COPPE
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia
UFRJ

NOVOS ALGORITMOS DE PROGRAMAÇÃO DINÂMICA APLICADOS À CADEIA DE SUPRIMENTO DE MINERAÇÃO

João Marcelo Leal Gomes Leite

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Produção, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Produção.

Orientador: Lino Guimarães Marujo

Rio de Janeiro
Novembro de 2022

NOVOS ALGORITMOS DE PROGRAMAÇÃO DINÂMICA APLICADOS À CADEIA DE SUPRIMENTO DE MINERAÇÃO

João Marcelo Leal Gomes Leite

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE PRODUÇÃO.

Orientador: Lino Guimarães Marujo

Aprovada por: Prof. Lino Guimarães Marujo
Prof. Edilson Fernandes de Arruda
Prof. Davi Michel Valladão
Prof. José Leandro Felix Salles
Prof. Virgílio José Martins Ferreira Filho

RIO DE JANEIRO, RJ – BRASIL
NOVEMBRO DE 2022

Leite, João Marcelo Leal Gomes

Novos algoritmos de Programação Dinâmica aplicados à Cadeia de Suprimento de Mineração/João Marcelo Leal Gomes Leite. – Rio de Janeiro: UFRJ/COPPE, 2022.

XVII, 110 p.: il.; 29, 7cm.

Orientador: Lino Guimarães Marujo

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Produção, 2022.

Referências Bibliográficas: p. 79 – 110.

1. Mineração. 2. Cadeia de Suprimentos. 3. Otimização Estocástica. 4. Processo de Decisão de Markov. 5. Programação Dinâmica. I. Marujo, Lino Guimarães. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Produção. III. Título.

Em memória da minha
madrinha e do Ralphinho.

Agradecimentos

A minha família, principalmente meu pai, minha mãe e Vanessinha, pela companhia e suporte para toda a vida.

Aos meus orientadores e amigos, Edilson Arruda e Lino Marujo, por todo apoio e orientação não apenas na tese, mas durante todo o doutorado.

Às minhas princesas, Maria Lícia e Ana Luísa, por cada olhar que serviu muito como incentivo.

À minha banca, Davi, José Leandro e Virgílio, pela disponibilidade e contribuição.

À COPPE (professores, colegas e funcionários), por me proporcionarem estes anos de conhecimento.

A CAPES, pelo apoio financeiro.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

NOVOS ALGORITMOS DE PROGRAMAÇÃO DINÂMICA APLICADOS À CADEIA DE SUPRIMENTO DE MINERAÇÃO

João Marcelo Leal Gomes Leite

Novembro/2022

Orientador: Lino Guimarães Marujo

Programa: Engenharia de Produção

Este trabalho apresenta um modelo de otimização estocástica para a cadeia logística de mineração, envolvendo todos os seus elos, das minas até os clientes finais. Pela natureza sequencial das decisões e pelos longos horizontes envolvidos, utilizou-se a modelagem de Processos de Decisão Markoviano (*Markov Decision Processes - MDP*). Comparando com os trabalhos existentes, verificou-se que muitos novos trabalhos com otimização estocástica em mineração foram publicados recentemente, porém nenhum englobava todos os elos da cadeia simultaneamente, principalmente dada sua complexidade que dificulta a obtenção da solução em tempos computacionais razoáveis (*mal da dimensionalidade*). Para contornar este problemas, foi necessário o desenvolvimento de dois novos algoritmos de programação dinâmica que atuassem na redução do espaço de estados e espaço de ações. O TABA é baseado em agregação temporal e expande a aplicação deste tipo de algoritmo para uma nova classe de MDP. O LSPSI atua na redução do espaço de ações, melhorando o processo de busca introduzido no passo de avaliação de política pelo algoritmo PSI. Estes algoritmos, quando combinados, trouxeram grande ganho de performance e sobrepuseram a eficiências tanto dos algoritmos mais tradicionais, iteração de valor e iteração de política, como de algoritmos mais modernos, SDDP. No final, o modelo e os algoritmos são testados em um problema numérico da cadeia de mineração e os resultados obtidos comprovam a importância da otimização estocástica considerando a visão completa da cadeia e a eficiência dos algoritmos propostos.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

NEW DYNAMIC PROGRAMMING ALGORITHMS APPLIED TO MINING SUPPLY CHAIN

João Marcelo Leal Gomes Leite

November/2022

Advisor: Lino Guimarães Marujo

Department: Production Engineering

This work presents a stochastic optimization model for the mining supply chain, involving all its links, from mines to final clients. Due to the sequential nature of the decisions process and the long terms involved, Markov Decision Processes (MDP) modeling was used. Comparing with the existing works, it was found that many new stochastic optimization in mining papers were recently published, but none encompassed all supply chain links simultaneously, mainly given its complexity that makes it difficult to obtain the solution in reasonable computational times (*curse of the dimensionality*). To circumvent this problem, it was necessary to develop two new dynamic programming algorithms to reduce the state space and action space. TABA is based on temporal aggregation and expands the application of this type of algorithm to a new class of MDP. LSPSI works to reduce the action space, improving the search process introduced by in the policy evaluation step by PSI algorithm. These algorithms, when combined, brought great performance gains and overlapped the efficiencies of both more traditional, as value iteration and policy iteration, and more modern algorithms, as SDDP. In the end, the model and the algorithms are tested in a numerical problem of the mining chain and the results obtained prove the importance of stochastic optimization considering supply chain complete view and the efficiency of the proposed algorithms.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

A	Conjunto de todas as decisões possíveis, p. 17
$A(s)$	Conjunto de todas as decisões possíveis para o estado s, p. 17
I	Matriz identidade, p. 20
P^{π}	Matriz com dimensão $\|S\| \times \|S\|$, em que cada elemento representa a probabilidade de transição do estado s para os estado s', aplicando-se uma política determinística estacionária π, onde s são representados nas linhas e s' nas colunas da matriz , p. 20
$Q(s, a)$	Fator Q que aproxima o custo de estar no estado s e tomar a ação a, p. 23
S	Espaço de estados, p. 17
T	Conjunto de todos os períodos de decisão possíveis, p. 17
$V(s)$	Função de custo partindo do estado s, p. 18
$V^*(s)$	Valor ótimo para a função de custo, p. 19
$V^{\pi}(s)$	Função de custo obtida aplicando-se a política π, p. 19
$V^{\pi^*}(s)$	Valor ótimo para a função de custo obtido aplicando-se a política ótima π^*, p. 19
Π	Conjunto de todas as políticas possíveis, p. 19
$\boldsymbol{\theta}$	Vetor de pesos da função de política paramétrica aproximada, p. 25
$\hat{V}(s, \mathbf{w})$	Função paramétrica aproximada para a função de custo, p. 25
λ	Fator de desconto $0 \leq \lambda < 1$ por período, p. 18
$\mathbb{R}$	Conjunto dos números reais, p. 18

$\mathbb{E}_{\xi}\{\cdot\}$	Valor esperado com relação à variável aleatória ξ, p. 13
$\mathbf{w}$	Vetor de pesos da função de custos paramétrica aproximada, p. 25
π	Política de decisão, p. 19
$\pi(a\|s,\boldsymbol{\theta})$	Função paramétrica aproximada para política, p. 25
$\pi(s)$	Política de decisão determinística, dependente apenas do estado s, p. 19
π^*	Política ótima, p. 19
ξ_t	Variável aleatória que representa as incertezas do modelo no período t, p. 17
ξ_t^1	Primeiro elemento do vetor ξ_t: capacidade de fluxo operacional no porto, p. 40
ξ_t^2	Segundo elemento do vetor ξ_t: capacidade de fluxo operacional no PAA, p. 40
ξ_t^3	Terceiro elemento do vetor ξ_t: demanda dos clientes de contrato, p. 40
ξ_t^4	Quarto elemento do vetor ξ_t: preço do mercado spot, p. 40
ξ_t^5	Quinto elemento do vetor ξ_t: custo da navegação internacional partindo do porto, p. 40
a	Vetor de decisão no tempo t, p. 17
a_t	Vetor de decisão no tempo t, p. 17
a_t^1	Primeiro elemento do vetor a_t: volume de produção das minas, p. 40
a_t^2	Segundo elemento do vetor a_t: volume transportado do porto para o PAA, p. 40
a_t^3	Terceiro elemento do vetor a_t: volume transportado do porto para os clientes de contrato, p. 40
a_t^4	Quarto elemento do vetor a_t: volume transportado do porto para o mercado spot, p. 40

a_t^5 Quinto elemento do vetor a_t: volume transportado do PAA para os clientes de contrato, p. 40

a_t^6 Sexto elemento do vetor a_t: volume transportado do PAA para o mercado spot, p. 40

c^1 Capacidade de produção nas minas, p. 42

c^2 Custo de produção, p. 42

c^3 Capacidade de armazenagem no porto, p. 42

c^4 Capacidade de armazenagem no PAA, p. 42

c^5 Custo do transporte marítimo (navegação) local, p. 42

c^6 Preço de venda para os clientes de contrato, p. 42

c^7 Penalidade contratual por não atendimento da demanda, p. 42

c^8 Produção mínima nas minas, p. 42

d Regra de decisão, p. 19

$d(s)$ Regra de decisão determinística, dependente apenas do estado s, p. 19

d_t Regra de decisão no período t, p. 19

i Taxa de desconto para cálculo do VPL por período, p. 42

$p(s'|s,a)$ Probabilidade de transição para o estado s', ao tomar a decisão a no estado s, p. 17

$r(s,a)$ Recompensa esperada obtida no período t ao tomar a decisão a no estado s, p. 18

$r(s,a,s')$ Recompensa obtida no período t ao tomar a decisão a no estado s e o sistema seguir para o estado s', p. 18

r^π Vetor de dimensão $|S|$ com a recompensa esperada em t para cada estado, aplicando-se uma política determinística estacionária π, p. 20

s Vetor de estado no tempo t, p. 17

s' Vetor de estado no tempo t+1, p. 17

s_t Vetor de estado no tempo t, p. 17

s_t^1 Primeiro elemento do vetor s_t: capacidade de fluxo operacional no porto, p. 39

s_t^2 Segundo elemento do vetor s_t: capacidade de fluxo operacional no PAA, p. 39

s_t^3 Terceiro elemento do vetor s_t: demanda dos clientes de contrato, p. 39

s_t^4 Quarto elemento do vetor s_t: preço do mercado spot, p. 39

s_t^5 Quinto elemento do vetor s_t: custo da navegação internacional partindo do porto, p. 39

s_t^6 Sexto elemento do vetor s_t: volume armazenado no porto, p. 39

s_t^7 Sétimo elemento do vetor s_t: volume armazenado no PAA, p. 39

s_{t+1} Vetor de estado no tempo t+1, p. 17

t Escalar representando o período de decisão, p. 17

v^π Vetor de dimensão $|S|$ com os valores da função de custo para cada estado, aplicando-se uma política determinística estacionária π, p. 20

Lista de Abreviaturas

ABD	Decomposição Acelerada de Benders (*Accelerated Benders Decomposition*), p. 14
ADP	Programação Dinâmica Aproximada (*Approximate Dynamic Programming*), p. 2
AVI	Iteração de Valor Aproximada (*Approximate Value Iteration*), p. 35
DES	Simulação de Eventos Discretos (*Discrete Event Simulation*), p. 11
DP	Desvio Padrão, p. 63
IC	Intervalo de Confiança, p. 63
LSPSI	Algoritmo de Iteração de Grupo de Políticas com Busca Local (*Local Search Policy Set Iteration*), p. 44
MDP	Processos de Decisão Markoviano (*Markov Decision Processes*), p. vi
MSSP	Programação Estocástica de Multiestágios (*Multi-Stages Stochastic Programming*), p. 11
PAA	Porto de Armazenagem Avançada, p. 38
PI	Iteração de Política (*Policy Iteration*), p. 26
PO	Pesquisa Operacional, p. 1
PSIAE	Extensão do Algoritmo de Iteração de Conjunto de Políticas (*Policy Set Iteration Algorithm Extension*), p. 55
PSI	Iteração de Conjunto de Políticas (*Policy Set Iteration*), p. 27
RL	Aprendizado por Reforço (*Reinforcement Learning*), p. 2

SA	Recozimento Simulado (*Simulated Annealing*), p. 14
SDDP	Programação Dinâmica Dual Estocástica (*Stochastic Dual Dynamic Programming*), p. 16
SMART	Técnica de Recompensa Média de Semi-Markov (*Semi-Markov Average Reward Technique*), p. 34
SMDP	Processos de Decisão de Semi-Markov (*Semi-Markov Decision Processes*), p. 20
SSMDP	Processos de Decisão Markoviano Parcialmente Estacionários (*semi-stationary Markov decision process*), p. 44
TABA	Algoritmo Baseado na Agregação Temporal (*Time Aggregation Based Algorithm*), p. 44
TSSP	Programação Estocástica de Dois Estágios (*Two-Stages Stochastic Programming*), p. 11
VI	Iteração de Valor (*Value Iteration*), p. 28
VMI	Estoque Gerido pelo Fornecedor (*Vendor-Managed Inventory*), p. 12
VNS	Busca de Variável na Vizinhança (*Variable Neighborhood Search*), p. 16
VPL	Valor Presente Líquido, p. 42
VSI	Iteração de Conjunto de Valores (*Value Set Iteration*), p. 29

Capítulo 1

Introdução

1.1 Motivação

A indústria da mineração é uma atividade econômica bastante relevante no Brasil e no mundo. Em 2021, esta foi responsável por aproximadamente 80% da balança comercial brasileira e atingiu um faturamento total de R$ 339 bilhões (cerca de 5% do PIB nacional), segundo dados do Instituto Brasileiro de Mineração (Ibram) [2]. No mundo, a mineração também tem um papel bastante significativo, e seu faturamento total chegou a 656 bilhões de dólares em 2020 [3]. Por ser uma indústria globalizada, sua cadeia logística é bastante complexa. As primeiras etapas envolvem a extração nas minas, o transporte ferroviário e o carregamento nos portos dos países produtores. Em seguida ocorre o transporte marítimo até os clientes nos países consumidores.

Essa cadeia logística complexa é um terreno fértil para o uso da pesquisa operacional (PO). LEITE *et al.* [1] realizaram uma revisão bibliográfica sobre a aplicação de PO na cadeia logística de mineração. Apesar das diversas incertezas presentes na cadeia estudada (por exemplo: capacidades operacionais, custos de transporte, demanda e preço do minério), os autores constataram que a aplicação de modelos estocásticos apresentou crescimento significativo apenas nas últimas décadas e ainda há espaço para desenvolvimento de novos trabalhos nessa área. Um maior detalhamento sobre as aplicações de PO na indústria da mineração pode ser encontrado na Seção 2.1.

Entre os diversos modelos de otimização estocástica existentes, *processos de decisão markovianos* (Markovian Decision Processes - MDP) se destacam por conseguir tratar todas as informações disponíveis e retornar o melhor conjunto de decisões (política) considerando longos intervalos de tempo (horizontes infinitos). Por conta dessas características, MDP são bastante utilizados em problemas de gestão de estoques (*inventory management problems*). Tais problemas envolvem decisões de

produção e transferência de estoques em face às incertezas na demanda, nos prazos de entrega e nos níveis de produção. Tipicamente, uma política ótima de estoque prescreve decisões com base nos níveis de estoque observados [4–7]. Modelos de gestão de estoques serão abordados mais detalhadamente na Seção 2.2. Apesar de todo sucesso atingido em outras indústrias, LEITE *et al.* [1] também constataram que o uso de MDP na otimização da cadeia logística da mineração é muito baixo, se limitando a alguns elos da cadeia.

Avalia-se que a pouca utilização de MDP na indústria da mineração pode ser causada por dificuldades na modelagem e na solução de problemas complexos, conhecida na literatura como *maldição da dimensionalidade* (*curse of dimensionality*) [4, 8, 9]. De fato, problemas de dimensão relativamente moderada podem gerar espaços de estados e de decisões com dimensões muito elevadas, dificultando ou inviabilizando sua solução. A busca por algoritmos que contornem esta dificuldade motivou o surgimento de extensiva literatura, notadamente nas áreas de: agrupamento de estados [10–12], agregação temporal [13–15] redução do espaço de ações [16], aprendizado por reforço (*reinforcement learning - RL*) [17] e programação dinâmica aproximada (*approximate dynamic programming - ADP*) [18]. A Seção 2.3 traz um maior detalhamento dessas linhas de pesquisa. Ambas as áreas promoveram considerável avanço em domínios específicos, embora muitas de suas soluções sejam aproximações sub-ótimas e a *maldição da dimensionalidade* continue sem solução definitiva.

1.2 Problema e Objetivo

Nesse contexto, espera-se que este trabalho responda às seguintes perguntas:

1. Como a modelagem por meio de MDP pode contribuir para o planejamento logístico da cadeia de mineração?

2. É possível melhorar o desempenho dos algoritmos de MDP já existentes para solução de problemas de grande escala?

Para resolver o problema proposto e responder às perguntas listadas acima, este trabalho tem por objetivo:

1. Desenvolver uma nova modelagem para o planejamento logístico da indústria da mineração que englobe todas as suas etapas (minas, ferrovia, porto, navegação) por meio de MDP, e validar os resultados por meio de comparações com outras técnicas de otimização estocástica;

2. Desenvolver novos algoritmos que possam ser aplicados em conjunto e tenham bom tempo de resposta para problemas envolvendo cadeias logísticas com-

plexas, a fim de contornar a *maldição da dimensionalidade* e mantendo os benefícios inerentes aos MDP.

1.3 Relevância e Originalidade

Como foi possível constatar em [1], a aplicação de otimização estocástica na cadeia logística de mineração ganhou força a partir da última década. Os problemas existentes trazem grande complexidade devido à quantidade de elos envolvidos, bem como às incertezas existentes em todo o processo. Várias técnicas de otimização estocástica já foram usadas, principalmente em partes isoladas da cadeia, tais como: planejamento de produção nas minas considerando sua arquitetura e alocação de equipamentos e navios nos portos. Quando aplicada de forma isolada, apenas os resultados locais são otimizados, não necessariamente atingindo o ótimo global para toda a cadeia de suprimentos. LEITE *et al.* [1] também concluíram que há poucos trabalhos envolvendo otimização estocástica em toda as etapas de forma conjunta, das minas até os clientes finais. Apesar dos resultados comprovados para problemas de gestão de estoques, os autores não localizaram aplicação da modelagem MDP na cadeia da mineração. Complementando o levantamento feito em [1], foram identificados apenas dois trabalhos que aplicam o MDP em problemas de mineração, sendo o escopo em elos específicos e não em toda a cadeia logística [19, 20]. Quando se trabalha apenas com alguns dos elos da cadeia, apenas estes têm seus custos otimizados, podendo não trazer a maior eficiência para toda cadeia logística. Por outro lado, quando se trabalha com todos os elos, espera-se que a solução traga o aumento da eficiência e a redução do custo logístico de toda a cadeia, não ficando restrita às partes isoladas. Considerando a importância e os valores envolvidos na indústria da mineração, citados na Seção 1.1, os impactos na melhoria da performance e redução de custos ao se buscar a otimização global serão extremamente significativos.

Por outro lado, para que a aplicação da modelagem MDP em uma cadeia tão complexa tenha êxito, é necessário que o tempo computacional para obtenção das soluções seja baixo. Desta forma, conseguir avançar com a fronteira do conhecimento dos algorítimos de programação dinâmica, ajudando a contornar a *maldição da dimensionalidade* é fundamental, e contribuirá para o avanço da aplicação do MDP em problemas complexos. Nesta tese, dois algoritmos inspirados nos trabalhos recentes de [21, 22] são propostos. O primeiro usa a ideia de grupos de políticas desenvolvidas em [21] para reduzir o espaço de decisões. O segundo utiliza os conceitos de partição de estados e seus pontos de equilíbrio para criar cadeias agregadas, reduzindo o espaço de estados [22]. Como os dois algoritmos atuam em frentes diferentes, o primeiro no espaço de ações e o segundo no espaço de estados, é possível combiná-los aumentando ainda mais a eficiência final. Já que os dois algoritmos

propostos avançam com trabalhos apresentados recentemente [21, 22], eles são inéditos na literatura. Conforme será mostrado na Seção 4, os resultados obtidos foram promissores e apontam para um possível contorno à *maldição da dimensionalidade*.

1.4 Delimitação

Além de definir o objetivo e o problema que será estudado nesta tese, também é importante definir seus limites de atuação.

Na modelagem para o planejamento logístico da indústria da mineração por meio de MDP, foi elaborado um modelo específico, com características comuns a essa indústria. Contudo, o modelo não é genérico e aplicável de forma direta a todas as cadeias desta indústria. Espera-se que no futuro, adaptações do modelo proposto permitam sua aplicação em outras cadeias da atividade de mineração, ou até mesmo em outros segmentos com características comuns, tais como a indústria de petróleo e gás. O modelo a ser apresentado aborda os problemas de planejamento da produção e gerenciamento de estoques. Não serão tratados os problemas específicos das minas, como: modelos de arquitetura de minas, bem como o planejamento de produção e extração de minério. No porto, não serão tratados os problemas de roteirização de esteiras nem o planejamento do pátio de armazenagem. Também não serão tratados os problemas de alocação de equipamentos nos pontos de estoque e alocação de navios e trens nos modais de transportes. Para finalizar, também não será abordado o problema de expansão de capacidade, nem de mistura.

Os dois algoritmos desenvolvidos são específicos para aplicação em problemas com modelagem de MDP e serão comparados tanto com algoritmos clássicos, como com alguns dos mais modernos em programação dinâmica. Além da performance isolada, os dois serão combinados em um algoritmo final. Os detalhes específicos desses algoritmos, bem como as características dos problemas que favorecem suas implementações, estão descritos na Seção 3.2.

Para finalizar, na aplicação da modelagem e do algoritmo em um caso prático, não se entrará em detalhes sobre as rotinas de obtenção e a armazenagem dos dados. Assume-se que os dados estão disponíveis de forma estruturada, prontos para serem utilizados. Neste trabalho, tanto a modelagem como os algoritmos dela resultantes foram implementados usando a linguagem Python, os pseudo-códigos serão apresentados oportunamente.

1.5 Definição dos termos

Tendo em vista a natureza interdisciplinar do trabalho, esta seção contém um glossário dos termos técnicos mais utilizados no decorrer do trabalho.

Cadeia de Suprimentos ou Cadeia Logística: "Cadeia de Suprimentos consiste em todas as partes envolvidas, direta ou indiretamente, no cumprimento de uma entrega a um cliente. A cadeia de suprimentos engloba não apenas o produtor e os fornecedores, mas também os transportadores, os pontos de armazenagem, os pontos de venda, e até os próprios clientes" [23].

Cadeia de Markov: é um processo estocástico no qual a probabilidade de transição para o próximo estado depende apenas do estado atual. Sistemas com essa propriedade são também chamados de *markovianos*;

Minas: local onde é feita a operação de extração do minério de ferro. Após a extração, o minério é escoado pelos trens e levado para o embarque no porto;

Navegação Marítima: Transporte do minério por navio, do porto de origem até uma região próxima ao cliente final;

Porto: onde o minério é armazenado até o carregamento nos navios. Durante a armazenagem pode ocorrer a mistura de produtos básicos, gerando novas composições de produtos comerciais;

Porto de Armazenagem Avançada: local próximo ao cliente final onde o minério pode ser descarregado e armazenado, aguardando a entrega aos clientes finais. Durante a armazenagem pode ocorrer a mistura de produtos básicos, gerando novas composições de produtos comerciais;

Transporte Ferroviário: meio de transporte usando o sistema ferroviário geralmente utilizado para levar o minério de ferro da mina até o porto onde será embarcado nos navios.

Valor Presente Líquido: teoria financeira em que a mesma quantia em dinheiro vale mais se recebida hoje do que no futuro. O valor do dinheiro reduz com o passar do tempo. O valor total de um projeto é a soma dos resultados obtidos em cada período, trazidos a valor presente, considerando um desconto que aumenta para fluxos com prazos mais distantes.

1.6 Organização do texto

O texto está organizado da seguinte forma. O Capítulo 2 faz uma breve revisão bibliográfica nas áreas de estudo definidas na Seção 1.2 apontando os caminhos a serem seguidos por este trabalho. Partindo da fronteira do conhecimento atual, o Capítulo 3 descreve o modelo e os algoritmos propostos. O Capítulo 4 apresenta os resultados obtidos em um problema teórico, comprovando os ganhos obtidos com as

metodologias propostas. E, para finalizar, o Capítulo 5 faz as análises finais e indica os próximos passos para avanços do conhecimento.

Capítulo 2

Revisão Bibliográfica

Seguindo a linha de raciocínio apresentada na seção 1.1, a revisão bibliográfica irá cobrir três áreas do conhecimento. Inicialmente, na seção *"2.1 Aplicações de PO na cadeia de mineração"*, será feito um mapeamento das aplicações de PO na cadeia de mineração, onde ficará claro até onde já se avançou neste tema. Este levantamento indicará oportunidades na direção da otimização estocástica. Seguindo este caminho, em seguida, na seção *"2.2 Modelos estocásticos aplicados em logística"*, será colocado o foco em problemas de otimização estocástica aplicados à logística. Serão comparadas as aplicações encontradas quando se abre o escopo para todas as cadeias logísticas, com as encontradas na indústria da mineração. Espera-se mostrar claramente as oportunidades existentes relacionadas aos modelos de otimização estocástica para cadeias de suprimentos, e assim identificar as lacunas e possibilidades de avanço na cadeia logística de mineração. Esta análise deixará claro que o desenvolvimento de modelos MDP aplicados à cadeia de mineração é uma oportunidade real de desenvolvimento de pesquisa. Para finalizar, na seção *"2.3 MDP e seus algoritmos de solução"*, dada a complexidade da cadeia logística de mineração e a já comentada *maldição da dimensionalidade* que assombra os MDP, será explorado o estado da arte dos algoritmos e técnicas para solução destes problemas. Será a partir desta fronteira do conhecimento que serão propostos os dois novos algoritmos que ajudarão na solução do problema de otimização estocástica, envolvendo toda a cadeia de mineração, usando MDP.

2.1 Aplicações de PO na cadeia de mineração

Para iniciar o mapeamento das aplicações de PO na cadeia de mineração, será usado como base o trabalho realizado por LEITE *et al.* [1].

Os autores iniciam o trabalho analisando e dividindo a cadeia logística de mineração em quatro elos até o cliente final. Eles apresentam os problemas mais comuns abordados em cada um destes elos.

Minas: englobam tantos os problemas em minas abertas quanto em minas subterrâneas, tais como: modelos de arquitetura de minas (*layout and design models*); problemas de planejamento e produção de minas (*mine production and scheduling problems*); e modelos de alocação de equipamentos operacionais (*operational equipment allocation models*).

Transporte Ferroviário: envolve os problemas de planejamento ferroviário (*train scheduling problems*);

Porto: cobre os modelos de planejamento de pátio de armazenagem (*stockyard planning models*) e os modelos de roteirização de esteiras (*conveyor routing models*) na área portuária;

Navegação marítima: engloba os modelos de planejamento e alocação de navios (*vessel allocation and scheduling models*);

Eles também criam uma quinta categoria (cadeia de suprimentos) para os problemas que envolvem mais de um elo.

Cadeia de Suprimentos: envolve todas as aplicações que englobam mais de um estágio da cadeia logística, tais como: problemas de planejamento de produção da cadeia de suprimentos (*supply chain production, planning and scheduling problems*); problemas de mistura (*blending problems*); e problemas de expansão de capacidade (*capacity expansion problems*).

A Figura 2.1 mostra a ilustração da cadeia com seus elos. Os números entre parênteses mostram a quantidade de artigos pesquisados em cada uma das categorias. É possível perceber que a maioria dos trabalhos se dedica a apenas um elo da cadeia, em particular destaca-se a *mina* com 105 artigos. Apenas 88 pesquisas englobavam mais de um elo da cadeia.

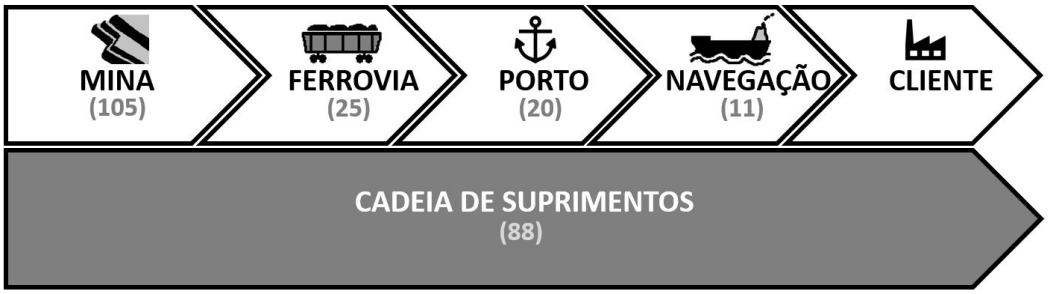

Figura 2.1: Classificação da Cadeia de Suprimentos da Mineração. Fonte: [1]

Em seguida, os autores detalharam o escopo dos 88 trabalhos que englobavam mais de um elo na cadeia e constataram que apenas 1 englobava todos os quatro

elos da cadeia de mineração [24] (Figura 2.2). Este panorama aponta que problemas envolvendo todos os elos da cadeia de suprimentos da cadeia de mineração ainda não são muito estudados e representam uma área a ser explorada.

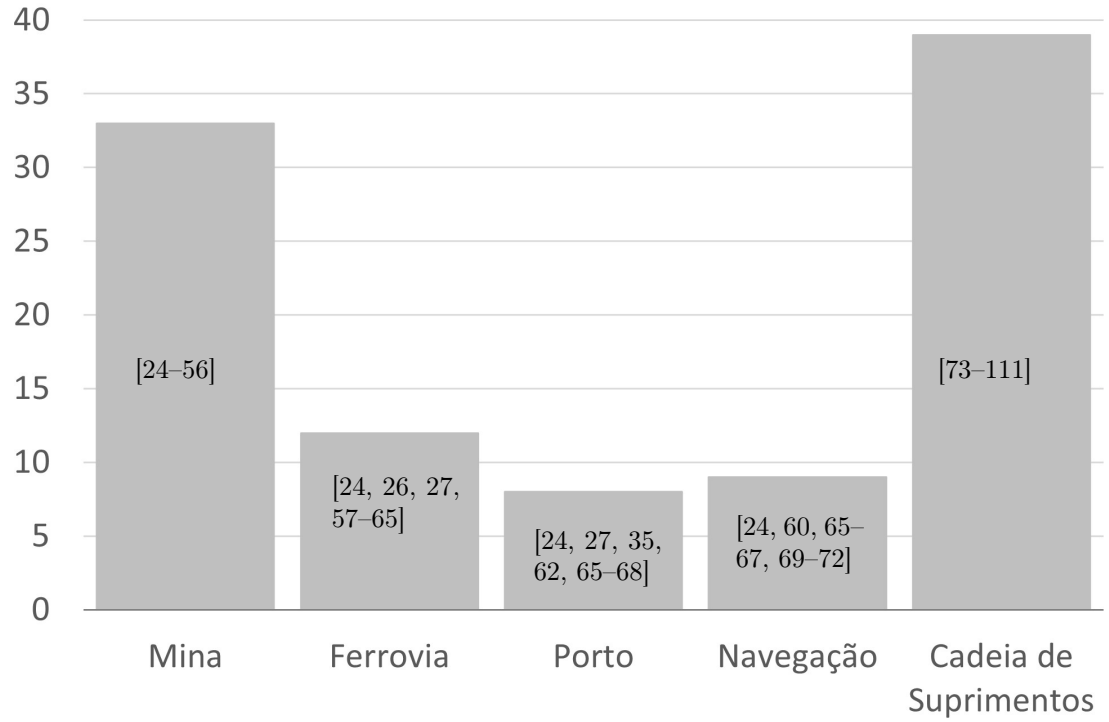

Figura 2.2: Análise dos Artigos que Englobam mais de um Elo da Cadeia. Fonte: [1]

Outro recorte realizado pelos autores explorou a abordagem estocástica ao longo das décadas. Apesar das diversas incertezas presentes na cadeia de mineração, como por exemplo capacidades operacionais, custos de transporte, demanda e preço do minério, a aplicação de modelos estocásticos é relativamente recente, tendo apresentado crescimento nas últimas décadas, conforme mostra a Figura 2.3.

A Figura 2.4 mostra os modelos estocásticos encontrados em [1]. Entre os modelos estocásticos mais usados encontram-se:

- **Simulação Condicional (*Conditional Simulation*):** É mais usada nos problemas envolvendo o planejamento das minas. Nos *modelos de arquitetura de minas* e *nos problemas de planejamento e produção das minas*, o mapeamento do grau de pureza do minério em toda a mina é uma informação fundamental para a otimização do problema. Por outro lado, não é viável realizar a pesquisa geológica em toda a região da mina, ela é feita apenas em alguns pontos espalhados. A simulação condicional é utilizada para inferir a

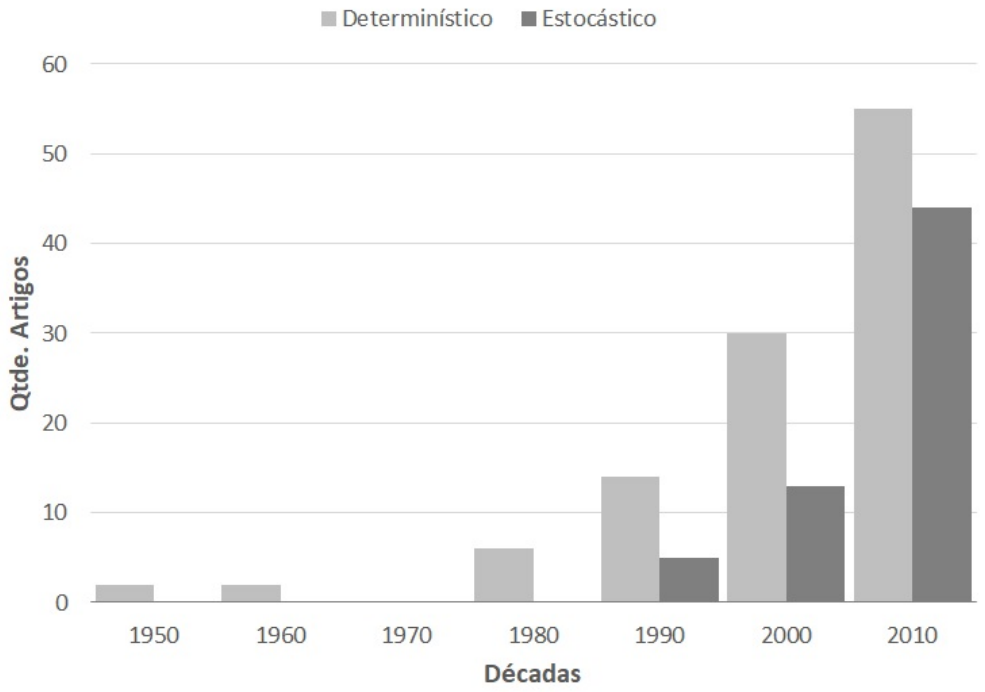

Figura 2.3: Modelos determinísticos e estocásticos aplicados à indústria da mineração. Fonte: [1]

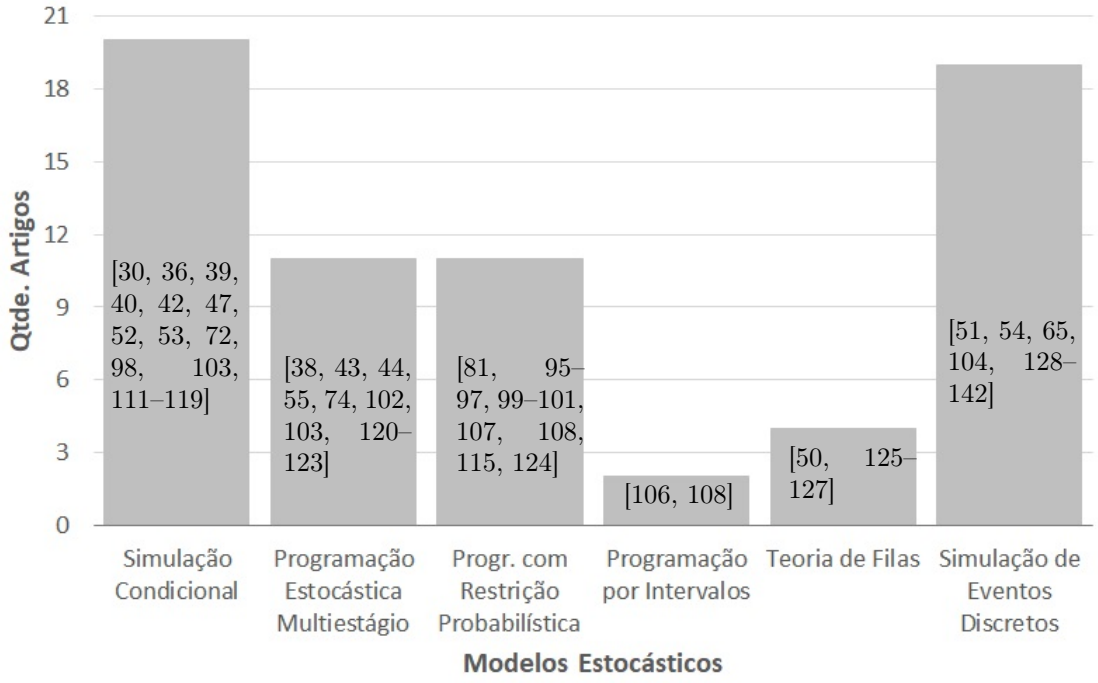

Figura 2.4: Modelos estocásticos na cadeia da mineração. Fonte: [1]

pureza da mina nos pontos não estudados, com base no resultados obtidos nos pontos pesquisados. Com a simulação condicional, busca-se emular a incerteza geológica através da geração de diversos cenários. Conforme a escavação da mina avança e os valores reais são apurados, esses novos dados são utilizados para atualizar a simulação condicional da região ainda não minerada [42, 117].

