UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE CIÊNCIAS

DEPARTAMENTO DE ESTATÍSTICA E MATEMÁTICA APLICADA

PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM E MÉTODOS QUANTITATIVOS

MESTRADO ACADÊMICO EM MODELAGEM E MÉTODOS QUANTITATIVOS

ARTHUR HERMONT FONSECA MURTA

APLICAÇÃO DE TÉCNICAS DE APRENDIZADO POR REFORÇO NA SOLUÇÃO DO PROBLEMA DE CORTE DE ESTOQUE MULTIPERÍODO ESTOCÁSTICO

FORTALEZA

2021

ARTHUR HERMONT FONSECA MURTA

APLICAÇÃO DE TÉCNICAS DE APRENDIZADO POR REFORÇO NA SOLUÇÃO DO PROBLEMA DE CORTE DE ESTOQUE MULTIPERÍODO ESTOCÁSTICO

Dissertação apresentada ao Curso de Mestrado Acadêmico em Modelagem e Métodos Quantitativos do Programa de Pós-Graduação em Modelagem e Métodos Quantitativos do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Modelagem e Métodos Quantitativos. Área de Concentração: Modelagem e Métodos Quantitativos.

Orientador: Prof. Dr. Anselmo Ramalho Pitombeira Neto.

FORTALEZA

2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M965a Murta, Artur Hermont Fonseca.
Aplicação de técnicas de aprendizado por reforço na solução do problema de corte de estoque multiperíodo estocástico / Artur Hermont Fonseca Murta. – 2021.
83 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Modelagem e Métodos Quantitativos, Fortaleza, 2021.
Orientação: Prof. Dr. Anselmo Ramalho Pitombeira Neto.

1. Problema de corte de estoque estocástico. 2. Aprendizado por reforço. 3. Programação dinâmica aproximada. 4. Otimização combinatória. I. Título.

CDD 510

ARTHUR HERMONT FONSECA MURTA

APLICAÇÃO DE TÉCNICAS DE APRENDIZADO POR REFORÇO NA SOLUÇÃO DO PROBLEMA DE CORTE DE ESTOQUE MULTIPERÍODO ESTOCÁSTICO

Dissertação apresentada ao Curso de Mestrado Acadêmico em Modelagem e Métodos Quantitativos do Programa de Pós-Graduação em Modelagem e Métodos Quantitativos do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Modelagem e Métodos Quantitativos. Área de Concentração: Modelagem e Métodos Quantitativos.

Aprovada em: 26/03/2021

BANCA EXAMINADORA

Prof. Dr. Anselmo Ramalho Pitombeira Neto (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Arthur Plinio de Souza Braga
Universidade Federal do Ceará (UFC)

Prof. Dr. Bruno de Athayde Prata
Universidade Federal do Ceará (UFC)

À minha família.

AGRADECIMENTOS

Ao Prof. Dr. Anselmo Ramalho Pitombeira Neto, por me orientar em minha dissertação de Mestrado. Obrigado pela confiança e por me atender com paciência todas as vezes que bati em sua porta. Agradeço por todos os ensinamentos compartilhados de forma admirável, e por me guiar nos primeiros passos da pós-graduação. Muito obrigado por tudo!

Aos meus pais, irmãos e cunhado, que nos momentos de minha ausência dedicados ao estudo superior, sempre fizeram entender que o futuro é feito a partir da constante dedicação no presente!

Aos membros do OPL – Pesquisa Operacional em Produção e Logística – pelas discussões muito produtivas.

A todos os amigos e colegas que de uma forma direta ou indireta, contribuíram, ou auxiliaram na elaboração do presente estudo, pela paciência, atenção e força que prestaram em momentos menos fáceis. Em especial a Slymara e a Thais pela revisão e criticas à esse trabalho.

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

“Sem dados você é apenas mais uma pessoa com uma opinião.” (DEMING apud KELLER, 2012, p. 388)

RESUMO

O problema de corte de estoque é um problema de otimização combinatória que consiste em cortar objetos maiores a fim de produzir peças menores para atender uma dada demanda de forma a minimizar as perdas de material. Nesta dissertação, aborda-se uma variante multiperíodo estocástica na qual o problema é resolvido em múltiplos períodos de tempo e não se sabe exatamente qual é a demanda futura, a qual é modelada como uma variável aleatória. Essa variante corresponde mais fielmente à realidade das empresas, em que normalmente não se conhece antecipadamente a demanda em cada período de tempo. Primeiramente, o problema de corte de estoque multiperíodo estocástico foi modelado como um processo de decisão markoviano. Uma solução para o problema corresponde a uma política de decisão ótima, a qual é definida como sendo qual ação a ser tomada a cada período de tempo de forma a minimizar o custo esperado total a longo prazo. Algoritmos exatos para calcular uma política ótima requerem grande esforço computacional quando o tamanho do problema cresce, por isso foram utilizadas técnicas de aprendizado por reforço por meio de um algoritmo de iteração de política aproximada com o uso de um filtro bayesiano. Experimentos computacionais foram realizados para ilustrar a aplicação da abordagem com uso de dados reais de corte de barras de aço em obras de construção civil. Os resultados indicam que o desempenho da política obtida pela abordagem proposta foi até cinquenta vezes melhor do que o desempenho utilizando uma política míope, a qual não leva em conta o impacto futuro de decisões tomadas no presente.

Palavras-chave: problema de corte de estoque estocástico; aprendizado por reforço; programação dinâmica aproximada; otimização combinatória.

ABSTRACT

The cutting stock problem is a combinatorial optimization problem that consists of cutting larger objects in order to produce smaller pieces to meet a given demand in order to minimize material losses. This dissertation addresses up a multiperiod stochastic variant in which the problem is solved in multiple periods of time and we do not know exactly what the future demand will be, which is modeled as a random variable. This problem variant corresponds more closely to the reality of companies, which usually do not know in advance the demand for each time period. First, the stochastic multiperiod cutting stock problem was modeled as a Markovian decision process. A solution to the problem corresponds to an optimal decision policy, which is defined as what action to be taken every time to minimize the expected total cost. Exact algorithms to calculate an optimal policy require large computational effort when the problem size grows, then reinforcement learning techniques were used through an approximate policy iteration algorithm using a Bayesian filter. Computational experiments were performed to illustrate the application of the approach to real data on cutting steel bars in construction industry. The results indicate that the performance of the policy obtained by the proposed approach was up to fifty times better than the performance using a short-sighted policy, which does not take into account the future impact of decisions taken in the present.

Keywords: stochastic cutting stock problem; reinforcement learning; approximate dynamic programming; combinatorial optimization.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

DINCON 2019	Conferência Brasileira de Dinâmica, Controle e Aplicações
DQN	Deep Q-Network
MDP	Processo de Decisão Markoviano
RL	Aprendizado por Reforço

LISTA DE SÍMBOLOS

$P(B\|A)$	Probabilidade condicional de B em relação a A
S	Conjunto de estados
A	Conjunto de ações
R	Conjunto de recompensas
t	T-ésimo período de tempo
S_t	Estado no tempo t
A_t	Ação no tempo t
R_t	Recompensa obtida no tempo t
L	Tamanho do objeto que deve ser cortado
i	Índice do itens que devem ser produzidos
l_i	Tamanho do item i
N_i	Número de peça do item i que devem ser produzidas
J	Conjunto de possíveis padrões de corte
j	Possíveis padrões de corte
a_{ij}	Número de peças do item i no padrão j
p_j	Perda correspondente ao padrão j
x_j	Número de vezes que o padrão j foi cortado
π	Uma política arbitrária
v_π	Avaliação da política π
$\pi(a\|s)$	Chande de tomar a ação a estando no estado s de acordo com a política π
$p(s', r\|s, a)$	Probabilidade de transicionar para o estado s' com a recompensa r estando no estado s e tomando a ação a
v_0	Valor inicial para avaliação de política
v_k	Avaliação da política π na k-ésima iteração da Equação 2.8
$\pi(s)$	Ação que deve tomada no estado s de acordo com a política π
a	Ação arbitraria tomada no estado s

$q_\pi(s,a)$	Valor esperado de tomar a ação a no estado s e apos isso seguir a ação indicada por $\pi(s)$
π'	Política gananciosa
π_*	Política ótima
v_{π_*}	Avaliação da política ótima
$\xrightarrow{A}$	Avaliação da política
$\xrightarrow{M}$	Melhoria da política
$\Theta(f)$	Ordem assintótica da função f
$F(n)$	N-ésimo número da sequência de Fibonacci
$R(s_t,a_t)$	Recompensa de estar no estado s_t e tomar a ação a_t
$P^a_{ss'}$	Matriz de transição com as probabilidades de transicionar de um estado $s_t = s$ para o estado $s_{t+1} = s'$ após a decisão $a_t = a$
$a_t \sim \pi(s_t;\theta)$	a_t segue a distribuição com os parâmetros s_t e θ
γ	Fator de desconto
$q(s,a)$	Valor esperado da recompensa ao tomar a ação a estando no estado s
$\mathscr{I}$	Conjunto dos possíveis itens de cada tamanho
l_I	Tamanho do i-ésimo item
d_t	Demanda no período t
d_{It}	Demanda do i-ésimo item no período t
r_t	Vetor de estoque no inicio do período t
r_{It}	Estoque do item I no inicio do período t
$r^+{}_t$	Vetor de estoque no final do período t
r^+_{It}	Estoque do item I no final do período t
$r^-{}_t$	Vetor de demanda não atendida no período t
r^-_{It}	Demanda não atendida do item I no período t
$h^+{}_t$	Vetor de custo associado ao manter estoque no final do período t
h^+_{It}	Custo associado ao manter o item I no final do período t
$h^-{}_t$	Vetor do custo associado a demanda não atendida no período t

h_{It}^{-}	Custo associado ao não atender do item I no período t
m_1	Número máximo de objetos que podem ser cortados
m_2	Vetor com a capacidade máxima de cada item no inventário
$c(\mathbf{s}, \mathbf{x})$	Função de contribuição de tomar a ação a ação x durante o estado s

SUMÁRIO

1 INTRODUÇÃO

O problema de corte de estoque é um problema de otimização combinatória que consiste em cortar objetos maiores a fim de produzir peças menores para atender uma dada demanda enquanto otimiza uma função objetivo, que pode ser, minimizar as perdas de material (KANTOROVICH, 1960; KANTOROVICH; ZALGALLER, 1951). Esses problemas são muito importantes no planejamento da produção em muitas indústrias, como as indústrias barras de aço, placas de metal ou madeira, chapas de vidro, plástico, de bobinas de papel ou alumínio, entre outros.

A versão multiperíodo do problema de corte de estoque consiste basicamente em resolver a cada período de tempo um problema de corte de estoque. Para atender uma demanda de itens, pode-se ou não antecipar a demanda da produção, isso permite produzir de forma antecipada alguns itens, que ao produzi-los agora geram uma menor perda de material (POLDI; ARENALES, 2010) .

A versão multiperíodo estocástica consiste em não saber exatamente qual é essa demanda futura, a qual é modelada como uma variável aleatória. Essa variante corresponde mais fielmente à realidade das empresas, em que normalmente não se conhece antecipadamente a demanda em cada período de tempo. Ao tratar esse problema como o problema clássico (determinístico) e tentar resolvê-lo de forma eficiente a cada período de tempo, pode-se gerar uma acúmulo de itens ou ter perdas materiais desnecessárias. A motivação para este trabalho é tentar contribuir com uma forma eficiente de tratar esse problema, pois o acúmulo de estoque e as perdas de material podem impactar financeiramente as empresas de forma muito significativa.

Os objetivos desse trabalho de mestrado são: Propor uma variante multiperíodo estocástica do problema clássico de corte de estoque, o qual deve ser otimizado em vários períodos subsequentes e a demanda em cada período de tempo é incerta; propor uma formulação do problema de corte de estoque multiperíodo estocástico como um problema de decisão markoviano; e desenvolver um algoritmo de iteração de política aproximada, que é uma técnica de aprendizado por reforço, para resolver o problema proposto.

Para avaliar a abordagem de solução proposta, foram realizados alguns experimentos computacionais, que comparam a política proposta com uma política míope que desconsidera a influência das demandas futuras e consiste na solução do problema clássico determinístico a cada período de tempo. Os resultados apresentados nessa dissertação são uma extensão de resultados apresentados no artigo “Uma abordagem baseada em programação dinâmica aproximada para o

problema de corte de estoque multiperíodo estocástico", publicado na Conferência Brasileira de Dinâmica, Controle e Aplicações - Conferência Brasileira de Dinâmica, Controle e Aplicações (DINCON 2019).

O restante do trabalho é organizado da seguinte forma. No capítulo 2, realiza-se uma revisão sobre conceitos relevantes ao trabalho, tais como: otimização combinatória, o problema do corte de estoque, programação dinâmica e aprendizado por reforço. No capítulo 3 apresenta-se a formulação matemática e a metodologia adotada. No capítulo 4 discutem-se os resultados obtidos a partir de dados reais e simulados. Por fim, no capítulo 5 apresentam-se as conclusões dos resultados alcançados nesse trabalho.

2 REVISÃO DA LITERATURA

2.1 Otimização Combinatória

O problema geral de otimização combinatória pode ser definido como: seja $\mathscr{F}$ uma família de subconjuntos de um conjunto finito $\mathbf{E}$ e seja $w : \mathbf{E} \rightarrow \mathscr{R}$ uma função de peso definida nos elementos de $\mathbf{E}$. O objetivo do problema de otimização combinatória é encontrar $\mathbf{F}^* \in \mathscr{F}$ de modo que $w(\mathbf{F}^*) = min_{\mathbf{F} \in \mathscr{F}} w(\mathbf{F})$ onde $w(\mathbf{F}) := \sum_{e \in \mathscr{F}} w(e)$ (PAPADIMITRIOU; STEIGLITZ, 1998). Em outras palavras, a otimização combinatória consiste em encontrar um subconjunto $\mathbf{F}$ de todos os possíveis subconjuntos $\mathscr{F}$, que respeitam as restrições do problema, que minimiza a função objetivo.

Na seção a seguir, descrevem-se algumas das aplicações da otimização combinatória.

2.1.1 Aplicações

Nessa seção serão apresentados exemplos de aplicações de problemas de otimização combinatória.

O problema do corte de materiais pode ser que fábrica têxtil precisa cortar um tecido grande para a confecção de algumas peças de roupa, porém sempre há desperdício de material ao fazer tais cortes. Os pedaços que sobram não são grandes o suficiente para produzir novas peças. O objetivo dessa análise é descobrir como cortar esse tecido de forma que minimize o desperdício desse material. Esse é o problema de corte em duas dimensões, mas existe também a versão dele em uma e três dimensões.

O problema de empacotamento é o inverso do problema de corte de materiais onde o objetivo é determinar a melhor maneira de agregar objetos de forma que ocupem o menor espaço possível. De vez em quando neste problema o tamanho máximo é predefinido e o objetivo é guardar o maior número de objetos neste compartimento. No transporte de cargas em um navio o objetivo é minimizar o número de contêineres necessários para transportar a carga.

No problema de escalonamento da mão-de-obra, há um conjunto de tarefas a serem feitas por um grupo de funcionários habilitados. A decisão a ser tomada no problema é quais funcionários farão quais tarefas de forma a minimizar os custos, seja com mão de obra profissional, estrutura ou atividades extras remuneradas. Esse problema geralmente tem restrições associadas a limites de cargas de trabalho, tempo máximo entre plantões etc.

O escalonamento de tarefas é o problema onde, dentro de uma determinada fábrica,

diversas pequenas tarefas são necessárias para a produção de um produto. Essas tarefas tem regras de precedência e de configuração entre si. Assim, temos que alocá-las em cada máquina disponível de forma que minimizemos os custos e o tempo de produção.

O problema de localização de facilidades por ser exemplificado por uma fábrica de café que quer se expandir para uma nova cidade, portanto ela precisa entregar o café para alguns supermercados, nesta cidade tem alguns locais disponíveis para ela montar o seu centro de distribuição, deseja-se determinar qual o local aonde ela pode montar o seu centro de distribuição de forma que todos os clientes sejam atendidos a um custo mínimo.

Já no problema de distribuição de bens de consumo essa mesma fábrica do exemplo anterior, precisa agora entregar o café que esta nos seus centros de distribuição para alguns supermercados da região. A fábrica terá que decidir quais caminhões irão atender cada cliente, otimizando quais rotas serão utilizadas, para que se gaste o mínimo de combustível e as entregas sejam efetuadas na data prevista.

Ao projetar uma placa de circuito impresso tem que se decidir o posicionamento dos componentes de forma a minimizar sua área total e o tamanho das ligações internas de forma a minimizar os gastos com a impressão. Mas essas placas estão sujeitas a restrições de dissipação térmica, tamanho máximo de algumas ligações de alta frequência, entre outras restrições.

O planejamento da produção é realizado para atender a demanda por um produto ou serviço em uma determinada janela de tempo. É preciso decidir quanto que vai ser produzido para atender toda a demanda e minimizar os custos. Existem restrições de quanto produtos pode-se produzir por vez e o quanto pode-se estocar. O não atendimento de uma demanda pode ou não ser aceitável visto que esses produtos podem possuir validade curta ou podem acontecer atrasos na entrega.

2.1.2 Técnicas

Os problemas de otimização combinatória precisam ser modelados matematicamente para que seja possível usar recursos computacionais para resolvê-los. Algumas técnicas de resolver esses problemas são:

– Programação Linear e Programação Inteira: A programação linear proposta por Leonid Kantorovich consiste em escrever o problema com as variáveis de decisão e as restrições lineares que são imposta sobre elas, o que queremos com essas variáveis de decisão é encontrar quais devem ser o valor delas para minimizarmos uma função custo sem

desobedecer as restrições do problema (KANTOROVICH, 1940) . O calculo desse sistema pode ser feito através de um algoritmo chamado Simplex que é bem eficiente para essa classe de problemas. Mas nem sempre todas as variáveis de decisão podem ser contínuas, como, por exemplo, se temos que decidir quantas pessoas serão alocadas para uma determinada função. Uma solução como "2.6 pessoas" não é uma boa resposta, e além disso o arredondamento nesse tipo de problema nem sempre é uma boa solução. Nesse tipo de problema é mais adequada a programação inteira, a qual tem como base resolver relaxações do problema original até que se obtenha uma solução satisfatória. As principais técnicas utilizadas são o método *branch-and-bound* e o método de planos de corte.

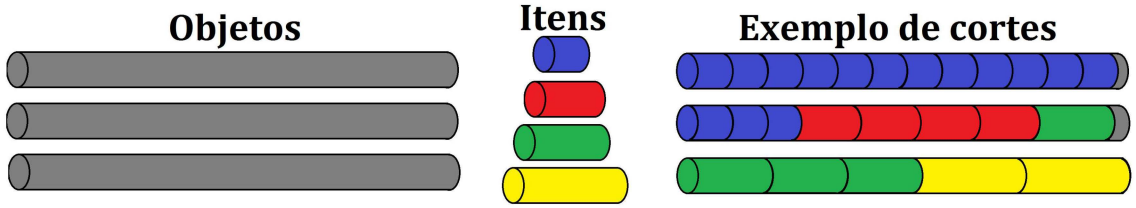
- Programação por Restrições: A programação por restrições proposta por Jaffar e Lassez (JAFFAR; LASSEZ, 1987) é baseada na programação lógica de restrição, que incorpora restrições na programação lógica (MCCARTHY, 1958). Essa técnica consiste em utilizar inferência logica para agrupar as restrições impostas sobre as variáveis do problema. Restrições complexas podem ser escritas de forma simples por ser modelado com uma linguagem declarativa (MIYAZAWA; SOUZA, 2015).
- Métodos Heurísticos: Em problemas muito complexos encontrar a solução exata é muito custoso computacionalmente. Quando a solução ótima não é necessária, os métodos heurísticos podem ser utilizados. Mesmo eles não garantindo encontrar a solução ótima para o problema, são capazes de retornar uma solução satisfatória em um tempo adequado.
- Algoritmos Aproximados: Existem alguns problemas em que se faz necessário saber quão boa uma solução é comparada com a solução ótima. Os métodos heurísticos sozinhos não conseguem fornecer essa informação. Nesses casos, algoritmos aproximados podem ser utilizados. Diferente dos métodos heurísticos, onde não é possível saber o quão longe ou perto a solução obtida está da solução mais otimizada, os algoritmos aproximados conseguem obter essa informação.

