

LUIS HENRIQUE LEÃO DO NASCIMENTO

Orientador: Marco Antonio Moreira de Carvalho

**UM ALGORITMO HEURÍSTICO APLICADO A
HOMOGENEIZAÇÃO DAS CARACTERÍSTICAS FÍSICAS
DE PRODUTOS**

Ouro Preto
Agosto de 2016

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**UM ALGORITMO HEURÍSTICO APLICADO A
HOMOGENEIZAÇÃO DAS CARACTERÍSTICAS FÍSICAS
DE PRODUTOS**

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

LUIS HENRIQUE LEÃO DO NASCIMENTO

Ouro Preto
Agosto de 2016



UNIVERSIDADE FEDERAL DE OURO PRETO

FOLHA DE APROVAÇÃO

Um Algoritmo Heurístico Aplicado a Homogeneização das Características
Físicas de Produtos

LUIS HENRIQUE LEÃO DO NASCIMENTO

Monografia defendida e aprovada pela banca examinadora constituída por:

Dr. MARCO ANTONIO MOREIRA DE CARVALHO – Orientador
Universidade Federal de Ouro Preto

Dr. ANDRÉ LUÍS SILVA
Universidade Federal de Ouro Preto

Msc. MARCELO LUIZ SILVA
Universidade Federal de Ouro Preto

Ouro Preto, Agosto de 2016

Resumo

A evolução e a disseminação da tecnologia, permitiu que em ambientes industriais, os diferentes processos produtivos sejam confiados a métodos computacionais, os quais têm sido aplicado com eficiência, contribuindo para redução de custos, aumento da produtividade, entre outros. Os processos produtivos devem ser otimizados visando algum critério pré-estabelecido como, por exemplo, a minimização da utilização de matéria prima ou a minimização de mão de obra. Em certas indústrias existe a necessidade que os produtos fabricados possuam uma qualidade satisfatória no que tange as características físicas dos mesmos, para que seu preço de revenda e sua qualidade não sejam afetados. Para atender este critério a linha de produção deve ser planejada e otimizada visando fabricar os produtos de um mesmo lote sem que haja a necessidade de interrupção de produção dos mesmos este problema é conhecido como o *Problema de Minimização de Descontinuidades*(MDP). É apresentado um método simples para a solução deste problema, baseado em um algoritmo clássico existente na teoria do grafos com algumas pequenas modificações. São reportados resultados preliminares obtidos em experimentos iniciais para um conjunto de instâncias da literatura.

Abstract

The development and dissemination of technology allowed that, in industrial environments, the different production processes to be entrusted to computational methods, which have been applied effectively, helping to reduce costs and increase productivity, among others. The production processes must be optimized considering predefined criteria. For example, minimizing the raw material waste or minimizing the use of manpower. In certain industries there is a need that the manufactured products have a satisfactory quality regarding the physical characteristics, so their sale price and quality are not affected. To meet this criterion the production line should be planned and optimized aiming to manufacture the products of the same lot without the need to interrupt production this problem is known as the Minimization of Discontinuities Problem. A simple method for the solution of this problem, based on an existing classical algorithm in graph theory with some minor modifications is presented. We report preliminary results obtained in initial experiments for a set of instances from the literature.

Dedico este trabalho a minha família, namorada e amigos que me apoiaram nesta jornada.

Agradecimentos

Agradeço a todos professores do departamento, em especial ao meu orientador Marco.

Sumário

1	Introdução	1
1.1	Motivação	3
1.2	Objetivos	3
1.3	Organização do Trabalho	4
2	Revisão da Literatura	5
2.1	Trabalhos Práticos	5
2.2	Trabalhos Teóricos	7
2.3	Trabalhos Relacionados	8
3	Fundamentação Teórica	9
3.1	Pré-Processamento por Dominância	11
3.2	Busca em Largura	12
4	Desenvolvimento	14
4.1	Representação Computacional	14
4.2	Busca em Largura aplicada ao MDP	15
4.3	Sequenciamento de Padrões	16
5	Experimentos	19
5.1	Comparação de Resultados	19
6	Plano de Atividades Restantes	21
7	Conclusões	22
	Referências Bibliográficas	26

Lista de Figuras

3.1	Aplicação da BFS em um grafo.	13
4.1	Grafo correspondente a instância anterior.	15
4.2	Aplicação da BFS modificada no grafo de exemplo.	17
5.1	Gráfico de comparação de resultados.	20

Lista de Tabelas

3.1	Instância MDP.	10
3.2	Exemplo de nova instância após a aplicação do pré-processamento por dominância.	12
4.1	Instância MDP.	15
4.2	Sequenciamento de padrões a partir da BFS.	18
4.3	Instância MDP depois do sequenciamento.	18
5.1	Comparação de resultados.	20
6.1	Cronograma de atividades para monografia II.	21
7.1	Instâncias do grupo A.	23
7.2	Instâncias do grupo B.	23
7.3	Instâncias do grupo C.	23
7.4	Instâncias do grupo D.	24
7.5	Instâncias do grupo E.	24
7.6	Instâncias do grupo F.	24
7.7	Instâncias do grupo G.	24
7.8	Instâncias do grupo H.	24
7.9	Instâncias do grupo I.	25

Capítulo 1

Introdução

O advento da Revolução Industrial na segunda metade do século XVIII, proporcionou um grande avanço tecnológico em todo o mundo, principalmente no contexto industrial onde ocorreu a substituição em larga escala de processos manuais por processos automatizados ou semi-automatizados. Esse acontecimento possibilitou a otimização da mão de obra, do tempo de trabalho e do uso de matéria prima, com o intuito de aumentar a produtividade e consequentemente a lucratividade a partir do comércio dos bens produzidos.

Nesse contexto, atualmente, a inteligência computacional e seus métodos têm sido utilizados nas indústrias, basicamente em todas as fases da produção, desde a obtenção da matéria-prima até a entrega do produto acabado ao consumidor final. Etapas intermediárias incluem o planejamento da produção, o processamento de matéria prima, a estocagem e a manipulação de produtos.

Em algumas indústrias há a necessidade de transformar unidades maiores de matéria prima em produtos menores com diferentes formas e tamanhos. Este processo pode ser caracterizado como um Problema de Corte, que possui como objetivo realizar cortes em uma determinada unidade de matéria prima a fim de se obter unidades menores (ou *peças*), respeitando algum objetivo pré-estabelecido, como por exemplo, maximizar a quantidade de unidades menores cortadas, ou minimizar a sobra de matéria-prima utilizada. Neste trabalho, considera-se que estas unidades maiores de matéria prima são chapas, por exemplo, de metal, vidro, papel ou madeira.

Um plano (ou *padrão*) de corte pode ser caracterizado como a disposição das peças dentro de uma chapa de matéria prima, semelhante a um gabarito. Cada padrão é unicamente reconhecido por possuir uma quantidade finita de peças e para cada uma há uma posição associada. A disposição de cada peça dentro destes padrões, interfere diretamente nos diferentes objetivos dos problemas de corte.