- **Simulação de Eventos Discretos (*Discrete Event Simulation - DES*):** A maior parte dos trabalhos que fazem uso de DES é encontrada nos *problemas de roteirização de esteiras* nos portos [65, 138]. Isto se deve à variação nas configurações de porto, uma vez que cada sistema de esteiras possui uma configuração própria e única, tornando a montagem de um modelo único adequado a todos os portos inviável. A prática mais comum é o desenho de uma modelagem DES específica para cada porto, onde é possível simular cenários e apurar os tempos de fila e gargalos operacionais.

- **Programação Estocástica de Dois Estágios (*Two-Stage Stochastic Programming - TSSP*) ou Multiestágios (*Multi-Stage Stochastic Orogramming - MSSP*):** São usadas tanto nos *problemas de planejamento e produção de minas*, quanto nos *problemas de planejamento de produção da cadeia de suprimentos* sob incerteza, veja por exemplo [44, 55, 103, 120, 143].

- **Programação Matemática com Restrições Probabilísticas (*Chance-Constraint Programming*):** É bastante usada para tratar o *problema de mistura*, que envolve a elaboração de uma receita apropriada para o produto final a partir das matérias primas disponíveis. Esta abordagem é mais eficiente para problemas com um único período (ou estágio) [115, 124] e enfrenta dificuldades para problemas multi-períodos [108].

- **Programação Linear por Intervalos (*Interval Linear Programming*):** Entre os modelos usados mais raramente, programação linear por intervalos tem a desvantagem de apresentar seus resultados em intervalos [108], dificultando seu entendimento e sua aplicação.

- **Teoria de Filas (*Queuing Theory*):** Também foram encontrados poucos trabalhos envolvendo teoria de filas. A maioria está ligada aos *modelos de alocação de equipamentos operacionais* [50, 125]. Devido à complexidade da cadeia da mineração, é difícil deduzir analiticamente as equações da *teoria de filas* e é mais fácil aplicar a DES.

Complementando a revisão bibliográfica realizada por LEITE *et al.* [1], foram encontrados dois trabalhos envolvendo MDP e a indústria da mineração. ZHAO

et al. [20] utiliza um MDP para resolver um problema de estoque gerido pelo fornecedor (*vendor-managed inventory - VMI*), com horizonte infinito. O autor utiliza o algoritmo de iteração de política. Este modelo é aplicado à cadeia logística de uma empresa de carvão, com um depósito central e quatro armazéns avançados ao longo do Rio Yangtze, na China. Ele não contempla as etapas iniciais da produção (por exemplo: mina, ferrovia, porto) e também não trata o *problema de mistura*. ARCHAMBEAULT [19] utilizou um MDP para a otimização do planejamento de minas, se concentrando na etapa da extração. Em seu modelo, a sequência da extração dos blocos já está pré-definida, cabendo decidir a quantidade a ser extraída, dependendo do preço de mercado no momento e da pureza prevista.

2.1.1 Conclusão

Após esta pesquisa sobre aplicações de PO na cadeia logística de mineração, foi possível constatar que:

1. Devido à complexidade e ao tamanho global da cadeia logística da mineração, há poucos trabalhos cobrindo todos os elos, das minas até os clientes finais;

2. Apesar de todas as incertezas existentes, a aplicação de modelos e algoritmos de otimização estocástica cresceu apenas na última década. Mesmo com este forte crescimento, ainda há diversos problemas não resolvidos que podem ser explorados;

3. Parte do motivo para a predominância da modelagem determinística ocorre em função da complexidade da cadeia da mineração, dificultando a implementação e a solução de problemas estocásticos. Entre os modelos estocásticos já consagrados na literatura, constata-se que o uso de MDP é muito baixo, também se apresentando como uma oportunidade a ser explorada.

Em função destas conclusões, na próxima seção, serão explorados os principais modelos de otimização estocástica aplicados à logística. Confrontando os modelos aplicados à todas as cadeias com os encontrados na cadeia de mineração, espera-se que fique claro as oportunidade existentes a serem exploradas na cadeia de mineração.

2.2 Modelos estocásticos aplicados em logística

Modelos estocásticos encontraram um terreno bastante fértil na logística. Há diversas aplicações de modelos estocásticos envolvendo os mais diversos tipos de problemas logísticos, desde decisões estratégicas, como planejamento de redes, até atividades operacionais do dia-a-dia, como roteirização. Como apontado na seção

anterior, o foco desta tese está apontado para problemas logísticos envolvendo o planejamento de cadeias logísticas com vários elos, tais como [144]: desenho de rede logística (*supply chain network design*) [145–149], planejamento de produção (*production scheduling and planning*) [150, 151] e controle de estoque (*inventory control*) [6, 7, 152, 153]. Por este motivo, dentro de todas as modelagens disponíveis no universo da otimização estocástica [9], serão focados os modelos comumente aplicados aos problemas acima: TSSP [154–157], MSSP [156, 157], MDP [4, 8, 10, 158–160], RL [9, 17] e ADP [9, 18, 161, 162]. Cada um deles apresenta pontos positivos e negativos quando aplicados a problemas de planejamento de cadeias logísticas. Estes pontos serão analisados com maiores detalhes nas próximas sub-seções.

2.2.1 Programação estocástica de dois estágios (*two-stages stochastic programming - TSSP*)

A TSSP, como o próprio nome diz, assume que o problema pode ser dividido em dois estágios, sendo o resultado total a soma dos resultados nos dois estágios. No primeiro, deve-se tomar uma decisão (x), baseada nas informações existentes neste momento (c, A, b e W), que irá influenciar o cenário do segundo estágio. Entretanto, parte das informações do segundo estágio ($q(\xi)$, $T(\xi)$ e $h(\xi)$) depende de uma variável aleatória ($\xi = \xi(\omega)$) que só será conhecida após a decisão do primeiro estágio e antes da decisão do segundo estágio ($y(\omega)$). O modelo clássico de TSSP pode ser escrito como [154–157]:

$$\begin{aligned} \min z = {} & c^T x + \mathbb{E}_\xi\{\min q(\omega)^T y(\omega)\} \\ \text{(s.t.)} \quad & Ax = b \\ & T(\omega)x + Wy(\omega) = h(\omega) \\ & x \geq 0,\ y(\omega) \geq 0 \end{aligned}$$

, onde $\mathbb{E}_\xi\{\cdot\}$ significa o valor esperado com relação à variável aleatória ξ.

Nos modelos de TSSP aplicados ao planejamento de rede logística, é comum colocar o desenho da rede no primeiro estágio (localizações das unidades, investimentos em capacidades, compra de equipamentos e etc.) e a operação (produção e transporte dos produtos até o cliente final) no segundo estágio [163–169]. Nos problemas envolvendo planejamento de produção e gestão de estoques, diversos autores definem o planejamento de produção como variável do primeiro estágio e as ações de correção envolvendo gestão dos estoques, distribuição dos produtos e a apuração dos desvios no segundo estágio [39, 52, 112, 170–179]. É possível misturar TSSP com programação matemática com restrições probabilísticas, quando o planejamento envolve o problema da mistura [167]. Em trabalhos mais recentes, também encontra-se

indicadores ambientais [164–166] e avaliação de risco [166, 173] na função objetivo. Para avaliar os possíveis cenários para o segundo estágio, a técnica mais utilizada é a aproximação do valor esperado da função de custo pela sua média amostral (*sample average approximation*). Enquanto [163, 167–169, 176] utilizam a simulação de Monte Carlo, [179] utilizou a amostragem por hipercubo latino. Como exceções, podemos citar [175], que utiliza a técnica de geração de cenários com o método de preservação dos quatro primeiros momentos (*moment matching*) combinado com a redução de cenários através da seleção de avanço acelerado (*fast foward selection*).

Na cadeia de mineração, a TSSP também é bastante utilizada no planejamento de produção nas minas. Nesta modelagem, é comum que a decisão dos blocos a serem minerados seja tomada no primeiro estágio, enquanto que a minimização dos desvios, em função da incerteza na qualidade do minério, é feita no segundo estágio. Nestes problemas, diversos autores utilizam simulação condicional (*conditional simulation*) para inferir a qualidade do minério em toda a área da mina, usando como base os pontos onde foi feita a amostragem [38, 39, 52, 55, 112, 123, 174]. São encontrados poucos trabalhos usando a TSSP que não estejam focados no planejamento de produção nas minas. Um exemplo é o modelo não linear desenvolvido por ZHANG e DIMITRAKOPOULOS [103, 180], que com uma abordagem mais ampla, define a capacidade dos sistemas de transportes no primeiro estágio e as variáveis operacionais no segundo. Esse modelo é interessante pois abrange a cadeia logística das minas até os clientes finais, os quais são categorizados como clientes de contrato ou clientes *spot*. Também são consideradas diversas incertezas no modelo, tais como: capacidade de produção, custo operacional, produtividade e qualidade nas plantas, qualidade na armazenagem, custo de transporte, preço final de venda e demanda.

Há diversos algoritmos para solução de modelos de TSSP porém o mais encontrado nesta pesquisa foi o algoritmo de decomposição acelerada de Benders (*accelerated Benders decomposition - ABD*) [181]: [163, 164, 169, 178]. Este algoritmo é combinado com relaxação lagrangeana em [164]. Outro algoritmo encontrado foi aproximações externas (*outer approximation algorithm*) [170]. Para acelerar a convergência, alguns autores utilizam diversas heurísticas e meta-heurísticas, tais como: ad hoc [103, 179, 180], busca de colônia de vírus (*virus colony search*) [166], enxame de partículas (*particle swarm*) [38, 55], evolução diferencial (*differential evolution*) [38, 55], heurística de concentração (*heuristic concentration*) [168], proteção progressiva (*progressive hedging*) [123] e recozimento simulado (*simulated annealing - SA*) [38, 39, 55, 112, 174].

De forma resumida, é possível constatar que a TSSP é bastante utilizada em problemas de otimização estocástica envolvendo planejamento logístico, incluindo a cadeia da mineração. O grande motivador da utilização desta modelagem é a facilidade de implementação e solução, principalmente através da ABD, dado que

o problema envolve apenas dois estágios. Entretanto, esta mesma facilidade serve como limitação pois a TSSP não é aplicável em problemas com múltiplos estágios e longos horizontes de tempo.

2.2.2 Programação estocástica de multi-estágios (*multi-stages stochastic programming - MSSP*)

Seguindo a mesma lógica da TSSP e tentando contornar a limitação de apenas dois estágios, a MSSP permite que os problemas tenham diversos estágios ($t \in \{1, 2, \cdots, H\}$) em que deve-se tomar decisões sob incerteza. No primeiro estágio ($t = 1$) parte das informações são conhecidas (c_1, W_t e h_1) quando é tomada a primeira decisão (x_1). A partir do segundo estágio ($t \in \{2, 3, \cdots, H\}$), antes da tomada de decisão ($x_t(\omega_t)$), uma evento aleatório ocorre dando origem à uma variável aleatória ($\xi = \xi(\omega)$) que irá influenciar as informações disponíveis naquele estágio ($c_t(\omega)$, $h_t(\omega)$ e $T_{t-1}(\omega_t)$). Normalmente, são utilizadas árvores de decisão para representar as incertezas em cada estágio do modelo, fazendo com que a quantidade de cenários cresça exponencialmente com o aumento dos estágios, que também pode ser visto na modelagem clássica de MSSP proposta em [156, 157]:

$$\begin{aligned}
\min z = c_1^T x_1 + \mathbb{E}_{\xi_2}\{\min c_2(\omega)^T x_2(\omega_2) + \ldots + \mathbb{E}_{\xi_H}\{\min c_H(\omega)^T x_H(\omega_H)\}\ldots\} \\
\text{(s.t.)} \quad W_1 x_1 = h_1, \\
T_1(\omega_2) x_1 + W_2 x_2(\omega_2) = h_2(\omega), \\
\ldots \vdots \\
T_{H-1}(\omega_H) x_{H-1}(\omega_{H-1}) + W_H x_H(\omega_H) = h_H(\omega), \\
x_1 \geq 0; x_t(\omega_t) \geq 0, \qquad t = 2, \ldots, H
\end{aligned}$$

Usando a modelagem MSSP para problemas de desenho de rede logística, muitos autores colocam as decisões de montagem da infra-estrutura da rede no primeiro estágio e as decisões de operação logística nos demais estágios [182–184]. Entretanto, é possível trabalhar com horizontes mais longos entre os estágios e permitir que as decisões estratégicas ocorram em todos eles [111]. Nos problemas de planejamento de produção e gestão de estoques, os modelos de MSSP permitem que em diversos horizontes de tempo sejam tomadas decisões de planejamento operacional que serão avaliadas após o conhecimento das variáveis aleatórias, normalmente associadas à demanda [185–190].

Há poucas aplicações de modelos de MSSP na indústria da mineração. Provavelmente devido ao aumento exponencial da complexidade. Apenas um artigo aplicou o MSSP no planejmaneto de produção nas minas [191], usando como base o artigo [55]. Os outros dois artigos encontrados já não entram no detalhamento da extração

nas minas. ARIGONI [102] utilizaram a MSSP para otimizar o custo de compra de carvão em uma usina termo-elétrica norte americana. Cada estágio representa um ano de operação e as decisões são operacionais (volumes de compra, de produção e de estoques), considerando incerteza no preço do carvão. REUS *et al.* [143] construíram um modelo que, além da aleatoriedade do preço, também considera a possibilidade de acidentes e incidentes.

Assim como na TSSP, alguns autores trabalham com simulação Monte Carlo [102, 182] ou hipercubo latino [183] para gerar os possíveis cenários. Também há diversos trabalhos que utilizam o algoritmo de ABD [183, 184]. Além disso, foram encontrados algoritmos de relaxação Lagrangeana [111] e metaheurísticas tais como: busca variável de vizinhança (*variable neighborhood search - VNS*) [188, 191], MSDV (multi-objetivo baseado no algoritmo de evolução diferencial auto-adaptativa - *self-adaptive differential evolution*) [188], programação dinâmica estocástica (*stochastic dynamic programming*) [187] e SA [191]

O algoritmo de programação dinâmica dual estocástica (*Stochastic Dual Dynamic Programming - SDDP*) desenvolvido por PEREIRA e PINTO [192] vem obtendo ótimos resultados nos problemas de planejamento energético, até para horizontes mais longos [192–196]. O SDDP aproxima as funções de custo esperado da programação dinâmica estocástica (*stochastic dynamic programming*) através de funções lineares por partes, obtidas através das soluções duais dos problemas de otimização em cada estágio. O uso do SDDP em logística não é tão difundido [190, 197] e foi localizado apenas um trabalho aplicando este algoritmo à cadeia logística de mineração [143]. Apesar de todos os ótimos resultados obtidos até o momento, o SDDP apresenta duas limitações importantes para que haja garantia da otimalidade global do resultado: sua função objetivo precisa ser convexa e suas variáveis de decisão devem ser contínuas, não podendo ser binárias ou inteiras. Recentemente, foi proposta uma alternativa para sua aplicação em problemas com variáveis binárias (*Stochastic Dual Dynamic integer Programming - SDDiP*) [198]. Os resultados apontados pelos autores são bastante promissores, mas ele ainda precisa ser testado e diferentes tipos de problemas.

Pode-se concluir que a grande dificuldade em se trabalhar com modelos de MSSP é o aumento exponencial de cenários relacionados ao aumento de estágios. Desta forma, é difícil encontrar exemplos de aplicação em problemas com longos horizontes de tempo ou com horizontes infinitos. A Tabela 2.1 mostra a quantidade de estágios utilizadas nos trabalhos pesquisados. Os trabalhos com maior horizonte utilizam o SDDP, porém este apresenta a limitação da função objetivo ser convexa e as variáveis necessitarem ser contínuas.

Qtd. Estágios	Trabalhos
3	[185, 188]
4	[183, 186]
5	[102]
7	[111]
8	[182]
9	[187]
10	[192]
12	[197]
15	[143]
20	[191]
30	[189]
50	[190]
60	[194, 196]
84	[195]

Tabela 2.1: MSSP: Quantidade de estágios nos artigos pesquisados

2.2.3 Processos de decisão markovianos (*Markov decision processes - MDP*)

Processos de decisão markovianos é uma abordagem desenvolvida especialmente para a solução de problemas com decisões sequenciais, em especial aqueles envolvendo um longo horizonte de tempo, onde busca-se uma política de ações, baseadas no estado atual do sistema, que otimize a sua resposta, considerando os ganhos presentes e futuros [4]. Considere um sistema dinâmico com tempo discreto, podendo ser infinito ($t \geq 0$ e $t \in T$, onde T é o conjunto de todos os períodos possíveis). Em cada período t, o tomador de decisão observa o estado do sistema ($s_t = s \in S$, onde S é o conjunto de todos os estados possíveis para o sistema) e escolhe uma decisão ($a_t = a$), dentre o conjunto de decisões possíveis para o estado s ($A(s)$), que está contido no espaço de todas as decisões (A): $a_t = a \in A(s) \subset A$.

Ao aplicar a ação a, o sistema sofre uma transição dependente da variável aleatória ξ_t, que representa as incertezas do modelo, do estado s para um novo estado s' ($s_{t+1} = s' \in S$), cuja probabilidade é $p(s'|s,a)$ independente de t. Por ser uma função de probabilidade indexada, tem-se que $p : S \times A \times S \to [0,1]$ e $\sum_{s' \in S} p(s'|s,a) = 1,\ \forall s \in S$ e $\forall a \in A(s)$. Desta forma, podemos descrever o estado seguinte através da seguinte função de transição:

$$s_{t+1} = f(s_t, a_t, \xi_t). \tag{2.1}$$

Além da transição para o estado s', ao se aplicar a ação a estando no estado s, em cada período é obtida uma recompensa $r(s,a,s')$, determinada pela função $r : S \times A \times S \to \mathbb{R}$, com limites inferiores e superiores e independente de t. É

possível calcular a recompensa esperada dependente apenas de s e a:

$$r(s,a) = \sum_{s' \in S} r(s,a,s')p(s'|s,a). \tag{2.2}$$

Para um MDP com horizonte finito, $r_N(s_{t=N})$ é a recompensa final ao atingir o estado s_t no último período ($t = N$).

Estando o sistema no estado s, a função de custo ($V(s)$), que será maximizada, é construída usando as recompensas esperadas obtidas em cada período ($r(s,a)$). Considerando problemas com horizonte infinito, há três principais formas de se avaliar a função de custo, sendo a *recompensa total descontada esperada* o critério de otimalidade mais aplicado e estudado [4].

1. *Recompensa Total Esperada*: Representa a soma de todas as recompensas obtidas até $t = \infty$ períodos de tempo, conforme a Equação (2.3). Em alguns casos, esse limite pode não existir. Quando o limite existe, é possível calcular $V(s)$ conforme a Equação (2.3).

$$V(s) \equiv \lim_{N \to \infty} \mathbb{E}_s \left\{ \sum_{t=1}^{N} r(s_t, a_t) \right\}. \tag{2.3}$$

2. *Recompensa Total Descontada Esperada*: Representa o cálculo do valor presente de todas as recompensas esperadas até $t = \infty$ períodos de tempo, considerando um fator de desconto $0 \leq \lambda < 1$ por período. É calculada conforme a Equação (2.5). Se $r(s,a)$ for limitado conforme a Equação (2.4), o limite da Equação (2.5) existe e é possível calcular $V(s)$ conforme a Equação (2.6).

$$\sup_{s \in S;\, a \in A} |r(s,a)| < \infty, \tag{2.4}$$

$$V(s) \equiv \lim_{N \to \infty} \mathbb{E} \left\{ \sum_{t=1}^{N} \lambda^{t-1} r(s_t, a_t) \right\}, \tag{2.5}$$

$$V(s) = \mathbb{E} \left\{ \sum_{t=1}^{\infty} \lambda^{t-1} r(s_t, a_t) \right\}. \tag{2.6}$$

3. *Ganho ou recompensa médios*: Representa a recompensa média quando o sistema atinge o estado de equilíbrio, conforme a Equação (2.7), quando esta equação tem solução:

$$V(s) \equiv \lim_{N \to \infty} \frac{1}{N} \mathbb{E} \left\{ \sum_{t=1}^{N} r(s_t, a_t) \right\}. \tag{2.7}$$

Como já foi dito inicialmente, o objetivo final do MDP é obter uma regra de decisão para todos os períodos (d_t) que maximize a *função de custo*. Para sistemas *markovianos*, nos quais as probabilidades de transição e as recompensas dependem exclusivamente do estado atual s, não dependendo de t ou de estados anteriores s_{t-n}, é provado que as regras de decisão determinísticas e dependentes exclusivamente do estado s são capazes de maximizar o sistema ($d : S \rightarrow A$) [4, 10]. Definindo *política* ($\pi = \{d_{t=1}, d_{t=2}, \ldots\}$) com uma sequência de regras de decisão e Π o conjunto de todas as políticas disponíveis, o sistema pode ser maximizado através de uma *política de controle estacionário* $\pi \in \Pi$, $\pi = \{d, d, \ldots\}$ e $\pi_t(s) = \pi(s) = d(s)$.

Considerando a função de custo como a *Recompensa Total Descontada Esperada* para problemas com horizonte infinito, o seu valor, aplicando-se a política π ($V^{\pi}(s)$), pode ser calculado como:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left\{\sum_{t=0}^{+\infty} \lambda^t r(s_t, \pi(s_t)) | s_0 = s\right\}, \forall s \in S. \tag{2.8}$$

O objetivo final do MDP é achar a política $\pi^* \in \Pi$ que maximiza a função de custo para cada estado $s \in S$, obedecendo as Equações (2.1) e (2.2):

$$V^*(s) = V^{\pi^*}(s) = \max_{\pi \in \Pi} V^{\pi}(s), \forall s \in S.$$
$$\pi^* = \arg\max_{\pi \in \Pi} V^{\pi}(s), \forall s \in S.$$

Analisando a Equação (2.8), ela pode ser reescrita como:

$$V^{\pi}(s) = r(s, \pi(s)) + \lambda \mathbb{E}_{\pi}\left\{\sum_{t=1}^{+\infty} \lambda^{t-1} r(s_t, \pi(s_t)) | s_1 = s'\right\}, \forall s \in S. \tag{2.9}$$

Entretanto, pela própria Equação (2.8) tem-se que:

$$\mathbb{E}_{\pi}\left\{\sum_{t=1}^{+\infty} \lambda^{t-1} r(s_t, \pi(s_t)) | s_1 = s'\right\} = V^{\pi}(s'), \forall s' \in S. \tag{2.10}$$

Substituindo (2.10) em (2.9), chega-se a equação recursiva de Bellman:

$$V^{\pi}(s) = r(s, \pi(s)) + \lambda V^{\pi}(s'), \forall s \text{ e } s' \in S. \tag{2.11}$$

Utilizando a notação vetorial e considerando que:

- v^{π} (vetor da função de custo): vetor com dimensão $|S|$, em que cada elemento representa o valor esperado da função de custo para cada estado, aplicando-se a política π;

- r^{π} (vetor de recompensas): vetor com dimensão $|S|$, em que cada elemento representa a recompensa esperada em $t+1$ para cada estado, aplicando-se a politica π;

- P^{π} (matriz de probabilidades de transição): matriz com dimensão $|S| \times |S|$, em que cada elemento representa a probabilidade de transição do estado $s = i$ para os estado $s' = j$, aplicando-se a politica π, onde i são as linhas e j as colunas da matriz.

- I: matriz identidade.

Continuando a trabalhar com a *Recompensa Total Descontada Esperada*, é possível escrever a equação de Bellman de forma vetorial como [4]:

$$v^{\pi} = r^{\pi} + \lambda P^{\pi} v^{\pi} \tag{2.12}$$

Sendo possível isolar o vetor da função de custo:

$$v^{\pi}(I - \lambda P^{\pi}) = r^{\pi} \tag{2.13}$$

$$v^{\pi} = (I - \lambda P^{\pi})^{-1} r^{\pi} \tag{2.14}$$

Esses resultados (equações (2.12) e (2.14)) são bastante utilizados no principais algoritmos para resolver MDP com *Recompensa Total Descontada Esperada.*

Apesar do sucesso obtido em diversas aplicações, o MDP possui algumas limitações, tais como:

- Intervalo de tempo discreto: em sua modelagem original, os MDP assumem intervalos de tempo discretos e com mesmo tamanho. Entretanto, através dos *semi-Markov decision processes - SMDP* é possível modelar o tempo continuamente e as decisões podem ser tomadas toda vez que houver uma mudança de estado no sistema. Os tempos entre mudanças no sistema são variáveis aleatórias que dependem do estado e das ações tomadas.

- Ações e estados discretos: os espaços de estados e ações não são contínuos. A metodologia mais comum para adaptar espaços contínuos é a discretização. Para melhorar a precisão, utiliza-se intervalos menores, aumentando a quantidade de estados e ações discretas. Esta solução, por outro lado, acaba levando à *maldição da dimensionalidade* (ver abaixo). Outra alternativa é modelar esses espaços de estados e ações através de funções aproximadas. Essa metodologia é conhecida como programação dinâmica aproximada (*approximate dynamic programming - ADP*), que será tratada com mais detalhes na Seção 2.2.5.

- *Maldição da dimensionalidade*: Os principais algoritmos de solução dos MDP têm seus tempos computacionais aumentados exponencialmente quando os espaços de estados e ações aumentam. Este fato, torna-se a principal barreira para aplicação de MDP em problemas com grande complexidade. Até hoje, buscam-se algoritmos que ajudem a contornar esse problema. Na Seção 2.3 será feita uma revisão do progresso já obtido até hoje.

Maiores detalhes sobre MDP e SMDP, incluindo suas limitações, podem ser obtidos em [4, 8, 10, 158–160].

Voltando para o foco em problemas de otimização estocástica em cadeias de suprimentos, MDP podem ser aplicados em diversos problemas. Os mais recorrentes, e que serão objeto de foco neste trabalho por envolverem mais de um elo da cadeia, são aqueles relacionados à: gestão de estoques e planejamento de produção. É possível encontrar revisões bibliográficas específicas sobre estes temas [199, 200]. Apesar de não ser o foco desta tese, há aplicações de MDP em outros tipos de problemas envolvendo logística, tais como: consolidação de carga [201], terceirização de logística reversa[202] e definição de políticas de preços [203, 204].

No planejamento de produção, utiliza-se MDP para decidir as próximas ações dos equipamentos produtivos, baseado nos estoques, nas filas e no que está sendo fabricado no momento da decisão, considerando as incertezas na demanda e no próprio processo produtivo. YIH e THESEN [205] aplicam a modelagem SMDP para definir o deslocamento dos produtos através de esteiras entre as estações de trabalho. ARRUDA e DO VAL [161] definem as ações de uma única máquina, que trabalha em lotes, sendo capaz de realizar todas as etapas na fabricação de diversos produtos. Na indústria da mineração, conforme já citado na Seção 2.1, ARCHAMBEAULT [19] utilizou um MDP para a otimização do planejamento de produção das minas. Seu modelo, além de cobrir apenas a etapa de extração do minério, considera que a sequência dos blocos já está pré-definida, tratando apenas a decisão de produção ou postergação dos mesmos.

Entre todos os problemas logísticos, MDP é bastante aplicado na gestão de estoques, apresentando ótimos resultados. Na modelagem mais básica, a variável de estado é composta pelo nível do estoque de cada produto em cada elo da cadeia. A decisão a ser tomada é a compra, ou fabricação, de novos produtos considerando a incerteza na demanda. Além da incerteza na demanda, também é possível considerar outras variações tais como a chegada de produtos e perecibilidade [206, 207]. A recompensa engloba o custo de aquisição, ou de fabricação, de estocagem e a receita de venda. Em seu livro clássico, PUTERMAN [4] já apresenta um exemplo simples de gestão de estoques. Dado a eficácia da aplicação de MDP para esse tipo de problema, diversos autores usam-no para provar a otimalidade de políticas de estoques, em especial a (s, S) (onde s é o nível do estoque mínimo para a efetivação

de um pedido e S o nível de estoque máximo admitido). FEINBERG e LEWIS [208] provam a otimalidade desta política para problemas que incluem pedidos pendentes e FLEISCHMANN e KUIK [209] para problemas com possibilidade de retorno de mercadorias. Ainda analisando a política (s, S), BIJVANK *et al.* [210] provam que ela não é ótima quando há vendas perdidas e o autor propõe uma política (s, S) modificada. Como citado anteriormente, um dos principais problemas enfrentados pelos MDP quando aplicado a cadeias mais complexas é a *maldição da dimensionalidade*. Além da busca por algoritmos mais eficientes (que será visto na Seção 2.3), outra alternativa encontrada é trabalhar na modelagem. Em vez de considerar a quantidade total de produto como estado do sistema, SUN *et al.* [211] dividem esta quantidade em duas variáveis: a quantidade de referência, calculado no período anterior, e a quantidade adicional. Trabalhando com um exemplo da indústria siderúrgica, os autores provam que a metodologia usada reduz o espaço de estados, acelerando a convergência do modelo. Na indústria da mineração, conforme já citado na Seção 2.1, ZHAO *et al.* [20] utilizaram um MDP para resolver um problema de VMI na cadeia de suprimentos de carvão com dois elos (depósito central e subsidiárias).

Como é possível observar, apesar de ter bastante sucesso no planejamento de produção e na gestão de estoques, os trabalhos com MDP normalmente são aplicados a cadeias de suprimentos pequenas com poucos elos. A principal causa desta limitação é a *maldição da dimensionalidade*, dado que o aumento da complexidade da cadeia logística significa aumento dos espaços de estados e ações. Algoritmos que ajudem a solucionar esse problema contribuirão na difusão e popularização do MDP. Na Seção 2.3 serão abordados os principais algoritmos usados para solução de MDP, sinalizando os avanços já obtidos e suas limitações.

2.2.4 Aprendizado por reforço (*reinforcement learning - RL*)

Seguindo a modelagem de decisões sequenciais utilizada no MDP para maximização da *recompensa total descontada esperada*, o RL sugere uma outra abordagem para a solução. Em vez de utilizar a Equação de Bellman (2.11) para atualizar a função de custo para todos os estados em cada iteração, o RL propões que o aprendizado ocorra a medida que o agente interaja com o meio-ambiente. Desta forma, o RL consegue duas importantes reduções na análise do espaço de estados. Primeiramente, em cada iteração, apenas o valor do estado atual é atualizado. O algoritmo aposta que os estados mais importantes serão os mais visitados, e por isso terão uma aproximação mais acurada para suas funções de custo. A segunda vantagem é que ele não trabalha com todas as transições possíveis partindo do estado s e tomando a ação a, pois utiliza apenas a informação da transição observada para o estado s'. A solução obtida pelo RL é uma solução aproximada, já que apenas os estados visitados são

atualizados. Como o aprendizado é feito através de tentativa e erro interagindo com o próprio sistema, o RL é bastante aplicado quando as probabilidades de transição ou as recompensas não são conhecidas previamente [9, 17].

No RL, o cálculo da função de custo através da Equação de Bellman (2.11) é substituído pelos fatores Q (*Q-factors*). Os fatores Q ($Q(s,a)$) aproximam o custo de estar no estado s e tomar a ação a ($Q : S \times A \to \mathbb{R}$):

$$Q(s,a) := (1-\alpha)Q(s,a) + \alpha \left\{ r(s,a,s') + \lambda \max_{a \in A(s')} Q(s',a) \right\}$$

onde α é a taxa de aprendizagem ($0 < \alpha \leq 1$).

Uma das aplicações de maior sucesso da técnica de aprendizado por reforço (RL) é no planejamento de produção, mais especificamente no planejamento dinâmico de estações de trabalho (*dynamic job shop scheduling*). Assim como no MDP, as variáveis de estado normalmente englobam os estoques, as filas de espera e o que está sendo produzido no momento. Em quase todos os trabalhos, a chegada dos novos pedidos é incerta, porém outras variações como a quebra de equipamentos [212] podem ser inseridas. Os problemas mais simples envolvem uma única estação de trabalho [213] e os mais complexos modelam diversas estações que realizam etapas do processo produtivo [214, 215]. Seguindo uma modelagem similar, o RL também pode ser usado no planejamento de serviços [216].

Dada sua modelagem semelhante ao MDP e seu sucesso na solução de problemas mais complexos, RL também é bastante aplicado a problemas de gestão de estoque em cadeias de suprimentos com vários elos [217–219]. Nesses problemas, normalmente a variável de estado é a quantidade em estoque em cada componente da cadeia, mas outras informações podem ser consideradas, tais como o tempo de validade dos produtos [219]. A decisão quase sempre encontrada é a quantidade de produto a ser pedida para repor seu estoque, em cada elo, contudo é possível incluir decisões sobre o meio de transporte [218]. Além da incerteza na demanda, outras variações podem ser consideradas tais como: taxa de câmbio [218], tempo de produção [218, 220] e tempo de transporte [218, 220]. Assim como ocorre com o MDP, alguns autores utilizam o RL para avaliar políticas de estoque, obtendo sucesso em cadeias maiores [220–222].

Apesar dos esforços, não foram encontrados trabalhos de RL no planejamento de cadeia logística da indústria da mineração. Em compensação, foram encontrados trabalhos na indústria de óleo e gás, que também possui uma cadeia de suprimentos complexa e global. Além de modelar a gestão de estoques, é comum também a otimização do planejamento de produção, o que causa um aumento na complexidade dos modelos. [223] propôs uma abordagem que combina simulação baseada em agentes e aprendizado por reforço (*learning-agent-based model*), englobando toda a

cadeia de valor. Neste modelo, cada agente da cadeia avalia seu estado baseado nas informações dos agentes em sua vizinhança. Com base nessas informações, o mesmo agente toma decisões que impactarão o sistema e os agentes vizinhos. Essa abordagem foi utilizada também para o controle de redes de filas em [224]. Já em [225] o modelo combina decisões estratégicas (compra de petróleo bruto) e operacionais aplicadas a uma refinaria. As primeiras são obtidas a partir de um modelo de MDP, ao passo que as últimas resultam de um modelo de MSSP que opera numa escala de tempo mais rápida. As variáveis de estado do MDP são os níveis de estoque e de preços e as incertezas estão nos preços do óleo cru e dos produtos acabados.

