Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Telemática

Paradigma de Programação Dinâmica Discreta Em Problemas Estocásticos de Investimento e Produção

Autor: **Edilson Fernandes de Arruda**
Orientador: **Prof. Dr. João Bosco Ribeiro do Val**
Co-orientador: **Prof. Dr. Anthony Almudevar**
Dept. of Biostatistics and Computational Biology. University of Rochester

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: **Automação e Controle**.

Banca Examinadora

Prof. Dr. João Bosco Ribeiro do Val (Presidente) DT/FEEC/UNICAMP
Prof. Dr. José Leandro Félix Salles ..... DEE/UFES
Prof. Dr. Marcelo Dutra Fragoso ..... CSR/LNCC
Prof. Dr. Rafael Santos Mendes ..... DCA/FEE/UNICAMP
Prof. Dr. Wagner Caradori do Amaral ..... DCA/FEE/UNICAMP

Campinas SP, 31 de maio de 2006.

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Ar69p

Arruda, Edilson Fernandes de
Paradigma de programação dinâmica discreta em problemas estocásticos de investimento e produção / Edilson Fernandes de Arruda. – Campinas, SP: [s.n.], 2006.

Orientadores: João Bosco Ribeiro do Val e Anthony Almudevar.
Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Markov, Processos de. 2. Programação estocástica. 3. Progamação dinâmica. I. Val, João Bosco Ribeiro do. II. Almudevar, Anthony. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Titulo em Inglês: The paradigm of discrete dynamic programming in stochastic investment and production problems.
Palavras-chave em Inglês: Markov processes, Stochastic programming, Dynamic programming.
Área de concentração: Automação e Controle.
Titulação: Doutor em Engenharia Elétrica
Banca examinadora: José Leandro Felix Salles, Marcelo Dutra Fragoso, Rafael Santos Mendes e Wagner Caradori do Amaral
Data da defesa: 31/05/2006

Resumo

Apresenta-se um modelo de controle por intervenções para o problema de produção e estoque de vários itens, com diversos estágios de produção. Este problema pode ser solucionado via programação dinâmica discreta (PD) por um operador de custo descontado. Para contornar a dificuldade de obtenção da solução ótima via PD ao se considerar um número razoável de classes de itens e suas etapas de produção, esta tese desenvolve-se em duas linhas. A primeira delas consiste em tomar uma noção de estabilidade estocástica no sentido Foster-Lyapunov para caracterizar a família de soluções candidatas a ótima, originando uma classe de políticas que geram um subconjunto de estados que são recorrentes positivos. Dessa forma, é possível propor políticas sub-ótimas que sejam estáveis, e cuja consideração de otimalidade possa ser desenvolvida apenas no subconjunto de estados recorrentes, simplificando a tarefa da PD e focando nos estados mais freqüentados no longo prazo. A segunda linha de abordagem consiste em desenvolver técnicas de PD aproximada para o problema, através de uma arquitetura de aproximação fixa aplicada a um subconjunto amostra do espaço de estados. Um avanço analítico é alcançado por observar como uma arquitetura de aproximação pode capturar adequadamente a função valor do problema, vista como uma projeção da função valor na arquitetura. Condições para que um algoritmo de PD aproximada convirja para essa projeção são obtidas. Essas condições são independentes da arquitetura utilizada. Um algoritmo derivado dessa análise é proposto, a partir do monitoramento da variação de passos sucessivos.

Palavras-chave: Processos de Decisão Markovianos, Controle por Intervenções, Estabilidade Estocástica, Programação Dinâmica Aproximada, Programação Neuro-Dinâmica.

Abstract

We propose an intervention control model for a multi-product, multi-stage, single machine production and storage problem. The optimal policy is obtained by means of discrete dynamic programming (DP), through a discounted cost contraction mapping. In order to overcome the difficulty of obtaining the optimal solution for problems with a reasonable number of products and production stages, we take two different approaches. The first one consists in using a notion of stochastic stability in the Foster-Lyapunov sense to characterize the candidate policies, thus originating a class of policies that induce a subset of positive recurrent states. Therefore, one can propose suboptimal policies that are stable and seek optimality only in the subset of recurrent states, in such a way that simplifies the DP task and focuses on the states which are visited more frequently in the long run. The second approach consists in developing approximate dynamic programming techniques for the problem, by means of a fixed approximation architecture applied to a sample subset of the state space. A novel result is obtained by observing how an approximation architecture can adequately capture the value function of the problem, which is viewed as a projection of the value function into the architecture. We obtain conditions for an approximate DP algorithm to converge to this projection. These conditions are architecture independent. An algorithm derived from this analysis is proposed that monitors the variation between successive iterates.

Keywords: Markov Decision Problems, Control by Interventions, Stocahstic Stability, Approximate Dynamic Programming, Neuro-Dynamic Programming

Em memória de meu tio Pedro

Agradecimentos

Aos professores João Bosco e Anthony Almudevar, pela amizade e pela orientação.

A meus pais e minha esposa, pelo carinho e pela paciência.

Ao meu filho Pedro, por me mostrar que existem coisas muito mais difíceis que desenvolver uma tese de doutorado.

Aos colegas do DT, pela amizade, pela convivência e por proporcionar um excelente ambiente de trabalho.

À CAPES, pelo apoio financeiro.

Sumário

Lista de Figuras

Lista de Tabelas

Trabalhos Publicados Pelo Autor

1. Arruda, EF, do Val, JBR & Almudevar A (2005). Function approximation for a production and storage problem under undertainty, *Proceedings of the IEEE International Conference on Mechatronics & Automation*, Niagara Falls, Canada, 665-670

2. Arruda, EF, do Val, JBR & Almudevar A (2004). Stability and optimality of a discrete production and storage model with uncertain demand, *Proceedings of the 43rd IEEE Conference on Decision and Control*, Nassau, 3354–3360

3. Arruda, EF & do Val, JBR (2003). A discrete dynamic programming approach to the production and storage problem, *Annals of the XXXV Brazilian Symposium on Operations Research*, Natal-RN, Brazil, pp. 2267 - 2276 doc pdf

4. Arruda, EF & do Val, JBR (2003). On stochastic production and storage problems with interventions at the end of production stages, *In: Mathematical Programing in Rio: A Conference in Honour of Nelson Maculan*, Búzios-RJ, Brazil, pp. 29-34

5. Arruda, EF & do Val, JBR (2002). Problema de controle impulsional para vários itens em produção, *Anais do XIV Congresso Brasileiro de Automática*, Natal-RN, Brazil, pp. 1385-1390

Glossário

CMH - Cadeia de Markov Homogênea

EC - Expansão de Capacidade

P&E - Produção e Estoque

PD - Programação Dinâmica

PDA - Programação Dinâmica Aproximada

PDM - Processo de Decisão Markoviano

PDRA - Programação Dinâmica Com Representação Aproximada

PMDP - Processo Markoviano Determinístico por Partes

Capítulo 1

Introdução

Em sistemas de investimento e produção é comum a presença de incertezas na demanda e nos tempos de realização de etapas de produção. Essas características estão presentes em problemas de produção e estoque, objeto de estudo deste trabalho.

O problema de produção e estoque (P&E) consiste numa variação do problema clássico de controle de estoque sujeito a demanda aleatória por itens. Ao contrário do modelo clássico, no qual se analisa quando e quanto repor ao estoque, num mecanismo que sugere que os itens são adquiridos de terceiros, nos problemas de produção e estoque introduz-se um detalhamento do mecanismo de produção. A idéia é acoplar a decisão sobre os instantes mais propícios a paralisações e retomadas de produção ao problema de atender a demanda com itens em estoque, controlando-se dinamicamente a taxa de processamento de itens frente à demanda observada.

Decisões de planejamento devem determinar quando iniciar e qual a taxa de produção adequada, face às incertezas nas previsões de demanda e nos custos de paralisação e retomada de produção. O processo decisório também deve levar em conta incertezas adicionais tais como possíveis devoluções de mercadoria, atrasos na produção devido à quebra de máquinas, atrasos na estocagem dos itens acabados, atrasos na entrega dos pedidos dos clientes, desistência de pedidos, etc.

O gerenciamento adequado do problema de produção e estoque deve perseguir o equilíbrio entre a oferta e a demanda durante toda a operação (da empresa), de modo a minimizar os custos esperados de estocagem (superprodução), falta de itens disponíveis (sub-produção) e períodos de inatividade, tais como: gastos com paralisação ou reinício da produção, custos de reajustes (setup) de máquinas para produzir tipos diferentes de itens, etc.

Quando modelados por meio de processos markovianos determinísticos por partes (PMDP), os problemas P&E são normalmente solucionados por meio de soluções viscosas de equações quasi-variacionais, vide (Salles e do Val 2001), (Gatarek 1992), (Moresino, Pourtallier e Tidball 1999). Em (Monticino e Weisinger 1995) e (Arruda e do Val 2002) utilizou-se programação dinâmica discreta para encontrar políticas ótimas de limiar, supondo tempo de conclusão exponencialmente distribuído. A utilização de programação dinâmica discreta associada a eventos foi sugerida em (Almudevar 2001) para o controle de PMDP. Nesse trabalho, o valor esperado do custo em cada estado é calculado com base na existência ou não de chegada de demanda até o início do próximo período de decisão. Essa formulação não contempla, contudo,

custos de paralisação e custos de ajuste e reinício da produção (custos de setup).

Um modelo discreto é bastante natural para problemas P&E, visto que os instantes de decisões estão normalmente associados ao término de etapas de produção. Mesmo que a duração destas seja de natureza aleatória, pode-se utilizar um modelo discreto com instantes de decisão aleatórios. Naturalmente, esse modelo não poderia deixar de levar em conta os custos de setup associados às ações de controle aplicadas ao sistema. Além de adequado ao problema, espera-se que o modelo discreto apresente uma solução mais tratável se comparado ao modelo a tempo contínuo (Salles e do Val 2001).

Este trabalho propõe um modelo P&E de vários produtos baseado em discretização por eventos. Entre eventos, considera-se que os níveis de estoque permanecem constantes. Assim, o modelo proposto incorpora as características gerais de PMDP, a saber, evolução determinística intercalada por saltos aleatórios. Além disso, o modelo proposto admite custos de setup. Trata-se de um modelo mais geral que os encontrados na literatura, por não estar restrito a problemas que apresentem processos de chegada de demanda poissonianos. É proposto, para esse modelo, um método de solução baseado em programação dinâmica (PD). Por se tratar de um modelo discreto, sua solução pode ser expressa em termos de programação dinâmica discreta, tal qual em (Monticino e Weisinger 1995) e (Arruda e do Val 2002).

O controle do processo de produção é efetuado por meio de intervenções de que podem ser aplicadas no início de cada período discreto na evolução do sistema. Uma política de controle estacionária define os estados de intervenção e não intervenção do sistema. Os estados de intervenção são visitados apenas instantaneamente e podem, portanto, ser excluídos do espaço de estados do problema. Dessa forma, uma política de controle por intervenções (ou controle impulsional) dá origem à cadeia de Markov na qual o sistema opera (em horizonte infinito). Nesse sentido, o controle por intervenções proposto pode ser visto como uma generalização da classe de políticas de controle de processos de decisão markovianos (PDM) padrão, que define a cadeia de Markov correspondente, além das probabilidades de transição entre os estados pertencentes a essa cadeia.

A política de controle ótima para o modelo P&E proposto pode ser obtida por meio da iteração de um operador de programação dinâmica em tempo discreto. A cada estado do sistema, associa-se um operador de intervenção e um operador de não intervenção. O operador de PD é definido como o mínimo entre esses dois operadores auxiliares e deve ser aplicado em todo o espaço de estados do sistema. Trata-se de um operador contrativo cujo ponto fixo é a função valor do problema, que define a política de controle ótima. Conhecendo-se a política de controle ótima, pode-se determinar os estados de não intervenção, pertencentes à cadeia de Markov em que o sistema opera, e os estados de intervenção que, por não serem freqüentados pelo sistema, podem ser eliminados da cadeia. Mais detalhes sobre esse assunto podem ser encontrados na Seção 2.6.2. No Capítulo 2 introduz-se o modelo proposto, a proposta de controle por intervenções e o operador de programação dinâmica destinado a obter a política de controle ótima. Além disso, apresenta-se uma discussão comparativa entre PDM's padrão e o problema de controle impulsional no modelo P&E proposto.

A solução via PD é elegante e concisa. Contudo, ao se considerar um número razoável de classes de produtos e estágios de produção, verifica-se que a obtenção da solução ótima via PD se torna inviável devido ao grande número de estados. Para contornar esse problema, o trabalho desenvolve-se em duas linhas distintas. A primeira delas consiste em caracterizar a

estabilidade estocástica das políticas de controle candidatas a solução ótima por meio do critério de recorrência positiva, através de funções de Foster-Lyapunov. Nessa abordagem, a classe de políticas de controle factíveis admite apenas soluções candidatas estocasticamente estáveis. Estabelece-se condições necessárias e suficientes para que uma política de controle induza um subconjunto finito de estados recorrentes positivos. Dessa forma, é possível obter políticas de controle sub-ótimas que estabeleçam uma região de estabilidade à qual o sistema atinge em tempo finito (em valor esperado) a partir de qualquer estado inicial. Essa é a região que o sistema tende a freqüentar, isto é, a região mais significativa do ponto de vista de ocupação relativa. Isso a torna mais determinante no custo associado à política de controle empregada.

De posse dessas informações, um procedimento destinado a obter políticas de controle sub-ótimas pelo processo reverso é proposto, a partir de uma região de estabilidade arbitrária. A partir de uma região de estabilidade arbitrada, define-se no conjunto complementar a essa região uma política arbitrária e adequada de controle a ser aplicada. O papel dessa política é estabelecer uma função de Foster-Lyapunov que defina o subconjunto de estados recorrentes positivos desejado (isto é, um *conjunto atrator*, que é a própria região de estabilidade do problema). O segundo passo do procedimento é a determinação das ações de controle na região de estabilidade. Estas ações são obtidas por meio do operador de PD, iterado unicamente nos estados pertencentes a essa região, utilizando-se uma condição de contorno arbitrária. Na prática, esse procedimento reduz significativamente o número de estados no domínio do operador de PD e possibilita um enfoque maior na região de estabilidade que, por ser mais freqüentada em horizonte infinito, contribui mais significativamente no custo final do problema. No Capítulo 3 introduz-se a noção de estabilidade empregada nesse trabalho e apresenta-se condições necessárias e suficientes para a estabilidade estocástica de políticas de controle por intervenção. O procedimento de obtenção de soluções sub-ótimas descrito acima é detalhado no Capítulo 5.

A segunda linha adotada para tratar problemas de grande porte é a utilização de programação dinâmica aproximada (PDA). Em problemas de grande porte apenas métodos de PDA que utilizam representação aproximada são viáveis. Trata-se de métodos com aproximação paramétrica da função valor, sendo o número de parâmetros muito menor que a cardinalidade do espaço de estados. Métodos de representação exata, da mesma forma que a PD, tornam-se inviáveis devido ao excesso de estados. Este trabalho apresenta resultados analíticos obtidos para esses métodos que garantem convergência para a resposta ótima, vista como uma projeção de um ponto fixo do operador de PD na arquitetura de aproximação utilizada. Esse resultado apresenta uma caracterização da resposta obtida pelo algoritmo aproximado. Outros métodos de convergência garantida existentes na literatura não apresentam essa caracterização do ponto de acumulação e nada se pode inferir sobre a natureza deste. Além disso, os resultados obtidos definem condições de convergência independentemente da arquitetura de aproximação utilizada. Resultados existentes na literatura apenas garantem convergência para classes particulares de arquiteturas de aproximação. Apresenta-se um algoritmo de convergência garantida que reproduz as condições analíticas de convergência para a resposta ótima através da monitoração de iterações sucessivas. Trata-se de um procedimento bastante simples de controle de passo aplicado a cada iteração de modo a garantir conformidade entre as iterações do algoritmo aproximado e as condições analíticas para a convergência para a resposta ótima. Os resultados analíticos referentes à convergência de algoritmos PDA com representação aproximada, bem como o algoritmo proposto encontram-se no Capítulo 4.

No intuito de aplicar PDA com aproximação paramétrica no problema P&E estudado, introduz-se no Capítulo 6 uma arquitetura de aproximação para problemas P&E. Trata-se de uma arquitetura de aproximação polinomial, na qual a função valor é aproximada por um polinômio em função dos níveis de estoque/déficit. Esse esquema de aproximação pode ser visto como uma aproximação linear que tem como parâmetros os coeficientes do polinômio. Visando melhorar a qualidade da solução aproximada assim obtida, utiliza-se uma representação exata em um conjunto finito de estados vizinhos do estado de estoque nulo. Essa é uma maneira de trazer mais confiabilidade às ações de controle nas imediações do estado de estoque nulo. A região de representação exata é estabelecida arbitrariamente, de maneira análoga à região de estabilidade no algoritmo sub-ótimo baseado nas condições de estabilidade estocástica. No entanto, no algoritmo de PDA, essa região perde a caracterização de região de estabilidade. São discutidas questões relativas à parametrização e convergência do algoritmo de PDA proposto, com base em exemplos numéricos ilustrativos. Comparações entre as soluções ótimas e aproximadas são apresentadas de forma a ilustrar o reflexo dos erros de aproximação no custo associado à política de controle obtida pelo algoritmo aproximado. Finalmente, o Capítulo 7 conclui este trabalho.

Capítulo 2

O Modelo e o Controle

2.1 Preliminares

Apresenta-se, nesse capítulo, o problema de produção e estoque (P&E) de múltiplos produtos (itens). Introduz-se um modelo estocástico discreto e utiliza-se o paradigma de controle por intervenções a fim de se modular a evolução do sistema.

Após uma descrição do sistema e algumas possíveis opções de modelagem, formula-se um problema de controle por intervenções e define-se um operador de programação dinâmica discreta destinado a obter a solução do problema, isto é, a política de controle ótima.

Este capítulo contém, adicionalmente, uma discussão comparativa entre o modelo de controle por intervenções apresentado e problemas de decisão markovianos (PDM) padrão. Embora exista hoje uma relação estreita entre os modelos, existem algumas diferenças que serão exploradas.

2.2 O Problema de Produção e Estoque de Múltiplos Itens

Problemas de produção e estoque (P&E) têm como objetivo modelar as decisões de controle em um sistema de produção. Tipicamente, necessita-se tomar decisões sobre quando e como produzir de modo a minimizar o custo de operação do sistema em um dado horizonte de tempo.

Apresenta-se aqui um problema P&E destinado a modelar gargalos de produção em sistemas de manufatura multi-produto. Assume-se que uma única unidade de produção é responsável pelo beneficiamento de vários produtos e que a unidade fabril não pode produzir mais de um produto a cada instante de tempo. O processo de manufatura de um lote de unidades de cada classe de produtos passa por estágios de produção seqüenciais bem definidos. Assume-se, para todas as classes de produtos, que um dado estágio de produção, uma vez iniciado, não pode ser interrompido.

O problema P & E é análogo ao problema de expansão de capacidade (EC), (Luss 1982) e ambos podem ser tratados por meio de processos markovianos determinísticos por partes (PMDP), veja (Davis, Dempster, Sethi e Vermes 1987), (Mancinelli e Gonzalez 1997), (Jean-Marie e Tidball 1997), (do Val e Salles 1999), (Salles e do Val 2001), (Arruda e do Val 2002),

(Salles e do Val 2002), dentre outros. Os PMDP, introduzidos por (Davis 1984), são adequados para a modelagem de problemas P&E por capturarem suas características fundamentais, a saber: trajetória determinística intercalada por saltos aleatórios.

Assume-se aqui que a demanda por cada produto é aleatória, o que é natural em sistemas de manufatura. Considerando a natureza discreta do processo de manufatura dividido em estágios de produção, o sistema é modelado como um *problema de controle por intervenções*, que pode, por sua vez, ser visto como uma generalização de um *processo de decisão markoviano* (PDM) em tempo discreto, em que o espaço de estados é modificado pela política de controle. A forma de decisão por intervenções difere da forma padrão de um PDM, vide por exemplo (Puterman 1994). As decisões de controle são tomadas em instantes de inspeção (ou de decisão) previamente definidos, que podem estar relacionados à finalização de etapas de produção. Quando produzindo, o sistema é inspecionado ao fim de cada estágio de produção; quando paralisado, o sistema é inspecionado depois de um período de espera pré-especificado. Os intervalos de tempo compreendendo qualquer estágio de produção ou período de espera podem ser determinísticos ou aleatórios. Define-se um período da evolução do sistema como o tempo entre duas inspeções sucessivas; dessa forma, o tempo de duração de cada período pode ser determinístico ou aleatório. A cada inspeção deve-se decidir por manter o sistema no mesmo *modo de produção* até a próxima inspeção ou por aplicar uma intervenção, pagando-se, nesse caso, um custo de intervenção. Cada intervenção aplicada pode mudar o artigo em produção, paralisar a produção ou reiniciar a produção de uma dada classe de produtos, caso o sistema esteja paralisado.

Em problemas P&E, o controlador do sistema deve definir uma seqüência de ações de controle tal que o controle aplicado a cada inspeção implique na minimização do valor esperado da soma (ou acumulado) dos custos futuros.

2.3 Notas Bibliográficas

Há na literatura um grande número de referências que tratam de problemas P&E nas áreas de teoria de sistemas e pesquisa operacional. Em uma linha de abordagem, problemas P&E foram modelados como problemas de controle de fluxo em manufatura. Tais problemas possuem natureza contínua, sendo mais adequados a processos de produção em grande escala. Nesse tipo de modelo, o nível de estoque é tratado como uma variável real. Um modelo clássico de um único produto e uma única máquina sujeita a falhas com taxa de demanda constante foi apresentado em (Akella e Kumar 1986). Algumas variações desse problema foram propostas e abordadas em (Perkins e Srikant 1999) e (Salama 2000). Problemas com uma única máquina e vários estágios de produção foram estudados em (Perkins e Srikant 1997) e (Perkins e Srikant 1998), enquanto um problema de um único produto, com várias máquinas e vários estágios de produção foi tratado por (Song e Sun 1998). Esse modelo não incorpora custos de intervenção e considera um funcional de custo linear. A possibilidade de ocorrência de acréscimo temporários na capacidade de produção foi considerada no modelo em (Huang, Hu e Vakili 1998). Para uma revisão de problemas de fluxo em manufaturas recomenda-se (Sethi, Yan, Zhang e Zhang 2002).

Em outra linha de abordagem, alguns modelos exploraram a similaridade entre problemas

P&E e problemas de filas. Dentre estes, destacamos (Federgruen e Zipkin 1986), (Goyal e Giri 2003), (Gravish e Graves 1981) e (Gullu 1998). Gravish e Graves (1981) demonstraram a otimalidade de uma política caracterizada por dois níveis de estoque críticos para um problema com custo de paralisação/reinício de produção (setup). Seguindo a mesma linha, Federgruen e Zipkin (1986) demonstraram a otimalidade de uma política com um único nível de estoque crítico para um problema sem custo de setup. Um problema com níveis de capacidade aleatórios foi abordado em (Gullu 1998), enquanto um modelo considerando depreciação foi estudado em (Goyal e Giri 2003).

Introduzidos por (Davis 1993), processos markovianos determinísticos por partes (PMDP) são bastante utilizados na modelagem de problemas P&E. Sistemas de manufatura sujeitos a falhas foram estudados em (Boukas e Haurie 1990), (Boukas, Zhu e Zhang 1994), (Presman, Sethi e Zhang 1995) e (Yan e Zhang 1997). Problemas de controle contínuo foram estudados em (Boukas e Haurie 1990), (Boukas et al. 1994) e (Presman et al. 1995), ao passo que uma combinação de controle contínuo e por intervenções foi utilizada em (Yan e Zhang 1997). Outros modelos P&E que utilizaram ferramentas de PMDP foram apresentados em (Arruda e do Val 2002), (do Val e Salles 1999), (Jean-Marie e Tidball 1997), (Mancinelli e Gonzalez 1997) e (Salles e do Val 2001). Problemas de múltiplos produtos produzidos por uma única máquina foram estudados em (Arruda e do Val 2002), (Arruda 2002), (Jean-Marie e Tidball 1997) e (Mancinelli e Gonzalez 1997). Já do Val e Salles (1999) e Salles e do Val (2001) abordaram um problema de um único produto e única máquina com intervenções em tempo contínuo utilizando técnicas de PMDP.

Neste trabalho, introduz-se um modelo discreto para um problema P&E de vários produtos e vários estágios, onde uma única unidade fabril é responsável pela produção de vários produtos. O modelo estudado pode também ser visto como um processo por eventos discretos inserido em um processo markoviano determinístico por partes (PMDP) que é inspecionado em instantes de interesse à medida que o sistema evolui. Esses instantes de interesse coincidem com o fim de um estágio de produção, um instante de chegada de demanda ou o fim de um tempo de espera pré-definido. Uma vantagem do método proposto sobre modelos em tempo contínuo é a possibilidade de se representar políticas de produção por meio de cadeias de Markov em tempo discreto, e assim solucionar o problema de controle através de programação dinâmica discreta, veja seção 2.5.1. O fato de se considerar decisões de controle em determinados instantes, ao invés de em tempo contínuo, constitui uma resposta natural a problemas P&E, dado que muitas tarefas em processos industriais não podem ser interrompidas uma vez iniciadas. Além disso, a solução de modelos contínuos baseados em PMDP's é normalmente obtida por procedimentos bem mais complexos que a programação dinâmica discreta, destinados a resolver as desigualdades variacionais características desse tipo de formulação, veja (Gatarek 1992). Contrastando com modelos de máquinas sujeitas a falhas, que concentram a atenção na possibilidade de quebra das máquinas e muitas vezes modelam a taxa de chegada de demanda como constante, veja por exemplo (Boukas et al. 1994), adotamos a hipótese mais realista de que a demanda em cada período é aleatória. Quebras de máquinas podem ser levadas em conta na distribuição do tempo de duração dos estágios na evolução do sistema, veja essa extensão na Seção 2.4.1.

2.4 Um Modelo de Produção e Estoque

Considere um sistema de produção e estoque responsável pela produção em lotes de J tipos diferentes de produtos. A cada instante de tempo há no máximo uma classe de produtos em produção. Tipicamente, um único lote de η_j unidades do produto j, $j \in \mathcal{J} := \{1, \ldots, J\}$ é completado em Γ_j unidades de tempo, sendo que Γ_j pode ser uma quantidade aleatória. No intervalo $[0, \Gamma_j]$ distinguem-se sub-intervalos disjuntos Θ_{ji}, $i \in \mathcal{I}_j := \{1, \ldots, I_j\}$, tais que

$$[0, \Gamma_j] = \bigcup_{i=1}^{I_j} \Theta_{ji}.$$

Cada sub-intervalo representa um estágio de produção e indica o tempo gasto em uma tarefa específica no processo de produção do produto j, como montagem, pintura, etc. O tempo necessário para se concluir um estágio de produção $i \in \mathcal{I}_j$ é a duração do intervalo Θ_{ji} e o valor I_j representa o total de estágios necessários para a produção de um novo lote de η_j unidades do produto j. Os tempos de duração dos intervalos Θ_{ji} podem ser determinísticos ou aleatórios. Assim, o tempo total de produção de um lote de itens de um dado produto j, Γ_j, pode ser determinístico ou aleatório.

Seja $\{T_0, \ldots, T_k, \ldots\}$ a seqüência de instantes de decisões de controle. Os intervalos $T_k - T_{k-1}$ podem ser determinísticos ou aleatórios. Eventualmente, $T_k - T_{k-1}$ é igual à duração do intervalo Θ_{ji} para algum $j \in \mathcal{J}$ e $i \in \mathcal{I}_j$.

Pode-se identificar três processos vetoriais que descrevem a evolução do sistema estudado:

- $T_k \to n_{T_k}$, o nível de estoque no princípio de cada período k, um processo J-dimensional. O processo $\{n_{T_k}\}$ assume valores no conjunto $\mathcal{N} := \prod_{j=1}^{J} \mathcal{N}_j$, $\mathcal{N}_j := \{N_j^-, \ldots, N_j^+\}$, sendo que os inteiros N_j^- e N_j^+ representam o mínimo e o máximo nível de estoque admissível para o produto j (eventualmente, $N_j^- = -\infty$ e $N_j^+ = \infty$).

- $T_k \to i_{T_k}$, o estágio de produção no princípio de cada período k; um processo J-dimensional que assume valores no conjunto $\mathcal{I} = \prod_{j=1}^{J} \mathcal{I}_j$.

- $T_k \to j_{T_k}$, o *modo de produção* no princípio de cada período k. O processo j é unidimensional e assume valores no conjunto $\mathcal{J}_0 := \{0\} \cup \mathcal{J}$.

Por simplicidade, denota-se $n_{T_k} = n_k = [n_k(1) \ldots n_k(J)]^T$, $i_{T_k} = i_k = [i_k(1) \ldots i_k(J)]^T$ e $j_{T_k} = j_k$, sendo que $v_k(m)$ indica a m-ésima componente do vetor v_k e o sobrescrito T indica o transposto de um vetor. O processo $\{j_k\}$ mapeia o modo de produção ao longo do tempo; $j_k = j \in \mathcal{J}$ indica que o sistema está produzindo o produto j à máxima taxa possível, ao passo que $j_k = 0$ indica que o sistema está paralisado. Define-se o processo conjunto $T_k \to z_k := (n_k, i_k, j_k)$ para representar a evolução do sistema.

Em cada período k, o sistema está produzindo um novo lote de η_j unidades do produto j à máxima taxa possível ou está paralisado. Deve-se decidir, ao final de cada estágio de produção, se o sistema deve ser mantido no mesmo modo de produção. Nesse caso, faz-se $j_{k+1} = j_k$ e inicia-se a tarefa seguinte no processo de produção do produto j ou, se $j_k = 0$, mantém-se o sistema paralisado por mais um período. Se a decisão tomada é a de intervir no sistema, o novo

modo de produção é indicado por j_{k+1}, e obviamente $j_{k+1} \neq j_k$. Uma penalidade é paga toda vez que se verifica uma mudança no modo de produção do sistema.