2.2 Problema de Corte de Estoque

O problema de corte de estoque (GILMORE; GOMORY, 1961) é um problema muito estudado desde os anos 60. Ele é encontrado em várias indústrias, como a têxtil (FARLEY, 1988), a de papel (POLTRONIERE *et al.*, 2008; RESPICIO *et al.*, 2002), a metalúrgica (HENDRY *et al.*, 1996; NONåS; THORSTENSON, 2000), a de movelaria (GRAMANI, 2001) entre outras.

Esse problema é ilustrado na Figura 1, consiste em escolher como cortar itens a partir de objetos maiores que estão disponíveis para o corte de forma a atender uma demanda desses itens e ao mesmo tempo minimizar os custos e as perdas nos cortes (MURTA; PITOMBEIRA NETO, 2019). Na sua formulação clássica como um problema de programação linear inteiro, considera-se que as demandas são determinísticas e a decisão é tomada somente para um período de tempo.

Figura 1 – Ilustração do problema de corte de estoque



Fonte: (COSTA, 2016).

A formulação do problema clássico de corte de estoque unidimensional (PAULL, 1956; GILMORE; GOMORY, 1961; LIROV, 1992; WäSCHER *et al.*, 2007) pode ser definida como: dado um objeto de tamanho L, devem ser cortadas N_i peças menores de tamanho l_i, $i \in \{1,2,...,m\}$ de forma a atender a demanda e desperdiçando a menor quantidade de material possível. Seja a_{ij} o número de peças do tipo i cortadas no padrão j, e p_j a perda correspondente ao padrão j. O conjunto de possíveis padrões J deve satisfazer as seguintes restrições:

$$\sum_{i=1}^{m} a_{ij} l_i + p_j = L, \quad \forall j \in J. \tag{2.1}$$

Seja x_j o número objetos em estoque cortados seguindo o padrão j. O problema de corte de estoque é definido pelo seguinte modelo de programação linear inteira:

$$\min \quad \sum_{j \in J} p_j x_j, \tag{2.2}$$

s.a.

$$\sum_{j \in J} a_{ij} x_j \geq N_i, \quad i = 1,...,m, \tag{2.3}$$

$$x_j \in \{0,1,2,\dots\}, \quad j \in J, \tag{2.4}$$

o que significa que pelo menos N_i peças do item i devem ser produzidas.

2.2.1 Extensões do problema de corte de estoque

O problema padrão de corte de estoque pode ser estendido para a sua versão multiperíodo em que a demanda vai ser atendida durante múltiplos períodos de tempo (POLDI;

ARENALES, 2010; PRATA *et al.*, 2015; PITOMBEIRA-NETO; PRATA, 2019; BIRGIN *et al.*, 2020; POLDI; ARAUJO, 2016; MA *et al.*, 2019). Ao modelar esse problema, durante a tomada de decisão em que se escolhe quanto de quais padrões de corte deve ser selecionado a cada período de tempo, as demandas futuras já foram definidas e elas são conhecidas. Portanto, pode-se antecipar o corte de alguns itens de forma que o custo de manter esse item guardado seja menor que o custo de produzir esse item quando ele vai ser requisitado.

Uma outra modificação do problema de corte de estoque assume que a demanda é desconhecida. Ao modelar esse problema, durante a tomada de decisão a demanda futura não é conhecida, se conhece apenas uma previsão de demanda. Isso faz com que exista a possibilidade da demanda real ser maior ou menor que os itens disponíveis, podendo ficar uma parte da demanda sem ser atendida ou sobra de estoque. Essa falta geralmente é muito penalizada. Essa modificação é explorada em (KRICHAGINA *et al.*, 1998; ALEM *et al.*, 2010), por meio de um modelo de programação estocástica, mas os autores consideram somente um período de decisão, e não múltiplos períodos.

2.3 Programação Dinâmica

Programação dinâmica (BELLMAN, 1954) é um método de resolução de problemas complexos, utilizada em problemas que podem ser subdivididos de forma temporal ou de forma sequencial. Após quebrar esses problemas em subproblemas e depois combinar as soluções desses subproblemas, obtemos a solução do problema inicial. Para se poder utilizar a programação dinâmica em um problema, o mesmo tem que ter as seguintes propriedades:

- O problema principal pode ser quebrado em subproblemas e ao combinar a solução ótima desses subproblemas resolve-se o problema principal .
- Os subproblemas são recorrentes portanto um mesmo subproblema aparece diversas vezes ao subdividirmos o problema principal. Guardar a solução desses subproblemas em memória acelera a resolução do problema principal.

O processo de decisão markoviano satisfaz essas duas propriedades. A ideia principal em programação dinâmica e em aprendizado por reforço (seção 2.4) é utilizar função de valores para procurar por políticas ótimas.

2.3.1 Processo de decisão markoviano

O processo de decisão markoviano é um método de tomada de decisão de forma sequenciada. Nesse processo, tem-se um agente tomador de decisão que interage com o ambiente, e essas interações ocorrem de forma sequencial ao longo do tempo. A cada instante de tempo o agente irá observar o ambiente e coletará as informações do estado atual. Dadas essas informações o agente toma uma decisão. Como resultado da interação com o agente, o estado realizará uma transição para um novo estado e o agente receberá uma recompensa pela sua ação executada.

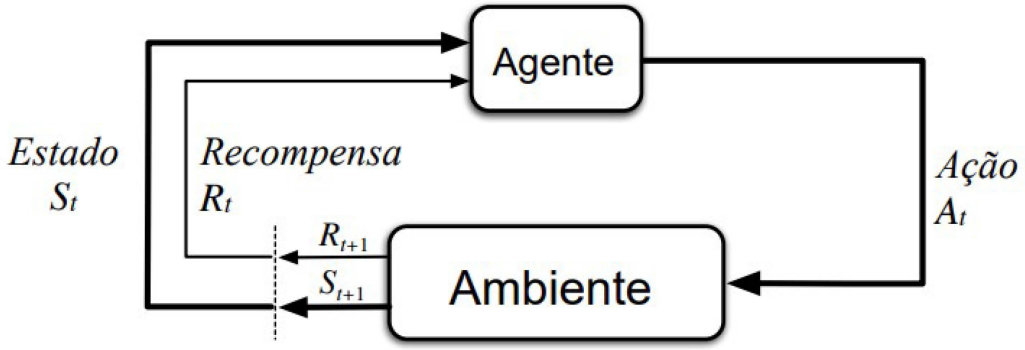
A sequência de estar em um estado, tomar uma ação, transicionar para um novo estado e receber um recompensa por essa transição, acontecendo de formas sucessiva, cria uma trajetória. Durante todo o processo o objetivo do agente é maximizar a recompensa total a ser recebida por tomar as ações. Isso significa que ele não quer só maximizar a recompensa imediata como também a recompensa acumulada que ele recebera ao longo das tomadas de decisões.

No processo de decisão markoviano, tem-se um conjunto de estados $\mathscr{S}$, um conjunto de ações $\mathscr{A}$, e uma função de recompensa R. A cada período de tempo $t = 0,1,2,...$ o agente recebe uma representação do ambiente como sendo $S_t \in \mathscr{S}$. Baseado nesse estado, o agente seleciona uma ação $A_t \in \mathscr{A}$. No próximo período de tempo $t+1$ o ambiente irá transicionar para um novo estado $S_{t+1} \in S$. Ao chegar no novo estado o agente receberá uma recompensa numérica $R_{t+1} \in R$ por ter tomado a ação A_t quando estava no estado S_t. A interação do agente com o ambiente produzirá uma trajetória:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, S_3, A_3, ... \tag{2.5}$$

A Figura 2 representa um diagrama dessa dinâmica.

Figura 2 – Diagrama da interação agente-ambiente como um processo de decisão markoviano



Fonte: (SUTTON; BARTO, 2018a). Traduzida pelo autor.

Em um processo de decisão markoviano finito, os conjuntos de estado, ação e recompensas tem um número finito de elementos, isso faz com que as variáveis aleatórias R_t e S_t tenham distribuição de probabilidade discreta bem definidas sendo dependente apenas do estado anterior e a ação tomada. A probabilidade de haver uma transição para um estado $s' \in \mathscr{S}$ e da recompensa assumir um valor $R = r$, dado que o estado no tempo t é $s \in \mathscr{S}$ e a ação tomada é $a \in \mathscr{A}$ é dada por:

$$p(s', r|s, a) \doteq \Pr\left\{S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a\right\}, \quad \forall s', s \in \mathscr{S}, r \in \mathscr{R}, a \in \mathscr{A}. \tag{2.6}$$

A função de transição p define a dinâmica do processo de decisão markoviano (SUTTON; BARTO, 2018a; MNIH *et al.*, 2015).

2.3.2 Avaliação de política

Dada uma política de decisão π, a obtenção de seu valor $v_\pi(s)$ para cada estado é conhecido como a avaliação da política π, e pode ser realizado por meio da Equação de Bellman:

$$v_\pi(s) \doteq \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_\pi(s')\right], \forall s \in \mathscr{S}, \tag{2.7}$$

onde $\pi(a|s)$ é a probabilidade de tomar a ação a estando no estado s de acordo com a política π, $p(s', r|s, a)$ é a probabilidade de transicionar para o estado s' com a recompensa r estando no estado s e tomando a ação a. A resolução da Equação 2.7 é muito custosa computacionalmente, portanto métodos de solução iterativos são mais indicados. O valor inicial para v_0 pode ser escolhido de forma arbitrária e a regra de atualização do valor da avaliação da política pode ser obtida aplicando a Equação de Bellman para v_π (2.7) como uma regra de atualizaçãoA

$$v_{k+1}(s) \doteq \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_k(s')\right], \forall s \in \mathscr{S}, \tag{2.8}$$

Quando $k \to \infty$, o valor de v_k converge para o valor de v_π, quando as condições de garantia da existência de v_π são atendidas. Esse algoritmo é chamado de avaliação de política interativa.

2.3.3 Melhoria de política

Um dos motivos de calcular a função de valor de uma política é para podermos melhora-la de forma incremental. Supondo que uma política π tem uma função de valor v_π, e para algum estado s gostaria de saber se tomar a ação $a \neq \pi(s)$ será melhor ou pior que seguir

a ação $\pi(s)$ para o estado s. Uma forma de descobrir isso seria selecionar a em s e após isso sempre seguir a política π. O valor esperado dessa maneira é

$$q_\pi(s,a) \doteq \sum_{s',r} p(s',r|s,a)\left[r+\gamma v_\pi(s')\right] \tag{2.9}$$

Caso o valor de $q_\pi(s,a)$ for maior que o valor de $v_\pi(s)$ significa que é melhor selecionar a em s e depois disso seguir π, do que seguir π sempre. Isso implica que é esperado que a política que toda vez está no estado s e seleciona a ação a e no restante dos estados tenha o mesmo comportamento da política π seja melhor que a política π.

Dada uma política e a sua função de valor, é possível avaliar a mudança da política em cada um dos estados possíveis, e em cada um deles selecionar a ação que tem a melhor avaliação de acordo com $q_\pi(s,a)$. Essa nova política, chamada de política gananciosa π', é

$$\pi'(s) \doteq_{a\in\mathscr{A}} q_\pi(s,a) \tag{2.10}$$

$$=_{a\in\mathscr{A}} \sum_{s',r} p(s',r|s,a)\left[r+\gamma v_\pi(s')\right] \tag{2.11}$$

onde $argmax_a$ é o valor de a que maximiza a expressão. Se dois ou mais valores de a maximizam a expressão qualquer um deles pode ser selecionado. A política gananciosa escolhe a melhor ação a curto prazo de acordo com v_π. A criação de uma nova política que vai ser tão boa ou melhor que a política original ao selecionar ações que maximizam a função de valor respeitando da política original, é conhecida como melhoria de política.

Supondo que uma nova política gananciosa, π', é tão boa, mas não melhor que a antiga política π. Portanto $v_\pi = v_{\pi'}$, e a partir de 2.11 para todo estado $s \in \mathscr{S}$:

$$v_{\pi'} = \max_a \sum_{s',r} p(s',r|s,a)\left[r+\gamma v_{\pi'}(s')\right] \tag{2.12}$$

Como essa é a equação da otimalidade de Bellman, então, $v_{\pi'}$ deve ser igual à função de valor ótima v_* e tanto π quanto π' devem ser políticas ótimas. A melhoria de política deve sempre nos dar políticas melhores que a original a não ser que a política original seja ótima.

2.3.4 Iteração de política

Uma vez que tenhamos a política π e ela seja melhorada usando v_π para a política π', podemos calcular $v_{\pi'}$ e com esse valor calcularmos a ainda melhor política π''. Com isso obtemos a sequência monotônica de melhoria de política e a sua função de valor:

$$\pi_0 \xrightarrow{A} v_{\pi_0} \xrightarrow{M} \pi_1 \xrightarrow{A} v_{\pi_1} \xrightarrow{M} \pi_2 \xrightarrow{A} \cdots \xrightarrow{M} \pi_* \xrightarrow{A} v_{\pi_*} \tag{2.13}$$

onde $\xrightarrow{A}$ é a avaliação da política e $\xrightarrow{M}$ é a melhoria da política. Cada política é melhor que a anterior, a não ser que já seja a ótima. Como o processo de decisão markoviano tem um número finito de políticas, esse processo converge para a política ótima em um número finito de iterações. Essa maneira de encontrar a política ótima é chamada de iteração de política. O Algoritmo 1 descreve de forma iterativa como podemos encontrar a política ótima (SUTTON; BARTO, 2018b).

Algoritmo 1: Iteração de politica para estimar $\pi \approx \pi_*$

início
  Inicializa $V(s)$ e $\pi(s)$, $\forall s \in \mathscr{S}$, com valores arbitrários
  enquanto $Politica - estavel \neq verdade$ **faça**
    enquanto $\Delta > \theta$ `// um pequeno valor positivo que determina a precisão da estimação` **faça**
      $\Delta \leftarrow 0$
      para cada $s \in \mathscr{S}$ **faça**
        $v \leftarrow V(s)$
        $V(s) \leftarrow \sum_{s',r} p(s',r|s,\pi(s))[r+\gamma V(s')]$
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
      fim
    fim
    $Politica - estavel \leftarrow verdade$
    para cada $s \in \mathscr{S}$ **faça**
      $acao - antiga \leftarrow \pi(s)$
      $\pi(s) \leftarrow_{a \in \mathscr{A}} \sum_{s',r} p(s',r|s,a)[r+\gamma V(s')]$
      se $acao - antiga \neq \pi(s)$ **então**
        $Politica - estavel \leftarrow falso$
      fim
    fim
  fim
fim

2.3.5 Iteração de valor

A avaliação da política como um passo da iteração de política pode ser truncada sem perder as garantias de convergência da iteração de política. Um caso especial desse truncamento ocorre quando a avaliação da política para após apenas uma atualização para cada estado. Esse algoritmo é chamado de *iteração de valor*, e tem como base a seguinte iteração

$$v_{k+1}(s) \doteq \max_a \sum_{s',r} p(s',r|s,a)\left[r+\gamma v_k(s')\right], \quad \forall s \in \mathscr{S}. \tag{2.14}$$

Para um v_0 arbitrário, a sequência $\{v_k\}$ converge para a função de valor ótima v_* nas mesmas condições que garante a existência de v_*.

Também pode-se notar que a atualização da iteração de valor é a mesma da atualização da avaliação de politica 2.7, com a diferença sendo que a iteração de valor requer a melhor ação seja sempre tomada. O Algoritmo 2 descreve os passos da iteração de valor.

Algoritmo 2: Algoritmo de iteração de valor para estimar $\pi \approx \pi_*$

Entrada: um $\theta > 0$ pequeno que determina a precisão da estimação

saída: A politica determinística, $\pi \approx \pi_*$ de modo que

$\pi(s) = _{a\in\mathscr{A}} \sum_{s',r} p(s',r|s,a)[r+\gamma V(s')]$

início

Inicializa V(s), para todos $s \in \mathscr{S}$, com valores arbitrários

enquanto $\Delta > \theta$ **faça**

$\Delta \leftarrow 0$

para cada $s \in \mathscr{S}$ **faça**

$v \leftarrow V(s)$

$V(s) \leftarrow arg\max_a \sum_{s',r} p(s',r|s,a)[r+\gamma V(s')]$

$\Delta \leftarrow \max(\Delta, |v - V(s)|)$

fim

fim

fim

2.4 Aprendizado por Reforço

O aprendizado por reforço, Aprendizado por Reforço (RL)-*Reinforcement learning*, é um subcampo do aprendizado de máquina cujo objetivo é aprender uma política ótima (uma

regra de decisão), dada uma função de recompensa ou custo (SUTTON; BARTO, 2018a). Está intimamente relacionado à programação dinâmica estocástica (BERTSEKAS *et al.*, 1995). Uma das desvantagens da programação dinâmica é que é necessário conhecermos a forma como ocorre a transição de estados para que se possa avaliar/melhorar as políticas e no RL não é necessário conhecer a função de transição.

Aprendizado por reforço tem dois desafios principais: a maldição da dimensionalidade e a indisponibilidade do modelo de transição de estados. A maldição da dimensionalidade se refere ao fato de que, para encontrar uma política ideal, passamos por todos os estados em $\mathscr{S}$. Isso é proibitivo computacionalmente para espaços de estado infinitos ou muito grandes, que é o caso da maioria dos problemas práticos. Isso geralmente é abordado com a aproximação da função de valor. Em vez de calcular $v(s)$ para todos os estados, usamos uma função paramétrica $v(s;\theta)$ na qual o vetor do parâmetro θ tem um tamanho muito menor que o conjunto de estados. Podemos usar como aproximadores de função de valor, por exemplo, um modelo de regressão linear ou uma rede neural. Em seguida, usamos técnicas estatísticas e de simulação para estimar os valores dos parâmetros. A indisponibilidade do modelo de transição de estado é resolvida usando simulação e valores Q.

O valor Q de uma política π, também chamada função de ação-valor, são semelhantes às funções de valor v_π, mas são definidas para o par (estado, ação):

$$q_\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t)) | s_0 = s, a_0 = a, \pi\right], \tag{2.15}$$

isto é, os valores Q em um determinado estado s e a ação a correspondem ao valor esperado da recompensa acumulada iniciando no estado s, escolhendo a ação a e seguindo a política π. Os valores Q também satisfazem a Equação de Bellman.

A principal vantagem dos valores Q é que ao encontrar os valores ótimos, nos permitem calcular prontamente a política ótima, agindo com rapidez, sem conhecer o modelo de transição de estados. A partir da Equação de Bellman (2.12), tem-se que precisamos conhecer o modelo de transição para calcular o valor esperado e, portanto, a melhor ação. Por outro lado, a Equação de Bellman para valores Q afirma que:

$$q_*(s,a) = \mathbb{E}[r(s_t,a_t) + \gamma \max_{a' \in \mathscr{A}} q_*(s_{t+1}, a') | s_t = s, a_t = a], \quad \forall s \in \mathscr{S}, a \in \mathscr{A} \tag{2.16}$$

Portanto, em um determinado estado s, a ação ótima a_* possa ser calculada simplesmente encontrando a ação que maximiza o valor Q correspondente:

$$a_* =_{a \in \mathscr{A}} q_*(s,a). \tag{2.17}$$

O uso de valores Q motivou o desenvolvimento de um popular algoritmo on-line baseado em iteração de valor conhecido como Q-learning (WATKINS, 1989), o qual é uma técnica livre de modelo que pode ser usada para encontrar a política ótima. Observe na Equação (2.16) que, no momento t com $s_t = s$ e $a_t = a$, o termo $r(s_t, a_t) + \gamma \max_{a'} q^*(s_{t+1}, a')$ é uma estimativa do valor Q ótimo $q^*(s,a)$, em que s_{t+1} é o próximo estado observado. Então, a partir de uma trajetória $s_0, a_0, r_1, s_1, a_1, r_2, s_2, ...$ Q-learning modifica em tempo de execução os valores Q aplicando uma equação de atualização (semelhante à aproximação estocástica de Robbins-Monro):

$$q(s,a) \leftarrow q(s,a) + \alpha[r(s_t,a_t) + \gamma \max_{a'} q(s_{t+1},a') - q(s,a)], \quad s_t = s, a_t = a, t = 0,1,... \tag{2.18}$$

em que $0 < \alpha < 1$ é chamado de *taxa de aprendizado*. Valores diferentes de α afetam a taxa de convergência do algoritmo, e a determinação de valores adequados depende do problema específico.