Resumindo, é possível observar que os algoritmos de RL são bastante utilizados em problemas complexos envolvendo cadeias logísticas com vários elos. Seu sucesso se deve à obtenção de boas soluções para problemas computacionalmente intratáveis por meio de algoritmos clássicos de MDP. Entre as desvantagens dos algoritmos de RL estão a solução sub-ótima e a necessidade de um período de testes para aprendizado. Esse período de testes exige que no início sejam testadas soluções aleatórias não-ótimas. Somente após um período de aprendizado é que o modelo passa a fornecer boas respostas.

2.2.5 Programação dinâmica aproximada (*approximate dynamic programming - ADP*)

Como mencionado na Seção 2.2.3, os modelos de MDP enfrentam dificuldades quando o número de estados, a quantidade de ações e o número de possíveis transições crescem. Uma das alternativas para tentar contornar este problema é utilizar funções paramétricas aproximadas. O ADP continua utilizando a modelagem MDP para problemas com decisões sequenciais, porém utiliza funções paramétricas aproximadas para:

- Função de custo: em vez de atualizar a função de custo para cada estado ($V(s)$), estima-se uma aproximação ($\hat{V}(s, \mathbf{w})$) onde atualiza-se apenas o vetor de parâmetros ($\mathbf{w}$).

$$\hat{V}(s, \mathbf{w}) \approx V^{\pi}(s) \tag{2.15}$$

  onde $\mathbf{w} \in \mathbb{R}^d$ é o vetor de pesos da função de custos paramétrica aproximada.

- Política: em vez de atualizar a política para cada estado ($\pi(s)$), estima-se uma aproximação ($\pi(a|s, \boldsymbol{\theta})$) para a probabilidade da ação a ser tomada dado que o sistema está no estado s, e atualiza-se apenas o vetor de parâmetros ($\boldsymbol{\theta}$).

$$\pi(a|s, \boldsymbol{\theta}) = p\left\{a_t = a | s_t = s, \boldsymbol{\theta}_t = \boldsymbol{\theta}\right\} \tag{2.16}$$

onde $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ é o vetor de pesos da função de política paramétrica aproximada.

Com as funções paramétricas aproximadas, reduz-se o esforço computacional pois apenas os vetores de pesos precisam ser atualizados $(\mathbf{w}, \boldsymbol{\theta})$ [9, 18, 161, 162].

Dados os bons resultados obtidos para problemas mais complexos, há diversas aplicações de ADP para problemas de planejamento de produção e serviços. Como utiliza a mesma modelagem MDP, a variável de estado normalmente engloba a situação das estações de trabalho e as filas de atendimento. A incerteza usualmente está na demanda [216, 226–230].

Na gestão de estoques, a modelagem tradicional continua considerando os níveis de estoque e os produtos em trânsito como as principais variáveis de estado utilizadas para a tomada de decisão dos pedidos de reposição de estoques, com incerteza na demanda [231, 232]. Também não foram encontradas aplicações de ADP na indústria da mineração. Na indústria de óleo e gás, foi encontrada uma aplicação otimizando o fornecimento global de petróleo [233].

Foi possível observar que ADP é uma alternativa para a *maldição da dimensionalidade*, a qual pode ser aplicada a problemas bastante complexos de forma a produzir boas soluções sub-ótimas em tempos computacionais razoáveis. O ponto negativo dessa metodologia é exatamente o fato de que as soluções fornecidas são sub-ótimas, dado que ela utiliza aproximações da função de custo ou de política. Ademais, como é difícil atestar a qualidade das aproximações à priori, também não se pode garantir que a solução obtida seja de boa qualidade.

2.2.6 Conclusão

Analisando a aplicação de modelos estocásticos em cadeias de suprimentos, principalmente nos problemas de planejamento de produção e gestão de estoques, é possível concluir que:

1. O TSSP é bastante utilizado porém tem como grande limitador trabalhar apenas com dois estágios;

2. Apesar de resolver a limitação dos dois estágios, o MSSP tem sua complexidade bastante aumentada para longos horizontes de tempo. O SDDP consegue contornar esse aumento de complexidade, mas apenas para problemas com função objetiva convexa e variáveis contínuas;

3. O MDP possui bastante sucesso, principalmente considerando longos horizontes de tempo. Entretanto, perde eficiência computacional quando aplicado a cadeias grandes e complexas.

4. Também usando a modelagem MDP para problemas com decisões sequenciais, tanto o RL quanto a ADP são soluções bastante aplicadas em cadeias complexas. A grande desvantagem dessas abordagens são suas soluções sub-ótimas.

Comparando esses algoritmos com a realidade da indústria da mineração, percebe-se que seus longos horizontes de tempo, bem como sua natureza de decisões sequenciais, indicam a utilização da abordagem MDP, que pode ser resolvida com seus algoritmos tradicionais ou RL ou ADP. Esses algoritmos serão aprofundados na próxima seção. Espera-se mapear até onde já se avançou para contornar a *maldição da dimensionalidade* e ter uma indicação de novos caminhos possíveis, tornando mais eficiente a aplicação do MDP à cadeia de mineração.

2.3 MDP e seus algoritmos de solução

Como já foi dito anteriormente (seções 1.1 e 2.2.3), MDP são bastante utilizados em problemas de gestão de estoques para modelar decisões de produção e de estocagem (*inventory management problems*). Entretanto, modelos MDP perdem eficiência computacional a medida que aumenta a quantidade de estados e ações. Nesta seção, serão estudados os algoritmos existentes para se obter a solução ótima em problemas com horizonte infinito. Espera-se que essa análise abra caminho para novas oportunidades de melhoria, viabilizando a aplicação de MDP em cadeias complexas como a da mineração. Como também já foi informado na Seção 2.2.3, o função de custo mais usada é a *recompensa total descontada esperada* [4], sendo assim, as avaliações dos algoritmos serão concentradas no uso dessa função.

2.3.1 Iteração de política (*policy iteration - PI*)

O algoritmo de iteração de política rivaliza com o de iteração de valor pelo posto de algoritmo clássico para a solução de MDP. No algoritmo de iteração de política, inicia-se com a escolha de uma política qualquer (π_n), onde n representa a iteração percorrida pelo algoritmo. Em seguida, obtém-se a função de custo dessa política nesta iteração n (v_n) por meio do passo de avaliação de política, baseado na Equação (2.14). Utilizando essa atualização da função de custo, obtém-se então uma nova politica através de um passo de melhoria de política, que escolhe a ação que maximiza a recompensa total a partir de cada estado $s \in S$, Equação (2.17). Retorna-se então ao passo de avaliação de política se o critério de parada não for satisfeito. O algoritmo termina quando a função de custo da nova política coincide com aquela da política anterior, o que significa que a política ótima foi encontrada. O pseudocódigo do procedimento está detalhado no Algoritmo 1 abaixo.

Algoritmo 1: Iteração de Política

Entrada: Defina $n = 0$ e escolha $\pi_0 \in \Pi$.

Saída: A solução ótima para o vetor da função de custo v^*, e a política ótima estacionária π^*

1 **repita**

Passo de avaliação de política

2 Calcule v_n:

$$(I - \lambda P^{\pi_n})v_n = r^{\pi_n}$$

Passo de melhoria de política

3 Escolha π_{n+1} (sendo possível, escolha $\pi_{n+1} = \pi_n$):

$$\pi_{n+1} \in \arg\max_{\pi \in \Pi} \{r^{\pi} + \lambda P^{\pi} v_n\} \tag{2.17}$$

$n \leftarrow n + 1$

4 **até** $\pi_n = \pi_{n-1}$;

5 Defina $\pi^* = \pi_n$ e $v^* = v_n$

6 **retorna** v^*, π^*

Buscando melhorar a performance, principalmente o tempo computacional, alguns autores desenvolveram variações para o algoritmo PI. CHANG [21] considera um conjunto de políticas, em vez de apenas uma, a cada passo de avaliação de política. O conjunto contém a melhor política calculada na iteração anterior, além de políticas sorteadas aleatoriamente. Desta forma, o autor introduz uma possibilidade de melhoria de política no passo que originalmente fazia apenas avaliação. O algoritmo foi denominado de iteração de conjunto de políticas (*policy set iteration - PSI*). Buscando facilitar a avaliação da política e evitando o cálculo de inversão de matrizes, BERTSEKAS [234] utiliza um número finito de iterações de valores (ver Seção 2.3.2) nesta etapa, criando um algoritmo modificado chamado de iteração de política-λ (λ-*policy iteration*).

Entre os trabalhos pesquisados aplicando MDP em cadeias logísticas, é possível encontrar alguns poucos exemplos de aplicação do algoritmo de PI. ARCHAMBEAULT [19] aplica o algoritmo de PI original. YIH e THESEN [205] utilizam um procedimento empírico para reduzir o tamanho dos estados e da matriz de probabilidades de transição. Os autores fazem a redução consultando pessoas experientes no problema modelado e considerando apenas os estados e transições mais relevantes. ZHAO *et al.* [20] utilizam um procedimento de eliminação de ações na etapa de busca pela nova política ótima, após a avaliação da política anterior.

O grande problema do PI é o cálculo da função de custo por meio de inversão de matriz, passo 2 do Algoritmo 1. Esse cálculo consome muito tempo computacional e faz uso extensivo de memória, sendo inviável para problemas muito complexos. Por esse motivo não são encontradas muitas aplicações deste algoritmo puro e alguns

autores usam procedimentos para redução da complexidade do problema, mesmo que assim percam a garantia da solução ótima.

Dentre as variações propostas para o PI, o PSI apresenta uma alternativa interessante na tentativa de redução da quantidade de passos de melhoria de política, e consequentemente reduzindo a busca no espaço de ações. Melhorias no passo de avaliação modificado, que aumentem a quantidade de políticas testadas sem comprometer o tempo computacional, podem apresentar bons resultados e indicam o caminho a ser seguido neste trabalho.

2.3.2 Iteração de valor (*value iteration - VI*)

O primeiro autor a desenvolver o algoritmo de VI foi BELLMAN [8]. O algoritmo usa a Equação (2.18), baseada na Equação de Bellman na forma vetorial (2.12), para atualizar o cálculo da função de custo (v_n) para a iteração n. O pseudo-código para o VI está detalhado no Algoritmo 2. O algoritmo é interrompido quando a modificação na função de custo é considerada insignificante, passo 3 do Algoritmo 2.

$$v_{n+1} := \max_{\pi \in \Pi} \{r^{\pi} + \lambda P^{\pi} v_n\} \tag{2.18}$$

Algoritmo 2: Iteração de Valor

Entrada: Defina $n = 0$ e escolha v_0 e $\varepsilon > 0$.

Saída: A solução ótima para o vetor da função de custo v^*, e a política ótima estacionária π^*

1 **repita**

2 Para cada $s \in S$, calcule $v_{n+1}(s)$ conforme:

$$v_{n+1}(s) := \max_{a \in A_s} \left\{ r(s,a) + \sum_{s' \in S} \lambda p(s'|s,a) v_n(s') \right\} \tag{2.19}$$

$n \leftarrow n + 1$

3 **até** $\|v_n - v_{n-1}\| < \varepsilon$;

4 Para cada $s \in S$, escolha:

$$\pi(s) \in \arg\max_{a \in A_s} \left\{ r(s,a) + \sum_{s' \in S} \lambda p(s'|s,a) v_n(s') \right\}$$

5 Defina $\pi^* = \pi$ e $v^* = v_n$

6 **retorna** v^*, π^*

Também buscando a otimização do tempo computacional, vários autores propuseram variações para o VI. Em 2001, [235] utilizou a busca heurística com iteração

de valor (ou iteração de política) como alternativa para solucionar MDP. Em vez de buscar a melhor política para todos os estados, a busca heurística otimiza um estado inicial específico, desconsiderando os estados que não podem ser atingidos a partir deste estado inicial. O autor baseou seu algoritmo em duas buscas heurísticas: A* (quando a solução tem a forma de um grafo acíclico ou um caminho) e OA* (quando a solução tem a forma de uma árvore de decisão). O algoritmo desenvolvido (LAO*) incorpora a solução através de ciclos e utiliza a programação dinâmica na etapa de atualização dos custos e das ações. Os autores mostram que o LAO* pode ser utilizada com sucesso em MDP.

Uma outra variação desenvolvida foi o algoritmo de iteração de valor com informação parcial (*partial information value iteration algorithm - PIVI*), apresentado em [236, 237]. A ideia por trás do algoritmo é usar modelos cuja acurácia cresça conforme as iterações avancem. Ou seja, no início, quando ainda está longe da solução, o algoritmo utiliza uma modelagem simplificada com o objetivo de reduzir o tempo computacional. Conforme as iterações vão avançando, o algoritmo aumenta a complexidade do modelo, gradualmente caminhando para a modelagem completa.

Usando uma metodologia similar ao seu trabalho anterior [21], CHANG [238] propôs uma variação chamada de iteração de conjunto de valores (*value set iteration - VSI*). Além de utilizar o vetor de valores calculado na iteração anterior, o algoritmo também utiliza um grupo de vetores de valores calculados no sorteio aleatório de políticas. O autor demonstra que essa metodologia nunca usa mais iterações que a VI.

Não há muitos trabalhos aplicando a VI em cadeias logísticas, principalmente devido ao aumento da complexidade em função da grande quantidade de estados e decisões. Ainda assim, é possível encontrar alguns trabalhos que utilizam o algoritmo de VI puro [19, 206, 210]. Por outro lado, [161] optou por uma solução aproximada batizada como VI truncado, que percorre apenas um subconjunto finito do espaço de estados em cada iteração.

2.3.3 Agregação temporal (*time aggregation*)

Em 2002, CAO *et al.* [13] propuseram um novo algoritmo para a solução do custo médio de horizonte infinito para MDP, chamado de *agregação temporal*. O algoritmo proposto se baseia no PI. Inicialmente, a agregação temporal é aplicada em problemas onde parte dos estados possuem apenas uma ação disponível. Pela agregação temporal, a cadeia de Markov original é reduzida para apenas os estados onde há mais de uma ação a ser escolhida, criando um problema SMDP. Apesar dos estados com apenas uma decisão possível não aparecerem na nova cadeia embutida, suas informações (probabilidades de transição e função de custo) são usadas para se criar

as probabilidades de transição e função de custo da nova cadeia. O nome agregação temporal é dado pois todo período de tempo em que a cadeia original percorre os estados com apenas uma ação disponível é agregado em apenas uma transição na nova cadeia embutida. Quanto maior for a quantidade de estados com apenas uma ação possível, maior será o ganho computacional do algoritmo de agregação temporal, comparando-se com o método tradicional de PI.

Comparando com a agregação de estados, na qual é feita uma partição do espaço de estados e uma nova cadeia agrupa os estados pertencentes a uma mesma partição e um mesmo meta-estado, os autores provam que na agregação temporal as propriedades de Markov da cadeia original permanecem na nova cadeia e que a solução ótima da nova cadeia é a mesma da cadeia original. Na agregação de estados, devido a aproximação das partições nos meta-estados, não é garantido que nova cadeia seja *markoviana* e, mesmo que seja, o resultado encontrado será sub-ótimo [10–12].

Para problemas onde todos os estados possuem mais de uma ação possível (e que precisa-se definir a ação ótima) CAO *et al.* [13] propõem uma adaptação, criando uma partição do espaço de estado em n conjuntos. Para cada um desses n conjuntos, utiliza-se a metodologia da agregação temporal fixando-se a ação para os outros $n-1$ conjuntos. Otimiza-se cada conjunto por vez, retornando para os conjuntos iniciais até que se ache a política ótima para todos os conjuntos. Os autores deixam claro que o benefício dessa metodologia para problemas em que todos os estados possuem mais de uma ação possível ainda não estaria claro. Ainda trabalhando com o critério de recompensa média em horizonte infinito, recentemente, uma abordagem de dois estágios com agregação de tempo foi proposta por ARRUDA e FRAGOSO [14, 239]. Os autores utilizam uma política fixa na maioria do espaço de estados e otimizam as ações apenas no subconjunto de estados mais visitados. A política fixa então é atualizada por meio de um passo de melhoria de política, gerando uma nova condição de contorno para a otimização da iteração seguinte. Os passos de otimização e melhoria de política são alternados até a convergência.

Em seu trabalho original, CAO *et al.* [13] desenvolvem um algoritmo que é uma variação do iteração de política específica para resolver MDP com agregação temporal e custo médio. O novo algoritmo se baseia no fato, provado pelos autores, de que a cadeia embutida pode ser modelada como um MDP fracionário. Seguindo esta característica, novos algoritmos foram desenvolvidos, tais como: variações da iteração de valor [240–242] e da iteração de política [13, 240, 243], programação linear [242], baseados em performance de gradiente (*performance based-based algorithms*)[242] e utilizando técnicas de aprendizado por reforço [243]. Em vez de trabalhar no algoritmo, ARRUDA e FRAGOSO [244, 245] propõem a aplicação de uma transformação [4] que converte um SMDP em um MDP padrão e assim torna possível a aplicação de algorítmos convencionais de programação dinâmica. Os au-

tores utilizam a agregação temporal para transformar o MDP original em um SMDP e depois aplicam uma transformação que retornar à modelagem MDP porém mais simples que a original.

Apesar das primeiras aplicações serem focadas em custo médio, foi natural o surgimento de adaptações da agregação temporal para problemas com *recompensa total descontada esperada*. Antes do trabalho de CAO *et al.* [13], HAUSKRECHT *et al.* [246] já havia dado os primeiros passos utilizando modelos hierárquicos e aplicando o conceito de cadeia embutida para esse tipo de problema. Mais recentemente, [15] extrapolam a metodologia de duas etapas apresentada em [14, 239] para o problema com critério de custo total descontado. Na primeira etapa, o algoritmo considera uma política fixa para uma parte do espaço de estados, otimizando as ações para o restante. Na segunda etapa, ele resolve um problema SSP (*stocastic shortest path*) para atualizar a função de custo e em seguida atualizar a política fixa. O algoritmo alterna as etapas um e dois até atingir a convergência.

As primeiras aplicações da agregação temporal foram em problemas nos quais parte dos estados possui apenas uma ação disponível [13] ou estrutura semelhante. Por exemplo, WAN e CAO [247] mostra que a metodologia de agregação temporal também se ajusta muito bem em MDP de dois níveis com as seguintes características: quando o tempo de permanência de cada modo é incontrolável e quando os conjuntos das distribuições de configuração inicial após uma mudança de modo são independentes das configurações antes da mudança de modo. Neste caso, a otimização do nível 1 pode ser feitas através de agregação temporal e a do nível 2 através de cadeias de Markov embutidas. Em outro exemplo, XU e CAO [243] aplica a agregação temporal no problema de controle ótimo com amostragem Lebesgue. Pela método de amostragem Lebesgue, as decisões aplicadas em um sistema só são revisadas quando um controle atinge um valor determinado, antes disso mantém-se a decisão da última revisão. Os autores mostram que é possível converter esse problema de controle ótimo em um MDP com tempo discreto com características que permitem a aplicação da agregação temporal com sucesso.

A busca por adaptações à metodologia de agregação temporal, que facilitem sua implementação em problemas onde todos os estados possuem mais de uma ação viável, ainda está aberta. Baseado nas ideias de agregação temporal, [22, 248] sugerem uma nova alternativa para reduzir o espaço de estados. Os autores fazem uma partição do espaço de estados, e trabalham com as transições entre os grupos e dentro dos grupos. Eles usam essa metodologia para calcular a distribuição estacionária de longo prazo de uma cadeia de Markov, sem otimização. Os próprios autores indicam como próximos passos o uso desta técnica para avaliação de políticas, possibilitando a otimização em MDP. Esta tese de doutorado irá seguir este caminho apontado pelos autores.

Entre os trabalhos pesquisados envolvendo otimização de cadeias logísticas, nenhum utilizou agregação temporal.

A agregação temporal se apresenta como uma alternativa interessante na redução da busca no espaço de estados. Entretanto, sua aplicação e desempenho em MDP onde todos os estados possuem mais de uma ação disponível ainda é uma área com possibilidades de desenvolvimento e também indicam outro caminho a ser seguido nesta tese.

2.3.4 Aprendizado por reforço (*reiforcement learning - RL*)

Como previamente mencionado na Seção 2.2.4, aprendizado por reforço (RL) busca soluções aproximadas sem a utilização direta da Equação de Bellman (2.11). Um dos algoritmos clássicos de RL é o *Q-learning*, que faz uso de fatores Q, onde cada fator $Q(s, a)$ estima o valor de estar no estado s e aplicar a ação a. Os fatores $Q(s, a)$ são calculados de forma iterativa em função das observações dos resultados obtidos no passado, por isso o método é chamado de aprendizado por reforço. O Algoritmo 3 descreve os passos do *Q-learning*. Nele a escolha das decisões a_n é feita de forma ε-gulosa porém a atualização de $Q(s_n, a_n)$ é feita utilizando a decisão a que maximize $Q(s_{n+1}, a)$.

Algoritmo 3: *Q-learning*

Entrada: Escolha $\alpha \in]0, 1]$, $\varepsilon > 0$, $0 < \lambda < 1$, n_{max}, defina $n = 0$ e inicialize $Q(s, a)\ \forall\ s \in S$ e $a(s) \in A$

Saída: A solução ótima para o vetor da função de custo v^*, e a política ótima estacionária π^*

Defina um s_n inicial

enquanto $n <= n_{max}$ **faça**

  Escolha uma ação a_n possível em s_n, usando uma política ε-gulosa em Q

  Tome a ação a_n e observe $r(s_n, a_n)$ e s_{n+1}

  Atualize:

$$Q(s_n, a_n) := Q(s_n, a_n) + \alpha \left[r(s_n, a_n) + \lambda \max_{a \in A(s')} Q(s_{n+1}, a) - Q(s_n, a_n) \right]$$

  $s_n = s_{n+1}$ e $n = n + 1$

fim

Defina $\forall s \in S$:

$$v^*(s) = \max_a Q(s, a); \text{ e}$$
$$\pi^*(s) = \arg\max_a Q(s, a)$$

retorna v^*, π^*

Uma variação do *Q-learning* é o algoritmo SARSA. Nele, a escolha de a_{n+1}

também é feita de forma ε-gulosa. Este mesmo a_{n+1} é utilizado na atualização de $Q(s_n, a_n)$ e é utilizado como a_n na próxima iteração. O nome SARSA vem da sequencia: $s_n, a_n, r_n, s_{n+1}, a_{n+1}$ [17, 18, 249].

Algoritmo 4: *SARSA*

Entrada: Escolha $\alpha \in]0, 1]$, $\varepsilon > 0$, $0 < \lambda < 1$, n_{max}, defina $n = 0$ e inicialize $Q(s, a)\ \forall\ s \in S$ e $a(s) \in A$

Saída: A solução ótima para o vetor da função de custo v^*, e a política ótima estacionária π^*

Defina um s_n inicial

Escolha uma ação a_n possível em s_n, usando uma política ε-gulosa em Q

enquanto $n <= n_{max}$ **faça**

  Tome a ação a_n e observe $r(s_n, a_n)$ e s_{n+1}

  Escolha uma ação a_{n+1} possível em s_{n+1}, usando uma política ε-gulosa em Q

  Atualize:

$$Q(s_n, a_n) := Q(s_n, a_n) + \alpha \left[r(s_n, a_n) + \lambda Q(s_{n+1}, a_{n+1}) - Q(s_n, a_n) \right]$$

  $s_n = s_{n+1}$, $a_n = a_{n+1}$ e $n = n + 1$

fim

Defina $\forall s \in S$:

$$v^*(s) = \max_a Q(s, a); \text{ e}$$
$$\pi^*(s) = \arg\max_a Q(s, a)$$

retorna v^*, π^*

A programação dinâmica em tempo real (*real-time dynamic programming*) mistura o conceitos de aprendizado por reforço com programação dinâmica. Nela, a atualização dos valores dos estados é feita somente no estados visitados, como no $Q - Learning$ e no $SARSA$. Para atualização dos valores, utiliza-se o valore esperados como na iteração de valor.

Uma técnica que ajuda na convergência dos algoritmos de aprendizado por reforço com política ε-gulosa é iniciar os valores $Q(s, a)$ com estimativas otimistas. Assim, nas primeiras iterações, conforme os $Q(s, a)$ visitados tem seus valores atualizados para valores menores mais reais, o algoritmo é incentivado a percorrer os demais estados com valores otimistas, fazendo com que o algoritmos explore mais estados e ações nas iterações iniciais. [17, 18].

Trabalhando com SMDP e recompensa média, [250] desenvolveu a técnica de recompensa média de semi-Markov (*semi-Markov average reward technique - SMART*). Os autores utilizam o cálculo de recompensas médias com tempo de permanência, conforme apresentado em [4], e fazem a estimativa da recompensa usando

o método de diferenças temporais com decaimento exponencial.

Dentre os trabalhos aplicados a cadeias logísticas pesquisados, o algoritmo de RL mais usado é o Algoritmo *Q-learning* [213, 219–221]. [219] também utiliza o SARSA em seu trabalho. Alguns autores desenvolveram variações do *Q-learning* buscando melhoria na solução e redução do tempo computacional. Comparando sua variação chamada de Q-III [214] com o *Q-learning*, o Q-III permite que o agente atualize as experiências obtidas por meio de uma ação para todas as ações semelhantes; essa experiência tem caráter permanente. [212] e [215] combinaram o *Q-learning* com outras técnicas: o VNS e um módulo de aprendizado desacoplado (*off-line*).

Nos problemas envolvendo modelagem SMDP, o SMART é bastante utilizado [217, 218]. Uma abordagem que combina ADP com RL, utiliza uma variação desse algoritmo denominada λ-SMART [216].

Entre os algoritmos empregados mais ocasionalmente, o *R-learning* é utilizado em [223], com o intuito de reduzir a necessidade de definição de parâmetros e suas calibrações em um processo de simulação. Já [222] faz uso de um procedimento denominado algoritmo de aprendizado por reforço baseado em casos (*case-based reinforcement learning).*

Os modelos de RL são alternativas bastante interessantes para contornar a *maldição da dimensionalidade*. Eles apresentam bons resultados para problemas bastante complexos. Entretanto, devido à utilização de funções aproximadas, eles não garantem a otimalidade em suas soluções. A grande vantagem do RL ocorre nos problemas onde as probabilidades de transição ou as recompensas não são conhecidas previamente. Neste caso, os algoritmos tradicionais como VI e PI não podem ser aplicados.

2.3.5 Programação dinâmica aproximada (*approximate dynamic programming - ADP*)

Uma das alternativas para contornar a *maldição da dimensionalidade* é o uso de funções aproximadas. Um dos primeiros trabalhos avaliando métodos estocásticos aproximados foi elaborado por ROBBINS e MONRO [251], que propõem um algoritmo de aproximação estocástica e provam a sua convergência. Utilizando uma abordagem distinta, BELLMAN [8] utiliza uma aproximação da função de custo em dimensão reduzida.

Apesar da originalidade desses trabalhos pioneiros, o uso da programação dinâmica aproximada só veio a crescer nas últimas décadas. Atualmente, o uso de aproximações (lineares ou não) para a função de custo (Equação (2.15)) é uma das técnicas mais utilizadas. O vetor de parâmetros $\mathbf{w}$ é atualizado a cada iteração (n). Uma técnica bastante utilizada é o gradiente-descendente estocástico (*stochastic*

gradient-descent) [17]:

$$\mathbf{w}_{n+1} := \mathbf{w}_n - \frac{1}{2}\alpha \bigtriangledown \left[V^{\pi}(s_n) - \hat{V}(s_n, \mathbf{w}_n)\right]$$

Quando não é possível calcular a matriz de transição, a estratégia utilizada pelo algoritmo de iteração de valor aproximada (*approximate value iteration - AVI*) é a geração de cenários em cada iteração. A atualização da função de custo é feita com o decaimento exponencial, dando peso maior para a realização mais recente [18, 252]. Buscando aumentar a velocidade de convergência, [162] incorpora a busca local ao AVI obtendo bons resultados.

Outro algoritmo bastante empregado é o de iteração de política com mínimos quadrados (*least squares policy iteration*). Este utiliza a estrutura do algoritmo de PI, combinando a avaliação de políticas através das diferenças temporais com mínimos quadrados (*least squares temporal differences*) e a melhoria de política exata [18, 252]. O método de mínimos quadrados é utilizado para se aproximar a função de custo a partir de observações do sistema.

Trabalhos mais recentes utilizam uma aproximação finita para os erros da função de custo, com vistas a garantir a convergência. [253] propõem um novo método, que faz uso de inferência bayesiana para definição do período de exploração em algoritmos que combinem ADP com RL [254].

Outra alternativa, menos utilizada, para reduzir a complexidade dos problemas é utilizar funções aproximadas de políticas (Equação (2.16)).

Dentre os trabalhos pesquisados, não foram encontradas funções aproximadas de políticas. A aproximação linear da função de custo foi o método mais utilizado [216, 226, 227, 230, 231]. Já CHENG e DURAN [233], além da aproximação linear, decompuseram o sistema global de distribuição de petróleo em subsistemas (um para cada continente), simplificando a sua solução. [232] também combina a decomposição com a aproximação paramétrica. No primeiro método, decompõe-se o problema de roteirização de entrega em subproblemas (um para cada cliente) e aproxima-se a função de custo pelo valor ótimo de um problema de programação inteira não-linear, similar ao problema da mochila, respeitando a frota total de distribuição. No segundo método, utiliza-se um aproximação paramétrica para função de custo.

Analisando os trabalhos que aplicam ADP às cadeias logísticas, o cenário encontrado reflete a diversidade de algoritmos disponíveis. A iteração de política aproximada por mínimos quadrados (*least square approximate policy iteration*) foi a abordagem mais utilizada nos trabalhos pesquisados [226, 227, 230]. Entretanto, vários outros algoritmos foram encontrados. [231] utiliza o algoritmo de aprendizado em tempo real por diferença temporal (*on-line temporal-difference learning*), ao passo que [216] opta pelo algoritmo λ-SMART, que combina ADP com RL. ZÉPHYR

et al. [228] propõem um algoritmo com otimização aproximada via programação linear generalizada (*generalized linear programming*) e LI *et al.* [229] utilizaram dois algoritmos, um baseado em simulação (*simulation-based algorithm*) e um com agregação (*aggregation algorithm*).

Os modelos ADP também são alternativas bastante interessantes para contornar a *maldição da dimensionalidade*, para problemas complexos. O principal ponto negativo é que devido à utilização de funções aproximadas, eles não garantem a otimalidade em suas soluções.

2.3.6 Comparação final

Comparando os algoritmos para MDP, é possível constatar que:

- O PI utiliza menos iterações do que o VI; entretanto, uma iteração de PI é muito mais custosa que uma de VI. O PI apresenta melhores resultados em problemas menores. Quando o número de estados é muito grande, o tempo gasto por iteração aumenta muito, tornando-se proibitivo [10, 21, 160, 255]. Por outro lado, o VI é o mais utilizado, o mais estudado e o mais fácil de ser implementado [4, 237, 238, 255];

- A variação PSI apresenta uma alternativa interessante para a redução da busca no espaço de ações, principalmente se o poder de busca do seu passo de avaliação modificado for aprimorado;

- A agregação temporal também apresenta-se com uma alternativa interessante, porém atuando na redução da busca no espaço de estados. Para aplicá-la a problemas envolvendo a indústria de mineração é necessário aumentar sua eficiência em problemas onde todos os estados possuem mais de uma ação disponível;

- ADP e RL são alternativas interessantes para problemas com muitas variáveis de estados e muitas decisões possíveis, quando não se conhece a matriz de transição à priori. Os trabalhos analisados mostram que esses algoritmos conseguem convergir em tempos computacionais aceitáveis, melhores que os PI e VI. Entretanto, as soluções obtidas não são ótimas, dado que tanto a ADP quanto o RL trabalham com funções aproximadas;

A utilização de métodos de aproximação se justifica quando os algoritmos com solução ótima global não conseguem convergir em tempo computacional aceitável. Por este motivo, conseguir aplicar uma metodologia que garanta a solução ótima com tempo computacional aceitável sempre terá mais valor que o uso de técnicas aproximadas. Neste sentido, a agregação temporal se apresenta como uma alternativa

eficiente na redução do espaço de estados, porém ainda necessitando de aprimoramento, principalmente em problemas em que todos os estados possuem mais de uma decisão possível. Ou seja, conseguir avançar com a aplicação da agregação temporal em problemas onde todos os estados possuem mais de uma decisão, irá contribuir para a aplicação de MDP em problemas complexos, especialmente na otimização da cadeia de suprimentos de mineração, onde até hoje ela nunca foi aplicada. Por outro lado, atuar na frente da redução do espaço de ações irá complementar a agregação temporal e aumentar a eficiência do algoritmo. Neste sentido, aprimorar o algoritmo PSI tornando a melhoria do passo de avaliação de política mais eficiente, irá reduzir a necessidade de passos de melhoria de política, reduzindo o tempo gasto com avaliação das ações.

Na próxima seção serão propostos dois novos algoritmos, sendo um baseado na agregação temporal e outro no PSI, que aplicados em conjunto reduzem a busca no espaço de estados e ações, e apresentaram ótimos resultados em uma classe específica de problemas complexos. A otimização da cadeia de mineração pode ser enquadrada nessa classe específica, e estes novos algoritmos serão aplicados em um problema desta cadeia. Tanto os novos algoritmos, quanto a aplicação de MDP envolvendo todos os elos da cadeia de mineração serão avanços inéditos que serão buscados neste trabalho.

Capítulo 3

Metodologia

Neste capítulo serão estabelecidas as duas abordagens inovadoras propostas por esse trabalho. Primeiramente, na seção 3.1, será introduzida a modelagem MDP aplicada a cadeia completa da mineração. Em seguida, na seção 3.2, serão apresentados os dois novos algoritmos para solução de problemas MDP com alta complexidade.

3.1 Modelagem de MDP para planejamento logístico da indústria da mineração

A modelagem proposta considera a cadeia de mineração completa, a começar pelas minas, considerando o transporte ferroviário até o porto, de onde o minério é transportado via navegação até um porto de armazenagem avançada (PAA), ou até os clientes finais. Estes podem ser clientes com os quais se tem um contrato de fornecimento, ou clientes eventuais, denominados na literatura como clientes de mercado *spot*. A Figura 3.1 mostra o modelo esquemático. As elipses negras com letras brancas são os elos de produção e armazenagem na cadeia.