Para atender a demanda de toda a linha de produção, ou seja, das peças que precisam ser produzidas, é preciso realizar o corte dos diferentes padrões várias vezes. Desta forma, caracteriza-se o Problema de Corte de Estoque, amplamente estudado na literatura e que

geralmente tem como objetivo minimizar a quantidade de padrões a serem cortados.

Um *estágio* pode ser definido como uma subdivisão de um processo em uma sequência de sub-processos. No meio industrial um estágio da produção pode ser caracterizado como uma unidade de tempo em que determinado padrão de corte é processado. A ordem em que os padrões são processados, pode interferir diretamente na produtividade, ocasionando custos desnecessários. É preciso então planejar a sequência em que os padrões serão cortados atendendo algum critério de qualidade, como por exemplo:

- Produzir peças de um mesmo lote que possuam características físicas semelhantes;
- Minimizar o estoque intermediário;
- Minimizar o número de interrupções na produção para a realização de ajustes na máquina de produção.

Cada diferente critério estabelece um problema de otimização diferente, definindo assim os Problemas de Sequenciamento de Padrões.

No contexto do tema abordado neste trabalho, é interessante minimizar o número de vezes em que o processamento de uma peça qualquer é interrompido, ou seja, o número de *descontinuidades* em sua produção. Uma descontinuidade surge quando uma determinada peça está sendo processada a partir de um padrão em um determinado estágio, e em nos estágios posteriores a referida peça não é produzida, porém, volta a sê-lo em algum estágio posterior.

Uma descontinuidade na produção de uma determinada peça pode ser ocasionada pela necessidade de produção de outro tipo de peça. Esse fato pode ocasionar uma pausa na linha de produção para a troca da cor da tinta utilizada pela máquina ou a troca de lâminas de corte, por exemplo. Ao fabricar peças diferentes, a mesma matéria prima é utilizada e, posteriormente, ao retomar a produção de um tipo de peça específica, as características da matéria prima podem ser diferentes, originando variações de características físicas.

Em nichos específicos de mercado, as características físicas dos produtos são um fator determinante no seu valor de revenda e aceitação pelos consumidores. Diferentes métricas podem ser utilizadas para inspecionar a qualidade de um produto, porém, uma que se destaca particularmente nas indústrias de processamento de madeira, vidro e cerâmica é a homogeneidade das características físicas, tais como textura, cor, padronagem e tonalidade.

Nestas indústrias, diferentes peças são utilizadas para a montagem de um único bem de consumo, como móveis, ou usadas em conjunto, principalmente na construção civil, como placas de revestimento e vidros. A introdução de variações acentuadas nestas características nos produtos de um mesmo lote pode causar uma diminuição considerável no valor de revenda dos mesmos. Ou seja, produtos pertencentes ao mesmo lote devem ser o mais parecidos possíveis, para que o seu preço de venda não seja afetado pelas variações acentuadas nas

características físicas. Para atender o critério anteriormente citado é preciso sequenciar os padrões visando diminuir o número de descontinuidades presentes, caracterizando assim o *Problema de Minimização de Descontinuidades*(MDP).

1.1 Motivação

Há duas motivações principais para esse trabalho. Primeiramente, pela sua relevância teórica, em que o trabalho apresentado pertence a classe NP-Difícil. Segundo, por se tratar de um problema de aplicação prática no contexto industrial, tornando-se essencial para o planejamento e execução do processo produtivo de diversas indústrias, tais como as relacionadas a produtos de madeira, vidro, papel e cimentícios. Ainda, o tema é relevante nas áreas de movelaria e construção civil, com aplicações práticas diretas na engenharia e indústria. A criação de novos modelos para solução do problema em questão pode contribuir para melhoria da eficiência da indústria nacional.

1.2 Objetivos

De maneira geral, esse trabalho consiste em desenvolver um método de inteligência computacional para a resolução do MDP. A partir do objetivo principal, temos os seguintes objetivos específicos:

- Realizar pesquisa para geração de embasamento teórico e revisão bibliográfica sobre MDP;
- Elaborar uma heurística consistente que possa ser utilizada no problema e que permita a obtenção rápida de soluções próximas da solução ótima sem que se perca a vantagem da busca sistemática;
- Pesquisar técnicas para melhoria fina de produções;
- Avaliar o método implementado considerando dados reais e também com problemas teste publicamente disponíveis, realizando uma análise crítica considerando outros métodos da literatura;
- Buscar a aplicação prática dos métodos desenvolvidos em contextos reais, a fim de que também seja constituído um avanço para as indústrias nacionais.

Outros produtos deste projeto de pesquisa serão trabalhos publicados em eventos nacionais, os quais contribuem para a promoção da Universidade Federal de Ouro Preto e também do tema tratado

1.3 Organização do Trabalho

Nas próximas seções o trabalho é detalhado. O Capítulo 2 apresenta a revisão da literatura de 1974 até 2015. A base conceitual do MDP é apresentada no Capítulo 3. No Capítulo 4 é apresentado uma implementação inicial do método proposto. Os experimentos utilizados são apresentados no Capítulo 5. No Capítulo 6 é apresentado o plano para conclusão do trabalho de pesquisa. As conclusões sobre esta primeira parte do trabalho são apresentadas no Capítulo 7.

Capítulo 2

Revisão da Literatura

Nesse capítulo, os principais trabalhos sobre o Problema de Minimização de Descontinuidades são descritos brevemente. Os trabalhos são divididos em:

- Práticos: propõem heurísticas e métodos exatos para a solução do problema;
- Teóricos: apresentam a complexidade do problema e o estudam o relacionamento entre os problemas de sequenciamento de padrões;
- Relacionados: apresentam certa similaridade com o problema tratado neste artigo mas são utilizados para a resolução de outros problemas.

2.1 Trabalhos Práticos

O primeiro trabalho a abordar o MDP foi o realizado por Dyson e Gregory (1974), integrado ao problema de corte de estoque. Inicialmente, foi utilizado o método simplex revisado para determinar a composição dos padrões de corte para atender toda a demanda por peças com o menor desperdício de matéria-prima possível. A fase de sequenciamento de padrões foi modelada como o Problema do Caixeiro Viajante, em que os padrões são as vértices do grafo, havendo arestas entre todos os pares de vértices. As distâncias foram calculadas levando em consideração a quantidade de peças que não são comuns a cada par de padrões. Entretanto, essa modelagem foi descartada, principalmente devido a limitação de desempenho computacional da época para solução do Problema do Caixeiro Viajante.

Como alternativa, os autores propuseram um método que resolve o problema de corte de estoque utilizando *branch-and-bound* e cria o sequenciamento dos padrões na mesma ordem que são gerados. Para isto foi utilizada a seguinte metodologia: em cada interação o *branch-and-bound* resolve o problema de corte de estoque considerando todos os padrões restantes a fim de determinar o menor desperdício, sendo que determinadas peças têm o valor de sua área aumentado por um parâmetro para de serem escolhidas primeiro. Esse parâmetro é calculado

para todas as peças que já foram processadas em algum padrão anterior e que ainda precisam ser cortadas para produzir a quantidade mínima para atender a demanda da indústria.

