

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO  
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO / ESCOLA DE ENGENHARIA

O Problema de Sequenciamento de Caminhões numa  
Estação de *Crossdocking* com Duas Máquinas:  
Formulação Indexada no Tempo, Relaxação  
Lagrangeana e Geração de Colunas

Marcelus Fabri Lima

**Orientador:** Prof. Dr. Martín Gomez Ravetti

Belo Horizonte  
Março, 2014

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO  
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO / ESCOLA DE ENGENHARIA

**O Problema de Sequenciamento de Caminhões numa  
Estação de *Crossdocking* com Duas Máquinas:  
Formulação Indexada no Tempo, Relaxação  
Lagrangeana e Geração de Colunas**

**Marcelus Fabri Lima**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais como requisito para a obtenção do título de Mestre em Engenharia de Produção.

**Área de Concentração:** Produção e Logística

**Linha de Pesquisa:** Modelos e Algoritmos de Produção e de Redes

**Orientador:** Prof. Dr. Martín Gomez Ravetti

**Belo Horizonte**

**Março, 2014**

L732p

Lima, Marcelus Fabri.

O problema de sequenciamento de caminhões numa estação de *Crossdocking* com duas máquinas [manuscrito] : formulação indexada no tempo, relaxação Langrangeana e Geração de Colunas / Marcelus Fabri Lima. – 2014.

x, 72 f., enc.: il.

Orientador: Martin Gomez Ravetti.

Dissertação (mestrado) Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f. 70-72.

1. Engenharia de produção - Teses. 2. Logística empresarial - Teses. I. Ravetti, Martin Gomez. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 658.5(043)



## FOLHA DE APROVAÇÃO

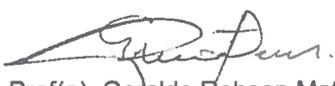
### **O Problema de Sequenciamento de Caminhões em uma Estação de Crossdocking Com Duas Máquinas: Formulação Indexada no Tempo, Relaxação Lagrangeana e Geração de Colunas**

**MARCELUS FABRI LIMA**

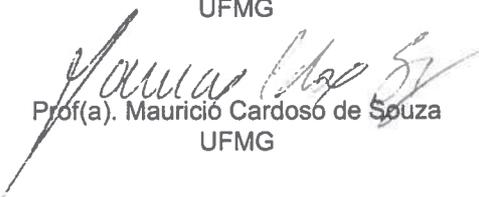
Dissertação submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA DE PRODUÇÃO, como requisito para obtenção do grau de Mestre em ENGENHARIA DE PRODUÇÃO, área de concentração PESQUISA OPERACIONAL E ENGENHARIA DE MANUFATURA, linha de pesquisa Modelos e

Aprovada em 31 de março de 2014, pela banca constituída pelos membros:

  
Prof(a). Martin Gomez Ravetti - Orientador  
UFMG

  
Prof(a). Geraldo Robson Mateus  
UFMG

  
Prof(a). Thiago Ferreira de Noronha  
UFMG

  
Prof(a). Maurício Cardoso de Souza  
UFMG

Belo Horizonte, 31 de março de 2014.



## ATA DA DEFESA DA DISSERTAÇÃO DO ALUNO MARCELUS FABRI LIMA

Realizou-se, no dia 31 de março de 2014, às 14:00 horas, sala 1010, da Universidade Federal de Minas Gerais, a 276ª defesa de dissertação, intitulada *O Problema de Sequenciamento de Caminhões em uma Estação de Crossdocking Com Duas Máquinas: Formulação Indexada no Tempo, Relaxação Lagrangeana e Geração de Colunas*, apresentada por MARCELUS FABRI LIMA, número de registro 2012652594, graduado no curso de ENGENHARIA DE PRODUÇÃO, como requisito parcial para a obtenção do grau de Mestre em ENGENHARIA DE PRODUÇÃO, à seguinte Comissão Examinadora: Prof(a). Martin Gomez Ravetti - Orientador (UFMG), Prof(a). Geraldo Robson Mateus (UFMG), Prof(a). Thiago Ferreira de Noronha (UFMG), Prof(a). Mauricio Cardoso de Souza (UFMG).

A Comissão considerou a dissertação:

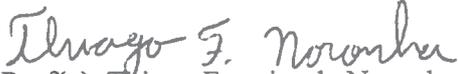
Aprovada

Reprovada

Finalizados os trabalhos, lavrei a presente ata que, lida e aprovada, vai assinada por mim e pelos membros da Comissão.  
Belo Horizonte, 31 de março de 2014.

  
Prof(a). Martin Gomez Ravetti ( Doutor )

  
Prof(a). Geraldo Robson Mateus ( Doutor )

  
Prof(a). Thiago Ferreira de Noronha ( Doutor )

  
Prof(a). Mauricio Cardoso de Souza ( Doutor )

# Resumo

O trabalho trata do problema de sequenciamento denotado  $F2/CD/\sum C_j^2$ . Esse problema é interpretado como um problema de *flowshop* com duas máquinas, com restrições de *crossdocking*, no qual a função objetivo busca minimizar a soma dos períodos de conclusão de todos os jobs que serão trabalhados pela máquina 2. É proposto um modelo de programação inteira com formulação baseada em indexação no tempo e em seguida são propostos e implementados os métodos exatos de relaxação Lagrangeana, Geração de Colunas e dois métodos heurísticos visando obter limites próximos da solução ótima do problema. Os métodos heurísticos buscam auxiliar a resolução da relaxação Lagrangeana e da Geração de Colunas obtendo uma solução viável, que será utilizada como limite superior, e calculando boas sequências de processamento de jobs para a máquina 1, respectivamente. A aplicação desses métodos tem o objetivo de identificar alternativas para solucionar o problema original e avaliar o desempenho de cada método destacado. O problema foi resolvido por meio do software comercial CPLEX e por heurística. Os resultados obtidos mostraram eficiência dos métodos implementados.

**Palavras-Chave:** Programação Linear Inteira, *Flow Shop*, *crossdocking*, Relaxação Lagrangeana, Geração de Colunas.

# Abstract

This work deals with a scheduling problem denoted as  $F2/CD/\sum C_j^2$ . That problem can be viewed as a flowshop problem with two machines and crossdocking constraints. The objective function aims to minimizing the sum of all job's conclusion dates, that are processed on machine 2. A time-indexed integer programming formulation is proposed and implemented. Also, are applied the exact methods Lagrangean relaxation, Column Generation and two heuristics methods, which target is get good upper bounds, close to the problem's optimal solution. The heuristics aim to get good upper bounds values and high quality sequences for the machine 1, providing information to the Lagrangean relaxation and the Column Generation, respectively. The objective of the application of these methods is identifying different alternatives to solve the original problem and evaluate each method's performance. The problem is solved by the commercial software CPLEX and heuristics. Computational results show this strategy is efficient.

**KeyWords:** Integer Linear Programming, Flow Shop, Crossdocking, Lagrangean Relaxation, Column Generation.

# Agradecimentos

Agradeço à Deus pelo dom da vida e companheirismo.

Ao professor Martín por suas orientações, que foram além da execução deste trabalho.

Aos meus amigos, colegas e professores que de uma forma ou de outra contribuíram para a realização deste trabalho.

À minha família pelo apoio e incentivo.

Ao meu Pai e minha Mãe pelo exemplo e apoio incondicional

# Sumário

<b>Lista de Figuras</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>x</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização . . . . .	1
1.2 Objetivos . . . . .	5
1.3 Justificativa . . . . .	6
<b>2 Revisão de Literatura</b>	<b>7</b>
2.1 <i>Crossdocking</i> . . . . .	7
2.1.1 Localização do centro de <i>crossdocking</i> . . . . .	8
2.1.2 <i>Layout</i> do centro de <i>crossdocking</i> . . . . .	9
2.1.3 Cadeia <i>crossdocking</i> de fornecimento . . . . .	9
2.1.4 Roteamento de veículos . . . . .	10
2.1.5 Atribuição de docas . . . . .	11
2.1.6 Sequenciamento de caminhões . . . . .	12
2.1.7 Sequenciamento interno de recursos . . . . .	14
2.1.8 Carga e descarga . . . . .	14
2.2 Relaxação Lagrangeana associada ao <i>Crossdocking</i> . . . . .	15
2.3 Geração de Colunas . . . . .	15

---

<b>3</b>	<b>Formulação</b>	<b>17</b>
3.1	Formulação matemática . . . . .	18
3.1.1	Parâmetros de Entrada . . . . .	19
3.1.2	Variáveis de Decisão . . . . .	20
3.1.3	Função Objetivo . . . . .	20
3.1.4	Modelo . . . . .	21
<b>4</b>	<b>Relaxação Lagrangeana</b>	<b>23</b>
4.1	Relaxação Lagrangeana . . . . .	24
4.1.1	Algoritmo do Subgradiente . . . . .	26
4.2	Formulação por Relaxação Lagrangeana . . . . .	28
4.3	Solucionando o Subproblema Lagrangeano . . . . .	30
4.3.1	Subproblema em X . . . . .	30
4.3.2	Subproblema em Y . . . . .	31
4.4	Encontrando um Limite Superior para o Problema . . . . .	33
4.5	Testes Computacionais . . . . .	35
4.5.1	Geração de Instâncias . . . . .	35
4.5.2	Características da máquina para teste . . . . .	36
4.5.3	Testes - Relaxação Lagrangeana . . . . .	37
4.5.4	Análise dos Testes com Relaxação Lagrangeana . . . . .	38
<b>5</b>	<b>Geração de Colunas</b>	<b>40</b>
5.1	Geração de Colunas . . . . .	40
5.1.1	Formulação Dantzig-Wolfe . . . . .	41
5.1.2	Resolvendo a Relaxação da Formulação Dantzig-Wolfe por Geração de Colunas . . . . .	42
5.2	Formulação - Geração de Colunas . . . . .	45
5.2.1	Formulação Problema Mestre Restrito e Subproblema de Precificação . . . . .	45

---

5.2.2	Método de Geração de Colunas - Heurística proposta . . . . .	49
<b>6</b>	<b>Testes Geração de Colunas</b>	<b>53</b>
6.1	Instâncias . . . . .	53
6.2	Descrição Teste 1 - Geração de Colunas . . . . .	54
6.2.1	Resultados Testes 1 - Geração de Colunas . . . . .	55
6.2.2	Análise Resultados Teste 1 - Geração de Colunas . . . . .	57
6.3	Descrição Teste 2 - Heurística e Geração de Colunas com garantia de Otimidade . . . . .	57
6.3.1	Resultados Teste 2 . . . . .	60
6.3.2	Análise Resultados Teste 2 . . . . .	60
6.4	Descrição Teste 3 - Heurística e Geração de Colunas sem garantia de Otimidade . . . . .	60
6.5	Resultados Teste 3 . . . . .	62
6.5.1	Análise Resultados Teste 3 . . . . .	62
<b>7</b>	<b>Conclusão e Trabalhos Futuros</b>	<b>68</b>
	<b>Referências Bibliográficas</b>	<b>70</b>

# Lista de Figuras

1.1	<i>Crossdocking</i> . . . . .	2
1.2	Operações internas no centro de <i>crossdocking</i> . . . . .	3
3.1	Sistema de <i>crossdocking</i> a ser abordado . . . . .	18
3.2	Abordagem de sequenciamento com indexação no tempo . . . . .	21
4.1	Gráfico representativo da Função do dual Lagrangeano $L(\pi)$ . . . . .	26
5.1	Gráfico representativo da convergência do algoritmo de geração de colunas. . . . .	44
5.2	Dinâmica entre PMR e SB . . . . .	48
5.3	Pseudo-Algoritmo heurística proposta . . . . .	51
6.1	Dinâmica Teste Geração de Colunas . . . . .	56
6.2	Pseudo-Algoritmo Teste 2 . . . . .	58
6.3	Dinâmica Teste 2 - Heurística e Geração de Colunas com garantia de Otimalidade . . . . .	59
6.4	Pseudo-Algoritmo Teste 3 . . . . .	61
6.5	Dinâmica Teste 3 - Heurística e Geração de Colunas sem garantia de Otimalidade . . . . .	62

# Lista de Tabelas

4.1	Sumário das Instâncias Geradas para Teste - Relaxação Lagrangeana . . .	36
4.2	Comparação entre Modelo Original, Relaxação Linear e Relaxação Lagrangeana . . . . .	39
5.1	Exemplo de Aplicação da heurística proposta . . . . .	49
5.2	Resultado iteração 1 da heurística proposta . . . . .	50
5.3	Resultado iteração 2 da heurística proposta . . . . .	50
6.1	Descrição Dados Coletados nos Testes . . . . .	54
6.2	Sumário das Instâncias Geradas para Teste - Geração de Colunas . . . .	55
6.3	Comparação entre Formulação Original, Relaxação Linear e Geração de Colunas . . . . .	64
6.4	Comparação entre Formulação Original, Relaxação Linear e Teste 2 . . .	65
6.5	Comparação entre Formulação Original, Relaxação Linear e Teste 3 . . .	66
6.6	Comparação entre Formulação Original, Testes 2 e Teste 3 . . . . .	67

# Capítulo 1

## Introdução

### 1.1 Contextualização

Em redes de distribuição tradicionais, o conceito de centro de distribuição é amplamente adotado por empresas que ofertam uma alta variedade e volume de produtos, geralmente originados de diversas fontes e sendo produzidos ou comprados para revenda. Esses bens são destinados a uma grande quantidade de clientes ou pontos de entrega. Em sistemas de grande escala, seria economicamente impraticável e ineficiente dedicar um carregamento exclusivo de um fornecedor a um cliente. Está cada vez mais comum caminhões viajarem com carga incompleta, pois os clientes estão mudando as características dos pedidos e exigindo entregas em pequenos lotes, o que aumenta o custo unitário de transporte. Somado a esse fato, está o elevado número de possíveis rotas, o que exige uma frota com grande número de veículos de transporte. Essas complicações deram origem aos centros de distribuição clássicos, que são unidades especializadas com a função de consolidar e armazenar produtos de diversas fontes para serem distribuídos a pontos de revenda ou clientes finais.

O papel principal de um centro de distribuição tradicional é estocar grandes quantidades de produtos, possibilitando pequenas entregas a destinos específicos. As operações de armazenagem requerem esforços de recebimento e separação de materiais, alocação e transporte interno de produtos, ocasionando custos de manuseio e acomodação, geralmente relacionados a investimentos em tecnologia da informação, equipamento para transporte, pessoal e espaço físico. Uma vez feita a armazenagem, o outro problema operacional é a atividade de coleta, ou *picking*, dos produtos para entrega. As estratégias de *picking* são estudadas pois afetam diretamente o desempe-

nho desse tipo de centro de distribuição, porque a eficiência da coleta determinará a eficiência de atendimento ao cliente e o nível de serviço. Assim, de forma semelhante às atividades de armazenagem, as atividades de *picking* são responsáveis por custos operacionais elevados, envolvendo manuseio e transporte interno de materiais nesses centros.

Por outro lado, o conceito de *crossdocking* surge com o propósito de buscar alternativas que minimizem alguns dos efeitos negativos descritos no centro de distribuição tradicional. Diferentemente deste centro, um centro de *crossdocking* funciona sem estocagens, apenas armazenamentos temporários para que a carga seja distribuída dos caminhões de chegada para os caminhões de saída. Seu principal objetivo é possibilitar a consolidação de cargas de diferentes tipos em caminhões com carga completa, tornando possível economias em custos de transporte [3]. Essas instalações não realizam as atividades de armazenagem e *picking*, pois a carga recebida por diversos fornecedores é imediatamente preparada para ser transferida para a área de despacho e destinada aos clientes. Os centros de *crossdocking* operam recebendo caminhões completos de diversos pontos de fornecimento, cada veículo é recebido em uma doca específica. Dentro do centro, as cargas são retiradas, separadas, combinadas e recarregadas em caminhões de saída, de acordo com os pedidos específicos dos clientes. Essas caminhões então deixam o centro com carga combinada, composta por produtos de diversos fornecedores, dedicada a um cliente ou destino específico, conforme a figura 1.1:

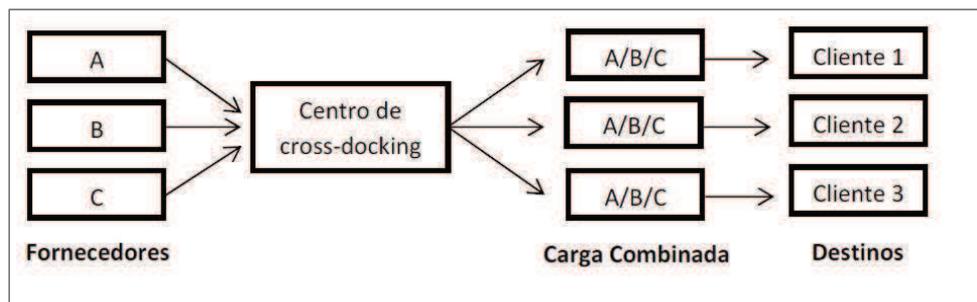


Figura 1.1: *Crossdocking* - Fonte: [30]

Dentro da instalação de *crossdocking*, as operações internas podem ser resumidas em chegada, descarregamento, ordenação, carregamento e despacho, conforme figura 1.2:

A figura 1.2 ilustra o caso particular, o qual estão presentes apenas duas docas em um centro de *crossdocking*. Destaca-se que as letras indicadas representam apenas

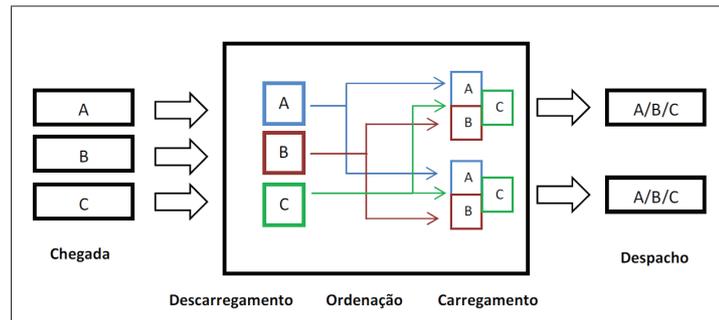


Figura 1.2: Operações internas no centro de *crossdocking* - Fonte: [30]

a origem dos produtos e não a quantidade ou a rotulação dos mesmos.