O problema enfrentado tipicamente possui decisões sequenciais durante um longo horizonte de tempo. Com base nas informações disponíveis ao longo da cadeia naquele momento (tipicamente capacidades, demandas, preços de venda, custos e posições de estoque) e considerando incertezas futuras (tais como capacidades, disponibilidades, demandas, preços e custos) deve-se tomar decisões operacionais (produção, transportes, reabastecimento de estoques e volumes vendidos). Esta dinâmica se repete ao longo de vários períodos (por exemplo, meses) e as empresas buscam as decisões que maximizem seus lucros. As decisões são tomadas levando em consideração apenas as informações disponíveis no período, não sendo importante as decisões, as informações e o lucro obtido em períodos anteriores, o objetivo é sempre maximizar o lucro futuro. Como já foi visto na Seção 2.2.3, o uso de MDP se enquadra perfeitamente nesta dinâmica. O fato de apenas as informações do período serem

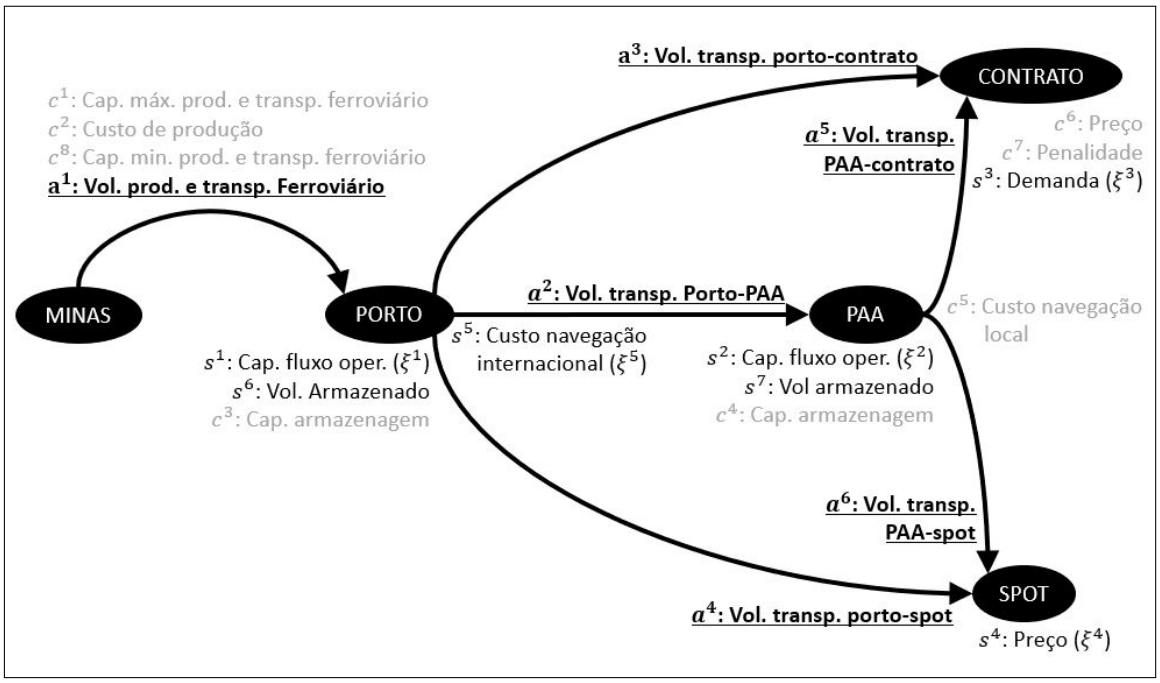

Figura 3.1: Modelagem da cadeia logística da mineração

relevantes para a tomada de decisão, garante a suposição *markoviana*. Os estados são as informações disponíveis, as ações são as decisões e o lucro total é apurado em função dos resultados financeiros mensais.

Seguindo a modelagem MDP apresentada na Seção 2.2.3, em cada período, o tomador de decisão precisa observar as seguintes informações, que variam ao longo do tempo (texto em preto, não sublinhado e sem negrito na Figura 3.1):

- Capacidade máxima do fluxo operacional no porto (s_t^1);
- Capacidade máxima do fluxo operacional no PAA (s_t^2);
- Demanda dos clientes com contrato (s_t^3);
- Preço do mercado spot (s_t^4);
- Custo da navegação internacional partindo do porto (s_t^5);
- Volume armazenado no porto (s_t^6);
- Volume armazenado no PAA (s_t^7).

O vetor de estados do sistema (s) tem 7 componentes, sendo representado por: $s_t = [s_t^1, s_t^2, s_t^3, s_t^4, s_t^5, s_t^6, s_t^7]$.

As variáveis de decisão são as setas em preto na Figura 3.1 (texto em preto, sublinhado e em negrito). Seja A o conjunto de todas as ações possíveis. Em

cada etapa o decisor deve escolher um vetor de ação ($a_t \in A$) de seis dimensões ($a_t = [a_t^1, a_t^2, a_t^3, a_t^4, a_t^5, a_t^6]$), a saber:

- Volume de produção e transporte ferroviário das minas até o porto (a_t^1);
- Volume transportado do porto para o PAA (a_t^2);
- Volume transportado do porto para os clientes com contrato (a_t^3);
- Volume transportado do porto para o mercado spot (a_t^4);
- Volume transportado do PAA para os clientes com contrato (a_t^5);
- Volume transportado do PAA para o mercado spot (a_t^6).

O modelo possui cinco incertezas que compõem o vetor da variável aleatória ($\xi_t = [\xi_t^1, \xi_t^2, \xi_t^3, \xi_t^4, \xi_t^5]$, que também estão representados na Figura 3.1:

- Variável aleatória que representa as incertezas na capacidade de fluxo operacional no porto (ξ_t^1);
- Variável aleatória que representa as incertezas na capacidade de fluxo operacional no PAA (ξ_t^2);
- Variável aleatória que representa as incertezas na demanda dos clientes com contrato (ξ_t^3);
- Variável aleatória que representa as incertezas no preço do mercado spot (ξ_t^4);
- Variável aleatória que representa as incertezas no custo da navegação internacional partindo do porto (ξ_t^5);

As capacidades de fluxo operacional no porto e no PAA dependem das manutenções programadas em função das vistorias realizadas nos equipamentos operacionais. Pode-se assumir que o cronograma de manutenção do mês corrente é conhecido e o dos próximos meses dependerá das ocorrências observadas nas vistorias. Nos contratos com os clientes, há um intervalo de demanda prevista e eles precisam informar com uma antecedência combinada qual volume necessitarão. Neste modelo, assume-se que a antecedência exigida é de um mês. Apesar dos preços do mercado *spot* de mineração e do custo da navegação internacional sofrerem importantes variações ao longo do tempo, sua volatilidade e suas negociações não são diárias. Assim, pode-se assumir que o preço do mês corrente é conhecido, pois já está negociado e a incerteza está nos preços futuros. Desta forma, apesar das incertezas envolvendo todas essas informações, no modelo proposto elas são conhecidas no início do período (mês)

e a incerteza está nos meses futuros. Além disso, as variáveis aleatórias possuem distribuições de probabilidades estacionárias, ou seja:

$$\xi_t^i \sim U^i, \forall t \in T \text{ e } i \in \{1, 2, \ldots, 5\},$$

onde U^i é uma distribuição de probabilidade genérica estacionária (independente de t) $\forall i \in \{1, 2, \ldots, 5\}$.

Conhecendo o vetor de estados (s_t), de decisão (a_t) e da variável aleatória (ξ_t), é possível escrever a função de transição ($s_{t+1} = f(s_t, a_t, \xi_t)$) para cada um dos elementos do vetor de estados:

$$s_{t+1}^1 = \xi_t^1 \tag{3.1}$$
$$s_{t+1}^2 = \xi_t^2 \tag{3.2}$$
$$s_{t+1}^3 = \xi_t^3 \tag{3.3}$$
$$s_{t+1}^4 = \xi_t^4 \tag{3.4}$$
$$s_{t+1}^5 = \xi_t^5 \tag{3.5}$$
$$s_{t+1}^6 = s_t^6 + a_t^1 - a_t^2 - a_t^3 - a_t^4, \tag{3.6}$$
$$s_{t+1}^7 = s_t^7 + a_t^2 - a_t^5 - a_t^6. \tag{3.7}$$

No modelo proposto, as seguintes informações não variam ao longo do tempo e estão com o texto em cinza na Figura 3.1:

- Capacidade de produção e transporte ferroviário máxima (c^1) e mínima (c^8) das minas até o porto: Considerando o conjunto de todas as minas disponíveis e o transporte ferroviário, é comum que manutenções e interrupções ocorram de forma distribuída, não havendo grandes alterações na capacidade total do sistema e mantendo a capacidade máxima constante. Por outro lado, é comum haver um volume mínimo de produção desejado, para evitar ociosidade nos equipamentos fixos;

- Custo de produção (c^2): Por ser uma operação muito baseada em investimentos em equipamentos, os custos são relativamente estáveis durante o período;

- Capacidade de armazenagem no porto (c^3) e capacidade de armazenagem no PAA (c^4): As capacidades de armazenagem nos portos não sofrem alterações, exceto em momentos pontuais de redimensionamento, que exigem grandes investimentos.

- Custo do transporte marítimo (navegação) local (c^5): Ao contrário da navegação de longo curso, que sofre influência dos fluxos mundiais de produtos e pode sofrer variações significativas de custo, a cabotagem local muitas vezes é uma operação dedicada com custos estáveis;

- Preço de venda para os clientes com contrato (c^6) e penalidade contratual por não atendimento da demanda (c^7): O preço para clientes de contrato é normalmente fixado em contrato, assim como as penalidades para o não atendimento da demanda.

As restrições que delimitam o espaço de ações viáveis para o estado s ($A(s)$) são:

- Cap. de prod. e transp. ferroviário: $a_t^1 \leq c^1$;
- Produção mínima: $a_t^1 \geq c^8$;
- Cap. fluxo oper. no porto: $a_t^2 + a_t^3 + a_t^4 \leq s_t^1$;
- Cap. fluxo oper. no PAA: $a_t^5 + a_t^6 \leq s_t^2$;
- Cap. armazenagem no porto: $s_t^6 + a_t^1 - a_t^2 - a_t^3 - a_t^4 \leq c^3$;
- Cap. armazenagem no PAA: $s_t^7 + a_t^2 - a_t^5 - a_t^6 \leq c^4$;
- Demanda dos clientes de contrato: $a_t^3 + a_t^5 \leq s_t^3$.

Seguindo a modelagem MDP apresentada na Seção 2.2.3, a recompensa ($r(s_t, a_t)$) é o lucro obtido em cada período (mês):

$$
\begin{aligned}
&r_t(s_t, a_t) = Receita(s_t, a_t) - CustoOper(s_t, a_t) - Penalidades(s_t, a_t),\\
&Receita(s_t, a_t) = c^6 \cdot (a_t^3 + a_t^5) + s_t^4 \cdot (a_t^4 + a_t^6),\\
&CustoOper(s_t, a_t) = c^2 \cdot a_t^1 + s_t^5 \cdot (a_t^2 + a_t^3 + a_t^4) + c^5 \cdot (a_t^5 + a_t^6),\\
&Penalidades(s_t, a_t) = c^7 \cdot (s_t^3 - a_t^3 - a_t^5).
\end{aligned}
$$

Complexos de mineração são operações que duram bastante tempo. Por exemplo, a mina do Cauê em Itabira está funcionando há 80 anos [256] e a mina de Carajás é prevista durar 400 anos [257], ambas operadas pela Vale no Brasil. Desta forma, com o modelo de intervalos mensais, será considerado que o horizonte de tempo é infinito.

Para calcular o lucro total, será utilizada a metodologia do valor presente líquido (VPL), pela qual os lucros obtidos nos diversos períodos (R_t) são trazidos a valor presente considerando uma taxa de desconto i ($0 \leq i < 1$), conforme a Equação (3.8).

$$
VPL = \sum_{t=0}^{n} \frac{R_t}{(1+i)^t} \tag{3.8}
$$

Considerando que:

- Os lucros nos diversos períodos são as recompensas do MDP:

$$R_t = r(s_t, \pi(s_t));$$

- A relação entre a taxa de desconto i e o fator de desconto λ é dada como:

$$\lambda = \frac{1}{1+i};$$

- O horizonte de planejamento é infinito e cada período corresponde a um mês: $n = \infty$

- O objetivo é achar a política estacionária (π^*) que maximize o VPL da operação (VPL^*) para cada estado inicial s:

$$VPL^* = V^*(s)$$

.

Desta forma, a equação da busca pela maximização do VPL pode ser descrita como a maximização da função de custo ($V^*(s)$) calculada pela *Recompensa Total Descontada Esperada*:

$$V^*(s) = \max_{\pi \in \Pi} \mathbb{E}_\pi \left\{ \sum_{t=0}^{+\infty} \lambda^t r(s_t, \pi(s_t)) | s_0 = s \right\}, \forall s \in S; \tag{3.9}$$

$$\pi^* = \arg \max_{\pi \in \Pi} V^\pi(s), \forall s \in S. \tag{3.10}$$

Com esta modelagem, conseguiu-se, de forma inédita, representar toda a cadeia de suprimentos da mineração, das minas até os clientes finais, como um MDP. Dada a natureza deste problema, com decisões sequencias sob incerteza, por um longo horizonte de tempo, dependente apenas de informações atuais disponíveis no sistema, a modelagem MDP se encaixa perfeitamente. Com ela, será possível obter a política ótima, válida para vários períodos, com as ações que maximizam o lucro para cada um dos possíveis estados do sistema. Entretanto, apesar de garantir a solução ótima global, conforme a complexidade do sistema aumenta, os algoritmos tradicionais para solução de MDP necessitam de tempos computacionais exponencialmente maiores, como é o caso do problema proposto. A busca por novos algoritmos, que atinjam as soluções ótimas em tempos computacionais mais curtos, ajuda na aplicabilidade desta modelagem em cadeias mais complexas. Na próxima seção, serão apresentados dois novos algoritmos, que podem ser aplicados conjuntamente e caminham nesta direção.

3.2 Dois Novos Algoritmos de Programação Dinâmica para MDP complexos, com muitos estados e ações.

Para resolver as equações (3.9) e (3.10) em problemas complexos, com muitos estados e ações, foram desenvolvidos dois novos algoritmos de programação dinâmica:

- Algoritmo Baseado em Agregação Temporal (*Time Aggregation Based Algorithm - TABA*);
- Algoritmo de Iteração de Grupo de Políticas com Busca Local (*Local Search Policy Set Iteration - LSPSI*).

Como atuam em frentes diferentes, TABA no espaço de estados e LSPSI no espaço de ações, eles podem ser usados conjuntamente.

Estes algoritmos, bem como seu uso conjunto, serão detalhados nas próximas seções.

3.2.1 Algoritmo Baseado na Agregação Temporal (*Time-aggregation-based algorithm - TABA*)

Observando a modelagem proposta na Seção 3.1, mais especificamente o vetor de estados (equações (3.1)-(3.7)), pode-se perceber que o problema possui uma construção característica, também encontrada em outros MDP: parte dos seus componentes de estados não dependem dos estados ou ações anteriores, apenas da variável aleatória com distribuição de probabilidade estacionária. Essa classe de MDP será chamada de Processos de Decisão Markoviano Parcialmente Estacionários (*semi-stationary Markov decision process - SSMDP*) e terá seu detalhamento apresentado abaixo. O TABA consegue explorar muito bem essa arquitetura dos SSMDP.

Processos de Decisão Markoviano Parcialmente Estacionários (*semi-stationary Markov decision process - SSMDP*)

Considere-se inicialmente MDP com vetores de estados ($s \in S$) multidimensionais, s^j ($j \in \{1, \ldots, I\}$) representando o j-ésimo componente de estados e I ($1 < I < \infty$) representando o número total de componentes de s. Agora, considere que parte dos componentes de estados $I - K$ ($K < I$) evoluem conforme um processo estocástico estacionário, independente do estado do sistema ou da política de controle empregada. Sem perda de generalidade, caso seja necessário, é possível reordenar o vetor de estados de modo que estes $I - K$ componentes sejam os últimos (Suposição 1).

Suposição 1. *Seja $S = S_c \times S_n$, onde S_c é o espaço das K primeiras componentes de estado e S_n corresponda às componentes restantes. Assume-se que:*

$$p(s'|s,a) = p(s'_1,\ldots,s'_K|s,a)p(\xi = (\, s'_{K+1},\ldots,s'_I)\,), \tag{3.11}$$

onde ξ é uma variável aleatória pertencente ao espaço S_n.

Definição 1. *Processos de decisão* markoviano *parcialmente estacionários (SSMDP) são MDP onde a Suposição 1 pode ser aplicada.*

Tanto a Suposição 1 como a Definição 1 de SSMDP são bastante encontrados em problemas reais, como em problemas de gestão de estoques em cadeias logísticas complexas. Normalmente, parte das componentes de estado são os níveis de estoque nos diferentes elos das cadeias, que dependem dos estados e das ações anteriores. Essas componentes podem ser organizadas como as K primeiras componentes. As componentes restantes $(I - K)$ representam as variáveis aleatórias tais como: demanda dos clientes com contrato, preços e custos de mercado. Estas variáveis seguem distribuições de probabilidade conhecidas que podem não depender dos estados e ações anteriores. Em alguns casos, quando as alterações dessas variáveis ocorrem no médio e longo prazos, pode-se considerar que seus valores no curto prazo são conhecidos antes da tomada de decisão, fazendo parte da variável de estados e podendo ser modeladas como os últimos $I - K$ elementos de um SSMDP.

O algoritmo TABA proposto é particularmente eficiente para SSMDP e será apresentado na próxima seção.

Algoritmo TABA

Considere uma partição do espaço de estados (S) em Z conjuntos disjuntos (G_z):

$$S = \bigcup_{z=1}^{Z} G_z$$
$$G_i \cap G_j = \emptyset, \forall\ i,j \leq Z \text{ tal que } i \neq j$$

Para cada conjunto (G_z) é definido:

- $\overline{V}^{G_z}$: o vetor com a função de custo para todos os estados $s \in G_z$. A cardinalidade deste vetor é $|G_z|$:

$$\overline{V}^{G_z} = [v(s)], s \in G_z$$

- μ_z: a distribuição normalizada de estado de equilíbrio dentro do conjunto G_z. É o vetor com a probabilidade no longo prazo de cada um dos estados $s \in G_z$,

dado que o sistema se encontra em G_z. A cardinalidade deste vetor também é $|G_z|$:

$$\mu_z = [\mu_z(s)], s \in G_z, \sum_{s \in G_z} \mu_z(s) = 1.$$

- $p_G(G'_z|s,a)$: a probabilidade de transição do conjunto G'_z ser acessado a partir do estado s aplicando-se a ação a:

$$p_G(G'_z|s,a) = \sum_{s' \in G'_z} p(s'|s,a) \tag{3.12}$$

Assumindo que μ_z é conhecido para todos os subconjuntos G_z da partição, pode-se calcular a função de custo do estado de equilíbrio como:

$$\hat{V}(G_z) = \mu_z \overline{V}^{G_z} \tag{3.13}$$

No TABA, essa função de custo do estado de equilíbrio é usada na atualização da função de custo (v_{n+1}), para a iteração $n+1$, no algoritmo VI (2.19):

$$v_{n+1}(s) = \max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} p_G(G'_z|s,a) \hat{V}_n(G'_z) \right\}, \tag{3.14}$$

onde a segunda parte do lado direito da Equação 3.14 é obtida através da Equação (3.13), para as estimativas da iteração n:

$$\hat{V}(G'_z) = \hat{V}_n(G'_z) = \mu_z \overline{V}_n^{G_z} = \sum_{s \in G_z} \mu_z(s) v_n(s) \tag{3.15}$$

O Algoritmo 5 é o pseudo-código do TABA. É possível observar que o custo computacional na Equação (3.17) é diretamente proporcional ao número de subconjuntos da partição do espaço de estados, dado por Z. Assim, substituindo (2.19) por (3.14) no algoritmo de iteração de valor (Equação (3.17)) pode trazer uma economia de tempo computacional significativa quando $Z \ll |S|$. Esta é a grande motivação pela busca do algoritmo baseado em agregação temporal apresentado nesta seção.

Convergência

A grande diferença do TABA encontra-se exatamente na equação de atualização da função de custo (3.17). Primeiramente, ela assume que a probabilidade de estado de equilíbrio μ_z é conhecida para todos os subconjuntos $G_z \in S$. De maneira geral, esta distribuição é função da política ótima π^*. Entretanto, é possível explorar a estrutura dos SSMDP na Definição 1 e fazer uma partição de modo que μ_z seja independente da política de controle, não varie ao longo do tempo e possa ser calculada uma única vez durante toda a execução do algoritmo.

Algoritmo 5: TABA

Entrada: Defina $n = 0$, uma função de custo inicial v_0, uma partição do estado de estados (S) em Z conjuntos disjuntos (G_z), a distribuição normalizada de estado de equilíbrio (μ_z) para cada subconjunto G_z, um fator de desconto arbitrário $\lambda \in [0, 1)$ e uma tolerância arbitrária ϵ

Saída: A solução ótima para a função de custo v^*, e a política ótima estacionária π^*

repita

  para *cada* G_z **faça**

$$\hat{V}_n(G_z) := \sum_{s \in G_z} \mu_z(s) v_n(s) \tag{3.16}$$

  fim

  para *cada* $s \in S$ **faça**

    Calcule:

$$v_{n+1}(s) := \max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} p_G(G'_z|s,a) \hat{V}_n(G'_z) \right\} \tag{3.17}$$

  fim

  $n \leftarrow n + 1$;

até $\|v_n - v_{n-1}\|_\infty \leq \epsilon$;

Para cada $s \in S$, escolha:

$$\pi(s) \in \arg \max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} p_G(G'_z|s,a) \hat{V}_n(G'_z) \right\}$$

Defina $\pi^* = \pi$ e $v^* = v_n$

retorna v^*, π^*

Suposição 2. *Usando a mesma notação da Definição 1, a Suposição 1 indica uma partição direta para SSMDP na qual cada subconjunto corresponde a uma única combinação dos primeiros K componentes do vetor de estado:*

$$G_z = \{s \in S : \{s_1, \ldots, s_K\} = s_z \in S_c \ e \ z \in \{1, \ldots, |S_c|\}$$

Usando a partição sugerida na Suposição 2, a distribuição do estado de equilíbrio para cada subconjunto G_z depende apenas de $\{s_{K+1}, \ldots, s_I\} = s_n \in S_n$ e pode ser calculada como:

$$\mu_z = [\mu_z(s_n)] = [P(\xi = s_n)], \ \forall \ s_n \in S_n$$

onde: $s \in G_z$, $s_n \in S_n$ e ξ é uma variável aleatória com distribuição de probabilidade conhecida.

Essa particularidade do SSMDP junto com a partição sugerida, garantem a convergência do TABA, cuja prova será apresentada a seguir.

Lema 1. *Seguindo a partição sugerida na Suposição 2 para SSMDP, a Equação* (3.17) *no Algoritmo 5 é equivalente à Equação* (2.19) *no Algoritmo 2.*

Demonstração. Substituindo (3.13) em (3.17) obtém-se:

$$\begin{aligned} v_{n+1}(s) &:= \max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} p_G(G'_z|s,a) \hat{V}_n(G'_z) \right\} \\ &= \max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} p_G(G'_z|s,a) \mu_z \overline{V}_t^{G'_z} \right\} \end{aligned}$$

Aplicando a Equação (3.15), tem-se que

$$\begin{aligned} v_{n+1}(s) &:= \max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} p_G(G'_z|s,a) \sum_{s' \in G_z} \mu_z(s') v_n(s') \right\} \\ &= \max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} \sum_{s' \in G_z} p_G(G'_z|s,a) \mu_z(s') v_n(s') \right\}. \end{aligned}$$

É possível perceber que $\mu_z(s') = P(\xi = s')$ na Equação (3.11) e substituindo a Equação (3.12) na última expressão, obtém-se:

$$v_{n+1}(s) = \max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{s' \in S} p(s'|s,a) v_n(s') \right\}.$$

Finalmente, pode-se concluir a prova verificando que a equação acima é igual à equação de atualização do algoritmo de VI (2.19). □

Assim, pode-se enunciar o Teorema 1:

Teorema 1. *Algoritmo 5 converge para a solução da Equação de Bellman* (2.11)*, quando aplicado a SSMDP com a partição sugerida na Suposição 2.*

Demonstração. Pelo Lema (1), tem-se que cada iteração do algoritmo TABA é equivalente a uma iteração do algoritmo clássico de iteração de valor. A convergência do TABA então é uma consequência desta equivalência, dado que a convergência do VI para a Equação de Bellman (2.11) já está provada [4]. □

Observação 1. *Importante observar que, apesar da atualização da função de custo no Algoritmo* (5) *ser equivalente à atualização no iteração de valor, o esforço computacional é menor, dado que o TABA usa apenas os Z valores agregados da Equação* (3) *ao invés de todo o vetor v_n. Esse fato, acarreta economia computacional significativa principalmente quando tem-se que $Z \ll |S|$.*

Versão preliminar do TABA

Antes de se chegar a versão atual do algoritmo TABA, uma versão preliminar foi publicada em [258]. Esta primeira versão utilizava uma atualização da função de custo menos eficiente. Nela, a função de custo do estado de equilíbrio era utilizada apenas em transições onde havia mudança de subconjunto, ou seja, nas quais o estado seguinte não pertencia ao mesmo subconjunto do estado original ($s \in G_{\mathbb{S}}$, $s' \in G_z$ e $G_z \neq G_{\mathbb{S}}$).

$$v_{n+1}(s) = \max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{s' \in G_{\mathbb{S}}} p(s'|s,a) v_n(s') + \lambda \sum_{z \neq \mathbb{S}} p_G(G_z|s,a) \hat{V}_n(G_z) \right\}$$

Observando o Algoritmo 6 é possível perceber que a primeira versão não capturava o ganho de eficiência da utilização da função de custo de estado de equilíbrio nas transições dentro do mesmo subconjunto. Além disso, sua implementação exigia o teste de verificação se s e s' pertenciam ao mesmo subconjunto ou não. Mesmo assim, os resultados apresentados já foram promissores, quando comparados com os algoritmos tradicionais, na solução de SSMDP.

Vantagens e inovações do TABA

A grande vantagem do TABA está no uso da função de custo de estado de equilíbrio nas transições, reduzindo o espaço de busca do próximo estado. No algoritmo de VI, essa busca percorre todo o espaço de estados ($|S|$), enquanto que o TABA percorre os subconjuntos da partição (Z). Assim, o TABA faz uma redução do espaço de estados sem perder as propriedades de Markov, de forma semelhante ao que ocorre na agregação temporal [13]. Dessa forma, garante-se que as soluções obtidas são ótimas globais também para o problema original.

Por outro lado, a principal diferença entre o TABA e a agregação temporal [13] está no espaço de estados iniciais. Na agregação temporal, também há uma redução no espaço dos estados iniciais. Para problemas onde todos os estados possuem mais de uma ação possível, os autores sugerem que se faça uma partição do espaço de estados e que o algoritmo inicial seja aplicado para cada subconjunto da partição isoladamente, considerando uma política fixa para os outros subconjuntos. Dessa forma, quando termina a otimização em um subconjunto da partição, é necessário rodar os outros subconjuntos para testar se a nova política é ótima para todos os estados. A solução só é considerada ótima depois de testar todos os subconjuntos. No TABA, não há essa redução do espaço de estados iniciais. Isso permite que o algoritmo já dê a solução ótima para todos os estados, não sendo necessário aplicá-lo separadamente em cada subconjunto, nem realizar testes considerando outros subconjuntos ao final.

Algoritmo 6: Versão Preliminar do TABA

Entrada: Defina $n = 0$, uma função de custo inicial v_0, uma partição do estado de estados (S) em Z conjuntos disjuntos (G_z), a distribuição normalizada de estado de equilíbrio (μ_z) para cada subconjunto G_z, um fator de desconto arbitrário $\lambda \in [0, 1)$ e uma tolerância arbitrária ϵ

Saída: A solução ótima para a função de custo v^*, e a política ótima estacionária π^*

repita
  para *cada* G_z **faça**

$$\hat{V}_n(G_z) := \sum_{s \in G_z} \mu_z(s) v_n(s)$$

  fim
  para *cada* $s \in S$ **faça**
    Calcule:

$$v_{n+1}(s) = \max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{s' \in G_{\mathbb{S}}} p(s'|s,a) v_n(s') + \lambda \sum_{z \neq \mathbb{S}} p_G(G_z|s,a) \hat{V}_n(G_z) \right\}$$

  fim
  $n \leftarrow n + 1$;
até $\|v_n - v_{n-1}\|_\infty \leq \epsilon$;
Para cada $s \in S$, escolha:

$$\pi(s) \in \arg\max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{s' \in G_{\mathbb{S}}} p(s'|s,a) v_n(s') + \lambda \sum_{z \neq \mathbb{S}} p_G(G_z|s,a) \hat{V}_n(G_z) \right\}$$

Defina $\pi^* = \pi$ e $v^* = v_n$
retorna v^*, π^*

O TABA se diferencia dos algoritmos de agregação de estados, pois esses fazem aproximações ao agrupar e unir os estados em conjuntos. Ou seja, todos os estados agrupados em um mesmo conjunto são tratados como iguais, apresentando a mesma probabilidade de transição e a mesma função de custo, fazendo com que a propriedade de Markov da cadeia original seja perdida. Desse modo, a solução obtida passa a ser uma aproximação e não o ótimo global do problema original. O TABA não reduz o espaço de estados iniciais e se aproveita das características do SSMDP para garantir a propriedade de Markov e obter o ótimo global.

3.2.2 Algoritmo de Iteração de Grupo de Políticas com Busca Local (*Local search policy set iteration - LSPSI*)

O segundo algoritmo proposto atua exclusivamente no espaço de ações, usando como base o algoritmo PSI desenvolvido por [21]. O algoritmo proposto (LSPSI) propõe uma redução no espaço de ações em boa parte das iterações, acelerando a convergência.

O PSI, apresentado em [21], busca acelerar a convergência do algoritmo de iteração de política fazendo o passo de avaliação de política não apenas para a melhor política identificada no passo de melhoria de política anterior. O algoritmo inclui no passo de avaliação de política uma amostra aleatória de outras políticas. No próximo passo de melhoria de política, o PSI usa o máximo entre os valores das funções de custo de todas as políticas avaliadas no passo anterior. Desta forma, ele incorpora uma possibilidade de melhoria no passo de avaliação de políticas. O algoritmo proposto (LSPSI) segue este mesmo caminho, incorporando uma possibilidade de melhoria no passo de avaliação de política. Comparando ao PSI, o LSPSI explora a natureza combinatorial do problema para aumentar esse poder de melhoria.

Algoritmo LSPSI

Comparando com o PSI, o LSPSI também parte da estrutura do algoritmo de iteração de política, combinando passos de avaliação e melhoria de política e também introduz a possibilidade de melhoria no passo de avaliação de política. Por outro lado, o LPSI, ao invés de utilizar inversão de matrizes, utiliza o algoritmo de iteração de valor no passo de avaliação de política. Assim como o PSI, ele não utiliza apenas a política identificada no último passo de melhoria de política. Em cada iteração do algoritmo VI no passo de avaliação de política, além da melhor política, o LSPSI também inclui uma amostra aleatória de outras ações possíveis ($H(s) \subset A(s)$) para cada estado $s \in S$, transformando o passo de avaliação de política em uma busca local no espaço de ações. Considerando o efeito combinatório de se avaliar $|H(s)|$ ações para cada estado s, em cada iteração são avaliadas $(|H(s)| + 1)^{|S|}$ políticas, trazendo um grande poder de melhoria, mesmo quando $|H(s)|$ é pequeno.

O tamanho de $H(s)$ é definido como um percentual fixo (α) do espaço de ações possíveis no estado s ($A(s)$), conforme a Equação (3.18). Para garantir que a avaliação da política seja ao menos tão boa quanto à anterior, na primeira iteração inclui-se de forma *gulosa* as ações do último passo de melhoria de política ($\pi_n(s)$) e nas iterações seguintes adiciona-se as ações selecionadas nas atualizações de valores da função de custo ($\pi_{n\_eval}(s)$), criando o subconjunto $D(s)$ conforme a Equação (3.19).

$$\frac{|H(s)|}{|A(s)|} = \alpha \leq 1, \tag{3.18}$$

$$D(s) = H(s) \cup \pi_n(s) \text{ ou } D(s) = H(s) \cup \pi_{n\_eval}(s), \tag{3.19}$$

Cada iteração de valor do passo de avaliação de política do LSPSI usa a atualização da função de custo conforme (3.20):

$$v_{n+1}(s) = \max_{a \in D(s)} \left\{ r(s,a) + \lambda \sum_{s' \in S} p(s'|s,a) v_n(s') \right\}, \tag{3.20}$$

onde $D(s)$ é construído conforme a Equação (3.19) em cada iteração. Dado que o algoritmo VI necessita de diversas iterações até sua convergência, fazer essa amostragem durante a etapa de avaliação de política com diferentes ações em cada uma das iterações garante que praticamente todas as ações serão testadas com um custo computacional reduzido. Ao final, para testar se a política encontrada na etapa de avaliação de política é ótima, é feito um passo de melhoria de política completo, considerando todas as ações. Caso o passo de melhoria de política encontre outra melhor, o algoritmo volta para o passo de avaliação de política modificado. Caso não encontre, a política final da avaliação já é a ótima. O Algoritmo 7 detalha os passos do algoritmo LSPSI.

É interessante observar que, assim como ocorre com o PSI, o algoritmo de iteração de política é um caso particular do LSPSI quando $\alpha = 0$ e $H(s) = \emptyset$. Por outro lado, quando $\alpha = 1$, $H(s)$ passa a ser todo o espaço de ações ($H(s) = A(s)$), e nesse caso o passo de avaliação de política passa a ser um passo de melhoria de política e o LSPSI se torna equivalente ao algoritmo de iteração de valor.

Convergência

A prova de convergência do LSPSI é baseada na prova de convergência do algoritmo de iteração de política apresentada em [4]. Provando-se que o passo de avaliação de política modificado do LSPSI preserva a Proposição 6.4.1 em [4], os passos seguintes são idênticos e a prova estará concluída.

Lema 2. (Proposição 6.4.1 em [4] adaptada)

Seja v_n^{*D} a avaliação de política definida no n-ésimo passo de avaliação de política do LSPSI. Então $v_{n+1}^{*D} \geq v_n^{*D}$.

Demonstração. Seja:

- $v_n^{*\pi}$ a avaliação de política definida no n-ésimo passo de avaliação de política do PI;

Algoritmo 7: Algoritmo LSPSI

Entrada: Defina $n = 0$, uma função de custo inicial v_0, uma política inical arbitrária $\pi_0 \in \Pi$, um fator de desconto arbitrário $\lambda \in [0, 1)$ e uma tolerância arbitrária ϵ

Saída: A solução ótima para a função de custo v^*, e a política ótima estacionária π^*

repita

Passo de avaliação de política modificado

$n\_eval \leftarrow 0$; $v_{n\_eval} \leftarrow v_n$; $\pi_{n\_eval} \leftarrow \pi_n$;

repita

para *cada* $s \in S$ **faça**

Escolha $H(s) \subset A(s)$ aleatoriamente; $D(s) \leftarrow H(s) \cup \pi_{n\_eval}(s)$

Calcule:

$$v_{n\_eval+1}(s) := \max_{a \in D(s)} \left\{ r(s,a) + \lambda \sum_{s' \in S} p(s'|s,a) v_{n\_eval}(s') \right\} \quad (3.21)$$

$$\pi_{n\_eval+1}(s) \in \arg\max_{a \in D(s)} \left\{ r(s,a) + \lambda \sum_{s' \in S} p(s'|s,a) v_{n\_eval}(s') \right\}$$

fim

$n\_eval \leftarrow n\_eval + 1$;

até $\|v_{n\_eval} - v_{n\_eval-1}\|_\infty \leq \epsilon$;

Passo de atualização de política

$v_n \leftarrow v_{n\_eval}$;

para *cada* $s \in S$ **faça**

Calcule:

$$v_{n+1}(s) := \max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{s' \in S} p(s'|s,a) v_n(s') \right\} \quad (3.22)$$

$$\pi_{n+1}(s) \in \arg\max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{s' \in S} p(s'|s,a) v_n(s') \right\}$$

fim

$n \leftarrow n + 1$;

até $\|v_n - v_{n-1}\| \leq \epsilon$;

Defina $\pi^* = \pi_n$ e $v^* = v_n$

retorna v^*, π^*

- v^{π_i} a avaliação de política definida na i-ésima iteração do algoritmo de iteração de valor no passo de avaliação de política do PI;
- v^{D_i} a avaliação de política definida na i-ésima iteração do algoritmo de iteração de valor no passo de avaliação de política do LSPSI.