Os resultados obtidos mostraram que a heurística proposta comparada com cálculos manuais apresentou pouca melhoria em uma indústria de produção de vidro.

Madsen (1979), primeiramente resolve o Problema de Corte de Estoque para calcular o mínimo de desperdício possível. Para a resolução do MDP foi proposta uma abordagem utilizada criando uma matriz C binária, quadrada e simétrica para representar quais padrões possuem pelo menos um par de peças em comum. Cada entrada c_{ij} da matriz é preenchida da seguinte maneira:

$$c_{ij} = \begin{cases} 1, & \text{se o padrão } i \text{ possui uma peça em comum com o padrão } j \\ 0, & \text{caso contrário} \end{cases}$$

Foi aplicado então a heurística de *Cuthill-McKee* (Cuthill e McKee, 1969) para o problema de Minimização de Largura de Banda em Matrizes Simétricas. Ao comparar o método proposto com a solução usada na empresa, houve uma significativa melhora no desperdício. Não houve relatos de melhoria das descontinuidades na empresa.

Madsen (1988), foi o primeiro artigo a focar somente no sequenciamento de padrões, visando diminuir o espalhamento de ordens de compras de um conjunto de clientes. O *Problema de Minimização de Espalhamento de Ordens* (*Minimization of Order Spread Problem* – MORP) tem como objetivo atender a ordens de compra dos clientes o mais rápido possível, de maneira que uma vez iniciada a produção das peças de uma determinada ordem, todas as outras peças relacionadas sejam produzidas imediatamente, visando minimizar a estocagem intermediária dos produtos.

Foi utilizada a formulação do Problema do Caixeiro Viajante proposta por (Dyson e Gregory, 1974) para o MDP, para resolver o problema de estoque intermediário de uma pequena empresa de vidro em um intervalo de 14 dias, para isso foi proposto uma heurística de três estágios.

Primeiramente foi utilizado o método proposto por Gilmore e Gomory (1961), para resolver o Problema de Corte em Estoque sem levar em consideração o espalhamento de ordens.

No seguinte estágio foi proposta uma matriz C de dimensão $n \times n$, em que n é a quantidade de padrões existentes, da seguinte forma:

- Inicialmente foi adicionado um padrão fictício, com nenhuma peça de corte, para que seja possível o aspecto circular do Problema do Caixeiro Viajante;
- A diagonal principal foi preenchida com um valor alto (10.000);
- Para todas as ligações entre o padrão fictício a outro padrão qualquer foi atribuído o valor 1.000;

- Para padrões que possuem apenas peças singulares, ou seja, peças que não estão contidas em outros padrões foi atribuído o valor 110;
- Para as posições restantes, os padrões que possuem n peças em comum o valor de $c_{i,j}$ é definido por $100 - 10 \times n$.

Os pesos foram atribuídos dessa maneira, para que ao solucionar o Caixeiro Viajante os padrões que possuem mais peças em comuns sejam cortados o mais próximo possível. Os padrões compostos apenas por peças singulares serão os últimos a serem processados, pois não existe descontinuidades em nenhuma de suas peças.

Para solução do Problema do Caixeiro Viajante foi utilizada a heurística *3-opt*. Este é um método de busca local em que três arestas presentes na solução $(k_1, k_2), (j_1, j_2)$ e (i_1, i_2) são trocadas por outras três novas arestas presentes no grafo e não presentes na solução, a fim de se obter uma melhoria na solução atual. Se existir alguma nova configuração melhor que a anterior, mantém-se essa solução. Caso contrário, escolhe-se novamente outras três arestas para análise.

Comparando os resultados obtidos a partir do primeiro estágio com a solução do Caixeiro Viajante, surpreendente os melhores resultados obtidos foram na minimização das descontinuidades, na qual foi reportada uma melhoria média de 30%. Já para o estoque intermediário ocorreu uma diminuição média de 18% entre os resultados.

2.2 Trabalhos Teóricos

Com o nome de *Consecutive Blocks Minimization*, Garey e Johnson (1979) mostraram que o MDP pertence a classe NP-Difícil, ou seja, não existe algoritmo conhecido que resolva este problema em tempo determinístico polinomial. Dom (2009) mostra que este problema aparece em várias aplicações práticas como:

- Escalonamento de Tarefas;
- Recuperação de Informação;
- Otimização de Linhas Férreas;
- Bioinformática;
- Arqueologia;
- Resolução de Sistemas Lineares;

Linhares e Yanasse (2002), apresentaram um estudo detalhado sobre o relacionamento dos diferentes Problemas de Sequenciamento de Padrões, mostrando que, apesar de possuírem mesma base conceitual, não são equivalentes entre si.

2.3 Trabalhos Relacionados

Recentemente, Haddadi et al. (2015), utilizando o *Consecutive Blocks Minimization* propôs dois métodos heurísticos para a redução da dimensão de sistemas lineares. Para a resolução do problema, primeiramente foi proposta uma matriz binária em que as linhas correspondem o número de equações presentes no sistema, e as colunas correspondem ao números de incógnitas do problema. É atribuído o valor não nulo para todas as posições em que uma incógnita qualquer do sistema está multiplicada por um coeficiente diferente de 0.

Através da matriz resultante é utilizado método *2-opt* (método 1), que consiste na troca de duas colunas quaisquer para tentar melhorar a solução e um método de *shift* (método 2), que consiste em colocar uma coluna em uma determinada posição e deslocar todas as outras uma posição para a direita. Foi proposto uma heurística que utiliza os dois métodos simultaneamente. Primeiramente o método 2 é aplicado até que não ocorra mais melhoria na solução, após o método 1 é aplicado de maneira semelhante.

Para a realização dos testes foram utilizadas um conjunto de instâncias geradas pelo próprio autor e outras instâncias reais de um problema equivalente (relacionado estações de trem). Estas instâncias foram divididas em categorias conforme suas dimensões e densidades. As instâncias foram fornecidas gentilmente pelo autor do artigo, para uma posterior comparação com o método proposto neste trabalho.

Comparando os resultados obtidos, o método 2 foi superior ao método 1 em todas as instâncias mostradas pelo autor quando aplicados individualmente, tanto no quesito do valor da solução quanto no tempo de execução dos algoritmos. Os resultados obtidos a partir da aplicação da heurística para todas as instâncias mostrou uma melhora em relação ao valor inicial das mesmas. Para as instâncias reais não foi possível comparar os resultados obtidos com os resultados originais, pois estes dados não foram disponibilizados.

Capítulo 3

Fundamentação Teórica

O Problema de Minimização de Descontinuidades (MDP) origina-se em um ambiente de produção industrial, em que é necessário realizar o corte de um conjunto S de diferentes padrões para produzir um conjunto finito de peças P para atender uma determinada demanda. A produção é considerada sequencial e incremental, ou seja, em um estágio todas as cópias de um padrão de corte específico devem ser cortadas antes que um padrão de corte diferente seja processado.