Segundo EAN (*European Article Numbering*) *International* [1], a utilização do *crossdocking* traz diversas vantagens para o sistema de distribuição, destacando-se:

- Redução dos custos de estoque e eliminação dos problemas operacionais de armazenagem e *picking*.
- Redução do custo unitário de transporte, visto que o sistema opera com carga completa nas caminhões.
- Redução da complexidade de entrega que é realizada de uma só vez com toda a variedade de produtos requerida pelo cliente.
- Aumento da vida de prateleira do produto, uma vez que ele passará menos ou nenhum tempo em estoque intermediário.
- Aumento da rotatividade do centro de *crossdocking*, já que o sistema opera com entregas em menor quantidade e mais frequentes.
- Redução da falta de estoque nas lojas dos clientes, devido ao ressuprimento contínuo e mais frequente proporcionado pela rede.
- Redução do estoque total em toda a cadeia de suprimentos, na qual o produto passa a fluir sem ser estocado.

Essas vantagens fazem do *crossdocking* uma importante estratégia logística e tem ganhado crescente atenção no cenário competitivo mundial. É cada vez mais comum, clientes exigirem tempos de entrega reduzidos e precisão das mesmas, impondo ao operador logístico flexibilidade e agilidade na programação e nas operações de distribuição. Diante desse cenário, a adoção de estratégias que reduzam custos operacionais

e viabilizam tal flexibilidade é fundamental para que a empresa logística seja competitiva no mercado, e a estratégia de *crossdocking* tem se mostrado adequado nesse sentido. Empresas como Wal-Mart, UPS e Toyota [5] já apresentaram casos de sucesso nos quais a adoção do *crossdocking* resultou em vantagens competitivas, visto que essas companhias possuem custos elevados de distribuição. Nesses mercados, também caracterizado pela alta competição, a redução de custos operacionais pode refletir em uma redução significativa do preço dos produtos ao consumidor final, criando assim vantagem competitiva sobre os concorrentes. No Brasil, a utilização do *crossdocking* pelas empresas que operam centros logísticos, incluindo varejistas, siderúrgicas e distribuidores em geral ainda é pequena, o que representa um grande potencial de estudo e trabalho visando a melhora da eficiência logística dessas empresas.

Por outro lado, apesar das vantagens obtidas com essa nova estratégia logística, alguns cuidados devem ser tomados. Para que a cadeia de suprimentos funcione de maneira eficiente, todos os participantes da cadeia devem estar envolvidos na busca da excelência no funcionamento do sistema, tanto na gestão de materiais, quanto na gestão da informação, por exemplo, conhecer a hora de chegada e partida de caminhões, a localização da frota e as informações da carga. A utilização dessas informações permite que as instalações planejem suas operações antes do recebimento das mercadorias, além de permitir também o planejamento e gerenciamento de sua capacidade. Segue abaixo algumas das ferramentas utilizadas na gestão de informações em cadeias de suprimentos [1]:

- *Electronic Data Interchange*: Intercâmbio de documentos e informações entre empresas.
- Códigos de barra.
- Rádio Frequência: Utilizada para identificar, localizar e determinar a condição dos itens.
- *Warehouse Management System*: Sistema de gestão por software com alto nível de controle e acuracidade do inventário.

O centro de *crossdocking*, assim como centros de distribuição tradicionais, introduz um ponto adicional entre a origem e o destino, em contraste com entregas realizadas ponto-a-ponto entre fornecedor e cliente. Carregamentos ponto-a-ponto são inviáveis economicamente em redes de distribuição de larga escala, entretanto, é a

maneira mais rápida com que as entregas cheguem ao cliente final, assumindo que quanto mais pontos intermediários, maior será o tempo necessário para concluir todo o trajeto entre fornecedor e cliente. Ao adicionar um novo ponto de descarga e carga na cadeia, o processo de distribuição se torna mais lento, gerando também uma quantidade considerável de dupla manipulação de produtos. Dessa forma, é necessário que se implemente métodos eficientes de transferência de carga em um centro de *crossdocking*, nos quais as descargas e cargas são eficientemente sincronizadas de forma a manter o nível de estoque baixo e assegurar as entregas a tempo [6]. Essa necessidade deu origem aos problemas de sequenciamento em um centro de *crossdocking*, que é o escopo deste trabalho. Esse problema consiste na determinação da sequência na qual caminhões são descarregados nas docas de entrada e carregados nas docas de saída, de forma a otimizar uma função objetivo definida previamente, como minimizar o tempo total de conclusão das tarefas (*makespan*), entre outras. Essa é uma decisão de curto prazo que surge no dia-a-dia da operação do centro de *Crossdocking*, e que impacta diretamente no nível de serviço ao cliente final.

## 1.2 Objetivos

De acordo com [6], o problema de sequenciamento de caminhões associado ao contexto de centro de *crossdocking*, procura decidir a sequência de processamento dos caminhões nas docas deste centro. Além disso, é destacado a importância que os processos sejam feitos de forma rápida e pontual, devido ao fato de que múltiplos pequenos carregamentos são processados em um intervalo de tempo que pode ser curto. A quantidade de docas a serem trabalhadas pode variar de acordo com o cenário trabalhado, não havendo uma quantidade fixa. Além disso, pode-se extrapolar o problema para mais de um centro de *crossdocking*, para problemas que envolvam, por exemplo, roteirização de veículos, localização de facilidades, entre outros.

O problema tratado nesse trabalho é definido da seguinte forma: em um centro de *crossdocking* na qual chegam  $n$  caminhões, cada um carregado com um tipo de produto, deve-se descarregar os produtos desses caminhões e carregá-los em  $m$  caminhões de saída, estes são responsáveis por carregar vários tipos de produtos para destinos específicos. Nessa abordagem, a doca de entrada é considerada como uma máquina (M1) que realiza a operação de descarregamento, os caminhões de chegada são jobs que devem ser processados em M1, e a doca de saída é considerada como uma máquina (M2) que realiza a operação de carregamento, os caminhões de saída são jobs

que devem ser processados por M2. Assim, um determinado caminhão de saída deixa o centro de *crossdocking* quando seu processamento pela máquina 2 estiver concluído. Com relação ao número de docas no centro, considera-se nesse trabalho que existem duas, uma doca de entrada e outra de saída. Dessa forma, o problema é sequenciar o descarregamento dos caminhões de chegada e o carregamento dos caminhões de saída de forma a minimizar o tempo total de conclusão das operações.

Assim, o objetivo desse trabalho é propor um algoritmo para resolver um problema de sequenciamento de caminhões em um centro de *crossdocking* que minimize o tempo total de conclusão das tarefas. É proposto um modelo de programação inteira com formulação baseada em indexação no tempo e em seguida são propostos e implementados dois métodos visando obter limites próximos da solução ótima do problema. Os métodos propostos são: Relaxação Lagrangeana e Geração de Colunas. A aplicação desses métodos tem o objetivo de identificar alternativas para solucionar o problema original e comparar o desempenho de cada método destacado.

No capítulo seguinte 2, é apresentada a revisão de literatura sobre o tema *crossdocking*. Em seguida, o modelo desenvolvido, e tomado como base para execução do trabalho, é discutido no capítulo 3. Após a apresentação do modelo, é abordado no capítulo 4 o método de relaxação Lagrangeana, mostrando a forma como a relaxação foi trabalhada e é feita a comparação entre o desempenho da relaxação linear e a relaxação Lagrangeana. No capítulo seguinte 5, o método de geração de colunas é exposto e a formulação obtida após a aplicação dessa técnica é apresentada, assim como a heurística proposta nesse trabalho. A descrição da estruturação dos testes, os resultados obtidos e as análises dos mesmos são expostos no capítulo 6. Por fim, no capítulo 7 é apresentada as conclusões obtidas por meio desse estudo.

## 1.3 Justificativa

Esse trabalho se justifica pela sua importância do ponto de vista prático e também acadêmico. Do ponto de vista prático devido ao impacto positivo que a abordagem de centros de *crossdocking* podem trazer para um sistema logístico, como abordado na Seção 1.1. Sob o ponto de vista acadêmico, o trabalho é justificado pela análise e comparação dos métodos que são aplicados, além de procurar solucionar problemas com instâncias maiores do que aquelas já solucionadas na literatura.

# Capítulo 2

## Revisão de Literatura

### 2.1 *Crossdocking*

Boysen e Fliedner [6] e Belle et al. [5] focam na revisão dos trabalhos presentes na literatura que possuem o tema *crossdocking* como foco principal. Esses artigos descrevem características próprias dos sistemas de *crossdocking*, associam trabalhos de diversos autores com problema de sequenciamento de caminhões e categorizam tais trabalhos em duas classes: Estratégicas ou táticas, que lidam com decisões a longo prazo, e operacionais ou técnicas, que trabalham com decisões a curto prazo. A diferença entre as duas revisões está no fato de Boysen e Fliedner [6] considerarem não apenas as características de um sistema *crossdocking* como Belle et al. [5] fizeram, mas também consideraram fazer um paralelo entre problemas de sequenciamento e o sistema discutido. Segundo os autores destacados acima, estações de *crossdocking*, assim como unidades de produção, enfrentam problemas envolvendo tomada de decisões desde o nível estratégico até o nível operacional. Esses problemas podem ser alocados de acordo com a seguinte classificação, ordenada do nível estratégico ao operacional:

- Localização do centro de *crossdocking*
- *Layout* do centro de *crossdocking*
- Cadeia *crossdocking* de fornecimento
- Roteamento de veículos
- Atribuição de docas

- Sequenciamento de caminhões
- Sequenciamento interno de recursos
- Carga e descarga

A seguir serão descritos os trabalhos relacionados aos tópicos acima, baseando-se nos trabalhos do autores citados.

### 2.1.1 Localização do centro de *crossdocking*

A decisão de onde instalar um ou mais centros de *crossdocking* faz parte de um conjunto de decisões estratégicas que compõem a elaboração da cadeia de abastecimento. O custo de instalação de um centro é uma função composta pelo custo fixo de instalação e um custo variável por unidade carregada. Dessa forma, trabalhos que lidam com problemas de localização de facilidades e que abordam a forma pela qual os bens devem fluir pela cadeia são relevantes nesse contexto. Boysen e Flidner [6] citam trabalhos de Campbell [8] e Klose e Drexler [17], que focam no problema de localização de facilidades, investigando a introdução de pontos intermediários em redes de distribuição. Dentre os autores que incorporam a questão de distribuição dos bens, além da localização, estão Gümüs e BookBinder [15], que modelam a distribuição de bens dos fornecedores aos clientes via *crossdocking*, porém nesse trabalho carregamentos diretos são permitidos e diferentes tipos de produtos são considerados (*multicommodity*). Em [15], os autores apresentam modelos para redução do custo total em três tipos de redes, cuja diferenciação está na variação da quantidade de fornecedores, produtos e clientes trabalhados. Segundo Belle et al. [5], em [15] um programa inteiro misto é modelado para solucionar o problema e foi concluído que o número ótimo de estações de *crossdocking* é uma função crescente da razão entre o custo fixo do caminhão sobre o custo fixo de instalação de um centro. Sung e Song [32] consideraram o mesmo problema, porém carregamentos diretos não são permitidos. Dada a demanda, o objetivo do modelo de programação inteira elaborado é calcular quantos terminais instalar e quantos veículos devem ser usados em cada centro de forma a minimizar os custos associados. O problema é NP-difícil e por isso o algoritmo de busca tabu foi utilizado para auxiliar na busca de melhores resultados. Jayaraman e Ross [16] utilizaram o mesmo algoritmo de busca tabu e também *Simulated Annealing* para construir heurísticas que auxiliam o modelo com formulação inteira a solucionar instâncias compostas

por um fornecedor, várias famílias de produtos, múltiplos centros de *crossdocking* e clientes que demandam vários tipos de produtos.

### 2.1.2 *Layout do centro de crossdocking*

Outra abordagem está relacionada ao *layout*, que é a forma e dimensão de um centro, assim como a forma e a dimensão de seus componentes internos [5]. Bartholdi e Gue [4] buscam determinar qual o melhor formato para uma estação de *crossdocking*. A conclusão é que o melhor formato depende do tamanho da instalação e do padrão de fluxo de materiais. Generalizando, à medida que a instalação cresce, os formatos mais eficientes são I, T e X, sucessivamente. Os autores indicam que o design I é mais apropriado para estações com 150 portas ou menos. Para mais de 200 portas, o X é o mais indicado e o formato T pode ser visto como uma fase intermediária. Apesar de os dois últimos apresentarem maior centralidade, eles possuem menor eficiência na execução das tarefas devido ao maior número de quinas. Gue e Kang [14] utilizam a simulação para estudar o comportamento das filas presentes em sistemas com um ou dois estágios de armazenamento. Os resultados indicam que para um estágio é melhor ter mais filas curtas do que poucas filas longas e que a configuração de um estágio é mais produtiva que o segundo modelo.

### 2.1.3 *Cadeia crossdocking de fornecimento*

Existem autores que trabalham com centros de *crossdocking*, porém de forma integrada, ou seja, a forma como esses centros se interagem. O objetivo é determinar o fluxo de produtos na cadeia, de forma a minimizar os custos durante o atendimento da demanda. Lim et al. [22] lidam com o problema de transbordo, cujo objetivo é identificar o fluxo de custo mínimo que atenda a demanda e as restrições de capacidade. Nesse problema, fornecedores, terminais intermediários e clientes são representados por diferentes nós, que são ligados por arcos, que possuem limitações de capacidade e custos associados. O modelo proposto pelos autores procura minimizar o estoque presente na rede. Além disso, considera-se janelas de tempo de fornecedores e clientes e custos de manutenção de estoque. Caso múltiplas partidas e entregas sejam permitidas dentro de uma janela de tempo, os autores concluem que uma rede de tempo-estendida (*time - expanded - network*) pode ser usada para formular o problema como um problema de fluxo de custo mínimo, que pode ser solucionado em tempo polinomial. Para os

demais casos, os autores provam que o problema é NP-difícil. Para o caso no qual somente uma partida ou entrega é permitida durante a janela de tempo e a chegada e partida de produtos são fixas, um algoritmo genético foi desenvolvido por Miao et al. [25]. Os resultados apresentados tiveram melhores resultados (qualidade da solução e tempo computacional) quando comparados a solução da formulação inteira do problema resolvida pelo solver CPLEX.

### 2.1.4 Roteamento de veículos

Muitas vezes as cargas destinadas a um centro de *crossdocking* necessitam ser recolhidas e entregues em diferentes pontos e, depois de passar pelo centro, ser entregues em diferentes localizações. Assim, ambas as etapas podem ser vistas como um problema de roteamento de veículos, que associado ao contexto de *crossdocking*, é objeto de trabalho de alguns autores. Lee et al. [19] foram os primeiros a dar ênfase nessa área que aborda roteamento de veículos e *crossdocking* simultaneamente. O foco dos autores é identificar uma sequência de roteamento ótima para coleta e entrega, que minimize a soma dos custos de transporte e os custos fixos dos veículos. Os autores apresentam um modelo de programação inteira para o problema, que, associado ao algoritmo de busca tabu é proposto para solucionar o problema. Wen et al. [34] propõem um modelo, cujo objetivo é minimizar a distância total percorrida pelos caminhões com restrições de janela de tempo. Cada veículo parte do centro e deve coletar cargas em fornecedores específicos, retornar ao centro de *crossdocking* e, após o transbordo de cargas, executar uma rota de entrega de produtos a clientes específicos, respeitando a janela de tempo de cada um dos pontos de coleta ou entrega. É proposto pelos autores um modelo de programação linear inteira mista e uma heurística baseada em busca tabu, que pode resolver problemas com até 200 nós alcançando resultados a 5% do ótimo em um curto intervalo de tempo. Ainda sobre a questão de roteamento de veículos, Morais [26] abordou o Problema de Roteamento de Veículos com *crossdocking* (VRPCD), que procura em minimizar os custos de transporte ao se coletar produtos em diversos fornecedores e entregá-los aos consumidores por um único centro, onde o processo de consolidação é realizado. Nesse problema, os processos de coleta e entrega são dependentes, uma vez que é permitido realizar a transferência de produtos entre os veículos no centro. Para tratar o problema foram apresentadas três heurísticas construtivas, uma baseada na metaheurística ILS (*Iterated – LocalSearch*), chamada ILS-VRPCD, e a metaheurística SPILS-VRPCD, que integra ILS-VRPCD e um modelo de Programação Linear Inteira, que apoia-se no problema de particionamento de

conjuntos. Os experimentos realizados comprovaram a eficiência dos algoritmos implementados. Além disso, o trabalho expôs que o tempo para realizar o processo de consolidação gera um grande impacto na atividade de roteamento. Tarantilis [9] aborda o problema de roteamento de veículos com múltiplas fontes associado a terminais de *crossdocking*. São avaliados sistemas com configurações abertas e fechadas e analisadas regras de dependência. O autor aborda um dado conjunto de pares de fornecedores e clientes, com demanda conhecida, o objetivo é definir as rotas de menor custo para o transporte dos produtos via *crossdocking*. O método proposto utiliza uma estrutura de busca local adaptada e também o algoritmo de busca tabu. Os experimentos computacionais exibem a eficiência e a efetividade, pois foi melhorado os limites superiores para a maioria das instâncias testadas.