Pelo Teorema 6.3.1 de [4] tem-se que v^{π_i} converge em norma para $v_n^{*\pi}$ fazendo-se $A(s) = \{\pi(s)\}$:

$$\lim_{i\to\infty} v^{\pi_i} \equiv v_n^{*\pi}$$

onde,

$$\begin{aligned}
v^{\pi_1}(s) &= r(s,\pi(s)) + \lambda \sum_{s'\in S} p(s'|s,\pi(s)) v^{\pi_0}(s') \\
v^{\pi_2}(s) &= r(s,\pi(s)) + \lambda \sum_{s'\in S} p(s'|s,\pi(s)) v^{\pi_1}(s') \\
&\vdots \\
v^{\pi_{i+1}}(s) &= r(s,\pi(s)) + \lambda \sum_{s'\in S} p(s'|s,\pi(s)) v^{\pi_i}(s') \\
&\vdots
\end{aligned}$$

Aplicando o Teorema 6.3.1 às primeiras iterações do algoritmo de iteração de valor no passo de avaliação de política do LSPSI e lembrando que na primeira iteração ele também parte da estimativa inicial da função de custo obtida no último passo de melhoria de política ($v^{D_0} = v^{\pi_0}$), tem-se que

$$\begin{aligned}
v^{D_1}(s) &= \max_{a\in D(s)} \left\{ r(s,a) + \lambda \sum_{s'\in S} p(s'|s,a) v^{\pi_0}(s') \right\} \\
v^{D_2}(s) &= \max_{a\in D(s)} \left\{ r(s,a) + \lambda \sum_{s'\in S} p(s'|s,a) v^{D_1}(s') \right\} \\
&\vdots \\
v^{D_{i+1}}(s) &= \max_{a\in D(s)} \left\{ r(s,a) + \lambda \sum_{s'\in S} p(s'|s,a) v^{D_i}(s') \right\} \\
&\vdots
\end{aligned}$$

Comparando as equações da primeira iteração, como $\pi(s) \in D(s)$, tem-se que:

$$\begin{aligned}
v^{D_1}(s) &= \max_{a\in D(s)} \left\{ r(s,a) + \lambda \sum_{s'\in S} p(s'|s,a) v^{\pi_0}(s') \right\} \\
&= \max \left\{ \max_{a\in H(s)} \left\{ r(s,a) + \lambda \sum_{s'\in S} p(s'|s,a) v^{\pi_0}(s') \right\}; \right. \\
&\qquad \left. r(s,\pi(s)) + \lambda \sum_{s'\in S} p(s'|s,\pi(s)) v^{\pi_0}(s') \right\} \\
&= \max \left\{ \max_{a\in H(s)} \left\{ r(s,\pi(s)) + \lambda \sum_{s'\in S} p(s'|s,\pi(s)) v^{\pi_0}(s') \right\}; v^{\pi_1}(s) \right\} \\
&\geq v^{\pi_1}(s)
\end{aligned}$$

Substituindo na segunda iteração e seguindo a Proposição 6.3.2 de [4], tem-se que $v^{D_2}(s) \geq v^{\pi_2}(s)$ e por indução $v^{D_i}(s) \geq v^{\pi_i}(s)$, $\forall i > 1$.

Fazendo o limite $i \to \infty$:

$$v_n^{*D} \equiv \lim_{i \to \infty} v^{D_i} \geq \lim_{i \to \infty} v^{\pi_i} \equiv v_n^{*\pi}$$

Ou seja, ao final do passo de avaliação de política, a função de custo obtida pelo passo de avaliação modificada do LSPSI é maior ou igual à função obtida pela avaliação de política do algoritmo de PI tradicional: $v_n^{*D} \geq v_n^{*\pi}$.

Para finalizar, a proposição 6.4.1 de [4] prova que $v_{n+1} \geq v_n$, onde v_n é a avaliação de política no n-ésimo passo do algoritmo de PI. Fazendo $v_n = v_n^{*D}$ e lembrando que usando o algoritmo de VI no passo de avaliação de política do PI tem-se que $v_{n+1} = v_{n+1}^{*\pi}$:

$$v_{n+1}^{*D} \geq v_{n+1}^{*\pi} = v_{n+1} \geq v_n = v_n^{*D}$$

□

Como, pelo Lema 2, a proposição 6.4.1 de [4] continua válida para o LSPSI, o Teorema 6.4.2 de [4] também continua válido, garantindo que o LSPSI termina em uma quantidade finita de iterações, com uma solução ótima para a função de custo e sua política.

Versão preliminar do LSPSI

Assim como ocorreu com o TABA, antes de se chegar a versão atual do algoritmo LSPSI, uma versão preliminar foi publicada em [258] com o nome de *Extensão do Algoritmo de Iteração de Conjunto de Políticas (Policy Set Iteration Algorithm Extension - PSIAE)*. Nesta primeira versão, não havia o passo de melhoria de política. Aplicava-se apenas uma vez o passo de avaliação de política modificado e o algoritmo encerrava quando esse passo convergisse. Desta forma, não havia garantia da solução ótima global. O Algoritmo 8 apresenta o pseudo código do PSIAE.

Vantagens e inovações do LSPSI

Quando comparado com o PSI, o LSPSI aumenta o poder de melhoria do passo de avaliação de política modificado. O PSI faz a avaliação das políticas usando a inversão de matrizes, obtendo apenas um vetor da função de custo para cada uma das políticas avaliadas: $N\_Pol_{(PSI)} = |\Delta| = |\Psi| + 1$. No passo de melhoria de política o PSI escolhe a melhor das $N\_Pol_{(PSI)}$ políticas. Já no LSPSI, em cada iteração de valor do passo de avaliação modificado, sorteia-se aleatoriamente $|H(s)| = \alpha \cdot |A(s)|$ ações para cada estado. Como, além das ações sorteadas, o

Algoritmo 8: Algoritmo PSIAE

Entrada: Defina $n = 0$, uma função de custo inicial v_0, uma política inical arbitrária $\pi_0 \in \Pi$, um fator de desconto arbitrário $\lambda \in [0, 1)$ e uma tolerância arbitrária ϵ

Saída: A solução ótima para a função de custo v^*, e a política ótima estacionária π^*

repita
  para *cada* $s \in S$ **faça**
    Escolha $H(s) \subset A(s)$ aleatoriamente; $D(s) \leftarrow H(s) \cup \pi_n(s)$
    Calcule:

$$v_{n+1}(s) := \max_{a \in D(s)} \left\{ r(s,a) + \lambda \sum_{s' \in S} p(s'|s,a) v_n(s') \right\}$$

$$\pi_{n+1}(s) \in \arg\max_{a \in D(s)} \left\{ r(s,a) + \lambda \sum_{s' \in S} p(s'|s,a) v_n(s') \right\}$$

  fim
  $n \leftarrow n + 1$;
até $\|v_n - v_{n-1}\| \leq \epsilon$;
Defina $\pi^* = \pi_n$ e $v^* = v_n$
retorna v^*, π^*

LSPSI também avalia a melhor ação do passo anterior, o total de ações avaliadas são $N\_Acoes_{(LSPSI)} = |D(s)| = |H(s)| + 1$. Considerando que esse procedimento se repete de forma independente para cada um dos estados ($s \in S$), a quantidade de políticas avaliadas em cada iteração de valor ($N\_Pol_{(LSPSI)}$) pode ser calculado como:

$$N\_Pol_{(LSPSI)} = \prod_{s \in S} N\_Acoes_{(LSPSI)} = (N\_Acoes_{(LSPSI)})^{|S|}$$

É possível observar que fazendo-se o número de ações sorteadas no LSPSI igual ao número de políticas sorteadas no PSI ($|H(s)| = |\Psi|$), o número de políticas avaliadas com LSPSI em cada iteração de valor do passo de avaliação é muito maior, crescendo exponencialmente com o número de estados, do que a quantidade de políticas avaliadas em cada passo de avaliação modificado completo do PSI.

$$N\_Pol_{(LSPSI)} = (|\Psi| + 1)^{|S|} \gg |\Psi| + 1 = N\_Pol_{(PSI)} \tag{3.23}$$

onde: $|H(s)| = |\Psi|$ e $|S| \gg 1$

Outro ponto importante do LSPSI é que há uma boa probabilidade do algoritmo convergir com apenas um passo de avaliação de política modificado.

Considere que:

- O passo de avaliação de política modificado precisa de diversas iterações de valores para convergir (*n_eval* no Algoritmo (7));

- em cada iteração de valor, $\alpha \cdot |A(s)|$ novas ações possíveis são avaliadas.

A probabilidade de não se avaliar a ação ótima diminui exponencialmente com o número de iterações do algoritmo de iteração de valor no passo de avaliação de política do LSPSI (n_eval). Em cada iteração, a probabilidade da ação ótima estar entre as sorteadas é igual a α. Considerando que em cada iteração o sorteio é feito de forma independente, a probabilidade de não se avaliar a ação ótima em um passo de avaliação modificado pode ser calculada como: $(1-\alpha)^{n\_eval}$. Mesmo para pequenos valores de α, a medida que o número de iterações de valores até a convergência cresce (n_eval), a probabilidade de não se avaliar a ação ótima já no primeiro passo da avaliação modificada de política diminui muito. Assim, no LSPSI, há uma boa probabilidade de já se achar a ação ótima no primeiro passo de avaliação de política modificado.

O PSIAE usava essa elevada probabilidade de se achar a decisão ótima com apenas um passo de avaliação de política modificado. Por outro lado, não havia garantia de que sempre seria localizada a ação ótima global. Com a inclusão do passo de melhoria de política no LSPSI, perde-se um pouco em eficiência computacional entretanto garante-se que a solução atinja o ótimo global. O passo de melhoria de política, testa todas as políticas possíveis, garantindo a otimalidade global. Caso uma política nova seja identificada nesse passo, o LSPSI retorna para um novo passo de avaliação de política modificada.

Como é possível perceber, o LSPSI é uma evolução do PSI pois aumenta o poder de busca no passo de avaliação modificado de política e também é um avanço comparado ao PSIAE pois garante a otimalidade global.

3.2.3 Combinação dos algoritmos: LSPSI + TABA

Como o TABA atua na redução do espaço de estados e o LSPSI na redução do espaço de ações, é possível combiná-los em um único algoritmo. Partindo da estrutura do LSPSI (Algoritmo 7), o algoritmo combinado LSPSI+TABA aplica o benefício do TABA nas iterações de valores. A equação de iteração de valor do passo de avaliação modificado (3.21) passa a ser:

$$v_{n\_eval+1}(s) = \max_{a \in D(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} p_G(G'_z|s,a) \hat{V}_{n\_eval}(G'_z) \right\}.$$

onde, $\hat{V}_{n\_eval}(G_z) := \sum_{s \in G_z} \mu_z(s) v_{n\_eval}(s) \ \forall G_z$

A equação de iteração de valor do passo de melhoria de política (3.22) passa a ser:

$$v_{n+1}(s) = \max_{a \in A(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} p_G(G'_z|s,a) \hat{V}_n(G'_z) \right\}.$$

onde, $\hat{V}_n(G_z) := \sum_{s \in G_z} \mu_z(s) v_n(s) \; \forall G_z$

A combinação do TABA com o LSPSI melhora bastante o tempo computacional na solução de SSMDP tanto quando comparado com os algoritmos mais tradicionais (por exemplo: PI e VI), como quando comparado com o TABA e o LSPSI aplicados isoladamente. No próximo capítulo será feita uma comparação entre todos os algoritmos, aplicando-os em um problema envolvendo a otimização de toda a cadeia logística de mineração. Os benefícios do TABA, LSPSI e do LSPSI+TABA ficarão claros neste Capítulo 4.

Algoritmo 9: Algoritmo TABA + LSPSI

Entrada: Defina $n = 0$, uma função de custo inicial v_0, uma partição do estado de estados (S) em Z conjuntos disjuntos (G_z), uma política inicial arbitrária $\pi_0 \in \Pi$, um fator de desconto arbitrário $\lambda \in [0, 1)$ e uma tolerância arbitrária ϵ

Saída: A solução ótima para a função de custo v^*, e a política ótima estacionária π^*

repita

Passo de avaliação de política modificado

$n\_eval \leftarrow 0$; $v_{n\_eval} \leftarrow v_n$; $\pi_{n\_eval} \leftarrow \pi_n$;

repita

para *cada* G_z **faça**

$$\hat{V}_{n\_eval}(G_z) := \sum_{s \in G_z} \mu_z(s) v_{n\_eval}(s)$$

fim

para *cada* $s \in S$ **faça**

Sorteie $H(s) \subset A(s)$; $D(s) \leftarrow H(s) \cup \pi_{n\_eval}(s)$. Calcule:

$$v_{n\_eval+1}(s) := \max_{a \in D(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} p_G(G'_z|s,a) \hat{V}_{n\_eval}(G'_z) \right\}$$

$$\pi_{n\_eval+1}(s) \in \arg\max_{a \in D(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} p_G(G'_z|s,a) \hat{V}_{n\_eval}(G'_z) \right\}$$

fim

$n\_eval \leftarrow n\_eval + 1$;

até $\|v_{n\_eval} - v_{n\_eval-1}\|_\infty \leq \epsilon$;

Passo de atualização de política

$v_n \leftarrow v_{n\_eval}$;

para *cada* G_z **faça**

$$\hat{V}_n(G_z) := \sum_{s \in G_z} \mu_z(s) v_n(s)$$

fim

para *cada* $s \in S$ **faça**

Calcule:

$$v_{n+1}(s) := \max_{a \in D(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} p_G(G'_z|s,a) \hat{V}_n(G'_z) \right\}$$

$$\pi_{n+1}(s) \in \arg\max_{a \in D(s)} \left\{ r(s,a) + \lambda \sum_{z=1}^{Z} p_G(G'_z|s,a) \hat{V}_n(G'_z) \right\}$$

fim

$n \leftarrow n + 1$;

até $\|v_n - v_{n-1}\| \leq \epsilon$;

Defina $\pi^* = \pi_n$ e $v^* = v_n$

retorna v^*, π^*

Capítulo 4

Resultados

Após a apresentação da modelagem MDP para otimização da cadeia de suprimentos na indústria da mineração, envolvendo todos os seus elos, e os novos algoritmos de programação dinâmica, neste capítulo, estes algoritmos serão usados para resolver um exemplo de otimização estocástica na cadeia de mineração. Com os resultados obtidos no exemplo, ficará mais clara a importância de serem considerados todos os elos da cadeia na sua otimização global. Também será possível observar a relevância de se considerar as incertezas existente neste processo e o impacto nas decisões ótimas. Por outro lado, aplicando os novos algoritmos propostos, será possível comparar suas performances frente aos tradicionais algoritmos já existentes para otimização estocástica de longo prazo.

4.1 O exemplo da cadeia de suprimentos na mineração

Para modelar a cadeia logística da indústria da mineração foi utilizado o modelo apresentado na Seção 3.1. Neste exemplo, cada período de tempo t representa um mês. Foram utilizados os seguintes valores para os parâmetros fixos:

- Capacidade de produção e transporte ferroviário das minas até o porto (c^1): 13.000 toneladas por mês;
- Custo de produção (c^2): \$12,00 por 1.000 toneladas;
- Capacidade de armazenagem no porto (c^3): 3.000 toneladas;
- Capacidade de armazenagem no PAA (c^4): 2.000 toneladas;
- Custo do transporte marítimo (navegação) local (c^5): \$1,00 por 1.000 toneladas;

- Preço de venda para os clientes com contrato (c^6): \$60,00 por 1.000 toneladas;
- Penalidade contratual por não atendimento da demanda (c^7): \$100,00 por 1.000 toneladas não atendidas;
- Produção mínima nas minas (c^8): 8.000 toneladas por mês.

As variáveis aleatórias representando os parâmetros com incerteza foram modelados através de variáveis discretas com as seguintes probabilidades, onde $p(x)$ significa "*probabilidade que x ocorra*":

- Variável aleatória que representa a capacidade de fluxo operacional no porto (ξ_t^1):

$$\xi_t^1 \in \{10; 11; 12\} \cdot 1.000 \text{ toneladas/mês}; \begin{cases} p(10) = 20\% \\ p(11) = 60\% \\ p(12) = 20\% \end{cases}$$

- Variável aleatória que representa a capacidade de fluxo operacional no PAA (ξ_t^2):

$$\xi_t^2 \in \{2; 3\} \cdot 1.000 \text{ toneladas/mês}; \begin{cases} p(2) = 40\% \\ p(3) = 60\% \end{cases}$$

- Variável aleatória que representa a demanda dos clientes com contrato (ξ_t^3):

$$\xi_t^3 \in \{8; 9\} \cdot 1.000 \text{ toneladas/mês}; \begin{cases} p(8) = 50\% \\ p(9) = 50\% \end{cases}$$

- Variável aleatória que representa o preço do mercado spot (ξ_t^4):

$$\xi_t^4 \in \{\$30; \$60; \$90\} \text{ por } 1.000 \text{ toneladas}; \begin{cases} p(\$30) = 25\% \\ p(\$60) = 50\% \\ p(\$90) = 25\% \end{cases}$$

- Variável aleatória que representa o custo da navegação internacional partindo do porto (ξ_t^5):

$$\xi_t^5 \in \{\$16; \$18; \$20\} \text{ por } 1.000 \text{ toneladas}; \begin{cases} p(\$16) = 20\% \\ p(\$18) = 60\% \\ p(\$20) = 20\% \end{cases}$$

O resumo destes parâmetros organizados por elo da cadeia estão descritos na Tabela 4.1.

Elo da cadeia	**Valores dos parâmetros**
Minas	Cap. de Produção: $c^1 = 13\ kt$ $(kt = 1.000t)$
	Custo de Prod.: $c^2 = \$12/kt$
	Prod. Mínima: $c^8 = 8\ kt$
Porto e PAA	Cap. Armaz. no Porto: $c^3 = 3\ kt$
	Cap. Fluxo Oper. no Porto: $\xi_t^1 \in \{10; 11; 12\}\ kt/mês; \begin{cases} p(10) = 20\% \\ p(11) = 60\% \\ p(12) = 20\% \end{cases}$
	Cap. Armaz. no PAA: $c^4 = 2\ kt$
	Cap. Fluxo Oper. no PAA: $\xi_t^2 \in \{2; 3\}\ kt/mês; \begin{cases} p(2) = 40\% \\ p(3) = 60\% \end{cases}$
Navegação	Custo Internacional: $\xi_t^5 \in \{\$16; \$18; \$20\}\ /kt; \begin{cases} p(\$16) = 20\% \\ p(\$18) = 60\% \\ p(\$20) = 20\% \end{cases}$
	Custo local: $c^5 = \$1/kt$
Clientes	Preço de Contrato: $c^6 = \$60/kt$
	Demanda de Contrato: $\xi_t^3 \in \{8; 9\}\ kt/mês; \begin{cases} p(8) = 50\% \\ p(9) = 50\% \end{cases}$
	Penalidade de Contrato: $c^7 = \$100/kt$
	Preço Mec. Spot: $\xi_t^4 \in \{\$30; \$60; \$90\}\ /kt; \begin{cases} p(\$30) = 25\% \\ p(\$60) = 50\% \\ p(\$90) = 25\% \end{cases}$

Tabela 4.1: Valores dos Parâmetros Utilizados no Modelo MDP

A modelagem proposta apresenta algumas similaridades com a proposta em [103]. Ambos os modelos separam os clientes em clientes com contrato e mercado spot e também modelam a cadeia de suprimentos como um fluxo dinâmico. Entretanto, existem algumas diferenças dignas de nota. Por exemplo, [103] não inclui navegação e o PAA em sua cadeia logística; por outro lado, o modelo apresentado neste trabalho não distingue os produtos de diferentes minas individualmente.

Nas próximas seções, esse exemplo será resolvido utilizando diversos algoritmos de programação estocástica. Na próxima seção e na seguinte, serão comparadas as performances entre os algoritmos e na Seção 4.4 serão avaliadas as decisões ótimas obtidas.

4.2 Avaliação dos α's no LSPSI e no LSPSI+TABA

Antes de fazer a comparação entre o LSPSI+TABA e os demais algoritmos, se faz necessário determinar o valor de α tanto no LSPSI+TABA quanto no LSPSI puro. O parâmetro α determina o percentual das ações sorteadas para teste na etapa de

avaliação de política. Quando $\alpha = 0$ os algoritmos se aproximam do algoritmo de iteração de políticas, não havendo melhoria no passo de avaliação de política. Já quando $\alpha = 1$ os algoritmos se aproximam do algoritmo de iteração de valor, pois em cada iteração do passo modificado de avaliação de política ele faz uma busca por melhoria em todo o espaço de ações. Desta forma, é de se esperar que para valores de α próximos a zero, o passo de avaliação de política seja mais rápido entretanto sejam necessários mais passos de melhoria de política, dado que poucas novas ações serão testadas no passo modificado de avaliação. Por outro lado, para valores mais altos de α, o passo de avaliação de política modificado se torna mais demorado mas espera-se que a quantidade de passos de melhoria de política seja menor. Também é possível observar pela Eq. (3.23) que mesmo valores pequenos de α já significam uma grande quantidade de políticas avaliadas, quando $|S|$ é grande.

Para comparação, foram testados os seguintes valores de α: $0,1\%$, $1,0\%$, $5,0\%$, 10% e 20%. Também foram avaliados os impactos para três valores diferentes de λ: 0,90, 0,95 e 0,99. Como os algoritmos LSPSI e LSPSI+TABA utilizam sorteio, para avaliar sua performance eles foram rodados diversas vezes com diferentes sementes no sorteio. Para cada combinação de λ e α os resultados obtidos com as diversas sementes foram avaliados:

- A média das quantidade de iterações até a convergência ("Qtde. Iterações - Média");
- O desvio padrão (DP) da quantidade de iterações até a convergência ("Qtde. Iterações - DP");
- O Intervalo de Confiança (IC) com 95% para a quantidade de iterações até a convergência ("Qtde. Iterações - IC (95%)");
- A média do tempo total (em segundos) até a convergência ("Tempo Total (s) - Média");
- O desvio padrão (DP) do tempo total (em segundos) até a convergência ("Tempo Total (s) - DP");
- O Intervalo de Confiança (IC) com 95% do tempo total (em segundos) até a convergência ("Tempo Total (s) - IC (95%)");

Inicialmente, a Tabela 4.2 apresenta os resultados de diferentes α's apenas para o algoritmo LSPSI puro. Foram utilizadas 50 sementes diferentes. Como era previsto, a quantidade de iterações diminui conforme α aumenta. Para $\lambda = 0,90$ o menor tempo total é obtido com $\alpha = 1,0\%$. Já para $\lambda = 0,95$ e $\lambda = 0,99$ o menor tempo total ocorre com $\alpha = 0,1\%$.

λ			α's				
			0,1%	**1,0%**	**5,0%**	**10%**	**20%**
0,90	Qtde. Iterações	Média	110,1	55,1	37,5	32,0	28,3
		DP	5,8	2,5	0,9	0,7	0,8
		IC (95%)	[98,8; 121,5]	[50,2; 60,0]	[35,7; 39,4]	[30,6; 33,4]	[26,8; 29,8]
	Tempo Total (s)	Média	166,1	**117,6**	166,4	219,7	313,6
		DP	11,5	13,5	3,1	3,9	18,9
		IC (95%)	[143,5; 188,7]	[91,1; 144,2]	[160,4; 172,4]	[212,0; 227,5]	[276,5; 350,6]
0,95	Qtde. Iterações	Média	129,3	68,1	46,7	41,0	37,7
		DP	5,0	1,5	0,9	0,4	0,7
		IC (95%)	[119,5; 139,0]	[65,2; 71,0]	[45,0; 48,4]	[40,1; 41,8]	[36,4; 39,0]
	Tempo Total (s)	Média	**171,2**	184,4	237,0	266,2	404,7
		DP	23,6	15,7	24,4	15,3	23,1
		IC (95%)	[124,9; 217,4]	[153,6; 215,2]	[189,2; 284,7]	[236,2; 296,2]	[359,3; 450,1]
0,99	Qtde. Iterações	Média	199,3	112,6	91,0	88,7	87,8
		DP	2,7	1,1	0,0	0,7	0,4
		IC (95%)	[194,0; 204,6]	[110,4; 114,8]	[91,0; 91,0]	[87,2; 90,1]	[87,1; 88,6]
	Tempo Total (s)	Média	**149,1**	165,4	289,4	504,2	877,7
		DP	13,5	21,5	2,8	35,7	57,2
		IC (95%)	[122,6; 175,6]	[123,2; 207,6]	[283,9; 294,9]	[434,1; 574,2]	[765,6; 989,8]

Tabela 4.2: Comparação entre α's no LSPSI

Já a Tabela 4.3 apresenta os resultados de diferentes α's para o algoritmo LSPSI+TABA. Como o tempo do LSPSI+TABA é menor, foram utilizadas 100 sementes na avaliação. Também é possível verificar que conforme α aumenta, a quantidade total de iterações reduz, entretanto o tempo total aumenta. Para o LSPSI+TABA, os menores tempos totais para todos os λ's são obtidos para $\alpha = 0,1\%$.

Como exemplo, a Figura 4.1 mostra a convergência do LSPSI+TABA para uma semente, os três valores de λ e os cinco valores avaliados de α. No eixo y é apresentado a variação da função de custo entre duas iterações consecutivas (delta da função de custo), calculado como o logaritmo da norma superior percentual. No eixo x, é representada a evolução do tempo. Foi considerado como critério de parada a maior variação percentual da função de custo entre todos os estados ser inferior a 10^{-4}. Quando o delta da função de custo sobe, significa que o passo de melhoria de política encontrou uma solução melhor. Nesse gráfico fica mais claro que com

λ			α's				
			0,1%	**1,0%**	**5,0%**	**10%**	**20%**
0,90	Qtde. Iterações	Média	118,7	59,6	40,0	34,3	30,7
		DP	6,6	1,9	0,7	0,6	0,5
		IC (95%)	[105,8; 131,6]	[55,8; 63,3]	[38,6; 41,4]	[33,2; 35,5]	[29,7; 31,6]
	Tempo Total (s)	Média	**5,8**	6,5	10,3	14,1	21,7
		DP	0,7	0,7	0,6	0,8	1,4
		IC (95%)	[4,3; 7,2]	[5,2; 7,8]	[9,1; 11,5]	[12,5; 15,8]	[19,1; 24,4]
0,95	Qtde. Iterações	Média	130,6	69,1	45,6	40,1	37,0
		DP	5,2	1,4	0,6	0,4	0,2
		IC (95%)	[120,5; 140,7]	[66,4; 71,8]	[44,4; 46,8]	[39,3; 40,8]	[36,5; 37,4]
	Tempo Total (s)	Média	**6,2**	7,1	11,3	16,5	25,2
		DP	0,6	0,4	0,9	1,0	1,8
		IC (95%)	[4,9; 7,4]	[6,2; 7,9]	[9,5; 13,1]	[14,6; 18,4]	[21,6; 28,8]
0,99	Qtde. Iterações	Média	204,1	116,0	95,0	93,0	92,0
		DP	2,8	0,8	0,1	0,0	0,0
		IC (95%)	[198,6; 209,5]	[114,4; 117,6]	[94,8; 95,2]	[93,0; 93,0]	[92,0; 92,0]
	Tempo Total (s)	Média	**5,8**	8,6	19,3	32,6	56,9
		DP	0,4	1,1	1,2	1,9	3,7
		IC (95%)	[5,1; 6,5]	[6,5; 10,7]	[17,0; 21,7]	[28,9; 36,2]	[49,7; 64,1]

Tabela 4.3: Comparação entre α's no LSPSI+TABA

valores menores de α a convergência do passo de avaliação de política modificado é mais rápida, porém o algoritmo necessita de mais passos de melhoria de política para convergir. Para valores de α maiores, a convergência da avaliação de política modificada é mais lenta, mas por outro lado a necessidade de passos de melhoria de política é menor. Pelos gráficos, é possível perceber que, mesmo precisando de mais passos de melhoria de política, a convergência final para $\alpha = 0,1\%$ é mais rápida para os três valores de λ.

Na próxima seção, será feita a comparação entre os diversos algoritmos de programação dinâmica utilizando o valor de $\alpha = 0,1\%$ tanto para o LSPSI como para o LSPSI+TABA.

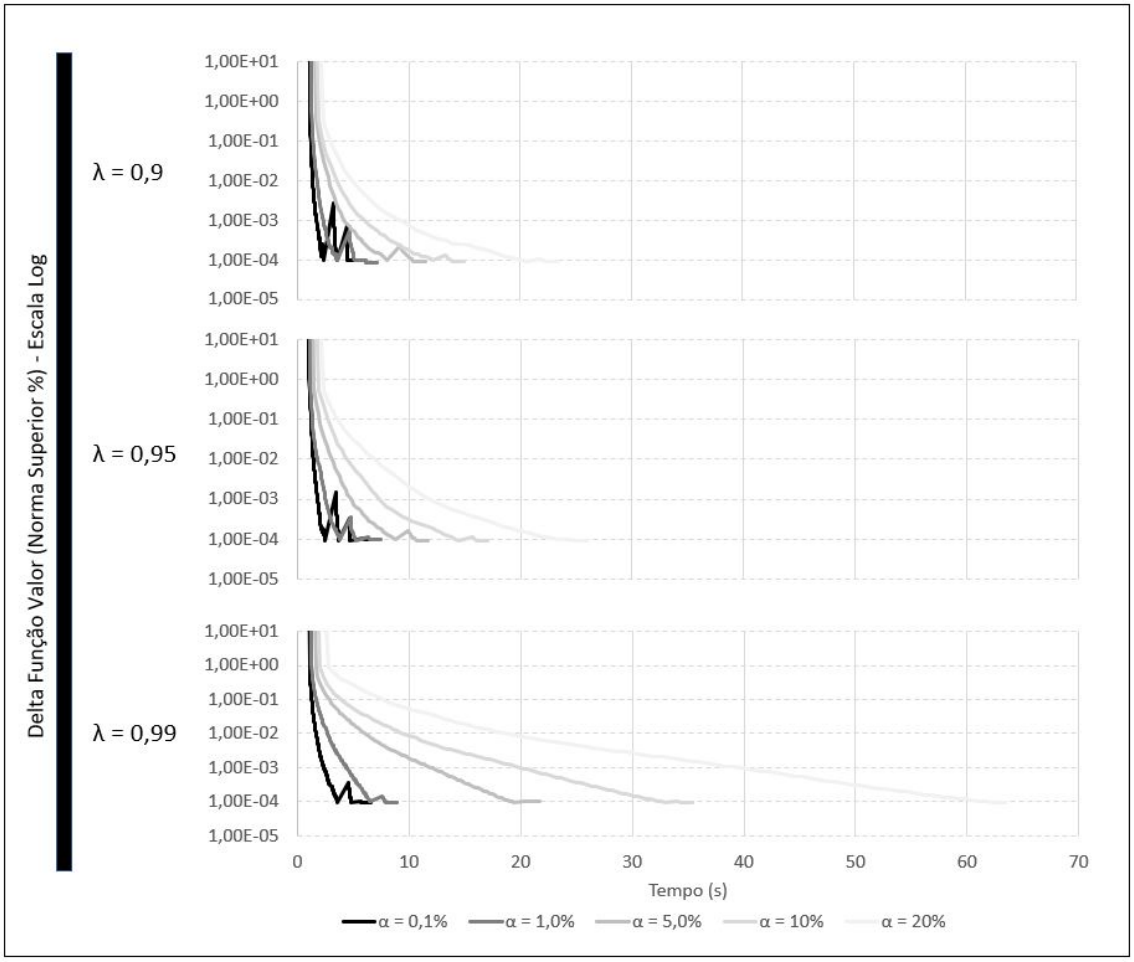

Figura 4.1: Convergência do LSPSI+TABA em função dos λ's e α's

4.3 Avaliação dos algoritmos

Para obter as decisões ótimas para o problema descrito na Seção 4.1, foram testados sete algoritmos diferentes:

1. Iteração de valor clássico com Gauss-Seidel (VI) [4];
2. Iteração de política (PI) [4]
3. Iteração de conjunto de políticas (PSI) CHANG [21];
4. Programação dinâmica dual estocástica (SDDP) PEREIRA e PINTO [192];
5. Iteração de grupo de políticas com busca local (LSPSI) com $\alpha = 0,1\%$;
6. Algoritmo baseado na agregação temporal (TABA);
7. Combinação do LSPSI com o TABA (LSPSI+TABA) com $\alpha = 0,1\%$

Considerando todas as combinações possíveis do problema estudado, existem 1.296 estados possíveis e avaliando todos os estados iniciais, as decisões possíveis e os estados finais, há mais de 1,5 milhões de transições viáveis. Os algoritmos foram testados utilizando-se três valores diferentes para o fator de desconto (λ): $0,9$, $0,95$ e $0,99$. Todos os modelos foram executados em um computador pessoal com processador Intel® Core™ i7-7500 CPU, @2.70GHz 2.9GHz com 8,00GB de memória RAM e usando o sistema operacional Windows 10.

Como era de se esperar, todos os algoritmos convergiram para as mesmas políticas ótimas para todos os valores λ. O resultado comparativo de desempenho dos algoritmos é apresentado na Tabela 4.4. Para todos os algoritmos, com exceção do SDDP foi utilizado como critério de convergência a variação percentual da função de custo ser inferior a 10^{-4}.

λ		**VI**	**PI**	**PSI**	**SDDP**	**LSPSI**	**TABA**	**LSPSI + TABA**
0,90	Qtde. Iterações	23	6	4	17	110,1	27	118,7
	Tempo Iter. (s)	50,0	49,7	71,8	0,83	1,51	1,17	0,05
	Tempo Tot. (s)	1.151	298	287	14,1	166,1	31,5	**5,8**
0,95	Qtde. Iterações	33	8	5	7	129,3	34	130,6
	Tempo Iter. (s)	64,1	61,8	97,6	1,93	1,32	1,32	0,05
	Tempo Tot. (s)	2.117	495	488	13,5	171,2	44,8	**6,2**
0,99	Qtde. Iterações	85	11	5	15	199,3	92	204,1
	Tempo Iter. (s)	55,8	54,2	206	1,35	0,75	1,24	0,03
	Tempo Tot. (s)	4.743	596	1.034	20,3	149,1	114	**5,8**

Tabela 4.4: Comparação entre os algoritmos

Os algoritmos de VI e PI foram os que precisaram de mais tempo até convergir. Na quantidade de iterações da Tabela 4.4, foram computadas apenas as iterações de melhoria de política tanto no PI como no PSI. No VI, o tempo de cada iteração é bastante longo, dado que ele avalia todas as ações possíveis. No PI o tempo é semelhante, dado que cada passo de melhoria de política também avalia todas as ações possíveis para cada estado. Como o número de iterações de melhoria de política no PI é menor do que a quantidade de iterações no VI, o tempo total do PI é menor que o do VI para todos os λ's.

O algoritmo PSI [21], base para o desenvolvimento do LSPSI, também foi implementado para servir como outra fonte de comparação. Na implementação utilizada, em cada passo de avaliação de política são testadas 10 políticas adicionais aleatoriamente. Como também era esperado, a quantidade de iterações no PSI é menor quando comparada com a do PI para todos os λ's. Entretanto, como o passo de avaliação de política se torna mais complexo, dado que testa mais políticas aleatórias, o ganho de tempo total do PSI comparado com o PI não é significativo para

$\lambda = 0,90$ e $\lambda = 0,95$. Por outro lado, o tempo total do PSI é maior que o do PI para $\lambda = 0,99$

Como o problema apresenta restrições lineares, sua curva de custo é convexa e não utiliza variáveis inteiras, também é possível resolvê-lo através do SDDP. Caso uma dessas premissas fosse quebrada, o SDDP não poderia mais ser aplicado. Para sua implementação, foi usada a biblioteca SDDP na linguagem de programação Julia [259] adaptada para problemas de horizonte infinito conforme [260]. Considerou-se como critério de atingimento de convergência variação de 10^{-4} entre os limites (*bounds*). Comparando com o VI, PI e PSI, o SDDP apresenta uma performance muito boa atingindo a convergência em um tempo substancialmente inferior, com uma quantidade de iterações semelhante porém um tempo médio por iteração muito inferior.

