DOUGLAS MATUZALEM PONTES BELO LANÇA

Orientador: Marco Antonio Moreira de Carvalho

# UM ALGORITMO HEURÍSTICO APLICADO À MINIMIZAÇÃO DO ESTOQUE INTERMEDIÁRIO EM SISTEMAS INDUSTRIAIS

Ouro Preto

Agosto de 2016

Universidade Federal de Ouro Preto
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

# UM ALGORITMO HEURÍSTICO APLICADO À MINIMIZAÇÃO DO ESTOQUE INTERMEDIÁRIO EM SISTEMAS INDUSTRIAIS

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.

DOUGLAS MATUZALEM PONTES BELO LANÇA

Ouro Preto
Agosto de 2016

UNIVERSIDADE FEDERAL DE OURO PRETO

FOLHA DE APROVAÇÃO

# Um Algoritmo Heurístico Aplicado à Minimização do Estoque Intermediário em Sistemas Industriais

DOUGLAS MATUZALEM PONTES BELO LANÇA

Monografia defendida e aprovada pela banca examinadora constituída por:

Dr. Marco Antonio Moreira de Carvalho – Orientador
Universidade Federal de Ouro Preto

Dr. Luiz Henrique de Campos Merschmann
Universidade Federal de Ouro Preto

Dr. Puca Huachi Vaz Penna
Universidade Federal de Ouro Preto

Ouro Preto, Agosto de 2016

# Resumo

Em ambientes industriais, diferentes processos produtivos são implementados nas várias linhas de produção. Estas linhas são divididas em estágios, de acordo com o tipo de produto fabricado. Em geral é necessário que os processos produtivos sejam otimizados de acordo com diferentes objetivos, como, por exemplo, a minimização da utilização de obra prima e a economia de energia. Também é necessário que a sequência em que os produtos são processados em cada estágio das linhas de produção seja otimizada para reduzir o uso de estoque intermediário, para aumentar o ritmo de produção e também o ritmo de entrega dos produtos fabricados aos consumidores. Existem diversos aspectos que interferem na eficiência das linhas de produção, dando origem aos problemas de otimização industriais. A otimização destes problemas na maioria das vezes não é trivial, portanto, exige estudo e investimento. Neste trabalho será descrito com detalhes o Problema de Minimização de Espalhamento de Ordens (*Minimization of Order Spread Problem - MORP*), que tem como foco a diminuição do uso de estoque intermediário nas indústrias. Para a resolução deste problema é proposta uma solução simples baseada em um método de percurso em grafo já existente na literatura. Os experimentos foram realizados a partir do conjunto de instâncias propostas na literatura e os resultados são comparados com o método considerado o estado da arte.

# Abstract

In industrial environments, differents production processes are implemented in various production lines. These lines are divided into stages according to the type of product manufactured. In general, it is necessary that the production processes are optimized according to different purposes, such as, for example, minimizing the use of raw material and energy saving. It is also necessary that the sequence in which product processed at each stage of production lines is optimized to reduce the use of intermediate stock, to increase the production rate and also the rate of delivery of the manufactured products to consumers. For different processes there are several aspects that affect the efficiency of production lines, giving rise to industrial optimization problems. The optimization of these problems most of the time is not trivial, therefore, requires study and investment. This work describes in details the Minimization of Order Spread Problem - MORP, which is focused on reducing the use of intermediate stock in industries. To solve this problem we propose a simple solution based on a graph search method from the literature. The experiments considered a set of instances from the literature and the results are compared to the state-of-the-art.

*Dedico este trabalho aos meus pais e meu irmão que me apoiaram.*

# Agradecimentos

Agradeço aos meus pais, irmão, namorada e amigos que me apoiaram.

# Sumário

# Lista de Figuras

# Lista de Tabelas

# Capítulo 1

# Introdução

Com os avanços da tecnologia, as indústrias tendem cada vez mais a investir na melhoria da qualidade dos produtos e dos métodos de produção. É de interesse destas indústrias minimizar os gastos relacionados a consumo de energia, mão de obra, maquinário, manipulação e estoque de produtos ou qualquer outro gasto relacionado com as linhas de produção. Estas variáveis podem ser trabalhadas para otimizar o processo de produção industrial e no intuito de maximizar a lucratividade. Desta forma, surge a mobilização de profissionais de engenharia e ciências exatas para analisar e otimizar todos os aspectos das linhas de produção de uma indústria, desde o recebimento da matéria prima até a entrega do produto acabado ao consumidor final.

Os produtos são classificados como semi-acabados quando faltam poucas etapas do processo produtivo para que os mesmos sejam terminados e enviados aos seus respectivos destinos finais, ou seja, em parte dos casos, os produtos passam por processos de acabamento estético antes de serem enviados ao consumidor final. Os produtos que são classificados como acabados já passaram por todas as etapas do processo produtivo ligadas a sua fabricação e portanto podem ser enviados ao consumidor final.

Uma linha de produção é uma forma de produção em série, onde vários funcionários, com ajuda de maquinário, trabalhando de forma sequencial, fabricam produtos semi-acabados ou acabados. A forma mais característica, a da montagem em série, foi inventada por Henry Ford, empresário estadunidense do setor automobilístico. A produção em massa, popularizada por Ford no início do século XX, se tornou um modo de produção muito difundido, pois permitia altas taxas de produção por trabalhador e ao mesmo tempo disponibilizava produtos a preços baixos.

Neste trabalho, considera-se uma indústria que fabrica vários produtos diferentes e diferentes clientes que possuem diferentes ordens de compra, ou seja, demandas por conjuntos de produtos. É necessário atender estas ordens o mais rápido possível, de maneira a diminuir o tempo de atendimento de todos os pedidos, fazendo com que o pedido seja despachado tão rápido quanto possível, reduzindo o uso do estoque intermediário. O estoque intermediário consiste no armazenamento de produtos fabricados, porém, ainda não preparados para entrega. A utilização deste tipo de estoque implica em utilização de mão de obra e maquinário adicionais para manipulação dos produtos e área física disponível para tanto. Além disto, os produtos podem ser danificados devido a acidentes ou intempéries.

A fabricação dos produtos é realizada em estágios, sendo que em cada um dos estágios um tipo de produto é fabricado. O intervalo entre o início e o fim da fabricação dos produtos de uma mesma ordem de compra é chamado de espalhamento de ordem, e é medido pelo número de estágios necessários para concluir a fabricação de todos os produtos de uma ordem de compra específica de um cliente.

O Problema de Minimização de Espalhamento de Ordens (Minimization of Order Spread, MORP) é um problema de sequenciamento de produção proposto originalmente no contexto de operações de corte na indústria de vidro, no qual, a estocagem intermediária das peças é custosa. A operação de corte consiste em, dadas unidades maiores de matéria prima, que podem ser barras, bobinas ou chapas de papel, madeira, vidro, ou metal, cortá-las em unidades menores de acordo com um padrão pré-estabelecido. Este padrão é a forma como as unidades menores estão dispostas nas unidades maiores.

