visitação ainda tem uma espera até ser devidamente atualizado; diferentemente da RNA, que consegue com informação de qualidade, inferir de uma forma mais efetiva e massiva, pelo menos para a instância simulada, sobre o comportamento de todo o espaço de estado apenas aprendendo com o padrão de amostras.

Os resultados para ambas as abordagens podem ser observados em sua totalidade na Tabela 4. Além disso, a PDA apresentou convergência em todos os casos, mesmo sem interpolações. Porém se mostrou competitiva nos casos onde os estados visitados foram utilizados para avaliar o espaço não coberto pelas iterações realizadas.

A programação dinâmica estocástica de horizonte finito de  $T$  intervalos de tempo e com  $n$  estados precisa resolver a equação recursiva de otimalidade um total de  $n(T - 1)$  vezes. Enquanto um algoritmo aproximado como o sugerido nesse trabalho necessita de um total de  $N_{it}T$ . É muito bem conhecido que  $n$  cresce exponencialmente com a dimensão do problema. Porém, esse crescimento não é único no caso aproximado. O que extrai-se dos ensaios usando a PDA no POR é que  $N_{it}$  depende tanto do número de estados quanto da abordagem interpolativa. Além disso, os ensaios mostraram que o crescimento da dimensão pode ser contornado com o bom uso de uma estratégia de interpolação (redes neurais, por exemplo), com a PDA.

Tabela 4 – Resultados finais do método para o problema bidimensional.

<i>metodo</i>	$N_{it}$	$\mathbb{E}(f.o.)$ [u.m.]	$t[s]$	$N_{resol}$	interpolação	gap [%]
PDE	-	50,64	1962	1100000	-	-
PDA	10000	66,28	2602	120000	Sem	31
PDA	10000	52,63	2602	120000	Delanauy	4
PDA	5000	52,65	1070	78000	Delanauy	4
PDA	300	52,01	155	3000	RNA	3

Em relação aos dois métodos utilizados notou-se que há interpolações que são afetadas diretamente pelas distribuições das vazões. Por exemplo, nas triangulações, as distribuições geram estados pouco prováveis que precisam ou ser visitados para se tornarem pontos da triangulação ou caírem dentro da malha, para se produzir uma melhor interpolação. Diferentemente das RNAs que só com os estados experimentados conseguem cobrir todo o espaço de busca. No caso específico do POR apresentados, as RNAs se mostraram uma ferramenta poderosa, reduzindo em mais de 90% o tempo de processamento do algoritmo clássico.

Além dos ganhos em complexidade, as RNAs possuem duas vantagens sobre outros métodos clássicos de interpolação. A primeira delas é sua versatilidade em serem aplicadas a sistemas maiores; por exemplo, seria mais simples utiliza-las em um POR com duas usinas do que as triangulações. A segunda vantagem é a economia de recursos de memória; no caso da PDE e da PDA com abordagem tabular, os valores de  $\pi_k$  e  $F_k$  são armazenadas em "tabelas". Nos exemplos de vazões markovianas um total de 1000 valores precisam ser armazenados para cada mês e para cada função ( $24 \cdot 10^3$  no total), o que pioraria na adição de mais um reservatório (totalizando  $24 \cdot 10^7$ ). Por outro lado com a rede simples utilizada, apenas um total de 30 valores (15 para cada função) precisaram ser armazenados (pesos

da rede).

## 6 Conclusões e Perspectivas Futuras

A dissertação desenvolveu metodologia para ampliar as possibilidades de usar programação dinâmica na otimização do planejamento da geração de energia elétrica através de sistemas de usinas hidrotérmicas. Para isso, explorou a associação de ideias de programação dinâmica aproximada com triangulações de Delaunay e redes neurais artificiais, com objetivo de reduzir o número de avaliações da equação recursiva de otimalidade da programação dinâmica. No conjunto, a metodologia permitiu atingir os benefícios de políticas de operação ótima obtidas por programação dinâmica estocástica com reduções significativas no esforço computacional. Usando a terminologia usual para expressar as dificuldades de complexidade computacional dos algoritmos de programação dinâmica, as ideias desenvolvidas permitiram contornar a “maldição da dimensionalidade” sem comprometer a qualidade das soluções encontradas.