Durante o processo produtivo, uma descontinuidade ocorre na produção de uma peça quando um padrão que contém a referida peça é processado em um determinado estágio e no estágio seguinte nenhuma cópia da peça é produzida, sendo que outras cópias serão produzidas em estágios posteriores, caracterizando assim uma interrupção na produção daquela peça. Ao fabricar peças diferentes, a mesma matéria prima é utilizada e, posteriormente, ao retomar a produção deste tipo de peça específica, as características da matéria prima podem ser diferentes, originando variações de características físicas. Sendo assim, descontinuidades na produção de peças podem ocasionar variações de textura, cor, padronagem e tonalidade em produtos de um mesmo lote. É preciso então um planejamento da linha de produção que a equilibre e, se necessário, priorize a minimização do número de descontinuidades, independente da duração das mesmas.

Uma instância MDP é representada por uma matriz M binária que relaciona os padrões de corte e as peças. Cada elemento m_{ij} ($i \in P, j \in S$) da matriz M é preenchida da seguinte forma:

$$m_{ij} = \begin{cases} 1, & \text{se o padrão } p_j \text{ possui a peça } i \\ 0, & \text{caso contrário} \end{cases}$$

A Tabela 3.1 apresenta um exemplo de instância MDP. O conjunto de peças, enumeradas de 1 a 6, está representado na horizontal, e o conjunto dos padrões de corte, enumerados de p_1 a p_6 , está representado na vertical.

Tabela 3.1: Instância MDP.

	p_1	p_2	p_3	p_4	p_5	p_6
1	1	1	0	0	0	0
2	1	0	1	0	0	0
3	0	0	0	1	1	0
4	0	0	0	1	0	1
5	0	1	0	0	1	0
6	0	0	1	0	0	1

No referido exemplo, o padrão p_1 é composto pelas peças 1, 2, e o padrão p_2 é composto pelas peças 1, 5 e assim por diante.

Uma solução para o MDP é dada por uma permutação π das colunas da matriz M , dando origem à uma Matriz Q^π , que consiste nos mesmos elementos da Matriz M , porém, com as colunas permutadas. Essa representação indica qual padrão de corte será processado em cada estágio da produção. A Tabela 3 apresenta dois possíveis sequenciamentos dos padrões com base na instância definida na Tabela 3.1. Na referida Tabela, para a primeira solução existe descontinuidades na produção das peças 1 e 3. Já para a segunda solução, há descontinuidades na produção das peças 1, 2, 4, 5 e 6.

No primeiro sequenciamento, definido na Tabela 3(a), temos $\pi = [p_5, p_2, p_4, p_6, p_3, p_1]$, o que indica que no primeiro estágio serão processadas as cópias do padrão p_5 , no segundo estágio serão processadas as cópias do padrão p_2 e assim sucessivamente até que no último estágio sejam processadas as cópias do padrão p_1 . O segundo sequenciamento, na Tabela 3(b), apresenta uma solução diferente, com um sequenciamento de padrões diferente, $\pi = [p_1, p_6, p_5, p_4, p_3, p_2]$. Os valores em negrito presentes na Tabela 3 indicam em quais estágios da produção ocorreu descontinuidade de uma determinada peça.

	p_5	p_2	p_4	p_6	p_3	p_1
1	0	1	0	0	0	1
2	0	0	0	0	1	1
3	1	0	1	0	0	0
4	0	0	1	1	0	0
5	1	1	0	0	0	0
6	0	0	0	1	1	0

(a)

	p_1	p_6	p_5	p_4	p_3	p_2
1	1	0	0	0	0	1
2	1	0	0	0	1	0
3	0	0	1	1	0	0
4	0	1	0	1	0	0
5	0	0	1	0	0	1
6	0	1	0	0	1	0

(b)

Uma maneira aproximada de determinar o número de descontinuidades na matriz Q^π é determinar o número de inversões de 0 para 1 em cada linha da mesma matriz. Embora o número de inversões seja maior do que o número de descontinuidades, ambos são proporcionais. Tendo definido a matriz Q^π , a quantidade de descontinuidades presentes é calculada pela Função 3.1.

$$Z_{MDP}^{\pi}(Q^{\pi}) = \sum_{j=1}^J \sum_{i=1}^I q_{ij}^{\pi} (1 - q_{ij-1}^{\pi}) \quad (3.1)$$

No sequenciamento ilustrado na Tabela (a), existem duas descontinuidades, uma para a peça 1 com duração de 3 estágios, e outra para a peça 3 com duração de 1 estágio. No segundo sequenciamento (Tabela (b)) há descontinuidades na produção das peças 1, 2, 4, 5 e 6 respectivamente por 4, 3, 1, 2 e 2 estágios. Claramente, o sequenciamento apresentado na primeira solução é melhor do ponto de vista de minimização de descontinuidades, calculada de acordo com a Função 3.1. O primeiro sequenciamento apresenta 7 inversões e o segundo um total de 11 inversões.

A partir da análise da Função 3.1 é possível definir a função objetivo do Problema de Minimização de Descontinuidades conforme mostrado na Função 3.2, que visa determinar uma permutação π das colunas da matriz M (dentre o conjunto de todas as permutações possíveis Π) que minimize o número máximo de descontinuidades.

$$\min_{\pi \in \Pi} Z_{MDP}^{\pi}(M) \quad (3.2)$$

3.1 Pré-Processamento por Dominância

O pré-processamento por dominância é uma técnica que consiste na remoção de padrões que possuem em sua composição todas as peças existentes em algum outro padrão do problema, a fim de eliminar redundâncias. Um padrão p_i que possui em sua composição um subconjunto de peças que engloba todas as peças de um padrão p_j é denominado como padrão *dominante*, ao passo que o padrão p_i é dito *dominado*.

Os padrões dominados podem ser removidos da entrada, diminuindo assim o tamanho do problema tratado. A partir da solução final proposta por algum método, os padrões dominados são sequenciados imediatamente após seus respectivos padrões dominantes em π , sem perda de otimalidade. A Tabela 3.2 apresenta um exemplo de instância antes (a) e após (b) da aplicação do pré-processamento por dominância entre padrões.

Analisando a Tabela 3.2(a) verifica-se que a composição do padrão p_3 , formado pelas peças 3 e 6 (ressaltadas em negrito) é um subconjunto das peças que compõem o padrão p_1 , formado pelas peças 1, 3, 4, 6 e 7. De maneira análoga, o padrão p_4 , formado pelas peças 1, 3 e 5 (também ressaltadas em negrito) consiste em um subconjunto das peças que compõem o padrão p_5 , formado pelas peças 1, 3, 4, 5 e 7. Dessa forma, o pré-processamento por dominância entre padrões elimina os padrões p_3 e p_4 da entrada, resultando em uma instância reduzida, conforme a Tabela 3.2(b). Em uma possível solução, os padrões p_1 e p_3 devem obrigatoriamente serem sequenciados consecutivamente, assim como os padrões p_5 e p_4 , sem causar qualquer prejuízo a solução do problema.