2.4.1 Aproximação da função de valor

O Q-learning pode ser subdividido em duas categorias: o Q-learning tradicional, que usa uma tabela-Q para inferir o valor Q, dado o estado atual da ação a ser tomada; o outro é Q-learning usando aproximação de função de valor. Essa aproximação pode ser feita através de uma rede neural ou de um modelo linear.

Em um modelo linear, a função de valor-estado v ou valor-ação Q é aproximada por uma combinação linear de características (chamados de *features* em inglês):

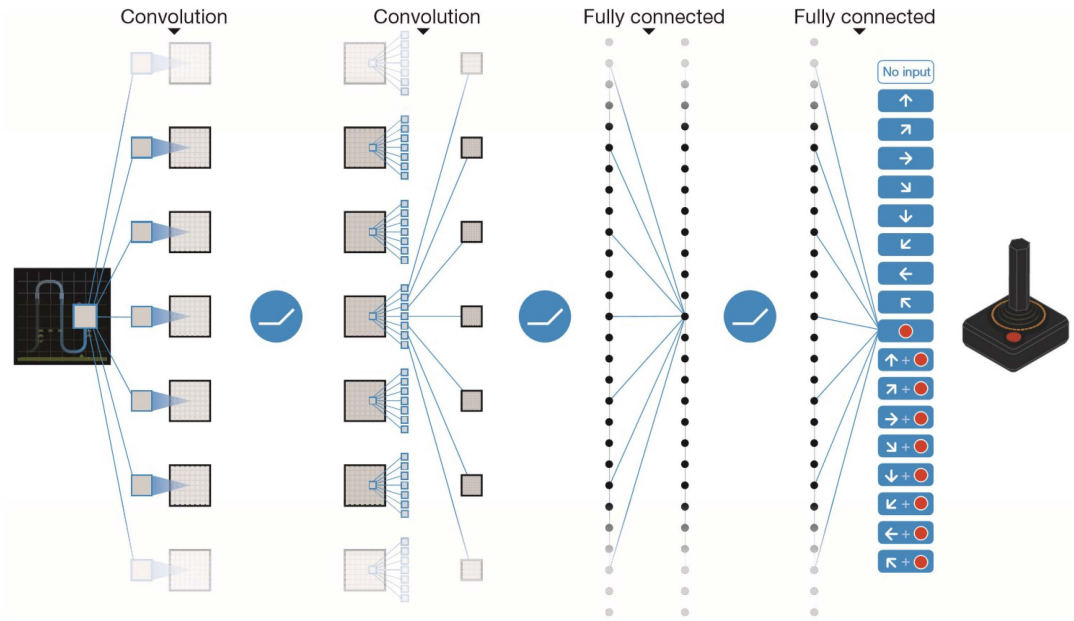
$$\bar{v}(s;\theta) = \sum_{k=1}^{K} \phi_k(s)\theta_k, \tag{2.19}$$

em que $\phi_k(.)$ são *funções-base* representando características dos estados s (as quais podem ser polinômios, funções trigonométricas, etc), e $\theta = (\theta)_{k=1}^{K}$ é um vetor de parâmetros. O modelo linear pode ser ajustado a partir de trajetórias de estados, ações e recompensas, com uso de métodos como mínimos quadrados ou gradiente descendente, de forma a aproximar a função de valor ótima ou de uma política específica.

Uma das arquiteturas aplicadas recentemente com grande sucesso para a aproximação de funções é a rede Q profunda, DQN – deep q-network, na qual uma rede neural profunda aproxima a função de valor q. Essa rede recebe o estado atual s_t e gera os valores Q para todas as ações possíveis. A rede neural profunda foi aplicada para treinar agentes para jogar os jogos

da Atari, nos quais o estado é a imagem da tela do jogo. Esta imagem é a entrada para um DQN – deep q-network, cujas saídas são os valores de Q para todas as ações possíveis, como mover para a esquerda ou direita ou pressionar o botão de disparo. Na Figura 3 pode ser visto uma representação da DQN onde a imagem da tela passa pela rede neural e a saída da rede escolhe ação a ser tomada (MNIH *et al.*, 2015).

Figura 3 – DQN Jogando Atari



Fonte: (MNIH *et al.*, 2015).

2.4.2 Aplicações recentes de aprendizado por reforço em otimização combinatória

Como em aprendizado por reforço não requer o conhecimento da função de transição de estados para encontrar a política ótima, ele pode ser aplicado em diversas áreas. A seguir, listam-se algumas aplicações recentes em otimização combinatória.

Samma *et al.* (2019) introduziu o *q-learning simulated annealing*. Os autores usam o algoritmo q-learning para aprender uma política online para definir o cronograma de temperatura. Eles aplicam a QLSA a três problemas no projeto de engenharia, que foram capazes de obter melhores soluções do que o recozimento simulado com uma programação de temperatura fixa.

Vinyals *et al.* (2015) propuseram uma nova estrutura de rede neural chamada *pointer network*. Este modelo é baseado na rede *Sequence-to-sequence* com a adição do mecanismo de atenção. Eles treinam a rede de maneira supervisionada, na qual a entrada é uma instância de um problema de otimização combinatória e a saída é a solução ótima prevista. Eles aplicam a

rede ao problema do caixeiro viajante e treinam a rede usando como rótulos soluções ideais ou subótimas de instâncias do problema do caixeiro viajante obtidas a partir de um solucionador ou algoritmo heurístico.

Khalil *et al.* (2017) usaram o aprendizado de reforço profundo para aprender uma política que tenta criar uma solução gananciosa para um problema de otimização combinatória. Os pesquisadores representam uma instância de problema por um gráfico e usam uma rede de incorporação de gráficos chamada *structure2vec* (DAI *et al.*, 2016) para extrair recursos do gráfico. Eles treinam a rede fornecendo muitas instâncias de problemas como entradas. A rede aprende uma política capaz de criar uma solução com adição sucessiva de nós com base na estrutura do gráfico. Eles mostram que suas redes treinadas foram capazes de aprender heurísticas construtivas gananciosas que são competitivas com a heurística Lin-Kerninghan no problema do caixeiro viajante.

Mnih *et al.* (2015) usaram avanços recentes no treinamento de redes neurais profundas para desenvolver um novo agente artificial, denominado *deep Q-network*, que pode aprender políticas bem-sucedidas diretamente de sensores de alta dimensão usando o *End-to-End Deep Reinforcement Learning* . Eles criam o primeiro agente artificial capaz de aprender a se destacar em diversas tarefas desafiadoras. Este agente usa uma rede convolucional profunda (LECUN *et al.*, 1998) que utiliza filtros convolucionais para explorar as correlações espaciais locais presentes nas imagens.

Kool *et al.* (2018) criaram um modelo que usa uma rede de atenção gráfica (MNIH *et al.*, 2015) para definir uma política estocástica para selecionar uma rota em problemas de roteamento de veículos, dada uma instância do problema. Essa rota é criado interativamente, um nó por vez. Quando uma rota parcial é construída, o problema restante é encontrar um caminho do último nó, através de todos os nós não visitados até o último nó.

3 METODOLOGIA

3.1 Política de decisão míope

Primeiramente, uma política de decisão míope foi proposta para servir como base para uma comparação com outros modelos – que serão desenvolvidos subsequentemente com programação dinâmica. Tal formulação foi nomeada de míope pois ela não tenta prever as demandas futuras. Para escolher quais são os melhores cortes, dado um planejamento de demandas, ela olha apenas para a demanda e o inventário atual e posteriormente calcula qual a melhor decisão para se tomar a cada período de tempo.

Seja $\mathscr{I} = \{1,2,...,I\}$ um conjunto dos possíveis itens de cada tamanho $l_1, l_2, ..., l_I$ que serão divididos a partir de um objeto de tamanho maior L. Em um dado período de tempo t, a demanda $\mathbf{d_t} = (d_{1t}, d_{2t}, ..., d_{It}) \in \mathbb{Z}_+^I$ deve ser satisfeita. Cada elemento do conjunto de padrões de cortes $\mathscr{J} = \{1,2,...,J\}$ possui uma perda de corte associada de $p_j \geq 0$. Esses padrões de cortes são associados ao vetor de itens $\mathbf{a}_j \in \mathbb{Z}_+^I, j \in \mathscr{J}$, onde cada componente representa a quantidade de vezes que o item correspondente é cortado no padrão. Vale notar que cada padrão deve satisfazer a restrição $\sum_{i=1}^{I} l_i a_{ij} \leq L$. Seja $\mathbf{x}_t = (x_{1t}, x_{2t}, ..., x_{Jt})$ onde x_{jt} representa a quantidade de objetos cortados de acordo com o padrão j.

Além das restrições básicas do problema de corte, quando o problema é multiperíodo, as sobras de itens devem ser levadas para os próximos períodos. Seja $\mathbf{r}_t = (r_{1t}, r_{2t}, \ldots, r_{It})$ o vetor de estoque no inicio do período e $\mathbf{r}_t^+ = (r_{1t}^+, r_{2t}^+, \ldots, r_{It}^+)$ o vetor de estoque após a demanda ser atendida. Manter o estoque para o próximo período tem um custo associado que é dado pelo vetor $\mathbf{h}^+ = (h_1^+, h_2^+, \ldots, h_I^+)$. Seja $\mathbf{r}_t^- = (r_{1t}^-, r_{2t}^-, \ldots, r_{It}^-)$ o vetor da demanda não atendida, que tem um vetor de custo correspondente $\mathbf{h}_t^- = (h_{1t}^-, h_{2t}^-, \ldots, h_{It}^-)$ em que esse vetor é o custo correspondente a não atender a demanda. Foi assumido que a demanda não atendida não é levada para o próximo período. Logo, o problema de corte de estoque *míope* no tempo t é dado pelo

seguinte modelo de programação linear inteira:

$$\min \quad \mathbf{p}^T\mathbf{x}_t + \mathbf{h}^+\mathbf{r}_t^+ + \mathbf{h}^-\mathbf{r}_t^- \tag{3.1}$$

s.t.

$$\mathbf{A}\mathbf{x}_t - \mathbf{r}_t^+ + \mathbf{r}_t^- = \mathbf{d}_t - \mathbf{r}_t, \tag{3.2}$$

$$1\mathbf{x}_t \leq m_1, \tag{3.3}$$

$$\mathbf{r}_t^+ \leq \mathbf{m}_2, \tag{3.4}$$

$$\mathbf{r}_t^+, \mathbf{r}_t^- \in \mathbb{Z}_+^I, \tag{3.5}$$

$$\mathbf{x}_t \in \mathbb{Z}_+^J, \tag{3.6}$$

onde $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_J)$ é a matriz cujas colunas correspondem aos padrões de corte, m_1 é o número máximo de objetos que podem ser cortados ou por conta de estoque ou por conta de alguma restrição de capacidade, e $\mathbf{m}_2$ é o vetor que guarda a capacidade máxima de cada item no inventário.

Na formulação míope, a função-objetivo (3.1) quer minimizar a soma das perdas de corte, do custo de inventário e do custo atrelado a demanda perdida. A restrição (3.2) está relacionada ao número total de itens cortados dado por $\mathbf{A}\mathbf{x}_t$, o inventário inicial $\mathbf{r}_t$, a demanda $\mathbf{d}_t$, o inventário final $\mathbf{r}_t^+$ e a demanda não atendida $\mathbf{r}_t^-$. Note que essas restrições são mais flexíveis que a formulação clássica, que é $\mathbf{A}\mathbf{x}_t \geq \mathbf{d}_t$, a qual não leva em conta o estoque inicial nem a possibilidade de não atender a demanda já que o corte ocorre depois da solicitação de demanda. A restrição (3.3) limita o numero de objetos que podem ser cortados no problema. A restrição (3.4) limita o numero de itens que podem ser guardados no estoque. Nessa aplicação, a demanda não atendida é muito penalizada para que essas soluções não ocorram e o numero de objetos que podem ser cortados e de itens que podem ser guardados são grandes o suficientes para não interferir nas soluções. As restrições (3.5) e (3.6) são restrições de integralidades para o estoque e os padrões de corte selecionados.

A solução do modelo (3.1) - (3.6) tenta atender a demanda escolhendo quantas vezes usar cada padrão de corte de forma a equilibrar as perdas de corte, o custo do inventário e o custo de não atendimento da demanda. Além disso, pode-se notar que quando a demanda for menor que o estoque inicial o termo $\mathbf{d}_t - \mathbf{r}_t$ será negativo. A solução ótima será não produzir nenhuma peça e o inventário final será $\mathbf{r}_t^+ = \mathbf{r}_t - \mathbf{d}_t$.

O Algoritmo 3 descreve a política míope, a qual reage a demanda observada a cada instante de tempo. Fazendo isso não leva em consideração o impacto da decisão ao longo do

tempo.

Algoritmo 3: Política míope

Entrada: $\mathbf{p}, \mathbf{h}^+, \mathbf{h}^-, \mathbf{A}, \mathbf{r}_0^+$

$\mathbf{r}_0 \leftarrow 0$ `// Inicialização do inventario`

início

  para $t \leftarrow 1 \ldots T$ **faça**

    $\mathbf{r}_t \leftarrow \mathbf{r}_{t-1}^+$ `// Atualiza o inventario`

    Resolve a o modelo definido pelas equações (3.1) - (3.6)

    Atualiza $\mathbf{r}_t^+ = \mathbf{A}\mathbf{x}_t + \mathbf{r}_t^- + \mathbf{r}_t - \mathbf{d}_t$ `// obitido diretamente do modelo`

  fim

fim

3.2 Formulação por processo de decisão markoviano

O problema de corte multiperíodo com demanda estocástica foi formulado como um processo de decisão markoviano composto por um conjunto de estados $\mathbf{s} \in \mathbf{S}$, um conjunto de ações $\mathbf{x} \in \mathbf{X}$, uma função de contribuição $c(\mathbf{s}, \mathbf{x})$, uma função de transição de estados $\mathbf{s}' = f(\mathbf{s}, \mathbf{x}, \mathbf{w})$, e $\mathbf{w}$ é uma fonte de incerteza. Define-se por política $\pi \in \Pi$ uma função que associa a cada estado $\mathbf{s}$ uma ação $\mathbf{x}$ a ser tomada.

Neste problema, foi definido que o estado $\mathbf{s}_t \in \mathbf{S}$ no tempo discreto t como sendo $\mathbf{s}_t = (\mathbf{r}_t, \mathbf{d}_t)$, em que $\mathbf{r}_t = (r_{1t}, r_{2t}, ..., r_{It})$ é um vetor de estoque inicial de itens no tempo t e $\mathbf{d}_t = (d_{1t}, d_{2t}, ..., d_{It})$ é o vetor de demandas. Foi assumido que a demanda é sempre atendida com uma produção sob demanda, que é uma configuração típica em problemas de corte de estoque. A decisão corresponde ao vetor $\mathbf{x}_t = (x_{1t}, x_{2t}, ..., x_{Jt}) \in \mathbb{Z}_+^J$, no qual x_{jt} representa o número de vezes que cada padrões de corte j foi aplicado nos objetos durante o tempo t, com perda de corte associada $\mathbf{p} = (p_1, p_2, ..., p_J)$, no qual p_j representa a perda associada a cada um dos padrões de corte.

Após a escolha dos cortes e a demanda sendo atendida, a quantidade de itens que sobra de cada tipo de item será guardada em um vetor de estoques finais $\mathbf{r}_t^+ = (r_{1t}^+, r_{2t}^+, \ldots, r_{It}^+)$ o qual será mantido até o próximo período $t+1$ a um custo dado por um vetor $\mathbf{h}^+ = (h_1^+, h_2^+, ..., h_I^+)$, em que h_i^+ é o custo de manter uma unidade do item i por uma unidade de tempo. Seja $\mathbf{r}_t^- = (r_{1t}^-, r_{2t}^-, \ldots, r_{It}^-)$ o vetor da demanda não atendida, que tem um vetor de custo correspondente $\mathbf{h}^- = (h_1^-, h_2^-, \ldots, h_I^-)$ em que esse vetor é o custo correspondente a não atender a demanda. A

demanda não atendida não é levada para o próximo período. A função de contribuição é dada pela soma do custo das perdas de corte e do custo de manter estoques para o próximo período:

$$c(\mathbf{s}_t, \mathbf{x}_t) = \mathbf{p}\mathbf{x}_t + \mathbf{h}^+\mathbf{r}_t^+ + \mathbf{h}^-\mathbf{r}_t^-, \tag{3.7}$$

A função de transição $\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{x}_t, \mathbf{w}_{t+1})$ é dada por $(\mathbf{r}_{t+1}, \mathbf{d}_{t+1}) = f(\mathbf{r}_t, \mathbf{d}_t, \mathbf{x}_t)$, em que o estoque inicial no tempo $t+1$ é dado por:

$$\mathbf{r}_{t+1} = \max\{\mathbf{A}\mathbf{x}_t - (\mathbf{d}_t - \mathbf{r}_t), \mathbf{0}\}, \tag{3.8}$$

onde $\mathbf{d}_t$ é um vetor de demandas aleatório e $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_J)$ é uma matriz cuja coluna $\mathbf{a}_j$ corresponde ao padrão de corte j. Note também que o estoque inicial de itens no tempo $t+1$ corresponde ao estoque final de itens no tempo t, i.e., $\mathbf{r}_{t+1} = \mathbf{r}_t^x$.

Em teoria, pode-se encontrar uma política ótima por meio da solução da Equação de Bellman:

$$v(\mathbf{s}) = \min_{\mathbf{x}_t \in \mathscr{X}_t} \{c(\mathbf{s}_t, \mathbf{x}_t) + \gamma \mathbb{E}[v(\mathbf{s}_{t+1}) | \mathbf{s}_t = \mathbf{s}, \mathbf{x}_t]\}, \quad \forall \mathbf{s} \in \mathscr{S}, \tag{3.9}$$

em que $\mathscr{X}_t$ é um conjunto de decisões viáveis definido pelas restrições:

$$\mathscr{X}_t = \left\{ \begin{array}{l} \mathbf{A}\mathbf{x}_t - \mathbf{r}_t^+ + \mathbf{r}_t^- = \mathbf{d}_t - \mathbf{r}_t, \\ \mathbf{1}\mathbf{x}_t \leq m_1, \\ \mathbf{r}_t^+ \leq \mathbf{m}_2, \\ \mathbf{r}_t^+, \mathbf{r}_t^- \in \mathbb{Z}_+^I, \\ \mathbf{x}_t \in \mathbb{Z}_+^J \end{array} \right\} \tag{3.10}$$

Para este processo de decisão markoviano, foi definida uma política determinística markoviana estacionária como um conjunto de funções $\pi := \{x(s_0), x(s_1), \dots\}$ de modo que $x : \mathscr{S} \to \mathbb{Z}_+^J$, com π em uma classe Π de políticas, que retorna um vetor de decisão $\mathbf{x} \in \mathbb{Z}_+^J$ de acordo com o estado $\mathbf{s} \in \mathscr{S}$. O índice de tempo pode ser eliminado, pois a política é estacionária e o mesmo vetor de decisão é retornado sempre que o sistema estiver no mesmo estado. O valor v_π de uma política para um estado inicial $\mathbf{s}_0 = \mathbf{s}$ é dado pelo valor esperado total descontado:

$$v_\pi(\mathbf{s}) = \lim_{T \to \infty} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t c(\mathbf{s}_t, \pi(\mathbf{s}_t)) | \mathbf{s}_0 = \mathbf{s}, \pi \right], \tag{3.11}$$

em que a esperança é tomada em relação à distribuição conjunta de todos os estados futuros $(\mathbf{s}_1, \mathbf{s}_2, ...\mathbf{s}_T)$ e $0 < \gamma < 1$ é um fator de desconto. Uma política $\pi^* \in \Pi$ é ótima se

$$v_{\pi^*}(\mathbf{s}) \leq v_\pi(\mathbf{s}), \quad \forall \mathbf{s} \in \mathscr{S}, \forall \pi \in \Pi. \tag{3.12}$$

Além disso, seja $v(\mathbf{s})$ a função de valor ótima, definida como

$$v(\mathbf{s}) \coloneqq \min_{\pi \in \Pi} v_\pi(\mathbf{s}), \quad \forall \mathbf{s} \in \mathscr{S} \tag{3.13}$$

A função de valor ótima que satisfaz a equação de Bellman (POWELL, 2011):

$$v(\mathbf{s}) = \min_{\mathbf{x} \in \mathscr{X}_\mathbf{s}} \{c(\mathbf{s}, \mathbf{x}) + \gamma \mathbb{E}[v(\mathbf{s}')|\mathbf{s}, \mathbf{x}]\}, \quad \forall \mathbf{s} \in \mathscr{S}, \tag{3.14}$$

em que o valor esperado é tomado em relação à probabilidade condicional $\mathbb{P}(\mathbf{s}'|\mathbf{s}, \mathbf{x})$ de uma transição de um estado $\mathbf{s}$ para um novo estado $\mathbf{s}'$ e $\mathscr{X}_\mathbf{s}$ é o conjunto invariante no tempo de restrições dado por (3.10), que dependem do estado $\mathbf{s}$.