Considerando a produção em larga escala, existem vários padrões diferentes a serem cortados, e diferentes sequências de corte otimizam determinados objetivos das indústrias. Entre estes objetivos cita-se a homogeneidade das características físicas do produto final, o ritmo da produção, o ritmo da entrega dos pedidos de compra, a economia com armazenamento intermediário e o gasto com mão de obra.

Quando o espalhamento das ordens de compra é minimizada, o tempo em que os clientes aguardam pelo atendimento de seus pedidos – e consequentemente a utilização do estoque intermediário – é minimizado. Existe, também, a tendência a minimizar a heterogeneidade das características físicas dos produtos.

## 1.1 Motivação

O Problema de minimização de espalhamento de ordens possui aplicação industrial ampla, portanto, o estudo deste problema pode contribuir para a otimização das linhas de produção de diversas indústrias. O problema tratado neste trabalho tem equivalência com outros problemas na literatura, por exemplo, MOSP, MDP e *Talent Scheduling*, fazendo com que este estudo possa servir de base para estudos de outros problemas relacionados. Este problema é classificado como NP-Difícil (Garey e Johnson, 2002), portanto, existe também a motivação teórica.

## 1.2 Objetivos

De uma maneira geral, este trabalho consiste em realizar pesquisa para geração de embasamento teórico para compreensão dos conceitos relacionados ao problema de minimização de espalhamento de ordens e aos problemas de sequenciamento de produção para elaborar uma heurística consistente que alcance bons resultados e se aplique a contextos reais dos processos de produção das industrias nacionais. São objetivos específicos:

1. Realizar pesquisa para geração de embasamento teórico e revisão bibliográfica sobre problema de minimização de espalhamento de ordens (MORP);

2. Elaborar uma heurística consistente que possa ser utilizada no contexto do problema tratado que permita a obtenção rápida de soluções próximas ou melhores que as soluções ótimas encontradas anteriormente sem que se perca a vantagem da busca sistemática – inicialmente considerando problemas específicos, mas com uma possibilidade de generalização;

3. Pesquisar técnicas para melhoria fina de soluções (polishing);

4. Buscar a aplicação prática dos métodos desenvolvidos em contextos reais, a fim de que também seja constituído um avanço para as indústrias nacionais;

Além dos objetivos principais, outros produtos deste projeto de pesquisa serão trabalhos publicados em periódicos e eventos nacionais.

## 1.3 Organização do Trabalho

O restante deste trabalho esta organizado da seguinte forma. O Capítulo 2 apresenta a revisão da literatura de 1988 até 2015. A base conceitual do MORP é apresentada no Capítulo 3. A implementação inicial do método proposto é apresentada no Capítulo 4 e os experimentos são descritos 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

Neste capítulo os trabalhos publicados anteriormente serão analisados em ordem cronológica, colocando em evidência os avanços nas heurísticas e métodos propostos para o Problema de Minimização de Espalhamento de Ordens. Trabalhos que abordam problemas relacionados ao MORP também são abordados. Os trabalhos revisados foram publicados entre 1988 a 2016.

## 2.1 O Trabalho de Madsen (1988)

O Problema de Minimização de Espalhamento de Ordens foi proposto originalmente por Madsen (1988) ao estudar o processo de corte na indústria de vidro. De acordo com este trabalho, o custo de estocagem intermediária em tal industria é custosa e de alto risco para a integridade física das peças. As peças de vidro não podem ser estocadas em pilhas altas, devem ser manuseadas com cuidado devido a fragilidade do material e quando estocadas existia a dificuldade em diferenciá-las. No intuito de minimizar a utilização de estocagem e o manuseio das peças de vidro, foi proposta uma estratégia de solução em três estágios, descritos a seguir.

No primeiro estágio, resolve-se o problema de corte de estoque (definição dos padrões de corte) sem levar em consideração o espalhamento de ordens. Para este primeiro estágio foi usado o método proposto por Gilmore e Gomory (1966).

No segundo estágio, constrói-se a matriz de distâncias $C$ baseada no resultado do estágio 1. Esta matriz é construída da seguinte forma: Para produtos contidos em ordens de compras *singulares* (ou seja, uma ordem de compra única para um produto específico), denotadas por $p_s, p'_s, p''_s, \ldots$, o valor da $c_{p_s,j}$ e $c_{j,p_s}$ ($j \neq p_s$ e $j$ variando entre 1 e o número máximo de ordens) é definido como 1.000; o valor das entradas $c_{p'_s,p''_s}$ e $c_{p''_s,p'_s}$ é definido como 110; para as ordens $i$ e $j$ ($i \neq j$) que possuem $n$ produtos em comum o valor de $c_{i,j}$ é definido por $100 - 10 \times n$. Os elementos da diagonal principal da matriz $C$ possuem valor 10.000 cada.

No terceiro estágio resolve-se o Problema do Caixeiro Viajante, considerando-se os produtos como vértices e a matriz de distâncias $C$. Para solução, foi empregado o método *3-opt* proposto por Lin (1965). Este método é uma heurística de melhoramento definida da seguinte maneira: dada uma solução inicial em forma de grafo, elimina-se três arestas – $(k_1, k_2), (j_1, j_2)$ e $(i_1, i_2)$ – e então são testados todas as combinações de ligações novas entre os vértices que estavam conectados por tais arestas. Se existir alguma nova configuração melhor que a anterior, ou seja, se o comprimento total da rota diminuir, mantêm-se a nova rota. Caso contrário, escolha novamente outras três arestas para análise.

A sequência dos vértices na rota estabelecida como solução do Problema do Caixeiro Viajante determina a ordem de fabricação dos produtos, visando minimizar o espalhamento de ordens. O algoritmo atribui um valor alto (10.000) para os elementos da diagonal principal da matriz $C$, para garantir que cada produto será fabricado apenas uma vez. Os elementos da diagonal principal da matriz representam as arestas que ligam um vértice a ele mesmo. O valor 110 é atribuído as arestas que representam conexões entre produtos (vértices) contidos apenas em ordens de compra singulares para que tais ordens sejam atendidas por último, pois não possuem espalhamento. Os valores atribuídos as demais arestas $(100 - 10 \times n)$ representam quantas ordens $(n)$ os dois produtos (vértices), que estão conectados, possuem em comum. Quanto maior este número, menor é o peso da aresta e mais probabilidade existe dos dois produtos em questão serem alocados em sequência.

O resultado apresentado pelo autor mostrou a diminuição em média de 18% do espalhamento de ordens e de 30% na minimização de descontinuidades em relação aos resultados obtidos com a execução do estágio 1. Uma *descontinuidade* ocorre quando a produção de um produto é iniciada, interrompida e então retomada. O problema de minimização de descontinuidades (*Minimization of Descontinuities* – MDP) tem como objetivo minimizar o número de descontinuidades sem considerar o tamanho da mesma, visando maximizar a homogeneidade das características físicas dos produtos. A aplicação proposta por Madsen (1988) resolveu o problema de minimização de descontinuidades com mais eficiência.