Tabela 3.2: Exemplo de nova instância após a aplicação do pré-processamento por dominância.

	p_1	p_2	p_3	p_4	p_5	p_6
1	1	1	0	1	1	0
2	0	1	0	0	0	1
3	1	1	1	1	1	1
4	1	0	0	0	1	1
5	0	0	0	1	1	1
6	1	0	1	0	0	0
7	1	1	0	0	1	1

(a)

	p_1p_3	p_2	p_5p_4	p_6
1	1	1	1	0
2	0	1	0	1
3	1	1	1	1
4	1	0	1	1
5	0	0	1	1
6	1	0	0	1
7	1	1	1	1

(b)

3.2 Busca em Largura

A Busca em Largura (*Breadth-First Search* - *BFS*) (Cormen, 2009) é um algoritmo de busca em grafos que, dado um vértice inicial, explora toda a vizinhança deste vértice. Durante a exploração, os vértices são visitados e inseridos em uma estrutura de fila. Terminada essa exploração, o primeiro vértice da fila é removido da mesma e é selecionado como novo vértice inicial e repete-se esse processo até que todo o grafo seja explorado, não visitando um vértice mais de uma vez. Caso exista mais de um componente, a busca em largura examina todo o componente, e logo após, um vértice presente em outro componente é selecionado para reiniciar a busca. Dessa maneira a busca em largura retorna uma lista com a ordem em que todos os vértices presentes no grafo foram explorados.

A Figura 3.1 apresenta um exemplo da BFS aplicada a uma grafo de exemplo mostrado na Figura 3.1a. O vértice 1 foi escolhido como inicial escolhido (em vermelho), como visto na Figura 3.1b. A partir desse vértice todos os vizinhos são visitados (em verde) vide Figura 3.1c e inseridos na fila f na ordem $f=[2, 3, 4]$. Os vértices 2 e 3 não possuem vizinhos que não tenham sido explorados ainda, portanto, são removidos de f e o próximo vértice a ter sua vizinhança explorada é o 4. O único vértice vizinho que não foi visitado é o vértice 5 (em azul) como mostrado na Figura 3.1d, portanto, este é inserido ao final de f . Todos os vértices de um mesmo componente foram visitados, portanto a f está vazia e a busca recomeça pelo vértice 6 (em amarelo), vide Figura 3.1e. Como não existem mais vértices não visitados, o algoritmo termina e retorna a sequência $[1, 2, 3, 4, 5, 6]$ em que os vértices foram percorridos.

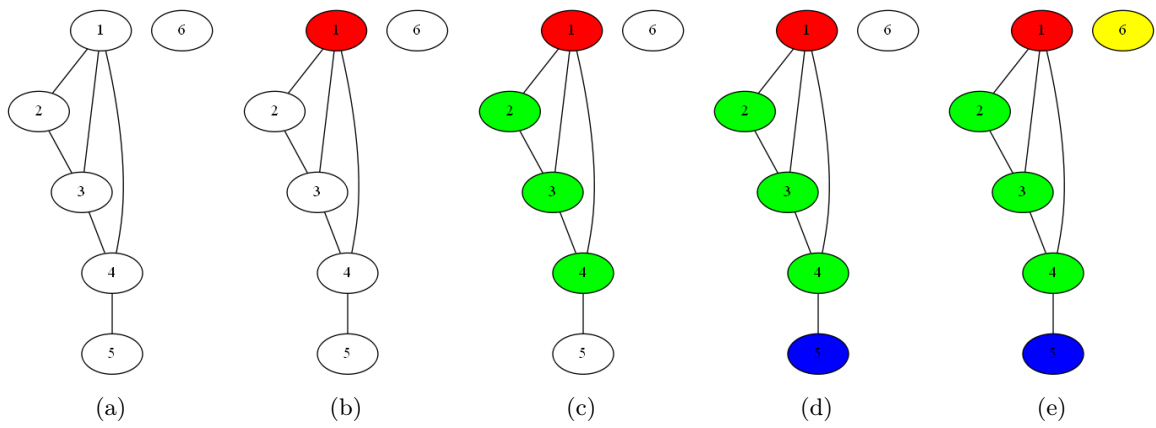


Figura 3.1: Aplicação da BFS em um grafo.

Capítulo 4

Desenvolvimento

Nesse capítulo é descrita com detalhes a heurística proposta e sua representação computacional, além de exemplos a fim de facilitar o entendimento dos métodos utilizados.

A heurística tem como idéia a representação do MDP na forma de um grafo e então a utilização de um algoritmo básico de percurso no mesmo, a fim de diminuir o número de discontinuidades.

4.1 Representação Computacional

As informações de entrada para o MDP são dados referentes à composição de cada padrão de corte e de cada peça. Essas informações são representadas em uma matriz binária semelhante a matriz descrita na Seção 3.

Para a representação do problema, a partir de uma instância é construído um grafo ponderado e não direcionado como descrito em (Yanasse, 1997), em que os vértices representam as peças, havendo ligação entre dois vértices quaisquer se as peças estão presentes em um mesmo padrão. A cada ocorrência desta característica, o peso da aresta correspondente é aumentado em uma unidade.

O grafo é construído da maneira citada anteriormente visando coletar a partir das peças, informações sobre os padrões. Peças que aparecem com frequência juntas em diferentes padrões, terão adjacência no grafo com um peso elevado. Portanto, os padrões que contém determinadas peças devem ser processados em sequência.

A Tabela 4.1 apresenta uma instância MDP de exemplo e a Figura 4.1 apresenta o grafo correspondente.

Inicialmente, o grafo é composto pelos cinco vértices que representam as cinco peças do problema. O padrão p_1 é formado pelas peças 1, 2 e 5, resultando na adjacência dos vértices que representam as peças do referido padrão. O padrão p_2 é composto pelas peças 1 e 5. Essa aresta já está presente no grafo, portanto, o peso da mesma é aumentado para duas unidades.

Tabela 4.1: Instância MDP.

	p_1	p_2	p_3	p_4	p_5	p_6
1	1	0	0	0	0	1
2	1	0	1	0	0	1
3	0	0	0	1	1	0
4	0	0	0	1	1	0
5	1	1	1	1	1	1

Os padrões restantes passam pelos mesmos processos citados anteriormente, até ser obtido o referido grafo final.

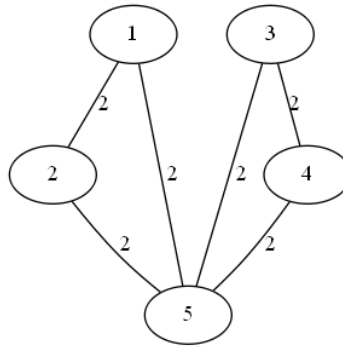


Figura 4.1: Grafo correspondente a instância anterior.

4.2 Busca em Largura aplicada ao MDP

A heurística aplicada ao MDP utiliza a BFS com algumas modificações. Primeiramente o vértice inicial é aquele que possui o menor grau, ou seja, aquele que possui menos adjacências com os demais vértices. Caso ocorra empate, o vértice inicial é selecionado de acordo com a ordem lexicográfica. A partir dos vizinhos do vértice inicial o próximo vértice a ser expandido é aquele que possuir a aresta de maior peso. De maneira análoga, caso ocorra empate é selecionado o vértice de menor ordem lexicográfica. Repete-se esse passos até que todos os vértices sejam visitados, retornando uma lista ϕ com a ordem de exploração dos vértices.