O primeiro capítulo fez a caracterização do problema de planejamento da geração de energia elétrica abordado no trabalho. Apresentou os principais componentes dos sistemas de energia elétrica e as principais etapas do planejamento da operação de sistemas hidrotérmicos. Em seguida, para contextualizar as contribuições da dissertação, fez uma análise das contribuições ao longo das últimas décadas na área de planejamento da operação de sistemas hidrotérmicos, com ênfase nas abordagens por programação dinâmica e propostas de aproximações para evitar as dificuldades de complexidade computacional; essas propostas moldaram o conjunto de conhecimentos hoje denominados como “programação dinâmica aproximada”.

O problema de planejamento da operação de sistemas hidrotérmicos foi discutido no segundo capítulo. Após a discussão dos aspectos principais do problema e apresentação do modelo matemático para sistemas hidroelétricos com um único reservatório, a análise foi ampliada para sistemas hidroelétricos com múltiplos reservatórios. Em seguida, foi realizado um estudo sobre as características aleatórias das vazões fluviais e detalhadas as propriedades exploradas nos capítulos seguintes.

Os fundamentos da programação dinâmica estocástica e aproximada foram resumidos no terceiro capítulo, incluindo a discussão dos aspectos de complexidade computacional que levaram às principais contribuições da dissertação, discutidas no quarto capítulo. Resumidamente, essas contribuições usam programação dinâmica aproximada com triangulações de Delaunay e redes neurais artificiais (RNA) para expandir as informações obtidas por estados visitados em algoritmos de programação dinâmica aproximada.

As ideias desenvolvidas foram avaliadas nos estudos de casos desenvolvidos no quinto capítulo, com foco na operação da usina de Ilha Solteira para um horizonte de um

ano, com vazões registradas pela Agência Nacional de Águas (ANA). Os experimentos foram divididos em dois grupos, de acordo com os modelos adotados para as vazões do reservatório: foram estudados cenários com afluências aleatórias independentes e cenários com vazões correlacionadas no tempo através de cadeias de Markov de primeira ordem — a interdependência das vazões através de cadeia de Markov aumenta a dimensão do estado e, conseqüentemente, a complexidade computacional do processo de solução através de algoritmos de programação dinâmica.

O primeiro conjunto de experimentos com vazões aleatórias usou o método clássico de programação dinâmica estocástica para encontrar a solução ótima global do problema, usada como métrica para avaliação dos benefícios de alternativas de solução por programação dinâmica aproximada. Em seguida, experimentos com programação dinâmica aproximada exploraram os conceitos de frequência de visitação, evidenciando a importância de evitar a avaliação da equação recursiva de otimalidade em estados improváveis. Esses experimentos mostraram também as vantagens de usar interpolações para ampliar os benefícios das técnicas de programação dinâmica aproximada, em relação à abordagem clássica de programação dinâmica.

O segundo conjunto de experimentos, com vazões aleatórias markovianas, ilustrou o crescimento da complexidade computacional com o método de programação dinâmica clássica e avaliou os benefícios do uso de programação dinâmica aproximada com triangulação de Delaunay e generalizações por redes neurais, para reduzir o esforço computacional de solução do problema. As melhores reduções de esforço computacional foram obtidas através de redes neurais; foram obtidas reduções de 90% no tempo computacional com apenas 3% de gap em relação ao método tradicional de programação dinâmica. Deve-se também observar que as generalizações das informações obtidas no melhor resultado nos estados visitados têm maior escalabilidade para aplicações em sistemas de dimensões maiores.

O próximo passo da pesquisa será a avaliação das ideias propostas na dissertação para sistemas com maior número de reservatórios e maior correlação temporal na aleatoriedade das vazões. Um aspecto importante para a investigação desses cenários é que as ideias de expansão das informações dos estados visitados propostas na dissertação podem ser usadas com diversas abordagens de programação dinâmica aproximada; por exemplo, podem ser incorporadas a algoritmos de programação dinâmica dual.