Se a função de valor ótimo for conhecida, resolvendo a equação de Bellman (3.14), uma política ótima é obtida agindo gulosamente em relação ao lado direito da equação (3.14). A resolução da equação de Bellman pode ser alcançada por meio de métodos de iteração de valor e política, descritos nas Seções 2.3.4 e 2.3.5. No entanto, ambos os métodos envolvem a iteração de todos os estados e todas as decisões, o que é computacionalmente inviável para grandes espaços de estados. Essa inviabilidade é conhecida como a *maldição da dimensionalidade*. Na próxima subseção, desenvolve-se uma abordagem com uso de aprendizado por reforço para obter uma política para o problema de corte de estoque com demandas estocásticas.

3.3 Solução por meio de aprendizado por reforço

O aprendizado por reforço, também conhecido por *programação dinâmica aproximada*, fornece um conjunto de estratégias para superar as maldições da dimensionalidade (POWELL, 2011; SUTTON; BARTO, 2018a; BERTSEKAS *et al.*, 1996). Em aprendizado por reforço, o Processo de Decisão Markoviano (MDP) não é resolvido exatamente varrendo todos os estados. Em vez disso, apenas um subconjunto dos estados possíveis é visitado através da simulação de uma trajetória que avança no tempo. Além disso, uma abordagem comum é usar uma aproximação de função de valor por meio de um modelo estatístico ou de aprendizado de máquina. Desta forma, a função de valor referente a um determinado estado pode ser estimada a partir do modelo.

3.3.1 Política baseada na aproximação da função de valor

Além do número esmagador de estados, um obstáculo notável na tentativa de resolver a formulação por processo de decisão markoviano do problema de corte de estoque com demandas

estocásticas é calcular a valor esperado na equação de Bellman (3.14). Adicionalmente, trabalha-se com o conceito de *estado pós-decisão* (ou simplesmente pós-estado), que corresponde ao estado após decidir sobre o valor de $\mathbf{x}_t$. Nesse caso, foi definido como o pós-estado, denotado por $\mathbf{s}_t^x \in \mathscr{S}^x$, o vetor de estoque final, ou seja, $\mathbf{s}_t^x = \mathbf{r}_t^+$. A função de valor pós-estado ótima $v(\mathbf{s}_t^x)$ satisfaz a seguinte forma da equação de Bellman (POWELL, 2011):

$$v(\mathbf{s}^x) = \mathbb{E}\Big[\min_{\mathbf{x}_t \in \mathscr{X}_t} \{c(\mathbf{s}_t, \mathbf{x}_t) + \gamma v(\mathbf{s}_t^x)\} | \mathbf{s}_{t-1}^x = \mathbf{s}^x\Big], \tag{3.15}$$

em que $\mathbf{s}_t = (\mathbf{r}_t, \mathbf{d}_t)$ é o estado de pré-decisão, e a valor esperado é calculado em relação às probabilidades de transição do pós-estado atual para o próximo estado de pré-decisão. Existem duas vantagens principais em usar o estado pós-decisão: a primeira refere-se ao fato de que, ao contrário de (3.14) , agora a valor esperado está fora do problema de minimização, o que permite resolvê-lo como um problema determinístico; a segunda se refere ao fato de que o pós-estado $\mathbf{s}_t^x = \mathbf{r}_t^+$ tem dimensionalidade menor que $\mathbf{s}_t = (\mathbf{r}_t, \mathbf{d}_t)$.

Portanto, se a função de valor pós-estado $v(\mathbf{s}^x)$ para todos os $\mathbf{s}^x \in \mathscr{S}^x$ for conhecida, uma política ótima π^* pode ser obtida escolhendo $\mathbf{x}_t^*$ em cada $t = 0, 1, \ldots$ de modo que

$$\mathbf{x}_t^* = \arg\min_{\mathbf{x}_t \in \mathscr{X}_t} \{c(\mathbf{s}_t, \mathbf{x}_t) + \gamma v(\mathbf{s}_t^x)\}. \tag{3.16}$$

No entanto, o grande tamanho do espaço de estados impede de calcular a função de valor de todos os pós-estados. Recorre-se então a um modelo linear para aproximar a função de valor

$$\bar{v}(\mathbf{s}^x; \theta) = \phi(\mathbf{s}^x)\theta, \tag{3.17}$$

em que $\phi(\mathbf{s}^x) = (\phi_1(\mathbf{s}^x), \phi_2(\mathbf{s}^x), ..., \phi_K(\mathbf{s}^x))$ é um vetor linha cujos componentes $\phi_k(.), k = 1, 2, ..., K$ são funções-base, e $\theta = (\theta_1, \theta_2, ..., \theta_K)$ é um vetor coluna de parâmetros. O uso de um modelo linear é conveniente, já que (3.16) será um problema de programação inteira, que de (3.7) e (3.17) e note que $\mathbf{s_t}^x = \mathbf{r_t}^+$, é dado por

$$\hat{\mathbf{x}}_t = \arg\min_{\mathbf{x}_t \in \mathscr{X}_t} \{\mathbf{p}\mathbf{x}_t + \mathbf{h}^+\mathbf{r}_t^+ + \mathbf{h}^-\mathbf{r}_t^- + \gamma\phi(\mathbf{r_t}^+)\theta\}, \tag{3.18}$$

e $\mathscr{X}_t$ é um conjunto de restrições dado por (3.10) . Observe que, para um determinado vetor de parâmetro θ, $\hat{\mathbf{x}}_t$ é apenas uma aproximação da decisão ótima, de modo que a política resultante não será ótima. Na próxima seção, propõe-se um algoritmo de iteração de valor aproximado para aprender os valores de θ em uma busca por boas políticas.

3.3.2 Aprendizado de política online

Em princípio, pode-se formular o problema de aprendizado da política como encontrar o vetor ótimo θ^* que minimiza o valor esperado do erro quadrático entre a função de valor exato $v(\mathbf{s}^x)$ e a função de valor aproximada $\bar{v}(\mathbf{s}^x;\theta^*)$:

$$\theta^* = \arg\min_{\theta\in\Theta} \mathbb{E}[(\mathbf{v}-\bar{\mathbf{v}}(\theta))^2], \tag{3.19}$$

em que $\mathbf{v} = v(\mathbf{s}^x)_{\mathbf{s}^x\in\mathscr{S}^x}$ é um vetor cujos componentes são os valores de estar em pós-estados $\mathbf{s}^x$ e $\bar{\mathbf{v}}(\theta) = \bar{v}(\mathbf{s}^x;\theta)_{\mathbf{s}^x\in\mathscr{S}^x}$ é um vetor cujos componentes são os valores aproximados de estar nos pós-estados dado θ, e o valor esperado é calculado em relação à distribuição de probabilidade estacionária dos estados dada a política ótima. Infelizmente, (3.19) não pode ser aproximado diretamente, uma vez que não se sabe o valor exato da função vetorial $\mathbf{v}$. Por isso, a equação (3.19) deve ser resolvida de forma indireta.

Uma abordagem popular para resolver (3.19) é usar um algoritmo de gradiente estocástico (SCHULMAN *et al.*, 2017). No entanto, definir uma regra de tamanho de passo que permite a convergência do algoritmo requer muito esforço experimental (GEORGE; POWELL, 2006). Em contraste, foi usada a técnica de filtragem bayesiana (também chamada de *aprendizado bayesiano*), da qual o filtro de Kalman é um caso particular, para ajustar os parâmetros θ, com a vantagem de não precisar ajustar uma regra de tamanho de passo. O esquema geral assemelha-se a uma iteração de valor aproximada, em que os valores estimados atuais de θ são usados na próxima iteração para aproximar a função de valor. Esta prática é chamada de *bootstraping* na literatura de aprendizagem por reforço.

Seja $\mathbf{s}_0, \hat{\mathbf{x}}_0, \mathbf{s}_0^x, \hat{v}_0, \mathbf{s}_1, \hat{\mathbf{x}}_1, \mathbf{s}_1^x, \hat{v}_1, \ldots \mathbf{s}_t, \hat{\mathbf{x}}_t, \mathbf{s}_t, \hat{v}_t, \ldots$ uma trajetória de estados, decisões, pós-estados e valores, em que as decisões $\hat{\mathbf{x}}_t$ e os valores $\hat{v}_t$ são os vetor de solução correspondente e valor da função objetivo dado pela solução de (3.18) . Observe que $\hat{v}_t$ é uma estimativa do valor do pós-estado $\mathbf{s}_{t-1}^x$ na etapa da iteração anterior. Em seguida, foi definido o seguinte modelo linear dinâmico (DLM) para aprender online os parâmetros θ:

$$\theta_t = \mathbf{G}_t\theta_{t-1} + \omega_t, \tag{3.20}$$

$$\hat{v}_t = \phi_t\theta_t + \nu_t, \tag{3.21}$$

em que $\mathbf{G}_t$ é uma matriz de evolução, $\phi_t = \phi(\mathbf{s}_{t-1}^x)$ é um vetor de características (*features* ou funções-base), $\omega_t \sim \mathscr{N}(\mathbf{0}, \mathbf{W}_t)$ é uma evolução temporal dos parâmetros com matriz de covariância $\mathbf{W}_t$, $\nu_t \sim \mathscr{N}(0, \sigma_t^2)$ é um erro observacional e $\mathscr{N}$ denota a distribuição normal

multivariada. O aprendizado bayesiano foi utilizado para atualizar o valor de θ a cada nova observação $\hat{v}_t$ da função de valor.

Dado o histórico de valores $D_{t-1} = \{\hat{v}_{t-1}, ...\hat{v}_0\}$, foi definido como probabilidade a priori $p(\theta_t|D_{t-1})$ e a posteriori $p(\theta_t|D_t)$ depois de observar $\hat{v}_t$, e $D_t = \{\hat{v}_t\} \cup D_{t-1}$. Como o principal interesse é no caso de demanda estacionária, foi assumido que os parâmetros θ não mudam com o tempo, de modo que $\theta_t = \theta$ para todos os t, que implica $\mathbf{G}_t$ é igual à matriz de identidade e $\omega_t = 0$. Também foi assumido que a variação de observação é constante ao longo do tempo, ou seja, $\sigma_t^2 = \sigma^2$ para todos os t.

Seja $\theta_0 \sim \mathcal{N}(\mathbf{m}_0, \mathbf{C}_0)$ uma distribuição a priori no tempo $t = 0$, especificada pelo tomador de decisão , em que $\mathbf{m}_0$ é um vetor médio e $\mathbf{C}_0$ uma matriz de covariância. Então, por conta do Teorema de Bayes, no tempo t, pode-se mostrar que a distribuição de probabilidades a posteriori é normal multivariada dada por $p(\theta_t|D_t) = \mathcal{N}(\mathbf{m}_t, \mathbf{C}_t)$, em que desenvolve-se abaixo como obter $\mathbf{m}_t$ e $\mathbf{C}_t$.

Além disso, observe que σ^2 é um parâmetro desconhecido. Também se utiliza o aprendizado bayesiano para ajustar seu valor online. Seja $p(1/\sigma^2|D_{t-1})$ a distribuição a priori no tempo t e $p(1/\sigma^2|D_t)$ a distribuição posteriori após observar $\hat{v}_t$. Se for assumida uma distribuição de probabilidades gama a priori em $1/\sigma^2$ com parâmetros $a_{t-1}/2$ e $b_{t-1}/2$, a distribuição a posteriori também será gama com parâmetros $a_t/2$ e $b_t/2$ dados por (WEST; HARRISON, 1999):

$$a_t = a_{t-1} + 1, \tag{3.22}$$

$$b_t = b_{t-1} + \hat{\sigma}_{t-1}^2(\hat{v}_t - \phi_t \mathbf{m}_{t-1})^2/q_t, \tag{3.23}$$

e $\hat{\sigma}_{t-1}^2 = b_{t-1}/a_{t-1}$ é uma estimativa de σ^2 no tempo $t-1$. Observe que a_{t-1}/b_{t-1} é a média da gama a priori no tempo $t-1$. Desta forma, $\hat{\sigma}_t^2 = b_t/a_t$ será usado como uma estimativa de σ^2 no tempo t.

Em cada etapa de tempo t, $\mathbf{m}_t$ e $\mathbf{C}_t$ é atualizado online de acordo com as seguintes equações recursivas:

$$\mathbf{m}_t = \mathbf{m}_{t-1} + \mathbf{B}_t(\hat{v}_t - \phi_t \mathbf{m}_{t-1}), \tag{3.24}$$

$$\mathbf{C}_t = \frac{\hat{\sigma}_t^2}{\hat{\sigma}_{t-1}^2}(\mathbf{C}_{t-1} - \mathbf{B}_t \mathbf{B}_t' q_t), \tag{3.25}$$

$$\mathbf{B}_t = \mathbf{C}_{t-1}\phi_t q_t^{-1}, \tag{3.26}$$

$$q_t = \phi_t' \mathbf{C}_{t-1}\phi_t + \hat{\sigma}_t^2. \tag{3.27}$$

Então $\mathbf{m}_t$ é uma estimativa de θ no tempo t. Finalmente, pode-se juntar todos os elementos e declarar o algoritmo proposto 4 para aproximar uma política ótima de forma online.

Algoritmo 4: Algoritmo de aprendizado de política

Entrada: $\mathbf{p}, \mathbf{h}^+, \mathbf{h}^-, \mathbf{A}, \mathbf{r}_0^+, \gamma, p(.), \phi(.), \mathbf{m}_0, \mathbf{C}_0, a_0, b_0$

saída: $\hat{\theta}$

$\theta_0 \leftarrow \mathbf{m}_0$, $\hat{\sigma}_0^2 \leftarrow b_0/a_0$

início

  para $t \leftarrow 1 \ldots T$ **faça**

    $\mathbf{r}_t \leftarrow \mathbf{r}_{t-1}^+$ `// Estoque inicial`

    Amostrar a demanda $\mathbf{d}_t \sim p(\mathbf{d}_t)$ `// Por simulação de Monte Carlo`

    $\mathbf{s}_t \leftarrow (\mathbf{r}_t, \mathbf{d}_t)$ `// Estado antes da ação no tempo` t

    $\hat{\mathbf{x}}_t \leftarrow \arg\min_{\mathbf{x}_t \in \mathscr{X}_t} \{c(\mathbf{s}_t, \mathbf{x}_t) + \gamma \bar{v}(\mathbf{s}_t^x; \theta_{t-1})\}$ `// Decisão de quanto cortar`

    $\hat{v}_t \leftarrow \min_{\mathbf{x}_t \in \mathscr{X}_t} \{c(\mathbf{s}_t, \mathbf{x}_t) + \gamma \bar{v}(\mathbf{s}_t^x; \theta_{t-1})\}$

    $\phi_t \leftarrow \phi_t(\mathbf{s}_{t-1}^x)$ `// Note que` $\mathbf{s}_{t-1}^x = \mathbf{r}_{t-1}^+$

    Atualizar $\mathbf{m}_t, \mathbf{C}_t, a_t, b_t, \hat{\sigma}_t^2$ `// equações (3.22) - (3.27)`

    $\theta_t \leftarrow \mathbf{m}_t$

  fim

  $\hat{\theta} \leftarrow \theta_T$

fim

3.3.3 Uma aproximação da função de valor com características lineares

Note que, para aplicar o Algoritmo 4, é necessário definir as características (*features* ou funções-base) da aproximação da função de valor. Nesta seção, definem-se funções-base $\phi(.)$ como funções lineares do estado. Neste caso, para $i = 1, 2, \ldots, I$, definiram-se características $\phi_i(\mathbf{s}^x) = r_{it}^+$, as quais correspondem ao estoque final para o item i no momento t. Isso resulta em uma aproximação linear da função de valor, dada por

$$\bar{v}(\mathbf{s}^x; \theta) = \sum_{i=1}^{I} r_{it}^+ \theta_i^+.$$

No entanto, foi notado que o uso de tal modelo estimaria o valor zero sempre que o estoque final fosse zero. Portanto, definiram-se características binárias adicionais que capturam se o estoque é

maior ou igual a zero. Isto é, para $i = 1, 2, \ldots, I$, foi definido

$$\begin{aligned}\phi_{i+I}(\mathbf{s}^x) &= 1 \quad \text{if} \quad r_{it}^+ > 1, \\ &= 0 \quad \text{caso contrário.}\end{aligned}$$

Para implementar as características binárias no modelo de otimização, definiram-se variáveis binárias auxiliares $z_{it} = 0$ se $r_{it}^+ > 1$ e $z_{it} = 1$ caso contrário. Dessa forma, a aproximação da função de valor será

$$\bar{v}(\mathbf{s}^x;\theta) = \sum_{i=1}^{I} r_{it}^+ \theta_i^+ + \sum_{i=1}^{I} (1 - z_{it})\theta_i^0, \tag{3.28}$$

$$= \mathbf{r}_t^+ \theta^+ + (\mathbf{1} - \mathbf{z}_t)\theta^0, \tag{3.29}$$

em que $\theta = (\theta^+, \theta^0)$. Portanto, a decisão em um momento t, dada pela equação (3.18), tomará a forma do seguinte modelo de programação linear inteira:

$$\min \quad \mathbf{p}\mathbf{x}_t + \mathbf{h}^+\mathbf{r}_t^+ + \mathbf{h}^-\mathbf{r}_t^- + \mathbf{r}_t^+\theta^+ + (\mathbf{1} - \mathbf{z}_t)\theta^0 \tag{3.30}$$

s.a.

$$\mathbf{A}\mathbf{x}_t - \mathbf{r}_t^+ + \mathbf{r}_t^- = \mathbf{d}_t - \mathbf{r}_t, \tag{3.31}$$

$$\mathbf{1}\mathbf{x}_t \leq m_1, \tag{3.32}$$

$$\mathbf{r}_t^+ \leq \mathbf{m}_2 \odot \mathbf{z}_t, \tag{3.33}$$

$$\mathbf{z}_t \leq \mathbf{r}_t^+, \tag{3.34}$$

$$\mathbf{r}_t^+, \mathbf{r}_t^- \in \mathbb{Z}_+^I, \tag{3.35}$$

$$\mathbf{x}_t \in \mathbb{Z}_+^J, \tag{3.36}$$

$$\mathbf{z}_t \in \{0,1\}^I, \tag{3.37}$$

em que $\odot$ denota o produto de Hadamard. Observe a adição das restrições (3.33) e (3.34), que são necessárias para garantir a condição $z_{it} = 0$ se $r_{it}^+ > 1$ e $z_{it} = 1$ caso contrário, para todos os $i = 1, 2, \ldots, I$.

4 RESULTADOS

Nesta seção, apresentam-se os resultados de testes computacionais que comparam a política obtida por aprendizado por reforço, dada pelo Algoritmo 4, com uma política míope dada pelo Algoritmo 3. Foram realizados 5 experimentos, rotulados de A a E. O experimento A é o experimento-base, e os outros são variantes em que se alteram um ou mais fatores.

O ambiente para a realização dos teste foi em um computador AMD Ryzen 7 2700X, CPU 3.7 GHz com 32 Giga bytes de Ram executando em Windows 10 versão 1809/64 bits. O programa usado para resolver os modelos é o IBM ILOG® CPLEX® Optimization Studio versão 12.09.00. Para a construção dos modelos foi utilizada a linguagem *Python*, versão 3.7.0, com as bibliotecas: NumPy 1.18.1, Matplotlib 3.1.3, Pandas 1.0.3 e Pyomo 5.6.9.