A BFS foi escolhida por ser um algoritmo simples para se percorrer um grafo de acordo com as adjacências de cada vértice (ou seja, aproveitando a vantagem da modelagem em grafos específica para este problema) e pela sua complexidade computacional. Como cada vértice entra na fila apenas uma vez e que lista de adjacências de cada vértice é examinada apenas uma vez a complexidade será $O(n + m)$, em que n representa o número de vértices e m representa o número de arestas.

O algoritmo 1 apresenta o pseudocódigo para o sequenciamento das peças. A entrada consiste do grafo G obtido a partir da instância de entrada. A função $f(G)$ retorna o vértice

que possui o maior grau entre todos presentes no grafo, a cada instante é verificado todos os vizinhos não explorados e adicionados na lista Q descendentemente pelo peso de sua aresta e o primeiro vértice é adicionado na lista ϕ de peças.

Algoritmo 1: Pseudocódigo da BFS modificada.

```

1  Entrada: Grafo de peças  $G$ .
2   $v \leftarrow f(G)$ ;
3  Marque  $v$  como visitado;
4   $\phi \leftarrow \emptyset$ ;
5  Fila  $Q \leftarrow v$ ;
6  enquanto  $Q \neq \emptyset$  faça
7       $T \leftarrow \emptyset$ ;
8      para todo vértice  $w$  vizinho de  $v$  faça
9          se  $w$  é marcado como não explorado então
10             Insira  $w$  ao final de  $T$ ;
11             Marque  $w$  como explorado;
12         fim
13     fim
14     Ordena  $T$  descendentemente pelo peso da aresta  $\{v, w\}$ ;
15     Insira todos os elementos de  $T$  no final de  $Q$ ;
16      $\phi \leftarrow v$ ;
17     Remova  $v$  de  $Q$ ;
18 fim

```

A Figura 4.2 apresenta a BFS modificada aplicada ao grafo da Figura 4.1. Inicialmente, nenhum vértice foi explorado, portanto a lista ϕ é vazia (Figura 4.2a). É preciso determinar o vértice inicial, porém todos vértices possuem o mesmo grau, então por ordem lexicográfica o vértice 1 é selecionado (em vermelho) e adicionado à lista ϕ conforme a Figura 4.2b. Os vértices 2 e 5 (em verde), vizinhos do vértice 1 são visitados e consequentemente colocados em ordem lexicográfica em ϕ , dado que possuem o mesmo peso nas arestas, vide Figura 4.2c. Por fim o vértice 5 é selecionado e seus vizinhos (em azul) são visitados e colocados na lista ϕ , vide Figura 4.2d. Ao final, temos $\phi = [1, 2, 5, 3, 4]$.

4.3 Sequenciamento de Padrões

A lista ϕ gerada pela BFS retorna somente uma ordenação das peças, entretanto uma solução viável para o MDP consiste na ordem de processamento dos padrões. É preciso então obter uma permutação π das colunas a partir da lista de peças ϕ . O método para este fim foi proposto por (Becceneri, 1999) e consiste em percorrer toda a lista de peças gerada

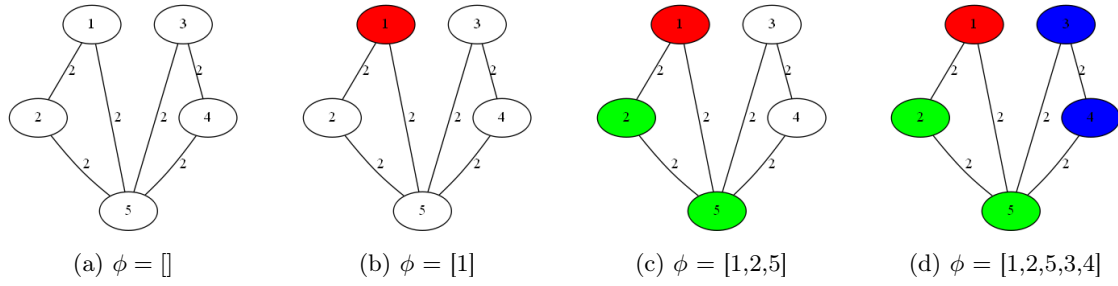


Figura 4.2: Aplicação da BFS modificada no grafo de exemplo.

pela BFS de forma incremental, ou seja, uma peça analisada por vez, verificando se algum padrão possui unicamente a peça corrente ou qualquer subconjunto da mesma com as peças analisadas anteriormente. Todos os padrões que atendem o critério descrito anteriormente são adicionados ao final de π , sem repetição. Caso mais que um padrão atenda ao critério de inserção na solução, o desempate é realizado em ordem lexicográfica.

O Algoritmo 2 apresenta o pseudocódigo para o sequenciamento dos padrões. A entrada consiste na lista de peças ϕ obtida a partir do sequenciamento das peças obtido pela BFS modificada. A função $c(p_i)$ retorna quais peças compõem o padrão p_i , e, a cada instante, o conjunto A de peças, representa as peças disponíveis a cada iteração do algoritmo.

Algoritmo 2: Pseudocódigo do sequenciamento de padrões

```

1 Entrada: Lista de peças  $\phi$ .
2  $A \leftarrow \emptyset$ ;
3 para cada peça  $i \in \phi$  faça
4    $A \leftarrow A \cup i$ ;
5   se  $c(p_i) \subseteq A$  então
6      $\mid$  Insira  $p_i$  ao final de  $\pi$ ;
7   fim
8 fim
```

A Tabela 4.2 apresenta o processo de sequenciamento de padrões dada a sequência de peças $\phi = [1, 2, 5, 3, 4]$ gerada no exemplo anterior. Na referida tabela, cada linha indica quais peças já foram analisadas. Primeiramente é verificado que nenhum padrão possui somente a peça $\{1\}$. Na segunda iteração, novamente nenhum padrão possui somente a peça $\{1\}$, $\{2\}$ ou $\{1, 2\}$. Ao analisar a peça $\{5\}$ é verificado que os padrões p_1, p_2, p_5 são compostos em sua totalidade por algum subconjunto das peças analisadas até então. Desta forma, os padrões são inseridos em ordem lexicográfica na solução. Na terceira iteração, a peça $\{3\}$ e todos os subconjuntos de peças analisadas são verificados e os padrões p_4 e p_5 são inseridos na solução final. Como todos os padrões já foram verificados o algoritmo termina e retorna a ordem em que os padrões serão processados.

Tabela 4.2: Sequenciamento de padrões a partir da BFS.

ϕ	π
1	
1,2	
1,2,5	p_1, p_2, p_3, p_6
1,2,5,3	$p_1, p_2, p_3, p_6, p_4, p_5$
1,2,5,3,4	$p_1, p_2, p_3, p_6, p_4, p_5$

A tabela 4.3 apresenta a matriz resultante a partir da $\pi = [p_1, p_2, p_3, p_6, p_4, p_5]$, havendo na mesma duas descontinuidades: na peça 1 e na peça 2.