Quando a produção de algum produto j está ativa, convenciona-se que o processo $k \to z_k$ evolui no subconjunto $S^j := \mathcal{N} \times \mathcal{I} \times \{j\}$; e sempre que o sistema estiver em produção, diz-se que o sistema evolui no *subconjunto de produção* $S' := \bigcup_{j \in \mathcal{J}} S^j$. Caso contrário, $k \to z_k$ evolui no *subconjunto de paralisação* $S^0 := \mathcal{N} \times \mathcal{I} \times \{0\}$. Assim, o espaço de estados do problema estudado é dado por $S := S' \cup S^0$. Ao decisor cabe determinar as transições entre os subconjuntos $S^j,\ j \in \mathcal{J}_0$, pagando uma taxa de intervenção a cada transição.

A idéia é acompanhar a demanda aleatória por itens, à medida que esta é observada ao longo do tempo, e não acumular itens em estoque ou sob encomenda. Os custos operacionais do sistema são o custo de estoque/deficit, o custo de produção e os custos de intervenção descritos anteriormente. Fazendo uso destes, busca-se minimizar o valor esperado do custo descontado de operação em horizonte infinito. O controle por intervenções aqui descrito está inserido no paradigma de *Problemas de Controle Impulsional*, veja (Salama 2000), (Salles e do Val 2001), (Arruda 2002), (Jean-Marie e Tidball 1997) e (Mancinelli e Gonzalez 1997), para alguns tratamentos de controle impulsional em problemas P&E.

Para completar a formulação, é necessário definir alguns outros elementos. Defina-se por τ_k o tempo de duração do k-ésimo período, a saber, $\tau_k = T_k - T_{k-1}$. Quando o sistema está em produção, isto é, quando $k \to z_k$ está evoluindo em S', o período τ_k corresponde ao tempo de conclusão de um estágio de produção. Dessa forma, $\tau_k =$ duração de Θ_{i_k}. Quando o sistema está paralisado, ($k \to z_k$ está evoluindo em S^0), o período τ_k é igual à duração do intervalo Δ_{i_k}, sendo que $\Delta_i,\ i \in \mathcal{I}$, é um intervalo pré-determinado entre decisões sucessivas em S^0, que pode ser determinístico ou aleatório.

Uma vez que um estágio de produção de algum produto j é iniciado, o esforço de produção continua sem interrupções até que se atinja o próximo estágio. Ao final do último estágio de produção de um dado produto j, um novo lote de η_j unidades desse produto é entregue e a componente $i_k(j)$ do vetor i_k retorna ao estágio 1. Visto que a produção é reiniciada ao final de cada ciclo de produção e considerando que o estágio de produção permanece constante entre dois instantes de decisão, a evolução do processo $k \to i_k$ é dada por

$$i_{k+1} = \begin{cases} i + \left(1 - i(j)\mathbb{1}_{\{i(j)=I_j\}}\right) e_j, & \text{se } z_k = (n, i, j) \in S' \\ i, & \text{se } z_k = (n, i, j) \in S^0 \end{cases},$$

sendo que e_j é um vetor coordenada, com 1 na j-ésima linha e zeros nas demais posições. Para qualquer evento E, o elemento $\mathbb{1}_E$ indica a função indicadora desse evento: $\mathbb{1}_E = 1$ quando o evento E se verifica e 0 caso contrário.

A distribuição de demanda em um certo estágio i resulta do acúmulo dos pedidos individuais registrados durante esse estágio. Assim, a distribuição de demanda em cada estágio depende da duração do estágio e da distribuição dos pedidos individuais. Dessa forma, quando o sistema está evoluindo em um dado subconjunto $S^j,\ j \in \mathcal{J}$, a distribuição total de demanda em um dado estágio de produção depende da distribuição dos pedidos individuais e da j-ésima componente do vetor i, que determina o tempo de duração σ_k do período k ($\sigma_k =$ duração de $\Theta_{j\,i(j)}$). Quando o sistema está evoluindo em S^0, a distribuição da demanda total depende das distribuições de pedido individuais e do estágio atual de produção, que determina o tempo de espera até a próxima decisão (duração de Δ_i).

As distribuições de demanda total para cada subconjunto $j \in \mathcal{J}$ são apresentadas abaixo:

$$p_d^{j\,i(j)} = P(d_k = d | j_k = j,\ i_k(j) = i(j)), \quad z \in S',$$

sendo que $d = [d_1,\ \ldots,\ d_J]^T$ pertence ao conjunto $\mathcal{D}^{j\,i(j)} = \prod_{m=1}^{J}\{0, \ldots, D_m^{j\,i(j)}\}$ e $D_m^{j\,i(j)}$ é a máxima demanda admissível do produto m para $j_k = j$ e $i_k(j) = i(j)$. A distribuição de demanda em S^0 é dada por

$$q_d^i = P(d_k = d | j_k = 0, i_k = i),\ z \in S^0,$$

sendo que $d = [d_1,\ \ldots,\ d_J]^T$ pertence ao conjunto $\mathcal{D}^{0\,i} = \prod_{m=1}^{J}\{0, \ldots, D_m^{0\,i}\}$ e $D_m^{0\,i}$ é a máxima demanda admissível do produto m quando $j_k = 0$ e $i_k = i$.

As transições do processo de estoque $k \to n_k$, com $z_k = (n, i, j)$, são dadas por

$$n_{k+1} = \begin{cases} n + \eta_j \mathbb{1}_{\{i(j)=I_j\}} e_j - d_k, & \text{se } z_k \in S' \\ n - d_k, & \text{se } z_k \in S^0. \end{cases}$$

Assim, as probabilidades de transição do processo $k \to z_k$ assumem a seguinte forma:

$$P(z_{k+1} = z | z_k = (n, i, j)) = \begin{cases} p_d^{j\,i(j)}, & \text{se } z = \left(n + \eta_j \mathbb{1}_{\{i(j)=I_j\}} e_j - d,\ i + \left(1 - i(j)\mathbb{1}_{\{i(j)=I_j\}}\right) e_j,\ j\right), \quad z_k \in S' \\ q_d^i, & \text{se } z = (n - d,\ i,\ j), \quad z_k \in S^0 \\ 0, & \text{caso contrário.} \end{cases} \tag{2.1}$$

A equação (2.1) mostra as probabilidades de transição $P(z_{k+1} = z | z_k = z')$ em todo estado $z' \in S$ para o processo livre, sem intervenções. A Figura 2.1 ilustra as probabilidades de transição do processo $k \to z_k$ em um dado *subconjunto de produção* $S^j \in S'$. Observe que, ao fim do último estágio de produção, a produção do produto j é reiniciada e η_j unidades desse produto são entregues.

A figura 2.2 ilustra as probabilidades de transição do processo $k \to z_k$ no *subconjunto de paralisação* S^0. Nesse subconjunto, na ausência de intervenções, o processo $k \to i_k$ permanece constante.

O processo sem intervenções forma uma cadeia de Markov discreta com $(J + 1)$ classes não comunicantes representadas pelos conjuntos $S^j,\ j \in \mathcal{J}_0$. Considere as seguintes hipóteses

A_1 Para toda classe de produtos $j = 1, \ldots, J$, temos

$$E[n_{k+I_j}(j) | z_k = (n_k, i, j)] > n_k(j),\ j_k, \ldots, j_{k+I_j} = j,\ \forall k \geq 0.$$

A_2 $q_0^i < 1,\ \forall i \in \mathcal{I}$.

A hipótese A_1 implica que o sistema de produção é capaz de satisfazer a demanda, em média; caso contrário, déficits cada vez maiores poderiam ser acumulados com o sistema produzindo à máxima taxa. Como conseqüência de A_1, se o sistema não visita o subconjunto S^0, i.e. $z_k \in S',\ \forall k$, pode-se verificar que $E\{n_k(j)\} \uparrow N_j^+$ para algum $j \in \mathcal{J}$. A hipótese A_2 implica existência de demanda para todo produto fornecido pelo sistema. Se nenhuma intervenção

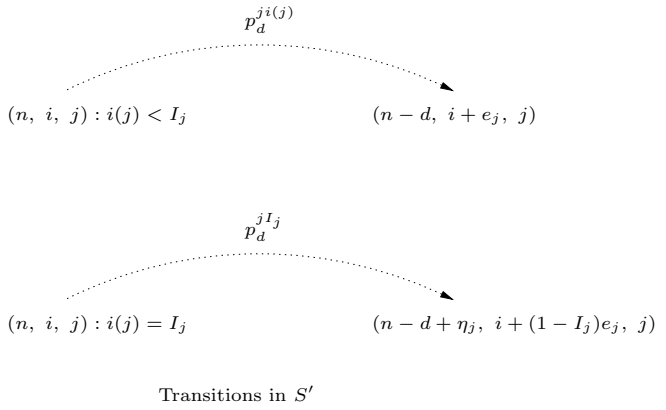

Fig. 2.1: Probabilidades de transição nos subconjuntos de produção, sem intervenções.

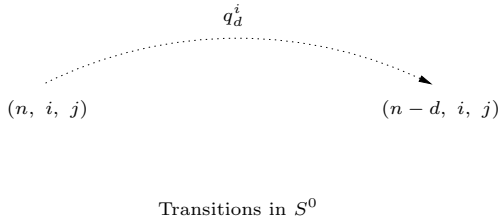

Fig. 2.2: Probabilidades de transição para o subconjunto de paralisação, sem intervenções.

ocorrer em S^0, $E\{n_k(j)\} \downarrow N_j^-$, para todo $j \in \mathcal{J}$. Dessa forma, pode-se concluir que o sistema é transiente a menos que se alterne períodos dentro e fora de S^0 de maneira a manter o nível de estoque finito. Particularmente, todo subconjunto $S^j,\ j \in \mathcal{J}_0$, forma uma cadeia de Markov transiente na ausência de intervenções.

Por meio de decisões de controle, deseja-se combinar as $(J+1)$ cadeias de Markov mencionadas no parágrafo anterior em classes de comunicação, de maneira a manter-se o processo de nível de estoque $k \to n_k$ finito (no sentido da média estocástica) num horizonte infinito de operação. Mais precisamente, busca-se condições para que a cadeia resultante possua um conjunto finito de estados recorrentes positivos, o que implica estabilidade estocástica no sentido Foster-Lyapunov, vide (Brémaud 1999). A discussão sobre estabilidade estocástica será retomada no próximo capítulo. A seguir apresenta-se extensões do modelo introduzido nesta seção.

2.4.1 Extensões do Modelo

Apresenta-se aqui duas possíveis extensões do modelo apresentado na Seção 2.4. A primeira extensão indica maneiras de se acomodar falhas de máquinas no modelo e a segunda introduz decisões em tempo contínuo quando o sistema está paralisado.

Quebra de Máquinas

A probabilidade de quebra de uma máquina durante um estágio de produção, quando não desprezível, pode ser levada em conta na distribuição final do tempo de duração deste estágio.

Uma idéia bem simples, explorada a seguir, é a de assumir-se um tempo constante de reparo r no caso de quebra de máquina.

Seja $p_l(i,\delta),\ l = 1,\ldots$ a probabilidade de que ocorram l quebras durante o estágio i, sendo δ = duração de Θ_i; obviamente, a possibilidade de quebra de máquinas não precisa ser considerada quando o sistema está paralisado. Dadas as definições acima, sabe-se que $\sigma_k = \delta + rl$ com probabilidade $p_l(i,\delta)\cdot dF_{\Theta_i}(\delta)$, sendo F_{Θ_i} a distribuição do tamanho do intervalo Θ_i. Naturalmente, as distribuições de chegada de demanda a cada estágio i devem ser calculadas de acordo com a distribuição do tempo de duração do período correspondente, considerando-se as probabilidades de quebra de máquina.

Um Modelo Híbrido Com Decisões a Tempo Contínuo em S^0

Em alguns sistemas, pode ser natural que, estando a produção paralisada ($z_k \in S^0$), opte-se por esperar que aconteça o primeiro salto de demanda antes de se definir a ação de controle a ser aplicada. Nesse caso, os intervalos de decisão $\Delta_i,\ i \in \mathcal{I}$ deixam de ser intervalos pré-determinados e passam a ser variáveis aleatórias a tempo contínuo. A título de exemplo, suponha que as chegadas de demanda para cada produto j são independentes e ocorrem em tempo contínuo a uma taxa $\delta_j > 0$, formando assim um processo clássico de Poisson. Assuma ainda que os pedidos individuais para todo produto j formam uma seqüência de variáveis independentes e identicamente distribuídas $(iid)\ \omega_k(j), k = 1,2,\ldots,\ \omega_k,\ k = 1,2,\ldots$, cada qual com distribuição $P(\omega_k(j) = m) = p_m(j),\ m = 1,2,\ldots,\ell(j)$, sendo $\ell(j)$ o máximo pedido individual aceito para o produto j. Seja τ_1 o instante de ocorrência do primeiro salto por demanda em um dado estado $z = (n,i,0)$ no subconjunto de paralisação. Pode-se definir uma medida de transição baseada na distribuição de pedidos apresentada acima. Para toda função $\phi : S \to \mathbb{R}$ a medida de transição é definida como

$$\mathcal{Q}[\phi](n,i,0) := E\big[\phi(z_{\tau_1})|z_{\tau_1^-} = (n,i,0)\big] = \sum_{j=1}^{J}\frac{\delta_j}{\lambda}\sum_{k=1}^{\ell(j)} p_k(j)\phi(n - k\,e_j, i, 0), \tag{2.2}$$

sendo que $\lambda := \sum_{j=1}^{J}\delta_j$. A evolução do processo em S^0 será caracterizada pela medida de transição $\mathcal{Q}$, até que uma intervenção transfira o sistema para o conjunto complementar S'.

Em geral, se existe uma taxa que caracteriza o processo de chegada de demanda, a evolução do sistema pode ser modelada por meio da classe de Processos Markovianos Determinísticos por Parte (PMDP), veja (Davis 1993). A demanda final em um dado intervalo de produção Θ_i é dada pelo processo de Poisson correspondente "modulado" pela distribuição da duração do intervalo Θ_{ji}.

2.5 O Problema de Controle Impulsional

Seja $n \to L(n,i)$ uma função representando o custo de estoque/déficit por unidade de tempo. Seja $\beta_{j\,i(j)}$ o custo instantâneo de produção de produtos do tipo $j \in \mathcal{J}$ em um dado estágio $i(j) \in \mathcal{I}_j$. A cada período $k \geq 0$, com $z_k = (n,i,j)$, paga-se um custo $h(z_k)$, definido como

$$h(n,i,j) = \begin{cases} (L(n,i) + \beta_{j\,i(j)})E\{\text{duração de } \theta_{j\,i(j)}\}, & \forall z \in S' \\ L(n,i)E\{\text{duração de } \Delta_i\}, & \forall z \in S^0. \end{cases} \tag{2.3}$$

No início de cada período k, uma intervenção pode ser aplicada ao sistema, a critério do decisor, resultando numa mudança no modo de produção. Uma intervenção pode paralisar a produção, reiniciá-la , ou ainda transferir os esforços de produção para um outro produto. Se uma intervenção é aplicada no período k, o processo $k \to z_k$ é instantaneamente transferido de $z_{k^-} = (n, i, j) \in S^j$ a um ponto $z_k = \bar{z}_{k^-}$ no conjunto de pontos de destino admissíveis $\text{dest}(z_{k^-}) := \{(n, i, \ell) \in S^\ell : \ell \neq j\}$. Além disso, paga-se um custo instantâneo de intervenção $g(z_{k^-}, z_k)$. Considere a seqüência de instantes de intervenção $\pi = \{k \geq 0 : z_{k^-} \neq z_k\}$, e defina-se por Π a classe de políticas de intervenção admissíveis tais que:

- $z_k \in \text{dest}(z_{k^-})$;
- $\Pi = \Pi(z_{k^-})$ só depende do ponto de origem (política estacionária).

Utiliza-se nesta tese programação dinâmica descontada, na qual um fator de desconto atenua os efeitos dos custos futuros. A cada período k, o fator de desconto é um número no intervalo $(0, 1)$ dependente do tempo de evolução do sistema, i.e. do valor de k, a ser multiplicado pelo custo observado nesse período. A cada período (j, i), define-se o fator de desconto como sendo

$$\alpha_{ji} = \begin{cases} e^{-\gamma\Theta_{ji(j)}}, & \text{se } z \in S' \\ e^{-\gamma\Delta_i}, & \text{se } z \in S^0, \end{cases} \tag{2.4}$$

sendo $\gamma > 0$ um escalar. Observe que, de acordo com a expressão (2.4), todo e qualquer fator de desconto assume valores no intervalo $(0, 1)$. A expressão (2.4) estabelece que os fatores de desconto são proporcionais ao tempo gasto em cada período $k \geq 0$. Além disso, $h(n, i, j)$ pode incluir um custo descontado em tempo contínuo no período correspondente. Nesse caso, a expressão (2.3) torna-se

$$h(n, i, j) = \begin{cases} (L(n, i) + \beta_{ji(j)}) E\{\int_0^{\text{duração de } \theta_{ji(j)}} e^{-\gamma t} dt\}, & \forall z \in S' \\ \\ L(n, i) E\{\int_0^{\text{duração de } \Delta_i} e^{-\gamma t} dt\}, & \forall z \in S^0. \end{cases} \tag{2.5}$$

Seja $E_z^\pi(X)$ o valor esperado da variável aleatória X, dado que a política de controle empregada é π e que a evolução do sistema inicia-se no ponto $z \in S$. Para uma dada política de intervenção $\pi \in \Pi$, o valor esperado do custo descontado de operação é dado por

$$V_\pi(z_0) = \lim_{N \to \infty} \sup \sum_{k=0}^{N} E_{z_0}^\pi \Big[\prod_{t=0}^{k-1} \alpha_{j_t i_t} \left(h(z_k) + g(z_{k^-}, z_k) \mathbb{1}_{\{k \in \pi\}} \right) \Big], \quad \forall z_0 \in S. \tag{2.6}$$

Na expressão acima, o produtório do lado direito da igualdade indica o fator de desconto em cada período da evolução do sistema. O objetivo do problema de controle impulsional é encontrar uma política estacionária de intervenção π tal que

$$V^*(z_0) = \min_{\pi \in \Pi} V_\pi(z_0), \ \forall z_0 \in S. \tag{2.7}$$

2.5.1 Solução do Problema de Controle Impulsional

Assuma que as funções $h : S \to \mathbb{R}^+$ e $g : S \times S \to \mathbb{R}^+$ são limitadas e defina:

$$T_0V(z) = \begin{cases} h(z) + \alpha_{ji} \sum_{d\in\mathcal{D}^{ji}} p_d^{ji} E\big[V(z_1)|d_0 = d\big], \ z \in S' \\ h(z) + \alpha_{ji} \sum_{d\in\mathcal{D}^{i}} q_d^{i} E\big[V(z_1)|d_0 = d\big], \ z \in S^0, \end{cases} \tag{2.8}$$

$$T_1V(z) = \min_{\bar{z}\in \mathrm{dest}(z)} g(z, \bar{z}) + T_0V(\bar{z}), \tag{2.9}$$

$$TV(z) = T_0V(z) \wedge T_1V(z), \tag{2.10}$$

sendo $a \wedge b := \min\{a, b\}$ e $V : S \to \mathbb{R}^+$. A esperança em (2.8) é calculada usando-se as probabilidades de demanda total $P(z_{k+1}|z_k = z)$, apresentadas em (2.1), para todo $z \in S$. O custo em um período sem intervenção é dado pela equação (2.8), enquanto a equação (2.9) representa o custo em um período com intervenção. O custo ótimo em um período, expresso por (2.10), é o menor destes dois custos. Usando argumentos clássicos de programação dinâmica (PD), pode-se caracterizar a solução do problema de controle impulsional por meio do resultado a seguir.

Teorema 1. *T é um operador contrativo e V^* é o único ponto fixo de T.*

Embora o teorema 1 descreva a solução do problema em uma forma simples, elegante e concisa, através do operador T, a solução é obtida por um procedimento de programação dinâmica e sofre, portanto, do conhecido *mal da dimensionalidade*: o crescimento combinatório do número de estados em problemas de grande porte, que torna tais problemas computacionalmente intratáveis.

Cabe ressaltar que a taxa de contração do operador T ($\tilde{\alpha}$), definido na equação (2.10) é limitada acima pelo fator de desconto definido para o problema, isto é

$$\tilde{\alpha} = \max_{j\in\mathcal{J},\ i\in\mathcal{I}} \alpha_{ji}.$$

Para o modelo híbrido da seção 2.4.1, o operador T_0 em (2.8), definido no subconjunto S^0, assume a forma

$$T_0V(z) = \frac{1}{\hat{\lambda}}\{L(n, i) + \lambda\mathcal{Q}[V](n, i, j)\}, \quad z \in S^0, \tag{2.11}$$

sendo que $\hat{\lambda} := \lambda + \gamma$ e γ é expresso na equação (2.4). Para se chegar à expressão (2.11), utiliza-se a versão a tempo contínuo da função de produção e estoque h (2.5) no subconjunto de paralisação S^0. Os operadores T_1 e T permanecem como definidos em (2.9)-(2.10). A verificação de contratividade do operador em (2.11) é relativamente simples e pode ser encontrada em (Arruda 2002, Lema 3.4). Nesse trabalho, mostra-se que a taxa de contração do operador em (2.11) é dada por $\tilde{\alpha} = \frac{\lambda}{\hat{\lambda}}$.

Observação 1. *É possível correlacionar os operadores determinísticos do problema P&E contínuo que utilizam PMDP (discretizado no tempo e no estado) com o problema P&E discreto modelado por meio de cadeia de Markov discreta apresentado neste trabalho.*

2.6 Notas Sobre Políticas de Intervenção

O problema de controle impulsional definido na Seção 2.5 é resolvido por meio de uma política de controle por intervenções π que satisfaça a expressão (2.7). Como mencionado na Seção 2.5.1, essa solução é obtida por meio da aplicação de um operador de programação dinâmica discreta no espaço de estados S do sistema. Trata-se do mesmo processo utilizado na obtenção de políticas ótimas em modelos de PDM's clássicos. No entanto, como brevemente mencionado na Seção 2.2, não se trata de um PDM clássico. Ainda assim, é possível utilizar-se operadores de PD clássicos na solução do problema. Uma discussão mais detalhada sobre esse assunto é apresentada a seguir, na Seção 2.6.2. Antes dessa discussão, apresenta-se uma seção dedicada a PDM's clássicos, que será útil no decorrer do capítulo.

2.6.1 Processos de Decisão Markovianos

Um PDM é um modelo de decisão seqüencial contendo os seguintes elementos:

1. Um conjunto de instantes de decisão.
2. Um conjunto de estados.
3. Um conjunto de ações de controle.
4. Uma função de custo instantâneo.
5. Um conjunto de probabilidades de transição.

A notação e a terminologia apresentadas acima e no decorrer desta seção, seguem de perto àquelas utilizadas em (Puterman 1994). Cada um dos componentes acima é descrito mais detalhadamente a seguir. A descrição apresentada abaixo é voltada a sistemas em tempo discreto, dada a natureza discreta da modelagem proposta para o problema P&E abordado neste trabalho.

Em um PDM, o *decisor*, *agente* ou *controlador* tem a oportunidade (ou o desafio) de influenciar o comportamento probabilístico do sistema à medida que este evolui no tempo. O objetivo é definir uma seqüência de ações de controle de modo a fazer com que o sistema tenha um desempenho ótimo com respeito a algum critério definido à priori.

Para um PDM em tempo discreto, define-se um conjunto discreto de instantes de decisão que coincidem com o início de cada período. O número total de períodos na evolução do sistema é denominado *horizonte do problema*. PDM's são, em geral, divididos em duas grandes classes: a classe de problemas em horizonte finito, e a classe complementar de problemas em horizonte infinito.

A cada instante de tempo, o sistema ocupa um *estado* e o conjunto de estados possíveis do sistema, comumente denominado *espaço de estados*, é denotado por S. Nos instantes de decisão, o controlador escolhe uma dentre as ações de controle disponíveis, dado o estado atual do sistema. O conjunto de ações de controle do sistema, comumente chamado A_s, pode ser finito ou infinito. Para os propósitos deste trabalho, assume-se que S é um conjunto enumerável e que A_s é um conjunto finito.

Como conseqüência da aplicação de uma dada ação de controle $a \in A_s$ em um dado período k, e em um estado $s \in S$, o sistema está sujeito a um custo imediato $h(s, a)$. Além disso, o próximo estado na evolução do sistema é determinado pela distribuição de probabilidade $p_k(\cdot|s, a)$. A função $p_k(j|s, a)$ é denominada *função distribuição de probabilidade* e satisfaz a seguinte relação:

$$\sum_{j \in S} p_k(j|s, a) = 1.$$

Para todo estado $j \in S$, busca-se minimizar o custo de operação, i.e., a somatória dos custos -$h(\cdot, \cdot)$- em cada período, ao longo da evolução do sistema. Ocasionalmente, pode-se definir um fator de desconto dependente do tempo de evolução do sistema. Nesse caso, busca-se minimizar a somatória dos custos descontados durante a evolução do sistema. O critério de custo descontado foi utilizado na modelagem do problema P&E da Seção 2.4.

Políticas de controle podem ser definidas como seqüências de regras de decisão a serem utilizadas durante a evolução do sistema. Uma dada política π especifica uma seqüência de regras de decisão, i.e. $\pi = d_1, d_2, \ldots$, tal que $d_t = \{d_t(s),\ s \in S\}$, sendo que $d_t(s)$ especifica a ação de controle que deve ser empregada se o sistema visita o estado s no período t de sua evolução.

Uma política de controle é classificada como estacionária quando $d_t = d$ para todo período t. Para políticas de controle estacionárias, toda ação de controle $d(s)$ depende unicamente do ponto de origem (s). Assim, uma política estacionária d pode ser especificada como um conjunto de ações de controle, cada qual associada a um dado estado $s \in S$. A ação de controle $d(s)$ deve ser empregada toda vez que o sistema visita o estado s. É sabido que a política ótima de um dado PDM é estacionária, veja (Puterman 1994) (seção 5.5). Assim, a busca pela política ótima de um dado PDM pode, sem perda de generalidade, ser restrita ao conjunto de políticas estacionárias.

A seguir, apresenta-se uma breve discussão acerca da natureza das ações de controle no contexto de controle impulsional, também denominado controle por intervenções.

2.6.2 Um Panorama Sobre Controle por Intervenções

Busca-se, nesta seção, apresentar uma breve discussão comparativa entre o controle por intervenções, também denominado *controle impulsional*, utilizado no modelo da Seção 2.4, e a classe de políticas de controle estacionárias definida em formulações clássicas de PDM's.

Inicia-se a discussão com um exemplo ilustrativo apresentado na Figura 2.3. Essa figura mostra um exemplo de política de intervenção para um problema de um único produto. A coluna do lado esquerdo representa o subconjunto de produção (S^1), ao passo que a coluna do lado direito representa o subconjunto de paralisação (S^0). O eixo ξ indica a progressão da produção; quando $\xi = \Gamma$ um novo lote de η itens é entregue ao sistema. Linhas pontilhadas indicam intervenção e linhas contínuas indicam não intervenção. Observe que a política ótima é permanecer produzindo para todo $n \leq n_*^{'}$ e paralisar a produção sempre que $n > n_*^{''}$. Quando paralisado, o sistema deve retomar a produção sempre que $n \leq n_*^{''}$. Observe que os estados $\{z \in S^1 : n > n_*^{'}\}$ jamais serão visitados pelo sistema (a não ser instantaneamente). Estes estados podem, portanto, ser excluídos da cadeia de Markov induzida pela política ótima. Essa mesma observação vale para o conjunto de estados $\{z \in S^0 : n \leq n_*^{''}\}$.

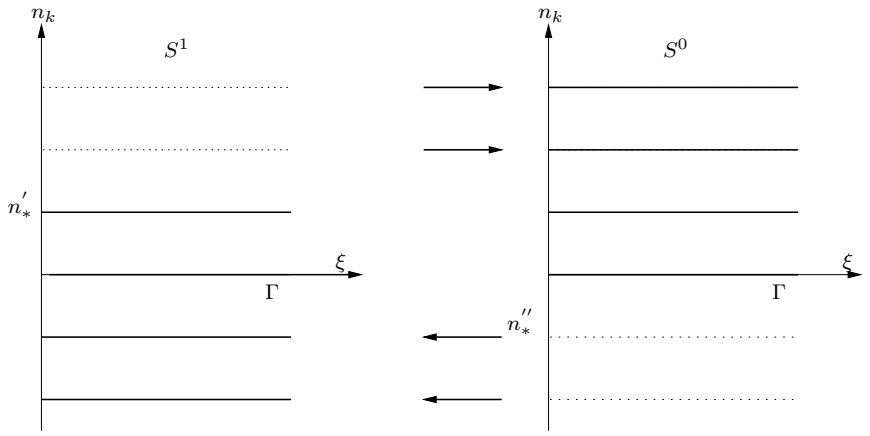

Fig. 2.3: Exemplo de Política de Intervenção

.

Como ilustrado na Figura 2.3, uma política de controle por intervenções estabelece a cadeia de Markov sobre a qual o sistema opera. Isto é, ela redefine o espaço de estados, excluindo os estados para os quais uma intervenção é prescrita. Essa característica não está presente em ações de controle associadas a PDM's padrão. Vale ressaltar que, além de definir o domínio da cadeia de Markov em regime estacionário, uma política de controle por intervenções também acumula a prerrogativa de controlar as transições do sistema. Esta última característica também é verificada nas políticas de controle associadas a PDM's padrão. Logo, uma política de intervenção pode ser vista como uma generalização de uma política de controle padrão em PDM's que agrega em si a definição da cadeia de Markov em que o sistema opera.

Para finalizar, convém recordar que uma política de controle padrão influencia a natureza estocástica dos estados pertencentes ao PDM, implicando em recorrência ou transiência. Já uma política de controle por intervenções define o espaço de estados em que o sistema opera e influencia a natureza estocástica de cada estado, implicando em recorrência ou transiência.