Os dados referentes aos experimentos são de uma empresa do ramo de construção civil. Para atender à demanda ao longo do tempo, barras de aço em estoque devem ser cortadas de acordo com os tamanhos e quantidades desejados pelos clientes. O tamanho das barras de aço em estoque que são cortadas é de 15 metros, e os tamanhos de itens demandados são 1.15m, 1.80m, 2.67m, 3.14m, 8.80m, 11.80m e 12.00m.

Inicialmente, os padrões de corte devem ser gerados. No Quadro 2 do Apêndice A encontram-se todos os dados referentes aos padrões de corte ótimos utilizados e as suas respectivas perdas. Eles foram gerados levantando cada possível combinação que respeita a restrição de tamanho do objeto a ser cortado e depois comparando essas combinações e removendo aquelas que não geram uma quantidade maior de itens de algum tipo do que outra combinação a função utilizada pode ser vista no Apêndice B . Os padrões ótimos são todos aqueles que não estão contidos em nenhum outro padrão ótimo e as suas sobras são menores que a menor das barras desejadas.

Assume-se que a demanda por barras $\mathbf{d}_t$ em um dado tempo t segue uma distribuição de probabilidades discreta $p(\mathbf{d}_t) = p(\mathbf{d}_t|N,\mathbf{q})p(N)$ em que $p(\mathbf{d}_t|N,\mathbf{q})$ é uma distribuição multinomial com parâmetros N e $\mathbf{q} = (q_1, q_2, ..., q_7)$, em que N é o total de itens demandados, q_i é a probabilidade de um item ter comprimento $l_i, i = 1, 2, \ldots, 7$, e $p(N)$ é uma distribuição uniforme discreta com parâmetros $N_{\min}$ e $N_{\max}$, que representam respectivamente a demanda total mínima e máxima por barras.

A medida de desempenho utilizada foi o custo médio rolante que foi computado de forma iterativa como sendo a soma de todo o custo até aquele momento dividido pelo tempo.

4.1 Resultados do Experimento A

Nesse experimento realizado foram utilizados o parâmetro $\mathbf{q} = (0.3, 0.1, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1)$ e as demandas $N_{\min} = 10$, $N_{\max} = 30$. Adotou-se o custo de estoque $\mathbf{h}^+$ como sendo proporcional a uma fração de 10% do comprimento de cada item, o custo de demanda não atendida $\mathbf{h}^-$ como sendo 150 vezes o comprimento de cada item e o fator de desconto de $\gamma = 0.95$. Foi usada uma aproximação linear da função de valor conforme a Equação (3.29) Adotou-se $\mathbf{m}_0 = 100; \mathbf{C}_0 = 1000; a_0 = 0,01; b_0 = 0,01$, que correspondem a distribuições de probabilidade a priori não informativas. Foi então aplicado o algoritmo 4 por um total de $T = 100000$ iterações.

A política treinada pelo Algoritmo 4 foi comparada à política míope (Algoritmo 3), a qual é equivalente a estabelecer $\theta = \mathbf{0}$. Ou seja, a política míope não leva em conta o valor do pós-estado ao resolver o modelo de programação linear inteira (3.30), considerando somente o custo associado à perda de corte e à manutenção de estoque para o próximo período. A medida de desempenho adotada foi $\bar{c}_t$, a qual corresponde ao custo médio até o tempo t. As duas políticas foram aplicadas a uma sequência de demandas de teste para um período de $T = 200$.

A Figura 4 exibe o custo médio e instantâneo das politicas testadas nesse experimento. É possível observar que o desempenho da política proposta é superior ao da política míope. A

Figura 4 – Custo instantâneo e médio da política míope e proposta durante o experimento A

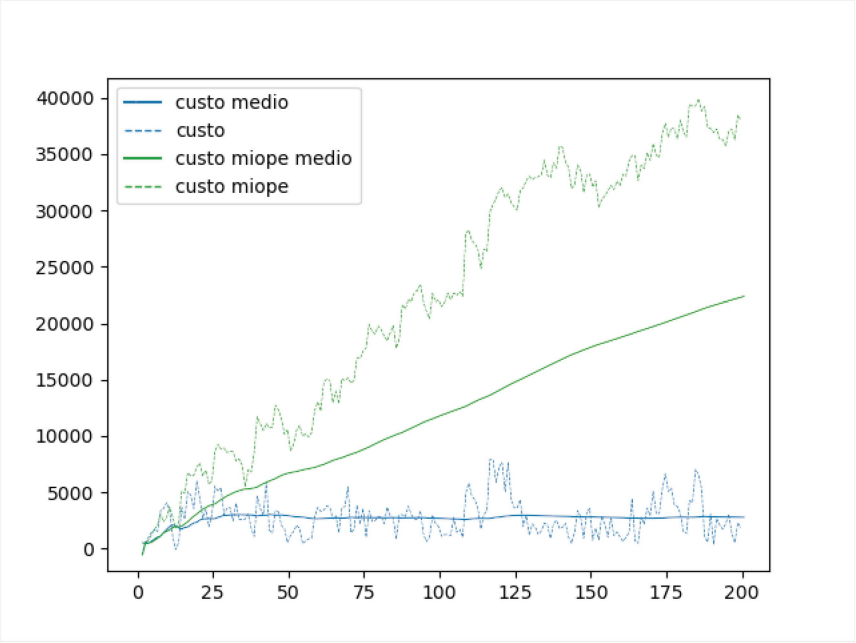


Figura 5 apresenta o estoque utilizando a política proposta e a Figura 6 apresenta o estoque

utilizando a política míope, cujo custo médio é crescente ao longo do tempo. Isso ocorre devido ao acúmulo de estoque de alguns tipos de peça no caso da política míope, a qual parece ser incapaz de controlar adequadamente os estoques.

Figura 5 – Estoque segundo a política proposta durante o experimento A

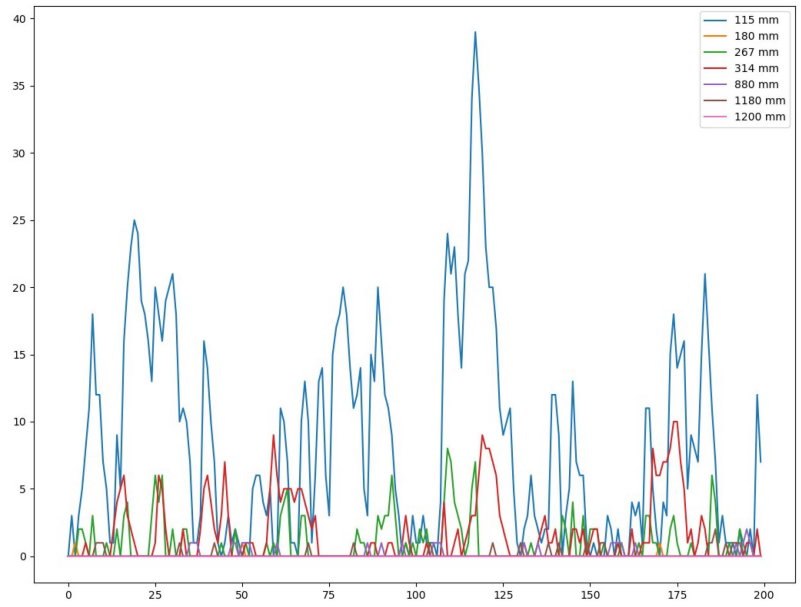
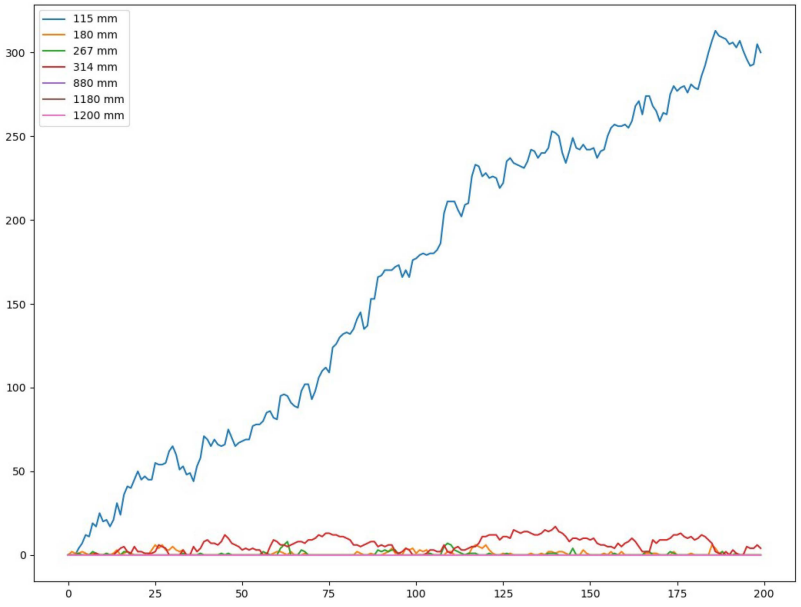


Figura 6 – Estoque segundo a política míope durante o experimento A

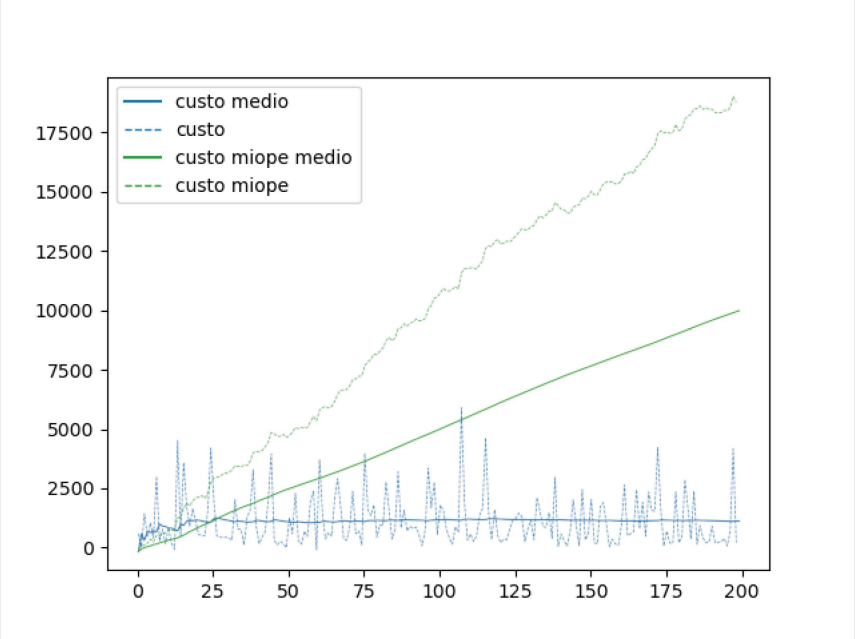


4.2 Resultados do Experimento B

No experimento B utilizou-se mesma estrutura do experimento A, mas alterando a chance de selecionar cada uma dos itens demandados para 14%. Essa mudança foi feita para visualizar o comportamento do estoque em uma distribuição de barras uniforme.

A Figura 7 exibe o custo médio e instantâneo das politicas testadas nesse experimento. É possível observar que o desempenho da política proposta continua superior ao da política míope. A Figura 8 apresenta o estoque utilizando a política proposta e a Figura 9 apresenta o estoque

Figura 7 – Custo instantâneo e médio da política míope e proposta durante o experimento B



utilizando a política míope. Ao modificar a demanda distribuindo igualmente a quantidade que cada item foi requisitado fez com que a procura do item mais curto caísse bastante como a política míope não olha as consequências que sobrar estoque tem a longo prazo e só está preocupada com as decisões a curto prazo, ocorre muito rapidamente um acumulo de itens.

4.3 Resultados dos Experimentos C e D

Para tentar reduzir o acúmulo de estoque a longo prazo foram gerados mais 15 padrões de corte subótimos. Esses padrões extras foram gerados selecionando todos os possíveis subconjuntos de itens e a partir deles gerando os padrões de corte ótimos, dessa maneira foram gerados 104 padrões, sendo que 89 deles são os padrões ótimos do conjunto completo de itens.

Figura 8 – Estoque segundo a política propostas durante o experimento B

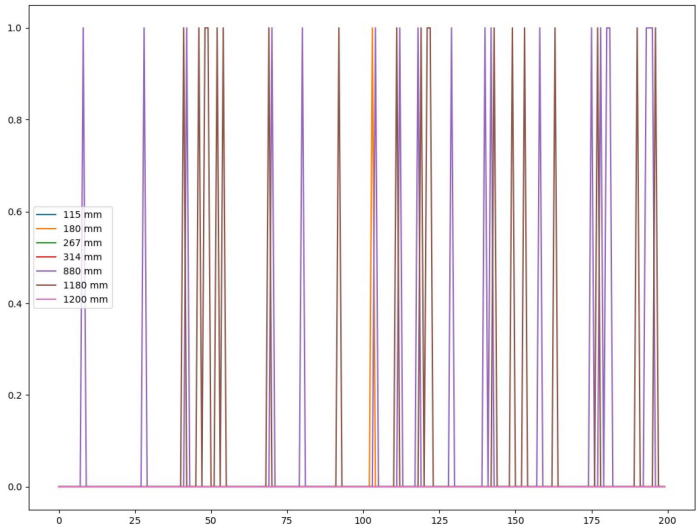
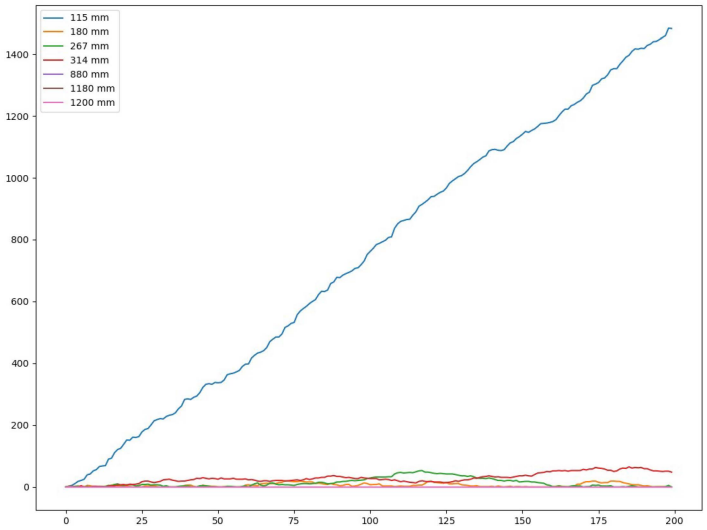


Figura 9 – Estoque segundo a política míope durante o experimento B



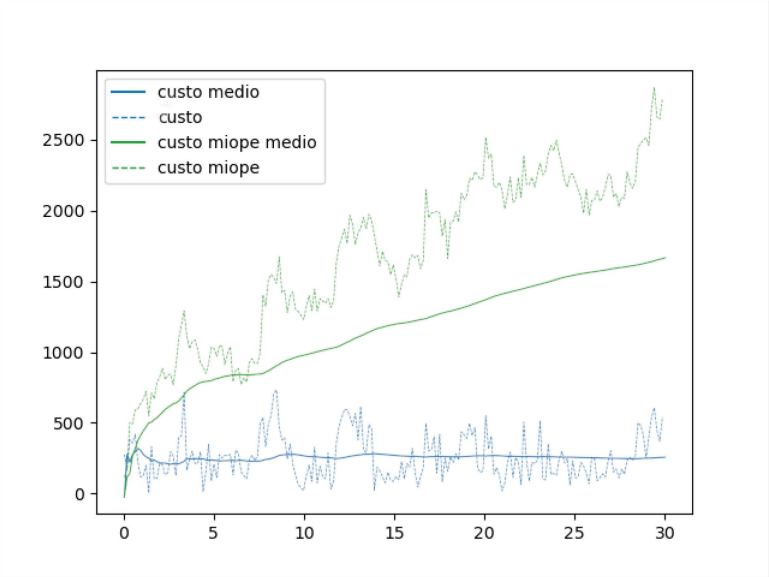
Esses 15 novos padrões podem ser vistos no Quadro 3 do Apêndice A. Vale notar que para cada padrão de corte do Quadro 3 há um padrão de corte no quadro 2 que é mais eficiente.

4.3.1 Experimento C

No experimento C foram utilizados os mesmos parâmetros do experimento A, mas modificando os padrões de corte disponíveis. A Figura 10 exibe o custo médio e instantâneo das politicas testadas nesse experimento. É possível observar que o desempenho da política proposta

continua superior ao da política míope. A Figura 11 apresenta o estoque utilizando a política

Figura 10 – Custo instantâneo e médio da política míope e proposta durante o experimento C



proposta e a Figura 12 apresenta o estoque utilizando a política míope. Ao modificar os padrões de corte disponíveis para serem selecionados a política míope continuou se comportando de forma pior que a política proposta por conta do acúmulo de itens ao longo do tempo. Mas em comparação ao experimento A ao adicionar os cortes subótimos o acúmulo do item de 115 mm foi menor pois era possível selecionar mais padrões que não contêm esse item.

4.3.2 Experimento D

No experimento D foram utilizado os mesmos parâmetros do experimento B, porém modificando os padrões de corte disponíveis. A Figura 13 exibe o custo médio e instantâneo das politicas testadas nesse experimento. É possível observar que o desempenho da política proposta continua superior ao da política míope. A Figura 14 apresenta o estoque utilizando a política proposta e a Figura 15 apresenta o estoque utilizando a política míope. Com essa distribuição das demandas a política míope não se comporta bem e em comparação com o experimento B a adição dos cortes não trazem um ganho significativo.

4.4 Resultado do Experimento E

Para o experimento E a realização de 50 simulações foram necessárias. O Quadro 1 apresenta os seguintes parâmetros: número de itens, demanda miníma, demanda máxima,

Figura 11 – Estoque segundo a política proposta durante o experimento C

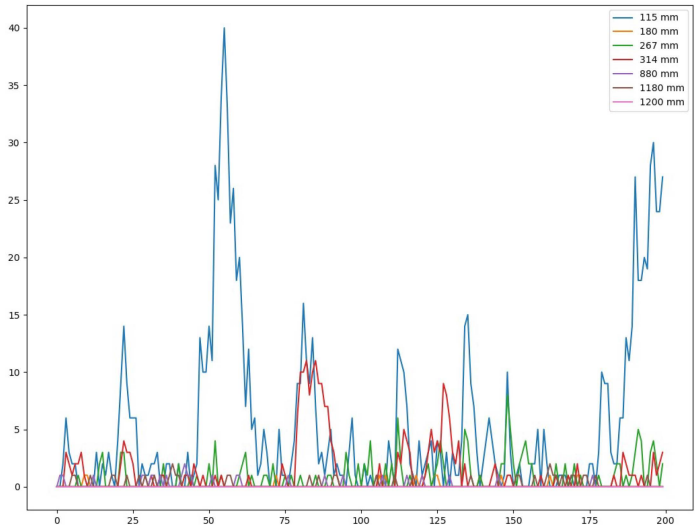
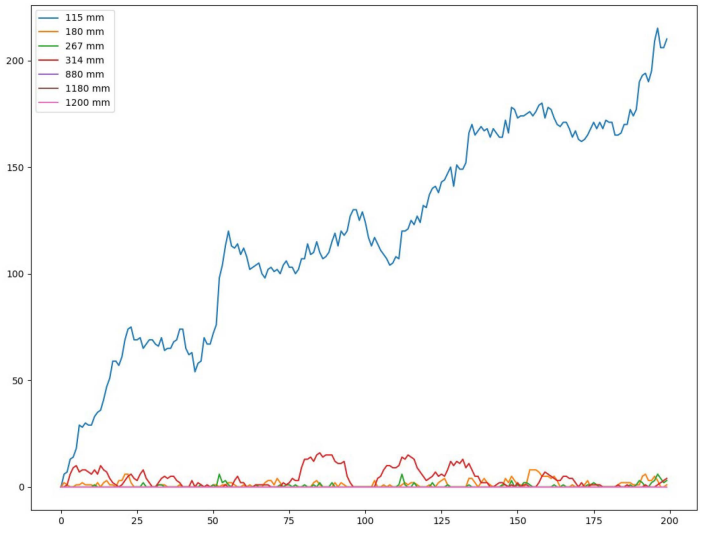


Figura 12 – Estoque segundo a política míope durante o experimento C

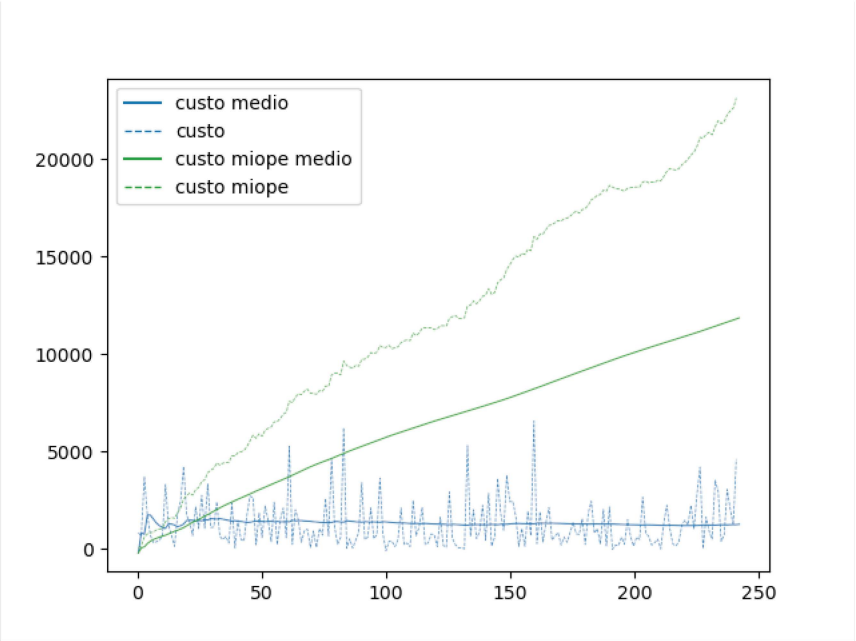


probabilidade de cada item, número de padrões de corte.