## 2.2 O Trabalho de Foerster e Wäscher (1998)

Em Foerster e Wäscher (1998), foi proposto o uso da metaheurística Recozimento Simulado (*Simulated Annealing* – SA) para solução do MORP. O método SA é uma meta-heurística que consistem em uma técnica de busca local probabilística, fundamentada numa analogia com a termodinâmica. O algoritmo deste método substitui a solução atual por uma segunda (situada na vizinhança da primeira), selecionada de acordo com a função objetivo e a variável $T$ (dita Temperatura, por analogia). Quanto maior a temperatura $T$, maiores as chances da solução selecionada substituir a anterior. A medida que o algoritmo progride, a variável $T$ é decrementada, fazendo com que a solução convirja para um ótimo local.

Os autores implementaram o procedimento de Madsen (1988), o procedimento *3-opt* e o SA utilizando as mesmas instâncias usadas por Madsen (1988). De acordo com a implementação dos autores, o *3-opt* superou o procedimento de Madsen (1988) e o SA superou o *3-opt*.

O SA apresentou redução considerável na média dos espalhamentos das ordens, porém, o procedimento de Madsen (1988) foi melhor em relação ao tempo de execução.

## 2.3 O Trabalho de Fink e Voß (1999)

No ano seguinte Fink e Voß (1999), empregaram diferentes heurísticas para solução do Problema de Minimização de Pilhas Abertas (ou MOSP, um problema correlato) e do MORP. O MOSP consiste em otimizar o uso de estoque primário (ainda no ambiente de produção) visando obedecer restrições que dizem respeito a capacidade de estoque, utilização de mão de obra e preservação da integridade física das peças.

Neste trabalho, foram geradas diferentes soluções iniciais pelos métodos de Inserção Mais Barata (*Cheapest Insertion* – CI) e Inserção Mais Barata do Pior Padrão (*Cheapest Insertion of the Worst Pattern* – CIW). Para a otimização destas soluções iniciais foram utilizados os métodos *2-opt*, uma nova implementação de SA e variações da Busca Tabu.

A idéia básica da Busca Tabu manter um histórico da busca realizada por um algoritmo específico de busca local e utilizar estas informações para evitar a exploração redundante do espaço de busca, visando também escapar de ótimos locais. Neste trabalho foram implementas as seguintes técnicas de Busca Tabu avançadas, utilizando como solução inicial o CIW:

**TSS:** Busca Tabu Estática (*Static Tabu Search*). Esta técnica tem como princípio básico proibir a inversão de alterações realizadas pelo algoritmo de busca local por um número fixo de iterações (duração tabu);

**TSTA:** Busca Tabu Estrita por Trajetória Aproximada (*Strict Tabu Search by Approximate Trajectory*. Esta técnica não permite que trajetórias semelhantes às já realizadas no espaço de busca pelo algoritmo e busca local sejam realizadas;

**TSTAE:** Busca Tabu Estrita por Trajetória Aproximada com Cinco Movimentos Aleatórios de Fuga (*Strict Tabu Search by Approximate Trajectory Including Five Random Escape Moves*). Semelhante a TSTA, entretanto, inclui cinco movimentos aleatórios de fuga, não especificados;

**TSTRE:** Busca Tabu Reativa (*Reactive Tabu Search Including Five Random Escape Moves*). Esta técnica é uma modificação da TSS para que o critério de duração tabu seja adaptada dinamicamente ao longo da busca, incluindo cinco movimentos aleatórios de fuga, não especificados;

**TSTRE5000:** Semlhante ao anterior, porém, aplicado a cada 5000 iterações.

Os métodos implementados foram comparados com métodos *3-opt* e a implementação de SA proposta por Foerster e Wäscher (1998). Os melhores resultados foram obtidos pela execução do TSRE5000, melhorando a qualidade de solução inicial em até $49,5\%$. O SA proposto por Foerster e Wäscher (1998) atingiu resultados entre $4,9\%$ e $15,9\%$ distantes dos melhores valores encontrados. As instancias utilizadas foram geradas aleatoriamente e fornecidas para testes neste trabalho.

## 2.4 O Trabalho de Kim et al. (2016)

## 2.5 Trabalhos Relacionados

O Problema de Escalonamento de Talentos (*Talent Scheduling Problem*) é um problema que pode ser visto como uma variação do MORP (Cheng et al., 1993). Este problema tem como objetivo minimizar o tempo que atores, equipamentos ou técnicos envolvidas nas produções de filmes permanecem no local da filmagem sem estar trabalhando. Considerando os talentos (atores, equipamentos ou técnicos) sendo ordens de compra e os dias de filmagem sendo produtos, monta-se a matriz $M$ definindo o valor da célula $m_{ij} = 1$ quando o talento $i$ é requisitado para filmar no dia $j$ e $m_{ij} = 0$ caso contrário. Desta forma, para reduzir o tempo ocioso dos talentos aplica-se técnicas para resolver o MORP levando em consideração a matriz $M$. Este problema se diferencia do MORP devido ao fato de considerar custos de atores, técnicos e equipamentos, ou seja, o espalhamento das ordens referentes aos talendos mais caros tende a ser minimizado com prioridade. Cheng et al. (1993) propôs a utilização do *branch-and-bound* e uma heurística que utiliza o *branch-and-bound* para resolver tal problema. A heurística usada é baseada na escolha da ramificação mais atrativa, ou seja, escolhen sempre o nó de menor valor. Posteriormente,o método *2-opt* é aplicado para trocar o conteúdo dos dias de filmagens, no intuito de aprimorar a solução. A heurística se mostrou mais eficiente em questão ao tempo, ambos chegando a mesma solução. Com base em uma instância real, o custo da filmagem relativo aos dias de espera dos atores, equipe e equipamentos de um filme foi reduzido de US$36.400 para US$17.900.

Há também o Problema de Escalonamento de Ensaio (*Scheduling a Rehearsal*), que possui como objetivo minimizar o tempo em que os músicos de uma orquestra permanecem no local do ensaio sem que estejam ensaiando. Durante um ensaio, diferentes movimentos são executados, porém, nem todos os músicos tocam em todos os movimentos, gerando ociosidade de alguns músicos de acordo com a estrutura da peça ensaiada. De forma análoga ao *Talent Scheduling Problem*, este problema se assemelha ao MORP e pode ser definido por ordens de compra sendo músicos e produtos sendo cada movimento do ensaio. A característica principal deste problema é levar em consideração o tempo de ensaio, ou seja, os espalhamentos de ordens podem variar de acordo com o tamanho de cada movimento do ensaio. Smith (2003) propôs uma modelagem usando programação por restrições. O autor apresentou apenas uma tabela de instância, tal tabela possui 5 músicos e 9 estágios do ensaio. O autor encontrou o resultado

17 unidades de tempo que, segundo ele, é o resultado ótimo e apresentou a sequencia de escalonamento.