## Referências

- ANA. *Sistema de acompanhamento de reservatórios (SAR)*. 2018. Agência Nacional de Águas. Disponível em: <[sar.ana.gov.br](http://sar.ana.gov.br)>. Acesso em: 04/2018. Citado 2 vezes nas páginas 37 e 78.
- ARVANITIDIS, N. V.; ROSING, J. Composite representation of a multireservoir hydroelectric power system. *IEEE Transactions on Power Apparatus and Systems*, PAS-89, p. 319–327, 1970. Citado na página 27.
- ARVANITIDIS, N. V.; ROSING, J. Optimal operation of multireservoir systems using a composite representation. *IEEE Transactions on Power Apparatus and Systems*, PAS-89, p. 327–335, 1970. Citado na página 27.
- BARBER, C.; DOBKIN, D.; HUHDANPAA, H. *The Quickhull algorithm for convex hulls*. 22. ed. <http://www.qhull.org>: ACM Trans. on Mathematical Software, 1996. 469-483 p. Citado na página 70.
- BELLMAN, R. *Dynamic Programming*. 1. ed. Princeton, NJ, USA: Princeton University Press, 1957. Citado 7 vezes nas páginas 26, 45, 46, 55, 57, 58 e 90.
- BERG, M. d. et al. *Computational Geometry: Algorithms and Applications*. 3. ed. Santa Clara, CA, USA: Springer-Verlag TELOS, 2008. Citado 2 vezes nas páginas 67 e 69.
- BERNHOLTZ, B.; GRAHAM, L. J. Hydrothermal economic scheduling part 1. solution by incremental dynamic programing. *Transactions of the American institute of electrical engineers.*, v. 79, p. 921–929, 12 1960. Citado na página 27.
- BERTSEKAS, D. P. *Dynamic Programming and Stochastic Control*. Orlando, FL, USA: Academic Press, Inc., 1976. Citado 2 vezes nas páginas 46 e 58.
- BERTSEKAS, D. P.; TSITSIKLIS, J. N. *Neuro-Dynamic Programming*. 1. ed. [S.l.]: Athena Scientific, 1996. Citado 2 vezes nas páginas 29 e 30.
- BRANDÃO, J. L. B. Performance of the equivalent reservoir modelling technique for multi-reservoir hydropower systems. *Water Resources Management*, v. 24, p. 3101–3114, 09 2010. Citado na página 30.
- CAI, X.; MCKINNEY, D. C.; LASDON, L. S. Solving nonlinear water management models using a combined genetic algorithm and linear programming approach. *Advances in water resources*, v. 4, p. 667–676, 06 2001. Citado na página 29.
- CASTELLETTI, A. et al. Neuro-dynamic programming for designing water reservoir network management policies. *Control Engineering Practice*, v. 15, p. 1031–1038, 02 2006. Citado na página 29.
- CEPEL. *NEWAVE - Modelo de Planejamento da Operação de Sistemas Hidrotérmicos Interligados de Longo e Médio Prazo*. 2019. Centro de Pesquisas de Energia Elétrica. Disponível em: <<http://www.cepel.br/produtos/newave-modelo-de-planejamento-da-operacao-de-sistemas-hidrotermicos-interligados-de-longo-e-medio-prazo.htm>>. Acesso em: 01/2019. Citado na página 25.

- CHANDRAMOULI, V.; RAMAM, H. Multireservoir modeling with dynamic programming and neural networks. *Journal of water resources planning and management*, p. 89–98, 03 2001. Citado na página 29.
- CONEJO, A. J.; BARINGO, L. *Power System Operations*. 1. ed. Springer: Princeton University Press, 2018. Citado 3 vezes nas páginas 18, 19 e 20.
- CROW, E. L.; SHIMIZU, K. *Lognormal distributions: theory and applications*. 1. ed. [S.l.]: Marcel Dekker, 1988. Citado 2 vezes nas páginas 38 e 39.
- DARIANE, A. B.; MORADI, A. M. Comparative analysis of evolving artificial neural network and reinforcement learning in stochastic optimization of multireservoir systems. *Hydrological sciences journal*, v. 61, p. 1141–1156, 03 2016. Citado na página 30.
- DURAN, H. et al. Optimal operation of multireservoir systems using an aggregation-decomposition approach. *IEEE Transactions on power apparatus and systems*, PAS-104, p. 2086–2093, 08 1985. Citado na página 28.
- ELETROBRAS. *Sistema de Informações do Potencial Hidrelétrico Brasileiro (SIPOT) - Potencial Hidrelétrico Brasileiro por Bacia*. 2017. Disponível em: <<http://eletrobras.com/pt/Paginas/Potencial-Hidreletrico-Brasileiro.aspx>>. Acesso em: 01/2019. Citado na página 24.
- FORTUNATO, L. A. et al. *Introdução ao planejamento da expansão e operação de sistemas de produção de energia elétrica*. 1. ed. Niteroi, RJ, Brasil: EDUFF - Editora universitária, 1990. Citado 5 vezes nas páginas 18, 19, 20, 21 e 37.
- GOOR, Q.; KELMAN, R.; TILMANT, A. Optimal multipurpose-multireservoir operation model with variable productivity of hydropower plants. *Journal of water resources planning and management*, p. 258–267, 05 2011. Citado na página 29.
- HALL, W. et al. *Optimum firm power output from a two reservoir system by incremental dynamic programming*. 1. ed. University of California, Los Angeles, USA: Water Reserch Center, 1969. Citado na página 27.
- HALL, W. A.; BURAS, N. The dynamic programming approach to water-resources development. *Journal of geophysical research*, v. 66, p. 517–520, 01 1961. Citado na página 26.
- HALL, W. A.; BUTCHER, W. S.; ESOGBUE, A. Optimization of the operation of a multiple-purpose reservoir by dynamic programming. *Water resources research*, v. 4, p. 471–477, 06 1968. Citado na página 26.
- HALL, W. A.; HOWELL, D. T. The optimization of single-purpose reservoir design with the application of dynamic programming to synthetic hydrology samples. *Journal of hydrology*, v. 1, p. 355–363, 12 1963. Citado na página 26.
- HAYKIN, S. S. *Neural networks and learning machines*. Third. Upper Saddle River, NJ: Pearson Education, 2009. Citado 2 vezes nas páginas 72 e 73.
- HEIDARI, M. et al. Discrete differential dynamic programming approach to water resources systems optimization. *Water resources research*, v. 7, p. 273–283, 04 1971. Citado na página 27.