Os tamanhos de corte de cada item foram selecionados de forma aleatória. A cada 10 experimentos foram adicionados novos padrões de corte. Os mesmos itens selecionados para os experimentos 1-10 também foram selecionados para os demais experimentos, os itens selecionados para os experimentos 11-20 também foram selecionados para os experimentos 21-50, assim suscetivamente.

Os itens selecionados e os padrões de cortes utilizados podem ser visto no Apêndice

Figura 13 – Custo instantâneo e médio da política míope e proposta durante o experimento C



A nos Quadros 4,5,6,7 e 8. A penalidade por manter um item em estoque é de 10% do tamanho do item, a penalidade por demanda não atendida foi 10 vezes o tamanho do item, a atenuação foi de $\gamma = 0,995$. Na aproximação linear da função de valor foram definidas funções base do tipo $\phi_i(\mathbf{r}^x) = r_i^x$, de forma que $\bar{v}(\mathbf{r}^x;\theta) = \sum_{i=1}^{7} r_i^x \theta_i$, i.e., a função de valor para um dado pós estado $\mathbf{r}^x$ é aproximada como uma combinação linear dos valores dos estoques finais r_i^x. Adotou-se $\mathbf{m}_0 = 100; \mathbf{C}_0 = 1000; a_0 = 0,01; b_0 = 0,01$.

Quadro 1 – Experimento E.

Simulação	1-5	6-10	11-15	16-20	21-25	26-30	31-35	36-40	41-45	46-50
Número de itens	3	3	6	6	9	9	12	12	15	15
Demanda Miníma	3	30	6	60	9	90	12	120	15	150
Demanda Máxima	9	90	18	180	81	810	36	360	45	450
Probabilidade de cada item	33,3%	33,3%	16,6%	16,6%	11,1%	11,1%	8,3%	8,3%	6,6%	6,6%
Número de padrões de corte	11	11	28	28	113	113	220	220	230	230
Ganho da política proposta	1,7%	4,9%	52,3%	98,0%	14,2%	97,5%	91,4%	67,8%	70,8%	84,1%

No Quadro 1 também está o ganho percentual da politica proposta sobre a politica míope, em media a politica proposta foi superior a míope, mas teve alguns casos pontuais em que

Figura 14 – Estoque segundo a política proposta durante o experimento C

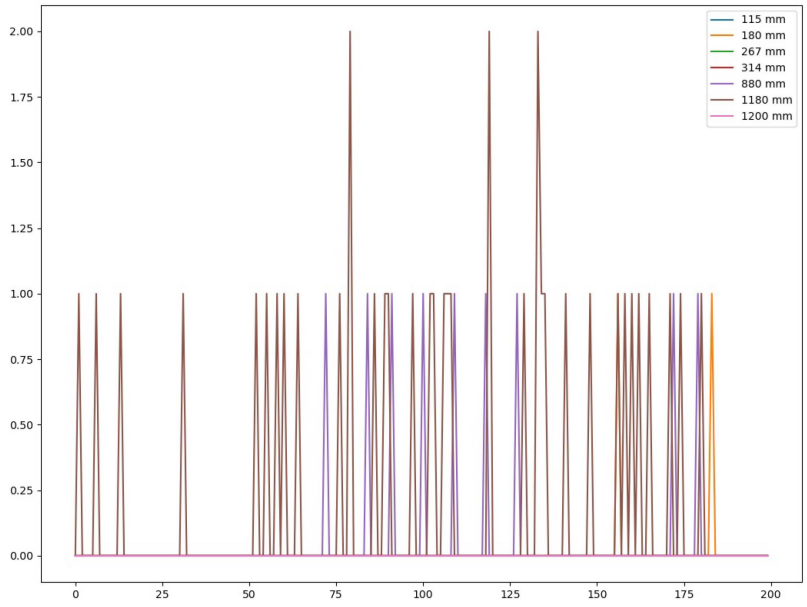
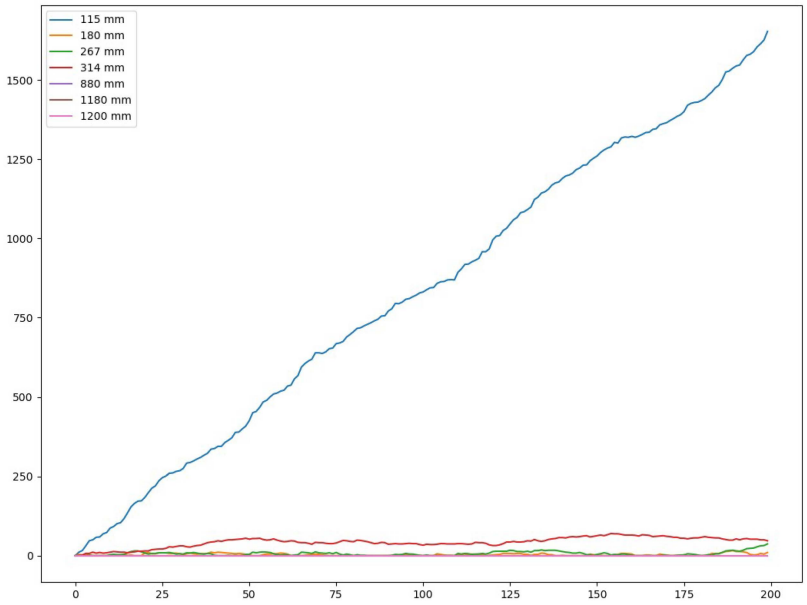


Figura 15 – Estoque segundo a política míope durante o experimento C



a demanda solicitada no período simulado divergiu da demanda esperada pelo modelo proposta fazendo com que a politica míope se saísse melhor. O resultado de todas as simulações podem ser vistos no Quadro 9 no Apêndice A.

5 CONCLUSÕES

Nessa dissertação abordou-se o problema de corte de estoque cuja a demanda por itens ocorre de forma estocástica ao longo do tempo, algo que acontece tipicamente em ambientes de produção. O problema de corte de estoques geralmente é tratado na literatura para uma única instância de tempo, bem diferente do que ocorre na prática onde o que sobra de estoque em um período é carregado para o próximo. A forma que a literatura trata esses problemas apresenta uma disparidade muito grande em relação as políticas propostas nessa dissertação uma vez que ela não enxerga o acúmulo de estoque.

Para contornar as limitações práticas da formulação clássica do problema de corte de estoque, formulou-se o problema de corte de estoque estocástico como um processo de decisão markoviano. Foi então proposta uma política de decisão com base na técnica de aprendizado por reforço. Os resultados dos experimentos computacionais realizados indicaram que a política proposta pode ter um custo médio até cinquenta vezes menor que uma política míope. Conclui-se que a política desenvolvida para tratar esse tipo de problema é muito eficiente para auxiliar a tomada de decisões que são necessárias serem executadas ao enfrentar esse problema na prática gerando custos acumulados muito menores comparados com uma política míope.

Como trabalhos futuros, sugere-se utilizar outros métodos de aproximação da tabela-Q como redes neurais (DQN), ou modificar a forma de aprendizado para o Q-learning duplo (HASSELT, 2010; HASSELT *et al.*, 2016). Uma outra alteração poderia ser utilizar modelos mais complexos como o corte de estoque bidimensional multiperíodo.

REFERÊNCIAS

ALEM, D. J.; MUNARI, P. A.; ARENALES, M. N.; FERREIRA, P. A. V. On the cutting stock problem under stochastic demand. **Annals of Operations Research**, [Netherlands], v. 179, n. 1, p. 169–186, 2010.

BELLMAN, R. The theory of dynamic programming. **Bulletin of the American Mathematical Society**, Providence, RI, v. 60, n. 6, p. 503–516, 1954.

BERTSEKAS, D.; TSITSIKLIS, J.; Tη, G. **Neuro-dynamic programming**. Belmont, MA: Athena Scientific, 1996. (Anthropological Field Studies). ISBN 9781886529106. Disponível em: https://books.google.com.br/books?id=WxCCQgAACAAJ. Acesso em: 30 de outubro de 2020

BERTSEKAS, D. P.; BERTSEKAS, D. P.; BERTSEKAS, D. P.; BERTSEKAS, D. P. **Dynamic programming and optimal control**. Belmont, MA: Athena Scientific, 1995. v. 1.

BIRGIN, E. G.; ROMãO, O. C.; RONCONI, D. P. The multiperiod two-dimensional non-guillotine cutting stock problem with usable leftovers. **International Transactions in Operational Research**, [United Kingdom], v. 27, n. 3, p. 1392–1418, 2020.

COSTA, L. L. S. **Um estudo sobre o problema de corte de estoque bidimensional 2-estágios**. 2016. 103 f. Dissertação (Mestrado) — Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica, Campinas, SP, 2016. Disponível em: http://www.repositorio.unicamp.br/handle/REPOSIP/307535. Acesso em: 03 de novembro de 2020

DAI, H.; DAI, B.; SONG, L. Discriminative embeddings of latent variable models for structured data. *In*: INTERNATIONAL CONFERENCE ON MACHINE LEARNING. 33., 2016, New York, NY. **Proceedings** [...] . [*S. l.*]: JMLR, 2016. p. 2702–2711.

FARLEY, A. A. Mathematical programming models for cutting-stock problems in the clothing industry. **Journal of the Operational Research Society**, United Kingdom, v. 39, n. 1, p. 41–53, 1988.

GEORGE, A. P.; POWELL, W. B. Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. **Machine Learning**, Netherlands, v. 65, n. 1, p. 167–198, Oct 2006. ISSN 1573-0565. Disponível em: https://doi.org/10.1007/s10994-006-8365-9. Acesso em: 23 de outubro de 2020

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. **Operations Research**, United States, v. 9, n. 6, p. 849–859, 1961. ISSN 0030364X, 15265463.

GRAMANI, M. C. N. **Otimização do processo de cortagem acoplado ao planejamento da produção**. 2001. Tese (Doutorado) — Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP, 2001. Disponível em: http://www.repositorio.unicamp.br/handle/REPOSIP/260294. Acesso em: 20 de novembro de 2020

HASSELT, H. Double q-learning. *In*: ANNUAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 24., 2010, Vancouver, Canada. **Advances in neural information processing systems 23** [...]. New York: Curran Associates, 2011. p. 2613–2621.

HASSELT, H. V.; GUEZ, A.; SILVER, D. Deep reinforcement learning with double q-learning. *In*: AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE, 13., 2016, Phoenix, Arizona, USA. **Proceedings** [...]. Palo Alto, CA: AAAI Press, 2016. v. 30, n. 1.

HENDRY, L.; FOK, K.; SHEK, K. A cutting stock and scheduling problem in the copper industry. **Journal of the Operational Research Society**, United Kingdom, v. 47, n. 1, p. 38–47, 1996.

JAFFAR, J.; LASSEZ, J.-L. Constraint logic programming. *In*: ACM SIGACT-SIGPLAN SYMPOSIUM ON PRINCIPLES OF PROGRAMMING LANGUAGES, 14., 1987, Munich, West German. **Proceedings** [...]. New York: ACM, 1987. p. 111–119. Disponível em:https://dl.acm.org/doi/10.1145/41625.41635. Acesso em: 30 de outubro de 2020

KANTOROVICH, L.; ZALGALLER, V. Calculation of rational cutting of stock. **Lenizdat,** Leningrad, v. 5, p. 11–14, 1951.

KANTOROVICH, L. V. A new method of solving of some classes of extremal problems. *In*: **Dokl. Akad. Nauk SSSR**. [*S.l.*: *s.n.*], 1940. v. 28, p. 211–214.

KANTOROVICH, L. V. Mathematical methods of organizing and planning production. **Management science**, United States, v. 6, n. 4, p. 366–422, 1960.

KHALIL, E.; DAI, H.; ZHANG, Y.; DILKINA, B.; SONG, L. Learning combinatorial optimization algorithms over graphs. *In*: ANNUAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 31., 2017, Long Beach, California, USA . **Advances in neural information processing systems 30** [...]. New York: Curran Associates, 2018. p. 6348–6358.

KOOL, W.; HOOF, H. van; WELLING, M. Attention, learn to solve routing problems! **arXiv.org**, [Ithaca, N. Y.], 2018. Disponível em: https://arxiv.org/abs/1803.08475. Acesso em: 23 de outubro de 2020

KRICHAGINA, E. V.; RUBIO, R.; TAKSAR, M. I.; WEIN, L. M. A dynamic stochastic stock-cutting problem. **Operations Research**, United States, v. 46, n. 5, p. 690–701, 1998.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. *et al.* Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Taipei, Taiwan, v. 86, n. 11, p. 2278–2324, 1998.

LIROV, Y. Knowledge based approach to the cutting stock problem. **Mathematical and computer modelling**, United Kingdom, v. 16, n. 1, p. 107–125, 1992.

MA, N.; LIU, Y.; ZHOU, Z. Two heuristics for the capacitated multi-period cutting stock problem with pattern setup cost. **Computers & Operations Research**, United Kingdom, v. 109, p. 218–229, 2019.

MCCARTHY, J. Programs with common sense. *In*: SYMPOSIUM AT THE NATIONAL PHYSICAL LABORATORY, 10., November 1958, Teddington, Middlesex, England. **Mechanization of Thought Processes**: proceedings [...]. London: H.M. Stationery Office, 1961. p. 1-10.

MIYAZAWA, F. K.; SOUZA, C. C. D. Introdução à otimização combinatória. *In*: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA, 34, 2015, Recife, PE. **Anais** [...]. Porto Alegre: SBC, 2021. p. 123-189.

MNIH, V.; KAVUKCUOGLU, K.; SILVER, D.; RUSU, A. A.; VENESS, J.; BELLEMARE, M. G.; GRAVES, A.; RIEDMILLER, M.; FIDJELAND, A. K.; OSTROVSKI, G. *et al.* Human-level control through deep reinforcement learning. **Nature**, United Kingdom, v. 518, n. 7540, p. 529, 2015.

MURTA, A. H. F.; PITOMBEIRA NETO, A. R. Uma abordagem baseada em programação dinâmica aproximada para o problema de corte de estoque multiperíodo estocástico. *In*: CONFERÊNCIA BRASILEIRA DE DINÂMICA, CONTROLE E APLICAÇÕES, 14., 2019, São Carlos, São Paulo. **Anais** [...]. São Carlos: EESC/ICMC-USP, 2019.

NONåS, S. L.; THORSTENSON, A. A combined cutting-stock and lot-sizing problem. **European Journal of Operational Research**, Netherlands, v. 120, n. 2, p. 327–342, 2000.

PAPADIMITRIOU, C. H.; STEIGLITZ, K. **Combinatorial optimization**: algorithms and complexity. [*S. l.*]: Courier Corporation, 1998.

PAULL, A. Linear programming: A key to optimum newsprint production. **Pulp and Paper Magazine of Canada**, Canada, v. 57, n. 1, p. 85–90, 1956.

PITOMBEIRA-NETO, A. R.; PRATA, B. de A. A matheuristic algorithm for the one dimensional cutting stock and scheduling problem with heterogeneous orders. **Top**, Germany, v. 28, p.1–15, 2019.

POLDI, K. C.; ARAUJO, S. A. de. Mathematical models and a heuristic method for the multiperiod one-dimensional cutting stock problem. **Annals of Operations Research**, [Netherlands], v. 238, n. 1-2, p. 497–520, 2016.

POLDI, K. C.; ARENALES, M. N. O problema de corte de estoque unidimensional multiperíodo. **Pesquisa Operacional**, Rio de Janeiro, v. 30, n. 1, p. 153–174, 2010.

POLTRONIERE, S. C.; POLDI, K. C.; TOLEDO, F. M. B.; ARENALES, M. N. A coupling cutting stock-lot sizing problem in the paper industry. **Annals of Operations Research**, [Netherlands], v. 157, n. 1, p. 91–104, 2008.

POWELL, W. **Approximate dynamic programming**: solving the curses of dimensionality. Wiley, 2011. (Wiley Series in Probability and Statistics). ISBN 9780470182956. Disponível em: https://books.google.com.nf/books?id=WWWDkd65TdYC. Acesso em: 07 de novembro de 2020

PRATA, B. de A.; PITOMBEIRA-NETO, A. R.; SALES, C. J. de M. An integer linear

programming model for the multiperiod production planning of precast concrete beams. **Journal of Construction Engineering and Management**, United States, v. 141, n. 10, p. 04015029, 2015.

RESPICIO, A.; CAPTIVO, M.; RODRIGUES, A. A DSS for production planning and scheduling in the paper industry. *In*: INTERNATIONAL CONFERENCE ON DECISION MAKING AND DECISION SUPPORT IN THE INTERNET AGE, 2002, Cork, Ireland. **Proceedings** [...]. Oxfordshire, U. K.: Oak Tree, 2002. p. 298–308

SAMMA, H.; MOHAMAD-SALEH, J.; SUANDI, S. A.; LAHASAN, B. Q-learning based simulated annealing algorithm for constrained engineering design problems. **Neural Computing and Applications**, United Kingdom, v. 32, p. 5147–5161, 2020. ISSN 1433-3058. Disponível em: https://doi.org/10.1007/s00521-019-04008-z. Acesso em: 23 de novembro de 2020

SCHULMAN, J.; WOLSKI, F.; DHARIWAL, P.; RADFORD, A.; KLIMOV, O. Proximal policy optimization algorithms. **arXiv.org**, [Ithaca, N. Y.], 2017. Disponível em: https://arxiv.org/abs/1707.06347. Acesso em: 11 de novembro de 2020

SUTTON, R.; BARTO, A. **Reinforcement learning**: an introduction. Cambridge, MA: MIT Press, 2018.(Adaptive Computation and Machine Learning series). ISBN 9780262039246. Disponível em: https://books.google.com.br/books?id=6DKPtQEACAAJ. Acesso em: 15 de novembro de 2020

SUTTON, R. S.; BARTO, A. G. **Reinforcement learning**: an introduction. [*S. l.*: *s. n.*], 2018.

VINYALS, O.; FORTUNATO, M.; JAITLY, N. Pointer networks. In: ANNUAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 29., 2015, Montreal, Canada. **Advances in neural information processing systems 28** [...]. New York: Curran Associates, 2016. p. 2692–2700.

WATKINS, C. J. C. H. **Learning from delayed rewards**. 1989. 234 f. Tese (doutorado) – King's College, Cambridge, 1989.

WEST, M.; HARRISON, J. **Bayesian Forecasting and Dynamic Models**. Springer New York, 1999. (Springer Series in Statistics). ISBN 9780387947259. Disponível em: https://books.google.com.br/books?id=jcl8lD75fkYC. Acesso em: 12 de novembro de 2020

WäSCHER, G.; HAUßNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. **Eur J Oper Res**, [Netherlands], v. 183, n. 3, p. 1109 – 1130, 2007.