O MORP, MOSP e MDP (problemas mencionados anteriormente neste trabalho) são problemas correlatos. Linhares e Yanasse (2002) estabeleceu relação entre estes afirmando que não são equivalente, ou seja, uma solução ótima para um não é solução ótima para os demais.

O MORP é também um caso especial do Problema de Minimização de Largura de Banda em Matrizes, segundo (Yanasse, 1997). Este problema consiste em permutar as linhas e as colunas de matrizes binárias para minimizar a distância entre os elementos não nulos e a diagonal principal. De acordo com Garey e Johnson (2002) o MORP é NP-Difícil por redução ao Partição de Triângulos.

# Capítulo 3

# Fundamentação Teórica

O Problema de Minimização de Espalhamento de Ordens, proposto por Madsen (1988), é relacionado ao contexto das indústrias em que uma parte da linha de produção confecciona um conjunto $J$ de produtos distintos para atender a demanda de um conjunto $I$ de ordens de compra, formadas por tais produtos. A produção é dividida em estágios em que cada estágio é fabricado um produto diferente.

Na linha de produção todos os produtos já fabricados são armazenados temporariamente no estoque intermediário. Este estoque intermediário é utilizado para armazenar os produtos de uma ordem de compra de um cliente. Uma vez que todos os produtos que compõem uma ordem de compra específica tenham sido fabricados, estes podem ser removidos do estoque intermediário.

O número de estágios necessários para a produção de todos os produtos de uma ordem, desconsiderando o primeiro estágio, é denominado espalhamento de ordem. O primeiro estágio é desconsiderado, porque, ordens de compra que possuem apenas um produto, são despachadas imediatamente sem que tal produto fique armazenado no estoque intermediário, ou seja, o tamanho do espalhamento de ordem quando a mesma possui apenas 1 produto é 0. Quando os produtos de uma mesma ordem não são fabricados consecutivamente, ou seja, existe um intervalo de estágios em que nenhum produto desta ordem é fabricado, caracteriza-se a *descontinuidade*. As descontinuidades influenciam diretamente no espalhamento de ordens. Uma ordem composta por $n$ produtos em que existem $m$ estágios de descontinuidade em sua produção, tem como espalhamento $(n-1)+m$ estágios.

Uma instancia para o MORP pode ser representada como uma matriz binária $P$ que relaciona ordens de compra e produtos. Quando o produto $j \in J$ está presente na ordem de compra $i \in I$, o valor da matriz $P$ na posição $p_{ij}$ é 1, caso contrário, o valor é 0. A Tabela 3.1 é um exemplo de instância para o MORP. As linhas, numeradas de $i_1$ a $i_6$, representam as ordens de compras dos clientes. As colunas, numeradas de $j_1$ a $j_6$, representam os produtos.

Uma solução para o MORP é uma permutação $\pi$ das $|J|$ colunas da matriz $P$ gerando uma matriz permutação $Q^{\pi}$. É de suma importância para o problema identificar os tamanhos dos

Tabela 3.1: Instância MORP.

| | $j_1$ | $j_2$ | $j_3$ | $j_4$ | $j_5$ | $j_6$ |
|---|---|---|---|---|---|---|
| $i_1$ | 1 | 1 | 0 | 0 | 0 | 0 |
| $i_2$ | 1 | 0 | 1 | 0 | 0 | 0 |
| $i_3$ | 0 | 0 | 0 | 1 | 1 | 0 |
| $i_4$ | 0 | 0 | 0 | 1 | 0 | 1 |
| $i_5$ | 0 | 1 | 0 | 0 | 1 | 0 |
| $i_6$ | 0 | 0 | 1 | 0 | 0 | 1 |

espalhamentos das ordens, medidos em número de estágios. Para identificar o comprimento dos espalhamentos de ordens a matriz $Q^\pi$ é definida com a propriedade de 1*s consecutivos*. Todo elemento da matriz $P$, de valor 0, compreendido entre dois elementos de valor 1 em uma mesma linha é preenchido com o valor 1 na matriz $Q^\pi$. Esta propriedade indica, na matriz $Q^\pi$, por quais estágios se estende a produção que atende cada ordem de compra. A matriz $Q^\pi = \{q_{ij}\}$ é definida por:

$$q_{ij} = \begin{cases} 1 & \text{se } \exists x, \exists y \mid \pi[x] \leq j \leq \pi[y] \text{ e } p_{ix} = p_{iy} = 1 \\ 0, & \text{caso contrário} \end{cases} \tag{3.1}$$

Na Equação 3.1, $\pi$ denota a permutação dos $|J|$ produtos definindo a sequência em que cada produto será fabricado, ou seja, $\pi[j]$ é o estágio da produção em que o produto $j$ será fabricado.

As Tabelas 3.2 representam exemplos de soluções para o MORP representadas da forma da matriz $Q^\pi$. As colunas representam o instante $\pi$ em que produtos serão fabricados, as linhas representam ordens de compra. Os elementos da matriz de valor 1 em negrito representam as descontinuidades.

Tabela 3.2: Exemplos de soluções em forma da matriz $Q^\pi$.

| | $j_5$ | $j_2$ | $j_4$ | $j_6$ | $j_3$ | $j_1$ |
|---|---|---|---|---|---|---|
| $i_1$ | 0 | 1 | **1** | **1** | **1** | 1 |
| $i_2$ | 0 | 0 | 0 | 0 | 1 | 1 |
| $i_3$ | 1 | **1** | 1 | 0 | 0 | 0 |
| $i_4$ | 0 | 0 | 1 | 1 | 0 | 0 |
| $i_5$ | 1 | 1 | 0 | 0 | 0 | 0 |
| $i_6$ | 0 | 0 | 0 | 1 | 1 | 0 |

$(a)$

| | $j_1$ | $j_6$ | $j_5$ | $j_4$ | $j_3$ | $j_2$ |
|---|---|---|---|---|---|---|
| $i_1$ | 1 | **1** | **1** | **1** | **1** | 1 |
| $i_2$ | 1 | **1** | **1** | **1** | 1 | 0 |
| $i_3$ | 0 | 0 | 1 | 1 | 0 | 0 |
| $i_4$ | 0 | 1 | **1** | 1 | 0 | 0 |
| $i_5$ | 0 | 0 | 1 | **1** | **1** | 1 |
| $i_6$ | 0 | 1 | **1** | **1** | 1 | 0 |

$(b)$

A matriz $Q^\pi$ é definida para auxiliar a Função 3.2 a encontrar qual o maior espalhamento de ordens dentre todas as $|I|$ ordens. A linha da matriz $Q^\pi$ que possuir o maior número de 1s consecutivos é a ordem com maior espalhamento. O comprimento do espalhamento de uma ordem $i$ é definido pela soma dos valores da linha $i$ da matriz $Q^\pi$ subtraído de um.

$$Z_{MORP}(Q^{\pi}) = \max_{i \in I} \left( \sum_{j=1}^{|J|} q_{ij} - 1 \right) \tag{3.2}$$