O Operador de Programação Dinâmica em Problemas de Controle Por Intervenção

Considere o operador T na equação (2.10) e observe que o domínio desse operador é o espaço de estados S e não o conjunto de estados de não intervenção induzido por uma determinada política de controle π, o qual será denotado por S_π. Observe também que a contratividade de T se deve inteiramente ao operador de não intervenção T_0, apresentado em (2.8).

Note que o operador T assume o mesmo valor do operador T_1 (2.9) nos estados de intervenção. Assim, T_1 é destinado a capturar as ações de intervenção, ao passo que T_0 define os estados no domínio da cadeia de Markov induzida pela política de controle, isto é, os estados de não intervenção.

A cada iteração, o operador T calcula uma função valor aproximada relativa à política gulosa (greedy) π, determinada pela aproximação da iteração anterior, para os estados de não intervenção através do operador contrativo T_0. Ao mesmo tempo, a função valor aproximada nos estados de intervenção é calculada através de T_1, utilizando os valores de T_0 recém calculados para os estados de não intervenção. Dessa forma, o operador T pode ser visto como uma

combinação de um operador de não intervenção T_0 e um operador de intervenção T_1. Note que os valores de T_0 calculados para os estados de intervenção são descartados em (2.10). O mesmo acontece para os valores de T_1 nos estados de não intervenção. Pode-se, portanto, definir o domínio de T_0 como os estados de não intervenção ($z \in S_\pi$). Analogamente, o domínio de T_1 pode ser definido como o conjunto estados de intervenção ($S \setminus S_\pi$). O espaço de estados original S é a união desses dois domínios.

Com base no que foi discutido acima, pode-se afirmar que o operador T da equação (2.10) é um operador generalizado, destinado a estender o domínio do problema de controle impulsional para todo o espaço de estados. A referida extensão é obtida através da composição de um operador de não intervenção T_0 e um operador de intervenção T_1. Essa extensão é útil considerando-se que os estados de intervenção de uma dada política de controle são desconhecidos à priori. A aplicação do operador T nos permite iterar em S e identificar os estados de intervenção à posteriori, obtendo assim o domínio da cadeia de Markov associada à política ótima e a função valor correspondente.

2.7 Considerações Finais e Contextualização

O propósito deste capítulo foi trazer um panorama geral do problema de produção e estoque (P&E) abordado neste trabalho e introduzir uma nova proposta de modelagem do problema por meio de programação dinâmica discreta. Apresentou-se um modelo estocástico em horizonte infinito com custo descontado para o problema P&E e políticas de controle por intervenção foram propostas para controlar a evolução do sistema. Definiu-se também um operador de programação dinâmica discreta destinado a encontrar a política de controle ótima, isto é, uma política de intervenção que minimiza um funcional de custo definido à priori em horizonte infinito.

Uma política de controle por intervenções pode ser vista como uma generalização de uma política padrão em PDM's que define o espaço de estados em que o sistema opera. Nesse sentido, a obtenção da política ótima se dá através da aplicação de um operador de programação dinâmica generalizado em todo o espaço de estados. A estrutura desse operador permite identificar a cadeia de Markov induzida pela política ótima ao mesmo tempo em que se obtém a função valor do problema.

Definida a modelagem do sistema e o procedimento de obtenção da solução ótima, pode-se pensar em caracterizar o comportamento estocástico do sistema. Essa caracterização será efetuada no capítulo seguinte, através dos conceitos de recorrência e transiência de estados pertencentes a uma cadeia de Markov. Esses conceitos estão ligados à estabilidade estocástica do sistema, isto é, à capacidade deste de se manter em uma região finita do espaço de estados em horizonte infinito. Estabilidade estocástica é, portanto, uma característica desejável para o problema. Por essa razão, pretende-se obter condições necessárias e suficientes para a estabilidade estocástica de problemas P&E.

Capítulo 3

Noções de Estabilidade Estocástica

3.1 Preliminares

Neste capítulo, utiliza-se a noção de estabilidade estocástica no sentido de Foster-Lyapunov para estabelecer condições necessárias e suficientes para a estabilidade de sistemas P&E (nesse mesmo sentido). Apresenta-se também uma noção geral da estrutura das cadeias de Markov de sistemas P&E controlados.

Os estados pertencentes à cadeia de Markov induzida por uma dada política de controle admissível são caracterizados. Essa caracterização tem estreita relação com a noção de estabilidade de políticas de controle admissíveis e define níveis de estoque críticos de modo a dividir o espaço de estados em três regiões distintas: uma região de aumento de estoque, uma região de decrescimento de estoque e uma região intermediária, denominada refúgio, onde o sistema tende a operar.

A noção de estabilidade em sistemas P&E, bem como a caracterização dos estados da cadeia de Markov induzida por ela serão utilizados no decorrer da tese no sentido de estabelecer rotinas de obtenção de políticas de controle sub-ótimas e estocasticamente estáveis. Essas rotinas são detalhadamente descritas no Capítulo 5.

Na próxima seção, apresenta-se uma breve introdução e comentários sobre estabilidade estocástica. Obtém-se, a seguir condições necessárias e suficientes para estabilidade estocástica nos sistemas P&E abordados. A isso se sucede uma caracterização dos estados pertencentes à cadeia de Markov induzida. Considerações finais são apresentadas em seguida, encerrando o capítulo.

3.2 Introdução

De maneira bastante geral, podemos dizer que um sistema dinâmico, seja ele determinístico ou estocástico, é estável se opera em uma região limitada ao longo de toda a evolução do sistema e as saídas dessa região limitada representam comportamentos transitórios, não persistentes. Dada a impossibilidade de se determinar em termos absolutos a região de operação de um processo de decisão markoviano (PDM), faz-se necessário definir conceitos probabilísticos que transmitam a noção de operação estável e instável em sistemas representados por PDM's. Os

conceitos de recorrência e transiência de cadeias de Markov homogêneas são utilizados para esse fim, veja por exemplo (Brémaud 1999) e (Meyn e Tweedie 1993).

O conceito de estabilidade de cadeias de Markov se estende aos sistemas que estas representam. Assim, estudar a estabilidade de um determinado PDM é equivalente a estudar a estabilidade da cadeia de Markov que o representa. Cada estado de uma dada cadeia pode ser classificado como recorrente ou transiente, sendo que o conceito de recorrência pode ser associado a estabilidade, ao passo que o conceito de transiência pode ser associado a comportamento instável. Isso não significa, contudo, que um sistema estocasticamente estável deva ser composto unicamente de estados recorrentes. Na concepção utilizada neste trabalho, estabilidade estocástica implica que estados transientes devem ser estados de passagem, levando a um conjunto de estados recorrentes, no sentido clássico de cadeias de Markov.

Dizemos que um estado é transiente se existe uma probabilidade positiva de que o sistema não mais retornará a esse estado. Se tal probabilidade for nula, o estado é recorrente. Estados recorrentes podem ser recorrentes nulos ou recorrentes positivos, sendo que à recorrência positiva associa-se uma noção mais forte de estabilidade. Neste trabalho utiliza-se o conceito de recorrência positiva de um conjunto de estados-alvo como fator indicativo de estabilidade no problema P&E estudado; todo estado transiente do sistema deve levar a cadeia ao conjunto alvo em tempo finito. Uma vez atingido o conjunto alvo, o sistema visita esse conjunto um número infinito de vezes (infinitely often) durante todo o horizonte (infinito) de operação.

Reproduz-se, a seguir, as definições matemáticas de recorrência, transiência e recorrência positiva mencionados anteriormente. Mais detalhes sobre estabilidade estocástica de sistemas descritos por cadeias de Markov homogêneas podem ser encontrados em (Brémaud 1999) e (Meyn e Tweedie 1993). Nas definições abaixo, τ_x representa o tempo de retorno a um dado estado $x \in S$.

Definição 1 (Transiência). *Um estado $x \in S$ é transiente se $P_x(\tau_x < \infty) < 1$.*

Definição 2 (Recorrência). *Um estado $x \in S$ é recorrente se $P_x(\tau_x < \infty) = 1$.*

Definição 3 (Recorrência Positiva). *Um estado $x \in S$ é recorrente positivo se x é recorrente e $E_x(\tau_x) < \infty$.*

3.3 Estabilidade de Sistemas P&E

Como vimos na seção 2.4, o problema P&E sem intervenções pode ser representado por $J+1$ cadeias de Markov homogêneas não comunicantes, cada qual representando um conjunto $S^j,\ j \in \mathcal{J}_0$. Ainda de acordo com o que foi mencionado na seção 2.4, os estados pertencentes às cadeias de Markov associadas aos conjuntos $S^j,\ j \in \mathcal{J}_0$, são transientes na ausência de intervenções de controle.

Deseja-se, portanto, regular o comportamento estocástico da cadeia de Markov controlada, de modo a manter o nível de estoque n_k finito (em valor esperado) para todo período k, obtendo-se assim um sistema P&E estocasticamente estável. No problema P&E da Seção 2.4, as políticas de controle admissíveis são aquelas que levam o sistema à estabilidade estocástica. Portanto, a minimização dos custos de operação através do problema de controle impulsional,

apresentado na Seção 2.5, é compatível com a estabilidade estocástica do sistema controlado. De fato, a busca por uma política ótima no sentido da seção 2.5 pode ser restrita ao conjunto de políticas de controle *estáveis*, i.e. políticas de controle que impliquem estabilidade estocástica do sistema P&E controlado.

Da teoria de processos de decisão markovianos, sabe-se que a política ótima de um PDM é estacionária, vide (Puterman 1994)-teorema 5.5.3. Dessa forma pode-se, sem perda de generalidade, restringir a análise de estabilidade estocástica do problema P&E estudado a políticas de intervenção estacionárias.

O critério de estabilidade a ser utilizado neste trabalho estabelece uma certa "região de atração" no espaço de estados do sistema, que faz as vezes de região de estabilidade. Para tanto, deve-se definir uma função de Lyapunov decrescente fora da região de atração. A idéia básica por trás desse critério é ilustrada na Figura 3.1. Note que à medida que a função de Lyapunov decresce, o processo $\{x_t : t \geq 0\}$ caminha em direção à origem.

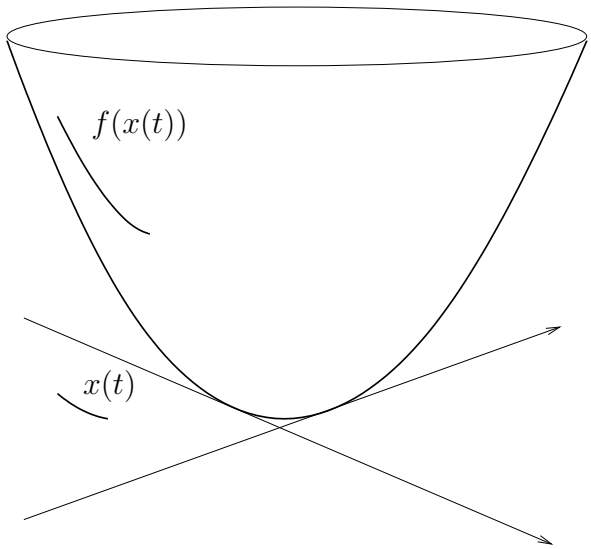

Fig. 3.1: Exemplo de Função de Lyapunov

Apresenta-se, a seguir, a definição do critério de estabilidade a ser utilizado.

Definição 4. *Uma política estacionária π é estável se a cadeia de Markov correspondente, com espaço de estados S_π, possui um conjunto finito e não vazio de estados recorrentes positivos.*

Sejam definidos um vetor de estoques negativos $n_B = [-B_1 \ \ldots \ -B_J]^T$ e um vetor de estoques positivos $n_C = [C_1 \ \ldots \ C_J]^T$, sendo $B_j, C_J \in \mathbb{N}$, $j \in \mathcal{J}$ constantes não negativas. Defina-se também as seguintes relações de ordem parcial:

$$x \succeq (\preceq)\ y \Rightarrow x_j \geq (\leq)\ y_j \ \forall j$$
$$x \succ (\prec)\ y \Rightarrow x \succeq (\preceq)\ y, \text{ e } x_j > (<)\ y_j \text{ para algum } j$$
$$x > y(<) \ \Rightarrow x_j > (<)\ y_j \ \forall j.$$

A principal hipótese desta seção aparece a seguir.

A₃ Existem políticas estacionárias π tais que para alguns vetores n_B e n_C, além de uma função limitada $n \to t_0(n) \in \mathbb{N}$ tais que

$$E\big[n_{k+t_0(n)} | n_k = n\big] > n, \text{ para todo } n \prec n_B, \tag{3.1a}$$
$$\text{e } E\big[n_{k+t_0(n)} | n_k = n\big] < n, \text{ para todo } n \succ n_C. \tag{3.1b}$$

A hipótese A_3 é natural em problemas P&E e indica que o sistema deve manter o nível de estoque limitado no sentido estocástico indicado acima. De fato, mostraremos no decorrer deste capítulo que a hipótese A_3 é condição necessária e suficiente para a estabilidade do sistema no sentido da Definição 4. Note, além disso, que o valor de estoque converge, em média, para uma região delimitada pelos valores n_B e n_C, em um sentido semelhante ao ilustrado na Figura 3.1.

Note em particular que a equação (3.1a) reforça a hipótese A_1 da Seção 2.4 e requer que o sistema seja capaz de satisfazer simultaneamente a demanda por todos os produtos ofertados, em valor esperado, o que indica uma acumulação de demanda não explosiva. A existência de uma política que satisfaça A_3 requer, no mínimo, que a capacidade de produção seja suficiente para que o nível de estoque aumente estritamente (em média) ao fim de um ciclo completo de produção no qual um lote de cada produto é completado em seqüência, sem interrupções. Comparando a equação (3.1b) e a hipótese A_2, verifica-se que a última é conseqüência da primeira, uma vez que (3.1b) implica existência de demanda.

Seja Π a classe de políticas estacionárias que satisfaçam A_3, seja S_F um subconjunto de S_π para alguma política $\pi \in \Pi$, definido abaixo

$$\begin{aligned} S_F &= \big\{\mathcal{N}_F \times \mathcal{I} \times \mathcal{J}_0\big\} \cap S_\pi, \\ \mathcal{N}_F &= \textstyle\prod_{j=1}^{J} \{-B_j, \ldots, C_j\}. \end{aligned} \tag{3.2}$$

Seja τ_F o tempo de retorno (ou tempo de chegada) do sistema a S_F, partindo de qualquer estado em S_π, para o processo controlado por uma política π. Verifica-se a seguir que, sob qualquer política de controle $\pi \in \Pi$, o processo π-controlado retorna a S_F um número infinito de vezes (do termo em inglês *infinitely often*).

Lema 1. *Assuma A_3. Então $E[\tau_F | z_0 = z] < \infty,\ \forall z \in S_\pi, z < \infty$.*

Demonstração. Os argumentos aqui apresentados seguem de perto aqueles contidos na prova do Teorema de Foster em (Brémaud 1999). Considere uma função $f : S \to \mathbb{R}_+$, tal que $f(n, \cdot, \cdot)$ é constante, a saber, $f(z = (n, i, j))$ é apenas função de n. Além disso, assuma que

$$E\big[f(z_{k+t_0(n)}) | z_k = (n,\ i,\ j)\big] < f(z),\ \forall z \notin S_F, z < \infty. \tag{3.3}$$

A hipótese A_3 implica na existência de uma função f satisfazendo a equação acima. Assuma, por exemplo, f como sendo a norma euclideana do vetor $n \in S_\pi$. Nesse caso, A_3 implica (3.3).

Defina uma seqüência de tempos de parada $\varphi = \{\varphi_0, \varphi_1, \varphi_2, \ldots\}$ para a cadeia de Markov controlada, sendo $\varphi_0 = 0$ e

$$\varphi_{k+1} = \begin{cases} \varphi_k + 1, & \text{se } z_{\varphi_k} \in S_F \\ \varphi_k + t_0(n_{\varphi_k}), & \text{se } z_{\varphi_k} \notin S_F. \end{cases}$$

sendo $n \to t_0(n)$ a função limitada definida em (3.1a)-(3.1b). Seja $\hat{z}_k := z_{\varphi_k}, k \geq 0$ e denote por $\hat{\tau}_F$ o tempo de retorno de $\hat{z}$ ao conjunto S_F. Defina também $Y_k = f(\hat{z}_k)\mathbb{1}_{\{k<\hat{\tau}_F\}}$ e seja $\hat{z}^k := \{\hat{z}_0, \ldots, \hat{z}_k\}$. Para um dado $z_0 = \hat{z}_0 \notin S_F$, sabe-se que

$$\begin{aligned} E\big[Y_{k+1}|\hat{z}^k\big] &= E\big[Y_{k+1}\mathbb{1}_{\{k<\hat{\tau}_F\}}|\hat{z}^k\big] + E\big[Y_{k+1}\mathbb{1}_{\{k\geq\hat{\tau}_F\}}|\hat{z}^k\big] \\ &= E\big[f(\hat{z}_{k+1})\mathbb{1}_{\{k<\hat{\tau}_F\}}|\hat{z}^k\big] = \mathbb{1}_{\{k<\hat{\tau}_F\}}E\big[f(\hat{z}_{k+1})|\hat{z}^k\big] \\ &\leq f(\hat{z}_k)\mathbb{1}_{\{k<\hat{\tau}_F\}} - \epsilon\mathbb{1}_{\{k<\hat{\tau}_F\}} = Y_k - \epsilon\mathbb{1}_{\{k<\hat{\tau}_F\}} \end{aligned}$$

para algum $\epsilon > 0$. Extraindo os valores esperados, obtém-se:

$$0 \leq E_{z_0}[Y_{k+1}\big] \leq E_{z_0}\big[Y_k\big] - \epsilon P_{z_0}(\hat{\tau}_F > k).$$

E iterando a expressão acima, pode-se concluir que

$$0 \leq E_{z_0}[Y_0\big] - \epsilon \sum_{\ell=0}^{k} P_{z_0}(\hat{\tau}_F > \ell).$$

Mas $Y_0 = f(z_0)$, com probabilidade um, e $\sum_{\ell=0}^{\infty} P_{z_0}(\hat{\tau}_F > \ell) = E_{z_0}\big[\hat{\tau}_F\big]$. Portanto, para todo estado $z \notin S_F$,

$$E_z\big[\hat{\tau}_F\big] \leq \epsilon^{-1} f(z) < \infty. \tag{3.4}$$

A desigualdade (3.4) é obtida observando-se que a norma $||z|| < \infty$. Já para $z \in S_F$, usando a análise de primeiro passo, temos

$$E_z\big[\hat{\tau}_F\big] = 1 + \sum_{y \notin S_F} p_{zy} E_y[\hat{\tau}_F] < \infty. \tag{3.5}$$

Por definição, qualquer trajetória $\{\hat{z}_0, \ldots, \hat{z}_k\}$ constitui uma subseqüência da trajetória $\{z_0, \ldots, z_{\varphi_k}\}$, com $z_0 = \hat{z}_0$ e $z_{\varphi_k} = \hat{z}_k$. Em particular, $\tau_F = \varphi_{\hat{\tau}_F}$ e $\{\hat{z}_0, \ldots, \hat{z}_{\hat{\tau}_F}\}$ é uma subseqüência de $\{z_0, \ldots, z_{\varphi_{\hat{\tau}_F}}\}$, com

$$\varphi_{\hat{\tau}_F} \leq \arg\max_{z \notin S_F}(t_0(n)) \cdot \hat{\tau}_F.$$

Considerando A_3, tem-se que $\arg\max_{z\notin S_F}(t_0(n)) = \bar{t}_0 < \infty$. Extraindo o valor esperado da expressão acima, obtém-se

$$E\big[\tau_F\big] = E\big[\varphi_{\hat{\tau}_F}\big] \leq \bar{t}_0 \cdot E\big[\hat{\tau}_F\big] < \infty.$$

A última desigualdade é obtida usando-se (3.4)–(3.5), e assim a demonstração se faz completa. □

O Lema 1 garante a existência de um conjunto recorrente positivo S_F. Dado que S_F é finito e considerando que o tempo esperado de retorno a S_F é finito, todo estado recorrente pertencente ao conjunto S_F é, de fato, recorrente positivo. Podemos, em conseqüência dessas considerações, formular o seguinte Corolário ao Lema 1.

Corolário 1. *Toda política estacionária π que satisfaça A_3 é estável.*

Comentário 1. *Note que a cadeia de Markov com espaço de estados S_π pode incluir estados transientes e pode não ser irredutível, i.e. pode possuir mais de uma classe de comunicação, veja (Brémaud 1999). Se a cadeia de Markov controlada for irredutível e considerando que recorrência positiva é uma propriedade de classe, o Lema 1 implica que todos os estados em S_π são recorrentes positivos.*

Apresenta-se abaixo um exemplo simples a fim de ilustrar uma situação em que π define uma cadeia controlada redutível e uma outra situação em que π define uma cadeia irredutível. Além disso, indica-se estados transientes associados a altos níveis de estoque nos exemplos apresentados.

Exemplo 1. *Considere um problema P&E responsável pela produção de um único produto e com um único estágio de produção. Para esse sistema $\mathcal{N} = \mathbb{Z}$, $\mathcal{I} = \{1\}$ e $\mathcal{J} = \{0, 1\}$. Suponha que o sistema é controlado por uma política π tal que $z \in S^1$, com probabilidade um, se $n \leq 0$ e $z \in S^0$, com probabilidade um, se $n > 0$ e*

$$P(z_1|z_0 = z) = \begin{cases} 0.5, & se \quad z_1 = (n + \delta, 1, 1) \\ 0.1, & se \quad z_1 = (n, 1, 1) \\ 0.4, & se \quad z_1 = (n - \delta, 1, 1), \end{cases} \qquad z \in S^1$$

$$P(z_1|z_0 = z) = \begin{cases} 0.5, & se \quad z_1 = (n, 1, 0) \\ 0.1, & se \quad z_1 = (n - \delta, 1, 0) \\ 0.4, & se \quad z_1 = (n - 2\delta, 1, 0). \end{cases} \qquad z \in S^0$$

Assume-se primeiro que $\delta = 2$ e então faz-se $\delta = 1$. Com o sistema operando sob a política π, tem-se

$$\begin{aligned} E\big[n_{k+1}|n_k = n\big] &= (n + \delta) \cdot 0.5 + n \cdot 0.1 + (n - \delta) \cdot 0.4 \\ &= n + 0.1\delta, \quad \forall n \leq 0, \end{aligned} \tag{3.6}$$

$$\begin{aligned} E\big[n_{k+1}|n_k = n\big] &= n \cdot 0.5 + (n - \delta) \cdot 0.1 + (n - 2\delta) \cdot 0.4 \\ &= n - 0.9\delta, \quad \forall n \geq 1. \end{aligned} \tag{3.7}$$

Note que a cadeia controlada satisfaz A_3 com $n_B = 0$ e $n_C = \delta$ para todo inteiro δ positivo. Se $\delta = 2$, o grafo de transição para o sistema controlado pode ser decomposto em dois grafos desconexos, como mostrado na figura 3.2. Note que a política de controle define um limite superior no nível de estoque para cada subgrafo. Os limitantes superiores na figura 3.2 são, respectivamente, $n = 1$ e $n = 2$ e cada subgrafo representa uma cadeia irredutível assumindo valores pares e ímpares de nível de estoque, respectivamente. Aplicando o Lema 1 e o Corolário 1 para o presente problema, conclui-se que a cadeia controlada é estável e retorna infinitas vezes ao conjunto $S_F = \{1, 2\} \times \{1\} \times 0$.

Se, por outro lado, faz-se $\delta = 1$, o grafo de transição do sistema controlado, mostrado na figura 3.3, é irredutível no subconjunto $(\{-\infty, 1\}, \times\mathcal{I} \times \mathcal{J}) \cap S_\pi$, e aplicando o Lema 1 e o Corolário 1 obtém-se que a cadeia controlada é estável e retorna infinitas vezes ao conjunto

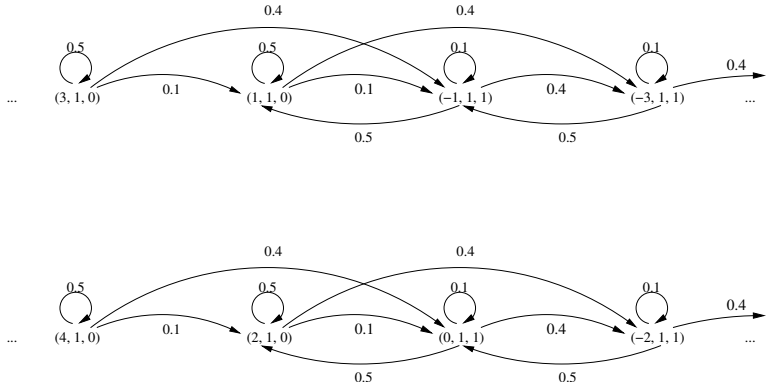

Fig. 3.2: Grafos de Transição para $\delta = 2$.

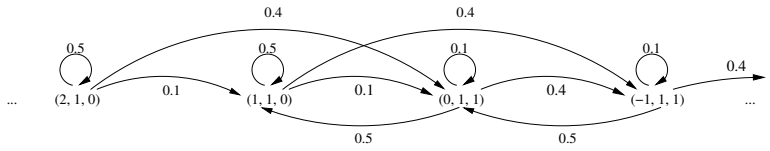

Fig. 3.3: Grafos de Transição para $\delta = 1$.

$S_F = \{1\} \times \{1\} \times 0$. *Além disso, dado que a cadeia controlada é irredutível, todos os estados pertencentes a* S_π *são recorrentes positivos, veja comentário 1.*

Observe em ambos os exemplos que a política de controle define um limitante superior ao nível de estoque, o que torna transientes os estados associados a níveis de estoques maiores que o limitante superior definido.

3.3.1 Estabilidade Implica A_3

Lema 2. *Assuma que uma dada política* π *é estável. Então a hipótese* A_3 *é verdadeira.*

Demonstração. Considerando a hipótese inicial de que a cadeia é estável, a política π define um conjunto estável S_F nos termos da equação (3.2). Assuma, por contradição, que A_3 não se verifica.

Seja uma função $f : S \to \mathbb{R}_+$ tal que

$$f(z) = \max_{j \in \mathcal{J}} |n(j)|.$$

Para negar A_3 é suficiente estabelecer $z \in S_\pi, z \notin S_F$ tal que $E[f(z_t)|z_0 = z] \geq f(z)$ para todo $t \geq 0$. Selecione um estado $z_0 \in S_\pi \setminus S_F$ tal que

$$f(z_0) > f(x),\ \forall x \in S_F. \tag{3.8}$$

Nessas condições, pode-se identificar uma seqüência de tempos de parada $\varphi = \{\varphi_0, \varphi_1, \varphi_2, \ldots\}$, $\varphi_0 = 0$, tal que $\varphi_k < \varphi_{k+t}, \forall t > 0$ e

$$E\big[f(z_{\varphi_{k+1}})|z_{\varphi_k}\big] \geq f(z_{\varphi_k}),\ \forall k \geq 0, \tag{3.9}$$

e verifica-se, portanto, que $\varphi_1 < \infty$ com probabilidade um. Defina $\hat{z}_k := z_{\varphi_k},\ k \geq 0$, $f_k := f(\hat{z}_k)$ e $\hat{z}^k = \{\hat{z}_0, \dots \hat{z}_k\}$. Considerando que o processo $z_k, k \geq 0$ apresenta a propriedade forte e Markov, a equação pode ser escrita na forma

$$E\big[f_{k+1}|\hat{z}^k\big] \geq f_k,\ \forall k \geq 0. \tag{3.10}$$

Seja τ_F o tempo de retorno de $\hat{z}_k, k \geq 0$ ao conjunto S_F; Segue da hipótese de estabilidade que $E[\tau_F] < \infty$. A equação (3.10), por sua vez, implica

$$0 \leq \sum_{k=0}^{\infty} \big(E\big[f_{k+1}|\hat{z}^k\big] - f_k\big)\, \mathbb{1}_{\{\tau_F > k\}} \tag{3.11}$$

$$= E\big[f_{\tau_F}|\hat{z}^{\tau_F - 1}\big] - f(z_0) + \sum_{k=1}^{\tau_F - 1} \big(E\big[f_k|\hat{z}^{k-1}\big] - f_k\big)\, \mathbb{1}_{\{\tau_F > k\}} \tag{3.12}$$

e dado que $E_{z_0}\left[E\big[f_k|\hat{z}^{k-1}\big] - f_k\right] = 0, \forall k \geq 1$, obtém-se que $E_{z_0}\big[f_{\tau_F}\big] - f(z_0) \geq 0$. Considerando, contudo, a escolha de z_0 em (3.8), tém-se necessariamente $f_{\tau_F} < f(z_0)$. Essa contradição mostra que A_3 é uma condição necessária de estabilidade. □

Do Corolário 1 e do Lema 2, segue a seguinte caracterização.

Teorema 2. *Assuma A_1–A_2. Uma política estacionária π é estável se e somente se A_3 é verdadeira.*

A conjugação de A_3 com o teorema 2 implica que qualquer política estacionária $\pi \in \Pi$ divide o espaço de estados estendido S_π em três subregiões distintas:

- Uma região de aumento de estoque, para altos níveis de demanda sob encomenda;
- Um *refúgio* S_F, a região na qual a política de controle tende a estabilizar o sistema, região essa visitada um número infinito de vezes;
- Uma região de estoque decrescente, para altos níveis de estoque.

Naturalmente, a política ótima define uma cadeia de Markov homogênea (CMH) estável e um refúgio ótimo S_F^*.

3.4 Considerações Finais e Contextualização

Foram apresentadas, neste capítulo, condições necessárias e suficientes para estabilidade estocástica de sistemas P&E. Além disso, os estados da cadeia de Markov induzida pela política de controle foram caracterizados.