O primeiro algoritmo proposto a ser testado é o LSPSI com $\alpha = 1,0\%$. Quando comparado com o VI, PI e PSI, o LSPSI apresenta uma performance melhor, tendo um tempo total inferior. O ganho de tempo é superior a 50% comparado ao melhor entre os três algoritmos para $\lambda = 0,95$ e $\lambda = 0,99$. A quantidade total de iterações é maior, porém o tempo médio por iteração é inferior. Por outro lado, quando comparado ao SDDP, o tempo computacional é maior. O SDDP apresenta um ganho de tempo próximo a 90% para os três λ's.

A configuração do problema permite a caracterização como SSMDP (Definição 1) e a implementação do TABA, agrupando os estados conforme o nível inicial dos estoques, componentes s^6 e s^7. Como as outras componentes do vetor de estado (s^1 a s^5) são independentes e identicamente distribuídos, elas podem fazer parte do subconjunto S_n na Suposição 1. O vetor de distribuição do estado de equilíbrio (μ) é fixo e o mesmo para todos os grupos, podendo ser calculado facilmente. Esta agregação, faz com que o TABA apresente uma performance ainda melhor que o LSPSI, e consequentemente bastante superior ao VI, PI e PSI. O ganho é maior para λ maior. Para $\lambda = 0,90$, a redução do tempo foi de 166,1s para 31,5s. Já para $\lambda = 0,99$ o tempo total foi de 114s para o TABA contra 149,1s para o LSPSI. Comparando com o SDDP, o TABA apresentou tempo computacional maior para todos os λ's. A performance do TABA piora significativamente quando o valor de λ aumenta. Para $\lambda = 0,90$, o SDDP converge em 14,1s, enquanto que o TABA precisa de 31,5s. Já para $\lambda = 0,99$, o SDDP precisa de 20,3s e o TABA 114s.

Finalmente, os melhores tempos computacionais foram obtidos com a combinação dos dois algoritmos propostos: LSPSI e o TABA (LSPSI+TABA). Comparando com os algoritmos tradicionais (VI, PI e PSI) o ganho computacional é superior a 95% sendo mais significativo para valores maiores de λ. Para $\lambda = 0,99$ o PI precisou de 596s e o LSPSI+TABA de apenas 5,8s. O LSPSI+TABA necessita de mais iterações do que os algoritmos tradicionais, porém graças às reduções no espaço de

busca de estados e de ações, o tempo de cada iteração é consideravelmente inferior, sendo da ordem de grandeza de 10^{-2}s.

Mesmo quando comparado com o SDDP, o LSPSI+TABA apresentou uma performance superior. O LSPSI+TABA sofreu menos impacto com o aumento de λ. Comparando com o SDDP, para $\lambda = 0,9$ o LSPSI+TABA teve uma redução de quase 60% do tempo computacional (de 14,1s para 5,8s). Para $\lambda = 0,95$ essa redução é superior a 50% (de 13,5s para 6,2s) e para $\lambda = 0,99$ essa redução é ainda maior atingindo 70% (de 20,3s para 5,8s). O LSPSI+TABA também precisa de mais iterações do que o SDDP, porém o tempo médio por iteração também é menor.

Para finalizar, a Figura 4.2 resume a performance dos sete algoritmos para os três valores de λ. Em cada gráfico, o eixo y representa o tempo computacional total até a convergência e as barras são a performance por algoritmo. Primeiramente, é possível reparar que os algoritmos VI, PI e PSI são bastante sensíveis ao aumento do λ. Nos gráficos fica ainda mais nítido o ganho computacional obtido com o LSPSI+TABA, comparando com os demais algoritmos.

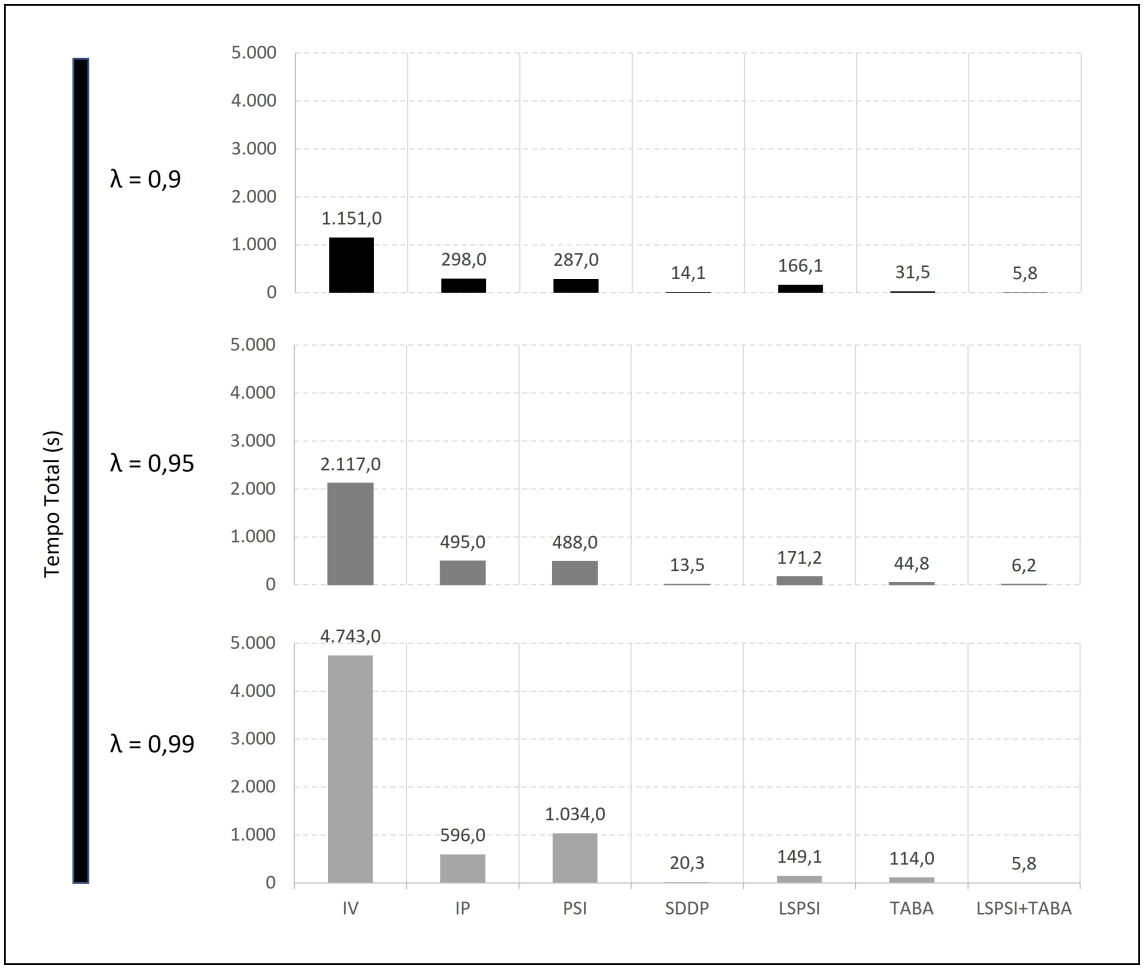

Figura 4.2: Comparativo entre os algoritmos

4.4 Decisões ótimas

Nesta seção serão analisadas as decisões ótimas obtidas e seus impactos na cadeia logística. Será usada a mesma nomenclatura da Seção 3.1. O vetor de estados s possui 7 componentes:

- Capacidade de fluxo operacional no porto (s_t^1);
- Capacidade de fluxo operacional no PAA (s_t^2);
- Demanda dos clientes com contrato (s_t^3);
- Preço do mercado spot (s_t^4);
- Custo da navegação internacional partindo do porto (s_t^5);
- Volume armazenado no porto (s_t^6);
- Volume armazenado no PAA (s_t^7).

O vetor de decisão a e a política ótima para o estado s ($\pi^*(s)$) possuem 5 componentes:

- Volume de produção e transporte ferroviário das minas até o porto (a_t^1);
- Volume transportado do porto para o PAA (a_t^2);
- Volume transportado do porto para os clientes com contrato (a_t^3);
- Volume transportado do porto para o mercado spot (a_t^4);
- Volume transportado do PAA para os clientes com contrato (a_t^5);
- Volume transportado do PAA para o mercado spot (a_t^6).

O VPL esperado ($V^*(s)$), estando no estado inicial s é calculado conforme a Equação (3.9), aplicando a política ótima conforme a Equação (3.10).

Abaixo, são apresentados alguns resultados obtidos considerando $\lambda = 0.95$:

- $\boldsymbol{s = [10, 2, 8, 30, 20, 3, 2],\ \pi^*(s) = [8, 0, 8, 0, 0, 0],\ V^*(s) = \$6.884 \cdot 10^3}$: O porto e o PAA iniciam com os estoques cheios. Como o preço do mercado spot é baixo ($30), a decisão ótima é produzir e entregar apenas o volume contratual.
- $\boldsymbol{s = [10, 2, 8, 30, 20, 0, 0],\ \pi^*(s) = [10, 2, 8, 0, 0, 0],\ V^*(s) = \$6.786 \cdot 10^3}$: O porto e o PAA começam com seus estoques vazios e o preço do mercado spot é baixo ($30). A decisão é produzir o volume máximo de capacidade ($10kt$), atender o volume contratual ($8kt$) e armazenar $2kt$ no PAA.

- $\boldsymbol{s = [10, 2, 8, 60, 20, 0, 0],\ \pi^*(s) = [10, 0, 8, 2, 0, 0],\ V^*(s) = \$6.792 \cdot 10^3}$: O porto e o PAA começam com seus estoques vazios e o preço do mercado spot é médio ($60). Comparando com o exemplo anterior, em vez de armazenar $2kt$ no PAA, esse volume é direcionado para venda no mercado spot.

- $\boldsymbol{s = [10, 3, 8, 90, 20, 3, 2],\ \pi^*(s) = [8, 0, 6, 4, 2, 0],\ V^*(s) = \$7.066 \cdot 10^3}$: O porto e o PAA iniciam com os estoques cheios e o preço do mercado spot é alto ($90). Como o limite de volume no porto é $10kt$ e há $2kt$ no PAA, o sistema só consegue entregar $12kt$, atendendo as $8kt$ do contrato e entregando $4kt$ para o mercado spot, dado seu preço alto. Para atender a produção mínima e dado o limite de escoamento priorizado para a entrega a cliente, $1kt$ termina o período armazenado no porto.

- $\boldsymbol{s = [12, 3, 8, 90, 16, 3, 2],\ \pi^*(s) = [9, 0, 6, 6, 2, 0],\ V^*(s) = \$7.230 \cdot 10^3}$: O porto e o PAA iniciam com os estoques cheios e o preço do mercado spot é alto ($90). Como o limite de volume no porto é $12kt$ e há $2kt$ no PAA, o sistema só consegue entregar $14kt$, atendendo as $8kt$ do contrato e entregando $6kt$ para o mercado spot, dado seu preço alto. Também dado o limite no porto de $12kt$ e como já há um volume inicial estocado de $3kt$, apenas $9kt$ são produzidas. Devido ao preço alto do mercado spot e à alta capacidade de volume no porto, esse é o maior VPL.

- $\boldsymbol{s = [12, 3, 8, 90, 16, 0, 2],\ \pi^*(s) = [12, 0, 6, 6, 2, 0],\ V^*(s) = \$7.194 \cdot 10^3}$: O PAA inicia com o estoque cheio, o porto vazio e o preço do mercado spot é alto ($90). Como o estoque no porto está vazio, a produção é equivalente ao limite no porto de $12kt$. Como o limite de volume no porto é $12kt$ e há $2kt$ no PAA, o sistema só consegue entregar $14kt$, atendendo as $8kt$ do contrato e entregando $6kt$ para o mercado spot, dado seu preço alto.

Variando o lambda para $\lambda = 0.99$, as decisões ótimas são as mesmas, apenas o VPL aumenta, dado que a taxa de desconto é menor. Como o peso do curto prazo diminui, a diferença percentual entre os VPLs dos diferentes cenários iniciais também fica menor:

- $\boldsymbol{s = [10, 2, 8, 30, 20, 3, 2],\ \pi^*(s) = [8, 0, 8, 0, 0, 0],\ V^*(s) = \$34.434 \cdot 10^3}$;
- $\boldsymbol{s = [10, 2, 8, 30, 20, 0, 0],\ \pi^*(s) = [10, 2, 8, 0, 0, 0],\ V^*(s) = \$34.331 \cdot 10^3}$;
- $\boldsymbol{s = [10, 2, 8, 60, 20, 0, 0],\ \pi^*(s) = [10, 0, 8, 2, 0, 0],\ V^*(s) = \$34.332 \cdot 10^3}$;
- $\boldsymbol{s = [10, 3, 8, 90, 20, 3, 2],\ \pi^*(s) = [8, 0, 6, 4, 2, 0],\ V^*(s) = \$34.606 \cdot 10^3}$;
- $\boldsymbol{s = [12, 3, 8, 90, 16, 3, 2],\ \pi^*(s) = [9, 0, 6, 6, 2, 0],\ V^*(s) = \$34.770 \cdot 10^3}$;

- $\boldsymbol{s = [12, 3, 8, 90, 16, 0, 2],\ \pi^*(s) = [12, 0, 6, 6, 2, 0],\ V^*(s) = \$34.734 \cdot 10^3}$.

Variando o lambda para $\lambda = 0.90$, as decisões ótimas continuam as mesmas, porém o VPL diminui, dado que a taxa de desconto é maior:

- $\boldsymbol{s = [10, 2, 8, 30, 20, 3, 2],\ \pi^*(s) = [8, 0, 8, 0, 0, 0],\ V^*(s) = \$3.435 \cdot 10^3}$;
- $\boldsymbol{s = [10, 2, 8, 30, 20, 0, 0],\ \pi^*(s) = [10, 2, 8, 0, 0, 0],\ V^*(s) = \$3.339 \cdot 10^3}$;
- $\boldsymbol{s = [10, 2, 8, 60, 20, 0, 0],\ \pi^*(s) = [10, 0, 8, 2, 0, 0],\ V^*(s) = \$3.351 \cdot 10^3}$;
- $\boldsymbol{s = [10, 3, 8, 90, 20, 3, 2],\ \pi^*(s) = [8, 0, 6, 4, 2, 0],\ V^*(s) = \$3.624 \cdot 10^3}$;
- $\boldsymbol{s = [12, 3, 8, 90, 16, 3, 2],\ \pi^*(s) = [9, 0, 6, 6, 2, 0],\ V^*(s) = \$3.789 \cdot 10^3}$;
- $\boldsymbol{s = [12, 3, 8, 90, 16, 0, 2],\ \pi^*(s) = [12, 0, 6, 6, 2, 0],\ V^*(s) = \$3.753 \cdot 10^3}$.

Também é importante perceber que o problema apresenta algumas soluções múltiplas. Observe o 4º exemplo acima: $\boldsymbol{s = [10, 3, 8, 90, 20, 3, 2],\ \pi^*(s) = [8, 0, 6, 4, 2, 0]}$. Como já foi observado, o sistema só consegue entregar $12kt$, atendendo as $8kt$ do contrato e entregando $4kt$ para o mercado spot. Dada as limitações de vazão no porto e no PAA, destes $12kt$, $10kt$ partirão do porto e $2kt$ do PAA. Há várias soluções com o mesmo custo que obedecem essa restrição. Na decisão escolhida pelo algoritmo, o porto envia $6kt$ para os clientes de contrato e $4kt$ para os clientes spot. Já o PAA envia $2kt$ para os clientes de contrato. Uma solução equivalente ($\boldsymbol{\pi^*(s) = [8, 0, 8, 2, 0, 2]}$) seria o porto enviar $8kt$ para os clientes de contrato e $2kt$ para os clientes spot, e o PAA enviar $2kt$ para os clientes spot. Ambas as soluções são equivalentes:

- As restrições de vazão no porto e no PAA são atendidas;
- São entregues $8kt$ para os clientes em contrato e $4kt$ para os clientes spot, resultando em receitas idênticas;
- São entregues $10kt$ a partir do porto e $2kt$ a partir do PAA, resultando em custo de frete de navegação iguais;
- O sistema termina com a mesma situação de estoque final: $1kt$ no porto e $0kt$ no PAA.

Neste mesmo exemplo há outras combinações que também resultariam em soluções equivalentes, como por exemplo: $\boldsymbol{\pi^*(s) = [8, 0, 7, 3, 1, 1]}$. Dada esta flexibilidade de atendimento a partir do porto e a partir do PAA, no dia-a-dia, esta a escolha poderia ser feita pela própria operação caso algum cliente solicite alguma entrega emergencial ou beneficiando algum cliente com maior relacionamento comercial.

Finalmente, analisando os resultados e as políticas obtidas, é possível obter algumas conclusões sobre este problema:

- As decisões ótimas não apresentam grandes variações em função de diferentes valores de λ;
- Para alguns estados, o sistema apresenta soluções múltiplas;
- Como a penalidade contratual é maior do que o preço *spot* mais caro, o volume contratual é sempre atendido;
- A capacidade de fluxo operacional no porto e no PAA são o grande gargalo da cadeia logística;
- Olhando especialmente para o porto, sua capacidade de fluxo operacional é inferior à capacidade de produção das minas, então o sistema não prioriza a formação de estoques no porto;
- Quando o preço spot é baixo ($30), o volume contratual é completamente atendido, porém os produtos excedentes não são enviados para o mercado spot. Eles são usados para recompor os estoques no porto e no PAA;
- Já quando o preço spot chega a $60 ou $90, os produtos restantes após o atendimento à demanda contratual são direcionados para venda no mercado spot;
- Como o preço de mercado e duas alternativas do mercado spot são lucrativos, superando os custos de produção, a cadeia é lucrativa e apresenta VPL maior do que zero;
- O volume produzido é altamente dependente da capacidade de vazão no porto e do preço do mercado spot, assim seu planejamento não deve ser feito de forma isolada e deve considerar os demais elos da cadeia.

Capítulo 5

Conclusão

Em um mundo globalizado, dinâmico e com mercados extremamente competitivos, as empresas buscam o máximo de eficiência em seus processos. Na indústria da mineração, que possui um papel importantíssimo tanto no Brasil como globalmente, não é diferente. As empresas procuram a melhor forma de gerenciar toda a cadeia de suprimentos, desde as minas até os clientes finais, maximizando seus lucros.

Neste contexto, ferramentas que auxiliem às tomadas de decisões são muito bem vindas. Entre elas, destaca-se a otimização estocástica pela capacidade de sugerir as melhores decisões, considerando as incertezas existentes em todo o processo. Dada a natureza sequencial do problema, MDP se apresenta como uma alternativa bastante promissora porém, devido à complexidade da cadeia da mineração, sua modelagem e sua convergência ainda são pontos a serem resolvidos. Como modelar toda a cadeia de mineração em um MDP? Como garantir que a solução ótima de um MDP tão complexo seja obtida de forma rápida?

Analisando os trabalhos de otimização estocástica na industria da mineração, é possível perceber que a maioria deles se concentra em apenas um elo da cadeia e poucos cobrem das minas até os clientes finais (Figura 2.2). Estes trabalhos buscam a otimização local dos elos envolvidos, contudo não garantem a otimização global da cadeia. Também é interessante notar, que apesar da natureza sequencial de longo prazo, a maioria dos trabalhos utiliza modelos estocásticos de curto prazo, pois estes modelos apresentam convergência mais rápida. Neste contexto, o primeiro objetivo desta tese foi criar um modelo que englobe todos os elos da cadeia logística de mineração e que também leve em consideração sua característica de longo prazo. Por este motivo, foi escolhida a modelagem MDP.

Para conseguir cumprir este primeiro objetivo, além da modelagem, é necessário que o algoritmo utilizado consiga a convergência para a decisão ótima global em um período de tempo factível. De forma geral, os algoritmos de otimização estocástica enfrentam a *maldição da dimensionalidade*, pois sua convergência se torna exponencialmente mais difícil conforme o modelo se torna maior, com mais incer-

tezas, estados e decisões possíveis. Uma das alternativas utilizadas é a utilização de aproximações, que facilitem a convergência dos resultados porém não garantem que a solução encontrada seja a ótima global. Desta forma, o desafio da modelagem da cadeia da mineração só será completamente transposto se houver um algoritmo que permita a solução deste problema tão complexo em um tempo computacional viável. Assim, surge o segundo objetivo desta tese, desenvolver um algoritmo de solução para MDP que consiga obter a decisão ótima global em um rápido intervalo de tempo, considerando toda a complexidade deste problema.

Para cumprir o primeiro objetivo, o modelo MDP proposto engloba todos os principais elos da cadeia:

- Produção nas minas e seu escoamento pela ferrovia;
- Armazenagem e carregamento no porto;
- Navegação internacional até os clientes finais ou porto de armazenagem avançada;
- Armazenagem e carregamento no porto de armazenagem avançada;
- Os dois principais tipos de clientes finais: de contrato ou eventuais (*spot*).

Ele também considera as principais incertezas no processo:

- Capacidade de escoamento no porto;
- Custo da navegação internacional;
- Capacidade de escoamento do porto de armazenagem avançado;
- A demanda dos clientes de contrato;
- O preço do minério no mercado spot.

E permite que sejam escolhidas as decisões ótimas envolvendo:

- Produção e escoamento ferroviário das minas até o porto;
- Volume armazenado e carregado no porto;
- Volume transportado, armazenado e carregado no porto de armazenagem avançada;
- Volume entregue para os clientes de contrato;
- Volume negociado e entregue para os clientes spot.

Estas decisões são tomadas considerando o horizonte de longo prazo (infinito) e maximizam o VPL desta operação, considerando suas receitas e custos totais.

Comparando com os demais trabalhos existentes, esta tese é pioneira em conseguir unir em um único modelo, todos os elos da indústria da mineração, com suas principais incertezas, para que as melhores decisões em cada etapa sejam tomadas garantindo a maximização do VPL total da cadeia.

Para solucionar um modelo tão complexo, com mais de 1,5 milhões de transições viáveis, foi necessário desenvolver um algoritmo que atuasse em duas frentes: redução do espaço de estados e redução do espaço de ações.

Com foco na redução do espaço de estados, esta tese apresentou o TABA, um algoritmo para MDP baseado em agregação temporal para uma classe de MDP com componentes do espaço de estados estacionários (SSMDP). O TABA agrupa os estados de acordo com as componentes de estados não estacionárias, que dependem do estado e ação inicial. Na busca dos próximos estados, o TABA só precisa percorrer estes agrupamentos e ainda assim a convergência para a solução ótima é comprovadamente obtida. Desta forma, ele reduz o espaço de estados a ser pesquisado e converge muito mais rapidamente, quando comparado com algoritmos de programação dinâmica tradicionais. O TABA se torna mais eficiente quando o agrupamento dos estados leva a um número de grupos de estados significativamente menor do que a quantidade total de estados existentes. A aplicação do TABA à cadeia logística da mineração é bastante eficaz dado que, apesar da grande quantidade de incertezas, seus valores são conhecidos no momento da tomada de decisão, caracterizando-os como SSMDP. Como exemplos, temos que apesar das flutuações do preço de venda no mercado spot e do custo da navegação internacional, seus contratos são fechados baseados no preço atual conhecidos no momento da assinatura.

Para atuar na redução do espaço de ações, esta tese propôs uma abordagem baseada na iteração de política, que inclui um passo de avaliação de política modificado com a inclusão de uma busca local de novas políticas. O algoritmo chamado de LSPSI, utiliza a iteração de valor, incluindo uma amostra pequena de novas ações, no passo de avaliação de política. Apesar da amostra usada na busca local ser pequena, considerando a quantidade de estados envolvidos e a quantidade de iterações necessárias do VI até sua convergência, o LSPSI consegue avaliar uma quantidade significativa de políticas, em um tempo bastante rápido, durante o passo de avaliação de política modificado. O passo de melhoria de política garante a otimização global do algoritmo, porém a busca local reduz a sua necessidade, contribuindo para a sua eficiência.

Finalmente, por atuarem em frentes diferentes, a tese combina os dois algoritmos anteriores e introduz o LSPSI+TABA, que atua tanto na redução do espaço de estados quanto no espaço de ações.

O algoritmo LSPSI+TABA, se baseia nos estudos de CAO *et al.* [13], CHANG [21] propondo melhorias inéditas que avançam a fronteira do conhecimento. Quando comparado com o PSI [21], o LSPSI aumenta o poder de busca do passo modificado de avaliação de política, avaliando uma quantidade de políticas muito maior, sem comprometer a eficiência computacional. Desta forma, o LSPSI reduz a quantidade de passos de melhoria de política. Já o TABA avança na fronteira que expande a aplicação dos algoritmos de agregação temporal [13], extrapolando para outros modelos os benefícios inicialmente obtidos apenas para MDP com estados em que há apenas uma ação viável. O TABA trabalha com a redução no espaço de estados, semelhante ao que ocorre na agregação temporal, mas consegue fazê-lo para problemas com várias ações possíveis em todos os estados e que sejam caracterizados como SSMDP. Com essas melhorias propostas e combinando os dois algoritmos, o LSPSI+TABA permite que a otimização de um problema tão complexo como a otimização de toda cadeia de uma empresa de mineração seja obtida em poucos segundos, facilitando a aplicação prática pelas empresas.

Para medir a performance dos algoritmos propostos e compará-los com os algoritmos existentes, eles foram testados em um problema numérico que modela uma cadeia logística de mineração. Neste problema, o LSPSI+TABA pôde ser comparado com o LSPSI e o TABA isoladamente, além dos algoritmos tradicionais de interação de valor e iteração de política, e de algoritmos mais modernos como o PSI e o SDDP. Quando comparado com os algoritmos tradicionais e com o PSI, o LSPSI+TABA é mais de 50 vezes mais rápido que o PI e do que o PSI, e mais de 100 vezes mais rápido que o VI, independentemente do valor adotado para λ. Quando aplicados isoladamente, o TABA apresentou uma melhor performance do que o LSPSI, entretanto o LSPSI+TABA é 6 vezes mais rápido para $\lambda = 0,90$ e 17 vezes mais rápido para $\lambda = 0,99$. Para finalizar, quando comparado com o SDDP, algoritmo que vem obtendo sucesso na programação estocástica apesar de algumas limitações de implementação, o LSPSI+TABA também se mostra mais eficiente, sendo 3 vezes mais rápido para todos os λ's.

Analisando as soluções obtidas em poucos segundos pelo LSPSI+TABA, uma primeira grande vantagem da modelagem MDP pode ser percebida: o modelo fornece as decisões ótimas para todos os estados possíveis, não sendo necessário rodar a solução para cada condição inicial. Desta forma, a política ótima fornecida pode ser aplicada durante vários meses e em simulações de cenários futuros. Também é possível perceber que o volume ótimo produzido nas minas e transportado para o porto, primeiros elos da cadeia, é altamente dependente da capacidade de vazão no porto e do preço do mercado spot, elos mais avançados. Desta forma, um modelo que considera todos os elos, apresenta resultados mais precisos do que aqueles que otimizam apenas um elo isoladamente. Outros resultados mais pontuais também

podem ser confirmados pelo modelo, tais como: as decisões ótimas não são muito dependentes de λ e o fluxo operacional no porto é o grande gargalo.

Avaliando os benefícios conjuntos da modelagem MDP para a cadeia da mineração e a introdução de novos algoritmos, como o LSPSI+TABA, que obtém a solução rapidamente, fica claro que a aplicação da otimização estocástica mesmo em cadeias tão complexas é uma ferramenta viável e que traz ótimos resultados. As respostas podem ser obtidas rapidamente, viabilizando sua aplicação no dia-a-dia das empresas. Suas soluções, que avaliam todas as incertezas existentes, trazem grande valor quando comparadas com os modelos determinísticos. Não é por acaso, que a otimização estocástica é uma área de pesquisa em constante crescimento e aplicabilidade, principalmente nas cadeias logísticas.

Para finalizar, os resultados obtidos nesta tese encorajam a continuidade das pesquisas nas duas frentes, tanto da modelagem MDP, como nos algoritmos LSPSI e TABA. Na modelagem, novas complexidades podem ser introduzidas como por exemplo a variação na capacidade de transporte da ferrovia e a diferença de pureza de minério extraída em diferentes minas. Na literatura de pesquisa operacional, este último tipo de problema onde produtos com diferentes características são combinados na formação do produto final, que deve ter uma qualidade pré-definida, é conhecido como problema de mistura e pode ser um próximo passo de evolução para o modelo proposto. Outro possível caminho é a introdução de uma dependência temporal na evolução dos preços, tanto de venda do minério como o da navegação internacional. Já na frente dos algoritmos, o caminho esperado para o TABA é a busca pelo relaxamento da suposição de estacionaridade parcial (SSMDP), permitindo que o TABA seja aplicado em mais classes de MDP. Para o LSPSI, uma frente que pode trazer bons resultados é a avaliação de diferentes estratégias de amostragem nas ações adicionais avaliadas no passo de avaliação de política modificado. Estratégias mais eficientes, poderão acelerar ainda mais a convergência do algoritmo.

Referências Bibliográficas

[1] LEITE, J., ARRUDA, E., BAHIENSE, L., et al. "Modeling the integrated mine-to-client supply chain: a survey", *International Journal of Mining, Reclamation and Environment*, pp. 1–47, 4 2019. ISSN: 1748-0930. doi: 10.1080/17480930.2019.1579693. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/17480930.2019.1579693>.

[2] CNN BRASIL. "Faturamento do setor de mineração do Brasil sobe 62% em 2021, diz levantamento". 2022. Disponível em: <https://www.cnnbrasil.com.br/business/faturamento-do-setor-de-mineracao-do-brasil-sobe-62-em-2021/-diz-levantamento/>.

[3] STATISTA. "Mining – Statistics & Facts". 2017. Disponível em: <https://www.statista.com/topics/1143/mining/>.

[4] PUTERMAN, M. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. 1st ed. New York, NY, USA, John Wiley & Sons, Inc., 1994. ISBN: 0471619779.

[5] BEYER, D. *Markovian demand inventory models*. New York, NY, Springer, 2010. ISBN: 978-0-387-71604-6.

[6] AXSÄTER, S. *Inventory Control*, v. 225, *International Series in Operations Research & Management Science*. 3rd editio ed. Cham, Springer International Publishing, 2015. ISBN: 978-3-319-15728-3. doi: 10.1007/978-3-319-15729-0.

[7] SNYDER, L., SHEN, Z. *Fundamentals of supply chain theory*. Hoboken, NJ, John Wiley & Sons, 2019. ISBN: 978-1-119-02497-2.

[8] BELLMAN, R. "A Markovian decision process", *Journal of Mathematics and Mechanics*, v. 6, pp. 679–684, 1957.

[9] POWELL, W. "A unified framework for stochastic optimization", *European Journal of Operational Research*, v. 275, n. 3, pp. 795–821, 6

2019. ISSN: 0377-2217. doi: 10.1016/J.EJOR.2018.07.014. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0377221718306192>.

[10] BERTSEKAS, D. *Dynamic Programming and Optimal Control Volume II*. Belmont, Massachusetts, Athena Scientific, 1995.

[11] REN, Z., KROGH, B. "State aggregation in Markov decision processes", *Proceedings of the IEEE Conference on Decision and Control*, v. 4, pp. 3819–3824, 2002. ISSN: 01912216. doi: 10.1109/CDC.2002.1184960.

[12] LI, L., WALSH, T., LITTMAN, M. "Towards a unified theory of state abstraction for MDPs". In: *International Symposium on Artificial Intelligence and Mathematics*, 2006.

[13] CAO, X., REN, Z., BHATNAGAR, S., et al. "A time aggregation approach to Markov decision processes", *Automatica*, v. 38, n. 6, pp. 929–943, 6 2002. ISSN: 00051098. doi: 10.1016/S0005-1098(01)00282-5.

[14] ARRUDA, E., FRAGOSO, M. "Solving average cost Markov decision processes by means of a two-phase time aggregation algorithm", *European Journal of Operational Research*, v. 240, n. 3, pp. 697–705, 2 2015. ISSN: 03772217. doi: 10.1016/j.ejor.2014.08.023.

[15] ARRUDA, E., FRAGOSO, M. "Discounted Markov decision processes via time aggregation". In: *2016 European Control Conference, ECC 2016*, pp. 2578–2583. Institute of Electrical and Electronics Engineers Inc., 1 2016. ISBN: 9781509025916. doi: 10.1109/ECC.2016.7810678.

[16] SALDI, N., YÜKSEL, S., LINDER, T. "On the Asymptotic Optimality of Finite Approximations to Markov Decision Processes with Borel Spaces", *Mathematics of Operations Research*, v. 42, n. 4, pp. 945–978, 3 2017. ISSN: 15265471. doi: 10.1287/MOOR.2016.0832.

[17] SUTTON, R., BARTO, A. *Reinforcement Learning: An Introduction*. Cambridge, Mass., MIT Press, 1998.

[18] POWELL, W. *Approximate Dynamic Programming Solving the Curses of Dimensionality*. New Jersey, USA, John Wiley & Sons, Inc., 2011.

[19] ARCHAMBEAULT, L. *Application of Markov decision processes to mine optimisation a real option approach*. Tese de Doutorado, McGill University, Montreal, 2006.

[20] ZHAO, Q., CHEN, S., LEUNG, S., et al. "Integration of inventory and transportation decisions in a logistics system", *Transportation Research Part E: Logistics and Transportation Review*, v. 46, n. 6, pp. 913–925, 11 2010. ISSN: 1366-5545. doi: 10.1016/J.TRE.2010.03.001. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1366554510000360>.

[21] CHANG, H. "Policy set iteration for Markov decision processes", *Automatica*, v. 49, n. 12, pp. 3687–3689, 12 2013. ISSN: 0005-1098. doi: 10.1016/J.AUTOMATICA.2013.09.010. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0005109813004433>.

[22] ARRUDA, E., FRAGOSO, M., OURIQUE, F. "A multi-cluster time aggregation approach for Markov chains", *Automatica*, v. 99, pp. 382–389, 1 2019. ISSN: 0005-1098. doi: 10.1016/J.AUTOMATICA.2018.10.027. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0005109818305016>.

[23] CHOPRA, S., MEINDL, P. *Supply Chain Management Strategy, Planning, and Operation*. Sixth edit ed. Edingurgh Gate, Pearson Education Limited, 2016.

[24] SINGH, G., SIER, D., ERNST, A. T., et al. "A mixed integer programming model for long term capacity expansion planning: A case study from The Hunter Valley Coal Chain", *European Journal of Operational Research*, v. 220, n. 1, pp. 210–224, 2012. ISSN: 0377-2217. doi: https://doi.org/10.1016/j.ejor.2012.01.012. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221712000318>.

[25] BJØRNDAL, T., HERRERO, I., NEWMAN, A., et al. "Operations research in the natural resource industry", *International Transactions in Operational Research*, v. 19, n. 1-2, pp. 39–62, 2012. ISSN: 1475-3995. doi: 10.1111/j.1475-3995.2010.00800.x. Disponível em: <http://dx.doi.org/10.1111/j.1475-3995.2010.00800.x>.

[26] KOZAN, E., LIU, S. "Operations research for mining : a classification and literature review", *ASOR Bulletin*, v. 30, n. 1, pp. 2–23, 2011. Disponível em: <https://eprints.qut.edu.au/48661/>.