No sequenciamento da Tabela 3.2 letra $a$, o produto $j_5$ é o primeiro produto a ser fabricado, pois $j_5$ é a primeira coluna da tabela, seguida por $j_2, j_4, j_6, j_3, j_1$. A ordem de compra $i_5$ é a primeira a ser despachada para o próximo estágio da produção (ou consumidor final) pois é composta pelos produtos $j_5$ e $j_2$, sendo $j_2$ o último da ordem a ser produzido, porém, produzido no segundo estágio. A ordem $i_1$ possui dois produtos, $j_2$ e $j_1$, sendo o produto $j_2$ produzido no estágio 2 e o produto $j_1$ produzido no estágio 6, sendo assim, o espalhamento desta ordem tem tamanho 4. Para a ordem em questão ($i_1$), o produto $j_2$ foi armazenado no estoque intermediário até o estágio 6, em que o último produto da ordem é fabricado e então a ordem completa pode ser despachada. A Tabela 3.2($a$) é melhor solução em relação a Tabela 3.2($b$), pois, o maior espalhamento contido na primeira solução é menor que o maior espalhamento contido na segunda solução.

A função objetivo do MORP visa minimizar o maior espalhamento das ordens, ou seja, minimizar a maior soma em qualquer linha da matriz $Q^{\pi}$. Com base na análise da Função 3.2, é necessário definir uma permutação $\pi$ que a minimize. A função objetivo é definida de acordo com a Função 3.3, onde $\Pi$ representa o número máximo de permutações possíveis da matriz $Q^{\pi}$. Em outras versões, a função objetivo do MORP visa minimizar o espalhamento médio das ordens de compra.

$$\min_{\pi \in \Pi} \; Z_{MORP}(Q^{\pi}) \tag{3.3}$$

Como mencionado anteriormente, para o MORP, o número de descontinuidades das soluções não é levado em consideração, portanto, uma solução que possui várias descontinuidades menores é melhor que uma solução que possui uma descontinuidade maior. Em suma, o Problema de Minimização de Descontinuidades trata do número de descontinuidades, ao contrário do MORP, que trata da duração das mesmas.

## 3.1 Pré-Processamento Por Dominância

O pré-processamento por dominância é feito para reduzir o tamanho das instâncias removendo os dados redundantes. Os dados são considerados redundantes quando os mesmos podem ser ignorados ao resolver o problema sem que sua estrutura seja afetada. O pré-processamento por dominância aplicado ao MORP remove os produtos que estão presentes em todas as ordens de compra em que um segundo produto também esta presente.

Dados dois produtos $p_i$ e $p_j$, seja $o(p_i)$ uma função que retorna todas as ordens de compra pelo produto $p_i$. Caso o produto $p_j$ conste exatamente em um subconjunto das ordens de compra em que $p_i$, consta (ou seja $o(p_j) \subseteq o(p_i)$), dizemos que $o(p_j)$ é *dominado* por $p_i$. Pode-se, sem perda de generalidade, considerar $p_i$ e $p_j$ como sendo um único produto, sequenciando-os consecutivamente na solução.

Antes da aplicação de um método para a solução do MORP, para diminuir o tamanho das instâncias, os produtos dominados podem ser removidos das mesmas. Depois de obtida uma solução para a instância reduzida, os produtos dominados são inseridos imediatamente após seus respectivos produtos dominantes em $\pi$, mantendo a otimalidade da solução. A Tabela 3.3 apresenta um exemplo de instância em que é possível reduzir seu tamanho através do processamento por dominância.

Tabela 3.3: Exemplo de nova instância (a) antes do pré-processamento e (b) após a aplicação do pré-processamento por dominância.

| | $j_1$ | $j_2$ | $j_3$ | $j_4$ | $j_5$ | $j_6$ |
|---|---|---|---|---|---|---|
| $i_1$ | 1 | 0 | 0 | 1 | 1 | 0 |
| $i_2$ | 1 | 1 | 1 | 0 | 0 | 0 |
| $i_3$ | 0 | 0 | 1 | 1 | 0 | 0 |
| $i_4$ | 1 | 1 | 1 | 0 | 1 | 0 |
| $i_5$ | 0 | **1** | 0 | 0 | 1 | **1** |
| $i_6$ | 0 | **1** | 0 | 0 | 0 | **1** |

$(a)$

| | $j_1$ | $j_2$ | $j_3$ | $j_4$ | $j_5$ |
|---|---|---|---|---|---|
| $i_1$ | 1 | 0 | 0 | 1 | 1 |
| $i_2$ | 1 | 1 | 1 | 0 | 0 |
| $i_3$ | 0 | 0 | 1 | 1 | 0 |
| $i_4$ | 1 | 1 | 1 | 0 | 1 |
| $i_5$ | 0 | 1 | 0 | 0 | 1 |
| $i_6$ | 0 | 1 | 0 | 0 | 0 |

$(b)$

Analisando a Tabela 3.3 (a) verifica-se que o produto 6 está presente nas ordens 5 e 6, e o produto 2 está contido nas ordens 2, 4, 5, 6, ou seja, o conjunto de ordens de compra que contem o produto 6 é subconjunto das ordens de compra que contem o produto 2. O produto 6 é dominado por 2, portanto, o produto 6 é removido sem alterar a solução do problema, mostrado na Tabela 3.3 (b).

## 3.2 Limitante Inferior

O limite inferior, para problemas de minimização como o MORP, é um valor para solução menor ou igual ao valor da solução ótima. Em métodos de solução compostos por múltiplas iterações, o limite inferior pode ser usado como critério de parada além de critérios já existentes: uma vez atingida uma solução com valor igual ao limite inferior, não existe forma de melhorá-la, logo, a execução pode ser interrompida. O valor do limite inferior, em alguns casos, pode não ser viável, ou seja, o método para resolução do problema não achará tal resposta pois esta pode quebrar restrições do problema.

O limite inferior pode auxiliar o método de solução de forma que iterações que não podem melhorar a solução não são executadas, diminuindo assim o tempo de execução.

Um limite inferior para MORP é o numero de produtos que compõem a maior ordem de compra da instância. Não existe forma de obter espalhamento de ordens menor que este limite inferior sem modificar a instância. A Tabela 3.4 possui limite inferior igual a 4.