A partir da caracterização da região de estabilidade S_F será desenvolvido no Capítulo 5 um algoritmo de PD aproximada destinado a obter soluções sub-ótimas estáveis. Para tanto, explora-se a caracterização da região de estabilidade, utilizando-se políticas arbitrárias, desde que estas definam uma região de estabilidade S_F qualquer. A região S_F é arbitrada à priori

porque não é possível estabelecer a região de operação S_F^* da política ótima sem o conhecimento prévio da política de intervenção ótima. Uma vez arbitrada a região de estabilidade desejada, itera-se o algoritmo de PD no interior dessa região fazendo uso de alguma condição de contorno arbitrária, obtendo-se assim as ações de controle correspondentes. Ao arbitrar S_F, é importante estabelecer uma região que contenha a região de estabilidade induzida pela política ótima, S_F^*. Dessa forma, as ações de controle referentes a todo estado no subconjunto S_F^* são definidas a partir da convergência do operador de PD aplicado. Infelizmente, não se pode verificar se a região arbitrada contém, de fato, o conjunto S_F^*, que é desconhecido à priori. No entanto, sabe-se que quanto maior a região S_F arbitrada, maior a probabilidade que isso se verifique.

Essa abordagem é proposta a partir da observação de que a região de estabilidade é mais significativa do ponto de vista de freqüência relativa. Em outras palavras, os estados nessa região são muito mais visitados em horizonte infinito que os demais e, portanto, contribuem de forma mais significativa para o custo final do problema. Assim, espera-se que uma boa aproximação da política ótima na região de estabilidade implique numa boa aproximação da função valor nessa região.

Capítulo 4

Programação Dinâmica Aproximada

4.1 Preliminares

Obtém-se, neste capítulo, um algoritmo de programação dinâmica aproximada (PDA) que, utilizando aproximação paramétrica de candidatas a função valor, apresenta convergência garantida para qualquer esquema de aproximação utilizado. Introduz-se, além disso, o conceito de projeção ótima, uma função pertencente ao esquema de aproximação (ou arquitetura de aproximação) que corresponde a um ponto fixo do operador de PD em uma norma definida à priori.

São dois os resultados novos obtidos. Primeiramente, introduz-se um algoritmo convergente para qualquer arquitetura de aproximação arbitrária e, em segundo lugar, apresenta-se uma caracterização do ponto de acumulação obtido pelo algoritmo.

4.2 Introdução

Introduzida por (Bellman 1957), a programação dinâmica (PD) é uma ferramenta concisa e elegante na solução de problemas que envolvam otimização de um custo cumulativo ao longo de vários estágios, tais como Processos de Decisão Markovianos (PDM). Uma desvantagem desse método é o crescimento exponencial no número de variáveis de estado à medida que o número de dimensões do problema é aumentado. Esse fenômeno, conhecido como *mal da dimensionalidade*, torna intratáveis computacionalmente problemas de grande porte.

No intuito de tratar as dificuldades oriundas do mal da dimensionalidade e reduzir o custo computacional associado à PD, diversos métodos de programação dinâmica aproximada (PDA) foram propostos, veja por exemplo (Hernández-Lerma 1989), (Bertsekas e Tsitsiklis 1996), (Sutton e Barto 1998), (Si, Barto, Powell e Wunsch 2004). Uma das abordagens utilizadas em PDA envolve a solução de aproximações sucessivas do problema real. Essas approximações melhoram gradativamente durante o processo iterativo de busca da solução aproximada. Para maiores detalhes acerca dessa abordagem, veja (Hernández-Lerma 1989, seção 2.4). Existe uma relação clara entre essa formulação e outros métodos de PDA, especialmente em se tratando de métodos *online*, tais como Q-Learning, onde a cada iteração o operador de programação dinâmica é aproximada por uma realização do sistema.

Outros métodos de PDA utilizam aproximações da função valor do problema. Esses métodos são conhecidos como métodos de *aprendizado por repetição* (do inglês *reinforcement learning*), veja (Sutton e Barto 1998), métodos de *programação neuro-dinâmica*, veja (Bertsekas e Tsitsiklis 1996) ou simplesmente métodos de *PD com representação aproximada*. Essa última designação será utilizada no decorrer deste trabalho para identificar os algoritmos propostos por fazer referência direta à aproximação da função valor por meio de representação paramétrica.

No decorrer desse capítulo apresenta-se uma discussão mais detalhada sobre métodos de PDA, com atenção particular aos métodos envolvendo aproximação da função valor do problema. O Capítulo 6 trata da utilização desses métodos na obtenção de soluções aproximadas para problemas P&E de múltiplos produtos, em situações para as quais seria inviável a obtenção de soluções exatas.

4.3 Métodos com Representação Tabular

Representações completas da função valor no espaço de estados S, seja em métodos exatos ou aproximados, são denominadas *representações tabulares*. Em representação tabular, os valores $V^*(x)$ para todo $x \in S$ são armazenados em uma tabela. Tratamos, nesta subseção, de métodos de PD aproximada utilizando representação tabular. Tais métodos podem ser aplicados quando o modelo probabilístico do problema não está disponível, ou ainda como alternativas de baixo custo computacional no intuito de se obter soluções aproximadas para problemas de PD.

Métodos exatos de PD, como iteração de valor, são aplicáveis quando um modelo explícito da estrutura de custos e das probabilidades de transição está disponível ao controlador. Em alguns casos, embora o referido modelo não esteja disponível, o sistema pode ser simulado. Isso significa que o espaço de estados e o conjunto de políticas de controle são conhecidos e pode-se simular, para uma dada política de controle π, o custo de um período $h(\cdot, \pi)$, assim como as transições probabilísticas de qualquer estado x para um estado sucessor y de acordo com as probabilidades de transição p_{xy} definidas por π. A partir desses dados, é possível obter estimativas dos custos esperados e das probabilidades de transição, através de métodos de aproximação estocástica que se valem de simulações sucessivas. De posse dessas estimativas, pode-se, num segundo momento, aplicar algum método exato de PD ao problema.

As estimativas dos custos unitários e das probabilidades de transição podem também ser obtidas implicitamente no processo de estimação direta da função valor do problema. Nesse caso, o problema é resolvido em uma única etapa, na qual o processo implícito de estimação das probabilidades de transição e o processo de obtenção de uma função valor aproximada são realizados concomitantemente. Essa abordagem se mostra bastante atrativa na solução de problemas *online*, i.e. problemas envolvendo simulação de um sistema cujas características probabilísticas sejam desconhecidas à priori.

Na literatura, o conceito de estimação de parâmetros é comumente associado a planejamento, e contrasta com o conceito de simulação do sistema, sendo este último relacionado com o processo de "aprendizado" de uma política de controle. A utilização conjunta de ambos os conceitos foi advogada por (Sutton 1990). Esse trabalho traz uma discussão conceitual relacionada a métodos de PD aproximada.

Diversos algoritmos que incorporam as idéias discutidas acima e utilizam representação

tabular podem ser encontrados na literatura; veja (Bertsekas e Tsitsiklis 1996) e (Sutton e Barto 1998). Dada a abrangência do tema e a pouca relevância de métodos com representação tabular em problemas de PD com espaço de estados de grande dimensão, apresenta-se neste trabalho apenas dois métodos clássicos dessa natureza. Serão discutidos a seguir os seguintes métodos de PD com representação tabular: *Q-Learning* e o *Método de Varredura Prioritária*.

O primeiro, por sua simplicidade e facilidade de implementação, é o método de simulação mais popular na área de programação dinâmica aproximada, freqüentemente utilizado como parâmetro de comparação. O segundo, embora menos difundido, é aplicado com sucesso a muitos problemas e também aplicável a sistemas para os quais existe um modelo estocástico disponível.

4.3.1 Q-Learning

Introduzido por (Watkins e Dayan 1989), esse método trabalha diretamente com estimativas da função valor ótima, de maneira análoga ao método de iteração de valor. Essas estimativas são obtidas por meio de avaliações dos custos associados a ações de controle específicas em estados selecionados do espaço de estados.

Ao invés de trabalhar diretamente com estimativas da função valor ótima, esse método utiliza o que se convencionou chamar de *Fatores-Q*, isto é, estimativas do valor ótimo para pares estado-ação-de-controle, na forma:

$$Q^*(x,\pi(x)) = h(x,\pi(x)) + \alpha \sum_{y\in S} p_{xy}(\pi(x))V^*(y), \forall x \in S \tag{4.1}$$

sendo α o fator de desconto. Assim, temos que

$$V^*(x) = \min_{\pi(x)} Q^*(x,\pi(x)), \forall x \in S.$$

Considerando a equação acima, pode-se escrever o algoritmo de iteração de valor na forma

$$Q^{k+1}(x,\pi(x)) = h(x,\pi(x)) + \alpha \sum_{y\in S} p_{xy}(\pi(x)) \min_{\pi(y)\in\pi} Q^k(y,\pi(y)), \forall (x,\pi(x)). \tag{4.2}$$

Pode-se também incorporar um *passo* variável β às iterações do algoritmo acima, obtendo-se assim a iteração abaixo

$$Q^{k+1}(x,\pi(x)) = (1-\beta)Q^k(x,\pi(x)) + \beta \sum_{y\in S} \big[h(y,\pi(y)) + \alpha p_{xy}(\pi(x)) \min_{\pi(y)\in\pi} Q^k(y,\pi(y))\big], \forall (x,\pi(x)). \tag{4.3}$$

O método Q-Learning utiliza uma versão aproximada da iteração acima, na qual o valor esperado expresso na somatória da equação acima é substituído por uma única amostra, isto é, uma realização do sistema a partir do estado x na forma

$$Q^{k+1}(x,\pi(x)) = (1-\beta)Q^k(x,\pi(x)) + \beta\big[h(x,\pi(x)) + \alpha \min_{\pi(y)\in\pi} Q^k(y(w),\pi(y))\big], \forall (x,\pi(x)). \tag{4.4}$$

Na equação acima, o estado y é gerado a partir do par $(x, \pi(x))$ por meio de simulação, de acordo com as probabilidades de transição $p_{xy}(\pi(x))$, que são inerentes ao processo de simulação mas podem não estar disponíveis ao controlador. Dessa forma, o algoritmo Q-Learning pode ser visto como uma combinação de simulação e iteração de valor. Equivalentemente, esse método também pode ser visto como uma aplicação do bastante difundido método de aproximação estocástica de Robbins-Monro (Robbins e Monro 1951).

Foi demonstrado que o algoritmo (4.4) converge para o valor Q^* em (4.1) quando o parâmetro β decai a uma taxa apropriada, a saber

$$\sum_{k=0}^{\infty} \beta_k = \infty,$$
$$\sum_{k=0}^{\infty} \beta_k^2 < \infty,$$

sendo β_k o fator de decaimento na iteração k, e todos os pares *estado-ação de controle* são visitados um número infinito de vezes. As condições de decaimento do parâmetro β_k foram introduzidas no contexto de métodos de aproximação estocástica em geral por (Robbins e Monro 1951). Para mais detalhes e provas de convergência do método de $Q-$Learning, recomenda-se uma consulta a (Watkins e Dayan 1989) ou (Bertsekas e Tsitsiklis 1996, pag. 248).

4.3.2 Método de Varredura Prioritária

O Método de Varredura Prioritária (do inglês *Prioritized Sweeping*), introduzido por (Moore e Atkeson 1993), baseia-se na utilização prioritária de esforço computacional em estados que apresentem maior potencial de variação na função valor entre duas iterações sucessivas. Este método pode ser utilizado em conjunto com qualquer algoritmo de PD exato ou aproximado. Em sua versão exata, o método de varredura prioritária pode ser visto como um algoritmo assíncrono de PD, no qual o esforço computacional é prioritariamente empregado de acordo com um critério pré-definido. Por se tratar de um esquema de priorização do esforço computacional empregado nas atualizações sucessivas das estimativas da função valor, o método de varredura prioritária pode ser aplicado em conjunção com qualquer algoritmo exato ou aproximado de programação dinâmica. Assim, na ausência de um modelo estocástico para o sistema, pode-se aplicar o método de varredura prioritária em conjunto com o algoritmo de Q-Learning ou qualquer outro algoritmo baseado em simulação.

Segue abaixo um exemplo de utilização do método de varredura prioritária em conjunção com PD, tal como apresentado em (Moore e Atkeson 1993). No Algoritmo 1, o estado $x_{recente}$ é o próximo estado a ser atualizado, sendo determinado pelo algoritmo utilizado em conjunto com o método de varredura prioritária. Esse estado pode ser obtido por meio de simulação ou enumeração, por exemplo. É importante ressaltar que as idéias associadas ao método de varredura prioritária se aplicam independentemente da ordem em que os estados são atualizados.

Note no Algoritmo 1 que todo estado visitado (atualizado) é adicionado ao topo da fila e atualizado. No passo seguinte, estima-se a influência da alteração na estimativa da função

Algoritmo 1 Método de Varredura Prioritária com Operador PD

1. Promover estado $x_{recente}$ ao topo da fila de prioridade. Fazer $k = 0$.
2. Enquanto $k < L_M$ e a fila de prioridades não for esvaziada:
   (a) $k \leftarrow k + 1$.
   (b) Remover o estado x do topo da fila de prioridades.
   (c) $\rho_{novo} \leftarrow \max_{\pi(x)} \left[h(x, \pi(x)) + \alpha \sum_{y \in S} p_{xy}(\pi(x))V(y)\right]$
   (d) $\Delta_{\max} \leftarrow |\rho_{novo} - V(x)|$
   (e) $V(x) \leftarrow \rho_{novo}$
   (f) Para todo y no conjunto de predecessores de x, i.e. $\{y \in S : p_{yx}(\pi(x)) > 0,\ \pi(x)$ factível$\}$ faça:
   - $P \leftarrow \max_{\pi(x)} p_{yx}(\pi(x))\Delta_{\max}$
   - Se $P > \epsilon$ (uma pequena tolerância) e se y não pertence à fila de prioridades ou se P é maior que a prioridade atual de y, atribua a y a nova prioridade P.

valor do estado visitado sobre os seus estados predecessores, ou seja, aqueles estados a partir dos quais este pode ser acessado em um único passo. Com base nessa estimativa atualiza-se a fila de prioridades. Então remove-se o estado no topo da nova fila, atualiza-se a estimativa de sua função valor e a fila de prioridades. Esse processo é repetido até que a fila seja esvaziada ou até que L_M estados sejam removidos do topo da fila. Quando uma dessas alternativas é contemplada, retorna-se ao passo 1, no qual um novo estado $x_{recente}$ é obtido e atualizado, dando início a uma nova iteração. O estado $x_{recente}$ pode ser parte de uma trajetória simulada, como é o caso quando empregamos algoritmos online. As iterações se sucedem até que um critério de parada seja contemplado.

A idéia inerente ao método de varredura prioritária é atualizar com mais freqüência estados mais "promissores" do espaço de estados de maneira a acelerar a convergência do algoritmo usado em conjunção com o método. A manutenção de uma fila de estados prioritários é a maneira pela qual os estados promissores são identificados. Naturalmente, há um custo computacional associado à implementação da fila de prioridades requerida pelo método e tal custo deve ser levado em consideração em qualquer análise de desempenho do algoritmo.

Para o problema abordado nessa tese, o modelo markoviano é conhecido. Portanto, a aplicação de métodos de simulação puros é inadequada. Ora, ao se simular um modelo conhecido, recursos computacionais valiosos são aplicados a fim de se obter uma estimativa do modelo já conhecido. Pode-se, portanto, utilizar os dados conhecidos à priori a fim de acelerar o processo de obtenção de uma solução aproximada, aumentando assim a eficiência do processo de busca da solução.

A utilização de métodos com representação tabular é pouco atrativa em problemas P&E de grande escala, considerando-se a impossibilidade de se armazenar os dados desse tipo de problema em representação tabular, devido à dimensão do espaço de estados. Assim, métodos com

representação aproximada devem ser utilizados. Esses métodos são introduzidos na próxima seção.

4.4 Métodos com Representação Aproximada

Em contraste com representações tabulares, discutidas na subseção anterior, representações utilizando arquiteturas que envolvam parâmetros de dimensão pequena em relação à cardinalidade do espaço de estados S são denominadas *representações compactas* ou *representações aproximadas*. Em uma representação compacta típica, apenas o vetor de parâmetros r e a estrutura geral da arquitetura de aproximação A são armazenados; os custos aproximados $\mathcal{V}(x)$ são gerados apenas quando estritamente necessário. Se, por exemplo, $\mathcal{V}(x)$ é a saída de uma rede neural, r é o vetor de pesos associados a essa rede; se $\mathcal{V}(x)$ é uma função polinomial, r é o vetor de coeficientes do polinômio.

Métodos programação dinâmica com representação aproximada (PDRA) são métodos sub-ótimos que utilizam aproximações da função valor ótima V^* do problema. Esses métodos requerem a definição de uma arquitetura de aproximação arbitrária; por exemplo o conjunto dos polinômios em S de ordem 2, ou um conjunto de nós e camadas que defina a estrutura de uma rede neural. Definida a arquitetura, busca-se em seu domínio uma função que aproxime V^* satisfatoriamente. Essa função é denominada "função valor aproximada". Definições formais dos elementos mencionados neste parágrafo são apresentadas a seguir.

Arquitetura de Aproximação Conjunto arbitrário de funções paramétricas. Cada elemento desse conjunto corresponde a um parâmetro.

Função Valor Aproximada Elemento da arquitetura correspondente a um parâmetro fixo.

Conjunto de Amostragem $\mathbf{M} \subset \mathbf{S}$ Conjunto de estados nos quais o operador de programação dinâmica é aplicado.

Operador de Projeção Operador que obtém funções valor aproximadas a partir de pares $(x, V(x))$, sendo x um estado pertencente ao conjunto amostragem e $V(x)$ o resultado da aplicação operador de programação dinâmica nesse estado.

Convém mencionar que a arquitetura de aproximação é definida à priori pelo usuário e permanece inalterada durante a execução de um algoritmo de PDRA.

Denota-se por A a arquitetura de aproximação utilizada, por $\mathcal{R}$ o conjunto de parâmetros admissíveis e por $\mathcal{V} := A(r),\ r \in \mathcal{R}$ um elemento da arquitetura, isto é, uma candidata a função valor aproximada. A estrutura da função $\mathcal{V}$ deve ser definida de forma que a avaliação de $\mathcal{V}(x)$ para qualquer estado $x \in S$ possa ser facilmente obtida.

Ao final do processo, substitui-se a função valor ótima $V^*(x),\ x \in S$ por uma aproximação $\mathcal{V}(x) = A(r, x)$ e utiliza-se no estado x um controle sub-ótimo $\tilde{\pi}(x)$ que satisfaça à expressão

$$\tilde{\pi}(x) = \arg\min_{\pi} E\big[h(x,\pi) + \sum_{y \in S} p_{xy}(\pi)\mathcal{V}(y)\big]. \tag{4.5}$$

Na expressão acima, $p_{xy}(\pi)$ denota a probabilidade de transição do estado x ao estado y. O diagrama na Figura 4.4 detalha o processo de obtenção do custo aproximado para um dado estado $x \in S$. Aplicando-se um vetor $r \in \mathcal{R}$ à arquitetura de aproximação A, obtém-se uma função valor aproximada $\mathcal{V}$. Para todo $x \in S$, o custo aproximado $\mathcal{V}(x)$ pode ser facilmente obtido aplicando-se x à aproximação $\mathcal{V}$.

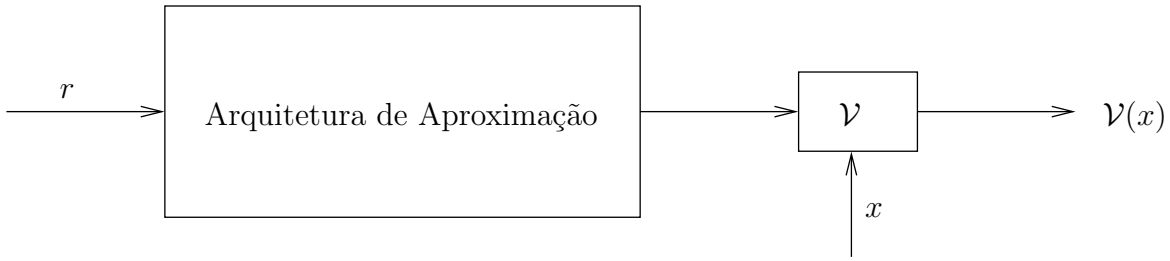

Fig. 4.1: Estrutura da Aproximação de Funções Valor

Busca-se, através do uso de aproximações, solucionar problemas com um grande número de estados utilizando-se arquiteturas de aproximação A associadas a vetores de parâmetros r de pequenas dimensões. Pretende-se, dessa forma, obter um algoritmo aproximado com custo computacional significativamente reduzido em relação ao algoritmo de PD padrão. O vetor de parâmetros r é melhorado iterativamente até que se obtenha uma função $\mathcal{V}$ que aproxime V^* satisfatoriamente. Assim, a determinação da função valor aproximada $\mathcal{V}$ envolve:

1. Definir a arquitetura de aproximação A;
2. Obter o vetor de parâmetros r de maneira a minimizar uma medida de erro entre V^* e $\mathcal{V}$.

4.4.1 Simulação e Treinamento

Vale ressaltar que, embora se deseje aproximar a função valor ótima V^* através de uma função paramétrica $\mathcal{V} \in A$, não se dispõe de um conjunto de amostras $(x, V^*(x))$ que poderia ser usado para se obter $\mathcal{V}$ por meio da minimização de uma medida de erro, por exemplo o erro quadrático médio. Deve-se, portanto, obter funções valor sub-ótimas através de um operador (exato ou aproximado) de programação dinâmica e tentar melhorá-las iterativamente. Cada função valor gerada no processo é aproximada por um elemento da arquitetura $\mathcal{V} \in A$ e aplica-se a esse elemento o operador de PD que gerará uma nova função valor a ser utilizada na iteração seguinte.

Cabe, portanto, definir um mapeamento do espaço de funções valor ao espaço de funções valor aproximadas A. Dado que pretende-se solucionar problemas de grande dimensão, é conveniente definir um mapeamento que gere aproximações a partir de um conjunto reduzido de amostras. Esse mapeamento, denotado por $\mathcal{P}_A$, é denominado **operador de projeção** e gera uma função valor aproximada $\mathcal{V} \in A$ a partir de um conjunto finito de pares $(x, V(x))$, sendo $x \in S$ um estado do sistema e $V(x)$ sua função valor, de maneira a minimizar uma medida de erro definida à priori (por exemplo, o erro quadrático médio). A Figura 4.2 abaixo ilustra o processo de obtenção de uma função valor aproximada a partir de pares de treinamento $(x, V(x))$.

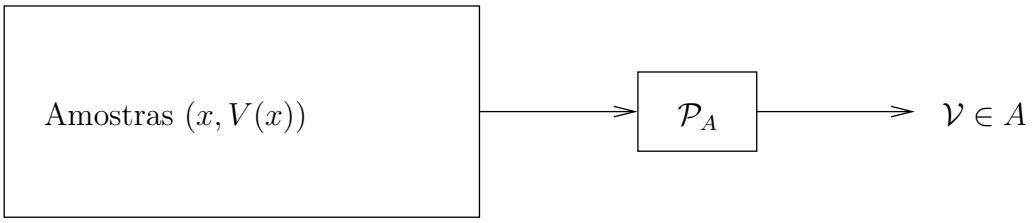

Fig. 4.2: Funcionamento do Operador de Projeção $\mathcal{P}_A$

Algoritmos PDRA alternam passos de obtenção de pares de treinamento, através de um operador de programação dinâmica (exato ou aproximado), e aplicações do operador de projeção $\mathcal{P}_A$. Esses passos se repetem um número indeterminado de vezes, até que um critério de parada pré definido seja satisfeito. Um esquema ilustrativo em malha fechada é apresentado na Figura 4.3.

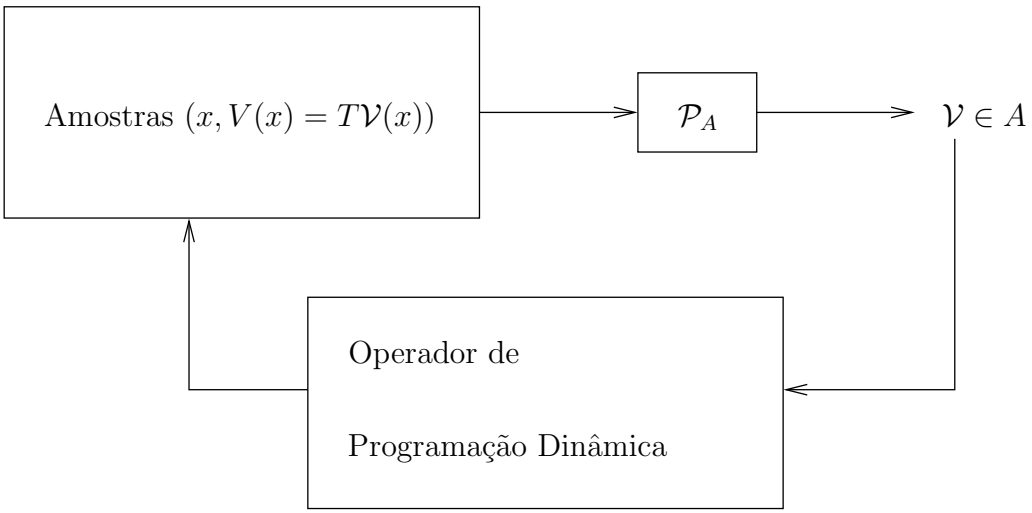

Fig. 4.3: Esquema em Malha Fechada do Algoritmo PDRA

Observando o esquema ilustrativo na figura 4.3, verifica-se que o algoritmo de PDRA é um procedimento recursivo em que, a cada iteração

1. Obtém-se pares de treinamento $x, V^k(x) = T\mathcal{V}^k(x)$;
2. Aplica-se o operador de projeção P_A a esses pares de treinamento, obtendo-se uma função aproximada $\mathcal{V}^{k+1}$.

É pertinente questionar se o procedimento recursivo apresentado acima converge. Em caso positivo, poder-se-ia indagar para quais arquiteturas de aproximação a convergência é garantida e o que se pode inferir sobre o ponto de acumulação. Esses pontos são abordados na próxima seção, na qual se apresenta uma discussão detalhada sobre os resultados existentes na literatura.

4.4.2 Notas Bibliográficas

A idéia de se utilizar uma arquitetura de aproximação no intuito de reduzir a complexidade e o espaço de busca de problemas de programação dinâmica tem sido bastante discutida e utilizada na literatura, veja (Si et al. 2004), (Bertsekas e Tsitsiklis 1996), (Sutton e Barto 1998). Essa abordagem mostrou-se satisfatória em algumas aplicações práticas; veja por exemplo (Tesauro 1992). Não obstante, existem relatos de divergência que põem em dúvida a universalidade dessa técnica. Veja, por exemplo, (Boyan e Moore 1995).

A partir dos relatos de divergência, surgiram vários algoritmos com convergência garantida, veja por exemplo: (Gordon 1995), (Baird 1995), (Tsitsiklis e Roy 1996), (Tsitsiklis e Roy 1999), (Reynolds 2002). Trata-se de algoritmos cuja convergência está condicionada ao emprego de uma determinada arquitetura de aproximação. (Gordon 1995) demonstrou que a combinação de um operador de projeção $\mathcal{P}_A$ não-expansivo em norma infinita com um operador de PD contrativo é estável e possui convergência garantida. Um resultado similar foi obtido por (Tsitsiklis e Roy 1996). A partir desses resultados, outros algoritmos de convergência garantida foram apresentados por (Baird 1995) e (Reynolds 2002) para o caso de arquiteturas de aproximação lineares. Nesse contexto, a função valor aproximada é dada por uma combinação linear de um vetor de parâmetros. Outros algoritmos com convergência garantida foram apresentados em (Perkins e Precup 2003) e (Szepesvári 2001). O volume (Si et al. 2004) contém uma coleção de resultados recentes no campo de programação dinâmica aproximada.

Embora interessantes, os algoritmos citados no parágrafo anterior se restringem a classes reduzidas de arquiteturas de aproximação, notadamente arquiteturas lineares e combinações convexas com propriedades específicas, tais como aquelas advogadas em (Gordon 1995). Mais detalhes sobre a abordagem proposta em (Gordon 1995) podem ser encontrados no Apêndice A. Estabelece-se, nesta tese, o objetivo de se obter resultados mais gerais no que se refere à convergência de algoritmos PDRA, aplicáveis a arquiteturas de aproximações arbitrárias. Nesse contexto, concentra-se a atenção em problemas com custo descontado. Introduz-se, para esses problemas, um procedimento sistemático de geração de algoritmos PDRA convergentes independente da arquitetura de aproximação utilizada. Isso é realizado por meio da monitoração de iterações sucessivas do algoritmo aproximado, num mecanismo análogo ao de controle de passo em problemas de otimização.

Introduz-se nesta tese o conceito de *expansões controladas*, isto é, operadores de projeção expansivos, mas que produzem mapeamentos contrativos quando combinados com operadores DP contrativos. Abandonando a noção de operadores de projeção não expansivos apresentada em (Gordon 1995), que foi provada convergente mas não necessariamente para a melhor aproximação, demonstra-se que expansões controladas em norma infinita possibilitam convergência para a projeção da função valor ótima no espaço da arquitetura de aproximação, independente da arquitetura de aproximação utilizada. Esse é um resultado encorajador, embora essencialmente teórico, uma vez que a avaliação em norma infinita é impraticável para espaços de estados de grande dimensão.

Entretanto, os resultados obtidos para a norma infinita podem ser estendidos para qualquer norma de subconjunto de interesse. Nesse contexto, pode-se derivar algoritmos que convirjam para a projeção de um ponto fixo do operador de PD - T - com respeito a uma norma de subconjunto, por exemplo a norma infinita no conjunto de M introduzido na Seção 4.4, assumindo-se

que T é contrativo com relação a essa norma. Esse algoritmo pode ser obtido por meio de uma monitoração da diferença entre iterações sucessivas do algoritmo aproximado, destinada a assegurar que o operador de projeção satisfaça à definição de expansão controlada em relação à norma de subconjunto utilizada. Além disso, o mecanismo de controle de diferença entre iterações sucessivas pode também ser usado a fim de assegurar a contratividade do operador de PD aproximada, que será introduzido na próxima seção, com respeito à norma de subconjunto. Nesse caso, embora não se tenha obtido garantias de desempenho, elimina-se a possibilidade de divergência e agrega-se confiabilidade ao algoritmo aproximado.