### 2.1.5 Atribuição de docas

Boas alocações de docas podem aumentar a produtividade dos centros de *crossdocking* e reduzir os custos de manuseio [5]. Dessa forma, o problema de alocação de docas procura definir a alocação ótima dos caminhões às docas que serão carregados e descarregados. Assume-se que o número mínimo de docas seja o mesmo número de caminhões, assim cada caminhão será alocado a diferentes docas e aspectos envolvendo tempo não serão levados em consideração. Caso contrário, as docas serão encaradas como recurso escasso e devem ser sequenciadas ao longo do tempo, dando origem ao problema de sequenciamento de caminhões.

De acordo com Boysen e Fliedner [6], a alocação de docas pode ser executada em um horizonte de curto a médio prazo. Belle et al. [5] afirmam que muitos artigos trabalham com essa abordagem sob um horizonte de médio prazo. Portanto, cada doca é atribuída a um destino ou origem específica por cerca de 6 meses. Por outro lado, quando informações sobre a chegada de caminhões são fornecidas antecipadamente, a alocação de docas pode ser feita em um horizonte de curto prazo.

Gue [18] baseou sua análise na alocação de docas examinando o efeito que um sequenciamento planejado (*look-ahead-scheduling*) tem sobre o fluxo de materiais e no *layout* de um centro. O sequenciamento planejado faz oposição ao método "primeiro chegar - primeiro sair", dessa forma, observa-se todos os elementos de um horizonte definido e não apenas o próximo. Assim, para determinar o *layout* ideal, ou seja, que minimize a movimentação de cargas, o autor propõem a procura da solução no espaço de soluções possíveis com auxílio do algoritmo de busca local. O custo do *layout* pode

ser determinado se o fluxo de material resultante é conhecido. É exposto um algoritmo específico chamado *look – ahead*, que testa as soluções por meio de simulação. Os resultados indicam ser possível economizar entre 15% a 20% nos custos. Bozer e Carlo [7] consideram um centro de *crossdocking*, com o mesmo número de docas de entrada e saída, na qual o transbordo de cargas é feito à noite, podendo a atribuição mudar de uma noite para outra. Os autores propõem inicialmente uma formulação baseada no problema de atribuição quadrática retilínea para minimizar o esforço de manipulação de materiais. Essa é tida como ineficiente para solucionar instâncias de larga escala, considerando que é uma decisão de curto prazo e que deve ser tomada a cada dia. Assim, é utilizada uma heurística baseada em *Simulated Annealing*, que obtém a solução ótima em problemas pequenos e apresenta desempenho superior a heurísticas de troca de pares existentes na literatura para problemas grandes. O impacto do formato da estação sobre a movimentação de materiais também é analisado, chegando à conclusão que estações mais planas requerem menos esforços.

### 2.1.6 Sequenciamento de caminhões

O problema de sequenciamento de caminhões consiste em decidir onde e quando os veículos devem ser processados. Boysen e Fliedner [6] apresentam uma classificação com alguns trabalhos que abordaram o tema de sequenciamento de caminhões. Tal classificação foi feita baseada principalmente em um esquema sobre problemas determinísticos de sequenciamento de caminhões. Foi utilizada uma notação em tuplas, que é aplicada em sequenciamento de máquinas e problemas de filas. Como os problemas de *crossdocking* apresentam peculiaridades adicionais, como a atribuição de cargas a veículos de saída, a classificação proposta pelos autores inclui também atributos específicos desse tipo de problema. Essas características foram representadas pela notação  $\alpha | \beta | \gamma$ , relacionando-se com a forma de trabalhar das docas, as características operacionais e os objetivos que guiaram a otimização, respectivamente. A revisão apresentada por Belle et al. [5] também aborda a classificação citada anteriormente, porém seu foco principal está na classificação por características dos terminais.

Chen e Lee [10] consideraram o chamado problema de sequenciamento de *flow-shop crossdocking* com duas máquinas, cujo objetivo é sequenciar os caminhões de chegada e saída de forma a minimizar o *makespan* (tempo de conclusão do último job [23]). A modelagem é feita como um problema de *flow shop* com duas máquinas, porém com restrições de precedência, a fim de garantir que o caminhão de saída seja processado

depois que todas as tarefas de seus predecessores sejam concluídas. Além disso, preempção não é permitida, considera-se que todos os caminhões estão disponíveis no início do horizonte de planejamento e que estoque temporário pode ser formado até a chegada do caminhão de saída apropriado. Os autores provam que esse problema é fortemente NP-difícil e apresentam uma heurística baseada no algoritmo de Johnson, que fornece a solução ótima para o problema de *flowshop* com duas máquinas e com o objetivo de minimizar o *makespan* [31], e um algoritmo de *branch – and – bound*. Os resultados mostram que o problema pode resolver instâncias com até 60 caminhões utilizando até 5 minutos de processamento. Chen e Song [11] estendem esse trabalho considerando múltiplas docas de entrada e saída, denominando problema de *crossdocking* híbrido de dois estágios. Nesse caso múltiplos veículos podem ser carregados ou descarregados ao mesmo tempo, considerando que as docas são como máquinas paralelas. Os autores apresentam um modelo de programação inteira mista e propõem heurísticas baseadas no algoritmo de Johnson para solucionar o problema. Em ambos os trabalhos  $\beta = t_j = 0$ , ou seja, o tempo de transbordo é igual a zero. O modelo de Chen e Lee [10] segue a classificação  $E2 \mid t_j = 0 \mid C_{max}$ , ou seja, problema com duas docas exclusivas para carga ou descarga (E2), com o tempo de transbordo das cargas igual a zero ( $t_j = 0$ ) e cuja função objetivo é minimizar o *makespan*. Já Chen e Song [11] modelam o problema como  $E \mid t_j = 0 \mid C_{max}$ , ou seja, considera-se múltiplas docas, destinadas a descarga ou carga, trabalhando em paralelo, portanto a diferença entre [10] e [11] está na forma de trabalhar com as docas.

McWilliams et al. [24] consideram um centro de *crossdocking* com docas que podem ser de entrada ou saída, o tempo de transporte varia de acordo com as docas atribuídas e o objetivo é minimizar o *makespan*. Para solucionar o problema, é utilizado um algoritmo genético combinado com um modelo de simulação. Em termos de notação em tuplas, o modelo é determinado como  $E \mid p_j = p, no - wait, t_{i0} \mid C_{max}$ . Nesse caso  $\beta$  significa que todos os jobs tem o mesmo tempo de processamento, não podem aguardar em estoques intermediários e os tempos de transbordo para cada par de docas é dado como parâmetro.

Já Boysen [27] utiliza a programação dinâmica associada a um grafo direto acíclico. O caminho mínimo nesse grafo corresponde a solução ótima do problema tratado, que consiste em sequenciar caminhões de chegada e saída em um centro de *crossdocking*, no qual cada doca é exclusiva para operações de carregamento ou descarregamento e não é possível armazenar os produtos, o que é comum para bens que necessitam de refrigeração. Assim, o autor apresenta uma formulação, cujo o objetivo

é minimizar o tempo de fluxo, o tempo de processamento e o atraso dos caminhões de saída. Também são apresentadas heurísticas baseadas em *Simulated Annealing* para obter limites para instâncias reais, representadas por mais de 25 caminhões de chegada. Segundo Belle et al. [5], os modelos trabalhados em [27] seguem as seguintes notações  $E|p_j = p, no - wait, t_j = 0 | \sum C_0$  e  $E|p_j = p, no - wait, t_j = 0 | \sum T_o$ , sendo  $\sum C_0$  é o tempo de conclusão e  $\sum T_o$  é a soma dos atrasos dos jobs.

Para problemas com grandes instâncias, Li et al. [21] apresentam um método heurístico chamado Procura por Posições Dependentes do inglês (*Dependency Ranking Search*), que primeiramente elabora uma sequência viável de jobs, respeitando o número de docas disponíveis, e em seguida esses jobs são alocados às docas. O desempenho com relação ao CPLEX é pior para instâncias pequenas, porém para grandes instâncias o CPLEX não é capaz, na maioria dos casos, de identificar soluções em até duas horas. Já a heurística obtém a solução em poucos minutos em 8 a cada 10 instâncias. A notação utilizada nesse caso será  $M | t_{i0} | C_{max}$ , na qual M significa que a mesma doca pode receber caminhões de chegada e saída.

### 2.1.7 Sequenciamento interno de recursos

Li et al. [20] analisaram o problema de sequenciamento de recursos em um centro de *crossdocking* com uma aproximação baseada em *just-in-time*. O problema é modelado como um problema de sequenciamento de máquinas e resolvido com heurísticas, pois é um problema NP-difícil. Os autores propõem duas heurísticas envolvendo algoritmos genéticos, sendo que a primeira heurística calcula soluções próximas a ótima em um intervalo de tempo muito inferior ao tempo utilizado pelo *solver* CPLEX. Já a segunda heurística consegue obter o melhor resultado, porém sob o custo de um alto tempo de processamento computacional.

### 2.1.8 Carga e descarga

Sobre carga e descarga, Boysen e Fliedner [6] indicam que, embora essa atividade influencie o processamento de veículos em um centro de *crossdocking*, não é proveitoso relacionar decisões de carregamento com sequenciamento de caminhões, porque a carga dos caminhões de chegada não é anteriormente conhecida. Além disso, trabalhar com carregamento nos caminhões de saída pode requerer integrar roteamento de veículos, que implica em determinar a sequência de clientes a visitar e, portanto, con-

siderar como será feita a organização da carga no interior do veículo. O que resultaria em um problema com grande complexidade.

Belle et al. [5] destacam que na literatura, às vezes, o número de docas é limitado a apenas uma ou duas. Nesses casos, o foco não é modelar um sistema real de *crossdocking*, mas obter informações importantes estudando modelos simplificados.

## 2.2 Relaxação Lagrangeana associada ao *Crossdocking*

Luh e Hoitomt [28] abordam três problemas de sequenciamento em ambientes de manufatura em ordem crescente de complexidade. Primeiramente, o sequenciamento de jobs, que requerem apenas uma operação, é feito em paralelo. O segundo trata de problemas com múltiplas operações por job e restrições de precedência em paralelo, e por último, o problema de sequenciamento do tipo *job shop* é analisado. Os autores propõem formulações e utilizam a técnica de relaxação lagrangeana para decompor os problemas originais em subproblemas ao nível de job ou operação, que são mais facilmente resolvidos. As soluções obtiveram gaps de otimalidade inferiores a 5%, com cerca de 3 minutos de processamento, para problemas com aproximadamente 30 máquinas e 120 jobs.

## 2.3 Geração de Colunas

Akker et al. [2] estudaram o problema de sequenciamento de uma máquina e avaliam como a decomposição de Dantzig-Wolfe, associada com o método de geração de colunas, pode ser usada para reduzir os problemas causados pelo grande número de variáveis, que pode ser visto como obstáculo para modelos indexados no tempo. Esses modelos, aplicados em problemas de sequenciamento, discretizam o horizonte de planejamento em intervalos de tempo e, em cada intervalo, toma-se a decisão se um job específico é sequenciado em alguma máquina ou não.

Santos et al. [12] propuseram uma formulação de programação inteira e um algoritmo *branch and price* para solucionar um problema de roteamento de veículos com *crossdocking*. Os resultados mostram que a formulação escolhida fornece limites mais fortes do que outras formulações baseadas em fluxos em redes. Além disso, observa-se que o algoritmo *branch and price* produz soluções melhores, em relação à qualidade dos limites superiores e inferiores, quando comparada à solução fornecida pelo algoritmo

*branch and bound* aplicado a um programa linear para um mesmo problema.

Outro tema relacionado é o Problema de *pickup* e coleta com *crossdocking* introduzido por Santos et al. [13], cujo objetivo é reduzir os custos associados às atividades de transporte via *crossdocking*. A proposta é evitar a imposição que todos os veículos devam parar em uma doca, mesmo que esse veículo colete e entregue o mesmo conjunto de produtos, ou seja, não havendo necessidade de manuseio em sua carga, essa é uma imposição típica dos modelos que consideram o sistema *crossdocking*. Foi proposta uma formulação de programação inteira e um algoritmo *branch and price* para modelar e solucionar o problema. Os resultados mostram ser possível que soluções ótimas ou com pequenos gaps reduzem significativamente os custos totais.

## Capítulo 3

### Formulação

Em um centro de *crossdocking* no qual chegam  $n$  caminhões, cada um carregado com um tipo de produto que é demandado por um ou vários clientes, deve-se descarregar os produtos desses caminhões e carregá-los em  $m$  caminhões de saída, estes são responsáveis por carregar vários tipos de produtos para destinos específicos. Um determinado caminhão de saída deixa o centro de *crossdocking* quando seu processamento pela máquina 2 estiver concluído, ou seja, o job foi carregado com os produtos previamente determinados pela matriz de precedência. Assim, com relação ao número de docas na estação, considera-se nesse trabalho que existem duas, uma doca de entrada e outra de saída, que são consideradas como máquina 1 e 2, respectivamente. Dessa forma, o problema é sequenciar o descarregamento dos caminhões de chegada e o carregamento dos caminhões de saída de forma a minimizar o tempo total de conclusão das operações. A figura 3.1 ilustra a explicação acima. No desenho, três caminhões, cada um carregado com um tipo de produto, são processados pela máquina 1, que recebe um caminhão de cada vez. Após descarregar determinado caminhão, a carga deste é organizada de forma a atender os pedidos a serem entregues pelos dois caminhões de saída, que estão aguardando para serem processados e abastecidos pela máquina 2, que também recebe um caminhão de cada vez. Vale destacar que os símbolos presentes nos caminhões representam apenas a origem dos produtos e não a quantidade ou a rotulação dos mesmos. O objetivo é ilustrar como funciona a distribuição de cargas entre os caminhões que chegam e os que partem do centro de *crossdocking*, a quantidade alocada em cada veículo não é importante nesse estudo, apenas se o produto está disponível para ser carregado no caminhão de saída. Apesar da quantidade alocada não ser o foco, destaca-se que a formulação abrange os cenários onde vários produtos são processados em um terminal de *crossdocking*.

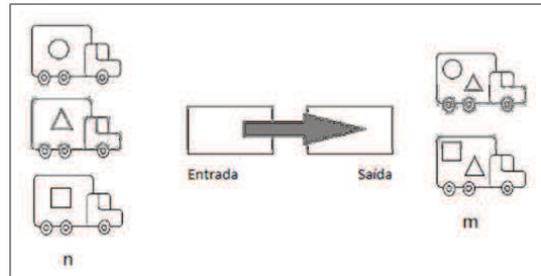


Figura 3.1: Sistema de *crossdocking* a ser abordado - Fonte: [30]

O problema acima será modelado matematicamente através de uma modificação do problema de sequenciamento do tipo *flowshop* clássico. Essa abordagem foi proposta inicialmente por Chen e Lee [10], na qual o problema de sequenciamento de caminhões em um centro de *crossdocking* é definido como uma extensão do problema de *flow shop* com duas máquinas em série (*two-machine crossdocking flow shop problem*). Como já foi explicitado, nessa abordagem, a doca de entrada é considerada como uma máquina (M1) que realiza a operação de descarregamento, os caminhões de chegada são jobs que devem ser processados em M1, e a doca de saída é considerada como uma máquina (M2) que realiza a operação de carregamento, os caminhões de saída são jobs que devem ser processados por M2. A particularidade que torna o problema de *crossdocking* diferente do problema de *flow shop* clássico é a introdução de um conjunto de restrições de precedência denominadas restrições de *crossdocking*.

### 3.1 Formulação matemática

Como a modelagem do problema será baseada em um problema de sequenciamento, será utilizada uma adaptação da notação adotada por Pinedo [23], que consegue capturar a estrutura de diversos modelos considerados na literatura, conforme descrito a seguir:

- $n$ : Número de jobs a serem processados na máquina 1;
- $m$ : Número de jobs a serem processados na máquina 2;
- $p_{ij}$ : Tempo de processamento do job  $j$  na máquina  $i$ ;
- $w_j^i$ : Peso do job  $j$  na máquina  $i$ , que é um fator de prioridade sobre os outros jobs a serem sequenciados naquela máquina.

- $C_j^i$ : Tempo de conclusão do job  $j$  na máquina  $i$ .

O problema será modelado como  $F2|CD|\sum C_j^2$ , ou seja, é um problema de *flowshop* com duas máquinas, com restrições de *crossdocking*, no qual a função objetivo busca minimizar a soma das datas de conclusão de todos os jobs que serão trabalhados pela máquina 2.