Tabela 4.3: Instância MDP depois do sequenciamento.

	p_1	p_2	p_3	p_6	p_4	p_5
1	1	1	0	1	0	0
2	1	0	1	1	0	0
3	0	0	0	0	1	0
4	0	0	0	0	1	1
5	1	1	1	1	1	1

Capítulo 5

Experimentos

Os experimentos computacionais foram realizados em um computador *Intel core i5* de 1.7 GHz com 4GB RAM sob o sistema operacional Windows 10. O código foi implementado em C++, compilado com g++ 4.4.1 e opção de otimização -O3.

São consideradas as instâncias propostas por Haddadi et al. (2015). Este conjunto de quarenta e cinco instâncias artificiais é dividido em nove grupos (*A-I*), com cinco instâncias cada. Cada instância foi gerada aleatoriamente atendendo o critério de que cada coluna possua pelo menos um valor diferente a zero, e cada linha possua pelo menos dois valores iguais a um. Estas instâncias foram originalmente utilizadas para o Problema de Minimização de Blocos Consecutivos, um problema correlato ao Problema de Minimização de Descontinuidades, conforme discutido no Capítulo 2. Desta forma, os resultados são apresentados para ambos os objetivos.

5.1 Comparação de Resultados

Na Tabela 5.1 é apresentada uma comparação preliminar entre o método proposto neste trabalho e os resultados obtidos por Haddadi et al. (2015), que disponibilizou as instâncias. A coluna \overline{B} apresenta a quantidade média de blocos iniciais. O valor médio para o número de blocos obtida pelo trabalho tomado como referência para cada conjunto de instância é mostrado na coluna *Haddadi*. Para a BFS modificada são apresentados o número médio de blocos finais na coluna *BFS-Blocos* e o número médio de descontinuidades na coluna *BFS-MDP*. A coluna *gap* apresenta a distância percentual entre os resultados obtidos e os de referência, calculada como $100 \times (\text{valor obtido} - \text{valor de referência}) / \text{valor de referência}$.

Analisando a Tabela 5.1 e a Figura 5.1 é possível verificar que a BFS modificada conseguiu reduzir em 20,67% o número médio de blocos consecutivos. O grupo de instâncias que obteve maior redução foi o grupo *G*, com diminuição de 41,18%, ao passo que o grupo *C* apresentou a menor redução, equivalente à 8,77% . Em relação ao método de referência, o *gap* médio foi de 18,40% variando de 11,73% no grupo *I* até 23,89% no grupo *F*. Os dados reportados

Tabela 5.1: Comparação de resultados.

Grupo	\bar{B}	<i>Haddadi</i>	<i>BFS-Blocos</i>	<i>BFS-MDP</i>	<i>gap</i>
<i>A</i>	416,0	253,0	298,2	198,2	18,10%
<i>B</i>	955,6	695,8	816,6	716,6	16,93%
<i>C</i>	1789,4	1358,6	1632,4	1532,4	20,15%
<i>D</i>	1027,2	552,0	636,4	536,4	15,29%
<i>E</i>	2370,2	1616,0	1953,0	1853,0	20,85%
<i>F</i>	4529,2	3308,4	4098,8	3998,8	23,89%
<i>G</i>	2078,8	1072,4	1222,6	1122,6	14,00%
<i>H</i>	4778,4	3125,4	3896,6	3796,6	24,67%
<i>I</i>	8998,8	6375,4	8123,6	8023,6	11,73%

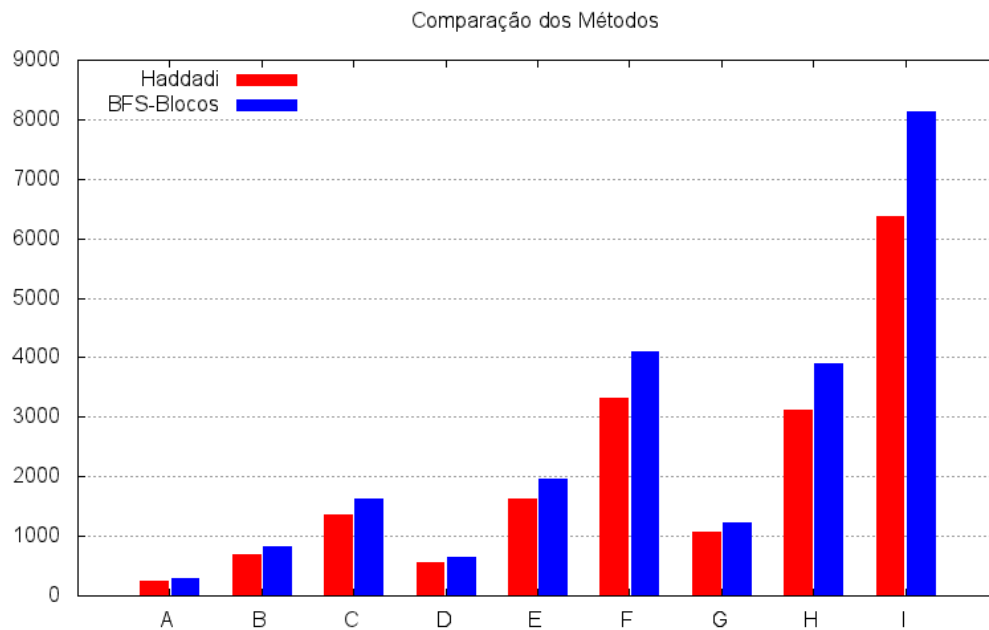


Figura 5.1: Gráfico de comparação de resultados.

indicam que o método proposto necessita de uma fase de melhoria fina, ou pós-otimização, a fim de que se torne possível obter soluções mais próximas das soluções de referência.

Não foi possível realizar comparações sobre o número de descontinuidades, dado que o trabalho de referência trata apenas do Problema de Minimização de Blocos Consecutivos, cuja função objetivo não é diretamente proporcional à função objetivo do MDP. Os resultados individuais para cada instância podem ser visualizados no Apêndice 7 e serem utilizados para futuras comparações.

Capítulo 6

Plano de Atividades Restantes

Na Tabela 6.1 são apresentadas informações sobre atividades planejadas para o trabalho na disciplina Monografia II, visando a conclusão deste trabalho de pesquisa.

Tabela 6.1: Cronograma de atividades para monografia II.

Tarefas	Mês 1	Mês 2	Mês 3	Mês 4
Pesquisar sobre métodos de refinamento	✓			
Implementar métodos de refinamento da heurística		✓	✓	
Executar experimentos			✓	
Analisar experimentos realizados			✓	✓
Descrever os experimentos computacionais				✓
Conclusão do texto da monografia				✓

As atividades basicamente se concentram em pesquisar, implementar e analisar métodos de melhoria fina para a heurística proposta, o que inclui métodos de busca local, metaheurísticas e programação linear. O novo método então será submetido a novos experimentos computacionais e análises adicionais.