APÊNDICE A – QUADROS

A.1 Padrões de corte

O Quadro 2 contém os padrões de corte onde cada linha representa um padrão de corte ótimo. Isso significa que não tem nenhum outro padrão de corte valido que contenha esse padrão de corte.

Quadro 2 – Padrões de corte ótimos

115 mm	180 mm	267 mm	314 mm	880 mm	1180 mm	1200 mm	sobra
7	2	0	1	0	0	0	21 mm
5	2	2	0	0	0	0	31 mm
8	0	2	0	0	0	0	46 mm
9	2	0	0	0	0	0	105 mm
4	4	1	0	0	0	0	53 mm
3	1	1	2	0	0	0	80 mm
2	4	2	0	0	0	0	16 mm
1	4	0	2	0	0	0	37 mm
0	8	0	0	0	0	0	60 mm
4	1	2	1	0	0	0	12 mm
7	0	1	1	0	0	0	114 mm
3	6	0	0	0	0	0	75 mm
1	6	1	0	0	0	0	38 mm
1	0	4	1	0	0	0	3 mm
2	7	0	0	0	0	0	10 mm
0	2	4	0	0	0	0	72 mm
10	0	0	1	0	0	0	36 mm
0	6	0	1	0	0	0	106 mm
5	5	0	0	0	0	0	25 mm
0	5	2	0	0	0	0	66 mm
3	0	1	0	1	0	0	8 mm
2	2	1	2	0	0	0	15 mm

Continua na próxima pagina

Quadro 2 - Padrões de corte ótimos- Continuação da pagina anterior

115 mm	180 mm	267 mm	314 mm	880 mm	1180 mm	1200 mm	sobra
4	4	0	1	0	0	0	6 mm
3	1	0	3	0	0	0	33 mm
0	0	1	1	1	0	0	39 mm
1	3	3	0	0	0	0	44 mm
0	0	3	2	0	0	0	71 mm
11	1	0	0	0	0	0	55 mm
0	0	2	0	1	0	0	86 mm
0	3	0	3	0	0	0	18 mm
6	4	0	0	0	0	0	90 mm
5	0	0	0	1	0	0	45 mm
3	0	3	1	0	0	0	40 mm
8	1	0	1	0	0	0	86 mm
6	1	0	2	0	0	0	2 mm
3	3	2	0	0	0	0	81 mm
0	0	2	3	0	0	0	24 mm
2	0	1	3	0	0	0	61 mm
5	3	0	1	0	0	0	71 mm
2	0	0	0	0	1	0	90 mm
0	1	0	4	0	0	0	64 mm
5	0	2	1	0	0	0	77 mm
0	0	1	0	0	1	0	53 mm
0	3	1	2	0	0	0	65 mm
3	3	1	1	0	0	0	34 mm
2	1	4	0	0	0	0	22 mm
8	3	0	0	0	0	0	40 mm
0	0	1	0	0	0	1	33 mm
0	2	3	1	0	0	0	25 mm
2	2	0	0	1	0	0	30 mm
1	1	1	0	1	0	0	58 mm

Continua na próxima pagina

Quadro 2 - Padrões de corte ótimos- Continuação da pagina anterior

115 mm	180 mm	267 mm	314 mm	880 mm	1180 mm	1200 mm	sobra
1	2	0	3	0	0	0	83 mm
2	0	0	1	1	0	0	76 mm
1	1	0	1	1	0	0	11 mm
3	0	4	0	0	0	0	87 mm
1	0	5	0	0	0	0	50 mm
1	1	2	2	0	0	0	43 mm
10	0	1	0	0	0	0	83 mm
0	3	2	1	0	0	0	112 mm
0	3	0	0	1	0	0	80 mm
4	1	3	0	0	0	0	59 mm
0	1	1	3	0	0	0	111 mm
6	3	1	0	0	0	0	3 mm
6	0	3	0	0	0	0	9 mm
6	1	2	0	0	0	0	96 mm
2	0	0	4	0	0	0	14 mm
1	1	0	0	0	0	1	5 mm
1	1	0	0	0	1	0	25 mm
2	0	0	0	0	0	1	70 mm
13	0	0	0	0	0	0	5 mm
9	1	1	0	0	0	0	18 mm
2	2	3	0	0	0	0	109 mm
2	2	2	1	0	0	0	62 mm
2	5	1	0	0	0	0	103 mm
3	1	0	0	1	0	0	95 mm
2	0	2	2	0	0	0	108 mm
2	3	0	2	0	0	0	102 mm
7	2	1	0	0	0	0	68 mm
1	1	3	1	0	0	0	90 mm
4	2	0	2	0	0	0	52 mm

Continua na próxima pagina

Quadro 2 - Padrões de corte ótimos- Continuação da pagina anterior

115 mm	180 mm	267 mm	314 mm	880 mm	1180 mm	1200 mm	sobra
6	1	1	1	0	0	0	49 mm
0	5	1	1	0	0	0	19 mm
0	0	0	1	0	1	0	6 mm
4	2	1	1	0	0	0	99 mm
5	0	1	2	0	0	0	30 mm
4	0	0	3	0	0	0	98 mm
2	5	0	1	0	0	0	56 mm
1	4	1	1	0	0	0	84 mm
7	0	0	2	0	0	0	67 mm

O Quadro 3 apresenta os padrões de corte subótimos que são padrões os quais as sobras deles cabem algum outro item, mas não cabem mais nenhuma outra unidade de um item que já está nele.

Quadro 3 – Padrões de corte subótimos

115 mm	180 mm	267 mm	314 mm	880 mm	1180 mm	1200 mm	sobra
0	0	0	4	0	0	0	244 mm
0	0	0	0	1	0	0	620 mm
0	1	0	0	0	1	0	140 mm
0	6	1	0	0	0	0	153 mm
0	1	0	1	1	0	0	126 mm
0	3	3	0	0	0	0	159 mm
0	0	0	0	0	0	1	300 mm
0	0	0	0	0	1	0	320 mm
0	0	5	0	0	0	0	165 mm
0	1	0	0	0	0	1	120 mm
0	4	0	2	0	0	0	152 mm

Continua na próxima pagina

Quadro 3 - Padrões de corte subótimos- Continuação da pagina anterior

115 mm	180 mm	267 mm	314 mm	880 mm	1180 mm	1200 mm	sobra
0	1	1	0	1	0	0	173 mm
0	1	2	2	0	0	0	158 mm
0	0	4	1	0	0	0	118 mm
0	0	0	1	1	0	0	306 mm

No Quadro 4 estão os padrões de corte para as simulações 1-10 do experimento E. A primeira linha é o tamanho de cada item utilizado no experimento.

Quadro 4 – Padrões de corte ótimos e subótimos para o experimento E - simulações 1-10

200 mm	300 mm	500 mm	Sobra
2	0	2	100 mm
0	3	1	100 mm
0	0	3	0 mm
0	5	0	0 mm
1	4	0	100 mm
7	0	0	100 mm
5	0	1	0 mm
3	3	0	0 mm
4	2	0	100 mm
0	1	2	200 mm
6	1	0	0 mm

No Quadro 5 estão os padrões de corte para as simulações 11-20 do experimento E. A primeira linha é o tamanho de cada item utilizado no experimento.

Quadro 5 – Padrões de corte ótimos e subótimos para o experimento E - simulações 11-20

200 mm	300 mm	500 mm	800 mm	1000 mm	1200 mm	Sobra
0	0	1	0	1	0	0 mm
1	0	1	1	0	0	0 mm
1	1	0	0	1	0	0 mm

Quadro 5 continuação da pagina anterior

200 mm	300 mm	500 mm	800 mm	1000 mm	1200 mm	Sobra
5	0	1	0	0	0	0 mm
1	1	2	0	0	0	0 mm
0	1	0	0	0	1	0 mm
1	4	0	0	0	0	100 mm
0	5	0	0	0	0	0 mm
0	0	1	1	0	0	200 mm
2	0	2	0	0	0	100 mm
2	1	0	1	0	0	0 mm
3	3	0	0	0	0	0 mm
6	1	0	0	0	0	0 mm
0	0	3	0	0	0	0 mm
7	0	0	0	0	0	100 mm
4	2	0	0	0	0	100 mm
1	0	0	0	0	1	100 mm
3	0	0	1	0	0	100 mm
0	2	0	1	0	0	100 mm
2	0	0	0	1	0	100 mm
3	1	1	0	0	0	100 mm
0	3	1	0	0	0	100 mm
0	0	0	0	1	0	500 mm
0	0	0	1	0	0	700 mm
0	1	0	0	1	0	200 mm
2	2	1	0	0	0	0 mm
0	1	2	0	0	0	200 mm
0	0	0	0	0	1	300 mm

No Quadro 6 estão os padrões de corte para as simulações 21-30 do experimento E. A primeira linha é o tamanho de cada item utilizado no experimento em milimetros.

Quadro 6 – Padrões de corte ótimos e subótimos para o experimento E - simulações 21-30

100	**200**	**300**	**500**	**700**	**800**	**900**	**1000**	**1200**	**Sobra**
3	1	1	0	1	0	0	0	0	0 mm
1	4	2	0	0	0	0	0	0	0 mm
3	1	0	2	0	0	0	0	0	0 mm
5	2	2	0	0	0	0	0	0	0 mm
3	0	4	0	0	0	0	0	0	0 mm
2	0	1	2	0	0	0	0	0	0 mm
3	0	0	0	0	0	0	0	1	0 mm
6	0	3	0	0	0	0	0	0	0 mm
8	0	0	0	1	0	0	0	0	0 mm
4	2	0	0	1	0	0	0	0	0 mm
1	7	0	0	0	0	0	0	0	0 mm
0	0	1	2	0	0	0	0	0	200 mm
0	3	3	0	0	0	0	0	0	0 mm
5	5	0	0	0	0	0	0	0	0 mm
11	2	0	0	0	0	0	0	0	0 mm
0	0	2	0	0	0	1	0	0	0 mm
4	0	1	0	0	1	0	0	0	0 mm
9	0	2	0	0	0	0	0	0	0 mm
1	1	0	0	0	0	0	0	1	0 mm
0	2	0	0	0	0	0	1	0	100 mm
0	0	1	0	0	0	0	0	1	0 mm
1	3	1	1	0	0	0	0	0	0 mm
4	1	3	0	0	0	0	0	0	0 mm
2	2	3	0	0	0	0	0	0	0 mm
15	0	0	0	0	0	0	0	0	0 mm
5	1	0	0	0	1	0	0	0	0 mm
4	1	0	0	0	0	1	0	0	0 mm
8	2	1	0	0	0	0	0	0	0 mm
3	3	2	0	0	0	0	0	0	0 mm
0	3	0	0	0	0	1	0	0	0 mm

Quadro 6 continuação da pagina anterior

100	200	300	500	700	800	900	1000	1200	Sobra
0	2	1	0	0	1	0	0	0	0 mm
0	0	0	0	0	0	0	1	0	500 mm
1	2	1	0	1	0	0	0	0	0 mm
0	0	0	3	0	0	0	0	0	0 mm
0	1	2	0	1	0	0	0	0	0 mm
1	1	4	0	0	0	0	0	0	0 mm
3	0	0	1	1	0	0	0	0	0 mm
0	0	0	0	0	0	0	0	1	300 mm
0	2	2	1	0	0	0	0	0	0 mm
3	2	0	0	0	1	0	0	0	0 mm
0	1	1	0	0	0	1	0	0	100 mm
0	0	0	1	0	0	0	1	0	0 mm
5	0	0	2	0	0	0	0	0	0 mm
10	0	0	1	0	0	0	0	0	0 mm
0	0	0	0	0	1	0	0	0	700 mm
2	0	0	1	0	1	0	0	0	0 mm
2	1	2	1	0	0	0	0	0	0 mm
0	3	1	1	0	0	0	0	0	100 mm
4	3	0	1	0	0	0	0	0	0 mm
0	7	0	0	0	0	0	0	0	100 mm
0	0	0	1	1	0	0	0	0	300 mm
2	0	2	0	1	0	0	0	0	0 mm
2	5	1	0	0	0	0	0	0	0 mm
0	0	1	1	1	0	0	0	0	0 mm
0	2	1	0	1	0	0	0	0	100 mm
2	1	1	0	0	1	0	0	0	0 mm
1	0	0	1	0	0	1	0	0	0 mm
1	0	0	0	2	0	0	0	0	0 mm
0	1	0	1	1	0	0	0	0	100 mm
1	3	0	0	0	1	0	0	0	0 mm

Quadro 6 continuação da pagina anterior

100	200	300	500	700	800	900	1000	1200	Sobra
1	1	0	1	1	0	0	0	0	0 mm
0	6	1	0	0	0	0	0	0	0 mm
10	1	1	0	0	0	0	0	0	0 mm
0	1	1	2	0	0	0	0	0	0 mm
0	1	1	0	0	0	0	1	0	0 mm
0	0	0	0	2	0	0	0	0	100 mm
0	0	0	0	1	1	0	0	0	0 mm
1	2	0	2	0	0	0	0	0	0 mm
1	0	2	0	0	1	0	0	0	0 mm
7	0	0	0	0	1	0	0	0	0 mm
5	0	0	0	0	0	0	1	0	0 mm
0	0	1	0	0	0	0	1	0	200 mm
5	1	1	1	0	0	0	0	0	0 mm
4	4	1	0	0	0	0	0	0	0 mm
7	0	1	1	0	0	0	0	0	0 mm
0	0	0	0	0	0	1	0	0	600 mm
1	2	0	0	0	0	0	1	0	0 mm
0	3	0	0	0	1	0	0	0	100 mm
5	0	1	0	1	0	0	0	0	0 mm
3	6	0	0	0	0	0	0	0	0 mm
3	1	0	0	0	0	0	1	0	0 mm
7	1	2	0	0	0	0	0	0	0 mm
0	0	2	0	0	1	0	0	0	100 mm
2	0	1	0	0	0	0	1	0	0 mm
2	4	0	1	0	0	0	0	0	0 mm
0	1	0	0	0	0	0	0	1	100 mm
6	3	1	0	0	0	0	0	0	0 mm
0	4	2	0	0	0	0	0	0	100 mm
2	2	0	0	0	0	1	0	0	0 mm
4	0	2	1	0	0	0	0	0	0 mm

Quadro 6 continuação da pagina anterior

100	200	300	500	700	800	900	1000	1200	Sobra
0	0	3	1	0	0	0	0	0	100 mm
3	0	1	0	0	0	1	0	0	0 mm
1	1	1	0	0	0	1	0	0	0 mm
0	0	0	1	0	0	1	0	0	100 mm
6	0	0	0	0	0	1	0	0	0 mm
2	3	0	0	1	0	0	0	0	0 mm
0	0	2	0	1	0	0	0	0	200 mm
13	1	0	0	0	0	0	0	0	0 mm
12	0	1	0	0	0	0	0	0	0 mm
9	3	0	0	0	0	0	0	0	0 mm
3	2	1	1	0	0	0	0	0	0 mm
7	4	0	0	0	0	0	0	0	0 mm
0	1	4	0	0	0	0	0	0	100 mm
0	5	0	1	0	0	0	0	0	0 mm
6	2	0	1	0	0	0	0	0	0 mm
1	0	3	1	0	0	0	0	0	0 mm
6	1	0	0	1	0	0	0	0	0 mm
0	0	5	0	0	0	0	0	0	0 mm
0	1	0	1	0	1	0	0	0	0 mm
8	1	0	1	0	0	0	0	0	0 mm
0	2	0	2	0	0	0	0	0	100 mm
0	4	0	0	1	0	0	0	0	0 mm
0	0	0	1	0	1	0	0	0	200 mm

No Quadro 7 estão os padrões de corte para as simulações 31-40 do experimento E. A primeira linha é o tamanho de cada item utilizado no experimento em milimetros.

Quadro 7 – Padrões de corte ótimos e subótimos para o experimento E - simulações 31-40

100	200	300	400	500	600	700	800	900	1000	1200	1300	Sobra
0	0	0	0	0	0	1	1	0	0	0	0	0 mm
0	0	1	0	0	2	0	0	0	0	0	0	0 mm

Quadro 7 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1200	1300	Sobra
0	1	4	0	0	0	0	0	0	0	0	0	100 mm
4	1	0	0	0	0	0	0	1	0	0	0	0 mm
1	0	2	2	0	0	0	0	0	0	0	0	0 mm
0	0	3	1	0	0	0	0	0	0	0	0	200 mm
2	1	0	1	0	0	1	0	0	0	0	0	0 mm
1	5	0	1	0	0	0	0	0	0	0	0	0 mm
5	1	0	0	0	0	0	1	0	0	0	0	0 mm
0	2	2	1	0	0	0	0	0	0	0	0	100 mm
2	0	0	0	0	0	0	0	0	0	0	1	0 mm
0	3	0	1	1	0	0	0	0	0	0	0	0 mm
5	0	0	1	0	1	0	0	0	0	0	0	0 mm
0	3	1	0	0	1	0	0	0	0	0	0	0 mm
0	0	0	0	0	2	0	0	0	0	0	0	300 mm
0	7	0	0	0	0	0	0	0	0	0	0	100 mm
4	3	0	0	1	0	0	0	0	0	0	0	0 mm
0	5	0	0	1	0	0	0	0	0	0	0	0 mm
0	0	0	0	1	0	1	0	0	0	0	0	300 mm
0	2	1	0	0	0	1	0	0	0	0	0	100 mm
0	0	0	2	0	1	0	0	0	0	0	0	100 mm
0	0	1	0	0	0	0	0	0	0	1	0	0 mm
1	0	0	2	0	1	0	0	0	0	0	0	0 mm
1	1	0	1	0	0	0	1	0	0	0	0	0 mm
6	0	0	0	0	0	0	0	1	0	0	0	0 mm
0	0	1	1	0	1	0	0	0	0	0	0	200 mm
8	0	0	0	0	0	1	0	0	0	0	0	0 mm
5	1	0	2	0	0	0	0	0	0	0	0	0 mm
0	0	2	2	0	0	0	0	0	0	0	0	100 mm
0	0	0	0	0	1	0	0	1	0	0	0	0 mm
0	0	0	1	0	0	0	1	0	0	0	0	300 mm
2	2	0	0	0	0	0	0	1	0	0	0	0 mm

Quadro 7 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1200	1300	Sobra
0	3	0	0	0	0	0	1	0	0	0	0	100 mm
4	0	0	0	1	1	0	0	0	0	0	0	0 mm
10	0	0	0	1	0	0	0	0	0	0	0	0 mm
4	1	3	0	0	0	0	0	0	0	0	0	0 mm
4	1	0	1	1	0	0	0	0	0	0	0	0 mm
0	0	0	0	3	0	0	0	0	0	0	0	0 mm
9	0	0	0	0	1	0	0	0	0	0	0	0 mm
0	3	0	2	0	0	0	0	0	0	0	0	100 mm
7	0	1	0	1	0	0	0	0	0	0	0	0 mm
1	3	1	0	1	0	0	0	0	0	0	0	0 mm
6	0	1	0	0	1	0	0	0	0	0	0	0 mm
0	1	0	1	0	0	0	0	1	0	0	0	0 mm
0	0	1	3	0	0	0	0	0	0	0	0	0 mm
1	3	0	0	0	0	0	1	0	0	0	0	0 mm
3	2	0	2	0	0	0	0	0	0	0	0	0 mm
3	0	0	0	0	0	0	0	0	0	1	0	0 mm
1	4	0	0	0	1	0	0	0	0	0	0	0 mm
5	0	0	0	2	0	0	0	0	0	0	0	0 mm
1	2	0	1	0	1	0	0	0	0	0	0	0 mm
8	1	0	0	1	0	0	0	0	0	0	0	0 mm
3	4	0	1	0	0	0	0	0	0	0	0	0 mm
3	1	0	0	0	0	0	0	0	1	0	0	0 mm
5	0	1	0	0	0	1	0	0	0	0	0	0 mm
0	0	0	0	0	0	2	0	0	0	0	0	100 mm
0	1	1	0	0	0	0	0	1	0	0	0	100 mm
3	0	0	3	0	0	0	0	0	0	0	0	0 mm
0	2	0	0	0	0	0	0	0	1	0	0	100 mm
2	1	1	0	0	0	0	1	0	0	0	0	0 mm
2	1	1	2	0	0	0	0	0	0	0	0	0 mm
1	0	2	0	0	0	0	1	0	0	0	0	0 mm