[27] PIMENTEL, B., MATEUS, G., ALMEIDA, F. "Mathematical models for optimizing the global mining supply chain". In: *Intelligent Systems in Operations: Methods, Models and Applications in the Supply Chain*, IGI Global,

pp. 133–163, Hershey, PA, 2010. ISBN: 9781615206063. Disponível em: <https://books.google.com.br/books?id=2f7Qjx3oofIC>.

[28] COSTA, F. *Aplicações de técnicas de otimização a problemas de planejamento operacional de lavra em minas a céu aberto*. Tese de Doutorado, PPGEM–UFOP, Ouro Preto, MG, Brasil, 2005.

[29] KLINGMAN, D., PHILLIPS, N. "Integer programming for optimal phosphate-mining strategies", *Journal of the Operational Research Society*, v. 39, n. 9, pp. 805–810, 1988. doi: 10.1057/jors.1988.140.

[30] GROENEVELD, B., TOPAL, E. "Flexible open-pit mine design under uncertainty", *Journal of Mining Science*, v. 47, n. 2, pp. 212–226, 3 2011. ISSN: 1062-7391. doi: 10.1134/S1062739147020080. Disponível em: <http://link.springer.com/10.1134/S1062739147020080>.

[31] ASKARI-NASAB, H., AWUAH-OFFEI, K., EIVAZY, H. "Large-scale open pit production scheduling using Mixed Integer Linear Programming", *International Journal of Mining and Mineral Engineering*, v. 2, n. 3, pp. 185–214, 2010. doi: 10.1504/IJMME.2010.037624.

[32] ASKARI-NASAB, H., POURRAHIMIAN, Y., BEN-AWUAH, E., et al. "Mixed integer linear programming formulations for open pit production scheduling", *Journal of Mining Science*, v. 47, n. 3, pp. 338, 2011. ISSN: 1573-8736. doi: 10.1134/S1062739147030117.

[33] SAMANTA, B., BHATTACHERJEE, A., GANGULI, R. "A genetic algorithms approach for grade control planning in a bauxite deposit". In: *Proceedings of the 32nd International Symposium on Applications of Computers and Operations Research in the Mineral Industry*, pp. 337–342, 2005.

[34] EIVAZY, H., ASKARI-NASAB, H. "A mixed integer linear programming model for short-term open pit mine production scheduling", *Mining Technology*, v. 121, n. 2, pp. 97–108, 2012. doi: 10.1179/1743286312Y.0000000006.

[35] MORENO, E., REZAKHAH, M., NEWMAN, A., et al. "Linear models for stockpiling in open-pit mine production scheduling problems", *European Journal of Operational Research*, v. 260, n. 1, pp. 212–221, 2017. ISSN: 0377-2217. doi: https://doi.org/10.1016/j.ejor.2016.12.014. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221716310335>.

[36] BAKHTAVAR, E., MAHMOUDI, H. "Development of a scenario-based robust model for the optimal truck-shovel allocation in open-pit mining", *Computers & Operations Research*, 8 2018. ISSN: 0305-0548. doi: 10.1016/J.COR.2018.08.003.

[37] FYTAS, K., HADJIGEORGIOU, J., COLLINS, J. L. "Production scheduling optimization in open pit mines", *International Journal of Surface Mining, Reclamation and Environment*, v. 7, n. 1, pp. 1–9, 1 1993. ISSN: 1389-5265. doi: 10.1080/09208119308964677. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/09208119308964677>.

[38] GOODFELLOW, R., DIMITRAKOPOULOS, R. "Simultaneous Stochastic Optimization of Mining Complexes and Mineral Value Chains", *Mathematical Geosciences*, v. 49, n. 3, pp. 341–360, 2017. ISSN: 1874-8953. doi: 10.1007/s11004-017-9680-3. Disponível em: <https://doi.org/10.1007/s11004-017-9680-3>.

[39] MONTIEL, L., DIMITRAKOPOULOS, R. "Optimizing mining complexes with multiple processing and transportation alternatives: An uncertainty-based approach", *European Journal of Operational Research*, v. 247, n. 1, pp. 166–178, 2015. ISSN: 0377-2217. doi: https://doi.org/10.1016/j.ejor.2015.05.002. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221715003720>.

[40] MONTIEL, L., DIMITRAKOPOULOS, R. "A heuristic approach for the stochastic optimization of mine production schedules", *Journal of Heuristics*, v. 23, n. 5, pp. 397–415, 2017. ISSN: 1572-9397. doi: 10.1007/s10732-017-9349-6.

[41] POURRAHIMIAN, Y., ASKARI-NASAB, H., TANNANT, D. D. "A multi-step approach for block-cave production scheduling optimization", *International Journal of Mining Science and Technology*, v. 23, n. 5, pp. 739–750, 2013. ISSN: 2095-2686. doi: https://doi.org/10.1016/j.ijmst.2013.08.019.

[42] RAMAZAN, S., DIMITRAKOPOULOS, R. "Traditional and New MIP Models for Production Scheduling With In-Situ Grade Variability", *International Journal of Surface Mining, Reclamation and Environment*, v. 18, n. 2, pp. 85–98, 2004. doi: 10.1080/13895260412331295367.

[43] GILANI, S., SATTARVAND, J. "Integrating geological uncertainty in long-term open pit mine production planning by ant colony optimization", *Computers & Geosciences*, v. 87, pp. 31–40, 2 2016. ISSN: 0098-3004. doi: 10.1016/J.CAGEO.2015.11.008.

[44] LAMGHARI, A., DIMITRAKOPOULOS, R. "Network-flow based algorithms for scheduling production in multi-processor open-pit mines accounting for metal uncertainty", *European Journal of Operational Research*, v. 250, n. 1, pp. 273–290, 4 2016. ISSN: 0377-2217. doi: 10.1016/J.EJOR.2015.08.051.

[45] NEHRING, M., TOPAL, E., KIZIL, M., et al. "An investigation to integrate optimum long-term planning with short planning in underground mine production scheduling". In: *Mine Planning and Equipment Selection (MPES) Conference / Fremantle, WA*, 2010.

[46] NEHRING, M., TOPAL, E., KIZIL, M., et al. "Integrated short- and medium-term underground mine production scheduling", *Journal of the Southern African Institute of Mining and Metallurgy*, v. 112, pp. 365–378, 2012. ISSN: 2225-6253.

[47] MONTIEL, L., DIMITRAKOPOULOS, R., KAWAHATA, K. "Globally optimising open-pit and underground mining operations under geological uncertainty", *Mining Technology*, v. 125, n. 1, pp. 2–14, 1 2016. ISSN: 1474-9009. doi: 10.1179/1743286315Y.0000000027.

[48] SARIN, S., WEST-HANSEN, J. "The long-term mine production scheduling problem", *IIE Transactions*, v. 37, n. 2, pp. 109–121, 2 2005. ISSN: 0740-817X. doi: 10.1080/07408170490447339.

[49] NEHRING, M., TOPAL, E., LITTLE, J. "A new mathematical programming model for production schedule optimization in underground mining operations", *Journal of the Southern African Institute of Mining and Metallurgy*, v. 110, pp. 437–446, 2010. ISSN: 2225-6253.

[50] TA, C., INGOLFSSON, A., DOUCETTE, J. "A linear model for surface mining haul truck allocation incorporating shovel idle probabilities", *European Journal of Operational Research*, v. 231, n. 3, pp. 770–778, 12 2013. ISSN: 0377-2217. doi: 10.1016/J.EJOR.2013.06.016.

[51] BENNDORF, J. "Application of efficient methods of conditional simulation for optimising coal blending strategies in large continuous open pit mining operations", *International Journal of Coal Geology*, v. 112, pp. 141–153, 2013. ISSN: 0166-5162. doi: https://doi.org/10.1016/j.coal.2012.10.008. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0166516212002492>.

[52] MATAMOROS, M., DIMITRAKOPOULOS, R. "Stochastic short-term mine production schedule accounting for fleet allocation, operational considerations and blending restrictions", *European Journal of Operational Research*, v. 255, n. 3, pp. 911–921, 2016. ISSN: 0377-2217. doi: https://doi.org/10.1016/j.ejor.2016.05.050. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221716303873>.

[53] PARICHEH, M., OSANLOO, M. "A simulation-based risk management approach to locating facilities in open-pit mines under price and grade uncertainties", *Simulation Modelling Practice and Theory*, v. 89, pp. 119–134, 12 2018. ISSN: 1569-190X. doi: 10.1016/J.SIMPAT.2018.09.015.

[54] BODON, P., FRICKE, C., SANDEMAN, T., et al. "Modeling the mining supply chain from mine to port: A combined optimization and simulation approach", *Journal of Mining Science*, v. 47, n. 2, pp. 202–211, 2011.

[55] GOODFELLOW, R., DIMITRAKOPOULOS, R. "Global optimization of open pit mining complexes with uncertainty", *Applied Soft Computing*, v. 40, pp. 292–304, 3 2016. ISSN: 1568-4946. doi: 10.1016/J.ASOC.2015.11.038. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1568494615007565>.

[56] KUMAR, A., CHATTERJEE, S. "Open-pit coal mine production sequencing incorporating grade blending and stockpiling options: An application from an Indian mine", *Engineering Optimization*, v. 49, n. 5, pp. 762–776, 2017. doi: 10.1080/0305215X.2016.1210312. Disponível em: <https://doi.org/10.1080/0305215X.2016.1210312>.

[57] GARCIA-FLORES, R., SINGH, G., ERNST, A., et al. "Medium-term rail planning at Rio Tinto Iron Ore". In: *Proceedings of the 19th International Congress on Modelling and Simulation (MODSIM 2011)*, pp. 311–317. Modelling and Simulation Society of Australia and New Zealand (MSSANZ), 2011.

[58] SINGH, G., GARCÍA-FLORES, R., ERNST, A., et al. "Medium-Term Rail Scheduling for an Iron Ore Mining Company", *Interfaces*, v. 44, n. 2, pp. 222–240, 2014. doi: 10.1287/inte.1120.0669. Disponível em: <https://doi.org/10.1287/inte.1120.0669>.

[59] NOBREGA, M. A. *Modelagem matemática de um sistema de produção e transporte de minério de ferro*. Tese de Doutorado, UNIVERSIDADE ESTADUAL DE CAMPINAS, Campinas, SP, Brasil, 1997.

[60] BOLAND, N. L., SAVELSBERGH, M. W. P. "Optimizing the Hunter Valley Coal Chain". In: Gurnani, H., Mehrotra, A., Ray, S. (Eds.), *Supply Chain Disruptions: Theory and Practice of Managing Risk*, Springer London, pp. 275–302, London, 2012. ISBN: 978-0-85729-778-5. doi: 10.1007/978-0-85729-778-5{_}10. Disponível em: <https://doi.org/10.1007/978-0-85729-778-5_10>.

[61] THOMAS, A., SINGH, G., KRISHNAMOORTHY, M., et al. "Distributed optimisation method for multi-resource constrained scheduling in coal supply chains", *International Journal of Production Research*, v. 51, n. 9, pp. 2740–2759, 2013. doi: 10.1080/00207543.2012.737955. Disponível em: <http://dx.doi.org/10.1080/00207543.2012.737955>.

[62] ABDEKHODAEE, A., DUNSTALL, S., ERNST, A., et al. "Integration of stockyard and rail network: a scheduling case study". In: *Proceedings of the Fifth Asia Pacific Industrial Engineering and Management Systems Conference*, 2004.

[63] THOMAS, A., VENKATESWARAN, J., SINGH, G., et al. "A resource constrained scheduling problem with multiple independent producers and a single linking constraint: A coal supply chain example", *European Journal of Operational Research*, v. 236, n. 3, pp. 946–956, 2014. ISSN: 0377-2217. doi: https://doi.org/10.1016/j.ejor.2013.10.006. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221713008084>.

[64] THOMAS, A., KRISHNAMOORTHY, M., VENKATESWARAN, J., et al. "Decentralised decision-making in a multi-party supply chain", *International Journal of Production Research*, v. 54, n. 2, pp. 405–425, 2016. doi: 10.1080/00207543.2015.1096977. Disponível em: <https://doi.org/10.1080/00207543.2015.1096977>.

[65] BALZARY, J., MOHAIS, A. "Consideration for Multi-objective Metaheuristic Optimisation of Large Iron Ore and Coal Supply Chains, from Resource to Market". In: Dimitrakopoulos, R. (Ed.), *Advances in Applied Strategic Mine Planning*, Springer International Publishing, pp. 297–316, Cham, 2018. ISBN: 978-3-319-69320-0. doi: 10.1007/978-3-319-69320-0{_}20.

[66] BOLAND, N., GULCZYNSKI, D., SAVELSBERGH, M. "A stockyard planning problem", *EURO Journal on Transportation and Logistics*, v. 1, n. 3, pp. 197–236, 2012. ISSN: 2192-4384. doi: 10.1007/s13676-012-0011-z. Disponível em: <https://doi.org/10.1007/s13676-012-0011-z>.

[67] BOLAND, N., GULEZYNSKI, D., JACKSON, M. P., et al. "Improved stockyard management strategies for coal export terminal at Newcastle". In: *Proceedings of the 19th International Congress on Modelling and Simulation (Perth, W.A. 12-16 December, 2011)*, pp. 718–724, 2011.

[68] HANOUN, S., KHAN, B., JOHNSTONE, M., et al. "An effective heuristic for stockyard planning and machinery scheduling at a coal handling facility". In: *Industrial Informatics (INDIN), 2013 11th IEEE International Conference on*, pp. 206–211. IEEE, 2013. doi: 10.1109/INDIN.2013.6622883.

[69] SHIH, L. H. "Planning of fuel coal imports using a mixed integer programming method", *International Journal of Production Economics*, v. 51, n. 3, pp. 243–249, 1997. ISSN: 0925-5273. doi: https://doi.org/10.1016/S0925-5273(97)00078-9. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925527397000789>.

[70] LIU, C. M. "A Blending and Inter-Modal Transportation Model for the Coal Distribution Problem", *International Journal of Operations Research*, v. 5, n. 2, pp. 107–116, 2008.

[71] ARIGONI, A., NEWMAN, A., TURNER, C., et al. "Optimizing global thermal coal shipments", *Omega*, v. 72, n. Supplement C, pp. 118–127, 2017. ISSN: 0305-0483. doi: https://doi.org/10.1016/j.omega.2016.12.001. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0305048316300792>.

[72] DONG, J., GAO, F., DAI, S., et al. "Purchasing and transport scheduling based on scenario tree in coal maritime supply chain with stochastic demand". In: *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pp. 3444–3449, 2014. doi: 10.1109/WCICA.2014.7053288.

[73] KUMRAL, M. "Optimal location of a mine facility by genetic algorithms", *Mining Technology*, v. 113, n. 2, pp. 83–88, 2004. doi: 10.1179/037178404225004940. Disponível em: <https://doi.org/10.1179/037178404225004940>.

[74] ZHOU, R. J., LI, L. J. "Joint capacity planning and distribution network optimization of coal supply chains under uncertainty", *AIChE Journal*, v. 64, n. 4, pp. 1246–1261, 2018. doi: 10.1002/aic.16012. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/aic.16012>.

[75] HENDERSON, J. M. "A Short-Run Model for the Coal Industry", *The Review of Economics and Statistics*, v. 37, n. 4, pp. 336–346, 1955. ISSN:

00346535, 15309142. Disponível em: <http://www.jstor.org/stable/1925847>.

[76] HENDERSON, J. M. *The Efficiency of the Coal Industry-An Application of Linear Programming*. Tese de Doutorado, Harvard University Press, Cambridge, Mass., 1958.

[77] TZENG, G. H., TEODOROVIĆ, D., HWANG, M. J. "Fuzzy bicriteria multiindex transportation problems for coal allocation planning of Taipower", *European Journal of Operational Research*, v. 95, n. 1, pp. 62–72, 1996. ISSN: 0377-2217. doi: https://doi.org/10.1016/0377-2217(95)00247-2. Disponível em: <http://www.sciencedirect.com/science/article/pii/0377221795002472>.

[78] PENDHARKAR, P. C. "A fuzzy linear programming model for production planning in coal mines", *Computers & Operations Research*, v. 24, n. 12, pp. 1141–1149, 1997. ISSN: 0305-0548. doi: https://doi.org/10.1016/S0305-0548(97)00024-5.

[79] MÁRQUEZ, L. G. "Optimización de una red de transporte combinado para la exportación del carbón del interior de Colombia", *Revista EIA*, pp. 103–113, 2011. ISSN: 1794-1237. Disponível em: <http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1794-12372011000200008&nrm=iso>.

[80] CANALES-BUSTOS, L., SANTIBAÑEZ-GONZÁLEZ, E., CANDIA-VÉJAR, A. "A multi-objective optimization model for the design of an effective decarbonized supply chain in mining", *International Journal of Production Economics*, v. 193, pp. 449–464, 2017. ISSN: 0925-5273. doi: https://doi.org/10.1016/j.ijpe.2017.08.012. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925527317302542>.

[81] CHENG, Q., NING, S., XIA, X., et al. "Modelling of coal trade process for the logistics enterprise and its optimisation with stochastic predictive control", *International Journal of Production Research*, v. 54, n. 8, pp. 2241–2259, 2016. doi: 10.1080/00207543.2015.1062568. Disponível em: <https://doi.org/10.1080/00207543.2015.1062568>.

[82] LIU, F., LV, T., SAJID, M., et al. "Optimization for China's coal flow based on matching supply and demand sides", *Resources, Conservation and Recycling*, v. 129, pp. 345–354, 2018. ISSN: 0921-3449. doi: https://doi.org/10.1016/j.resconrec.2016.08.013. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0921344916302051>.

[83] RAVINDRAN, A., HANLINE, D. L. "Optimal Location of Coal Blending Plants by Mixed-Integer Programming", *AIIE Transactions*, v. 12, n. 2, pp. 179–185, 1980. doi: 10.1080/05695558008974505. Disponível em: <https://doi.org/10.1080/05695558008974505>.

[84] BARBARO, R., RAMANI, R. "Generalized multiperiod MIP model for production scheduling and processing facilities selection and location", *Mining Engineering*, v. 38, n. 2, pp. 107–114, 1986.

[85] SHERALI, H. D., PURI, R. "Models for a coal blending and distribution problem", *Omega*, v. 21, n. 2, pp. 235–243, 1993. ISSN: 0305-0483. doi: https://doi.org/10.1016/0305-0483(93)90056-Q. Disponível em: <http://www.sciencedirect.com/science/article/pii/030504839390056Q>.

[86] CHANDA, E. K. C., DAGDELEN, K. "Optimal blending of mine production using goal programming and interactive graphics systems", *International Journal of Surface Mining, Reclamation and Environment*, v. 9, n. 4, pp. 203–208, 1995. doi: 10.1080/09208119508964748. Disponível em: <https://doi.org/10.1080/09208119508964748>.

[87] LYU, J., GUNASEKARAN, A., CHEN, C. Y., et al. "A goal programming model for the coal blending problem", *Computers & Industrial Engineering*, v. 28, n. 4, pp. 861–868, 1995. ISSN: 0360-8352. doi: https://doi.org/10.1016/0360-8352(95)00007-N. Disponível em: <http://www.sciencedirect.com/science/article/pii/036083529500007N>.

[88] LAI, J. W., CHEN, C. Y. "A cost minimization model for coal import strategy", *Energy Policy*, v. 24, n. 12, pp. 1111–1117, 1996. ISSN: 0301-4215. doi: https://doi.org/10.1016/S0301-4215(96)00091-2. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0301421596000912>.

[89] LIU, C. M., SHERALI, H. D. "A coal shipping and blending problem for an electric utility company", *Omega*, v. 28, n. 4, pp. 433–444, 2000. ISSN: 0305-0483. doi: https://doi.org/10.1016/S0305-0483(99)00067-5. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0305048399000675>.

[90] PENG, H. J., ZHOU, M. H., LIU, M. Z., et al. "A dynamic optimization model of an integrated coal supply chain system and its application", *Mining Science and Technology (China)*, v. 19, n. 6, pp. 842–846, 2009. ISSN:

1674-5264. doi: https://doi.org/10.1016/S1674-5264(09)60153-8. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1674526409601538>.

[91] SCHELLENBERG, S., LI, X., MICHALEWICZ, Z. "Benchmarks for the Coal Processing and Blending Problem". In: *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, pp. 1005–1012, New York, NY, USA, 2016. ACM. ISBN: 978-1-4503-4206-3. doi: 10.1145/2908812.2908945. Disponível em: <http://doi.acm.org/10.1145/2908812.2908945>.

[92] PENDHARKAR, P. C., RODGER, J. A. "Nonlinear programming and genetic search application for production scheduling in coal mines", *Annals of Operations Research*, v. 95, n. 1, pp. 251–267, 1 2000. ISSN: 1572-9338. doi: 10.1023/A:1018958209290. Disponível em: <https://doi.org/10.1023/A:1018958209290>.

[93] LIAO, Y. F., WU, C. H., MA, X. Q. "New hybrid optimization model for power coal blending". In: *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, v. 7, pp. 4023–4027. IEEE, 2005.

[94] XI-JIN, G., MING, C., JIA-WEI, W. "Coal blending optimization of coal preparation production process based on improved GA", *Procedia Earth and Planetary Science*, v. 1, n. 1, pp. 654–660, 2009. ISSN: 1878-5220. doi: https://doi.org/10.1016/j.proeps.2009.09.103. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1878522009001040>.

[95] CANDLER, W. "Coal blending – with acceptance sampling", *Computers & Operations Research*, v. 18, n. 7, pp. 591–596, 1991. ISSN: 0305-0548. doi: https://doi.org/10.1016/0305-0548(91)90066-Z. Disponível em: <http://www.sciencedirect.com/science/article/pii/030505489190066Z>.

[96] SHIH, J. S., FREY, H. C. "Coal blending optimization under uncertainty", *European Journal of Operational Research*, v. 83, n. 3, pp. 452–465, 1995. ISSN: 0377-2217. doi: https://doi.org/10.1016/0377-2217(94)00243-6. Disponível em: <http://www.sciencedirect.com/science/article/pii/0377221794002436>.

[97] KUMRAL, M. "Application of chance-constrained programming based on multi-objective simulated annealing to solve a mineral blending problem", *Engineering Optimization*, v. 35, n. 6, pp. 661–673, 2003. doi:

10.1080/03052150310001614837. Disponível em: <https://doi.org/10.1080/03052150310001614837>.

[98] CONRADIE, D. G., MORISON, L. E., JOUBERT, J. W. "Scheduling at coal handling facilities using Simulated Annealing", *Mathematical Methods of Operations Research*, v. 68, n. 2, pp. 277–293, 2008. ISSN: 1432-5217. doi: 10.1007/s00186-008-0221-1. Disponível em: <https://doi.org/10.1007/s00186-008-0221-1>.

[99] SAKALL\I, U. S., BAYKOÇ, O. F. "An optimization approach for brass casting blending problem under aletory and epistemic uncertainties", *International Journal of Production Economics*, v. 133, n. 2, pp. 708–718, 2011. ISSN: 0925-5273. doi: https://doi.org/10.1016/j.ijpe.2011.05.022. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925527311002398>.

[100] SAKALL\I, U. S., BAYKOÇ, O. F. "Strong guidance on mitigating the effects of uncertainties in the brass casting blending problem: a hybrid optimization approach", *Journal of the Operational Research Society*, v. 64, n. 4, pp. 562–576, 2013. ISSN: 1476-9360. doi: 10.1057/jors.2012.50. Disponível em: <https://doi.org/10.1057/jors.2012.50>.

[101] SAKALL\I, U. S., BAYKOÇ, O. F., BIRG\"OREN, B. "Stochastic optimization for blending problem in brass casting industry", *Annals of Operations Research*, v. 186, n. 1, pp. 141–157, 6 2011. ISSN: 1572-9338. doi: 10.1007/s10479-011-0851-1. Disponível em: <https://doi.org/10.1007/s10479-011-0851-1>.

[102] ARIGONI, A. *Optimization techniques in coal markets : a global cost minimization and a multi-stage procurement strategy.* Tese de Doutorado, Colorado School of Mines. Arthur Lakes Library, Golden, Colorado, USA, 2016.

[103] ZHANG, J., DIMITRAKOPOULOS, R. "Optimising a Mineral Supply Chain Under Uncertainty with Long-Term Sales Contracts". In: Dimitrakopoulos, R. (Ed.), *Advances in Applied Strategic Mine Planning*, Springer International Publishing, pp. 787–800, Cham, 2018. ISBN: 978-3-319-69320-0. doi: 10.1007/978-3-319-69320-0{_}45.

[104] BODON, P., FRICKE, C., SANDEMAN, T., et al. "Combining Optimisation and Simulation to Model a Supply Chain from Pit to Port". In: Dimitrakopoulos, R. (Ed.), *Advances in Applied Strategic Mine Planning*, Springer

International Publishing, pp. 251–267, Cham, 2018. ISBN: 978-3-319-69320-0. doi: 10.1007/978-3-319-69320-0{_}17.

[105] CHAKRABORTY, M., CHANDRA, M. K. "Multicriteria decision making for optimal blending for beneficiation of coal: a fuzzy programming approach", *Omega*, v. 33, n. 5, pp. 413–418, 2005. ISSN: 0305-0483. doi: https://doi.org/10.1016/j.omega.2004.07.005. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0305048304001008>.

[106] LI, W., HUANG, G. H., DONG, C., et al. "An Inexact Fuzzy Programming Approach for Power Coal Blending", *Journal of Environmental Informatics*, v. 21, n. 2, 2013. ISSN: 1684-8799.

[107] DAI, C., CAI, X. H., CAI, Y. P., et al. "A simulation-based fuzzy possibilistic programming model for coal blending management with consideration of human health risk under uncertainty", *Applied Energy*, v. 133, pp. 1–13, 2014. ISSN: 0306-2619. doi: https://doi.org/10.1016/j.apenergy.2014.07.092. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0306261914007831>.

[108] LIU, Y., HUANG, G. H., CAI, Y. P., et al. "Development of an inexact optimization model for coupled coal and power management in North China", *Energy Policy*, v. 37, n. 11, pp. 4345–4363, 2009. ISSN: 0301-4215. doi: https://doi.org/10.1016/j.enpol.2009.05.050. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0301421509003942>.

[109] BENALCAZAR, P., KAMIŃSKI, J., SAŁUGA, P. "The storage location problem in a coal supply chain: background and methodological approach", *Gospodarka Surowcami Mineralnymi*, v. 33, n. 1, pp. 5–14, 2017. ISSN: 2299-2324. doi: 10.1515/gospo-2017-0009. Disponível em: <https://doi.org/10.1515/gospo-2017-0009>.

[110] PROMBAN, M., KITTITHREERAPRONCHAI, O. "Robust Optimization for Coal Transportation Planning", *Journal of Engineering and Applied Science*, v. 12, n. 1, pp. 5609–5616, 2017. ISSN: 1816-949X.

[111] PIMENTEL, B., MATEUS, G., ALMEIDA, F. "Stochastic capacity planning and dynamic network design", *International Journal of Production Economics*, v. 145, n. 1, pp. 139–149, 2013. ISSN: 0925-5273. doi: https://doi.org/10.1016/j.ijpe.2013.01.019. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925527313000376>.

[112] DIMITRAKOPOULOS, R. "Conditional simulation algorithms for modelling orebody uncertainty in open pit optimisation", *International Journal of Surface Mining, Reclamation and Environment*, v. 12, n. 4, pp. 173–179, 1998. doi: 10.1080/09208118908944041. Disponível em: <https://doi.org/10.1080/09208118908944041>.

[113] DIMITRAKOPOULOS, R., FARRELLY, C., GODOY, M. "Moving forward from traditional optimization: grade uncertainty and risk effects in open-pit design", *Mining Technology*, v. 111, n. 1, pp. 82–88, 2002. doi: 10.1179/mnt.2002.111.1.82. Disponível em: <https://doi.org/10.1179/mnt.2002.111.1.82>.

[114] FARMER, I., DIMITRAKOPOULOS, R. "Schedule-based pushback design within the stochastic optimisation framework", *International Journal of Mining, Reclamation and Environment*, v. 32, n. 5, pp. 327–340, 7 2018. ISSN: 1748-0930. doi: 10.1080/17480930.2017.1289606.

[115] GRIECO, N., DIMITRAKOPOULOS, R. "Managing grade risk in stope design optimisation: probabilistic mathematical programming model and application in sublevel stoping", *Mining Technology*, v. 116, n. 2, pp. 49–57, 6 2007. ISSN: 1474-9009. doi: 10.1179/174328607X191038.

[116] KOUSHAVAND, B., ASKARI-NASAB, H., DEUTSCH, C. V. "A linear programming model for long-term mine planning in the presence of grade uncertainty and a stockpile", *International Journal of Mining Science and Technology*, v. 24, n. 4, pp. 451–459, 2014. ISSN: 2095-2686. doi: https://doi.org/10.1016/j.ijmst.2014.05.006.

[117] LAMGHARI, A., DIMITRAKOPOULOS, R. "A diversified Tabu search approach for the open-pit mine production scheduling problem with metal uncertainty", *European Journal of Operational Research*, v. 222, n. 3, pp. 642–652, 2012. ISSN: 0377-2217. doi: https://doi.org/10.1016/j.ejor.2012.05.029.

[118] HALATCHEV, R. A. "A model of discounted profit variation of open-pit production sequencing optimization". In: *Application of Computers and Operations Research in the Mineral Industry-Proc. of the 32nd Int. Symposium on the Application of Computers and Operations Research in the Mineral Industry, APCOM 2005*, pp. 315–323. AA Balkema Publishers, 2005.

[119] TOPAL, E., RAMAZAN, S. "Mining truck scheduling with stochastic maintenance cost", *Journal of Coal Science and Engineering (China)*, v. 18, n. 3, pp. 313–319, 2012. ISSN: 1866-6566. doi: 10.1007/s12404-012-0316-4.

[120] LAMGHARI, A., DIMITRAKOPOULOS, R., FERLAND, J. "A variable neighbourhood descent algorithm for the open-pit mine production scheduling problem with metal uncertainty", *Journal of the Operational Research Society*, v. 65, n. 9, pp. 1305–1314, 2014. ISSN: 1476-9360. doi: 10.1057/jors.2013.81.

[121] BOLAND, N., DUMITRESCU, I., FROYLAND, G. "A multistage stochastic programming approach to open pit mine production scheduling with uncertain geology. Optimization Online". 2008.

[122] MAI, N., TOPAL, E., ERTEN, O., et al. "A new risk-based optimisation method for the iron ore production scheduling using stochastic integer programming", *Resources Policy*, 11 2018. ISSN: 0301-4207. doi: 10.1016/J.RESOURPOL.2018.11.004. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0301420717302763>.

[123] LAMGHARI, A., DIMITRAKOPOULOS, R. "Progressive hedging applied as a metaheuristic to schedule production in open-pit mines accounting for reserve uncertainty", *European Journal of Operational Research*, v. 253, n. 3, pp. 843–855, 9 2016. ISSN: 0377-2217. doi: 10.1016/J.EJOR.2016.03.007. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221716301357>.

[124] TA, C. H., KRESTA, J. V., FORBES, J. F., et al. "A stochastic optimization approach to mine truck allocation", *International Journal of Surface Mining, Reclamation and Environment*, v. 19, n. 3, pp. 162–175, 9 2005. ISSN: 1389-5265. doi: 10.1080/13895260500128914. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/13895260500128914>.

[125] ERCELEBI, S., BASCETIN, A. "Optimization of shovel-truck system for surface mining", *Journal of the Southern African Institute of Mining and Metallurgy*, v. 109, pp. 433–439, 2009. ISSN: 2225-6253.

[126] REINHARDT, G., DADA, M., CHOPRA, S. "Coal Movement by Railroad in the Powder River Basin", *Energy Studies Review*, v. 11, n. 1, 2002.

[127] BINKOWSKI, M., MCCARRAGHER, B. J. "A Queueing Model for the Design and Analysis of a Mining Stockyard", *Discrete Event Dynamic Systems*, v. 9, n. 1, pp. 75–98, 1 1999. ISSN: 1573-7594. doi: 10.1023/A:1008397332376. Disponível em: <https://doi.org/10.1023/A:1008397332376>.

[128] RAJ, M., VARDHAN, H., RAO, Y. "Production optimisation using simulation models in mines: a critical review", *International Journal of Operational Research*, v. 6, n. 3, pp. 330–359, 2009. doi: 10.1504/IJOR.2009.026937.

[129] DOWD, P. A. "Risk assessment in reserve estimation and open-pit planning", *Transactions of the Institution of Mining and Metallurgy(Section A: Mining Industry)*, v. 103, 1994.

[130] ASAD, M., DIMITRAKOPOULOS, R. "Implementing a parametric maximum flow algorithm for optimal open pit mine design under uncertain supply and demand", *Journal of the Operational Research Society*, v. 64, n. 2, pp. 185–197, 2013. ISSN: 1476-9360. doi: 10.1057/jors.2012.26. Disponível em: <https://doi.org/10.1057/jors.2012.26>.

[131] DIMITRAKOPOULOS, R., MARTINEZ, L., RAMAZAN, S. "A maximum upside/minimum downside approach to the traditional optimization of open pit mine design", *Journal of Mining Science*, v. 43, n. 1, pp. 73–82, 1 2007. ISSN: 1573-8736. doi: 10.1007/s10913-007-0009-3. Disponível em: <https://doi.org/10.1007/s10913-007-0009-3>.

[132] CHATTERJEE, S., SETHI, M. R., ASAD, M. W. A. "Production phase and ultimate pit limit design under commodity price uncertainty", *European Journal of Operational Research*, v. 248, n. 2, pp. 658–667, 2016. ISSN: 0377-2217. doi: https://doi.org/10.1016/j.ejor.2015.07.012. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0377221715006451>.

[133] INTHAVONGSA, I., DREBENSTEDT, C., BONGAERTS, J., et al. "Real options decision framework: Strategic operating policies for open pit mine planning", *Resources Policy*, v. 47, pp. 142–153, 3 2016. ISSN: 0301-4207. doi: 10.1016/J.RESOURPOL.2016.01.009. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0301420716300022>.

[134] DINDARLOO, S., OSANLOO, M., FRIMPONG, S. "A stochastic simulation framework for truck and shovel selection and sizing in open pit mi-

nes", *Journal of the Southern African Institute of Mining and Metallurgy*, v. 115, pp. 209–219, 2015. ISSN: 2225-6253.

[135] DE FARIA, C., DA COSTA CRUZ, M. "Simulation Modelling of Vitória-Minas Closed-Loop Rail Network", *Transport Problems*, v. 10, n. SE, pp. 125–139, 2015. doi: 10.21307/tp-2015-067.

[136] MEIRELES, R. P. L. *Modelagem e Simulação da Malha Ferroviária em Circuito Fechado da Estrada de Ferro de Vitória a Minas*. Tese de Doutorado, Universidade Federal do Esp{\'\i}rito Santo, 2010.

[137] LE, V. T., JOHNSTONE, M., ZHANG, J., et al. "Complex Simulation of Stockyard Mining Operations". In: Gao, D., Ruan, N., Xing, W. (Eds.), *Advances in Global Optimization*, pp. 529–537, Cham, 2015. Springer International Publishing. ISBN: 978-3-319-08377-3.

[138] FIORONI, M. M., FRANZESE, L. A. G., ZANIN, C. E., et al. "Simulation of continuous behavior using discrete tools: Ore conveyor transport". In: *2007 Winter Simulation Conference*, pp. 1655–1662, 2007. doi: 10.1109/WSC.2007.4419786.

[139] XIAO-PING, B., YU-HONG, Z. A., YA-NAN, L. A. "A Novel Approach to Study Real-Time Dynamic Optimization Analysis and Simulation of Complex Mine Logistics Transportation Hybrid System with Belt and Surge Links", *Discrete Dynamics in Nature and Society*, v. 2015, pp. 1–8, 2015. doi: doi:10.1155/2015/601578.