Capítulo 7

Conclusões

Neste trabalho foram introduzidos a definição formal, uma revisão detalhada da literatura e uma implementação inicial de uma heurística para o Problema de Minimização de Descontinuidades. É possível concluir que este é um importante problema no contexto industrial, que pode reduzir os custos operacionais da produção de bens de consumo e também auxiliar na homogeneização das características físicas dos mesmos. Também foi observado que existem poucas propostas na literatura para a resolução deste problema.

A partir do experimento computacional preliminar realizado, pode-se dizer que o método inicial proposto apresentou resultados razoáveis também para a resolução de um problema relacionado, obtendo resultados próximos aos reportados na literatura. Os trabalhos futuros incluem o aprimoramento da heurística proposta, a realização de novos experimentos computacionais com a inclusão de outros conjuntos de instâncias.

Apêndice

Nas nove tabelas a seguir, a coluna *Instância* refere-se ao nome das instâncias, $|P|$ indica a quantidade de padrões, $|C|$ representa a quantidade de peças e os tempos de execução (expressos em segundos) são apresentados na coluna T . Novamente, estas instâncias foram originalmente utilizadas para o Problema de Minimização de Blocos Consecutivos, são apresentados duas colunas com os valores de solução obtidos: na coluna *Blocos* para a quantidade de blocos consecutivos e na coluna *Descontinuidades* para o MDP.

Tabela 7.1: Instâncias do grupo A.

Instância	$ P $	$ C $	T	<i>Blocos</i>	<i>Descontinuidades</i>
<i>A1</i>	100	200	0,23	284	184
<i>A2</i>	100	200	0,29	292	192
<i>A3</i>	100	200	0,26	293	193
<i>A4</i>	100	200	0,23	314	114
<i>A5</i>	100	200	0,21	308	208

Tabela 7.2: Instâncias do grupo B.

Instância	$ P $	$ C $	T	<i>Blocos</i>	<i>Descontinuidades</i>
<i>B1</i>	100	200	0,22	809	709
<i>B2</i>	100	200	0,22	824	724
<i>B3</i>	100	200	0,31	805	705
<i>B4</i>	100	200	0,24	833	733
<i>B5</i>	100	200	0,22	812	712

Tabela 7.3: Instâncias do grupo C.

Instância	$ P $	$ C $	T	<i>Blocos</i>	<i>Descontinuidades</i>
<i>C1</i>	100	200	0,27	1612	1512
<i>C2</i>	100	200	0,26	1591	1491
<i>C3</i>	100	200	0,23	1625	1525
<i>C4</i>	100	200	0,24	1626	1526
<i>C5</i>	100	200	0,26	1703	1603

Tabela 7.4: Instâncias do grupo D.

Instância	$ P $	$ C $	T	<i>Blocos</i>	<i>Descontinuidades</i>
<i>D1</i>	100	500	0,70	666	566
<i>D2</i>	100	500	0,63	681	581
<i>D3</i>	100	500	0,80	579	479
<i>D4</i>	100	500	0,74	646	546
<i>D5</i>	100	500	0,66	610	510

Tabela 7.5: Instâncias do grupo E.

Instância	$ P $	$ C $	T	<i>Blocos</i>	<i>Descontinuidades</i>
<i>E1</i>	100	500	0,64	1956	1856
<i>E2</i>	100	500	0,77	1933	1833
<i>E3</i>	100	500	0,86	1951	1851
<i>E4</i>	100	500	0,83	1936	1836
<i>E5</i>	100	500	0,80	1989	1889

Tabela 7.6: Instâncias do grupo F.

Instância	$ P $	$ C $	T	<i>Blocos</i>	<i>Descontinuidades</i>
<i>F1</i>	100	500	0,66	4033	3993
<i>F2</i>	100	500	0,64	4120	4020
<i>F3</i>	100	500	0,71	4091	3991
<i>F4</i>	100	500	0,60	4122	4022
<i>F5</i>	100	500	0,72	4128	4028

Tabela 7.7: Instâncias do grupo G.

Instância	$ P $	$ C $	T	<i>Blocos</i>	<i>Descontinuidades</i>
<i>G1</i>	100	1000	1,47	1231	1131
<i>G2</i>	100	1000	1,63	1201	1101
<i>G3</i>	100	1000	1,57	1230	1130
<i>G4</i>	100	1000	1,57	1264	1164
<i>G5</i>	100	1000	1,55	1187	1087

Tabela 7.8: Instâncias do grupo H.

Instância	$ P $	$ C $	T	<i>Blocos</i>	<i>Descontinuidades</i>
<i>H1</i>	100	1000	1,49	3957	3857
<i>H2</i>	100	1000	1,43	3799	3699
<i>H3</i>	100	1000	1,45	3936	3836
<i>H4</i>	100	1000	1,44	3736	3636
<i>H5</i>	100	1000	1,56	4055	3955

Tabela 7.9: Instâncias do grupo I.

Instância	$ P $	$ C $	T	$Blocos$	$Descontinuidades$
$I1$	100	1000	1,43	8165	8065
$I2$	100	1000	1,48	8208	8108
$I3$	100	1000	1,54	8062	7962
$I4$	100	1000	1,45	8115	8015
$I5$	100	1000	1,44	8068	7968

Referências Bibliográficas

- Becceneri, J. (1999). O problema de sequenciamento de padrões para minimização do número máximo de pilhas abertas em ambientes de corte industriais. *São José dos Campos. Tese (Doutorado em Ciências no Curso de Engenharia Eletrônica e Computação na Área de Informática)–ITA, Centro Tecnológico Aeroespacial.*
- Cormen, T. H. (2009). *Introduction to algorithms*. MIT press.
- Cuthill, E. e McKee, J. (1969). Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, pp. 157–172. ACM.
- Dom, M. (2009). *Recognition, Generation, and Application of Binary Matrices with the Consecutive Ones Property*. Cuvillier.
- Dyson, R. e Gregory, A. (1974). The cutting stock problem in the flat glass industry. *Operational Research Quarterly*, pp. 41–53.
- Garey, M. R. e Johnson, D. S. (1979). A guide to the theory of np-completeness. *WH Freeman, New York*.
- Gilmore, P. C. e Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations research*, 9(6):849–859.
- Haddadi, S.; Chenche, S.; Cheraitia, M. e Guessoum, F. (2015). Polynomial-time local-improvement algorithm for consecutive block minimization. *Information Processing Letters*, 115(6):612–617.
- Linhares, A. e Yanasse, H. H. (2002). Connections between cutting-pattern sequencing, vlsi design, and flexible machines. *Computers & Operations Research*, 29(12):1759–1772.
- Madsen, O. B. (1979). Glass cutting in a small firm. *Mathematical Programming*, 17(1):85–90.
- Madsen, O. B. (1988). An application of travelling-salesman routines to solve pattern-allocation problems in the glass industry. *Journal of the Operational Research Society*, pp. 249–256.

- Yanasse, H. H. (1997). On a pattern sequencing problem to minimize the maximum number of open stacks. *European Journal of Operational Research*, 100(3):454–463.