Tabela 3.4: Limitante inferior.

| | $j_1$ | $j_2$ | $j_3$ | $j_4$ | $j_5$ | $j_6$ |
|---|---|---|---|---|---|---|
| $i_1$ | 1 | 1 | 0 | 1 | 1 | 1 |
| $i_2$ | 1 | 0 | 1 | 0 | 0 | 0 |
| $i_3$ | 0 | 0 | 0 | 1 | 1 | 0 |
| $i_4$ | 0 | 0 | 0 | 1 | 0 | 1 |
| $i_5$ | 0 | 1 | 0 | 0 | 1 | 0 |
| $i_6$ | 0 | 0 | 1 | 0 | 0 | 1 |

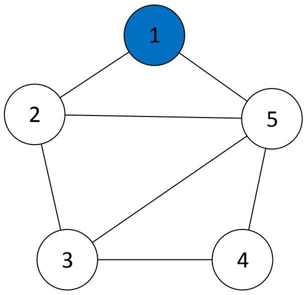
A Tabela 3.4 apresenta um exemplo de instância MORP em que o limitante inferior para o espalhamento máximo de uma ordem é dado pela ordem $i_1$. Esta é a ordem que possui mais produtos, ou seja, é a linha da matriz que possui maior número de uns. Não existe permutação $\pi$ das colunas da referida instância que possua espalhamento máximo menor que o número de produtos da ordem $i_1$.

## 3.3 Busca em Largura

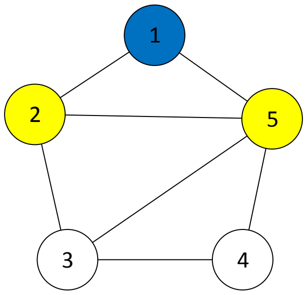
A Busca em Largura (ou *Breadth-First Search, BFS*) é um método de busca que explora sistematicamente todos os vértices de um grafo direcionado ou não-direcionado. A BFS é realizada da seguinte forma: dado um vértice inicial, o algoritmo explora todos seus vértices vizinhos. Então, para cada um desses vértices, explora-se todos seus vértices vizinhos que ainda não tenham sido explorados e assim por diante, até que o alvo da busca seja encontrado ou todos os vértices do grafo tenham sido explorados.

O algoritmo garante que nenhum vértice ou aresta seja visitado mais de uma vez. Para isso, utiliza-se uma estrutura de dados fila para garantir a ordem de exploração dos vértices: cada vértice recém-explorado é adicionado ao final da fila, e o próximo vértice a ter sua vizinhança atualizada é o primeiro da fila. A ordem de exploração dos vértices pode ser definida por diferentes critérios para que uma determinada característica seja priorizada, por exemplo, ordem lexicográfica ou peso das aresta de um vértice para os vizinhos.

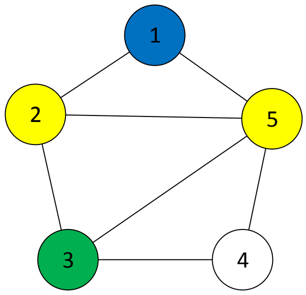
A Figura 3.1 exemplifica a execução da BFS em um grafo genérico. A fila é inicialmente vazia, então o vértice 1 (em azul na Figura 3.1a) é adicionado a fila e marcado como explorado. No segundo momento, os vizinhos do vértice 1 (vértices 2 e 5, em amarelo na Figura 3.1b), que não foram explorados, são adicionados a fila em ordem lexicográfica e marcados como explorados. No terceiro momento, os vizinhos do vértice 2 que não foram explorados (vértice 3 em verde na Figura 3.1c) são adicionados na fila em ordem lexicográfica. O vértice 2 possui como vizinhos os vértices 1, 3 e 5, sendo apenas o 3 não explorado, portanto, os vértices 1 e 3 não são adicionados a fila novamente. No quarto momento, os vizinhos do vértice 5 (vértice 4 em vermelho na Figura 3.1d) que não foram explorados são adicionados a fila em ordem lexicográfica. Todos os vértices do grafos estão marcados como explorados, portanto, a busca termina e a fila representa a ordem de exploração do grafo.



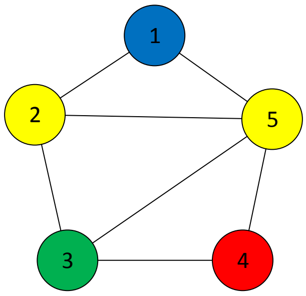

(a)




(b)




(c)




(d)

Figura 3.1: Busca em Largura

# Capítulo 4

# Desenvolvimento

Neste capítulo, são descritos a representação computacional e o método de solução do MORP proposto, em detalhes. A proposta deste capítulo é definir claramente a representação computacional e gerar solução inicial que será melhorada nos trabalhos futuros.

## 4.1 Representação Computacional

Como mencionado anteriormente, uma instância para o MORP é dada em forma de matriz binária, em que as linhas representam ordens de compras e as colunas representam produtos que compõem estas ordens. Para representação computacional do MORP, é definido um grafo $G$ em que vértices representam ordens de compra e uma aresta entre dois vértices existe quando as ordens de compra correspondentes possuem produtos em comum. O peso de cada aresta é definido pela quantidade de produtos em comum entre as ordens de compra.

A Tabela 4.1 apresenta uma instância MORP e a representação em grafo correspondente é apresentada na Figura 4.1.

Tabela 4.1: Representação em grafo.

| | $j_1$ | $j_2$ | $j_3$ | $j_4$ | $j_5$ | $j_6$ |
|---|---|---|---|---|---|---|
| $i_1$ | 1 | 1 | 0 | 0 | 0 | 1 |
| $i_2$ | 0 | 0 | 1 | 0 | 0 | 0 |
| $i_3$ | 1 | 0 | 0 | 0 | 0 | 1 |
| $i_4$ | 0 | 0 | 0 | 1 | 0 | 0 |
| $i_5$ | 0 | 1 | 0 | 1 | 1 | 0 |
| $i_6$ | 1 | 1 | 1 | 1 | 1 | 1 |

A ordem de compra $i_6$ da referida Tabela 4.1 possui todos os produtos disponíveis em sua composição e estes também compõem as demais ordens de compra, portanto, existe uma aresta entre o vértice 6 e todos os demais vértices. Especificamente, as ordens de compra $i_1$ e $i_5$ possuem três produtos em comum com a ordem de compra $i_6$ e um produto em comum entre si. Desta forma, o peso das arestas {1, 6} e {1, 5} é três e o peso da aresta {1, 5} é um. O mesmo processo se repete para as demais ordens de compra para construção do grafo.