O modelo desenvolvido adota uma formulação de indexação no tempo, ou seja, defini-se um horizonte de tempo  $T$ , esse horizonte é discretizado em  $t = 1, 2, \dots, T$  períodos ou datas. Em cada período  $t$ , é tomada a decisão se algum job inicia seu processamento ou não. Essa decisão é tomada com base nas restrições e na função objetivo do modelo. Esse tipo de formulação gera um número muito elevado de variáveis (número de máquinas x número de jobs x  $T$ ), porém gera limites mais fortes, ou seja, limites com valores próximos a solução ótima do problema, do que outros tipos de formulações encontradas na literatura.

Destaca-se que apesar desse trabalho basear-se na proposta apresentada por Chen e Lee [10], não é objetivo realizar a comparação entre os dois trabalhos. Além disso, a função objetivo dos modelos é diferente, inviabilizando que uma comparação seja feita. O modelo discutido nesse trabalho trata a função objetivo como  $\sum C_j^2$ , descrito acima. Já o outro modelo está preocupado em minimizar o tempo de conclusão da última tarefa, ou seja, minimizar o *makespan* ( $C_{max}$ ). Portanto, para que seja feito um paralelo entre os dois trabalhos é necessário que a função objetivo seja modificada.

### 3.1.1 Parâmetros de Entrada

- $p_{ij}$ : Tempo de processamento do job  $j$  na máquina  $i$ ;
- $T$ : Horizonte de tempo considerado - Uma primeira estimativa para o horizonte de tempo é a soma dos tempos de processamento de todos os jobs.
- Para representar as chegadas e saídas da estação de *crossdocking*, dois conjuntos de jobs são criados:

$J_1 = \{j_1^1, j_2^1, \dots, j_n^1\}$  representa os caminhões de chegada, que devem ser processados em M1.

$J_2 = \{j_1^2, j_2^2, \dots, j_m^2\}$  representa os caminhões de saída, que devem ser processados em M2.

As restrições de *crossdocking* são representadas pela condição abaixo:

Para cada job  $J_j^2 \in J^2$ , existe um subconjunto  $S_j$  de jobs em  $J^1$ , tal que  $J_j^2$  só pode ser processado em M2 se todos os jobs em  $S_j$  tiverem sido concluídos em M1. Considera-se que cada elemento do subconjunto  $S_j$  apresente pelo menos um elemento, ou seja, um determinado job  $\{j \in J^2\}$  possui ao menos um job  $\{j \in J^1\}$  como precedente.

### 3.1.2 Variáveis de Decisão

Na formulação indexada no tempo, para cada job  $j$  e para cada período  $t$ , cria-se uma variável que indica se o job  $j$  inicia seu processamento em  $t$ . Para máquinas M1 e M2, tem-se os dois conjuntos de variáveis respectivamente:

- A variável  $x_{jt}$  ( $\forall j \in J^1, \forall t \in T$ ), será igual a 1, se o job  $j$  começar no período  $t$ . Caso contrário, o valor da variável será igual a 0.
- A variável  $y_{jt}$  ( $\forall j \in J^2, \forall t \in T$ ), será igual a 1, se o job  $j$  começar no período  $t$ . Caso contrário, o valor da variável será igual a 0.

Ou seja, o valor da variável  $x$  é igual a 1, se job  $j$ ,  $\forall j \in J^1$ , inicia seu processamento na máquina M1 no período  $t$ . Da mesma forma, a variável  $y$  é igual a 1 se o job  $j$ ,  $\forall j \in J^2$  inicia seu processamento na máquina M2 no período  $t$ .

A Figura 3.2 ilustra a abordagem com indexação no tempo e os valores das variáveis a título de exemplificação. Observa-se que  $J_1^1$  inicia seu processamento no período 0, logo  $x_{10} = 1$ .  $J_2^1$  inicia seu processamento no período 3, logo  $x_{23} = 1$ , e assim por diante. Os demais valores de  $x_{jt}$  e  $y_{jt}$ , nos valores de  $t$  nos quais os jobs não são iniciados, são iguais a 0. Pode-se identificar também os tempos de processamento:  $p_{11} = 3$ ,  $p_{12} = 2$ ,  $p_{21} = 2$  e  $p_{22} = 1$  e os períodos de conclusão:  $C_1^1 = 3$ ,  $C_2^1 = 5$  e  $C_2^2 = 3$ . Onde  $C_j^M$  significa o tempo de conclusão do processamento job  $j$  na máquina  $M$ .

### 3.1.3 Função Objetivo

Conforme descrito anteriormente, a função objetivo do problema é tipo  $\min \sum C_j^2$ , minimizar a soma dos períodos de conclusão de todos os jobs na máquina 2. Na formulação indexada no tempo, considerando que o job  $J_j^2 \in J^2$  inicia seu processamento na máquina 2 no período  $t$ , ou seja,  $y_{jt} = 1$ , sua conclusão é dada pelo período de início  $t$

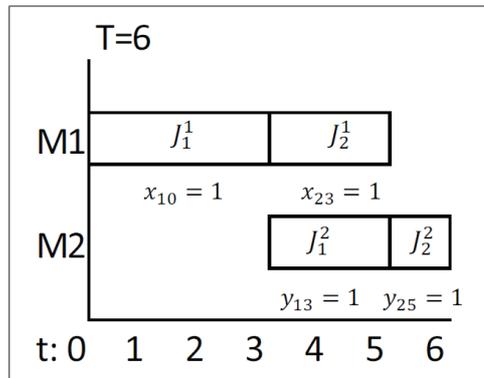


Figura 3.2: Abordagem de sequenciamento com indexação no tempo - Fonte: [30]

mais seu tempo de processamento  $p_{2j}$ . Somando essa expressão para todos os jobs na máquina 2, chega-se à função objetivo do problema:

$$\min \sum_{j \in J^2} \sum_{t=0}^{T-p_{2j}} (t + p_{2j}) y_{jt} \quad (3.1)$$

O leitor pode analisar o modelo matemático completo na próxima seção.

### 3.1.4 Modelo

$$\min Z = \sum_{j \in J^2} \sum_{t=0}^{T-p_{2j}} (t + p_{2j}) y_{jt}$$

sujeito a

$$\sum_{t=0}^{T-p_{1j}} x_{jt} = 1 \quad \forall j \in J^1 \quad (3.2)$$

$$\sum_{t=0}^{T-p_{2j}} y_{jt} = 1 \quad \forall j \in J^2 \quad (3.3)$$

$$\sum_{t=0}^{T-p_{2j}} ty_{jt} - \sum_{t=0}^{T-p_{1k}} (t + p_{1k})x_{kt} \geq 0 \quad \forall j \in J^2 \wedge \forall k \in S_j \quad (3.4)$$

$$\sum_{j \in J^1} \sum_{s=\max(0; t-p_{1j}+1)}^t x_{js} \leq 1 \quad \forall t \in T \quad (3.5)$$

$$\sum_{j \in J^2} \sum_{s=\max(0; t-p_{2j}+1)}^t y_{js} \leq 1 \quad \forall t \in T \quad (3.6)$$

$$x_{jt} \in 0, 1 \quad \forall j \in J^1 \wedge \forall t \in T \quad (3.7)$$

$$y_{jt} \in 0, 1 \quad \forall j \in J^2 \wedge \forall t \in T \quad (3.8)$$

O conjunto de restrições 3.2 diz que cada job  $J_j^1 \in J^1$  deve iniciar seu processamento em um e somente um período dentro do horizonte de planejamento  $T$ . O conjunto 3.3 indica que cada job  $J_j^2 \in J^2$  deve iniciar seu processamento em um e somente um período dentro do horizonte de planejamento  $T$ . O conjunto 3.4 são as restrições de precedência do tipo *crossdocking*. Assim, para cada relação de precedência existente na instância, cria-se uma restrição desse conjunto, na qual o período de início do job  $J_j^2 \in J^2$  deve ser maior ou igual ao período de conclusão de seu precedente  $J_k^1 \in S_j$ . Se o job possuir mais que um precedente, as restrições relativas aos precedentes, que são processados primeiro, tornam-se redundantes e prevalece aquela relativa ao job precedente com maior período de conclusão. As restrições indicadas por 3.5 garantem que um job  $J_j^1 \in J^1$  não inicia seu processamento enquanto outro estiver sendo processado na máquina 1, da seguinte forma: para cada período  $t \in T$ , a restrição verifica se algum job  $J_j^1 \in J^1$  começou a ser processado em um período maior ou igual a  $t - p_{1j} + 1$ . Caso afirmativo,  $J_j^1$  ainda está sendo processado, e nenhum job pode iniciar seu processamento em  $t$ . O conjunto de restrições 3.6 tem o mesmo objetivo que o conjunto anterior 3.5, no entanto é aplicado aos jobs  $J_j^2 \in J^2$ . Finalmente, os conjuntos 3.7 e 3.8 restringem os valores das variáveis  $x$  e  $y$ , que podem assumir apenas valores binários, ou seja, 0 ou 1.

# Capítulo 4

## Relaxação Lagrangeana

Nesse capítulo a aplicação do método da Relaxação Lagrangeana será abordada. Na Seção 4.1 será descrita a teoria na qual o método se apoia. Na seção seguinte 4.2, será descrito como foi feita a aplicação do método e será apresentada a formulação resultante. Na Seção 4.3 será exposto os procedimentos adotados para solucionar o Subproblema Langrangeano, que será dividido em duas etapas. A primeira etapa está contida na Subseção 4.3.1, que contém a descrição para solução do Subproblema em X, cujo objetivo é definir uma sequência para os jobs  $J_j^1 \in \mathcal{J}^1$ . Já a subseção 4.3.2 descreve como é feita a solução do Subproblema em Y, ou seja, objetivo é definir uma sequência para os jobs  $J_j^2 \in \mathcal{J}^2$ . Em seguida, será discutido na Seção 4.4 como é calculado o limite superior para o problema. A seção seguinte 4.5.3 expõem os resultados dos experimentos realizados. Finalizando, a seção 4.5 descreve alternativas para estudos futuros envolvendo o que foi discutido nesse capítulo.

A aplicação desse método foi motivada como uma alternativa para resolver o problema, visto que o modelo criado é indexado no tempo e para grandes instâncias apresenta um grande número de variáveis, o que dificulta a resolução do problema, como é exposto na Seção 4.5.3. Além disso, procura-se verificar os ganhos que o método pode alcançar com relação ao tempo de processamento para obtenção de um limitante inferior, em comparação com a relaxação linear, e também com relação aos ganhos que os valores do limitante superior obtidos com a aplicação da heurística lagrangeana podem gerar.

## 4.1 Relaxação Lagrangeana

A definição a seguir pode ser encontrada em [33]. Considere o seguinte problema de minimização:

$$z = \min\{cx : x \in X\}$$

Suponha que no exemplo acima, o conjunto de restrições  $X$  defina o subconjunto  $Z$ , de forma que ( $X \subset Z$ ). Assim, o problema acima pode ser escrito da seguinte forma:

$$z = \min\{cx : Dx \geq d, Bx \geq b, x \in Z_+^n\} \quad (4.1)$$

O subconjunto  $Z = \{x \in Z_+^n : Bx \geq b\}$  é definido por restrições que são mais tratáveis e, por isso, é mais fácil obter sua solução ótima, pois exige pouco tempo de processamento e memória computacional para ser solucionado. A base comparativa é em relação ao conjunto de restrições  $Dx \geq d$ , que são tidas como complicantes, pois exigem maior tempo e memória computacional para serem resolvidos do que o outro conjunto citado anteriormente. Assim, subconjunto  $Y = \{x \in Z_+^n : Dx \geq d\}$  representa as restrições consideradas como complicantes. Na prática, significa que o problema  $\min\{cx : x \in Z\}$  é solucionado de forma mais rápida do que o problema  $\min\{cx : x \in Y\}$ .

A Relaxação Lagrangeana é um procedimento que explora a otimização sobre um conjunto simplificado. Seu objetivo é obter limites duais por meio da relaxação de um conjunto de restrições complicantes e a consequente penalização das violações desse conjunto na função objetivo.

As variáveis associadas a cada restrição alocada na função objetivo é chamada de multiplicadores de Lagrange ou variáveis duais, denominadas aqui pela letra  $\pi$ . O foco está em selecionar valores para essas variáveis de forma a forçar a satisfação das restrições complicantes  $Dx \geq d$ . Assim, pode-se definir o Subproblema Lagrangeano como:

$$L(\pi) = \min_x\{cx + \pi(d - Dx) : Bx \geq b, x \in Z_+^n\} \quad (4.2)$$

Por definição, é um problema relativamente tratável. Para qualquer vetor  $\pi \geq 0$ , a função dual  $L(\pi)$  define um limite dual ou inferior para esse exemplo, sobre o valor ótimo  $z$  em 4.1. De fato, a solução ótima  $x^*$  desse problema satisfaz  $cx^* \geq L(\pi) + \pi(d - Dx^*)$ .

$Dx^*) \geq L(\pi)$ , ou seja, a primeira desigualdade resulta de  $x^*$  viável para 4.1 e  $\pi \geq 0$  e a segunda inequação é garantida uma vez que  $x^*$  é viável em 4.2.

O dual Lagrangeano é definido como o problema de maximização do limite dual sobre o conjunto de vetores de penalidades admissíveis ( $\pi$ ), ou seja, identificar o maior limite inferior possível por meio da maximização do Subproblema Lagrangeano :

$$z_{LD} = \max_{\pi \geq 0} L(\pi) = \max_{\pi \geq 0} \min_{x \in Z} \{cx + \pi(d - Dx)\} \quad (4.3)$$

Buscando obter mais informações, o dual Lagrangeano será reformulado como um programa linear, assumindo que o conjunto de restrições  $Z$  é não vazio e limitado. O subproblema Lagrangeano atinge seu valor ótimo em um ponto extremo  $x^t$  de  $\text{conv}(Z)$ , assim o dual Lagrangeano pode ser reescrito como:

$$z_{LD} = \max_{\pi \geq 0} \min_{t=1, \dots, T} \{cx^t + \pi(d - Dx^t)\} \quad (4.4)$$

Onde  $\{x^t\}_{t=1, \dots, T}$  é o conjunto de pontos extremos de  $\text{conv}(Z)$  ou  $\{x^t\}_{t=1, \dots, T}$  pode ser o conjunto de todos os pontos do conjunto  $Z$ . Introduzindo uma variável adicional  $\sigma$ , que representa o limite inferior sobre os valores  $(c - \pi D)x^t$ , pode-se reescrever 4.3 como o programa linear:

$$z_{LD} = \max \quad \pi d + \sigma \quad (4.5)$$

$$\pi Dx^t + \sigma \leq cx^t \quad t = 1, \dots, T \quad (4.6)$$

$$\pi \geq 0, \sigma \in \mathfrak{R}^1 \quad (4.7)$$

Cujo respectivo programa linear dual é:

$$z_{LD} = \min \sum_{t=1}^T (cx^t)\lambda_t \quad (4.8)$$

$$\sum_{t=1}^T (Dx^t)\lambda_t \geq d \quad (4.9)$$

$$\sum_{t=1}^T \lambda_t = 1 \quad (4.10)$$

$$\lambda_t \geq 0 \quad t = 1, \dots, T. \quad (4.11)$$

A partir da formulação acima, segue o seguinte resultado.

**Teorema 4.1** (Dualidade Lagrangeana).

$$z_{LD} = \min \{cx : Dx \geq d, x \in \text{conv}(Z)\} \quad (4.12)$$

Pela definição do conjunto de pontos  $\{x^t\}_{t=1}^T$ ,  $\text{conv}(Z) = \{x = \sum_{t=1}^T x^t \lambda_t : \sum_{t=1}^T \lambda_t = 1, \lambda_t \geq 0, t = 1, \dots, T\}$ . Assim, o valor da dual Lagrangeano é igual ao valor do programa linear obtido pela minimização de  $cx$  sobre o conjunto resultante da interseção das restrições complicantes  $Dx \geq d$  com o envelope convexo do conjunto mais simples  $Z$ .

### 4.1.1 Algoritmo do Subgradiente

O algoritmo do subgradiente é uma das alternativas existentes na literatura para solucionar o problema do dual Lagrangeano ( $L(\pi)$ ). Segundo Vanderbeck e Wolsey [33],  $L(\pi) = \min_{g \in G} (c - \pi D)x^g + \pi d$  é uma função côncava por partes de  $\pi$ , onde  $\{x^t\}_g \in G$  são todos os pontos do conjunto  $Z$ , como ilustrado na figura 4.1.

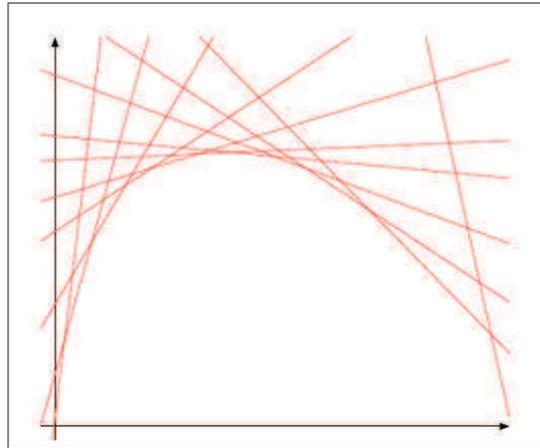


Figura 4.1: Gráfico representativo da Função do dual Lagrangeano  $L(\pi)$ , vista como uma função afim côncava por partes. Assume-se que  $\pi \in \mathbb{R}^1$  nessa representação. Cada parte é definida por um vetor  $x^t$  - Fonte: [33]

Por meio da maximização dessa função côncava não diferenciável, é possível solucionar o dual Lagrangeano. O seguinte método descreve os procedimentos necessários, que compõem o algoritmo do subgradiente, capaz de solucionar o dual Lagrangeano formulado na equação 4.4:

1. Inicializar  $\pi^0 = 0$ ,  $t = 1$ .
2. Iteração  $t$ ,
  - (a) Resolver o subproblema Lagrangeano 4.2 para obter o limite dual (limite inferior)  $L(\pi_t) = \max \{cx + \pi_t(d - Dx)\}$  e a solução ótima  $x^t$ .