- JACOBSON, D. H.; MAYNE, D. Q. *Differential Dynamic Programming*. 1. ed. [S.l.]: American Elsevier Publishing Company, 1970. Citado na página 27.
- LABADIE, J. W. Optimal operation of multireservoir systems: state-of-the art review. *Journal of water resources planning and management*, 04 2004. Citado na página 25.
- LARSON, R. E. *State increment dynamic programming*. 1. ed. New York, USA: American Elsevier Pub., 1968. Citado 3 vezes nas páginas 27, 46 e 56.
- LEE, J.-H.; LABADIE, J. W. Stochastic optimization of multireservoir systems via reinforcement learning. *Water resources research*, 05 2007. Citado na página 29.
- LITTLE, J. D. C. The use of storage water in a hydroelectric system. *Journal of the operations research society of America*, v. 3, p. 187–197, 03 1955. Citado na página 26.
- LYRA FILHO, C. *Contribuição ao planejamento da produção de energia em sistemas de potencia*. Tese (Doutorado) — UNICAMP, 1984. Citado 6 vezes nas páginas 21, 23, 26, 32, 34 e 78.
- MAASS, A. et al. *Design of Water-Resource Systems*. 1. ed. [S.l.]: Harvard University Press, 1962. Citado na página 26.
- MARTIN, B.; BRENNER, D. *Theory of Multivariate Statistics*. 1. ed. [S.l.]: Springer, 1999. Citado 2 vezes nas páginas 41 e 43.
- MARTINEZ, L.; SOARES, S. Comparison between closed-loop and partial open-loop feedback control policies in long term hydrothermal scheduling. *IEEE Transactions on Power Systems*, v. 17, n. 2, p. 330–336, 05 2002. Citado na página 30.
- MASSE, P. *Les réserves et la régulation de l'avenir dans la vie économique*. 1. ed. [S.l.]: Paris: Hermann et Cie, 1946. Citado na página 26.
- MCKERCHAR, A. Optimal monthly operation of interconnected hydro-electric power storages. *Journal of hydrology*, v. 25, p. 137–158, 08 1975. Citado na página 27.
- MURRAY, D. M.; YAKOWITZ, S. J. Constrained differential dynamic programming and its application to multireservoir control. *Water resources research*, v. 15, p. 1017–1028, 10 1979. Citado na página 28.
- NETO, T. A. A. et al. Comparison of stochastic and deterministic approaches in hydrothermal generation scheduling. *Water resources research*, v. 18, p. 175–180, 10 1984. Citado na página 24.
- ONS. *Sobre o SIN. O sistema em números*. 2018. Operador Nacional do Sistema Elétrico. Disponível em: <<http://ons.org.br/paginas/sobre-o-sin/o-sistema-em-numeros>>. Acesso em: 01/2019. Citado 2 vezes nas páginas 20 e 24.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 75.
- PEREIRA, M.; PINTO, L. Stochastic optimization of a multireservoir hydroelectric system: a decomposition approach. *Water resources research*, v. 21, p. 779–792, 06 1985. Citado 2 vezes nas páginas 22 e 28.