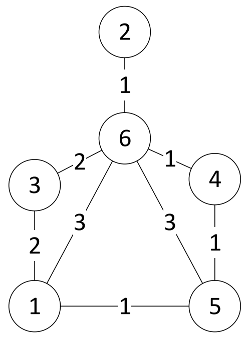



Figura 4.1: Grafo referente a Tabela 4.1

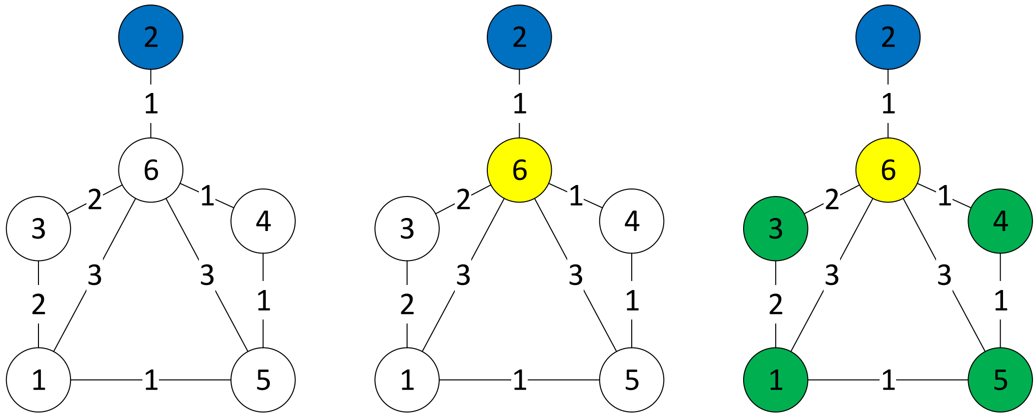
## 4.2 Busca em Largura Aplicada ao MORP

Neste trabalho, a BFS aplicada ao MORP utiliza como vértice inicial aquele de menor grau. A prioridade de exploração dos vértices é determinada pelo peso das arestas em ordem decrescente, ou seja, dado um vértice $v_1$ ligado aos vértices $v_2$ e $v_3$ por arestas de peso 3 e 5 respectivamente, a ordem de exploração será $v_3$ e $v_2$. O resultado da Busca em Largura é um vetor $\phi$ que representa a ordem em que os vértices foram explorados.

A Figura 4.2 exemplifica a execução da BFS no grafo representado na Figura 4.1. O vértice 2 (em azul em 4.2a) é o de menor grau, portanto, é o vértice inicial da busca. O vértice 2 é adicionado à fila e marcado como explorado. O referido vértice possui como vizinho apenas o vértice 6 (em amarelo em 4.2b), portanto, este vértice é adicionado à fila e marcado como explorado. O vértice 6 possui como vizinhos não explorados os vértices 1, 3, 4 e 5, (em verde em 4.2c). Estes vértices são marcados como explorados e adicionados à fila levando em consideração o peso das arestas que os ligam ao vértice 6 em ordem decrescente. A ordem em que os vértices são adicionados à fila é 1, 5, 3 e 4. Todos os vértices do grafo estão marcados como explorados, portanto, execução termina. Como mencionado, a fila representa a ordem em que os vértices foram explorados, denominada $\phi$, ou seja, cada posição do vetor $\phi$ representa uma ordem de compra específica e a sequência em que foram adicionadas representa a relação entre ordens de compra, levando em consideração o número de produtos em comum.

A Tabela 4.2 exemplifica a construção da fila de acordo com a execução demonstrada na Figura 4.2.

O Algoritmo 1 é o pseudocódigo da BFS modificada aplicada ao MORP desenvolvida neste trabalho. Como mencionado anteriormente, a BFS recebe um grafo G formado por ordens de compra e suas relações e retorna o vetor $\phi$ como sequencia resultante da BFS.



(a) (b) (c)

Figura 4.2: Busca em Largura aplicada ao MORP

Tabela 4.2: Construção da fila de acordo com a execução da BFS.

| Iteração | Fila |
|---|---|
| 1 | $i_2$ |
| 2 | $i_2, i_6$ |
| 3 | $i_2, i_6, i_1, i_5, i_3, i_4$ |

**Algoritmo 1:** Pseudocódigo da BFS modificada.

**Entrada:** Grafo de ordens $G$.<br>
$v \leftarrow f(G)$;<br>
Marque $v$ como visitado;<br>
$\phi \leftarrow \emptyset$;<br>
Fila $Q \leftarrow v$;<br>
**enquanto** $Q \neq \emptyset$ **faça**<br>
    $T \leftarrow \emptyset$;<br>
    **para todo** *vértice $w$ vizinho de $v$* **faça**<br>
        **se** *$w$ é marcado como não explorado* **então**<br>
            Insira $w$ ao final de $T$;<br>
            Marque $w$ como exlorado;<br>
        **fim**<br>
    **fim**<br>
    Ordena $T$ descendentemente pelo peso da aresta $\{v, w\}$;<br>
    Insira todos os elementos de $T$ no final de $Q$;<br>
    $\phi \leftarrow v$;<br>
    Remova $v$ de $Q$;<br>
**fim**

O Algoritmo 2 é o pseudocódigo da função que recebe o vetor de ordens de compra $\phi$, resultado da BFS, e retorna o vetor $\pi$ que representa a ordem que os produtos serão produzidos na linha de produção.

**Algoritmo 2:** Pseudocódigo da construção do vetor $\pi$

**Entrada:** Lista de ordens $\phi$.
$A \leftarrow \emptyset$;
**para cada** *ordem* $i \in \phi$ **faça**
  $A \leftarrow A \cup i$;
  **se** $c(p_i) \subseteq A$ **então**
    Insira $p_i$ ao final de $\pi$;
  **fim**
**fim**

## 4.3 Sequenciamento dos Produtos

Para obter bons resultados no sequenciamento de produtos é necessário observar tanto os produtos quanto as ordens de compra, portanto, o vetor $\phi$ auxilia na criação do vetor de permutações $\pi$. A construção de $\pi$ é feita como descrito a seguir. O vetor $\phi$ é percorrido analisando-se iterativamente cada ordem de compra na sequência em que são listados. Esta análise consiste em verificar quais produtos estão presentes em cada ordem. Uma vez que todas as ordens de compra de um produto específico tenham sido analisadas, este produto é então adicionado ao final do vetor de solução $\pi$. Devido ao fato de todas as ordens de compras estarem presentes no vetor $\phi$ é garantido que todos os produtos necessários para atender todas as ordens de compra estarão no vetor $\pi$.

A Tabela 4.3 demonstra a construção de $\pi$ a partir de $\phi$, gerado pela execução da BFS exemplificada na Figura 4.2. Na primeira iteração, o vetor $\pi$ está vazio pois não há produto exclusivamente na ordem de compra $i_2$. Na segunda iteração, o vetor $\pi$ recebe o produto $j_3$ pois o mesmo está presente nas ordens de compra $i_2$ e $i_6$ simultaneamente. Na terceira iteração, o vetor $\pi$ não recebe nenhum novo produto pois não há produto que atenda o critério de seleção. Na quarta iteração o vetor $\pi$ recebe os produtos $j_2$ e $j_5$. O produto $j_2$ está presente no conjunto de ordens $[i_1, i_5, i_6]$ e o produto $j_5$ está presente no conjunto de ordens $[i_5, i_6]$, ambos já analisados. O mesmo processo é repetido para as ordens de compra restantes, e por temos $\pi = [j_3, j_2, j_5, j_1, j_6, j_4]$.