- (b) Calcular a violação de  $x^t$  sobre as restrições dualizadas ( $d - Dx^t$ ). Esse passo fornece o subgradiente.
- (c) Atualizar a solução dual utilizando:  $\pi^{t+1} = \max \{ 0, \pi^t + \varepsilon_t(d - Dx^t) \}$ , onde  $\varepsilon_t$  é um parâmetro chamado passo, cujo cálculo do valor será definido em breve.
3. Se  $t < \tau$ , então aumentar uma unidade em  $t$  e retorna ao passo 2.  $\tau$  é o limite máximo de iterações e estabelecidas como parâmetro.

A ideia central desse método está na regra de atualização das variáveis duais no passo 2.c. Tal regra acarreta em um acréscimo nas variáveis duais ( $\pi$ ) associadas as restrições violadas e a um decréscimo para as demais restrições, embora não considere os pontos  $\{x^g\}$ , para  $g = 1, \dots, (t-1)$ , para atualizar  $\pi$ , o que pode resultar em um mal desempenho.

Além disso, a convergência do algoritmo está relacionada à escolha do tamanho do passo ( $\varepsilon$ ), pois um valor elevado pode causar oscilações e uma possível divergência. Em contrapartida, um valor muito pequeno pode provocar uma convergência lenta ou fazer com que a mesma ocorra em um ponto não ótimo. Dessa forma, é comum normalizar o tamanho do passo:  $\varepsilon = (\alpha_t)/(\|d - Dx^t\|)$ . Usualmente, os valores dos parâmetros utilizados são:

- $\alpha_t = C(LS - L(\pi_t))$  com  $C \in (0, 2)$ , onde o limite primal ou superior (LS) atua como um valor superestimado sobre o valor desconhecido do dual Lagrangeano ( $z_{LD}$ ).
- $\alpha_t$  forma uma série geométrica quando  $\alpha_t = C\rho^t$  com  $\rho \in (0, 1)$  e  $C > 0$ .
- $\alpha_t$  forma uma série divergente quando  $\alpha_t \rightarrow 0$  e  $\sum_t \alpha_t \rightarrow \infty$ . Por exemplo  $\alpha_t = \frac{1}{t}$ .

Uma forma natural de garantir a convergência é verificar quando o limite primal (superior) assume o valor do limite dual (inferior)  $z_{LD}$ .

Vale destacar que o tamanho do passo é sempre convergente, mas a convergência é muito lenta, por causa das sequências divergentes. Além disso, o parâmetro  $\tau$  presente no último passo 3, do algoritmo em destaque, permite que o número de iterações seja limitado. Outra heurística padrão de parada diz que o algoritmo pode ser interrompido quando o limite dual ou inferior  $LI = \max_t \{L(\pi^t)\}$  não obtiver melhora após várias iterações.

As condições de parada podem variar, dependendo do problema que está sendo resolvido e do resultado que se deseja encontrar. Nesse trabalho, as condições de parada consideradas foram:

1.  $\|d - Dx^t\| < \zeta$  - Subgradiente nulo ou desprezível, ou;
2.  $\varepsilon_t < \zeta$  - Tamanho do passo nulo ou desprezível, ou;
3.  $(LS - L(\pi_t)) < \zeta$  - Sendo  $\zeta$  um número pequeno arbitrariamente escolhido.

Concluindo, o algoritmo do subgradiente pode ser usado como heurística para produzir possíveis soluções para problema primal 4.12, embora não haja garantia que a restrição  $Dx \geq d$  seja atendida, por outro lado, a solução primal da formulação (4.8) - (4.11) atende a essa restrição.

Considerando que o candidato, à solução ótima, denotado por  $\hat{x}$  pode ser obtido como combinação convexa dos pontos gerados anteriormente  $x^g$  para  $g = 1, \dots, t$ . Possíveis regras de atualização seriam:

- $\hat{x} = \sum_{g=1}^t x^g \lambda_g$  onde  $\sum_{g=1}^t \lambda_g = 1$ .
- $\hat{x} = \delta \hat{x} + (1 - \delta)x^t$  com  $\delta \in (0,1)$ .

De acordo com Vanderbeck e Wolsey [33], a última regra é de maior interesse, porque maior peso é colocado sobre os pontos  $x^t$  gerados por último. Por meio do último passo(3), a teoria prevê a convergência de  $\hat{x}$  em direção à solução ótima em 4.12. Na prática, no entanto, deveria ser checado primeiro se  $\hat{x}$  respeita o conjunto  $Dx \geq d$  e, apenas em caso afirmativo, o valor associado seria armazenado como um limite superior sobre  $z_{LD}$ , o que é útil para monitorar a convergência. Caso  $\hat{x}$  respeite as condições de integralidade, então esse valor é tido como um limite superior.

## 4.2 Formulação por Relaxação Lagrangeana

Da mesma forma como foi descrito na Seção 4.1, um conjunto de restrições definidas como complicantes deve ser eliminada do problema e suas violações penalizadas na função do objetivo do novo problema. Na formulação original, o conjunto de restrições a ser relaxado será o conjunto 3.4, que representa as restrições de precedência do tipo *crossdocking* e são as restrições de acoplamento entre as variáveis  $x$  e  $y$ .

O número de restrições desse conjunto é o número de relações de precedência entre um job na máquina 2 ( $j \in J^2$ ) e seu precedente na máquina 1 ( $k \in S_j$ ). Por exemplo: Suponha que existem dois jobs a serem processados na máquina 2 e cada um deles possui 2 precedentes, o número de relações de precedência é 4. Para cada restrição do conjunto 3.4, associa-se um multiplicador  $\lambda$ , que representará o peso atribuído à violação daquela restrição na função objetivo. Dessa forma, o conjunto de multiplicadores de Lagrange do problema será representado por  $\lambda_{jk}$ , sendo ( $j \in J^2$ ) e ( $k \in S_j$ )  $\subset J^1$ . Para cada par ( $j, k$ ), temos a atribuição representada abaixo:

$$\sum_{t=0}^{T-p_{2j}} t.y_{jt} - \sum_{t=0}^{T-p_{1k}} (t + p_{1k}).x_{kt} \geq 0 \leftarrow \lambda_{jk} \quad (4.13)$$

Cada restrição do conjunto é representada por  $By - Ax \geq 0$ . Dessa forma, sua violação é dada por  $0 - By + Ax$ , ou seja,  $Ax - By$ , que é a diferença entre a data de conclusão de um precedente na máquina 1 e a data de início de seu sucessor na máquina 2. Considerando a situação em que o job na máquina 2 é sequenciado antes que seu predecessor seja concluído, pois a restrição foi eliminada do problema, essa violação será positiva, e a função objetivo será penalizada. Para cada par ( $j, k$ ), soma-se na função objetivo o fator  $\lambda_{jk} \cdot (Ax - By)$ , que representa a penalização pela violação daquela restrição em específico. Define-se portanto o subproblema lagrangeano  $L(\lambda)$ :

$$L(\lambda) = \min \sum_{j \in J^2} \sum_{t=0}^{T-p_{2j}} (t + p_{2j}).y_{jt} + \sum_{j \in J^2} \sum_{k \in S_j} \lambda_{jk} \left( \sum_{t=0}^{T-p_{1k}} (t + p_{1k}).x_{kt} - \sum_{t=0}^{T-p_{2j}} t.y_{jt} \right) \quad (4.14)$$

s.t. (3.2), (3.3), (3.5), (3.6), (3.7) e (3.8)

Definido o  $L(\lambda)$ , deve-se resolver o dual lagrangeano, que conforme descrito na seção 4.1, maximizará o valor do limite inferior para o problema original. Nesse trabalho, o método utilizado para solucionar o dual lagrangeano será o Algoritmo do Subgradiente descrito na subseção 4.1.1, cujo primeiro passo após a inicialização é resolver o subproblema lagrangeano  $L(\lambda)$  acima para o conjunto  $\lambda$  dado. Lembrando que a cada iteração do algoritmo os valores de  $\lambda$  são atualizados, definindo as penalidades (pesos) para cada restrição violada naquela iteração. Os passos para a resolução de  $L(\lambda)$  são descrito na seção seguinte.

## 4.3 Solucionando o Subproblema Lagrangeano

Analisando o modelo resultante, a aplicação da relaxação lagrangeana resultou em um subproblema com algumas características especiais, que tornam possível a decomposição desse subproblema em dois problemas independentes. É possível identificar que com a eliminação do conjunto de restrições 3.4, as variáveis  $x$  e  $y$  tornam-se desacopladas, dessa forma, não há restrições no novo problema que relacionam os valores de  $x$  com os valores de  $y$ . Portanto, separando também na função objetivo os termos que contém valores da variável  $x$  dos termos que contém valores da variável  $y$ , pode-se definir dois subproblemas menores independentes, cada qual com termos da função objetivo e restrições de uma das variáveis. Assim, a solução de cada um dos modelos separadamente é vista como uma alternativa. O valor de  $L(\lambda)$ , que será um limite inferior para o problema, é obtido agregando os subproblemas  $L(\lambda)_x$  e  $L(\lambda)_y$ .

### 4.3.1 Subproblema em X

Com os termos que contém  $x$  isolados da função objetivo do subproblema lagrangeano, obtêm-se o problema:

$$L(\lambda)_x = \min \sum_{j \in J^2} \sum_{k \in S_j} \lambda_{jk} \sum_{t=0}^{T-p_{1k}} (t + p_{1k}) x_{kt} \quad (4.15)$$

s.t. (3.2), (3.5) e (3.7) - Ou seja, restrições que contém somente  $x$ .

Alternativamente, podemos reescrever a função objetivo acima sob a perspectiva dos jobs em  $J^1$ , ao invés de avaliá-los indiretamente através do conjunto de precedentes dos jobs em  $J^2$ , conforme abaixo:

$$\min \sum_{j \in J^1} \sum_{t=0}^{T-p_{1j}} (t + p_{1j}) x_{jt} w_j^1 \quad (4.16)$$

onde  $w_j^1 = \sum_{i \in J^2} \lambda_{ij}$ .

Se  $j \notin S_i$ ,  $\lambda_{ij} = 0$ .

Como indicado acima,  $w_j^1$  é a soma das penalidades de todos os jobs na máquina 2 que possuem o job  $j$  como precedente na máquina 1. Esse valor é uma constante que representa o peso do job  $j$ , e será maior se este job for responsável por grandes violações das restrições de precedência.

O problema citado acima é conhecido como Tempo Total de Conclusão Ponderado ([23]), denotado por  $1 \parallel \sum C_j W_j$ , sendo  $w_j^1$  o peso do job  $j$ . Esse problema pode ser resolvido por ordenação de acordo com a regra WSPT (*Weighted Shortest Processing Time First*). De acordo com essa regra, a solução ótima é obtida ordenando-se os jobs em ordem decrescente de  $w_j^1 / p_{1j}$ , e pode ser obtida em  $O(n \log(n))$ .

**Teorema 4.2.** *A regra WSPT é ótima para o problema  $1 \parallel \sum C_j W_j$ . ([23])*

*Demonstração.* Suponha uma sequência  $S$  na qual existem dois jobs adjacentes  $j$  e  $k$ , sendo  $j$  seguido de  $k$ , tal que  $w_j^1 / p_{1j} < w_k^1 / p_{1k}$ ,  $w_j^1 > 0 \forall j$ , ou seja, não seguindo a regra WSPT. Dessa forma, o tempo total de conclusão ponderado  $T_{jk}$  dos jobs  $j$  e  $k$  é  $(t + p_{1j})w_j^1 + (t + p_{1j} + p_{1k})w_k^1$ . Troca-se então  $j$  e  $k$  de posição, mantendo os demais jobs em suas posições originais, denominando a nova sequência  $S'$ , onde o tempo total de conclusão ponderado  $T_{kj}$  dos jobs  $j$  e  $k$  é  $(t + p_{1k})w_k^1 + (t + p_{1k} + p_{1j})w_j^1$ . Como  $w_j^1 / p_{1j} < w_k^1 / p_{1k}$ , verifica-se que sob  $S'$ ,  $T_{kj}$  é sempre menor que  $T_{jk}$  em  $S$ , o que contradiz a otimalidade de  $S$ , provando que a regra WSPT chega a solução ótima do problema.  $\square$

### 4.3.2 Subproblema em Y

Com os termos que contém y isolados da função objetivo do subproblema lagrangeano, obtém-se o problema:

$$L(\lambda)_y = \min \sum_{j \in J^2} \sum_{t=0}^{T-p_{2j}} (t + p_{2j})y_{jt} - \sum_{j \in J^2} \sum_{k \in S_j} \lambda_{jk} \sum_{t=0}^{T-p_{2j}} t \cdot y_{jt} \quad (4.17)$$

s.t. (3.3), (3.6) e (3.8) - Restrições que contém y.

Definindo  $a_j = \sum_{k \in S_j} \lambda_{jk} \forall j \in J^2$  tem-se:

$$L(\lambda)_y = \min \sum_{j \in J^2} \sum_{t=0}^{T-p_{2j}} (t + p_{2j})y_{jt} - \sum_{j \in J^2} \sum_{t=0}^{T-p_{2j}} a_j \cdot t \cdot y_{jt}$$

Multiplicando-se os termos entre parênteses e colocando  $t + y_{jt}$  em evidência, chega-se a:

$$L(\lambda)_y = \min \sum_{j \in J^2} \sum_{t=0}^{T-p_{2j}} t \cdot y_{jt} (1 - a_j) + p_{2j} \cdot y_{jt}$$

Definindo  $w_j^2 = 1 - a_j \forall j \in J^2$ , podemos reescrever o problema da seguinte maneira:

$$L(\lambda)_y = \min \sum_{j \in J^2} \sum_{t=0}^{T-p_{2j}} t \cdot y_{jt} \cdot w_j^2 + \sum_{j \in J^2} \sum_{t=0}^{T-p_{2j}} y_{jt} \cdot p_{2j}$$

s. t. (3.3), (3.6) e (3.8).

O segundo termo da função objetivo acima é constante, pois representa a soma de processamento de todos os jobs na máquina 2. Além disso, todos os jobs presentes nesse termo serão sequenciados pelo conjunto de restrições 3.3. Dessa forma, a função objetivo restante é o somatório das datas de início ponderadas dos jobs na máquina 2, gerando um problema que podemos denominar Tempo Total de Início Ponderado, denotado  $1 \parallel \sum I_j W_j$ , sendo  $w_j^2$  o peso do job  $j$  em M2.

Para resolver este subproblema, usa-se a mesma regra utilizada para o problema da máquina 1, assim a solução ótima nesse caso também pode ser obtida de acordo com a regra WSPT em  $O(n \log(n))$ .

**Teorema 4.3.** *A regra WSPT é ótima para o problema  $\sum I_j W_j$*

*Demonstração.* Suponha uma sequência  $S$  na qual existem dois jobs adjacentes  $j$  e  $k$ , sendo  $j$  seguido de  $k$ , tal que  $w_j^2/p_{2j} < w_k^2/p_{2k}$ ,  $w_j^2 > 0 \forall j$ , ou seja, não obedecendo a regra WSPT (Prova por absurdo). Dessa forma, o tempo total de início ponderado  $T_{jk}$  dos jobs  $j$  e  $k$  é  $t.w_j^2 + (t + p_{2j})w_k^2$ . Troca-se então  $j$  e  $k$  de posição, mantendo os demais jobs em suas posições originais, denominando a nova sequência  $S'$ , onde o tempo total de conclusão ponderado  $T_{jk}$  dos jobs não é alterado. Sob  $S$ , o tempo total de início ponderado  $T_{jk}$  dos jobs  $k$  e  $j$  é  $t.w_k^2 + (t + p_{2k})w_j^2$ . Como  $w_j^2/p_{2j} < w_k^2/p_{2k}$ , verifica-se que sob  $S'$ ,  $T_{jk}$  em  $S$ , o que contradiz a otimalidade de  $S$ , provando que a regra WSPT chega à solução ótima do problema.  $\square$

Quando  $\sum_{k \in S_j} \lambda_{jk} > 1$  para determinado  $j$ ,  $w_j^2$  é negativo, contrariando a suposição que  $w_j^2 > 0 \forall j$ . Neste caso, dada uma sequência na máquina 2 gerada pela regra WSPT, deve-se sequenciar os jobs com os  $w_j^2 > 0$  a partir de  $t = 0$ , e os jobs com  $w_j^2 < 0$  a partir de  $t = T$ , mantendo a sequência gerada pela regra WSPT. Isso se dá, pois como a função objetivo é de minimização, deve-se associar um grande valor para  $t$ , ou seja, a data de início, para jobs com peso negativo e um valor baixo de  $t$  para jobs com peso positivo, minimizando assim a soma das datas de início ponderadas. Logo, observa-se que o problema só tem solução viável sob esta modelagem, pois o horizonte de tempo é limitado em  $t = T$ , caso contrário a solução ótima seria alocar jobs com  $w_j^2 < 0$  em  $t = \infty$ .