No contexto introduzido neste trabalho, algoritmos convergentes baseados na abordagem em (Gordon 1995) podem ser vistos como um caso particular. Nesses algoritmos, o passo de monitoração de iterações sucessivas é desnessário, uma vez que a propriedade de não expansão do operador de projeção limita a diferença entre as funções valor aproximadas obtidas em iterações sucessivas e assegura convergência.

4.5 O Algoritmo PDRA

Suponha que a solução de um PDM descontado exista e possa ser obtida por um mapeamento contrativo T. Nesse caso, a solução exata desse PDM coincide com o ponto fixo do operador T e pode ser obtida pelo algoritmo de iteração de valor (Puterman 1994)

$$V_{k+1} = TV_k,$$

a partir de qualquer função inicial $V_0 : S \to \mathbb{R}$. Seguindo a notação introduzida por (Gordon 1995), o algoritmo PDRA é definido abaixo:

$$\begin{array}{l} \mathcal{V}_0 \in A \\ \mathcal{V}_k = \mathcal{P}_A(T\mathcal{V}_{k-1}),\ \forall k \in \mathcal{N}, \end{array} \tag{4.6}$$

sendo que A é uma arquitetura de aproximação escolhida à priori, $\mathcal{P}_A$ é o operador de projeção introduzido na seção 4.4.1 e T é o mapeamento contrativo do algoritmo exato, como mencionado anteriormente. Note que V_k representa uma função valor exata, ao passo que $\mathcal{V}_k$ denota uma função valor aproximada, pertencente à arquitetura de aproximação A sendo utilizada. No decorrer deste trabalho, essa notação será mantida.

Quando convergente, um algoritmo PDRA converge para a função obtida da relação de ponto fixo

$$\bar{\mathcal{V}} = \mathcal{P}_A(T\bar{\mathcal{V}})$$

a qual pode ser avaliada por meio de iteração de valor aproximada. De maneira qualificar as soluções fornecidas pelo algoritmo (4.6), define-se a resposta ótima de um algoritmo PDRA como sendo o ponto $\mathcal{V}^* \in A$, tal que

$$||P_A(V^*) - \mathcal{V}^*||_M = 0,$$

sendo que na expressão acima V^* é interpretado como um ponto fixo do operador T com respeito à norma $||\cdot||_M$ e M é o conjunto de amostragem utilizado pelo operador de projeção $\mathcal{P}_A$. Vale

ressaltar que o ponto fixo do operador T com relação à norma $||\cdot||_M$ não é necessariamente único e pode não ter relação com o ponto fixo de T em relação à norma infinita, i.e. a solução exata do algoritmo PD. O ponto de acumulação $\mathcal{V}^*$ é denominado "*projeção ótima*". Vale ressaltar que o conceito de projeção ótima é uma novidade introduzida neste trabalho. Observe que a projeção ótima pertence à arquitetura de aproximação e pode, em princípio, ser alcançada pelo algoritmo PDRA. Neste trabalho, entende-se que definir a resposta ótima de um algoritmo aproximado como sendo a melhor aproximação da solução real pela arquitetura escolhida é um avanço, já que assim a avalição da resposta de um algoritmo aproximado desvincula-se de conceitos como *qualidade da aproximação*, muito difíceis de serem medidos na prática, já que a função valor do problema é desconhecida à priori.

A definição da resposta ótima carrega, à primeira vista, uma inconsistência, considerando-se que a função valor ótima V^* é, como mencionado anteriormente, desconhecida. Essa aparente inconsistência é contornada pela definição de uma medida de erro apropriada entre uma iteração do algoritmo PDRA com a iteração correspondente do algoritmo PD a partir de uma mesma solução inicial, pertencente à arquitetura de aproximação. Mostra-se que essa medida de erro decresce geometricamente para uma certa classe de operadores de projeção, possibilitando assim a obtenção da projeção ótima pelo algoritmo PDRA a partir de qualquer solução inicial. Visto que a projeção ótima depende da arquitetura de aproximação escolhida, a utilização de uma arquitetura de aproximação adequada é fundamental. Vale ressaltar que essa afirmação aplica-se a qualquer algoritmo de PD com representação aproximada. No entanto, uma discussão sobre a escolha da arquitetura de aproximação a ser utilizada foge do escopo deste trabalho.

(Gordon 1995) demonstrou que o algoritmo (4.6) converge quando o operador de projeção é não expansivo, ou seja

$$||\mathcal{P}_A(x) - \mathcal{P}_A(y)|| \leq ||x - y||$$

para todos x, y pertencentes ao domínio de $\mathcal{P}_A$, sendo $||\cdot||$ a norma infinita. Outros autores exploraram propriedades de projeções não expansivas, por exemplo (Reynolds 2002), para desenvolver algoritmos convergentes aplicáveis a arquiteturas de aproximação específicas. Não obstante, a resposta a esses algoritmos pode ser diferente da *projeção ótima* definida acima. Esse trabalho trata, mais adiante, de projeções expansivas, procurando uma teoria geral aplicável a qualquer arquitetura de aproximação. Mais que isso, define-se um algoritmo que pode convergir para a projeção ótima.

A fim de dar prosseguimento ao trabalho, assume-se que a arquitetura de aproximação é "bem comportada" no sentido da hipótese abaixo.

B_1 Seja S o espaço de estados do PDM a ser solucionado. Qualquer função valor finita $V : S \rightarrow \mathbb{R}$ possui uma projeção finita, $\mathcal{P}_A(V)$, no espaço das funções pertencentes à arquitetura de aproximação A.

Uma propriedade óbvia do operador de projeção é

$$\mathcal{P}_A(Y) = Y, \ \forall Y \in A. \tag{4.7}$$

A próxima seção traz condições suficientes para que o algoritmo (4.6) convirja para a projeção ótima $\mathcal{P}_A(V^*)$. Como mencionado acima, investiga-se as propriedades de operadores de projeção expansivos que apresentem propriedades desejáveis, possibilitando a convergência do algoritmo aproximado.

4.6 Mapeamentos Aproximados Contrativos

Ao invés de definir um mapeamento aproximado (algoritmo PDRA) para depois investigar suas propriedades, assume-se que o mapeamento aproximado utilizado $\hat{T} := \mathcal{P}_A \circ T$ é contrativo, sendo seu ponto fixo a solução do algoritmo PDRA; no decorrer do capítulo, mostra-se como construir um mapeamento aproximado dessa natureza.

O principal resultado desta seção aparece no teorema 3 e traz as condições suficientes para a convergência do algoritmo (4.6) à projeção ótima $\mathcal{P}_A(V^*)$. Considere, antes, o seguinte lema enunciado em (Williams e Baird 1993):

Lema 3. *Seja T um mapeamento contrativo em norma $||\cdot||$. Para toda função valor $V : S \to \mathbb{R}$*

$$||V - V^*|| \leq \frac{||V - TV||}{1 - \alpha}$$

sendo $\alpha \in (0, 1)$ o fator de contração do operador T.

Seja α a taxa de contração do algoritmo do mapeamento exato T e considere que o mapeamento aproximado $\hat{T} = \mathcal{P}_A \circ T$ possui uma taxa de contração γ. Uma simples aplicação do lema 3 ao mapeamento $\hat{T}$ mostra que a região de busca do mapeamento aproximado pode não incluir a projeção ótima $\mathcal{P}_A(V^*)$ se este se contrair a uma taxa menor que α. Portanto, para fazer com que a projeção ótima pertença à região de busca do algoritmo aproximado, considera-se que a taxa de contração do mapeamento $\hat{T}$ é estritamente maior que α.

Seja $||\cdot||_M$ a norma infinita no subconjunto de amostragem $M \subset S,\ M \neq \emptyset$ definido na Seção 4.4. Cabe recordar que o operador de projeção $\mathcal{P}_A$ utiliza apenas estados $s \in M$. No decorrer desta Seção, utiliza-se a seguinte hipótese:

B_2 O operador de PD T é contrativo com respeito à norma $||\cdot||_M$.

B_3 O operador de PDRA $\hat{T} = \mathcal{P}_A \circ T$ é contrativo com respeito à norma $||\cdot||_M$.

A partir da hipótese B_3, um limitante superior de erro pode ser gerado para o algoritmo aproximado por meio da aplicação da propriedade de contração do mapeamento aproximado $\hat{T} = \mathcal{P}_A \circ T$. Note que esse limite superior de erro é similar ao encontrado em (Bertsekas e Tsitsiklis 1996, pg. 333).

Lema 4. *Assuma B_1 e B_3 e seja γ a taxa de contração do operador $\hat{T}$ em norma $||\cdot||_M$. Então*

$$||\mathcal{P}_A(T^k\mathcal{V}_0) - \mathcal{V}_k||_M \leq \sum_{j=1}^{k-1} \gamma^j ||T^{k-j}\mathcal{V}_0 - \mathcal{P}_A(T^{k-j}\mathcal{V}_0)||_M$$

Demonstração.

$$\begin{aligned}
||\mathcal{P}_A(T^k\mathcal{V}_0) - \mathcal{V}_k||_M &= ||\mathcal{P}_A(T^k\mathcal{V}_0) - \mathcal{P}_A(T\mathcal{V}_{k-1})||_M \\
&= ||\mathcal{P}_A \circ T(T^{k-1}\mathcal{V}_0) - \mathcal{P}_A \circ T(\mathcal{V}_{k-1})||_M \\
&\leq \gamma||T^{k-1}\mathcal{V}_0 - \mathcal{V}_{k-1}||_M \\
&= \gamma||T^{k-1}\mathcal{V}_0 - \mathcal{V}_{k-1} - \mathcal{P}_A(T^{k-1}\mathcal{V}_0) + \mathcal{P}_A(T^{k-1}\mathcal{V}_0)||_M \\
&\leq \gamma||\mathcal{P}_A(T^{k-1}\mathcal{V}_0) - \mathcal{V}_{k-1}||_M + \gamma||T^{k-1}\mathcal{V}_0 - \mathcal{P}_A(T^{k-1}\mathcal{V}_0)||_M
\end{aligned}$$

Iterando a expressão acima, obtém-se

$$||\mathcal{P}_A(T^k\mathcal{V}_0) - \mathcal{V}_k||_M \leq \sum_{j=1}^{k-1}\gamma^j||T^{k-j}\mathcal{V}_0 - \mathcal{P}_A(T^{k-j}\mathcal{V}_0)||_M + \gamma^{k-1}||\mathcal{P}_A(T\mathcal{V}_0) - \mathcal{V}_1||_M$$

sendo que o último termo acima se anula. □

O limitante apresentado no Lemma 4 é de natureza geral e implica basicamente que o erro final do algoritmo é limitado pela somatória dos erros acumulados durante a execução do algoritmo aproximado. Seria possível melhorar esse limitante de erro? Sob certas condições, essa resposta é positiva. Tais condições são apresentadas na próxima definição.

Definição 5. *Um operador de projeção $\mathcal{P}_A$ é uma expansão controlada em norma $||\cdot||_M$ com respeito a um mapeamento contrativo T se, para todo x, y no domínio de $\mathcal{P}_A$,*

$$||\mathcal{P}_A(x) - \mathcal{P}_A(y)||_M \geq ||x - y||_M$$

e $\mathcal{P}_A \circ T$ é uma contração com taxa $\gamma \in (\alpha, 1)$.

Obviamente, o limite inferior na taxa de contração γ procede, dado que $\hat{T}$ é a combinação de uma expansão e um mapeamento contrativo e não pode, portanto, contrair a uma taxa maior que o operador T.

Lema 5. *Assuma $B_1 - B_3$ e suponha que $\mathcal{P}_A$ é uma expansão controlada em norma $||\cdot||_M$ com respeito ao operador contrativo T. Então*

$$||\mathcal{P}_A(T^k\mathcal{V}_0) - \mathcal{V}_k||_M \leq \gamma^{k-1}||T\mathcal{V}_0 - \mathcal{V}_1||_M$$

Demonstração. Uma vez que $\mathcal{P}_A$ é uma expansão controlada, temos por definição

$$\begin{aligned} ||\mathcal{P}_A(T^k\mathcal{V}_0) - \mathcal{V}_k||_M &= ||\mathcal{P}_A \circ T(T^{k-1}\mathcal{V}_0) - \mathcal{P}_A \circ T(\mathcal{V}_{k-1})||_M \\ &\leq \gamma||T^{k-1}\mathcal{V}_0 - \mathcal{V}_{k-1}||_M. \end{aligned} \tag{4.8}$$

Além disso

$$||T^{k-1}\mathcal{V}_0 - \mathcal{V}_{k-1}||_M \leq ||\mathcal{P}_A(T^{k-1}\mathcal{V}_0) - \mathcal{P}_A(\mathcal{V}_{k-1})||_M = ||\mathcal{P}_A(T^{k-1}\mathcal{V}_0) - \mathcal{V}_{k-1}||_M.$$

Substituindo a expressão em (4.8), obtém-se

$$||\mathcal{P}_A(T^k\mathcal{V}_0) - \mathcal{V}_k||_M \leq \gamma||\mathcal{P}_A(T^{k-1}\mathcal{V}_0) - \mathcal{V}_{k-1}||_M.$$

Iterando-se a expressão acima, obtém-se

$$||\mathcal{P}_A(T^k\mathcal{V}_0) - \mathcal{V}_k||_M \leq \gamma^{k-1}||T\mathcal{V}_0 - \mathcal{V}_1||_M.$$

□

Teorema 3. *Assuma B_1 e seja $\mathcal{P}_A$ uma expansão controlada com respeito ao operador contrativo T. Então o algoritmo* (4.6) *converge para $\mathcal{P}_A(V^*)$.*

Demonstração. Segue diretamente do lema 5 que

$$\begin{aligned}\limsup_{k\to\infty} ||\mathcal{P}_A(V^*) - \mathcal{V}_k||_M \leq & \limsup_{k\to\infty} ||\mathcal{P}_A(V^*) - \mathcal{P}_A(T^k\mathcal{V}_0)||_M \\ & + \limsup_{k\to\infty} ||\mathcal{P}_A(T^k\mathcal{V}_0) - \mathcal{V}_k||_M \\ & \leq 0 + \sup_{k\to\infty} \gamma^{k-1}||T\mathcal{V}_0 - \mathcal{V}_1||_M = 0\end{aligned}$$

□

Comentário 2. *Quando $M = S$, V^* é o único ponto fixo do operador T em norma infinita. Nesse caso, portanto, o algoritmo PDRA converge para $\mathcal{P}_A(V^*)$, isto é, a projeção da função valor ótima na arquitetura de aproximação.*

4.6.1 Geração de Expansões Controladas

Foram apresentados até aqui resultados interessantes aplicáveis a expansões controladas. Introduz-se, nesta seção, uma maneira simples de gerar uma expansão controlada. O Lema 6 mostra como gerar uma expansão controlada em norma $||\cdot||_M$ por meio da monitoração de iterações sucessivas do algoritmo PDRA.

Lema 6. *Assuma $B_1 - B_3$. Seja θ a taxa de contração do operador T na norma $||\cdot||_M$, e seja $\sigma \in (\theta, 1)$, e suponha que*

$$\theta||\mathcal{V}_k - \mathcal{V}_{k-1}||_M \leq ||\mathcal{V}_{k+1} - \mathcal{V}_k||_M \leq \sigma||\mathcal{V}_k - \mathcal{V}_{k-1}||_M, \tag{4.9}$$

se verifica para todo $k > 0$. Então $\mathcal{P}_A$ é uma expansão controlada com respeito a T.

Demonstração. A primeira desigualdade na equação (4.9) equivale a

$$\begin{aligned}||\mathcal{P}_A(T\mathcal{V}_k) - \mathcal{P}_A(T\mathcal{V}_{k-1})||_M &\geq \theta||\mathcal{V}_k - \mathcal{V}_{k-1}||_M \\ &\geq ||T\mathcal{V}_k - T\mathcal{V}_{k-1}||_M.\end{aligned}$$

Portanto, $\mathcal{P}_A$ é uma expansão. Além disso, a última desigualdade em (4.9) indica que operador $\hat{T} = \mathcal{P}_A \circ T$ é contrativo, com taxa $\theta \leq \sigma \leq 1$. Logo, temos que $\mathcal{P}_A$ satisfaz a Definição 5. □

Conclui-se dos resultados apresentados até aqui que, utilizando-se a desigualdade (4.9) no Lema 6 e monitorando-se iterações sucessivas do algoritmo PDRA, pode-se estabelecer algoritmos convergentes, tal que $\mathcal{P}_A$ seja uma expansão controlada, para qualquer arquitetura de aproximação arbitrária. Esse assunto é detalhado na próxima seção.

Algoritmo 2 Algoritmo PDRA Convergente Baseado em Expansões Controladas

Passo 0 Início

1. Escolha $\sigma \in (\theta, 1)$, $m \geq 1$
2. Estabeleça a tolerância $tol = \delta$
3. Escolha $\mathcal{V}_0 \in A$
4. $y \leftarrow \sigma^{-1}||\mathcal{P}_A(T\mathcal{V}_0) - \mathcal{V}_0||_M$
5. $k \leftarrow 0$

Passo 1 Atualização

1. $k \leftarrow k + 1$
2. $\mathcal{V}_k^{'} \leftarrow \mathcal{P}_A(T\mathcal{V}_{k-1})$
3. $\Delta \leftarrow \mathcal{V}_k^{'} - \mathcal{V}_{k-1}$

Passo 2 Controle de Expansão

1. $\rho = \max_r\{r||\Delta||_M \leq \sigma y\}$
2. $\mathcal{V}_k \leftarrow \mathcal{V}_{k-1} + \rho\Delta$
3. $y \leftarrow \rho||\Delta||_M$

Passo 3 Teste de Convergência
Se($y \leq tol$) Parar
Caso contrário Voltar ao Passo 1

4.6.2 Construindo Um Algoritmo Aproximado Convergente

Nesta seção, abordamos o problema de implementar algoritmos PDRA convergentes para arquiteturas de aproximação arbitrárias, com base nos resultados do Lema 6. O Algoritmo 2 foi idealizado para se ajustar às desigualdades (4.9) a cada iteração. Utilizando qualquer operador de projeção como base, emprega-se, a cada iteração (se necessário), um fator de ajuste na projeção obtida por esse operador, de modo a gerar uma expansão controlada.

Observe que a variável ρ no passo 2 do Algoritmo 2 pode ser interpretada como o máximo passo a ser empregado na direção Δ de modo a se satisfazer a desigualdade (4.9). O operador aproximado resultante é um mapeamento contrativo com taxa de contração σ e satisfaz a Definição 5. Como mencionado na Seção 4.6, o Algoritmo 2 converge para a projeção de um ponto fixo do operador T na norma $||\cdot||_M$.

Note-se que quando não é possível estabelecer a taxa de contração do operador T com respeito à norma de subconjunto $||\cdot||_M$, ainda assim o Algoritmo 2 é convergente para qualquer $\sigma < 1$ escolhido. Nesse caso, o Algoritmo 2 monitora a diferença entre iterações sucessivas de forma a garantir que o mapeamento aproximado resultante é $||\cdot||_M$-contrativo. Entretanto,

com os resultados obtidos neste capítulo, nada se pode inferir a respeito da qualidade da solução aproximada obtida dessa maneira. Algoritmos baseados em operadores de projeção não expansivos, e.g. (Gordon 1995), são casos particulares onde o passo de monitoração não se faz necessário para garantir convergência. A própria estrutura do operador de projeção garante que a diferença entre candidatas a função valor aproximada decresce a cada iteraçao. Para mais detalhes, veja o Apêndice A.

4.7 Considerações Finais e Contextualização

Estabeleceu-se, neste capítiulo, uma classe de algoritmos PDRA convergentes para qualquer arquitetura de aproximação. A convergência do método é obtida por meio da monitoração de iterações sucessivas do algoritmo aproximado, empregando-se um fator de ajuste de passo quando necessário.

Foram introduzidos os conceitos de *expansões controladas* e *projeção ótima*, sendo que este último denota um ponto fixo do operador de PD na norma infinita empregada em um subconjunto de amostragem M. Mostrou-se que a utilização de expansões controladas garante a convergência do algoritmo para a projeção ótima e introduziu-se um algoritmo que, por meio de monitoração de iterações sucessivas e emprego de um fator de ajuste quando necessário, transforma qualquer operador de projeção utilizado como base em uma expansão controlada. Dessa forma, pode-se garantir a convergência do método para a projeção ótima.

Os resultados obtidos possibilitam avanços em dois sentidos. Em primeiro lugar introduz-se um algoritmo convergente para qualquer arquitetura de aproximação arbitrária. Em segundo lugar, por meio do emprego de expansões controladas, a natureza do ponto de acumulação pode ser inferida. Resultados nesse sentido ainda não haviam sido obtidos na literatura.

Introduz-se, no Capítulo 6, um algoritmo PDRA baseado em aproximação polinomial da função valor para o problema P&E estudado. As idéias obtidas nesse algoritmo serão empregadas no sentido de garantir a convergência do algoritmo proposto por meio da monitoração de iterações sucessivas nos moldes do Lema 6 e do Algoritmo 2.

Capítulo 5

Estabilidade Estocástica e Soluções Aproximadas

5.1 Preliminares

Os resultados do Capítulo 3 permitem a definição de procedimentos de identificação de políticas estáveis de fácil implementação mesmo para problemas P&E complexos e de grande porte. Uma maneira de explorar esses resultados consiste em identificar funções de Foster-Lyapunov que definam um conjunto recorrente finito S_F.

Dessa forma, a implementação de uma política de controle pode ser considerada em dois passos:

1. Identificação de ações de controle no conjunto $S \setminus S_F$ que levem o sistema à região de estabilidade S_F em tempo finito, a partir de qualquer estado naquele conjunto;

2. Estabelecimento de ações de controle sub-ótimas no interior da região S_F.

O segundo passo pode, naturalmente, ser implementado iterando-se o operador de programação dinâmica na região S_F e estabelecendo-se valores de contorno aproximados para os estados vizinhos. Dessa forma, é possível evitar o esforço computacional de se calcular a política ótima em todo o espaço de estados S.

Neste capítulo, mostrar-se-á que o estabelecimento de funções de Foster/Lyapunov, assim como a conseqüente implicação de estabilidade estocástica, pode ser utilizado para simplificar o procedimento de obtenção de soluções aproximadas (sub-ótimas) em problemas de grande porte. Ao se mudar o foco do problema, concentrando-se em uma região finita do espaço de estados, pode-se estabelecer procedimentos simples para obter políticas sub-ótimas de controle que sejam naturalmente estáveis para o problema em estudo.

Métodos de programação dinâmica com representação aproximada, introduzidos no Capítulo 4, podem também ser utilizados na obtenção de soluções aproximadas em problemas com espaço de estados demasiadamente grandes, para os quais métodos de programação dinâmica sejam impraticáveis.

5.2 Freqüência de Visitas a Estados do Sistema

Esta seção traz como exemplo numérico um problema P&E de dois produtos. Esse exemplo é apresentado no intuito de ilustrar a existência de uma região de atração, ou região de estabilidade (esse último termo sendo empregado com mais freqüência no decorrer do trabalho) em problemas P&E. Para qualquer política de controle estável, existe uma região de atração. Simula-se o sistema em regime estacionário, empregando as ações de controle ótimas e efetua-se uma contagem do número de visitas a cada estado do sistema. O objetivo é mostrar que alguns estados são muito mais visitados que outros, o que indica uma tendência do sistema de retornar com mais freqüência a esses estados.

Utiliza-se como exemplo um problema híbrido de dois produtos, nos moldes da Seção 2.4.1. Os dados referentes a esse exemplo, doravante denominado Exemplo 1, são apresentados nas tabelas 5.1 e 5.2.

Tab. 5.1: Parâmetros do Exemplo 1.

γ	I_1	I_2	η_1	η_2	β_1	β_2	$L(n)$	$g(z,\bar{z})$
0.15	6	4	5	5	10	20	$10\|n_1\| + n_2^2$	10

Tab. 5.2: Duração dos Estágios no Exemplo 1.

Produto 1						Produto 2			
Estágios						Estágios			
1	2	3	4	5	6	1	2	3	4
0.40	0.50	0.60	0.10	0.15	0.05	0.40	0.60	0.50	0.30

As distribuições de demanda são construídas a partir de um processo clássico de chegada poissoniana. Nesse modelo, assume-se que as chegadas de demanda para cada produto j são independentes e ocorrem em tempo contínuo a uma taxa constante $\delta_j > 0$. Além disso, os pedidos individuais de cada produto formam uma seqüência de variáveis independentes e identicamente distribuídas (*iid*). As taxas de demanda por cada produto, assim como as distribuições dos pedidos individuais, são mostradas na Tabela 5.3.

Tab. 5.3: Distribuições de Demanda Para o Exemplo 1.

Produto 1							Produto 2			
λ_1	p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2	λ_2	p_1^1	p_2^1	p_3^1
0.3	0.0964	0.237	0.3151	0.237	0.0964	0.0182	0.3	0.4167	0.4167	0.1667

Para todo subconjunto de produção S^j, $j = 1, 2$, e todo estágio de produção $i(j) \in \mathcal{I}_j$, obtém-se a distribuição final de demanda para cada uma das duas classes de produtos a partir

da taxa de chegada de demanda e da duração do estágio de produção. A probabilidade de que k pedidos do produto j cheguem em um dado estágio de produção de duração t é dada por $\rho_k^j = e^{-\delta_j t} \cdot \delta_j^k / k!$. E a quantidade máxima de chegadas de demanda (pedidos) em um dado estágio de produção é obtida truncando-se a distribuição de probabilidade ρ_k^j em uma precisão pré-definida. No exemplo em questão, apenas as probabilidades $\rho_k^j > 0.05$ são selecionadas e a distribuição final de chegadas de demanda é obtida por meio de renormalização.

No subconjunto de paralisação S^0, define-se o instante de inspeção como sendo o instante de chegada do primeiro pedido. Assim, a probabilidade de transição em cada estágio é dada por (2.2) e o operador de programação dinâmica toma, em S^0, a forma de (2.11).

A Figura 5.1 refere-se à simulação da política ótima (obtida por meio de PD) para o Exemplo 1. Nela, mostra-se o número de visitas em função do estoque/déficit acumulado. Para cada nível de estoque, somou-se o número de visitas em todos os estágios de produção e o resultado está mostrado graficamente na Figura 5.1. Foram realizadas 1000 (mil) simulações distintas, a partir de estados iniciais distintos e aleatórios e contadas as visitas a cada estado $n \in \mathcal{N}$. Em cada uma delas, foram geradas 10^7 transições aleatórias. A Figura 5.1 traz o número médio de visitas a cada estado $n \in \mathcal{N}$.

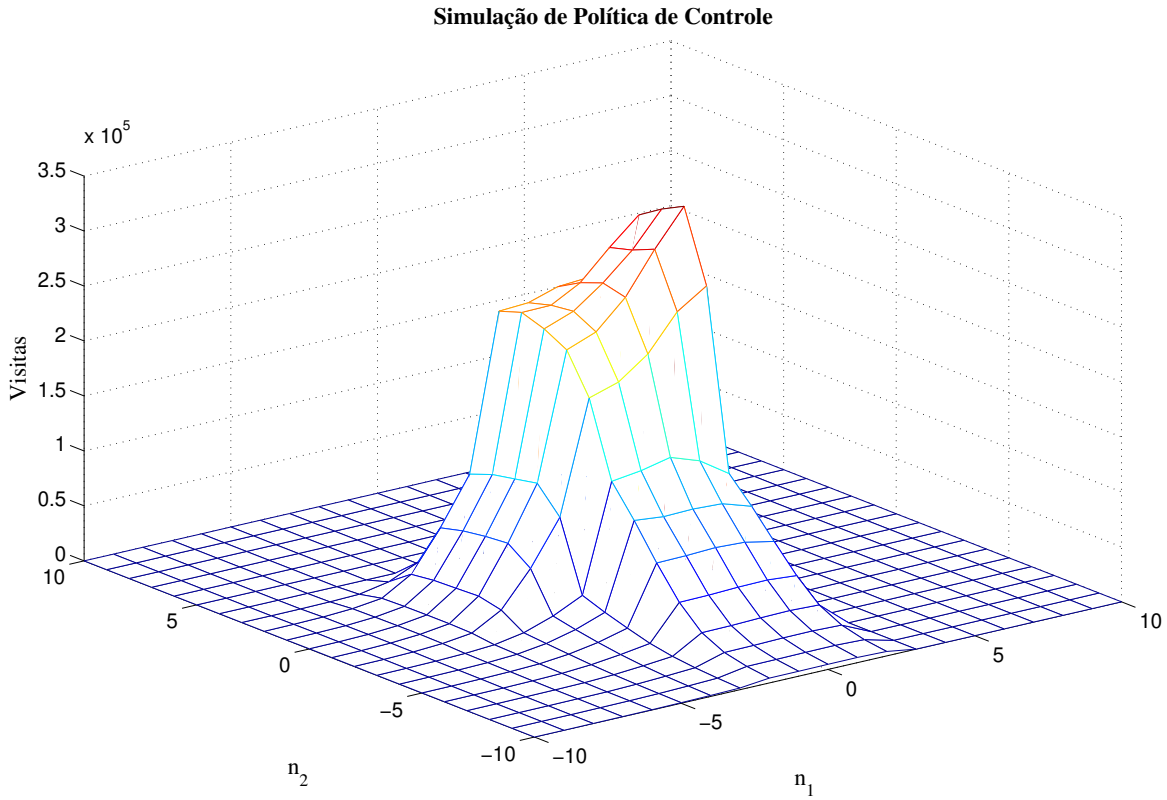

Fig. 5.1: Exemplo 1: Freqüência de Visitas Seguindo a Política Ótima.

Note na Figura 5.1 que, como esperado, as visitas se concentram em uma região finita do espaço de estados. Trata-se de uma região de atração, ou região de estabilidade. Essa observação

condiz com o conceito de estabilidade estocástica definido no capítulo 3 e mostra que, de fato, o nível de estoque é mantido dentro de limites finitos, em horizonte infinito, mediante a aplicação da política ótima de controle. No exemplo apresentado, a região de estabilidade compreende, aproximadamente, pontos no intervalo bidimensional $\mathcal{N}_F := \{-6, \ldots, 3\} \times \{-10, \ldots, 4\}$. Observe ainda que o número de visitas do sistema à região $\mathcal{N} \backslash \mathcal{N}_F$ é desprezível quando comparado ao número de visitas a estados no interior da região $\mathcal{N}_F$. Essa constatação ilustra o conceito de estabilidade estocástica em sistemas sujeitos a incertezas.