Tabela 4.3: Construção do vetor solução $\pi$ a partir do vetor $\phi$.

| Iteração | $\phi$ | $\pi$ |
|---|---|---|
| 1 | $[i_2]$ | $[\emptyset]$ |
| 2 | $[i_2, i_6]$ | $[j_3]$ |
| 3 | $[i_2, i_6, i_1]$ | $[j_3]$ |
| 4 | $[i_2, i_6, i_1, i_5]$ | $[j_3, j_2, j_5]$ |
| 5 | $[i_2, i_6, i_1, i_5, i_3]$ | $[j_3, j_2, j_5, j_1, j_6]$ |
| 6 | $[i_2, i_6, i_1, i_5, i_3, i_4]$ | $[j_3, j_2, j_5, j_1, j_6, j_4]$ |

# Capítulo 5

# Experimentos

Os experimentos computacionais foram realizados em um computador com processador *Intel i5 Quad Core* de 2.27 GHz com 3 GB RAM sob o sistema operacional Windows 7. O código do método proposto foi escrito em C++, compilado com g++ 4.4.1 e a opção de otimização -O3. Comparam-se o comprimento dos espalhamento de ordens obtidas e o tempo de execução da heurística proposta com o estado da arte.

## 5.1 Instâncias para o MORP

As instâncias utilizadas são as mesmas utilizadas por Fink e Voß (1999), gentilmente cedidas pelo autor. As instâncias foram geradas aleatoriamente. O parâmetro $v$ representa o nível de densidade das matrizes, quanto menor o $v$ mais densa é a instância. O parâmetro $m$ representa o nÚmero de linhas, ou seja, número de ordens de compra da instância. Há 4 conjuntos de instâncias para $m \approx 50$ e $v = \{0,25; 0,5; 0,75; 1\}$ com 100 instâncias cada, e 4 conjuntos de instância para $m \approx 60$ com 100 instancias para $v = \{0,25; 0,5\}$, 98 instâncias para $v = 0.75$ e 99 instâncias para $v = 1$ (soma total 797 instâncias).Para todas as instâncias o número de colunas é aproximado ao número de linhas ($m$).

A Tabela 5.1 expõe os resultados obtidos com a BFS aplicada ao MORP (*BFS-MORP*) comparados ao resultados do *Cheapest Insertion* (*Chins*) e Busca Tabu Reativa com 5000 iterações (*TSRE500*). A coluna $m$ representa a dimensão das instâncias, a coluna $v$ a densidade das instâncias, a coluna $S$ o valor médio dos espalhamentos de ordem encontrado pelos métodos citados, a coluna *gap* representa a distância, em porcentagem, entre o melhor resultado e os demais (calculado como *100×(valor da solução obtida - referência) /referência*), a coluna $T$ representa o tempo de execução de cada método.

Analisando a Tabela 5.1 observa-se que o método *BFS-MORP* obteve resultados nada razoáveis alcançando *gap* médio de 146,9% variando de 44,7% a 383,4%. Os melhores resultados foram alcançados para instâncias de densidade média ($v = 0,50$), apresentando *gap* de 46,5% e 44,7% para as instâncias com número de colunas igual a 50 e 60 respectivamente.

Tabela 5.1: Comparação com o melhor e o pior resultado de Fink e Voß (1999).

| $m$ | $v$ | *Chins* | | | *TSRE5000* | | | *BFS-MORP* | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $S$ | *gap* | $T$ | $S$ | *gap* | $T$ | $S$ | *gap* | $T$ |
| 50 | 0,25 | 19,69 | 40,2 | 0,2 | 14,04 | 0,0 | 142,6 | 24,92 | 77,5 | 0,2 |
| 50 | 0,50 | 17,65 | 42,0 | 0,2 | 12,43 | 0,0 | 79,5 | 18,21 | 46,5 | 0,3 |
| 50 | 0,75 | 8,54 | 75,4 | 0,2 | 4,87 | 0,0 | 58,6 | 9,33 | 91,6 | 0,3 |
| 50 | 1,00 | 2,53 | 83,3 | 0,2 | 1,38 | 0,0 | 35,1 | 6,25 | 352,9 | 0,2 |
| 60 | 0,25 | 23,44 | 42,8 | 0,4 | 16,41 | 0,0 | 159,5 | 29,99 | 82,7 | 0,2 |
| 60 | 0,50 | 21,03 | 43,9 | 0,4 | 14,61 | 0,0 | 142,6 | 21,14 | 44,7 | 0,5 |
| 60 | 0,75 | 9,45 | 77,3 | 0,4 | 5,33 | 0,0 | 98,2 | 10,47 | 96,4 | 0,3 |
| 60 | 1,00 | 2,90 | 100,0 | 0,4 | 1,45 | 0,0 | 58,9 | 7,01 | 383,4 | 0,3 |

O método *TSRE500* obteve os melhores resultados e o método *Chins* obteve os piores resultados dentre os métodos propostos por Fink e Voß (1999). O método *BFS-MORP* alcançou resultados piores do que os resultados do método *Chins*, portanto, para trabalhos futuros será implementado o melhoria da heurística com objetivo de obter soluções mais próximas do ótimo.

# Capítulo 6

# Planejamento

A Tabela 6.1 apresenta as informações sobre o planejamento para as próximas atividades.

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 os conceitos referentes ao Problema de Minimização de Espalhamento de Ordens. Também foi revisada a literatura referente a este problema, detalhado as abordagens empregadas e os resultados obtidos.

O problema foi formalmente descrito e uma implementação preliminar de heurística foi apresentada. Com base na pesquisa realizada foi possível compreender o problema e gerar uma solução inicial, aplicando uma busca em largura na representação do problema utilizando grafos. A partir dos experimentos computacionais preliminares, as soluções iniciais geradas obtiveram resultados ruins. Na continuidade deste trabalho, será realizada a melhoria da heurística proposta com o intuito de atingir resultados próximos ao ótimo já encontrado e a realização de novos experimentos computacionais com outros conjuntos de instâncias.

# Referências Bibliográficas


Cheng, T.; Diamond, J. e Lin, B. (1993). Optimal scheduling in film production to minimize talent hold cost. *Journal of Optimization Theory and Applications*, 79(3):479–492.

Fink, A. e Voß, S. (1999). Applications of modern heuristic search methods to pattern sequencing problems. *Computers Operations Research*, 26(1):17 – 34.

Foerster, H. e Wäscher, G. (1998). Euro best applied paper competition simulated annealing for order spread minimization in sequencing cutting patterns. *European Journal of Operational Research*, 110(2):272 – 281.

Garey, M. R. e Johnson, D. S. (2002). *Computers and intractability*, volume 29. wh freeman New York.

Gilmore, P. e Gomory, R. (1966). The theory and computation of knapsack functions. *Operations Research*, 14(6):1045–1074.

Kim, B.-I.; Ki, Y.; Son, D.; Bae, B. e Park, J.-S. (2016). An algorithm for a cutting problem in window frame production. *International Journal of Production Research*, 54(14):4327–4339.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal, The*, 44(10):2245–2269.

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. G. (1988). An application of travelling-salesman routines to solve pattern-allocation problems in the glass industry. *Operational Reseach Society Ltd*, 39(3):249 – 256.

Smith, B. M. (2003). Constraint programming in practice: Scheduling a rehearsal. *Re-search Report APES-67-2003, APES group.*

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.