## 4.4 Encontrando um Limite Superior para o Problema

Resolvido o subproblema lagrangeano  $L(\lambda)$  e obtido um limite inferior para o problema, o próximo passo é obter uma solução viável no espaço de soluções do problema original, visto que a eliminação das restrições de precedência torna o espaço de soluções maior, e uma solução encontrada nesse novo espaço não é necessariamente viável no espaço anterior. Encontrada essa solução, seu valor da função objetivo será um limite superior para o problema original.

Nesse problema, pode-se dizer que a obtenção de um limite superior ou solução viável qualquer é relativamente fácil. A questão é encontrar uma solução próxima da solução ótima do problema, o que proporcionaria maior rapidez na obtenção da solução ótima. A estratégia utilizada para obter soluções viáveis para o problema nesse trabalho é definida da seguinte forma:

Considera-se que a sequência na máquina 1 é aquela obtida pela solução do subproblema em X, ou seja, ordenando-se os jobs dessa máquina de acordo com a regra WSPT. Já a máquina 2, a solução obtida por meio da resolução do subproblema em Y pode violar as restrições de precedência de *crossdocking* existentes no problema original, já que estas foram eliminadas na aplicação do método de relaxação lagrangeana. Portanto, é necessário obter uma nova sequência na máquina 2 de forma a satisfazer tais restrições, sendo assim, uma nova solução viável para o problema original.

Para o caso especial, no qual todos os jobs gerados pelo subproblema em Y possuem peso negativo ( $w + j^2 < 0 \forall j \in J^2$ ), estes serão sequenciados no final do horizonte de tempo, conforme dito anteriormente. Dessa forma, a solução obtida pode ser viável no problema original. Pelo fato de os pesos serem negativos, a preposição 2 não é válida, ou seja, essa não é uma solução ótima para o problema original. Pode-se dizer também que é provável que existam soluções viáveis mais próximas da solução ótima, visto que nesta última os jobs foram sequenciados o mais tarde possível na máquina 2, o que acresce o valor na função objetivo. Logo, vale destacar que o procedimento adotado para obter soluções viáveis no problema original será sempre executado, pois o algoritmo do subgradiente necessita de um valor para o limite superior, a fim de identificar o gap de otimalidade.

Assim, simplificando, um limite superior é obtido quando uma sequência viável na máquina 2 é gerada, dada uma sequência fornecida pela máquina 1. Para que a sequência seja viável no problema original, um job na máquina 2 só pode ser inici-

ado após todos seus precedentes na máquina 1 terem sido completamente processados. Defini-se então, para cada job da máquina 2, sua *release date* ( $r_j, j \in J^2$ ), que representa a data a partir da qual aquele job pode ser sequenciado. A *release date* de um job  $j$  é calculada como  $r_j = \max_{k \in S_j} C_k$ , ou seja, a maior data de conclusão dentre os precedentes de  $j$  na máquina 1 na sequência dada. A sequência obtida deve minimizar o tempo total de conclusão dos jobs na máquina 2, que é a função objetivo do problema original. Esse procedimento não garante a solução ótima do problema original, pois não é garantido que a sequência fixada na máquina 1 é ótima, porém espera-se chegar a uma solução próxima à ótima.

O problema discutido é conhecido na literatura como Tempo Total de Conclusão com *Release Dates* em uma máquina, denotado  $1 | r_j | \sum C_j$ . Apesar de o problema  $1 || \sum C_j$  poder ser resolvido de maneira exata por simples ordenação em ordem não decrescente do tempo de processamento, a introdução de *release date* torna o problema NP-difícil. Resolver um problema NP-difícil dentro do procedimento de resolução do dual lagrangeano tornaria tal procedimento inviável na prática, devido ao alto custo computacional. Por esse motivo, é utilizado um algoritmo 2-aproximado proposto por Philips et al. [29]. Um algoritmo 2-aproximado significa que, em tempo polinomial, encontra-se uma solução no máximo duas vezes o valor ótimo da função objetivo do problema.

O procedimento inicia resolvendo o problema considerando preempção  $1 | r_j, prmp | \sum C_j$  pela regra do menor tempo de processamento: em cada instante de tempo  $t \in T$ , processar o job com o menor tempo de processamento restante.

A sequência gerada é então usada como entrada para o procedimento *Convert* proposto em [29], que converte uma sequência preemptiva em uma não preemptiva, ou seja, transforma uma solução de  $1 | r_j, prmp | \sum C_j$  em uma solução  $1 | r_j | \sum C_j$ , conforme abaixo:

**Algoritmo *Convert***

**Entrada:** Sequência preemptiva P

**Saída:** Sequência não-preemptiva N

1. Forme uma lista L de jobs, ordenada em ordem não decrescente das datas de conclusão preemptivas  $C_j^P$ . A data de conclusão preemptiva de um jobs  $j$  é a data de conclusão da última parte deste *job*.
2. Sequenciar usando a lista L, com a restrição de que nenhum job  $j$  inicie seu

processamento antes da data  $r_j$ . Isso produz uma sequência na qual os jobs iniciam na ordem dada por L.

O procedimento descrito é então aplicado para os jobs da máquina 2, obtendo, conforme explicado anteriormente, uma solução viável e um limite superior para o problema original, valor que será utilizado como uma etapa do algoritmo do subgradiente. Ressalta-se que esse procedimento será sempre executado, pois o algoritmo do subgradiente necessita de um valor para o limite superior, a fim de identificar o gap de otimalidade.

## 4.5 Testes Computacionais

### 4.5.1 Geração de Instâncias

As propriedades das instâncias geradas e aplicadas nos modelos foram:

- Existem dois grupos de instâncias: O primeiro possui jobs, cujos tempos de processamento foram gerados uniformemente entre 1 e 5. O segundo grupo apresenta jobs com tempos de processamento gerados uniformemente entre 1 e 50. Essa variação nos tempos de processamento tem como objetivo avaliar a influência do tamanho do horizonte de tempo no desempenho do algoritmo.
- Para cada grupo, foram gerados 3 instâncias de cada subgrupo: 5, 10, 15, 20 ou 50 jobs na máquina 1. O número de jobs na máquina 2 não pode ser superior ao número de jobs da máquina 1.
- Os jobs associados a máquina 1 são precedentes de pelo menos 1 job na máquina 2, respeitando a premissa do problema de que não existe job na máquina 1 sem sucessores. Na prática, se esse for o caso, este job deve ser sequenciado por último na máquina 1.
- O número de precedentes de cada job na máquina 2, também gerado por meio de uma distribuição uniforme, é no mínimo 1 e no máximo 50 % do número de jobs na máquina 1.

Cada subgrupo de instâncias geradas possuirá um código único para facilitar a referência ao longo da apresentação dos resultados e análises, no formato  $n - m - mp - g$ ,

onde  $n$  é o número de jobs na máquina 1,  $m$  é o número de jobs na máquina 2,  $mp$  é o número máximo de precedentes de cada job na máquina 2 e  $g$  é o grupo o qual pertence o subgrupo.

A tabela a seguir apresenta um resumo das instâncias geradas e suas respectivas características:

Tabela 4.1: Sumário das instâncias geradas para teste, separadas em grupo 1 e 2, e subgrupos, de acordo com o número de jobs na máquina 1 ( $JobsM1(n)$ ). Além disso, a tabela informa o número de jobs na máquina 2 ( $JobsM2(m)$ ), o número de precedentes do job  $j \in J^2$  ( $NP$ ), o tempo de processamento dos jobs ( $TP$ ), o código da instância  $Código$  e por último quantas instâncias foram geradas para cada subgrupo  $N.Inst.$

Grupo	Jobs M1(n)	Jobs M2(m)	NP	TP	Código	N. Inst.
1	5	2-3-4	3	U(1,5)	5-m-3-1	3
	10	3-5-7	5	U(1,5)	10-m-5-1	3
	15	5-8-11	8	U(1,5)	15-m-8-1	3
	20	6-10-14	10	U(1,5)	20-m-10-1	3
	50	15-25-35	15	U(1,5)	50-m-25-1	3
2	5	2-3-4	3	U(1,50)	5-m-3-2	3
	10	3-5-7	5	U(1,50)	10-m-5-2	3
	15	5-8-11	8	U(1,50)	15-m-8-2	3
	20	6-10-14	10	U(1,50)	20-m-10-2	3
	50	15-25-35	15	U(1,50)	50-m-25-2	3

Os valores dos tempos de processamento dos jobs, presentes nos grupos  $J^1$  ou  $J^2$ , foram definidos de forma aleatória, de acordo com a respectiva distribuição uniforme definida na tabela. O número máximo de precedentes foi estabelecido anteriormente, assim como os valores que representam  $n$  e  $m$ . A configuração definida busca avaliar os resultados quando a quantidade de jobs é alterada. Além disso, é analisado as interferências relativas ao tempo de processamento dos jobs e ao número de precedentes de um job que será processado pela máquina 2.

## 4.5.2 Características da máquina para teste

Os testes descritos nesse trabalho foram executados em uma máquina com processador Intel(R) Xeon (R) CPU X5450@2,40GHz, com 12 processadores, 12 GB de memória RAM em sistema operacional Linux Ubuntu 13.10. Foram utilizadas as linguagens de programação C, C++ e AMPL versão 2012911(Linux x86 64). O compilador GNU GCC e o software de otimização CPLEX versão 12.4.

### 4.5.3 Testes - Relaxação Lagrangeana

Foram feitos testes com o objetivo de avaliar e comparar os resultados das seguintes formulações: Completa, Relaxação Linear e Relaxação Lagrangeana. A primeira formulação contém todas as restrições (3.2) - (3.8) citadas no capítulo 3. A segunda contém as restrições (3.2) - (3.6), as variáveis foram relaxadas linearmente e, por último, foi aplicado a relaxação Lagrangeana descrita nesse capítulo.

O tempo de execução foi limitado a 1 hora de processamento. Após esse tempo foram registrados os seguintes dados: Limite Inferior e Superior, gap de otimalidade e tempo de máquina. Caso a solução ótima seja encontrada em menos de 1 hora, a execução é interrompida. A tabela 4.2 expõem os resultados.

#### 4.5.4 Análise dos Testes com Relaxação Lagrangeana

Por meio dos resultados expostos na tabela 4.2, é possível comparar os desempenhos da relaxação Lagrangeana e da relaxação linear. Verifica-se que a relaxação Lagrangeana fornece os mesmos limites inferiores calculados pela relaxação linear para todas as instâncias.

Por outro lado, a relaxação Lagrangeana apresenta dois fatores que podem justificar o esforço para sua implementação. O primeiro é o baixo tempo computacional utilizado para processamento, comparado com a relaxação linear do modelo completo, pois em nenhuma das instâncias a relaxação linear superou seu desempenho. Esse fato pode ser proveitoso ao se trabalhar com outras instâncias ainda maiores do que as que foram testadas nesse trabalho, visto que as duas últimas instâncias do grupo 2 levaram mais de 2 minutos cada para serem solucionadas pelo modelo relaxado. O segundo fator está associado ao cálculo do limitante superior, pois o seu valor não é muito distante do valor fornecido pelo modelo completo -, principalmente nas instâncias que o valor ótimo foi identificado pelo modelo original. Dessa forma, trabalhar na melhoria do cálculo do limite superior, associando-o a heurísticas, pode ser uma boa opção na busca de soluções ótimas para instâncias maiores.

Com relação aos testes realizados com o modelo completo, constata-se que não foi identificada a solução ótima para as maiores instâncias do grupo 1 e no grupo 2 apenas as menores instâncias tiveram sua solução ótima identificada. Esse fato expõe a dificuldade de resolução de modelos com indexação no tempo, uma vez que os parâmetros entre os dois grupos são os mesmos, com exceção do tempo de processamento.

Tabela 4.2: Comparação entre Modelo Original, Relaxação Linear e Relaxação Lagrangeana. A coluna (*Código*) refere-se a instância trabalhada, os itens (*RL*) e (*R.Lag*) significam Relaxação Linear e Relaxação Lagrangeana, respectivamente. Já as siglas (*FO*), (*LI*), (*LS*), (*T*) e (*GAP*) representam a função objetivo, limite inferior, limite superior, tempo de processamento em segundos e o GAP de otimalidade ( $\frac{LS - LI}{LS}$ ) respectivamente. Foi apresentada a solução ótima para relaxação linear.

Código	Modelo Original			RL		R. LAG			
	FO	GAP	T(seg)	FO	T(seg)	LI	LS	GAP	T(seg)
5-2-3-1	34	0%	0,2	28	0	28	34	16%	0
5-3-3-1	28	0%	0	26	0	26	28	7%	0
5-4-3-1	50	0%	0,2	44	0	44	50	11%	0
10-3-5-1	61	0%	3,2	49	0	49	61	19%	0
10-5-5-1	113	0%	44,7	97	0	97	117	17%	0
10-7-5-1	157	0%	50,48	141	0	141	166	15%	0
15-5-8-1	149	0%	3600	132	0	132	149	11%	0
15-8-8-1	240	3%	3600	208	0	208	259	19%	0
15-11-8-1	347	11%	3600	292	0	292	388	25%	0
20-6-10-1	190	0%	3350	168	0,1	168	190	11%	0
20-10-10-1	389,7	20%	3600	382	3	382	547	30%	0
20-14-10-1	481,1	15%	3600	476	1	476	625	24%	0
50-15-15-1	1383	13%	3600	1207	1	1207	1534	21%	0,1
50-25-15-1	3093	33%	3600	2071	7	2071	3225	35%	0,1
50-35-15-1	4725	36%	3600	3037	8	3037	4869	37%	0,2
5-2-3-2	226	0,0%	2,2	185	0	185	226	18%	0
5-3-3-2	436	0,0%	4	401	0,2	401	436	8%	0
5-4-3-2	327	0,0%	0,4	323	0,1	323	327	1%	0
10-3-5-2	361	6,0%	3600	319	1	319	361	14%	0
10-5-5-2	882	9,0%	3600	788	3	788	881	10%	0
10-7-5-2	1000	5,7%	3600	926	2	926	1000	7%	0
15-5-8-2	1705	17,2%	3600	1402	14	1402	1766	21%	0
15-8-8-2	2055	28,9%	3600	1453	9	1453	2147	32%	0
15-11-8-2	3907	24,3%	3600	2954	8	2954	3966	26%	0,1
20-6-10-2	1859	24,6%	3600	1396	5	1396	1575	11%	0
20-10-10-2	3642	22,2%	3600	2831	9	2831	3720	24%	0,1
20-14-10-2	4866	27,8%	3600	3509	3	3509	4750	26%	0,2
50-15-15-2	7950	24,5%	3600	6001	41	6001	7609	21%	0,4
50-25-15-2	18699	47,8%	3600	12737	33	12737	17327	26%	1,1
50-35-15-2	22456	49,0%	3600	15280	115	15280	22616	33%	1,4

# Capítulo 5

## Geração de Colunas

Nesse Capítulo é discutida a aplicação do método de Geração de Colunas e é apresentado o algoritmo proposto. Na Seção 5.1 é descrita a teoria na qual o método se apoia. Em seguida na Seção 5.2, é detalhado como foi feita a aplicação do método e apresentado a formulação resultante. Além disso, a heurística proposta utilizada como complemento do método também será discutida nessa Seção.

### 5.1 Geração de Colunas

No Capítulo 4, foi apresentado como é a abordagem do método da relaxação Lagrangeana frente ao problema discutido nesse trabalho. Já o método de geração de colunas apresenta uma outra forma de lidar com o problema em questão. Nesse caso, a otimização é feita a partir da seleção de soluções presentes no conjunto  $Z = \{ x \in Z_+^n : Bx \geq b \}$ , porém esses valores devem atender também as restrições que definem o conjunto  $Y = \{ x \in Z_+^n : Dx \geq d \}$ . Essa abordagem leva a formulação proposta por Dantzig-Wolfe (5.1) - (5.4), descrita na subseção 5.1.1, na qual variáveis são associadas a pontos do conjunto  $Z$ , como especificado nos teoremas 5.1 e 5.2 expostos a seguir:

**Teorema 5.1.** *Todo poliedro  $P = \{ x \in \mathfrak{R}^n : Ax \geq a \}$  pode ser representado da seguinte forma:*

$$P = \{ x \in \mathfrak{R}^n : x = \sum_{g \in G} \lambda_g x^g, \sum_{g \in G} \lambda_g = 1, \lambda \in \mathfrak{R}_+^{|G|} \}$$

Onde  $\{x^g\}_{g \in G}$  são os pontos extremos de  $P$ .

**Teorema 5.2.** *Todo conjunto de programação inteira  $X = \{ x \in Z^n : Ax \geq a \}$  pode*

ser representado na forma  $X = \text{proj}_x(Q_i)$ , onde:

$$Q_i = \{(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}_+^{|G|} : x = \sum_{g \in G} \lambda_g x^g, \sum_{g \in G} \lambda_g = 1\}$$

Onde  $\{x^g\}_{g \in G}$  é um conjunto finito de pontos inteiros em  $X$ .

Vale destacar que a solução dessa formulação fornece limites inferiores mais próximos a solução ótima do problema, comparados com aqueles obtidos pela relaxação linear do problema original em  $X$  e, além disso, é igual ao melhor limite calculado pela relaxação Lagrangeana das restrições  $Dx \geq d$ , justificando a escolha pelo método [33].