5.3 Estratégias Sub-Ótimas

Observando a Figura 5.1, pode-se inferir que a região mais significativa no espaço de estados do problema é a região de estabilidade. Considerando que os estados pertencentes a essa região são muito mais visitados que os demais em horizonte infinito, independentemente do estado inicial, pode-se afirmar que eles são os que mais contribuem para o custo final em horizonte infinito.

Tendo em vista que os estados na região de estabilidade são os que mais influenciam no custo final e ponderando que a influência de estados externos a essa região é limitada, pode-se vislumbrar maneiras de aproximar a influência desses últimos a fim de se estabelecer estratégias sub-ótimas de controle. O objetivo é reduzir o tempo de computação buscando, contudo, estratégias sub-ótimas que apresentem desempenho razoável.

Políticas sub-ótimas estáveis, isto é, políticas que levem o sistema à região de estabilidade em tempo finito a partir de qualquer estado finito, podem possuir desempenho razoável, dependendo das ações de controle prescritas para os estados no interior da região de estabilidade. Argumenta-se que uma boa aproximação na região S_F é fundamental, dado que os estados dessa região são mais determinantes do ponto de vista do custo final. Uma vez que a região de estabilidade é finita, torna-se praticável a possibilidade de se utilizar o operador de PD exato nesta região e aproximações na região contígua, isto é, nos estados vizinhos. Assim, pode-se agregar mais confiabilidade às ações de controle obtidas por um dado algoritmo aproximado para estados pertencentes à região de estabilidade. Com relação aos estados externos a essa região, a estabilidade da política implementada garante retorno à região de estabilidade em tempo finito.

Cabe ressaltar que a região de estabilidade, i.e. os estados mais visitados pelo sistema em horizonte infinito, é estabelecida pela política de controle implementada. Portanto, pode-se esperar que políticas cuja região de estabilidade coincida ou englobe a região de estabilidade da política ótima apresentem um bom desempenho. Conseqüentemente, é razoável empregar-se uma região de estabilidade ampliada na elaboração de algoritmos sub-ótimos para o problema P&E. Essa observação é levada em consideração no algoritmo aproximado introduzido na próxima seção.

5.4 Iteração de Valor com Truncamento

A análise da Seção 3.3 permite a definição de procedimentos interessantes na busca de políticas estáveis, de fácil cômputo mesmo em problemas em grande escala. Uma maneira de

explorar esses resultados consiste em concentrar a atenção em funções candidatas de Foster-Lypapunov que, após verificação, geram conjuntos recorrentes positivos S_F. No interior do conjunto S_F, a aplicação do operador de PD exata leva a uma função valor aproximada que pode ser obtida por meio de iterações em um domínio reduzido: os estados pertencentes à região S_F.

A idéia é estabelecer um *conjunto alvo* S_F e buscar ações de controle sub-ótimas nesse conjunto. Suponha que, uma vez definido o conjunto alvo, exista ao menos uma política heurística que leve o sistema em tempo finito a esse conjunto. Fica claro, portanto, que a definição de uma política heurística dessa natureza é condição suficiente para estabilidade estocástica.

Supondo que o sistema é capaz de suprir a demanda, em média, podem existir várias alternativas de políticas heurísticas para baixos níveis de estoque no conjunto $S \setminus S_F$, ou seja, estados $z \in S \setminus S_F : n < n_B$ que satisfaçam A_3. Para altos níveis de estoque, tendo em vista a existência de demanda expressa na hipótese A_2, introduzida na Seção 2.4, existe uma escolha trivial: paralisar a produção. Uma primeira alternativa para problemas com altos custos de setup (função g no modelo da Seção 2.4) seria utilizar a seguinte regra de decisão:

$$j_{k+1} = \begin{cases} 0, & \text{se } n_k > n_C \\ j, & \text{se } n_k < n_B,\ j_k = j \text{ e } i_k(j) < I_J, \\ j\mathbb{1}_{\{j<J\}} + 1, & \text{se } n_k < n_B,\ j_k = j \text{ e } i_k(j) = I_J. \end{cases}$$

Isto equivale a continuar a produção do produto que está sendo atualmente concluído até o último estágio. Ao se concluir a produção desse produto, prossegue-se a produção do produto seguinte até que esta seja concluída, e assim por diante. Ao se concluir a produção do produto J, recomeça-se o ciclo a partir do produto 1 (um). Esse processo continua até que a região S_F seja alcançada. Para estados cujo nível de estoque esteja acima do máximo nível de estoque da região S_F, a produção é paralisada até que o sistema atinja S_F. Trata-se de uma política que apresenta um número relativamente pequeno de intervenções no conjunto $S \setminus S_F$, ao mesmo tempo em que satisfaz a demanda por ítens. Por essa razão, esta política é atrativa em problemas com altos custos de intervenções (setup).

Apresenta-se ainda uma segunda alternativa de polítca heurística para problemas com baixos custos de intervenção. Essa política usa a seguinte regra de decisão:

$$j_{k+1} = \begin{cases} 0, & \text{se } n_k > n_C \\ j, & \text{se } n_k < n_B, \text{ e } j = \min\{l : l = \arg\min n_k(j)\}. \end{cases} \tag{5.1}$$

Esta regra de decisão corresponde a produzir, na região $\{z \in S \setminus S_F : n < n_B\}$, o produto cujo nível de estoque estiver no nível mais baixo. Em caso de empate, prossegue-se a produção do produto de menor índice. Isto é, o produto 1 (um) tem prioridade sobre o produto 2 (dois) e assim por diante. Neste cenário, pode haver um número significativo de intervenções na região de produção. Por essa razão, essa política é indicada para problemas com baixos custos de intervenção.

Dentro do conjunto S_F, a idéia é 'imitar' a política ótima tanto quanto possível, considerando o conjunto reduzido de informações disponíveis. Quanto melhor for a qualidade das ações de controle no interior do conjunto S_F, menor é a função valor dos estados nesse conjunto. Conjunto este que, em última instância, influencia de maneira mais significativa no custo final do problema.

Assume-se que a função de custo unitário $L : S \to \mathbb{R}$ é monotonicamente não crescente para estados $z_k : n_k \prec 0$, e monotonicamente não decrescente para estados $z_k : n_k \succ 0$. Assim, dado que a política heurística é estável e, portanto, satisfaz A_3 (vide Teorema 2), verifica-se que a expressão

$$E\big[L(n_{k+t_0(n)}, i_{k+t_0(n)}, j_{k+t_0(n)})|z_k = (n,i,j)\big] < L(n,i,j), \text{ para todo } n \notin S_F. \,, \tag{5.2}$$

na qual $t_0(n)$ é a função definida em A_3 e S_F é determinado pela política heurísitica utilizada. Dessa forma, a função de custo unitário $L : S \to \mathbb{R}$ é uma função de Foster/Lyapunov para o problema.

No procedimento proposto, define-se arbitrariamente um conjunto S_F e estabelece-se, no conjunto $S \setminus S_F$, uma política heurística com região de estabilidade S_F. Nessas condições, $L : S \to \mathbb{R}$ é uma função de Foster-Lyapunov para o problema.

Tendo definido a função de Foster/Lyapunov e a política correspondente fora da região S_F, estabelece-se condições de contorno nos estados $z \notin S_F$ a ser utilizada no procedimento de programação dinâmica. No interior da região S_F, aplica-se PD exata, com os elementos da seção 2.2, de forma a encontrar as ações de controle correspondentes para cada estado $z \in S_F$.

Nos experimentos deste trabalho duas alternativas de condição de contorno foram utilizadas, sendo que ambas são funções de Foster/Lyapunov. A primeira condição de contorno é um limite superior da função valor

$$V(z) = \int_0^\infty e^{-\gamma t} L(z) dt = \frac{L(z)}{\gamma}, \; z \notin S_F. \tag{5.3}$$

A segunda condição de contorno proposta é menos conservadora, trata-se do custo de estoque de um período, um limite inferior da função valor

$$V(z) = \begin{cases} L(n,i)E\{\text{duração de } \theta_{j\,i(j)}\}, & \forall z \in S' \\ L(n,i)E\{\text{duração de } \Delta_i\}, & \forall z \in S^0. \end{cases} \tag{5.4}$$

Comentário 3. *Ao se utilizar a condição de contorno* (5.3) *está-se, de fato, resolvendo até a otimalidade um problema P&E 'auxiliar', no qual o processo entra em um estado absorvente ao atingir qualquer estado $z \notin S_F$, pagando-se um custo instantâneo $L(z)$ no restante do horizonte de operação.*

Comentário 4. *Ao se utilizar a condição de contorno* (5.4) *está-se, de fato, resolvendo até a otimalidade um problema P&E 'auxiliar', no qual o processo entra em um estado absorvente ao atingir qualquer estado $z \notin S_F$ com custo final dado pelo custo de estoque de um período.*

5.4.1 Freqüência de Visitas

A fim de ilustrar o método de iteração de valor com truncamento apresentado na Seção 5.4, apresenta-se aqui uma simulação da freqüência de visitas aos estados em S utilizando-se as política de controle obtida pelo método utilizando-se as condições de contorno (5.3) e (5.4), respectivamente. Os experimentos apresentados nesta seção são similares ao experimento apresentado na Seção 5.2.

Em todos os exemplos apresentados nesta seção utilizou-se, nos estados externos a S_F a regra de decisão heurística apresentada na expressão (5.1), sugerida na Seção 5.4.

A Figura 5.2 refere-se ao Exemplo 1, introduzido na Seção 5.2. Na referida simulação, foi utilizada a condição de contorno (5.3).

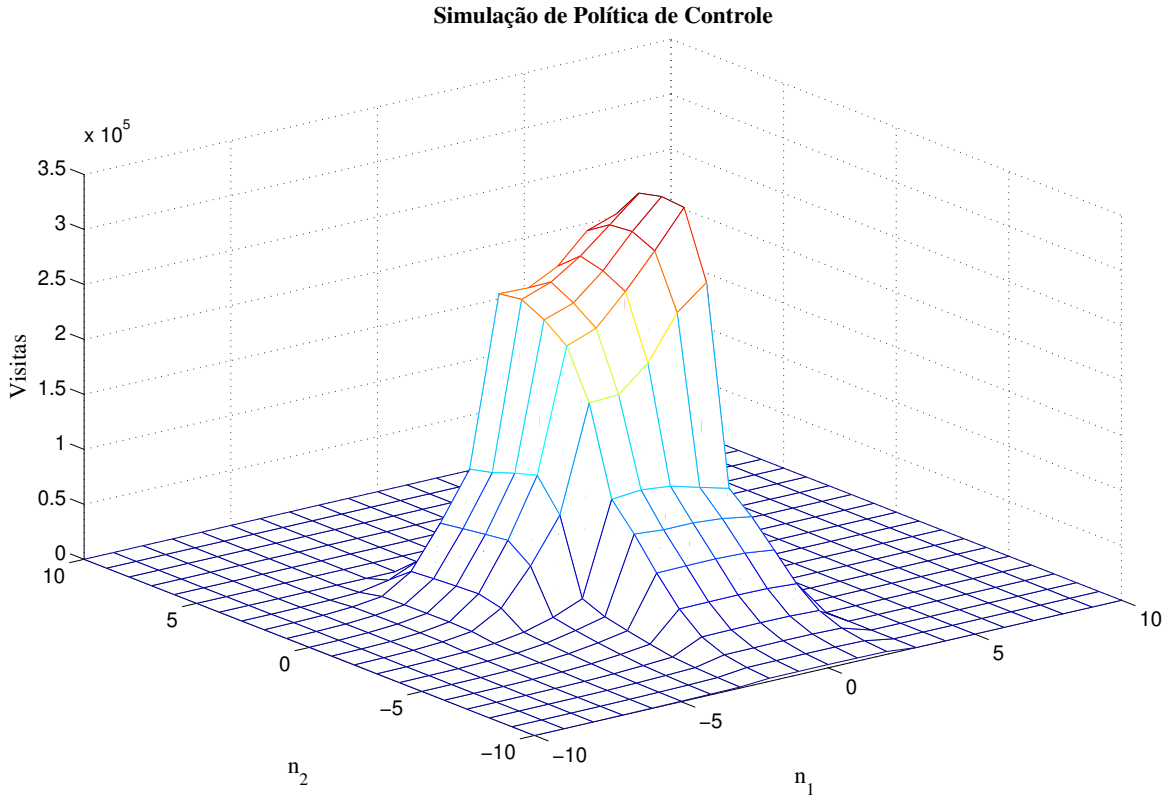

Fig. 5.2: Exemplo 1: Freqüência de Visitas Seguindo Uma Política Sub-Ótima I.

Estabeleceu-se valores arbirários $n_B = -10$ e $n_C = 10$ de modo a definir-se uma região de estabilidade

$$S_F = \mathcal{N}_F \times \mathcal{I} \times \mathcal{J}_0,$$

sendo $\mathcal{N}_F := \prod_{j=1}^{J} \{n_B, \ldots, n_C\}$. As ações de controle no subconjunto S_F são determinadas pelo método de iteração de valor com truncamento. Observe na Figura 5.2 que, como esperado, as visitas aos estados do sistema se concentram na região S_F. Assim, fica empiricamente verificado que a política heurística aplicada fora da região S_F é estabilizante. De fato, as figuras 5.1 e 5.2 são bastante similares e em ambas fica clara a existência de uma região de estabilidade.

A Figura 5.3 mostra a diferença entre o número de visitas utilizando a política ótima e o número de visitas do algoritmo com truncamento e condição de contorno (5.3). Note que a diferença no número de visitas é função do nível de estoque do produto 2 (dois), que possui custo quadrático. De maneira geral, a política sub-ótima apresenta menos visitas a níveis de estoque menores do produto 2 e mais visitas a níveis de estoque maiores do produto 2. Isso

indica que a política sub-ótima é mais conservadora, pois tende a manter os níveis de estoque em patamares maiores que a política ótima. Isso se deve ao fato de utilizarmos um limitante superior da função valor ótima como condição de contorno.

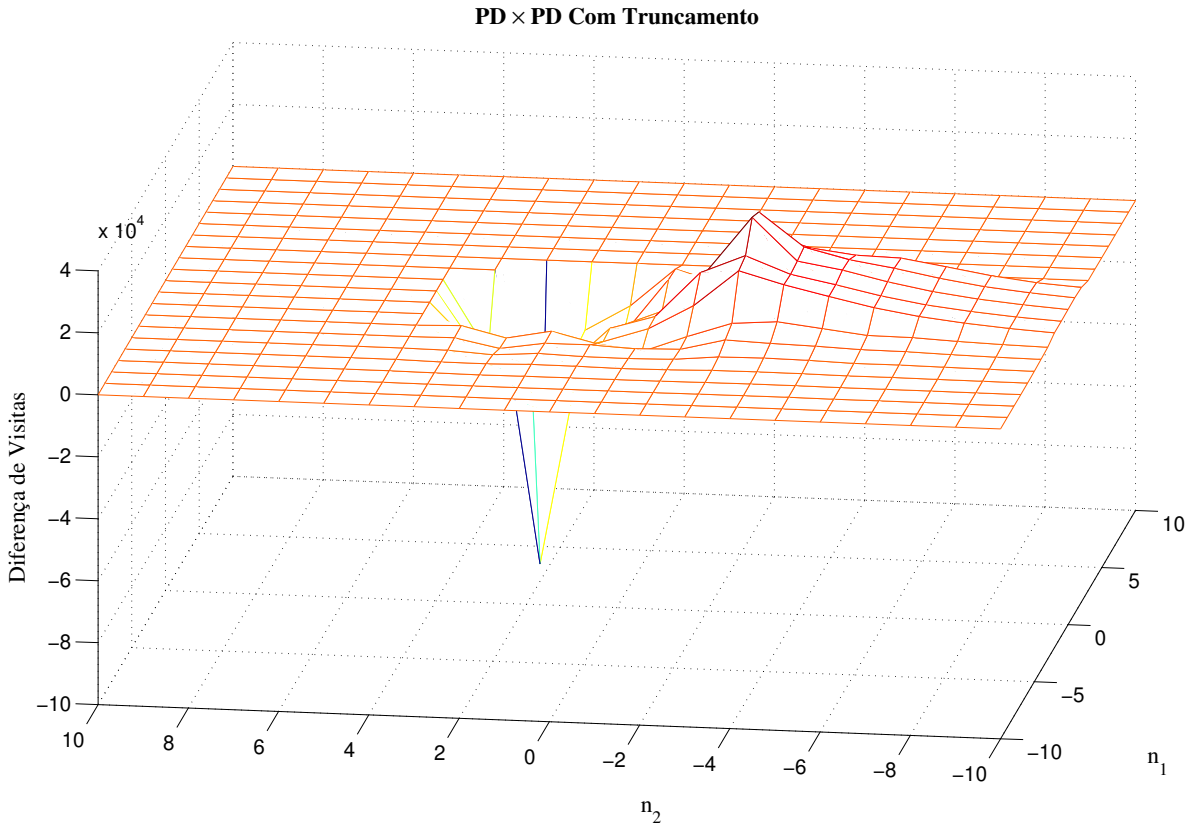

Fig. 5.3: Exemplo 1: Comparação Entre Política Ótima e Sub-Ótima I.

Nos demais experimentos desta seção, foi utilizada a condição de contorno (5.4). O Exemplo 1 foi simulado utilizando-se a política sub-ótima obtida pelo algoritmo de iteração de valor com truncamento. O número médio de visitas é apresentado na Figura 5.4.

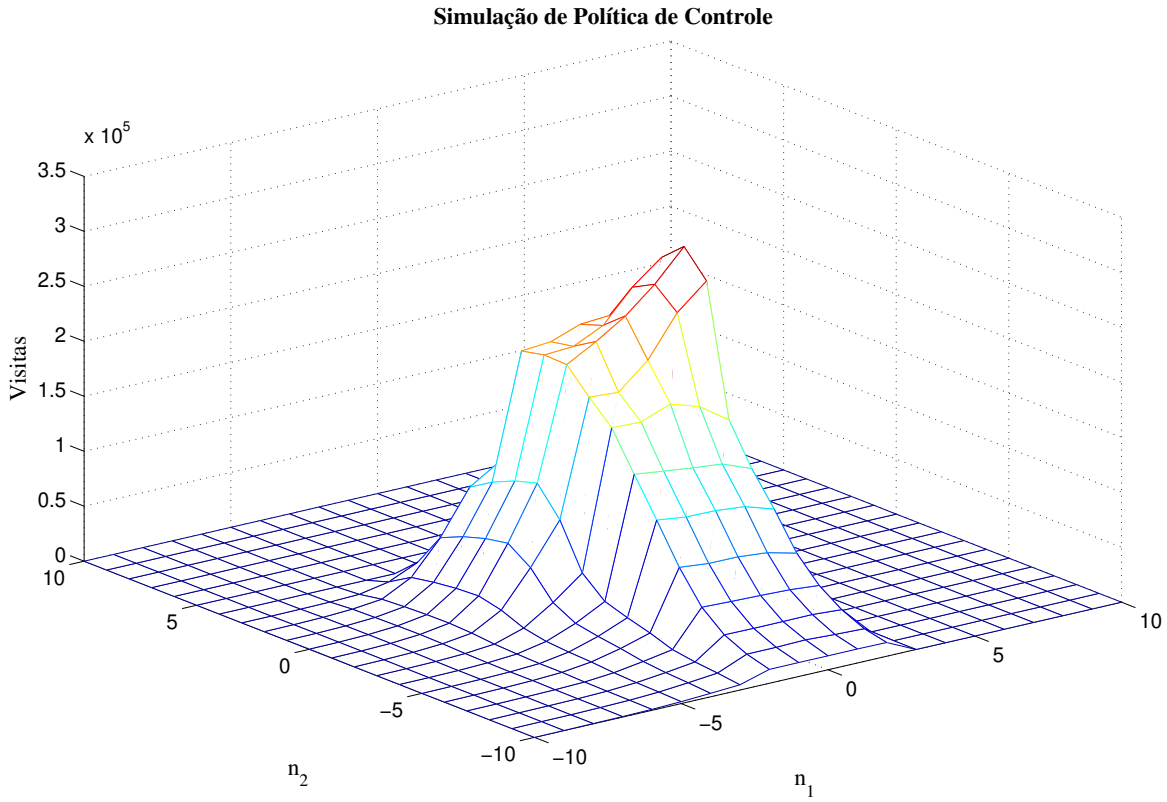

Fig. 5.4: Exemplo 1: Freqüência de Visitas Seguindo Uma Política Sub-Ótima II.

A Figura 5.5 mostra a diferença entre o número de visitas utilizando a política ótima e o número de visitas do algoritmo com truncamento e condição de contorno (5.4). Note que a diferença no número de visitas é função do nível de estoque do produto 2 (dois), que possui custo quadrático. De maneira geral, a política sub-ótima apresenta menos visitas a níveis de estoque maiores do produto 2 e mais visitas a níveis de estoque menores do produto 2. Isso indica que a política sub-ótima é menos conservadora, pois tende a manter os níveis de estoque em patamares menores que a política ótima. Isso se deve ao fato de utilizarmos um limitante inferior da função valor ótima como condição de contorno.

5.4.2 Funções de Custo Sub-Ótimas

Mostra-se, nesta seção, uma comparação entre a função valor ótima do Exemplo 1, utilizado para análise neste capítulo, e as funções valor aproximadas, referentes às políticas obtidas pelos algoritmos com truncamento da Seção 5.4. O objetivo é trazer uma comparação de desempenho entre a política ótima e as políticas sub-ótimas estáveis obtidas pelo algoritmo aproximado proposto.

A Figura 5.6 mostra a diferença entre a função valor ótima e a função valor da política sub-ótima obtida por meio do algoritmo de iteração de valor com truncamento, utilizando a condição de contorno (5.3), no subconjunto S^1. Os resultados nos demais subconjuntos

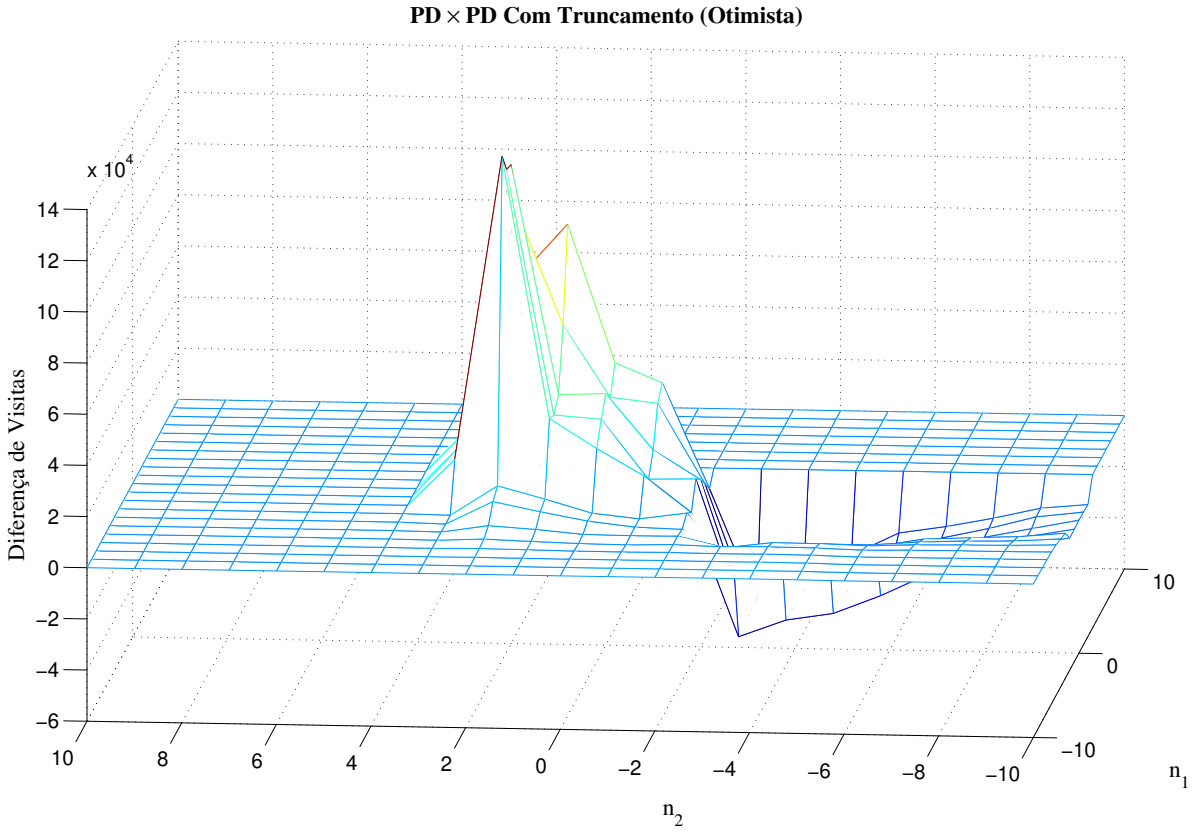

Fig. 5.5: Exemplo 1: Comparação Entre Política Ótima e Sub-Ótima II.

são similares e, por essa razão, omitidos. Note no detalhe que o erro máximo percentual entre as funções valor ótima e sub-ótima é da ordem de 5% na região de estabilidade $S_F = \{-10, \ldots, 10\} \times \{-10, \ldots, 10\} \times \mathcal{I} \times \mathcal{J}$. Esse resultado sugere que o emprego de uma política heurística estável no conjunto $S \setminus S_F$ tem efeito limitado na região de estabilidade S_F. Esse resultado pode ser esperado, uma vez que, como mostra a figura 5.2, a freqüência relativa de visitas utilizando a política sub-ótima é muito mais significativa na região S_F. Como esperado, os erros percentuais no conjunto $S \setminus S_F$, onde emprega-se uma política heurística pura, são mais significativos. A importância relativa desses erros é, no entanto, pequena, considerando que o objetivo do controlador é fazer com que o sistema evite operar nessa região.

A Figura 5.7 mostra a diferença entre a função valor ótima e a função valor da política sub-ótima obtida por meio do algoritmo de iteração de valor com truncamento, utilizando a condição de contorno (5.4). Observe que o erro máximo na região de estabilidade $S_F = \{-10, \ldots, 10\} \times \{-10, \ldots, 10\} \times \mathcal{I} \times \mathcal{J}$ é da ordem de 40%. Cabe ressaltar que, no geral, o desempenho da política sub-ótima se mantém razoável na região de estabilidade. Note ainda que os erros percentuais são mais elevados se comparados aos obtidos no experimento anterior, o que sugere que, no exemplo em questão, condições de contorno mais conservadoras apresentam melhor desempenho.

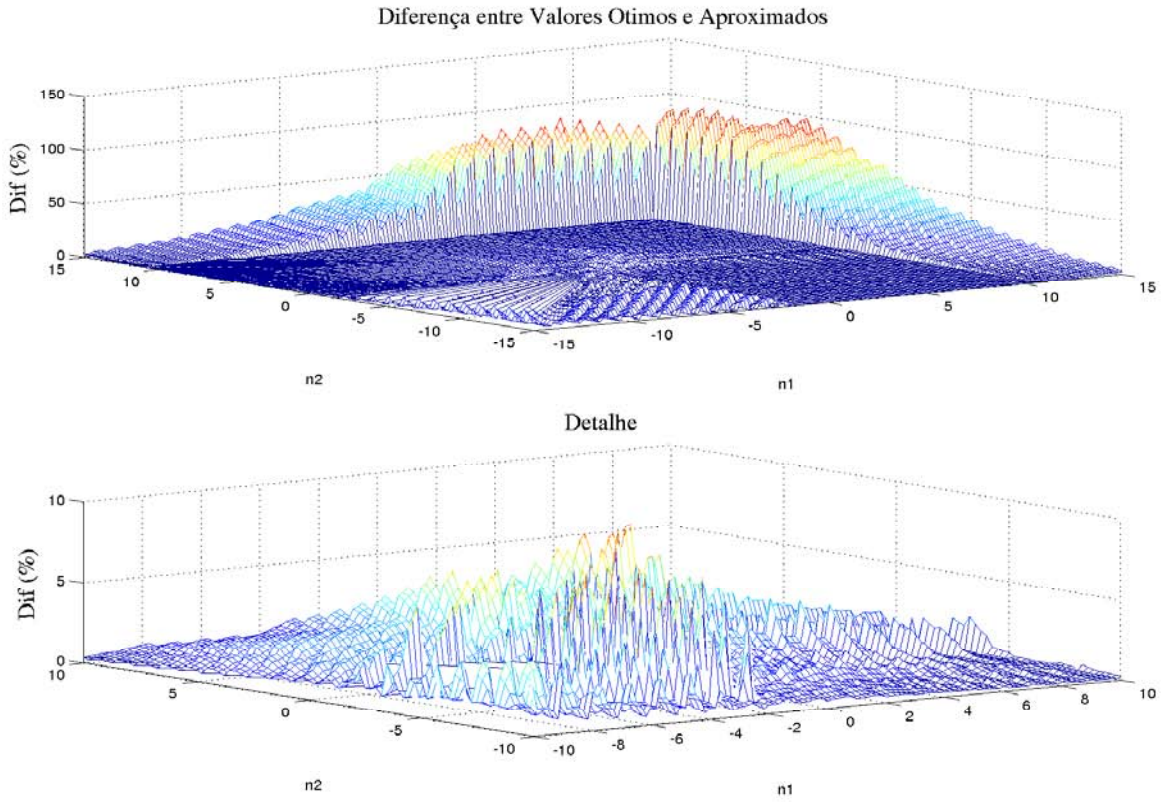

Fig. 5.6: Exemplo 1: Função Valor Ótima × Função Valor Sub-Ótima (em S^1) I.