Quadro 7 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1200	1300	Sobra
1	1	0	3	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	1	0	0	1	0	0	0	0	200 mm
2	1	0	0	1	1	0	0	0	0	0	0	0 mm
6	0	0	1	1	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	1	1	0	0	0	0	0	200 mm
0	3	3	0	0	0	0	0	0	0	0	0	0 mm
7	1	0	0	0	1	0	0	0	0	0	0	0 mm
12	0	1	0	0	0	0	0	0	0	0	0	0 mm
13	1	0	0	0	0	0	0	0	0	0	0	0 mm
0	5	0	1	0	0	0	0	0	0	0	0	100 mm
0	0	2	0	0	0	0	0	1	0	0	0	0 mm
0	0	0	0	1	0	0	0	0	1	0	0	0 mm
0	1	0	0	0	1	1	0	0	0	0	0	0 mm
3	2	1	0	1	0	0	0	0	0	0	0	0 mm
0	4	2	0	0	0	0	0	0	0	0	0	100 mm
3	1	2	1	0	0	0	0	0	0	0	0	0 mm
0	0	0	1	2	0	0	0	0	0	0	0	100 mm
3	1	0	0	2	0	0	0	0	0	0	0	0 mm
0	3	1	0	1	0	0	0	0	0	0	0	100 mm
2	3	0	0	0	0	1	0	0	0	0	0	0 mm
5	0	2	1	0	0	0	0	0	0	0	0	0 mm
6	2	0	0	1	0	0	0	0	0	0	0	0 mm
7	2	0	1	0	0	0	0	0	0	0	0	0 mm
1	0	0	0	1	0	0	0	1	0	0	0	0 mm
0	1	0	0	1	0	1	0	0	0	0	0	100 mm
0	0	1	1	0	0	1	0	0	0	0	0	100 mm
2	0	3	1	0	0	0	0	0	0	0	0	0 mm
0	0	0	2	1	0	0	0	0	0	0	0	200 mm
1	0	0	1	2	0	0	0	0	0	0	0	0 mm
0	1	3	1	0	0	0	0	0	0	0	0	0 mm

Quadro 7 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1200	1300	Sobra
7	4	0	0	0	0	0	0	0	0	0	0	0 mm
1	1	0	0	0	2	0	0	0	0	0	0	0 mm
2	2	1	0	0	1	0	0	0	0	0	0	0 mm
3	0	2	0	0	1	0	0	0	0	0	0	0 mm
1	0	0	0	0	1	0	1	0	0	0	0	0 mm
0	0	0	1	0	0	0	0	0	1	0	0	100 mm
4	0	0	1	0	0	1	0	0	0	0	0	0 mm
0	0	0	0	0	0	0	0	0	0	1	0	300 mm
2	2	3	0	0	0	0	0	0	0	0	0	0 mm
2	0	0	1	0	0	0	0	1	0	0	0	0 mm
0	0	1	0	1	1	0	0	0	0	0	0	100 mm
6	0	3	0	0	0	0	0	0	0	0	0	0 mm
1	2	0	0	0	0	0	0	0	1	0	0	0 mm
2	0	0	0	1	0	0	1	0	0	0	0	0 mm
0	0	0	2	0	0	1	0	0	0	0	0	0 mm
0	1	0	0	1	0	0	1	0	0	0	0	0 mm
1	0	1	0	1	1	0	0	0	0	0	0	0 mm
0	0	0	0	0	0	0	0	1	0	0	0	600 mm
0	2	1	2	0	0	0	0	0	0	0	0	0 mm
5	3	0	1	0	0	0	0	0	0	0	0	0 mm
0	0	2	1	1	0	0	0	0	0	0	0	0 mm
1	1	1	1	1	0	0	0	0	0	0	0	0 mm
0	1	0	3	0	0	0	0	0	0	0	0	100 mm
0	0	0	0	0	0	0	1	0	0	0	0	700 mm
3	6	0	0	0	0	0	0	0	0	0	0	0 mm
0	1	0	0	0	2	0	0	0	0	0	0	100 mm
4	0	1	0	0	0	0	1	0	0	0	0	0 mm
0	0	3	0	1	0	0	0	0	0	0	0	100 mm
6	1	1	1	0	0	0	0	0	0	0	0	0 mm
1	0	3	0	1	0	0	0	0	0	0	0	0 mm

Quadro 7 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1200	1300	Sobra
9	1	0	1	0	0	0	0	0	0	0	0	0 mm
6	1	0	0	0	0	1	0	0	0	0	0	0 mm
1	1	0	0	0	0	0	0	0	0	1	0	0 mm
2	0	0	0	0	1	1	0	0	0	0	0	0 mm
9	0	2	0	0	0	0	0	0	0	0	0	0 mm
7	0	0	0	0	0	0	1	0	0	0	0	0 mm
0	0	3	0	0	1	0	0	0	0	0	0	0 mm
10	1	1	0	0	0	0	0	0	0	0	0	0 mm
1	0	0	1	0	0	0	0	0	1	0	0	0 mm
1	3	0	2	0	0	0	0	0	0	0	0	0 mm
11	2	0	0	0	0	0	0	0	0	0	0	0 mm
3	0	0	0	0	2	0	0	0	0	0	0	0 mm
1	4	2	0	0	0	0	0	0	0	0	0	0 mm
0	2	0	0	1	1	0	0	0	0	0	0	0 mm
5	0	0	0	0	0	0	0	0	1	0	0	0 mm
2	0	1	1	0	1	0	0	0	0	0	0	0 mm
0	0	2	0	0	0	1	0	0	0	0	0	200 mm
0	1	0	1	0	0	0	1	0	0	0	0	100 mm
5	1	1	0	1	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	1	0	1	0	0	0	0	100 mm
0	0	1	0	2	0	0	0	0	0	0	0	200 mm
15	0	0	0	0	0	0	0	0	0	0	0	0 mm
11	0	0	1	0	0	0	0	0	0	0	0	0 mm
4	2	1	1	0	0	0	0	0	0	0	0	0 mm
0	6	1	0	0	0	0	0	0	0	0	0	0 mm
4	0	2	0	1	0	0	0	0	0	0	0	0 mm
2	1	2	0	1	0	0	0	0	0	0	0	0 mm
4	1	1	0	0	1	0	0	0	0	0	0	0 mm
3	0	1	0	0	0	0	0	1	0	0	0	0 mm
2	0	1	0	2	0	0	0	0	0	0	0	0 mm

Quadro 7 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1200	1300	Sobra
7	0	0	2	0	0	0	0	0	0	0	0	0 mm
3	3	2	0	0	0	0	0	0	0	0	0	0 mm
0	1	0	0	0	0	0	0	0	0	1	0	100 mm
0	2	2	0	1	0	0	0	0	0	0	0	0 mm
1	0	1	1	0	0	1	0	0	0	0	0	0 mm
6	3	1	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	0	0	0	0	0	0	1	200 mm
0	0	0	1	0	0	0	0	1	0	0	0	200 mm
0	0	2	0	0	0	0	1	0	0	0	0	100 mm
3	0	0	0	1	0	1	0	0	0	0	0	0 mm
0	0	1	1	0	0	0	1	0	0	0	0	0 mm
0	0	0	1	1	1	0	0	0	0	0	0	0 mm
2	3	1	1	0	0	0	0	0	0	0	0	0 mm
1	1	0	0	1	0	1	0	0	0	0	0	0 mm
1	1	4	0	0	0	0	0	0	0	0	0	0 mm
5	2	0	0	0	1	0	0	0	0	0	0	0 mm
0	2	0	1	0	1	0	0	0	0	0	0	100 mm
2	0	0	2	1	0	0	0	0	0	0	0	0 mm
2	5	1	0	0	0	0	0	0	0	0	0	0 mm
1	1	2	0	0	1	0	0	0	0	0	0	0 mm
7	1	2	0	0	0	0	0	0	0	0	0	0 mm
0	4	0	0	0	0	1	0	0	0	0	0	0 mm
0	2	1	0	0	0	0	1	0	0	0	0	0 mm
0	1	2	0	0	0	1	0	0	0	0	0	0 mm
8	2	1	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	0	0	0	0	1	0	0	500 mm
0	0	0	0	1	1	0	0	0	0	0	0	400 mm
0	0	0	0	1	0	0	0	1	0	0	0	100 mm
8	0	1	1	0	0	0	0	0	0	0	0	0 mm
0	1	2	0	0	1	0	0	0	0	0	0	100 mm

Quadro 7 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1200	1300	Sobra
1	0	0	0	0	0	2	0	0	0	0	0	0 mm
0	4	0	0	0	1	0	0	0	0	0	0	100 mm
2	0	1	0	0	0	0	0	0	1	0	0	0 mm
0	0	1	0	0	0	0	0	0	1	0	0	200 mm
3	0	1	1	1	0	0	0	0	0	0	0	0 mm
1	7	0	0	0	0	0	0	0	0	0	0	0 mm
5	5	0	0	0	0	0	0	0	0	0	0	0 mm
4	4	1	0	0	0	0	0	0	0	0	0	0 mm
0	0	1	0	1	0	1	0	0	0	0	0	0 mm
0	1	0	2	1	0	0	0	0	0	0	0	0 mm
4	0	1	2	0	0	0	0	0	0	0	0	0 mm
0	1	1	0	2	0	0	0	0	0	0	0	0 mm
9	3	0	0	0	0	0	0	0	0	0	0	0 mm
0	3	0	0	0	0	0	0	1	0	0	0	0 mm
3	0	0	1	0	0	0	1	0	0	0	0	0 mm
0	0	5	0	0	0	0	0	0	0	0	0	0 mm
0	1	1	0	0	0	0	0	0	1	0	0	0 mm
0	4	1	1	0	0	0	0	0	0	0	0	0 mm
1	1	1	0	0	0	0	0	1	0	0	0	0 mm
0	1	1	1	0	1	0	0	0	0	0	0	0 mm
0	2	0	0	2	0	0	0	0	0	0	0	100 mm
0	0	0	3	0	0	0	0	0	0	0	0	300 mm
1	2	1	0	0	0	1	0	0	0	0	0	0 mm
3	1	1	0	0	0	1	0	0	0	0	0	0 mm
2	2	0	1	1	0	0	0	0	0	0	0	0 mm
1	2	2	1	0	0	0	0	0	0	0	0	0 mm
3	2	0	0	0	0	0	1	0	0	0	0	0 mm
5	2	2	0	0	0	0	0	0	0	0	0	0 mm
0	1	1	1	1	0	0	0	0	0	0	0	100 mm
3	0	4	0	0	0	0	0	0	0	0	0	0 mm

Quadro 7 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1200	1300	Sobra
2	0	2	0	0	0	1	0	0	0	0	0	0 mm
1	2	0	0	2	0	0	0	0	0	0	0	0 mm
3	3	0	0	0	1	0	0	0	0	0	0	0 mm
4	2	0	0	0	0	1	0	0	0	0	0	0 mm
0	1	0	0	0	0	0	0	0	0	0	1	0 mm
3	1	0	1	0	1	0	0	0	0	0	0	0 mm
0	2	0	1	0	0	1	0	0	0	0	0	0 mm
2	4	0	0	1	0	0	0	0	0	0	0	0 mm

No Quadro 8 estão os padrões de corte para as simulações 41-50 do experimento E. A primeira linha é o tamanho de cada item utilizado no experimento em milimetros.

Quadro 8 – Padrões de corte ótimos e subótimos para o experimento E - simulações 41-50

100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500	Sobra
4	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0 mm
5	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	400 mm
6	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	300 mm
1	2	1	0	0	0	1	0	0	0	0	0	0	0	0	0 mm
5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
1	3	0	2	0	0	0	0	0	0	0	0	0	0	0	0 mm
7	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
9	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0 mm
0	4	2	0	0	0	0	0	0	0	0	0	0	0	0	100 mm
2	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0 mm
0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	1	3	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0 mm
0	1	1	0	2	0	0	0	0	0	0	0	0	0	0	0 mm
5	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0 mm
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0 mm
3	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
5	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0 mm

Quadro 8 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500	Sobra
5	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	100 mm
0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	200 mm
0	1	0	2	1	0	0	0	0	0	0	0	0	0	0	0 mm
1	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
1	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0 mm
10	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
7	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
3	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0 mm
3	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
1	2	0	1	0	1	0	0	0	0	0	0	0	0	0	0 mm
1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0 mm
0	0	3	0	0	1	0	0	0	0	0	0	0	0	0	0 mm
4	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	300 mm
5	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0 mm
0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0 mm
10	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
1	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	3	0	1	1	0	0	0	0	0	0	0	0	0	0	0 mm
0	5	0	1	0	0	0	0	0	0	0	0	0	0	0	100 mm
0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	200 mm
0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	100 mm
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0 mm
3	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0 mm
11	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
11	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	5	0	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
1	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0 mm
1	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0 mm
1	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	100 mm
13	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	1	2	0	0	1	0	0	0	0	0	0	0	0	0	100 mm
2	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0 mm

Quadro 8 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500	Sobra
0	6	1	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
7	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
0	4	0	0	0	0	1	0	0	0	0	0	0	0	0	0 mm
0	4	0	0	0	1	0	0	0	0	0	0	0	0	0	100 mm
2	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0 mm
2	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0 mm
0	2	1	2	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0 mm
9	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0 mm
0	2	0	1	0	0	1	0	0	0	0	0	0	0	0	0 mm
6	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0 mm
3	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0 mm
0	3	1	0	1	0	0	0	0	0	0	0	0	0	0	100 mm
6	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	2	2	1	0	0	0	0	0	0	0	0	0	0	0	100 mm
2	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0 mm
7	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	1	2	0	0	0	1	0	0	0	0	0	0	0	0	0 mm
3	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0 mm
2	1	2	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	100 mm
0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	100 mm
0	0	3	0	1	0	0	0	0	0	0	0	0	0	0	100 mm
0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	100 mm
2	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0 mm
6	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0 mm
2	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0 mm
5	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0 mm
6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0 mm
0	0	0	2	0	1	0	0	0	0	0	0	0	0	0	100 mm
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0 mm
8	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	600 mm
2	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0 mm

Quadro 8 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500	Sobra
0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0 mm
0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	100 mm
0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0 mm
4	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0 mm
4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0 mm
6	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	300 mm
5	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	100 mm
6	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	100 mm
5	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
4	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
3	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0 mm
1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0 mm
3	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0 mm
0	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0 mm
1	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0 mm
4	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0 mm
4	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0 mm
0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	200 mm
4	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0 mm
1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0 mm
2	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0 mm
4	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0 mm
0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	200 mm
1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0 mm
3	4	0	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	2	0	1	0	1	0	0	0	0	0	0	0	0	0	100 mm
2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0 mm
0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	300 mm
0	3	1	0	0	1	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	300 mm
4	3	0	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	100 mm
2	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0 mm

Quadro 8 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500	Sobra
1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0 mm
0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	100 mm
0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0 mm
0	0	2	1	1	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	200 mm
7	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0 mm
0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	200 mm
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	100 mm
2	2	0	1	1	0	0	0	0	0	0	0	0	0	0	0 mm
8	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	100 mm
3	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0 mm
5	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0 mm
3	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0 mm
3	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0 mm
0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0 mm
2	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0 mm
1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0 mm
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	400 mm
0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0 mm
0	3	0	2	0	0	0	0	0	0	0	0	0	0	0	100 mm
0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	100 mm
2	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0 mm
2	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0 mm
2	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	2	1	0	0	0	1	0	0	0	0	0	0	0	0	100 mm
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0 mm
3	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0 mm
9	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
5	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0 mm
6	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0 mm
0	3	0	0	0	0	0	1	0	0	0	0	0	0	0	100 mm
0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	100 mm

Quadro 8 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500	Sobra
2	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0 mm
1	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	100 mm
9	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	100 mm
4	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
3	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
3	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0 mm
7	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0 mm
0	2	2	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
4	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0 mm
2	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0 mm
8	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
2	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0 mm
1	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0 mm
1	1	2	0	0	1	0	0	0	0	0	0	0	0	0	0 mm
1	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
3	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0 mm
1	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0 mm
0	4	1	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
1	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0 mm
1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	700 mm
0	1	4	0	0	0	0	0	0	0	0	0	0	0	0	100 mm
0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	100 mm
3	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0 mm
1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0 mm
2	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0 mm
0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0 mm
0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	200 mm
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	500 mm
8	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0 mm
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0 mm
4	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0 mm

Quadro 8 continuação da pagina anterior

100	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500	Sobra
1	0	3	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
2	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0 mm
0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0 mm
0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	100 mm
3	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	200 mm
0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0 mm
0	2	0	0	1	1	0	0	0	0	0	0	0	0	0	0 mm
3	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0 mm
2	4	0	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	200 mm
1	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0 mm
0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	200 mm
0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	100 mm
0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	100 mm
1	3	1	0	1	0	0	0	0	0	0	0	0	0	0	0 mm
3	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0 mm
0	2	1	0	0	0	0	1	0	0	0	0	0	0	0	0 mm
1	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0 mm
5	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0 mm
1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0 mm
4	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0 mm
2	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0 mm

No Quadro 9 estão os resultados das simulações realizadas no experimento E.

Quadro 9 – Quadro com as simulações do experimento E

Simulações	1-5	6-10	11-15	16-20	21-25	26-30	31-35	36-40	41-45	46-50
1^a simulação	1,37%	5,97%	52,82%	98,02%	11,30%	97,55%	93,64%	54,33%	68,34%	83,19%
2^a simulação	-9,62%	3,69%	60,47%	97,96%	21,15%	97,59%	91,06%	73,15%	60,59%	79,76%
3^a simulação	7,71%	5,08%	43,48%	97,99%	11,79%	97,50%	87,89%	73,60%	73,48%	87,97%
4^a simulação	5,59%	6,36%	64,38%	98,02%	16,15%	97,43%	91,80%	62,58%	79,43%	82,17%
5^a simulação	5,85%	3,55%	52,22%	98,02%	15,10%	97,46%	89,88%	68,35%	73,90%	86,19%

APÊNDICE B – ALGORITMOS

O Código-fonte 1 tem a função que gera os padrões de corte ótimo. Ele está escrito em Python e recebe dois parâmetros de entrada: uma lista com o tamanho dos objeto que serão cortados e uma outra com o tamanho dos itens que devem ser cortados, e retorna um dicionario com os padrões de corte e as suas respectivas perdas.

Código-fonte 1 – Função que gera os padrões de corte em Python

```
1
2  def gerador_cortes(comps, modelos):
3
4    dicionario = {}
5      # Dicionario com os padroes de corte
6    for barra in comps:
7      cortes= [(0, 0, 0, 0, 0, 0, 0)]
8
9      num=[]
10     for item in modelos:
11       num.append(np.floor(barra/item))
12         # num recebe o quanto de cada item
13         # cabe em uma barra
14     lista=[range(int(i+1)) for i in num]
15         # Cria um range para cada possibilidade
16     pos= list(itertools.product(*lista))
17         # Todas as combina  es de cortes
18         # limitadas pelo numero maximo de cortes
19
20     for i in pos: # para cada combinacao
21       if np.dot(modelos,i)<=barra:
22             # verifica se ela eh possivel
23         for j in range(np.shape(cortes)[0]):
24             # se for uma combinacao valida compara
25             # com todas as combinacoes selecionadas
```

```
26              # anteriormente
27            if all([i[k]>=cortes[j][k] for k in range(7)]):
28              # checa se essa combinacao contem todas  as
29              # combinacoes de uma barra que
30              # ja estava na lista
31              cortes[j]=tuple(i)
32                  # caso sim substitui na lista
33            elif all([any([i[k]>cortes[p][k] for k in range
                 (7)]) for p in range(np.shape(cortes)[0])]):
34              # checa se essa nova combinacao eh maior
35              # por Pareto que todas as outras
36              # combinacoes que ja estavam na lista
37              cortes.append(i)
38                  # caso sim adiciona na lista
39          cortes=[tuple(l) for l in np.unique(cortes, axis=0)
                .tolist()]
40              # remove as combinacoes nao unicas
41              # e trasforma em tuplet de novo
42
43      dicionario[barra]=[cortes,[barra-np.dot(modelos,k) for
           k in cortes]]
44          # adiciona ao dicionario as possiveis
45          # combinacoes de corte e a sua perda
46
47  return dicionario
```