### 5.1.1 Formulação Dantzig-Wolfe

Assume-se que o conjunto  $Z$  é fechado e limitado e a seguinte formulação estendida do problema inteiro  $\min = \{cx : Dx \geq d, x \in Z\}$ , resultante do teorema 5.2 citado acima.

$$z^{DW} \min \sum_{g \in G} (cx^g) \lambda_g \quad (5.1)$$

$$\sum_{g \in G} (Dx^g) \lambda_g \geq d \quad (5.2)$$

$$\sum_{g \in G} \lambda_g = 1 \quad (5.3)$$

$$\lambda_g \in \{0, 1\} \forall g \in G \quad (5.4)$$

Onde  $\{x^g\}, g \in G$ , são todos os pontos extremos de  $Z$ .

O leitor pode observar que o modelo (4.8) - (4.11) do capítulo 4 é exatamente a relaxação linear do modelo proposto acima, pois qualquer ponto do conjunto  $Z$  pode ser obtido através da combinação convexa dos pontos extremos de  $Z$ . Assim:

$$z^{DW} = \min \{cx : Dx \geq d, x \in \text{conv}(Z)\} = z_{LD} \quad (5.5)$$

Onde  $z^{DW}$  representa o valor da relaxação linear de (5.1) - (5.4) e  $z_{LD}$  denota o valor do dual Lagrangeano calculado em (4.8) - (4.11).

### 5.1.2 Resolvendo a Relaxação da Formulação Dantzig-Wolfe por Geração de Colunas

Nessa seção será considerado como calcular o limite dual fornecido pela relaxação da formulação de Dantzig-Wolfe ( $DW$ ), utilizando-se o método de geração de colunas.

A relaxação linear de  $DW$  dada em (5.1) - (5.4) é usualmente denominada Problema Mestre (PM). Sua formulação é composta por um número muito elevado de variáveis, que são introduzidas dinamicamente ao longo do processo de otimização.

Assume-se que o conjunto  $Z$  é inteiro, limitado e  $\{x^g\} g \in G$  são todos os pontos extremos desse conjunto. Suponha que, em uma determinada iteração  $t$  do algoritmo, somente um subconjunto de pontos  $\{x^g\} g \in G^t$  com  $G^t \subset G$  são conhecidos. Dessa forma, defini-se o Problema Mestre Restrito (PMR):

$$z^{PMR} = \min \sum_{g \in G^t} (cx^g) \lambda_g \quad (5.6)$$

$$\sum_{g \in G^t} (Dx^g) \lambda_g \geq d \quad (5.7)$$

$$\sum_{g \in G^t} \lambda_g = 1 \quad (5.8)$$

$$0 \leq \lambda_g \leq 1 \quad \forall g \in G^t \quad (5.9)$$

O problema dual do modelo acima assume a seguinte forma:

$$\max \quad \pi d + \sigma \quad (5.10)$$

$$\pi D x^g + \sigma \leq c x^g \quad g \in G^t \quad (5.11)$$

$$\pi \leq 0, \quad \sigma \in \Re^1 \quad (5.12)$$

$$(5.13)$$

Assim,  $\lambda^*$  e  $(\pi^*, \sigma^*)$  representam as soluções primais e duais do PMR respectivamente.

Através das formulações primais e duais do PMR, seguem as seguintes observações sobre o método geração de colunas:

1. Dada solução dual corrente  $(\pi^*, \sigma^*)$ , o custo reduzido da coluna associada a solução  $x^g$  é  $cx^g - \pi^* D x^g - \sigma^*$ .
2. Defini-se o problema de precificação (*pricing*) como:  $\zeta = \min_{g \in G} (cx^g - \pi^* D x^g) = \min_{x \in Z} (c - \pi^* D)x$ . Assim, em vez de avaliar o custo reduzido de um grande número

mero de colunas, o problema de precificação pode ser executado implicitamente, ou seja, sem explicitar todas as variáveis, solucionando um programa inteiro sobre o conjunto  $Z$ .

3. A solução do PMR (Problema Mestre Restrito)  $z^{PMR} = \sum_{g^t} (cx^g)\lambda_g^* = \pi^*d + \sigma^*$  fornece um limite superior sobre  $z_{PM}$  (Solução do Problema Mestre, ou seja, formulação que considera todas as variáveis). O PM está resolvido quando  $\zeta - \sigma^* \geq 0$ , ou seja, quando não há mais colunas com o custo reduzido negativo.
4. O problema de precificação definido no segundo item é equivalente ao Subproblema Lagrangeano 4.2. Dessa forma, a cada iteração do problema de precificação um limite dual Lagrangeano é fornecido.
5. Um outro ponto de vista sobre o item 4, o valor de  $\pi^*$ , associado ao termo  $\zeta$  apresentado no item 2, representa a solução dual do PMR, forma uma solução viável de  $(\pi^*, \zeta^*)$  para o dual do PM:

$$\max\{\pi d + \sigma : \pi D x^g + \sigma \leq c x^g, g \in G, \pi \geq 0, \sigma \in \mathbb{R}^1\} \quad (5.14)$$

E, portanto,  $\pi^*d + \zeta$  fornece o limite inferior sobre  $z^{PM}$

6. Caso a solução  $\lambda^*$  do PMR seja inteira, o valor correspondente  $z^{PMR}$  será um limite superior para o  $Z_{PM}$ .

Destaca-se que no item 5, o problema de precificação sobre o dual do PMR pode ser visto como um problema de separação. Nesse caso, é avaliado se novas restrições serão incorporadas ao dual do PMR.

A seguir será descrito o algoritmo de Geração de Colunas aplicado à formulação (5.6) - (5.9) :

1. Inicialize os limites primais e duais  $LP = +\infty$  e  $LD = -\infty$ . Gere um subconjunto de pontos  $x^g$ , de forma que PMR seja viável.
2. Iteração  $t$ :
  - (a) Resolva PMR sobre o conjunto corrente de colunas  $\{x^g\}_{g \in G^t}$ . Armazene a solução primal  $\lambda^t$  e a solução dual  $(\pi^t, \sigma^t)$ .
  - (b) Cheque se  $\lambda^t$  é uma solução inteira para PMR e  $LP = LD$ . Se sim, atualize  $LP = LD$  e Pare.

(c) Resolva o Problema de Precificação:

$$(SP^t) \quad \zeta^t = \min \{ (c - \pi^t D)x : x \in Z \} \quad (5.15)$$

Armazenar  $x^t$  como solução ótima do *pricing*.

Se  $\zeta - \sigma^t = 0$ , fazer Limite Dual LD =  $z^{PMR}$  e Pare. O PMR foi resolvido. Caso contrário,  $\zeta - \sigma^t < 0$ , adicionar  $x^t$  ao conjunto  $G^t$  e incluir a coluna associada no PMR.

(d) Calcular o Limite Dual (subproblema Lagrangeano ou *pricing*):  $L(\pi^t) = \pi^t d + \zeta^t$ . Atualizar o Limite Dual LD =  $\max \{ LI, L(\pi^t) \}$ . Se LP = LD, então Pare.

3. Incrementar  $t$  e retornar para o passo 2.

O comportamento típico dos limites primais e duais ao longo do algoritmo de geração de colunas está ilustrado na figura 5.1.

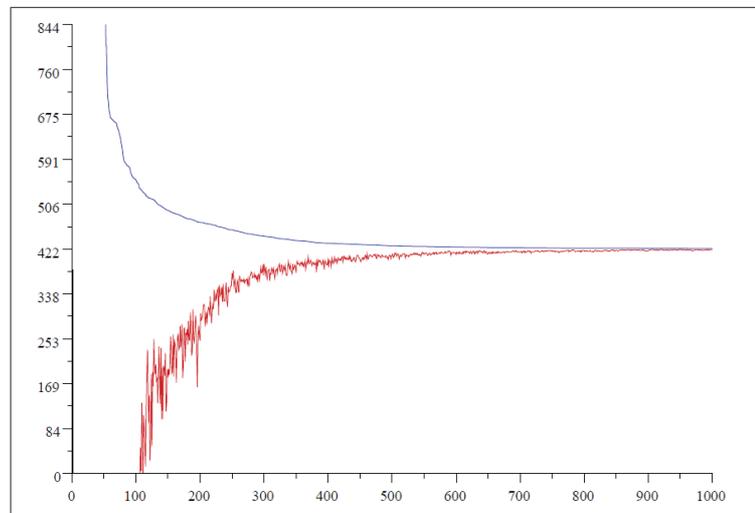


Figura 5.1: Gráfico representativo da convergência do algoritmo de geração de colunas. O eixo x é o tempo em segundos e o eixo y são os valores do limite primal (superior) e dual (inferior). A linha superior representa o limite primal, que é dado pelo PM, já a linha inferior é calculada com auxílio do subproblema e representa o limite dual. - Fonte: [33]

## 5.2 Formulação - Geração de Colunas

Nessa Seção é apresentado o Problema Mestre Restrito (PMR), bem como o subproblema de Precificação (SB). Além disso, é discutida a forma escolhida de inicialização do método abordado e a heurística que é associada ao mesmo. A introdução da geração de colunas nesse trabalho tem o objetivo de solucionar o problema tratado. Tal método é visto como uma alternativa em relação a relaxação Lagrangeana.

### 5.2.1 Formulação Problema Mestre Restrito e Subproblema de Precificação

A aplicação do método geração de colunas para problema tratado nesse trabalho segue a seguinte dinâmica. O problema mestre restrito (PMR) é responsável por identificar a melhor solução dado um subconjunto  $c'$  de colunas pertencentes a um conjunto maior  $C$ , que é composto por todas as colunas possíveis, ou seja,  $c' \subset C$ . Assim, o PMR calculará a melhor sequência de processamento dos jobs para a máquina 2.

Por outro lado, o subproblema (SB) fornece as colunas que irão abastecer o PMR, ou seja, os vetores que irão fazer parte do conjunto  $C$ . O SB procura minimizar o problema de precificação descrito em sua função objetivo, as restrições, as quais o SB está sujeito, identificam sequências viáveis para os jobs processados pela máquina 1. Assim, definido tal sequência, será possível calcular os valores possíveis para  $r_j$  (release date do job  $\{j \in J^2\}$ ), considerando as relações de precedência existentes.

Vale enfatizar que o método de geração de colunas pode identificar um limitante superior, um limitante inferior ou ainda nenhum valor que se encaixe nas duas classificações anteriores. Caso o SB não garanta a inexistência de colunas capazes de melhorar a solução do PMR, ou seja, o custo reduzido negativo, não é possível garantir a otimalidade da solução fornecida pelo PMR. Dessa forma, não há garantias com relação ao valor obtido, porém, caso o SB seja capaz de garantir que não exista coluna que possa causar uma melhora na solução do PMR, pode-se obter um limitante inferior ou superior. Para o caso do cálculo de um limitante inferior, o PMR fornece uma solução, porém não factível, ou seja, não viável para o problema original devido ao fato do PMR ser uma relaxação. Destaca-se, portanto, que o PMR é uma relaxação, porque não são consideradas todas as variáveis, motivo pelo qual o problema mestre é chamado de restrito. Já a obtenção do limitante superior ocorre quando o PMR possui solução viável. Nesse caso, como o SB garantiu que não existe coluna capaz de provocar uma

melhora da solução do PMR, pode-se dizer que a solução ótima foi obtida.

O PMR é descrito pelo conjunto de equações expostas abaixo, onde  $y_{jt}$  representa se um job  $\{j \in J^2\}$  será processado na máquina 2 no instante de tempo  $t$ . Já  $\lambda_c$  representa a variável de convexificação. Além das variáveis citadas, o modelo possui os seguintes parâmetros:  $r_j^c$ , que indica a data a partir da qual um job  $\{j \in J^2\}$  pode iniciar seu processamento, levando-se em conta o tempo de conclusão de seus jobs precedentes na máquina 1, ou seja, o *release date* do job  $\{j \in J^2\}$ . Cada coluna gerada  $c$  será inserida no conjunto  $c'$  ( $c' \subset C$ ), onde todas as colunas serão armazenadas. Dessa forma,  $r_j^c$  indica todos os  $c$  vetores de  $r_j$ . O conjunto  $J^2$  indica os jobs que devem ser processados pela máquina 2, o horizonte de tempo é representado por  $T$  e, por último,  $p_{2j}$  se refere ao tempo de processamento do job  $\{j \in J^2\}$  na máquina 2.

$$\{PMR\} \quad \min Z = \sum_{j \in J^2} \sum_{t=0}^{T-p_{2j}} (t + p_{2j}) y_{jt}$$

sujeito a

$$\sum_{t=0}^{T-p_{2j}} y_{jt} = 1 \quad \forall j \in J^2 \quad (5.16)$$

$$\sum_{j \in J^2} \sum_{s=\max(0; t-p_{2j}+1)}^t y_{js} \leq 1 \quad \forall t \in T \quad (5.17)$$

$$\sum_{t=0}^{T-p_{2j}} t y_{jt} - \sum_{c=1}^C r_j^c \lambda_c \geq 0 \quad \forall j \in J^2 \quad (5.18)$$

$$\sum_{c=1}^C \lambda_c = 1 \quad (5.19)$$

$$0 \leq y_{jt} \leq 1 \quad \forall j \in J^2 \wedge \forall t \in T \quad (5.20)$$

$$0 \leq \lambda_c \leq 1 \quad \forall c \in C \quad (5.21)$$

A função objetivo busca minimizar a soma dos tempos de conclusão ponderada dos jobs  $\{j \in J^2\}$  na máquina 2. A restrição 5.16 garante que todo job  $\{j \in J^2\}$  seja processado uma única vez. A restrição seguinte 5.17 inibe que dois jobs sejam processados ao mesmo tempo pela máquina 2. A inequação 5.18 garante que nenhum job  $\{j \in J^2\}$  inicie seu processamento antes da finalização do processamento dos seus respectivos jobs precedentes na máquina 1  $\{k \in S_j\}$ , onde  $S_j \forall \{j \in J^2\}$  é o conjunto

precedência, no qual  $\{k \in J^1\}$  são os jobs precedentes. Já a restrição 5.19 define que o somatório de todas as variáveis  $\lambda_c$  com  $c \subset C$  seja igual a 1. E as restrições 5.20 e 5.21 definem as variáveis  $y_{jt}$  e  $\lambda_c$  respectivamente.

O subproblema é responsável por gerar as colunas inseridas no PMR e é modelado de acordo com o problema de precificação referente ao mesmo. Sendo assim, os parâmetros  $\alpha_j \forall \{j \in J^2\}$  são representados pelas variáveis duais obtidas por meio da restrição 5.18 e  $\theta$  é representado pela variável dual referente a restrição 5.19. Os demais parâmetros já foram explicados anteriormente. Já as variáveis são representadas por  $r_j$ , ou seja, o *release date* para cada job  $\{j \in J^2\}$  e  $x_{jt}$ , que define se um job  $\{j \in J^1\}$  será processado na máquina 1 no instante de tempo  $t$ .

$$\{SB\} \quad \min \sum_{j \in J^2} r_j \alpha_j - \theta$$

sujeito a

$$\sum_{t=0}^{T-p_{1j}} x_{jt} = 1 \quad \forall j \in J^1 \quad (5.22)$$

$$\sum_{j \in J^1} \sum_{s=\max(0; t-p_{1j}+1)}^t x_{js} \leq 1 \quad \forall t \in T \quad (5.23)$$

$$r_j \geq \sum_{t=0}^{T-p_{1k}} (t + p_{1k}) x_{kt} \quad \forall j \in J^2 \wedge \forall k \in S_j \quad (5.24)$$

$$x_{jt} \in \{0, 1\} \quad \forall j \in J^1 \wedge \forall t \in T \quad (5.25)$$

$$r_j \geq 0 \quad \forall j \in J^2 \quad (5.26)$$

A função objetivo busca minimizar o problema de precificação relacionado ao PMR. As inequações 5.22 garantem que todo job  $\{j \in J^1\}$  seja processado uma única vez. As restrições 5.23 não permitem que mais de um job seja processado ao mesmo tempo na máquina 1. As restrições seguintes, 5.24, calculam o limite inferior para um dado  $r_j \{j \in J^2\}$ . Vale destacar que  $S_j$  é o conjunto de precedência, assim, para cada relação de precedência existente na instância, cria-se uma restrição desse conjunto, na qual o período de início do job  $j \in J^2$  deve ser maior ou igual à data de conclusão de seu precedente  $J_k^1 \in S_j$ . As restrições 5.25 e 5.26 definem as variáveis  $x_{jt}$  e  $r_j$ .

Destaca-se que o método de geração de colunas será iniciado a partir da seleção de uma sequência aleatória de jobs na máquina 1, ou seja, uma solução viável para as variáveis  $x_{jt}$ . Dessa forma, o PMR é resolvido pela primeira vez, as variáveis duais  $\alpha_j$  e  $\theta$  são armazenadas, e o método de geração de colunas é iniciado. A figura 5.2 ilustra a dinâmica entre o PMR e o SB presente no método de geração de colunas. Além disso, é ressaltado que as variáveis  $y_{jt}$  são definidas como fracionárias (5.20) no PMR, porém no modelo original as mesmas são declaradas como inteiras (3.8). Assim, deve-se obter uma solução inteira para que a mesma seja viável no problema original.