5.5 Considerações Finais e Contextualização

Apresentou-se neste capítulo um algoritmo sub-ótimo de iteração de valor com base no estabelecimento de uma região de estabilidade arbitrária e implementação de uma política heurística no subconjunto complementar à região de estabilidade. A política heurística deve ser definida de modo a trazer o sistema para a região de estabilidade, a partir de qualquer estado inicial, em tempo finito. Esse algoritmo possibilita a iteração do operador de programação dinâmica em um subconjunto finito do espaço de estados, a região de estabilidade. Dessa forma, é possível obter soluções aproximadas para problemas de grande porte, o que não seria possível utilizando programação dinâmica exata, devido ao conhecido *mal da dimensionalidade*.

Introduziu-se um exemplo numérico e foi mostrado que a freqüência relativa de visitas aos estados, assim como a função valor da política sub-ótima, se aproximam bastante da freqüência relativa e da função valor da política ótima, respectivamente. Os resultados sugerem que uma condição de contorno mais conservadora apresenta resultados melhores.

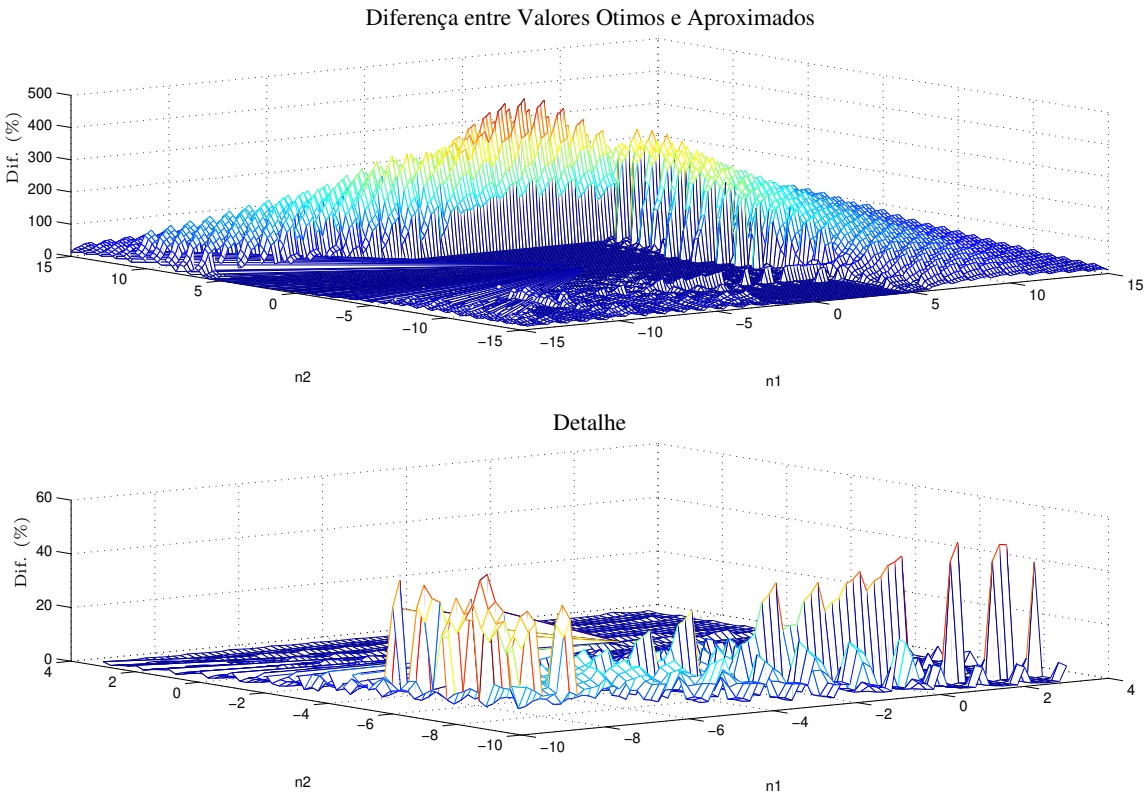

Fig. 5.7: Exemplo 1: Função Valor Ótima × Função Valor Sub-Ótima (em S^1) II.

Capítulo 6

Experimentos Numéricos

6.1 Preliminares

Apresenta-se, neste capítulo, um panorama dos resultados numéricos obtidos para um conjunto de problemas-teste, utilizando-se os métodos de programação dinâmica exata e o algoritmo PDRA com uma arquitetura de aproximação sugerida na Seção 6.2.

Efetua-se uma análise dos resultados a partir de variações dos parâmetros do algoritmo aproximado, assim como comparações entre os resultados obtidos pelos algoritmos exato e aproximado.

Introduz-se, na Seção 6.3, os exemplos numéricos utilizados nas análises contidas neste capítulo. Prossegue-se com experimentos numéricos destinados a avaliar as propriedades do algoritmo aproximado.

6.2 Métodos PDRA Para Problemas P&E

O primeiro passo na definição de um algoritmo PDRA para o problema P&E é o estabelecimento de uma arquitetura de aproximação arbitrária. Opta-se, neste trabalho, por utilizar uma aproximação polinomial em função da variável de estoque/déficit. De acordo com a terminologia utilizada na área, esse tipo arquitetura de aproximação pode ser vista como uma arquitetura linear. Assim, a função valor aproximada pode ser vista como uma combinação linear de funções-base polinomiais. Cada função base é um polinômio na variável de estoque/déficit n. Os parâmetros da arquitetura são os coeficientes dos polinômios utilizados. Resultados preliminares utilizando esse tipo de aproximação em problemas P&E de um item foram relatados em (Arruda, do Val e Almudevar 2005).

A maneira encontrada para aproximar a função valor V em função apenas da variável de estoque n é a de se estabelecer aproximações polinomiais distintas para cada estágio de produção $i \in \mathcal{I}$. Embora os coeficientes associados a cada polinômio possam diferir para cada estágio $i \in \mathcal{I}$, a arquitetura de aproximação, i.e. os polinômios-base, permanece constante. Uma vez que a região de interesse é a região de estabilidade induzida pela política de controle, emprega-se uma representação exata para os estados no interior do conjunto

$$S_F = \mathcal{N}_F \times \mathcal{I} \times \mathcal{J}_0, \tag{6.1}$$

sendo que $\mathcal{N}_F := \prod_{j=1}^{J}\{n_B, \ldots, n_C\}$ e n_B e n_C são valores arbitrários. Nos estados externos à região S_F emprega-se a aproximação polinomial descrita acima. Como mencionado anteriormente, define-se à priori uma aproximação polinomial para cada estágio de produção $i \in \mathcal{I}$. Obviamente, a arquitetura de aproximação permanece constante durante uma execução completa do algoritmo.

Para todo subconjunto $S^j,\ j \in \mathcal{J}_0$ e estágio de produção $i \in \mathcal{I}$, a função valor aproximada é dada por uma soma de polinômios de ordem ρ_j em função dos níveis de estoque de cada produto $j \in \mathcal{J}$. Se $n \in \mathcal{N}_F$, utiliza-se representação exata, caso contrário, utiliza-se a função valor aproximada descrita acima. Pode-se, portanto, escrever:

$$\mathcal{V}(n,i,j) = \begin{cases} w_1' P(n_1,\rho_1) + \ldots + w_J' P(n_n,\rho_J) + w_{J+1}', & z \notin S_F \\ \\ V(n,i), & s \in S_F \end{cases} \qquad z \in S^j. \tag{6.2}$$

sendo $w_j = [w_{j\rho_j}\ w_{j(\rho_j-1)}\ w_{j1}]',\ j \leq J$; w_{J+1} um escalar finito e $P(n_j,\rho) = [n_j^{\rho}\ n_j^{\rho-1}\ \ldots\ n_j^1]$. Vale ressaltar que os vetores de peso $w_j, j \in \mathcal{J}$ são diferentes em cada subconjunto S^j.

Cabe agora definir o operador de projeção $\mathcal{P}_A$ a ser utilizado a fim de se calcular os coeficientes da equação (6.2). Utiliza-se como medida de erro quadrático médio e calcula-se, a cada iteração, os coeficientes w_j que minimizam o erro quadrático médio entre os custos obtidos pelo operador de PD T e a aproximação correspondente em um subconjunto finito de S, denominado conjunto de amostragem. Assim, o operador de projeção $\mathcal{P}_A$ é o operador de minimização do erro quadrático médio entre o conjunto amostra e os elementos da arquitetura de aproximação. Por simplicidade, opta-se pela utilização de um conjunto de amostragem constante durante a execução do algoritmo aproximado. A minimização de erros quadráticos é realizada por meio de uma sub-rotina de decomposição por valores singulares. Vale ressaltar que o uso de normas euclidianas ponderadas é perfeitamente compatível com a referida sub-rotina. Para mais informações sobre o decomposição por valores singulares, recomenda-se consultar (Golub e van Loan 1996).

Para cada estágio $i \in \mathcal{I}$ e subconjunto S^j, o subconjunto de amostragem é um subconjunto de $S_A := \mathcal{N}_A \times i \times j$, doravante denominado S_A. A fim de se estabelecer $\mathcal{N}_A$, define-se previamente um limite máximo de estados para esse conjunto, denotado por ss. São então selecionados ss estados igualmente espaçados em $\mathcal{N}$ para compor $\mathcal{N}_A$, sendo excluídos todos os elementos que pertençam a $\mathcal{N}_F$. Um pseudo-código para o procedimento PDRA proposto é apresentado no Algoritmo 3.

6.2.1 Notas Sobre o Algoritmo PDRA

Vale ressaltar que o algoritmo PDRA apresentado na Seção anterior não garante, por si só, estabilidade da política de controle sub-ótima, como é o caso do algoritmo de iteração de valor com truncamento introduzido na Seção 5.4. Isso se deve ao fato de que as ações de controle são obtidas diretamente da função valor aproximada, de acordo com a Equação (4.5).

Algoritmo 3 Algoritmo PDRA

Passo 1 Estabelecer valores w_j iniciais para todo $i \in \mathcal{I}$ e $\mathcal{V}^0(n,i) = V^0(n,i)$ para todo $z \in S_F$

Passo 2 Calcular $T\mathcal{V}_k(z)$, para todo $z \in S_F$ e todo $z \in S_A$.

Passo 3 **Para** todo $i \in \mathcal{I}$ e todo subconjunto S^J

- $\mathcal{V}^k(n,i,j) = T\mathcal{V}(n,i,j),\ \forall z \in S_F$
- Determinar os pesos w_j, por meio de minimização (ponderada) de erros quadráticos nos pontos de amostragem $z \in S_A$.
- Fazer

$$\mathcal{V}^k(n,i,j) = \begin{cases} w'_1 P(n_1,\rho_1) + \ldots + w'_J P(n_n,\rho_J) + w'_{J+1}, & z \notin \mathcal{S}_F \\ V(n,i), & z \in \mathcal{S}_F \end{cases}$$

Fim Para

Passo 4 **Se** Critério de Parada

- Estabelecer ações de controle
- *PARAR*

Caso Contrário Retornar ao Passo 2

Fim Se

Este problema pode ser contornado por meio da implementação de uma política heurística que confere estabilidade ao sistema, no subconjunto complementar a uma região de estabilidade arbitrária, como sugerido na Seção 5.4. Na região de estabilidade arbitrada, pode-se utilizar as ações de controle obtidas por meio da aplicação da expressão (4.5). Dessa forma, obtém-se uma política sub-ótima estável, ao mesmo tempo em que as ações de controle da região de estabilidade são determinadas pelo algoritmo PDRA.

6.3 Dados dos Exemplos Numéricos

Os exemplos apresentados são problemas híbridos de dois produtos, com decisões a tempo contínuo em S^0, nos moldes da Seção 2.4.1. Os dados de custo referentes aos problemas estudados são apresentados na Tabela 6.1. Já as durações dos estágios de produção em S' são apresentados nas Tabelas 6.2 e 6.3.

Como mencionado na Seção 5.2, as distribuições de demanda são construídas a partir de um processo clássico de chegada poissoniana. Assume-se que as chegadas de demanda para cada produto j são independentes e ocorrem em tempo contínuo, a uma taxa constante $\delta_j > 0$. Os pedidos individuais de cada produto formam uma seqüência de variáveis independentes

Tab. 6.1: Parâmetros dos Exemplos Numéricos.

Exemplo	γ	I_1	I_2	η_1	η_2	β_1	β_2	$L(n)$	$g(z,\bar{z})$
Caso A	0.15	6	4	5	5	10	20	$10\|n_1\| + n_2^2$	10
Caso B	0.15	6	4	5	5	10	20	$10\|n_1\| + 10\|n_2\|$	10
Caso C	0.15	6	4	5	5	10	20	$n_1^2 + n_2^2 + 5\|n_2\|$	2.5
Caso D	0.25	5	5	1	2	5	8	$n_1^2 + n_2^2$	10
Caso E	0.25	5	5	1	2	5	8	$n_1^2 + n_2^2 + 5\|n_2\|$	2.5
Caso F	0.25	5	5	1	2	5	8	$n_1^2 + 5\|n_1\| + n_2^2 + 5\|n_2\|$	10

Tab. 6.2: Duração dos Estágios nos Exemplos Numéricos.

Exemplo	**Produto 1**						**Produto 2**			
	Estágios						Estágios			
	1	2	3	4	5	6	1	2	3	4
Caso A	0.40	0.50	0.60	0.10	0.15	0.05	0.40	0.60	0.50	0.30
Caso B	0.40	0.50	0.60	0.10	0.15	0.05	0.40	0.60	0.50	0.30
Caso C	0.40	0.50	0.60	0.10	0.15	0.05	0.40	0.60	0.50	0.30

Tab. 6.3: Duração dos Estágios nos Exemplos Numéricos.

Exemplo	**Produto 1**					**Produto 2**				
	Estágios					Estágios				
	1	2	3	4	5	1	2	3	4	5
Caso D	0.20	0.25	0.35	0.30	0.40	0.35	0.30	0.33	0.27	0.25
Caso E	0.20	0.25	0.35	0.30	0.40	0.35	0.30	0.33	0.27	0.25
Caso F	0.20	0.25	0.35	0.30	0.40	0.35	0.30	0.33	0.27	0.25

e identicamente distribuídas (*iid*). As taxas de demanda por cada produto, assim como as distribuições dos pedidos individuais, são mostradas nas Tabelas 6.4 e 6.5.

Vale relembrar da Seção 5.2 que, para todo subconjunto de produção S^j, $j = 1, 2$, e todo estágio de produção $i(j) \in \mathcal{I}_j$, obtém-se a distribuição final de demanda para cada classe de produtos a partir da taxa de chegada de demanda e da duração do estágio de produção. O processo de obtenção das probabilidades de transição é aquele introduzido na Seção 5.2, com precisão $\rho_k^j > 0.05$.

No subconjunto de paralisação S^0, define-se o instante de inspeção como sendo o instante de chegada do primeiro pedido. Assim, a medida de transição em cada estágio de produção é dada por (2.2) e o operador de programação dinâmica toma, em S^0, a forma de (2.11).

Tab. 6.4: Distribuições de Demanda Para os Exemplos Numéricos.

Exemplo	Produto 1			
	λ_1	p_1^1	p_2^1	p_3^1
Caso A	0.3	0.4167	0.4167	0.1667
Caso B	0.3	0.4167	0.4167	0.1667
Caso C	0.3	0.4167	0.4167	0.1667

Exemplo	Produto 2						
	λ_2	p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2
Caso A	0.3	0.0964	0.237	0.3151	0.237	0.0964	0.0182
Caso B	0.3	0.0964	0.237	0.3151	0.237	0.0964	0.0182
Caso C	0.3	0.0964	0.237	0.3151	0.237	0.0964	0.0182

Tab. 6.5: Distribuições de Demanda Para os Exemplos Numéricos.

Exemplo	**Produto 1**				
	λ_1	p_1^1	p_2^1	p_3^1	p_4^1
Caso D	0.3	0.265625	0.390625	0.265625	0.078125
Caso E	0.3	0.265625	0.390625	0.265625	0.078125
Caso F	0.3	0.265625	0.390625	0.265625	0.078125

Exemplo	**Produto 2**						
	λ_2	p_1^2	p_2^2	p_3^2	p_4^2	p_5^2	p_6^2
Caso D	0.3	0.096354	0.236979	0.315104	0.236979	0.096354	0.018229
Caso E	0.3	0.096354	0.236979	0.315104	0.236979	0.096354	0.018229
Caso F	0.3	0.096354	0.236979	0.315104	0.236979	0.096354	0.018229

6.4 Políticas Ótimas

As Figuras 6.1-6.2 mostram as políticas de controle ótimas para os Casos **A** e **B** da Seção 6.3, respectivamente. Esses exemplos são apresentados no intuito de se mostrar a estrutura básica da política ótima em problemas P&E. Os eixos horizontais representam os níveis de estoque de cada um dos produtos e o eixo vertical traz as ações de controle ótima. A fim de poder representar graficamente as ações de controle para cada estado do sistema, cada nível de estoque de um determinado produto j é discretizado em I_J pontos, de acordo com o estágio de produção do produto j. Por exemplo, se o produto j tem dois estágios de produção, o nível de estoque n_j será representado pelos pontos n_j e $n_j + 0.5$. O primeiro ponto corresponde ao primeiro estágio de produção e o segundo ponto representa o segundo estágio de produção. Dessa forma, cada par (n_j, i_j) está representado pelo ponto $n_j + \frac{i_j - 1}{I_j}$.

Para cada exemplo, três figuras são apresentadas, cada qual indicando as ações de controle em um subconjunto $S^j,\ j \in \{0, 1, 2\}$ do sistema. No subconjunto S^j, deve-se implementar as ações de controle u_j. Vale lembrar que o sistema se encontra paralisado em S^0 e produzindo

o produto j em S^j, $j \in \{1, 2\}$. Nas Figuras 6.1-6.2 abaixo, $u_j = 0$ indica que o sistema deve ser paralisado. Logo, se $u_j = 0$ e $j \neq 0$, uma intervenção deve ser aplicada ao sistema, transferindo-o de S^j para S^0. Analogamente, $u_j = 1$ indica que o sistema deve produzir o produto 1 e $u_j = 2$ indica que o sistema deve produzir o produto 2; naturalmente, intervenções devem ser aplicadas sempre que $u_j \neq j,\ j \neq 0$.

Observe nas Figuras 6.1-6.2 que todo subconjunto (S^1, S^2, S^0) possui duas regiões distintas:

- Uma *região de intervenção*, na qual a política ótima é transferir o sistema para um dos demais subconjuntos.
- Uma *região de não intervenção*, na qual a política ótima é não intervir no sistema

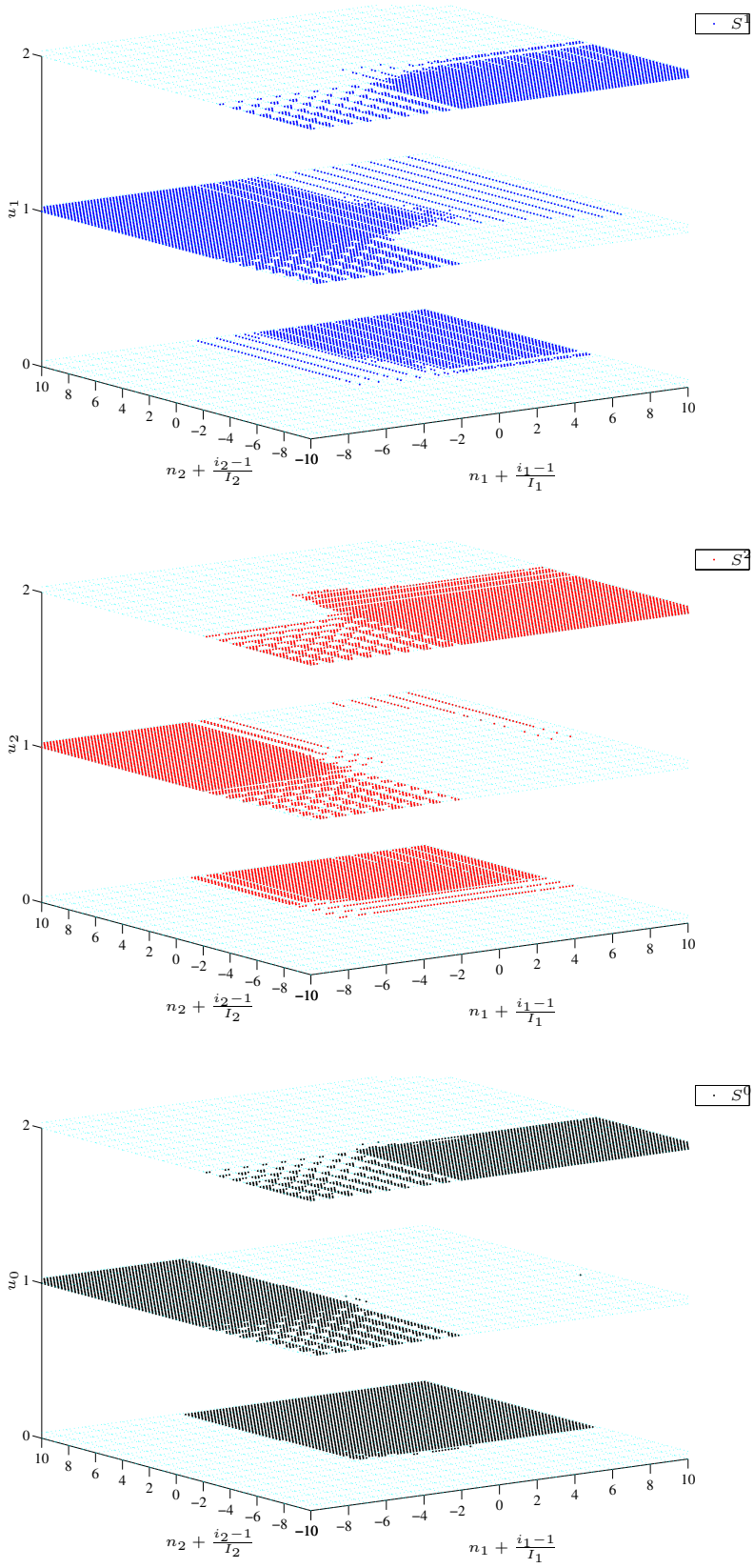

Fig. 6.1: Política Ótima Para o Caso A.

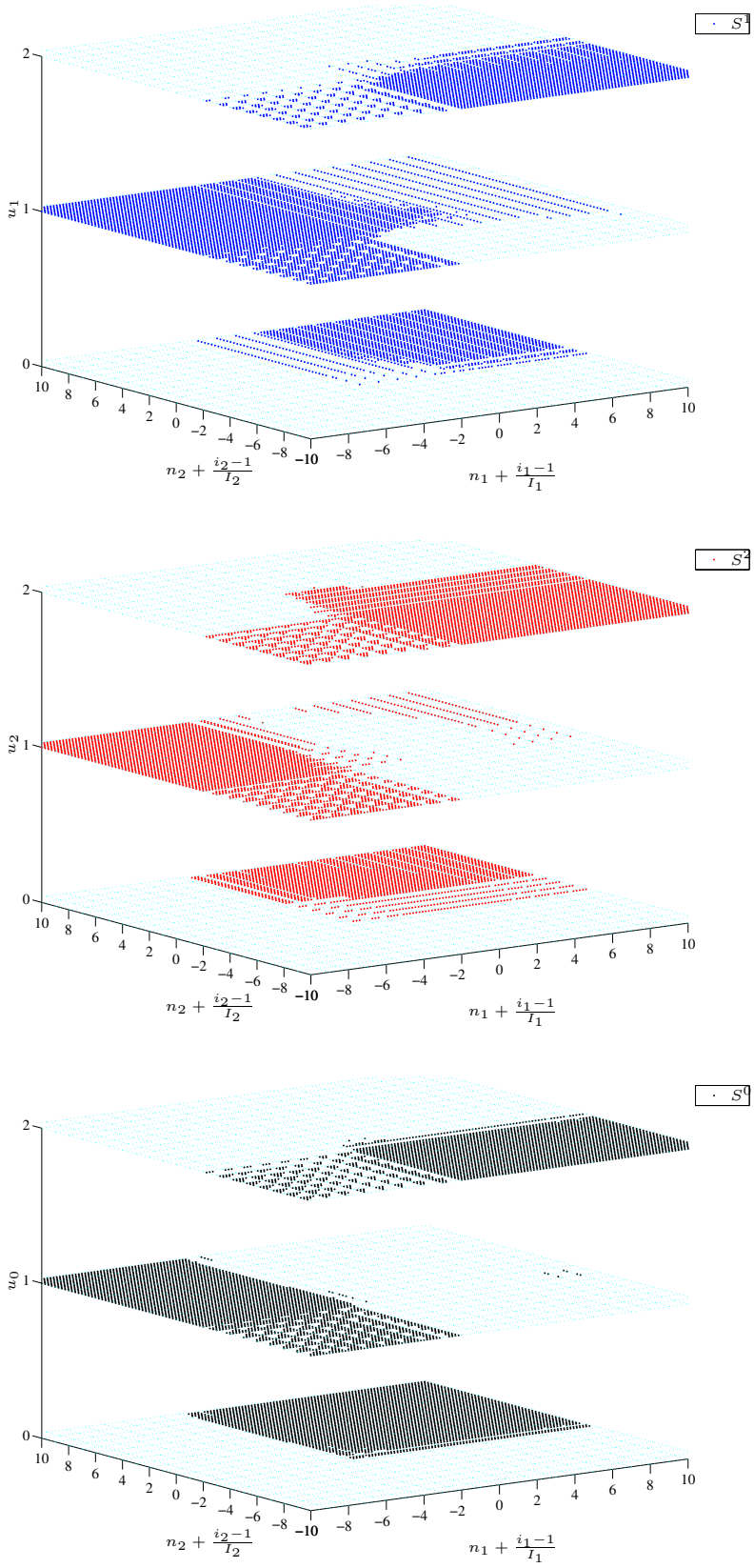

Fig. 6.2: Política Ótima Para o Caso B.

Observe também que podem existir intersecções entre as regiões de não intervenção de mais de um subconjuntos S^j. Isso significa que podem existir pares (n, i) para os quais a política ótima é de não intervenção em mais de um subconjunto. Isso acontece devido à existência de um custo de intervenção. Ora, podem existir situações em que a soma do custo final de um estado destino $z' = (n, i, j')$ de menor função valor com o custo de intervenção $g(z, z')$ supere a função valor no estado de partida $z = (n, i, j)$. Nesse caso, a política ótima prescreveria não intervenção tanto em S^j quanto em $S^{j'}$ para o par (n, i).

A Tabela 6.6 mostra os tempos de execução do algoritmo de programação dinâmica para os Casos A-F. Utilizou-se $N_j^- = -146$ e $N_j^+ = 54$ para todo $j \in \mathcal{J}$. Todas as simulação relatadas neste capítulo foram realizadas em um computador com processador Pentium 4, 2.8 GHz e 768 MB-RAM.

Tab. 6.6: Tempo de Execução do Algoritmo PD

Exemplo	**Tempo (s)**
Caso A	1347
Caso B	1459
Caso C	1470
Caso D	1364
Caso E	1364
Caso F	1441

6.5 Programação Dinâmica Com Representação Aproximada

Mostra-se, nesta seção, os resultados obtidos para os exemplos da Seção 6.3, utilizando-se os métodos aproximados da Seção 6.2 - Algoritmo 3. A Figura 6.3 contém a função valor ótima e a função aproximada obtida pelo algoritmo PDRA sem ponderação, para o subconjunto S^1, no Caso **A**. Os resultados nos demais subconjuntos (S^2 e S^0) são similares e foram omitidos aqui. Os resultados na Figura 6.3 foram obtidos utilizando-se os parâmetros $ss = 100$ (sample size), $n_B = -10$ e $n_C = 10$ e uma aproximação polinomial de ordem 2, isto é $\rho_1 = \rho_2 = 2$ na equação (6.2).

A Tabela 6.7 contém os erros quadráticos (%) médios, erros máximos (%), tempo de execução do algoritmo PDRA, ordem da aproximação polinomial (nos exemplos apresentados, utiliza-se sempre o mesmo valor para os graus polinomiais ρ_1 e ρ_2 na equação (6.2). Os erros quadráticos médios e erros máximos são calculados na região S_F induzida pelos valores n_B e n_C e expressa em (6.1). Esses erros são calculados com relação à função valor ótima obtida por meio de programação dinâmica.

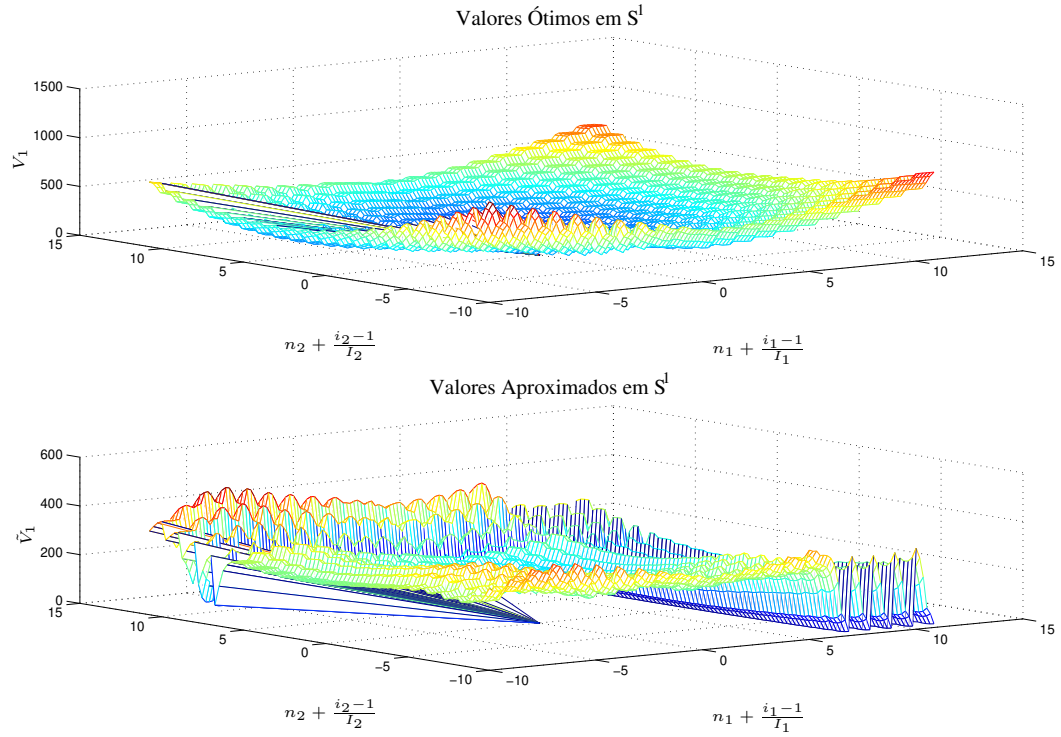

Fig. 6.3: Função Valor e Aproximações Para o Caso A - S^1

Tab. 6.7: Resultados Para o Algoritmo Sem Ponderação

Exemplo	**Ordem do Pol.**	**Er. Quad. Médio(%)**	**Er. Max. (%)**	**Tempo (s)**
Caso A	2	1,0142	99,07	25
	3	0,5736	99,04	42
	4	0,5119	99,03	62
Caso B	2	1,3185	219,10	25
	3	0.3735	114,96	38
	4	0,3592	113,82	58
Caso C	2	1,0723	98,97	29
	3	1,0722	98,97	41
	4	1,0723	98,97	63
Caso D	2	0,7090	89,90	28
	3	0,7089	89,89	44
	4	0,7089	89,89	60
Caso E	2	0,6633	89,81	38
	3	0,6672	89,81	56
	4	0,6672	89,81	73
Caso F	2	0,6115	89,83	46
	3	0,6966	90,41	76
	4	0,6992	90,41	100

Note na Tabela 6.7 que ocorre, em geral, uma melhora de desempenho à medida em que se aumenta o grau do polinômio de aproximação. Em outros casos, um aumento do grau do polinômio não implica em melhoria do desempenho. Observe também que polinômios de ordem maior requerem, como se poderia esperar, maior tempo de execução. Comparando-se a última coluna da Tabela 6.7 com a coluna correspondente na Tabela 6.6, verifica-se que o algoritmo aproximado é significativamente mais rápido que o exato.