- PEREIRA, M.; PINTO, L. Multi-stage stochastic optimization applied to energy planning. v. 52, p. 359–375, 05 1991. Citado na página 29.
- PICCARDI, C.; SONCINI-SESSA, R. Stochastic dynamic programming for reservoir optimal control: dense discretization and inflow correlation assumption made possible by parallel computing. v. 27, p. 729–741, 05 1991. Citado na página 29.
- POWELL, W. B. *Approximate Dynamic Programming: Solving the Curses of Dimensionality (Wiley Series in Probability and Statistics)*. New York, NY, USA: Wiley-Interscience, 2007. Citado 7 vezes nas páginas 30, 52, 59, 61, 62, 65 e 66.
- RAMAN, H.; CHANDRAMOULI, V. Deriving a general operating policy for reservoirs using neural network. *Journal of Water Resources Planning and Management*, v. 122, n. 5, p. 342–347, 1996. Citado na página 29.
- ROEFS, T. G.; BODIN, L. D. Optimal operation of multireservoir systems using a composite representation. *Water resources research*, v. 6, p. 410–421, 04 1970. Citado na página 27.
- ROSENTHAL, R. D. The status of optimization models for the operation of multireservoir systems with stochastic inflows and nonseparable benefits. *Tennessee Water Resources Research Center*, v. 75, p. 763–786, 05 1980. Citado 2 vezes nas páginas 22 e 25.
- ROSS, S. *A First Course in Probability*. 5. ed. [S.l.]: Pearson Prentice Hall, 1998. Citado 3 vezes nas páginas 38, 42 e 51.
- SAAD, M. et al. Fuzzy learning decomposition for the scheduling of hydroelectric power systems. v. 32, p. 179–186, 01 1996. Citado na página 29.
- SAAD, M. et al. Learning disaggregation technique for the operation of long-term hydroelectric power systems. v. 30, p. 3195–3202, 11 1994. Citado na página 29.
- SHAPIRO, A. Analysis of stochastic dual dynamic programming method. *European Journal of Operational Research*, v. 209, p. 63–72, 2011. Citado na página 29.
- SHARIF, M.; WARDLAW, R. Multireservoir systems optimization using genetic algorithms: Case study. *Journal of Computing in Civil Engineering*, v. 14, n. 4, p. 255–263, 2000. Citado na página 29.
- SHERKAT, V. et al. Stochastic long-term hydrothermal optimization for a multireservoir system. *IEEE Transactions on power apparatus and systems*, PAS-104, p. 2040–2051, 08 1985. Citado na página 28.
- SIQUEIRA, T. G. *Comparação entre Diferentes Abordagens de Programação Dinâmica no Planejamento da Operação Energética de Sistemas Hidrotérmicos de Potência*. Tese (Doutorado) — UNICAMP, 2009. Citado 2 vezes nas páginas 22 e 23.
- SOARES, M. P.; STREET, A.; VALLADAO, D. M. On the solution variability reduction of stochastic dual dynamic programming applied to energy planning. *European Journal of Operational Research*, v. 258, n. 2, p. 743–760, 2017. Citado na página 29.
- STROOCK, D. W. *An Introduction to Markov Process*. 2. ed. [S.l.]: Springer, 2000. Citado na página 41.

- TOLMASQUIM, M. T. *Geração de Energia Elétrica no Brasil*. 2. ed. [S.l.]: Editora Interciência, 2005. Citado 2 vezes nas páginas 20 e 24.
- TURGEON, A. Optimal operation of multireservoir power systems with stochastic inflows. *Water resources research*, v. 16, p. 275–283, 04 1980. Citado na página 28.
- TURGEON, A. A decomposition method for the long-term scheduling of reservoirs in series. *Water resources research*, v. 17, p. 1565–1570, 12 1981. Citado na página 28.
- WEC. *Energy Resources: Hydropower*. 2016. World Energy Council. Disponível em: <<https://www.worldenergy.org/data/resources/resource/hydropower/>>. Acesso em: 01/2019. Citado na página 24.
- YAKOWITZ, S. Dynamic programming applications in water resources. *Water resources research*, v. 18, 08 1982. Citado na página 25.
- YEH, W. W. Reservoir management and operations models: a state-of-the-art review. *Water resources research*, v. 21, 12 1985. Citado na página 25.
- ZAMBELLI, M. et al. Deterministic versus stochastic models for long term hydrothermal scheduling. p. 7 pp.–, 06 2006. Citado na página 30.