[140] VAN VIANEN, T., OTTJES, J., LODEWIJKS, G. "Belt conveyor network design using simulation", *Journal of Simulation*, v. 10, n. 3, pp. 157–165, 2016. doi: 10.1057/jos.2014.38. Disponível em: <`https://doi.org/10.1057/jos.2014.38`>.

[141] QUE, S., AWUAH-OFFEI, K., FRIMPONG, S. "Optimising design parameters of continuous mining transport systems using discrete event simulation", *International Journal of Mining, Reclamation and Environment*, v. 30, n. 3, pp. 217–230, 2016. doi: 10.1080/17480930.2015.1037056. Disponível em: <`https://doi.org/10.1080/17480930.2015.1037056`>.

[142] MCNEARNY, R., NIE, Z. "Simulation of a conveyor belt network at an underground coal mine", *Mineral Resources Engineering*, v. 09, n. 03, pp. 343–355, 2000. doi: 10.1142/S0950609800000299.

[143] REUS, L., PAGNONCELLI, B., ARMSTRONG, M. "Better management of production incidents in mining using multistage stochastic optimization", *Resources Policy*, v. 63, pp. 101404, 10 2019. ISSN: 0301-4207. doi: 10.1016/J.RESOURPOL.2019.101404.

[144] PEIDRO, D., MULA, J., POLER, R., et al. "Quantitative models for supply chain planning under uncertainty", *International Journal of Advanced Manufacturing Technology*, v. 43, n. 3-4, pp. 400–420, 7 2009. ISSN: 02683768. doi: 10.1007/s00170-008-1715-y.

[145] Drezner, Z., Hamacher, H. (Eds.). *Facility Location: Applications and Theory*. New York, Springer, 2004.

[146] SHEN, Z. "Integrated supply chain design models: A survey and future research directions", *Journal of Industrial and Management Optimization*, v. 3, n. 1, pp. 1–27, 2007. ISSN: 1553166X. doi: 10.3934/jimo.2007.3.1.

[147] MELO, M. T., NICKEL, S., SALDANHA-DA GAMA, F. "Facility location and supply chain management - A review", *European Journal of Operational Research*, v. 196, n. 2, pp. 401–412, 2009. ISSN: 0377-2217. doi: https://doi.org/10.1016/j.ejor.2008.05.007.

[148] GOVINDAN, K., FATTAHI, M., KEYVANSHOKOOH, E. "Supply chain network design under uncertainty: A comprehensive review and future research directions", *European Journal of Operational Research*, v. 263, n. 1, pp. 108–141, 11 2017. ISSN: 0377-2217. doi: 10.1016/J.EJOR.2017.04.009. Disponível em: <`https://www.sciencedirect.com/science/article/pii/S0377221717303429`>.

[149] GOVINDAN, K., CHENG, T. "Advances in stochastic programming and robust optimization for supply chain planning", *Computers and Operations Research*, v. 100, pp. 262–269, 12 2018. ISSN: 03050548. doi: 10.1016/j.cor.2018.07.027.

[150] MULA, J., POLER, R., GARCÍA-SABATER, G., et al. "Models for production planning under uncertainty: A review", *International Journal of Production Economics*, v. 103, n. 1, pp. 271–285, 9 2006. ISSN: 09255273. doi: 10.1016/j.ijpe.2005.09.001.

[151] MULA, J., PEIDRO, D., DÍAZ-MADROÑERO, M., et al. "Mathematical programming models for supply chain production and transport planning", *European Journal of Operational Research*, v. 204, n. 3, pp. 377–390, 8 2010. ISSN: 0377-2217. doi: 10.1016/J.EJOR.2009.09.008. Dispo-

nível em: <https://www.sciencedirect.com/science/article/abs/pii/S0377221709005694>.

[152] BELLMAN, R., GLICKSBERG, I., GROSS, O. "On the Optimal Inventory Equation", *Management Science*, v. 2, n. 1, pp. 83–104, 10 1955. ISSN: 0025-1909. doi: 10.1287/mnsc.2.1.83.

[153] PORTEUS, E. "Stochastic inventory theory". In: *Handbooks in Operations Research and Management Science*, v. 2, Elsevier Science Publishers B.V., cap. 12, pp. 605–652, North-Holland, 1 1990. doi: 10.1016/S0927-0507(05)80176-8.

[154] DANTZIG, G. B. "Linear programming under uncertainty", *Management Science*, v. 1, pp. 197–206, 1955.

[155] BEALE, E. M. L. "On Minimizing A Convex Function Subject to Linear Inequalities", *Journal of the Royal Statistical Society. Series B (Methodological)*, v. 17, n. 2, pp. 173–184, 1955. ISSN: 00359246. Disponível em: <http://www.jstor.org/stable/2983952>.

[156] SHAPIRO, A., DENTCHEVA, D., RUSZCZYNSKI, A. *Lectures on stochastic programming: modeling and theory*. Philadelphia, USA, MPS-SIAM, 2009.

[157] BIRGE, J. R., LOUVEAUX, F. *Introduction to stochastic programming*. Copenhagen, Springer, 2011. ISBN: 9781461402374.

[158] BELLMAN, R., DREYFUS, S. *Applied Dynamic Programming*. Princeton, N.J., Princeton University Press, 1962.

[159] BERTSEKAS, D. *Dynamic Programming and Optimal Control Volume I*. Belmont, Massachusetts, Athena Scientific, 1995.

[160] HOWARD, R. *Dynamic programming and Markov processes*. New York, The MIT Press, 1960.

[161] ARRUDA, E. F., DO VAL, J. B. R. "Stability and optimality of a multiproduct production and storage system under demand uncertainty", *European Journal of Operational Research*, v. 188, n. 2, pp. 406–427, 7 2008. ISSN: 0377-2217. doi: 10.1016/J.EJOR.2007.04.028. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0377221707004286>.

[162] ARRUDA, E., FRAGOSO, M., DO VAL, J. "Approximate dynamic programming via direct search in the space of value function approximations",

European Journal of Operational Research, v. 211, n. 2, pp. 343–351, 6 2011. ISSN: 03772217. doi: 10.1016/j.ejor.2010.11.019.

[163] SANTOSO, T., AHMED, S., GOETSCHALCKX, M., et al. "A stochastic programming approach for supply chain network design under uncertainty", *European Journal of Operational Research*, v. 167, n. 1, pp. 96–115, 11 2005. ISSN: 0377-2217. doi: 10.1016/J.EJOR.2004.01.046. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0377221704002292>.

[164] MARUFUZZAMAN, M., EKSIOGLU, S., HUANG, Y. "Two-stage stochastic programming supply chain model for biodiesel production via wastewater treatment", *Computers and Operations Research*, v. 49, pp. 1–17, 2014. ISSN: 03050548. doi: 10.1016/j.cor.2014.03.010.

[165] REZAEE, A., DEHGHANIAN, F., FAHIMNIA, B., et al. "Green supply chain network design with stochastic demand and carbon price", *Annals of Operations Research*, v. 250, n. 2, pp. 463–485, 3 2017. ISSN: 15729338. doi: 10.1007/s10479-015-1936-z.

[166] FATHOLLAHI-FARD, A., HAJIAGHAEI-KESHTELI, M. "A stochastic multi-objective model for a closed-loop supply chain with environmental considerations", *Applied Soft Computing Journal*, v. 69, pp. 232–249, 8 2018. ISSN: 15684946. doi: 10.1016/j.asoc.2018.04.055.

[167] QUDDUS, M., CHOWDHURY, S., MARUFUZZAMAN, M., et al. "A two-stage chance-constrained stochastic programming model for a bio-fuel supply chain network", *International Journal of Production Economics*, v. 195, pp. 27–44, 1 2018. ISSN: 09255273. doi: 10.1016/j.ijpe.2017.09.019.

[168] SANCI, E., DASKIN, M. "Integrating location and network restoration decisions in relief networks under uncertainty", *European Journal of Operational Research*, v. 279, n. 2, pp. 335–350, 12 2019. ISSN: 03772217. doi: 10.1016/j.ejor.2019.06.012.

[169] WESKAMP, C., KOBERSTEIN, A., SCHWARTZ, F., et al. "A two-stage stochastic programming approach for identifying optimal postponement strategies in supply chains with uncertain demand", *Omega (United Kingdom)*, v. 83, pp. 123–138, 3 2019. ISSN: 03050483. doi: 10.1016/j.omega.2018.02.008.

[170] GUPTA, A., MARANAS, C. "Managing demand uncertainty in supply chain planning", *Computers and Chemical Engineering*, v. 27, n. 8-9, pp. 1219–1227, 9 2003. ISSN: 00981354. doi: 10.1016/S0098-1354(03)00048-6.

[171] BARBAROSOĞLU, G., ARDA, Y. "A two-stage stochastic programming framework for transportation planning in disaster response", *Journal of the Operational Research Society*, v. 55, n. 1, pp. 43–53, 1 2004. ISSN: 0160-5682. doi: 10.1057/palgrave.jors.2601652. Disponível em: <https://www.tandfonline.com/doi/full/10.1057/palgrave.jors.2601652>.

[172] MAQSOOD, I., HUANG, G. H., SCOTT YEOMANS, J. "An interval-parameter fuzzy two-stage stochastic program for water resources management under uncertainty", *European Journal of Operational Research*, v. 167, n. 1, pp. 208–225, 11 2005. ISSN: 0377-2217. doi: 10.1016/J.EJOR.2003.08.068. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0377221704002929>.

[173] KHOR, C., ELKAMEL, A., PONNAMBALAM, K., et al. "Two-stage stochastic programming with fixed recourse via scenario planning with economic and operational risk management for petroleum refinery planning under uncertainty", *Chemical Engineering and Processing: Process Intensification*, v. 47, n. 9-10, pp. 1744–1764, 2008. ISSN: 02552701. doi: 10.1016/j.cep.2007.09.016.

[174] DIMITRAKOPOULOS, R. "Stochastic optimization for strategic mine planning: A decade of developments", *Journal of Mining Science*, v. 47, n. 2, pp. 138–150, 3 2011. ISSN: 1573-8736. doi: 10.1134/S1062739147020018. Disponível em: <https://doi.org/10.1134/S1062739147020018>.

[175] HU, Z., HU, G. "A two-stage stochastic programming model for lot-sizing and scheduling under uncertainty", *International Journal of Production Economics*, v. 180, pp. 198–207, 10 2016. ISSN: 09255273. doi: 10.1016/j.ijpe.2016.07.027.

[176] DILLON, M., OLIVEIRA, F., ABBASI, B. "A two-stage stochastic programming model for inventory management in the blood supply chain", *International Journal of Production Economics*, v. 187, pp. 27–41, 5 2017. ISSN: 09255273. doi: 10.1016/j.ijpe.2017.02.006.

[177] MEGAHED, A., GOETSCHALCKX, M. "Tactical supply chain planning under uncertainty with an application in the wind turbines industry", *Computers and Operations Research*, v. 100, pp. 287–300, 12 2018. ISSN: 03050548. doi: 10.1016/j.cor.2017.12.015.

[178] DOS SANTOS, F., OLIVEIRA, F. "An enhanced L-Shaped method for optimizing periodic-review inventory control problems modeled via two-stage stochastic programming", *European Journal of Operational Research*, v. 275, n. 2, pp. 677–693, 6 2019. ISSN: 03772217. doi: 10.1016/j.ejor.2018.11.053.

[179] NIKZAD, E., BASHIRI, M., OLIVEIRA, F. "Two-stage stochastic programming approach for the medical drug inventory routing problem under uncertainty", *Computers and Industrial Engineering*, v. 128, pp. 358–370, 2 2019. ISSN: 03608352. doi: 10.1016/j.cie.2018.12.055.

[180] ZHANG, J., DIMITRAKOPOULOS, R. "Stochastic optimization for a mineral value chain with nonlinear recovery and forward contracts", *Journal of the Operational Research Society*, v. 69, n. 6, pp. 864–875, 6 2018. ISSN: 0160-5682. doi: 10.1057/s41274-017-0269-5. Disponível em: <`https://www.tandfonline.com/doi/full/10.1057/s41274-017-0269-5`>.

[181] BENDERS, J. "Partitioning prodedures for solving mixed-variables programming problems", *Numerische Mathematik*, v. 4, pp. 238–252, 1962.

[182] FATTAHI, M., GOVINDAN, K., KEYVANSHOKOOH, E. "Responsive and resilient supply chain network design under operational and disruption risks with delivery lead-time sensitive customers", *Transportation Research Part E: Logistics and Transportation Review*, v. 101, pp. 176–200, 5 2017. ISSN: 13665545. doi: 10.1016/j.tre.2017.02.004.

[183] FATTAHI, M., GOVINDAN, K., KEYVANSHOKOOH, E. "A multi-stage stochastic program for supply chain network redesign problem with price-dependent uncertain demands", *Computers and Operations Research*, v. 100, pp. 314–332, 12 2018. ISSN: 03050548. doi: 10.1016/j.cor.2017.12.016.

[184] HADDADSISAKHT, A., RYAN, S. "Closed-loop supply chain network design with multiple transportation modes under stochastic demand and uncertain carbon tax", *International Journal of Production Economics*, v. 195, pp. 118–131, 1 2018. ISSN: 09255273. doi: 10.1016/j.ijpe.2017.09.009.

[185] KÖRPEOĞLU, E., YAMAN, H., AKTÜRK, M. "A multi-stage stochastic programming approach in master production scheduling", *European Journal of Operational Research*, v. 213, n. 1, pp. 166–179, 8 2011. ISSN: 03772217. doi: 10.1016/j.ejor.2011.02.032.

[186] SHABANI, N., SOWLATI, T. "A hybrid multi-stage stochastic programming-robust optimization model for maximizing the supply chain of a forest-based biomass power plant considering uncertainties", *Journal of Cleaner Production*, v. 112, pp. 3285–3293, 2016. ISSN: 09596526. doi: 10.1016/j.jclepro.2015.09.034.

[187] ESCUDERO, L., MONGE, J., MORALES, D. "On the time-consistent stochastic dominance risk averse measure for tactical supply chain planning under uncertainty", *Computers and Operations Research*, v. 100, pp. 270–286, 12 2018. ISSN: 03050548. doi: 10.1016/j.cor.2017.07.011.

[188] ZAHIRI, B., TORABI, S., MOHAMMADI, M., et al. "A multi-stage stochastic programming approach for blood supply chain planning", *Computers and Industrial Engineering*, v. 122, pp. 1–14, 8 2018. ISSN: 03608352. doi: 10.1016/j.cie.2018.05.041.

[189] SAWIK, T. "Two-period vs. multi-period model for supply chain disruption management", *International Journal of Production Research*, v. 57, n. 14, pp. 4502–4518, 7 2019. ISSN: 0020-7543. doi: 10.1080/00207543.2018.1504246. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/00207543.2018.1504246>.

[190] AVILA, D., PAPAVASILIOU, A., LÖHNDORF, N. "Batch learning in stochastic dual dynamic programming". 2021. Disponível em: <https://orbilu.uni.lu/bitstream/10993/49798/1/batchlearningsddp.pdf>.

[191] DEL CASTILLO, M. F., DIMITRAKOPOULOS, R. "Dynamically optimizing the strategic plan of mining complexes under supply uncertainty", *Resources Policy*, v. 60, pp. 83–93, 3 2019. ISSN: 0301-4207. doi: 10.1016/J.RESOURPOL.2018.11.019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0301420718302307>.

[192] PEREIRA, M., PINTO, L. "Multi-stage stochastic optimization applied to energy planning", *Mathematical Programming 1991 52:1*, v. 52, n. 1, pp. 359–375, 5 1991. ISSN: 1436-4646. doi: 10.1007/BF01582895. Disponível em: <https://link.springer.com/article/10.1007/BF01582895>.

[193] GJELSVIK, A., MO, B., HAUGSTAD, A. "Long- and Medium-term Operations Planning and Stochastic Modelling in Hydro-dominated Power Systems Based on Stochastic Dual Dynamic Programming". In: *Handbook of Power Systems I*, Springer, pp. 33–55, Berlin, Heidelberg, 2010.

doi: 10.1007/978-3-642-02493-1{_}2. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-02493-1_2>.

[194] BRIGATTO, A., STREET, A., VALLADÃO, D. "Assessing the cost of time-inconsistent operation policies in hydrothermal power systems", *IEEE Transactions on Power Systems*, v. 32, n. 6, pp. 4541–4550, 11 2016. ISSN: 08858950. doi: 10.1109/TPWRS.2017.2672204.

[195] STREET, A., BRIGATTO, A., VALLADÃO, D. "Co-optimization of energy and ancillary services for hydrothermal operation planning under a general security criterion", *IEEE Transactions on Power Systems*, v. 32, n. 6, pp. 4914–4923, 11 2017. ISSN: 08858950. doi: 10.1109/TPWRS.2017.2672555.

[196] SOARES, M., STREET, A., VALLADÃO, D. "On the solution variability reduction of Stochastic Dual Dynamic Programming applied to energy planning", *European Journal of Operational Research*, v. 258, n. 2, pp. 743–760, 4 2017. ISSN: 0377-2217. doi: 10.1016/J.EJOR.2016.08.068.

[197] FHOULA, B., HAJJI, A., REKIK, M. "Stochastic dual dynamic programming for transportation planning under demand uncertainty", *2013 International Conference on Advanced Logistics and Transport, ICALT 2013*, pp. 550–555, 2013. doi: 10.1109/ICADLT.2013.6568518.

[198] ZOU, J., AHMED, S., SUN, X. "Stochastic dual dynamic integer programming", *Mathematical Programming 2018 175:1*, v. 175, n. 1, pp. 461–502, 3 2018. ISSN: 1436-4646. doi: 10.1007/S10107-018-1249-5. Disponível em: <https://link.springer.com/article/10.1007/s10107-018-1249-5>.

[199] TAI, A., CHING, W. "Recent advances on Markovian models for inventory research", *International Journal of Inventory Research*, v. 3, n. 3, pp. 198, 2016. ISSN: 1746-6962. doi: 10.1504/IJIR.2016.081882.

[200] PAPADOPOULOS, C., LI, J., O'KELLY, M. "A classification and review of timed Markov models of manufacturing systems", *Computers and Industrial Engineering*, v. 128, pp. 219–244, 2 2019. ISSN: 03608352. doi: 10.1016/j.cie.2018.12.019.

[201] HIGGINSON, J., BOOKBINDER, J. "Markovian Decision Processes in Shipment Consolidation", *Transportation Science*, v. 29, n. 3, pp. 242–255, 8 1995. ISSN: 0041-1655. doi: 10.1287/trsc.29.3.242.

[202] SERRATO, M., RYAN, S., GAYTÁN, J. "A Markov decision model to evaluate outsourcing in reverse logistics", *International Journal of Production Research*, v. 45, n. 18-19, pp. 4289–4315, 9 2007. ISSN: 0020-7543. doi: 10.1080/00207540701450161. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/00207540701450161>.

[203] WU, S., LIU, Q., ZHANG, R. "The reference effects on a retailer's dynamic pricing and inventory strategies with strategic consumers", *Operations Research*, v. 63, n. 6, pp. 1320–1335, 11 2015. ISSN: 15265463. doi: 10.1287/opre.2015.1440.

[204] HU, P., SHUM, S., YU, M. "Joint inventory and markdown management for perishable goods with strategic consumer behavior", *Operations Research*, v. 64, n. 1, pp. 118–134, 1 2016. ISSN: 15265463. doi: 10.1287/opre.2015.1439.

[205] YIH, Y., THESEN, A. "Semi-Markov decision models for real-time scheduling", *International Journal of Production Research*, v. 29, n. 11, pp. 2331–2346, 1991. ISSN: 1366588X. doi: 10.1080/00207549108948086.

[206] FERREIRA, G., ARRUDA, E., MARUJO, L. "Inventory management of perishable items in long-term humanitarian operations using Markov Decision Processes", *International Journal of Disaster Risk Reduction*, v. 31, pp. 460–469, 10 2018. ISSN: 2212-4209. doi: 10.1016/J.IJDRR.2018.05.010. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2212420918305247>.

[207] KRISHNAKUMAR, S., ELANGO, C. "Perishable Inventory Control in Retrial Service Facility-Semi Markov Decision Process", *Intern. J. Fuzzy Mathematical Archive*, v. 15, n. 2, pp. 2320–3250, 2018. doi: 10.22457/ijfma.v15n2a15. Disponível em: <www.researchmathsci.orgdoi:http://dx.doi.org/10.22457/ijfma.v15n2a15>.

[208] FEINBERG, E., LEWIS, M. "On the convergence of optimal actions for Markov decision processes and the optimality of (s,S) inventory policies", *Naval Research Logistics (NRL)*, v. 65, n. 8, pp. 619–637, 12 2018. ISSN: 0894069X. doi: 10.1002/nav.21750. Disponível em: <http://doi.wiley.com/10.1002/nav.21750>.

[209] FLEISCHMANN, M., KUIK, R. "On optimal inventory control with independent stochastic item returns", *European Journal of Operational Research*, v. 151, n. 1, pp. 25–37, 11 2003. doi: 10.1016/S0377-2217(02)00592-1.

[210] BIJVANK, M., BHULAI, S., TIM HUH, W. "Parametric replenishment policies for inventory systems with lost sales and fixed order cost", *European Journal of Operational Research*, v. 241, n. 2, pp. 381–390, 3 2015. ISSN: 03772217. doi: 10.1016/j.ejor.2014.09.018.

[211] SUN, W., WANG, Y., ZHANG, F., et al. "Dynamic allocation of surplus by-product gas in a steel plant by dynamic programming with a reduced state space algorithm", *Engineering Optimization*, v. 50, n. 9, pp. 1578–1592, 9 2018. ISSN: 10290273. doi: 10.1080/0305215X.2017.1402013.

[212] SHAHRABI, J., ADIBI, M., MAHOOTCHI, M. "A reinforcement learning approach to parameter estimation in dynamic job shop scheduling", *Computers and Industrial Engineering*, v. 110, pp. 75–82, 8 2017. ISSN: 03608352. doi: 10.1016/j.cie.2017.05.026.

[213] WANG, Y., USHER, J. "Application of reinforcement learning for agent-based production scheduling", *Engineering Applications of Artificial Intelligence*, v. 18, n. 1, pp. 73–82, 2 2005. ISSN: 09521976. doi: 10.1016/j.engappai.2004.08.018.

[214] AYDIN, M., ÖZTEMEL, E. "Dynamic job-shop scheduling using reinforcement learning agents", *Robotics and Autonomous Systems*, v. 33, n. 2, pp. 169–178, 11 2000. ISSN: 09218890. doi: 10.1016/S0921-8890(00)00087-7.

[215] SHIUE, Y., LEE, K., SU, C. "Real-time scheduling for a smart factory using a reinforcement learning approach", *Computers and Industrial Engineering*, v. 125, pp. 604–614, 11 2018. ISSN: 03608352. doi: 10.1016/j.cie.2018.03.039.

[216] SCHÜTZ, H., KOLISCH, R. "Approximate dynamic programming for capacity allocation in the service industry", *European Journal of Operational Research*, v. 218, n. 1, pp. 239–250, 4 2012. ISSN: 03772217. doi: 10.1016/j.ejor.2011.09.007.

[217] GIANNOCCARO, I., PONTRANDOLFO, P. "Inventory management in supply chains: a reinforcement learning approach", *International Journal of Production Economics*, v. 78, n. 2, pp. 153–161, 7 2002. ISSN: 0925-5273. doi: 10.1016/S0925-5273(00)00156-0. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0925527300001560>.

[218] PONTRANDOLFO, P., GOSAVI, A., OKOGBAA, O., et al. "Global supply chain management: A reinforcement learning approach", *International Journal of Production Research*, v. 40, n. 6, pp. 1299–1317, 1 2002. ISSN: 0020-7543. doi: 10.1080/00207540110118640. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/00207540110118640>.

[219] KARA, A., DOGAN, I. "Reinforcement learning approaches for specifying ordering policies of perishable inventory systems", *Expert Systems with Applications*, v. 91, pp. 150–158, 1 2018. ISSN: 09574174. doi: 10.1016/j.eswa.2017.08.046.

[220] MORTAZAVI, A., ARSHADI KHAMSEH, A., AZIMI, P. "Designing of an intelligent self-adaptive model for supply chain ordering management system", *Engineering Applications of Artificial Intelligence*, v. 37, pp. 207–220, 1 2015. ISSN: 09521976. doi: 10.1016/j.engappai.2014.09.004.

[221] CHAHARSOOGHI, S., HEYDARI, J., ZEGORDI, S. "A reinforcement learning model for supply chain ordering management: An application to the beer game", *Decision Support Systems*, v. 45, n. 4, pp. 949–959, 11 2008. ISSN: 0167-9236. doi: 10.1016/J.DSS.2008.03.007. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167923608000560>.

[222] JIANG, C., SHENG, Z. "Case-based reinforcement learning for dynamic inventory control in a multi-agent supply-chain system", *Expert Systems with Applications*, v. 36, n. 3 PART 2, pp. 6520–6526, 2009. ISSN: 09574174. doi: 10.1016/j.eswa.2008.07.036.

[223] FULLER, D., FERREIRA FILHO, V., ARRUDA, E. "Oil industry value chain simulation with learning agents", *Computers and Chemical Engineering*, pp. 199–209, 3 2018. ISSN: 00981354. doi: 10.1016/j.compchemeng.2018.01.008.

[224] FULLER, D., ARRUDA, E., FERREIRA FILHO, V. "Learning-agent - based simulation for queue network systems", *Journal of the Operational Research Society*, pp. 1–17, 9 2019. ISSN: 0160-5682. doi: 10.1080/01605682.2019.1633232.

[225] SHIN, J., LEE, J. "Multi-timescale, multi-period decision-making model development by combining reinforcement learning and mathematical programming", *Computers and Chemical Engineering*, v. 121, pp. 556–573, 2 2019. ISSN: 00981354. doi: 10.1016/j.compchemeng.2018.11.020.

[226] ASTARAKY, D., PATRICK, J. "A simulation based approximate dynamic programming approach to multi-class, multi-resource surgical scheduling", *European Journal of Operational Research*, v. 245, n. 1, pp. 309–319, 8 2015. ISSN: 03772217. doi: 10.1016/j.ejor.2015.02.032.

[227] RETTKE, A., ROBBINS, M., LUNDAY, B. "Approximate dynamic programming for the dispatch of military medical evacuation assets", *European Journal of Operational Research*, v. 254, n. 3, pp. 824–839, 11 2016. ISSN: 03772217. doi: 10.1016/j.ejor.2016.04.017.

[228] ZÉPHYR, L., LANG, P., LAMOND, B., et al. "Approximate stochastic dynamic programming for hydroelectric production planning", *European Journal of Operational Research*, v. 262, n. 2, pp. 586–601, 10 2017. ISSN: 03772217. doi: 10.1016/j.ejor.2017.03.050.

[229] LI, X., WANG, J., FUNG, R. "Approximate dynamic programming approaches for appointment scheduling with patient preferences", *Artificial Intelligence in Medicine*, v. 85, pp. 16–25, 4 2018. ISSN: 18732860. doi: 10.1016/j.artmed.2018.02.001.

[230] MCKENNA, R., ROBBINS, M., LUNDAY, B., et al. "Approximate dynamic programming for the military inventory routing problem", *Annals of Operations Research*, 2019. ISSN: 15729338. doi: 10.1007/s10479-019-03469-8.

[231] VAN ROY, B., BERTSEKAS, D., LEE, Y., et al. "Neuro-dynamic programming approach to retailer inventory management". In: *Proceedings of the IEEE Conference on Decision and Control*, v. 4, pp. 4052–4057. IEEE, 1997. doi: 10.1109/cdc.1997.652501.

[232] KLEYWEGT, A., NORI, V., SAVELSBERGH, M. "The stochastic inventory routing problem with direct deliveries", *Transportation Science*, v. 36, n. 1, pp. 94–118, 2002. ISSN: 00411655. doi: 10.1287/trsc.36.1.94.574.

[233] CHENG, L., DURAN, M. "Logistics for world-wide crude oil transportation using discrete event simulation and optimal control", *Computers & Chemical Engineering*, v. 28, n. 6-7, pp. 897–911, 6 2004. ISSN: 0098-1354. doi: 10.1016/J.COMPCHEMENG.2003.09.025. Disponível em: <https://www.sciencedirect.com/science/article/pii/S009813540300231X>.

[234] BERTSEKAS, D. "Lambda-Policy Iteration: A Review and a New Implementation". In: *Reinforcement learning and approximate dynamic pro-*

gramming for feedback control, John Wiley & Sons, Inc., pp. 381–409, Hoboken, New Jersey, 2013.

[235] HANSEN, E., ZILBERSTEIN, S. "LAO*: A heuristic search algorithm that finds solutions with loops", *Artificial Intelligence*, v. 129, n. 1-2, pp. 35–62, 6 2001. ISSN: 00043702. doi: 10.1016/S0004-3702(01)00106-0.

[236] ALMUDEVAR, A., ARRUDA, E. "Optimal approximation schedules for a class of iterative algorithms, with an application to multigrid value iteration", *IEEE Transactions on Automatic Control*, v. 57, n. 12, pp. 3132–3146, 2012. ISSN: 00189286. doi: 10.1109/TAC.2012.2203053.

[237] ARRUDA, E., OURIQUE, F., LACOMBE, J., et al. "Accelerating the convergence of value iteration by using partial transition functions", *European Journal of Operational Research*, v. 229, n. 1, pp. 190–198, 8 2013. ISSN: 0377-2217. doi: 10.1016/J.EJOR.2013.02.029. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0377221713001628>.

[238] CHANG, H. "Value set iteration for Markov decision processes", *Automatica*, v. 50, n. 7, pp. 1940–1943, 7 2014. ISSN: 0005-1098. doi: 10.1016/J.AUTOMATICA.2014.05.009. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0005109814001848>.

[239] ARRUDA, E., FRAGOSO, M. "A two-phase time aggregation algorithm for average cost Markov decision processes", *Proceedings of the American Control Conference*, pp. 1615–1620, 2012. ISSN: 07431619. doi: 10.1109/ACC.2012.6315187.

[240] REN, Z., KROGH, B. "Markov decision processes with fractional costs", *IEEE Transactions on Automatic Control*, v. 50, n. 5, pp. 646–650, 5 2005. ISSN: 00189286. doi: 10.1109/TAC.2005.846520.

[241] SUN, T., ZHAO, Q., LUH, P. "Incremental value iteration for time-aggregated Markov-decision processes", *IEEE Transactions on Automatic Control*, v. 52, n. 11, pp. 2177–2182, 11 2007. ISSN: 00189286. doi: 10.1109/TAC.2007.908359.

[242] LI, Y., WU, X. "A unified approach to time-aggregated Markov decision processes", *Automatica*, v. 67, pp. 77–84, 5 2016. ISSN: 0005-1098. doi: 10.1016/J.AUTOMATICA.2015.12.022.

[243] XU, Y., CAO, X. "Lebesgue-sampling-based optimal control problems with time aggregation", *IEEE Transactions on Automatic Control*, v. 56, n. 5, pp. 1097–1109, 5 2011. ISSN: 00189286. doi: 10.1109/TAC.2010.2073610.

[244] ARRUDA, E., FRAGOSO, M. "Standard dynamic programming applied to time aggregated Markov decision processes", *Proceedings of the IEEE Conference on Decision and Control*, pp. 2576–2580, 2009. ISSN: 01912216. doi: 10.1109/CDC.2009.5400692.

[245] ARRUDA, E., FRAGOSO, M. "Time aggregated Markov decision processes via standard dynamic programming", *Operations Research Letters*, v. 39, n. 3, pp. 193–197, 5 2011. ISSN: 0167-6377. doi: 10.1016/J.ORL.2011.03.006.

[246] HAUSKRECHT, M., MEULEAU, N., KAELBLING, L. P., et al. "Hierarchical Solution of Markov Decision Processes using Macro-actions". In: *Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI1998)*, pp. 220–229, 1 1998. doi: 10.48550/arxiv.1301.7381.

[247] WAN, Y., CAO, X. "The control of a two-level Markov decision process by time aggregation", *Automatica*, v. 42, n. 3, pp. 393–403, 3 2006. ISSN: 0005-1098. doi: 10.1016/J.AUTOMATICA.2005.11.006.

[248] ARRUDA, E., FRAGOSO, M., OURIQUE, F. "Multi-partition time aggregation for Markov Chains", *2017 IEEE 56th Annual Conference on Decision and Control, CDC 2017*, v. 2018-Janua, pp. 4922–4927, 1 2018. doi: 10.1109/CDC.2017.8264387.

[249] WATKINS, C., DAYAN, P. "Q-learning", *Machine Learning*, v. 8, n. 3-4, pp. 279–292, 5 1992. ISSN: 0885-6125. doi: 10.1007/bf00992698.

[250] DAS, T., GOSAVI, A., MAHADEVAN, S., et al. "Solving semi-Markov decision problems using average reward reinforcement learning", *Management Science*, v. 45, n. 4, pp. 560–574, 1999. ISSN: 00251909. doi: 10.1287/mnsc.45.4.560.

[251] ROBBINS, H., MONRO, S. "A Stochastic Approximation Method". 1951. Disponível em: <`https://www.jstor.org/stable/2236626`>.

[252] BUSONIU, L., BABUSKA, R., DE SCHUTTER, B., et al. *Reinforcement Learning and Dynamic Programming Using Function Approximators*. Boca Raton, CRC Press, 7 2017. ISBN: 9781315217932. doi: 10.1201/9781439821091. Disponível em: <`https://www.taylorfrancis.com/books/9781439821091`>.

[253] WEI, Q., WANG, F., LIU, D., et al. "Finite-approximation-error-based discrete-time iterative adaptive dynamic programming", *IEEE Transactions on Cybernetics*, v. 44, n. 12, pp. 2820–2833, 12 2014. ISSN: 21682267. doi: 10.1109/TCYB.2014.2354377.

[254] RYZHOV, I., MES, M., POWELL, W., et al. "Bayesian exploration for approximate dynamic programming", *Operations Research*, v. 67, n. 1, pp. 198–214, 1 2019. ISSN: 15265463. doi: 10.1287/opre.2018.1772.

[255] BLONDEL, V., TSITSIKLIS, J. "A survey of computational complexity results in systems and control", *Automatica*, v. 36, n. 9, pp. 1249–1274, 9 2000. ISSN: 0005-1098. doi: 10.1016/S0005-1098(00)00050-9. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0005109800000509>.

[256] VALE. "Mina do Cauê". 2022. Disponível em: <http://www.vale.com/brasil/PT/initiatives/environmental-social/mina-do-caue/Paginas/default.aspx>.

[257] EDITORA ABRIL. "O tesouro da Serra de Carajás". 2016. Disponível em: <https://super.abril.com.br/historia/o-tesouro-da-serra-de-carajas/>.

[258] LEITE, J., ARRUDA, E., BAHIENSE, L., et al. "Mine-to-client planning with Markov Decision Process". In: *2020 European Control Conference (ECC)*, pp. 1123–1128, 2020.

[259] DOWSON, O., KAPELEVICH, L. "SDDP.jl: A Julia Package for Stochastic Dual Dynamic Programming", *https://doi.org/10.1287/ijoc.2020.0987*, v. 33, n. 1, pp. 27–33, 8 2020. ISSN: 15265528. doi: 10.1287/IJOC.2020.0987. Disponível em: <https://pubsonline.informs.org/doi/abs/10.1287/ijoc.2020.0987>.

[260] DOWSON, O. "The policy graph decomposition of multistage stochastic programming problems", *Networks*, v. 76, n. 1, pp. 3–23, 7 2020. ISSN: 1097-0037. doi: 10.1002/NET.21932. Disponível em: <https://onlinelibrary.wiley.com/doi/full/10.1002/net.21932https://onlinelibrary.wiley.com/doi/abs/10.1002/net.21932https://onlinelibrary.wiley.com/doi/10.1002/net.21932>.