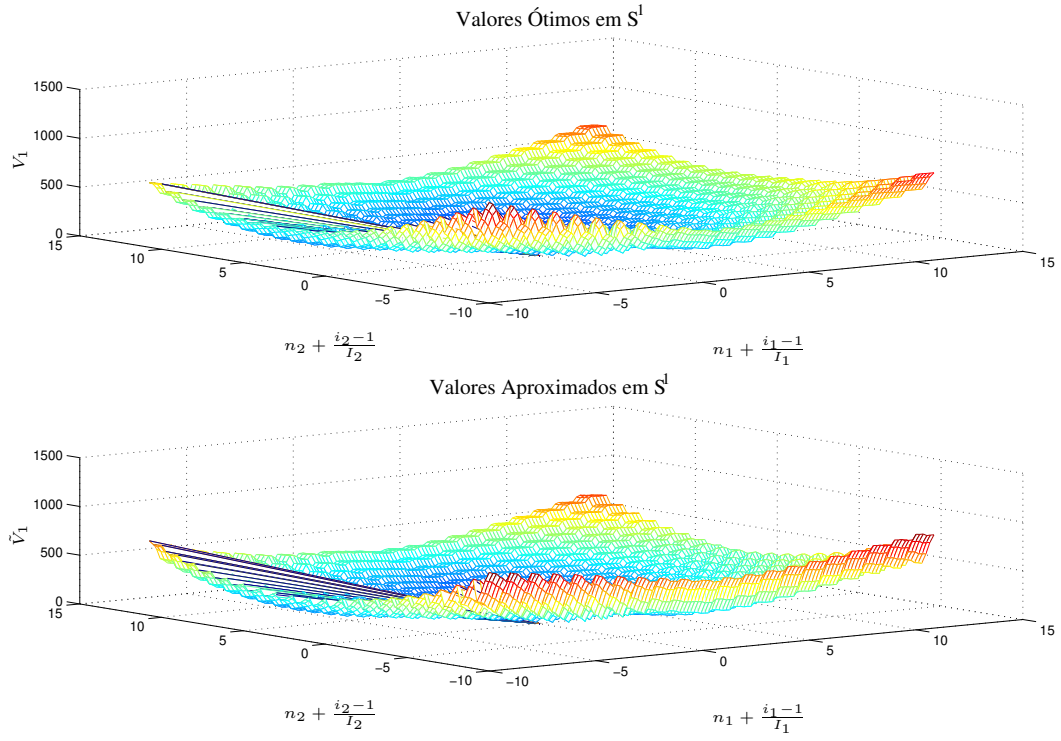

Fig. 6.4: Algoritmo Ponderado: Função Valor e Aproximações Para o Caso A - S^1

Embora os erros quadráticos médios relatados na Tabela 6.7 sejam satisfatórios, verifica-se que o erro máximo apresenta valores bem altos. Isso se deve ao fato de que valores maiores de custo são mais significativos do ponto de vista de erro quadrático. Assim, a minimização de erros quadráticos tende a aproximar melhor funções valor de magnitude maior, o que causa um erro percentual acentuada nas regiões de baixo nível de estoque, como a região arbitrária S_F. A fim de melhorar o desempenho em S_F, introduz-se um algoritmo aproximado que utiliza uma norma ponderada, destinada a atribuir maior peso às funções valor de menor magnitude, e expressa na forma:

$$\omega^k(n,i) = \frac{\frac{1}{\mathcal{V}^k(n,i)}}{\sum_{j\in\chi}\frac{1}{\mathcal{V}^k(n,i)}},\ (n,i)\in\chi$$

sendo χ o conjunto de estados-amostra.

A Figura 6.4 contém a função valor ótima e a função aproximada obtida pelo algoritmo PDRA com ponderação, no subconjunto S^1, para o Caso **A**. Os resultados nos demais sub-

Tab. 6.8: Resultados Para o Algoritmo Ponderado

Exemplo	Ordem do Pol.	Er. Quad. Médio(%)	Er. Max. (%)	Tempo (s)
Caso A	2	0,0311	38,61	41
	3	0,0195	64,41	47
	4	0,0590	58,02	81
Caso B	2	0,7601	167,12	41
	3	0.2701	98,31	57
	4	0,2820	102,38	76
Caso C	2	0,0492	79,99	56
	3	0,0134	33,90	97
	4	0,0055	29,37	106
Caso D	2	0,2090	83,94	60
	3	0,0027	10,09	64
	4	0,0739	80,20	106
Caso E	2	0,0021	14,80	54
	3	0,0014	11,33	80
	4	0,0005	7,37	92
Caso F	2	0,0069	19,73	63
	3	0,0108	22,91	66
	4	0,0075	20,56	95

conjuntos (S^2 e S^0) são similares e foram omitidas aqui. Os parâmetros do algoritmo são os mesmos utilizados no algoritmo não ponderado. Por meio de comparação entre as figuras 6.3 e 6.4, nota-se que o algoritmo ponderado apresenta oscilações bem menores na região S_F. Repare também que a aproximação obtida pelo algoritmo ponderado é claramente superior àquela obtida pelo algoritmo sem ponderação.

Os experimentos da Tabela 6.7 foram repetidos utilizando-se o algoritmo ponderado e os resultados são apresentados na Tabela 6.8. Observe na Tabela 6.8 que o algoritmo ponderado possui um desempenho significativamente melhor em termos de erro máximo na região S_F. A melhoria de desempenho é obtida a custo de um acréscimo no tempo de execução do algoritmo em relação ao algoritmo sem ponderação. Não obstante, o algoritmo ponderado permanece significativamente mais rápido que o algoritmo exato (comparar com o tempo de execução na primeira linha da Tabela 6.6). Também pode-se observar a tendência de melhoria de desempenho do algoritmo ao se utilizar polinômios de ordem maior. No entanto, em alguns casos o desempenho do algoritmo piora ao se utilizar aproximações de ordem maior. Isso se deve à tendência de termos de ordem maior introduzirem maior erro numérico. Essa tendência é compensada quando termos de ordem maior melhoram a precisão da aproximação de uma maneira mais significativa. Caso contrário, esses erros podem piorar a qualidade da aproximação em relação a polinômios de ordem menor, como é o caso em alguns dos experimentos relatados na Tabela 6.8. Naturalmente, polinômios de ordem maior tendem a tornar o algoritmo um pouco mais lento. Dado que o desempenho do algoritmo ponderado é, em geral, melhor que o do algoritmo sem ponderação, este será utilizado nos demais experimentos apresentados nesta seção.

O próximo experimento apresentado nesta seção visa ilustrar a influência do número de estados de amostragem utilizados no desempenho do algoritmo aproximado. Simula-se o Caso A, utilizando-se polinômios de aproximação de ordem 2, $n_B = -10$, $n_C = 10$. O parâmetro ss é variado e os resultados obtidos são descritos na Tabela 6.9.

Tab. 6.9: Variações de ss Para o Algoritmo Ponderado

ss	**Er. Quad. Médio(%)**	**Er. Max. (%)**	**Tempo (s)**
100	0,0311	38,61	41
225	0,0766	69,66	68
400	0,0560	62,01	124
625	0,0293	39,68	160
900	0,0338	47,13	203

Observe na Tabela 6.9 que o aumento do número de estados de amostragem não constitui, por si só, uma garantia de melhora de desempenho do algoritmo. De fato, foram verificadas apenas oscilações no erro quadrático médio, com melhora apenas quando $ss = 625$.

No último experimento apresentado nesta seção, são variados os parâmetros n_B e n_C. Utiliza-se o Caso A como problema de estudo com $ss = 100$. Para efeito de comparação com os demais experimentos, os erros são calculados no subconjunto fixo $\tilde{S} = \tilde{N} \times \mathcal{I} \times \mathcal{J}$, com $\tilde{N} = \prod_{j=1}^{j} \{-10, \ldots, 10\}$. Repare que $\tilde{S}$ coincide com S_F quando $n_B = -10$ e $n_C = 10$, como é o caso nas simulações anteriores; para os demais Casos na Tabela 6.10, $\tilde{S} \subset S_F$.

Tab. 6.10: Variações de n_B e n_C Para o Algoritmo Ponderado

n_B	n_C	**Er. Quad. Médio(%)**	**Er. Max. (%)**	**Tempo (s)**
-10	10	0,0311	38,61	41
-15	15	0,0507	53,91	83
-20	20	0,0001	6,66	214
-25	25	0	0,1123	331
-30	30	0	0,0086	440

Repare que há uma tendência de melhora na solução aproximada à medida que a região S_F é estendida. A qualidade da solução aproximada melhora sensivelmente no subconjunto $\tilde{S}$, e para os dois últimos casos da Tabela 6.10 obtém-se praticamente a função valor ótima nesse subconjunto. Esse fato reforça a idéia de que os estados mais distantes do estoque zero têm influência limitada na função valor dos estados próximos a essa região, notadamente na região de estabilidade induzida pela política de controle.

6.6 Considerações Finais

Sugeriu-se, um esquema de aproximação polinomial a ser utilizado em conjunto com um algoritmo de PDRA. Apresentou-se um pseudo-código do algoritmo em questão, utilizando

representação exata em um subconjunto do espaço de estados, no intuito de conferir mais confiabilidade às soluções sub-ótimas obtidas na região próxima ao estado de estoque nulo.

Foram apresentados experimentos numéricos destinados a ilustrar o desempenho do algoritmo de PDRA, utilizando-se a arquitetura de aproximação polinomial introduzida na Seção 6.2. Foi efetuada também uma análise de sensibilidade do algoritmo com relação aos parâmetros de aproximação utilizados.

Os resultados numéricos obtidos demonstram redução significativa no tempo de convergência do algoritmo PDRA com relação ao algoritmo clássico de iteração de valor. A qualidade das soluções aproximadas obtidas foi analisada e verificou-se um desempenho razoável dos algoritmos de PDRA propostos na região de baixos níveis de estoque/déficit.

As simulações sugerem que a utilização de um esquema de ponderação, com base na função valor aproximada no procedimento de obtenção dos pesos a cada iteração, tende a introduzir mais confiabilidade ao algoritmo. Os resultados obtidos sugerem uma melhora significativa da aproximação na região de baixos níveis de estoque/déficit, em relação ao algoritmo sem ponderação.

Capítulo 7

Considerações Finais

Introduziu-se nesta tese um modelo discreto para o problema de Produção e Estoque com vários produtos e múltiplos estágios de produção. O modelo apresentado pode ser visto como uma discretização a eventos de um modelo de processo markoviano determinístico por partes (PMDP). O modelo apresentado não se restringe a processos de chegada de demanda poissonianos e acomoda processos de chegada quaisquer. Além disso, o modelo introduzido permite a divisão do processo de produção em estágios bem definidos que, uma vez iniciados, não podem ser paralisados. Isso traz mais flexibilidade ao decisor e torna o modelo mais adequado para processos de produção que apresentem essas características. Temos assim, um modelo bastante geral com relação aos modelos encontrados na literatura, que incorpora a possibilidade de divisão do processo de produção em vários estágios.

Além das vantagens já mencionadas, o problema de controle por intervenções associado ao modelo apresentado pode ser resolvido por meio de programação dinâmica discreta, ao passo que a solução de PMDP's é obtida por processos mais complexos, como a resolução de equações de viscosidade (integro-diferenciais).

Infelizmente, associado ao procedimento conciso e elegante de programação dinâmica está o conhecido *mal da dimensionalidade*, isto é, a impossibilidade de obter soluções para problemas de grande porte, com número bastante elevado de estados de Markov, possivelmente infinitos. Assim, no intuito de tornar possível a obtenção de soluções sub-ótimas para problemas P&E de grande porte, esta tese desenvolveu-se em duas frentes. Primeiro concentrou-se as atenções em conceitos de estabilidade estocástica, tornando possível a definição de procedimentos destinados a obter soluções sub-ótimas estocasticamente estáveis. Num segundo momento, concentrou-se a atenção em algoritmos de programação dinâmica aproximada, a partir de uma arquitetura de aproximação fixa.

Apresentou-se condições necessárias e suficientes para a estabilidade estocástica do sistema estudado. A partir desses resultados, é possível definir políticas heurísticas de controle no exterior de uma região de estabilidade estabelecida à priori de forma a garantir estabilidade estocástica. Isso permite iterar o operador de programação dinâmica apenas na região de estabilidade desejada, utilizando-se condições de contorno definidas à priori. Os resultados para alguns sistemas de produção de dois produtos sugerem o bom desempenho de políticas de controle assim estabelecidas na região de estabilidade desejada.

No campo de algoritmos de programação dinâmica aproximada, alguns novos resultados

teóricos foram obtidos neste trabalho. Os resultados prévios existentes na literatura mostram a possibilidade de divergência de algoritmos de programação aproximada com representação paramétrica (aproximada) da função valor. Os algoritmos com convergência garantida são associados a arquiteturas de aproximação particulares, especialmente desenhadas para garantir a estabilidade do algoritmo aproximado. Além disso, esses algoritmos não possuem garantia de desempenho, isto é, nada se pode inferir sobre a solução obtida. Introduziu-se nesta tese um algoritmo com representação paramétrica, que converge para qualquer arquitetura de aproximação arbitrária. Além disso, a resposta do algoritmo, isto é, a solução aproximada obtida por este, foi caracterizada como um ponto fixo do operador de programação dinâmica com relação a uma norma de subconjunto.

Para o problema P&E estudado, sugeriu-se uma arquitetura de aproximação polinomial, com representação exata em um conjunto de estados próximos à região de estoque nulo. Efetuou-se experimentos numéricos a fim de avaliar o desempenho do algoritmo aproximado, além de análises de sensibilidade com relação aos parâmetros. Os resultados sugerem a adequação da arquitetura de aproximação proposta a problemas P&E com funções de custo de estoque/déficit polinomiais. Esses resultados mostram, além disso, que o algoritmo aproximado obtém resultados mais confiáveis para os estados com representação exata.

Como perspectivas para trabalhos futuros, pode-se buscar resultados equivalentes em termos de estabilidade estocástica para processos a tempo contínuo, notadamente processos markovianos determinísticos por partes (PMDP). Uma extensão natural desta tese envolve a generalização do modelo apresentado para problemas multi-produto com operação de várias máquinas. Outra perspectiva nessa área inclui o emprego de controle por horizonte retrocedente nos estados da região de estabilidade estocástica do problema. Essa abordagem também se aplica, naturalmente, a processos de decisão markovianos (PDM).

Na área de estabilidade programação dinâmica aproximada, novas caracterizações podem ser buscadas para o ponto fixo do algoritmo aproximado introduzido nesta tese. Pode-se, além disso, investigar a relação entre soluções obtidas para normas de subconjunto distintas. Existe também o potencial para investigar outros algoritmos com propriedades de convergência similares, no intuito de se buscar algoritmos com pontos de acumulação próximos á solução ótima do problema.

Referências Bibliográficas

Akella, R. e Kumar, P. R. (1986). Optimal control of production rate in a failure prone manufacturing system, *IEEE Trans. Automat. Contr.* **31**(2): 116–126.

Almudevar, A. (2001). A dynamic programming algorithm for the optimal control of piecewise deterministic Markov processes, *SIAM J. Control Optim.* **4**(1): 525–539.

Arruda, E. F. (2002). *Investimento e produção de múltiplos itens em presença de incertezas*, Master's thesis, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, Campinas/SP.

Arruda, E. F. e do Val, J. B. R. (2002). Problema de controle impulsional para vários itens em produção, *Anais do XIV Congresso Brasileiro de Automática*, Natal-RN, pp. 1385–1390.

Arruda, E. F., do Val, J. B. R. e Almudevar, A. (2005). Function approximation for a production and storage problem under undertainty, *Proceedings of the IEEE International Conference on Mechatronics & Automation*, Niagara Falls, Canada, pp. 665–670.

Baird, L. C. (1995). Residual algorithms: Reinforcement learning with function approximation, *International Conference on Machine Learning*, pp. 30–37.
*citeseer.csail.mit.edu/baird95residual.html

Bellman, R. (1957). *Dynamic programming*, Princeton University Press, Princeton, NJ.

Bertsekas, D. P. e Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*, Athena Scientific, Belmont.

Boukas, E. K. e Haurie, A. (1990). Manufacturing flow control and preventive maintenance: a stochastic control approach, *IEEE Trans. Automat. Contr.* **35**(9): 1024–1031.

Boukas, E. K., Zhu, Q. e Zhang, Q. (1994). Piecewise deterministic Markov process model for flexible manufacturing systems with preventive maintenance, *J. Opt. Theory & Appl.* **81**(2): 259–275.

Boyan, J. A. e Moore, A. W. (1995). Generalization in reinforcement learning: safely approximating the value function, *Advances in Neural Information Processing Systems*, Vol. 7, MIT Press.

Brémaud, P. (1999). *Gibbs fields, Monte Carlo simulation, and queues*, Springer-Verlag, New York.

Davis, M. H. A. (1984). Piecewise-deterministic Markov process: A general class of non-diffusion stochastic models, *J. R. Statist. Soc. B* **46**(3): 353–388.

Davis, M. H. A. (1993). *Markov models and optimization*, Chapman and Hall, London.

Davis, M. H. A., Dempster, M. A. H., Sethi, S. P. e Vermes, D. (1987). Optimal capacity expansion under uncertainty, *Advances in Applied probability* **19**: 156–176.

do Val, J. B. R. e Salles, J. L. F. (1999). Optimal production with preemption to meet stochastic demand, *Automatica* **35**(11): 1819–1828.

Federgruen, A. e Zipkin, P. (1986). An inventory model with limited production capacity and uncertain demands II. The discounted cost criterion, *Math. Ops. Res.* **11**(2): 208–215.

Gatarek, D. (1992). Optimality conditions for impulse control of piecewise deterministic processes, *Math. Control Signals Systems* **5**: 217–232.

Golub, G. H. e van Loan, C. F. (1996). *Matrix Computations*, 3 edn, John Hopkins University Press, Baltimore.

Gordon, G. (1995). Stable function approximation in dynamic programming, *Proceedings of the IMCL '95*.

Goyal, S. K. e Giri, B. C. (2003). The production-inventory problem of a product with time varying demand, production and deterioration rates, *Eur. J. Ops. Res.* **147**: 549–557.

Gravish, B. e Graves, S. C. (1981). Production/inventory systems with a stochastic production rate under a continuous review policy, *Comp. & Ops. Res.* **8**(3): 169–183.

Gullu, R. (1998). Base stock policies for production/inventory problems with uncertain capacity levels, *Eur. J. Ops. Res.* **105**: 43–51.

Hernández-Lerma, O. (1989). *Adaptive Markov control processes*, Springer-Verlag, New York.

Huang, L., Hu, J. e Vakili, P. (1998). Optimal control of a multi-stage manufacturing system: control of production rate and temporary increase in capacity, *Proceedings of the* 37^{th} *IEEE Conference on Decision and Control*, pp. 2130–2155.

Jean-Marie, A. e Tidball, M. (1997). Application of the impulsive control of piecewise-deterministic processes to multi-item single machine scheduling, *6th Viennese Workshop on Optimal Control, Dynamic Games, Nonlinear Dynamics and Adaptive Systems*, Vienne.

Luss, H. (1982). Operations research and capacity expansion problems: a survey, *Operations Research* **30**(5): 907–947.

Mancinelli, E. M. e Gonzalez, R. L. V. (1997). Multi-item single machine scheduling optimization. The case with piecewise deterministic demands, *Technical Report RR-3144*, Institut National de Recherche en Informatique et en Automatique.
*citeseer.nj.nec.com/mancinelli97multiitem.html

Meyn, S. P. e Tweedie, R. L. (1993). *Markov Chains and Stochastic Stability*, Springer-Verlag, New York.

Monticino, M. e Weisinger, J. (1995). Optimal cutoff strategies in capacity expansion problems, *Naval research logistics* **42**: 1021–1039.

Moore, A. W. e Atkeson, C. G. (1993). Prioritized sweeping: reinforcement learning with less data and less time, *Machine Learning* **13**: 103–130.

Moresino, F., Pourtallier, O. e Tidball, M. (1999). Using viscosity solution for approximations in piecewise deterministic Markov processes, *Technical Report 3687*, Institut National de Recherche en Informatique et en Automatique.

Perkins, J. R. e Srikant, R. (1997). Scheduling multiple part-types in an unreliable single machine manufacturing system, *IEEE Trans. Automat. Contr.* **42**(3): 364–377.

Perkins, J. R. e Srikant, R. (1998). Hedging policies for failure prone manufacturing systems: Optimality of JIT and bounds on buffer levels, *IEEE Trans. Automat. Contr.* **43**: 953–958.

Perkins, J. R. e Srikant, R. (1999). Failure-prone production systems with uncertain demand, *System modelling and optimization*, Chapman and Hall/CRC Research Notes in Mathematics, pp. 289–297.

Perkins, T. J. e Precup, D. (2003). A convergent form of approximate policy iteration, *in* S. T. S. Becker e K. Obermayer (eds), *Advances in Neural Information Processing Systems 15*, MIT Press, Cambridge, MA, pp. 1595–1602.

Presman, E., Sethi, S. e Zhang, Q. (1995). Optimal feedback production planning in a stochastic N-machine flowshop, *Automatica* **31**: 1325–1332.

Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*, John Wiley & Sons, New York.

Reynolds, S. I. (2002). The stability of general discounted reinforcement learning with linear function approximation, *Proceedings of the UK Workshop on Computational Intelligence*, Birmingham-UK, pp. 139–146.

Robbins, H. e Monro, S. (1951). A stochastic approximation method, *Ann. Math. Statist.* **13**: 400–407.

Salama, Y. (2000). Optimal control of a simple manufacturing system with re-starting costs, *Ops. Res. Letters* **26**: 9–16.

Salles, J. L. F. e do Val, J. B. R. (2001). An impulse control problem of a production model with interruptions to follow stochastic demand, *Eur. J. Ops. Res.* **132**: 123–145.

Salles, J. L. F. e do Val, J. B. R. (2002). Controle da expansão da capacidade de múltiplas instalações com demanda aleatória, *Anais do XIV Congresso Brasileiro de Automática*, Sociedade Brasileira de Automática, Natal-RN, pp. 1355–1360.

Sethi, S. P., Yan, H., Zhang, H. e Zhang, Q. (2002). Optimal and hierarchical controls in dynamic stochastic manufacturing systems: A survey, *Manuf. & Serv. Ops. Management* **4**(2): 133–170.

Si, J., Barto, A., Powell, W. e Wunsch, D. (2004). *Handbook of learning and approximate dynamic programming*, John Wiley & Sons-IEEE Press, Piscataway-NJ.

Song, D. e Sun, Y. (1998). Optimal service control of a serial production line with unreliable workstations and random demand, *Automatica* **34**(9): 1047–1060.

Sutton, R. S. (1990). Integrated architectures for learning, planning and reacting based on approximating dynamic programming, *Proceedings of the* 7^{th} *International Conference on Machine Learning*, Morgan Kaufmann, pp. 216–224.

Sutton, R. S. e Barto, A. G. (1998). *Reinforcement learning: an introduction*, MIT Press, Cambridge.

Szepesvári, C. (2001). Convergent reinforcement learning with value function interpolation, *Technical Report TR-2001-02*, Mindmaker Ltd., Budapest 1121, Konkoly Th. M. u. 29-33, HUNGARY.
*citeseer.csail.mit.edu/article/szepesv01convergent.html

Tesauro, G. (1992). Practical issues in temporal difference learning, *Machine Learning* **8**(3): 257–277.

Tsitsiklis, J. N. e Roy, B. V. (1996). Feature-based methods for large scale dynamic programming, *Machine Learning* **22**(1-3): 59–94.

Tsitsiklis, J. N. e Roy, B. V. (1999). Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing in high-dimensional financial derivatives, *IEEE Transactions on Automatic Control* **44**(10): 1840–1851.

Watkins, C. e Dayan, P. (1989). Q-learning, *Machine Learning* **8**: 279–292.

Williams, R. e Baird, L. (1993). Tight performance bounds on greedy policies based on imperfect value functions, *Technical Report NU-CCS-93-14*, Northeastern University.
*citeseer.ist.psu.edu/article/williams93tight.html

Yan, H. e Zhang, Q. (1997). A numerical method in optimal production and setup scheduling of stochastic manufacturing systems, *IEEE Trans. Automat. Contr.* **42**(10): 1452–1455.

Apêndice A

Algoritmos Convergentes Baseados Em Projeções Não Expansivas

Apresenta-se, neste apêndice, um resumo dos resultados acerca da convergência de operadores de projeção não expansivos contidos em (Gordon 1995). Com o propósito de facilitar a leitura, apresenta-se abaixo a definião de mapeamentos não expansivos encontrada em (Gordon 1995).

Definição 6. *Seja S um espaço vetorial completo com norma $||\cdot||$. A função $f : S \rightarrow S$ é um* mapeamento não expansivo *ou uma* não-expansão *se, para todos os pontos a e b pertencentes a S,*

$$||f(a) - f(b)|| \leq ||a - b||.$$

Claramente, todo mapeamento contrativo é também um mapeamento não expansivo. O principal resultado obtido por (Gordon 1995) é expresso no seguinte teorema, cuja demonstração é trivial.

Teorema 4. *Seja T o operador de PD para algum PDM com taxa de desconto $\gamma < 1$. Seja A uma arquitetura de aproximação com mapeamento $\mathcal{P}_A$. Suponha que $\mathcal{P}_A$ é um operador não expansivo em norma infinita. Então $\mathcal{P}_A \circ T$ tem um fator de contração γ; assim o algoritmo de iteração de valor aproximada converge em norma infinita a uma taxa γ quando aplicado a esse PDM.*

A partir desse resultado, introduz-se a definição de *mediadores*: operadores de projeção que satisfazem a definição 6, aos quais o Teorema 4 se aplica. Essa definição encontra-se transcrita abaixo.

Definição 7. *Um esquema de aproximação é um mediador se todo valor aproximado é a média ponderada de zero ou mais valores-alvo and possivelmente algumas constantes pré-determinadas. Os pesos envolvidos no cálculo da função aproximada $\mathcal{V}(x)$ podem depender do conjunto de amostragem M, mas não podem depender dos valores-alvo $V(y)$ nesse conjunto de amostragem. Mais precisamente, para um conjunto M fixo, se S tem n elementos, devem existir n números reais k_i, n^2 números reais não negativos β_{ij}, e n escalares não negativos β_i,*

tais que para todo $i \in S$*, tem-se*

$$\beta_i + \sum_{j \in S} \beta_{ij} = 1$$

e

$$\mathcal{V}(x) = \beta_x k_x + \sum_{y \in S} \beta_{xy} V(y).$$

A maior parte dos valores β_{xy} são normalmente nulos. Em particular β_{xy} deve ser zero se $j \notin M$.

Da definição acima verifica-se que, para *mediadores*, o valor aproximado em um dado estado é uma média ponderada dos valores-alvo, sendo que os pesos são determinados pela distâncias entre os estados, não sendo portanto, afetadas pelos valores-alvo $V(y)$, $y \in M$. O último teorema desse apêndice é uma clara aplicação do Teorema 4 aos operadores de projeção aqui denominados mediadores.

Teorema 5. *O mapeamento* $\mathcal{P}_A$ *associado a um mediador* A *é uma não-expansão em norma infinita; logo o algoritmo de iteração de valor aproximado* A *converge quando aplicado a qualquer MDP descontado.*

O algoritmo convergente derivado do Teorema 5 pode ser visto como um caso particular do Algoritmo 2 da Seção 4.6.2, sem o procedimento de controle de expansão (passo). O pseudo código para um algoritmo iterado na classe de projeções não expansivas é apresentado abaixo.

Algoritmo 4 Algoritmo PDRA Convergente Para Não-Expansões

Passo 0 Início

1. Escolha $\sigma \in (\theta, 1)$, $m \geq 1$
2. Estabeleça a tolerância $tol = \delta$
3. Escolha $\mathcal{V}_0 \in A$
4. $k \leftarrow 0$

Passo 1 Atualização

1. $k \leftarrow k + 1$
2. $\mathcal{V}_k \leftarrow \mathcal{P}_A(T\mathcal{V}_{k-1})$

Passo 3 Teste de Convergência
Se($y \leq tol$) Parar
Caso contrário Voltar ao Passo 1