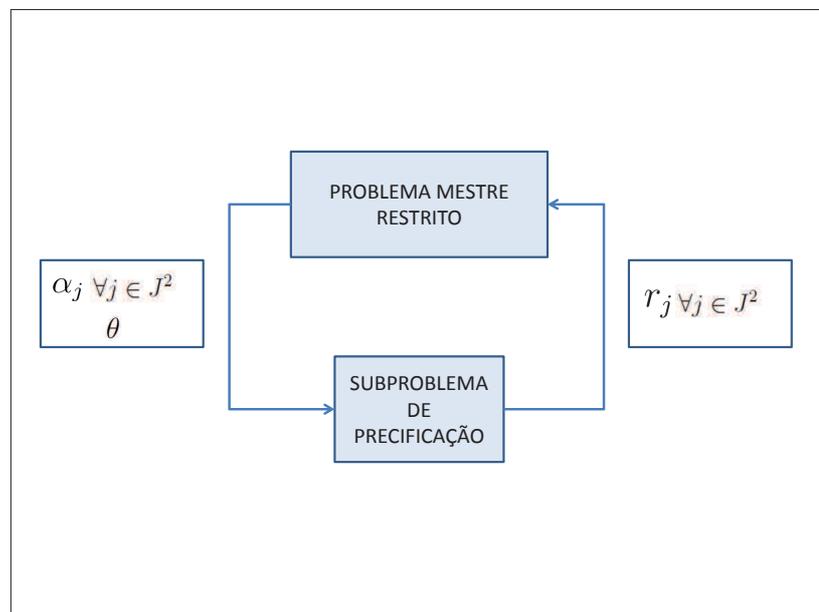


Figura 5.2: Dinâmica de trabalho entre o PMR e o SB para o problema em questão

O modelo SB (5.22 -5.26) pode ser associado a um problema de sequenciamento em uma única máquina. A função objetivo busca minimizar a soma ponderada das variáveis  $r_j$ , que são definidas por meio do sequenciamento estabelecido pelas variáveis  $x_{jt}$  e pelo conjunto de precedência  $S_j$ . Assim, dado que  $r_j$  é a data de conclusão mais tardia de um dos jobs que compõem o respectivo conjunto  $S_j$ , a função objetivo avalia qual é a melhor ordem de sequenciamento dos jobs  $J_k^1 \in S_j$ , de forma a minimizar o valor do termo  $\sum_{j \in J^2} r_j \alpha_j$ , avaliando em conjunto a ponderação estabelecida por  $\alpha_j$ . O valor de  $\theta$ , por se tratar de uma constante, não interfere no processo de otimização do modelo, que será realizado utilizando-se como base o *solver* CPLEX. A descrição da forma como será conduzida a resolução do método abordado será exposta adiante.

### 5.2.2 Método de Geração de Colunas - Heurística proposta

A introdução da heurística tem como objetivo resolver o SB, porém buscando reduzir o tempo total de processamento no processo de obtenção da solução. A forma como a heurística proposta trabalha é fornecendo colunas para o PMR, as mesmas são geradas a partir de um critério que busca identificar possíveis boas sequências de processamento dos job na máquina 1. Assim, uma vez que tais colunas são selecionadas criteriosamente, espera-se que os valores dos custos duais obtidos no PMR e repassados para o SB sejam melhores do que aqueles gerados por sequências aleatórias, melhores no sentido de reduzir o número de iterações necessárias para que o SB identifique a sequência ótima de processamento dos jobs na máquina e, com isso, reduza o número de colunas geradas pelo SB, acelerando o processo de resolução do mesmo.

Os procedimentos que compõem a heurística serão descritos por meio de um exemplo, a fim de demonstrar ao leitor a sua dinâmica. Em seguida, o pseudo-código relativo a heurística é exposto.

A tabela 5.1 é composta pelos dados da instância a ser trabalhada. Nesse caso, o número de caminhões que chegam para descarregar no terminal de *Crossdocking* são 3 ( $n=3$ ), a quantidade de caminhões que serão carregados na doca de saída são 2 ( $m=2$ ). Os tempos de processamento de cada caminhão a ser trabalhado na doca de entrada e na doca de saída estão expostos pelas colunas  $P_1$  e  $P_2$ , respectivamente. Por último, o conjunto de  $S_j$  indica os jobs ( $j \in J_1$ ) precedentes dos caminhões que serão carregados.

Tabela 5.1: Exemplo de Aplicação da heurística proposta

Tabela de Dados				
N	M	P1	P2	S <sub>j</sub>
		3	1	{1, 2}
3	2	2	2	{1, 3}
		3	-	

De posse dos dados da instância, o primeiro passo é a identificação de uma solução inicial, ou seja, uma sequência que seja viável para a máquina 1. Essa sequência irá possibilitar que o próximo passo seja dado, que é solucionar o PMR e a consequente coleta dos valores dos custos duais  $\alpha_j \forall \{j \in J^2\}$ .

Dessa forma, considerando a instância apresentada na tabela 5.1 e realizando os cálculos relativos aos itens da heurística, que serão apresentados nessa seção, tem-se a tabela 5.2:

Tabela 5.2: Resultado iteração 1 da heurística proposta

Item	$\alpha_1$	$\alpha_2$	$GS_1$	$GS_2$	$PrGS_1$	$PrGS_2$	Peso1	Peso2
Valor	1	2	{1, 2}	{1, 3}	$P_{11}+P_{12} = 3$	$P_{11}+P_{13} = 4$	3	2

Destaca-se que o peso 1 e 2 são calculados da seguinte forma: peso 1 é dado pela razão  $PrGS_1/\alpha_1 = 3$  e o peso 2 é dado por  $PrGS_2/\alpha_2 = 2$ . O Vetor Peso possui dois componentes, pois o conjunto  $J^2$  possui dois jobs. Além disso, de acordo com a tabela os valores das variáveis duais  $\alpha_1$  e  $\alpha_2$  são iguais a 1 e 2, respectivamente. O Grupo matriz de precedência do job  $j_1$  ( $GS_1$ )  $\{j \in J^2\}$  é composto pelos jobs  $\{j_1$  e  $j_2\}$   $\{j \in J^1\}$ , ou seja,  $GS_1 = \{j_1, j_2\}$ , o que significa que os jobs  $j_1$  e  $j_2$  devem ser descarregados na doca de entrada para que o job  $j_1$   $\{j \in J^2\}$  seja carregado na doca de saída. Já  $GS_2 = \{j_1, j_3\}$ . O valor dado por ( $PrGS_j$ )  $\{j \in J^2\}$  é a soma dos tempos de processamento dos jobs precedentes ao job  $\{j \in J^2\}$  em questão, ou seja, a soma dos job presentes em ( $GS_j$ )  $\{j \in J^2\}$ . Assim, para  $PrGS_1$  o valor é igual a soma dos tempos de processamento dos jobs  $\{j_1$  e  $j_2\}$ , jobs pertencentes a  $J^1$ . Da mesma forma,  $PrGS_2$  se refere aos tempos de processamento dos jobs  $\{j_1$  e  $j_3\}$  na máquina 1.

Concluído os cálculos, deve-se selecionar o peso cujo o valor é o menor dentre todos, neste caso é o peso 2, que é relativo ao  $j_2 \in J^2$ . Dado que este job possui como precedentes os jobs  $j_1$  e  $j_3 \in J^1$ , os mesmos serão alocados no vetor chamado sequência ( $S = \{1, 3\}$ ). Uma vez que estes jobs já foram sequenciados, é necessário que sejam retirados dos grupos de precedência ( $GS_j$ )  $\{j \in J^2\}$  para que não sejam mais considerados, além disso, o job, cujo peso foi selecionado, também será desconsiderado como candidato da "lista de pesos", lista que contém todos os pesos associados aos respectivos jobs  $\in J^2$ . Caso haja algum job  $\{j \in J^2\}$  cujo jobs precedentes já estão no vetor sequência, o mesmo também pode ser retirado da "lista de pesos".

A tabela 5.3 contém o resultado de cada item após a primeira iteração.

Tabela 5.3: Resultado iteração 2 da heurística proposta

Item	$\alpha_1$	$\alpha_2$	$GS_1$	$GS_2$	$PrGS_1$	$PrGS_2$	Peso1	Peso2
Valor	1	2	2	$\{\emptyset\}$	$P_{12} = 1$	0	$PrGS_1/\alpha_1 = 1$	Retirado Não Calculado

Na tabela 5.3 percebe-se que  $GS_1$  contém apenas o  $j_2 \in J^1$ , pois o  $j_1 \in J^1$  já havia sido sequenciado. Além disso,  $GS_2$  está vazio, pois seus jobs precedentes foram todos sequenciados na iteração 1. O peso a ser selecionado nessa iteração será aquele relativo

ao job  $j_1 \in J^2$ , pois é o único remanescente. Uma vez que  $GS_1$  é composto por apenas o  $j_2 \in J^1$ , este será acrescido ao vetor sequência, que será  $S = \{1, 3, 2\}$ . Como todos os pesos relativos aos jobs da máquina 2 já foram considerados e os jobs na máquina 1 estão sequenciados (vetor sequência), a heurística encerra-se e, como resultado, uma sequência de processamento para a máquina 1 é gerada de forma criteriosa.

A figura 5.3 expõem os passos seguidos no exemplo discutido por meio de um pseudo-código: float

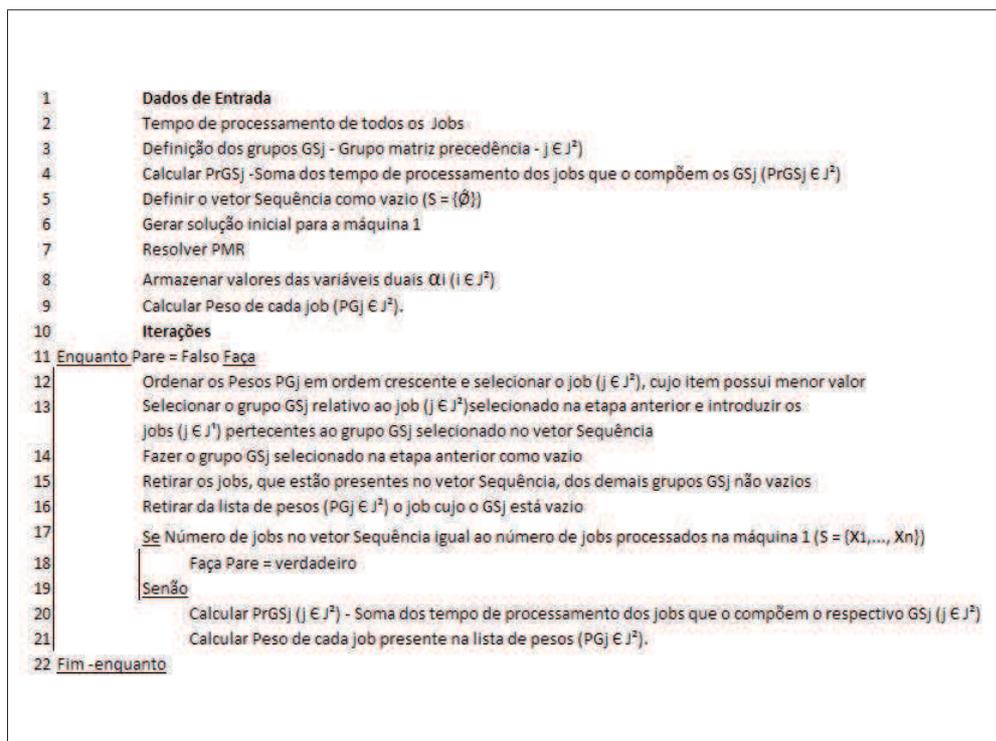


Figura 5.3: Pseudo-Algoritmo heurística proposta

A heurística descrita busca sequenciar os jobs na máquina 1 baseando-se na ponderação  $PrGS_j / \alpha_j$ , ou seja, é analisado a soma corrente dos tempos de processamento dos jobs  $\{j \in J^1\}$  que compõem o conjunto de precedência de determinado job ( $PrGS_j$ )  $\{j \in J^2\}$  e os custos  $\alpha_j$   $\{j \in J^2\}$  fornecidos pelo PMR. Os valores de  $PrGS_j$  são recalculados a cada iteração, pois a medida que os jobs da máquina 1 são sequenciados, os mesmos são desconsiderados para o cálculo dos pesos, mesmo que façam parte do conjunto  $GS_j$  inicial. Da mesma forma, a medida que um job  $j \in J^2$  é selecionado por apresentar um peso com menor valor, seu respectivo  $GS_j$  também é desconsiderado nas iterações seguintes. Assim, ao final obtém-se uma sequência de jobs

a serem processados na máquina 1. Pode-se perceber que os grupos  $(G_j) \forall \{j \in \mathcal{J}^1\}$ , que indicam para quais jobs  $\{j \in \mathcal{J}^2\}$  o job  $\{j \in \mathcal{J}^1\}$  é precedente, não foram utilizados da descrição exposta, porém são importantes para a implementação, uma vez que são referências para consultas.

# Capítulo 6

## Testes Geração de Colunas

Nesse Capítulo são descritos os testes feitos envolvendo o método geração de colunas e a heurística citados no Capítulo 5. Assim, são descritos nas Seções seguintes cada teste realizado, os respectivos resultados e as análises dos mesmos.

### 6.1 Instâncias

Os testes envolvendo geração de colunas foram executados em uma máquina com processador Intel(R) Xeon (R) CPU X5450@2,40GHz, com 12 processadores, 12 GB de memória RAM em sistema operacional Linux Ubuntu 13.10. Foram utilizadas as linguagens de programação C, C++ e AMPL versão 2012911(Linux x86 64) e o software de otimização CPLEX versão 12.4.

O tempo de execução foi limitado a 1 hora de processamento, caso a solução ótima seja encontrada em menos de 1 hora, a execução é interrompida. Após esse tempo são registrados os seguintes dados destacado na tabela 6.1:

A tabela 6.1 destaca que são coletados informações relativas ao PMR e o SB especificamente, ou seja, foi avaliado o limite inferior do SB por meio do resultado de sua função objetivo  $FO(SB)$  e de seu  $GAP(SB)$ . Assim, definido o limitante inferior para o problema é calculado o GAP para o método geração de colunas, de acordo com a instância em questão. O GAP do método é definido utilizando o limite inferior e também o resultado da função objetivo do PMR, interpretada como um limitante superior.

A tabela 6.2 descreve as instâncias utilizadas para os experimentos realizados

Tabela 6.1: Resumo dos dados coletados para cada instância avaliada. A coluna *Modelo* refere-se as formulações utilizadas para cada instância, que podem ser PMR, SB ou GC, que significam Problema Mestre Restrito, Subproblema e Geração de Colunas, respectivamente. Já as siglas FO, T(s) e GAP significam Função Objetivo, Tempo de processamento medido em segundo e GAP de otimalidade, respectivamente. Destacando que o GAP baseia-se na fórmula ( $\frac{LS - LI}{LS}$ ), porém para os casos em que a solução final não foi encontrada quando o processamento foi interrompido, o GAP é calculado de acordo com a fórmula ( $\frac{LS - \bar{LI}}{LI}$ ). Assim, ( $LI = \frac{LS}{GAP - 1}$ ).

Modelo	Dados Coletados
Modelo Original	FO, GAP e T(s)
Relaxação Linear	FO, GAP e T(s)
GC	FO(SB), FO(PMR), GAP(SB) e T(s)

envolvendo o método de geração de colunas.

Os valores dos tempos de processamento dos jobs, presentes nos grupos  $J^1$  ou  $J^2$ , foram definidos de forma aleatória, de acordo com a respectiva distribuição uniforme definida na tabela. O número máximo de precedentes foi estabelecido anteriormente, assim como os valores que representam  $n$  e  $m$ . As configurações definidas buscam avaliar os resultados quando a quantidade de jobs é alterada. Além disso, é analisado as interferências relativas ao tempo de processamento dos jobs e ao número de precedentes de um job que será processado pela máquina 2.

## 6.2 Descrição Teste 1 - Geração de Colunas

O teste em questão aborda o método de geração de colunas utilizando como base os modelos PMR ( 5.16 - 5.21 ) e o SB ( 5.22 - 5.26 ). Assim, uma solução inicial aleatória é gerada para inicialização do método, ou seja, uma sequência jobs para a máquina 1. Em seguida, o PMR é resolvido, os valores das variáveis duais calculadas são transferidos como parâmetros para o SB, que avalia se o conjunto de soluções corrente é suficiente para identificar a solução ótima, por meio do subproblema de precificação (SB), que é solucionado pelo *solver* CPLEX, assim como o PMR. Caso o resultado do SB for negativo, significa que há possibilidade de melhoria acrescentando ao conjunto de colunas geradas outra coluna, por outro lado, se a solução do SB for não negativa, significa que a solução ótima foi encontrada e o método é encerrado. A dinâmica descrita pode ser observada na figura 6.1.

Tabela 6.2: Sumário das instâncias geradas para teste, separadas em grupo 1 e 2, e subgrupos, de acordo com o número de jobs na máquina 1 ( $JobsM1(n)$ ). Além disso, a tabela informa o número de jobs na máquina 2 ( $JobsM2(m)$ ), o número máximo de precedentes do job  $j \in J^2 (NP)$ , o tempo de processamento dos jobs ( $TP$ ), o código da instância (*Código*) e por último quantas instâncias foram geradas para cada subgrupo ( $N.Inst$ ).

Grupo	Jobs M1(n)	Jobs M2(m)	NP	TP	Código	N. Inst.
1	5	2-3-4	3-4	U(1,5)	5-m-NP-1	8
	10	3-5-7	5-7	U(1,5)	10-m-NP-1	8
	15	5-8-11	5-8-11	U(1,5)	15-m-NP-1	9
	20	6-10-14	6-10-14	U(1,5)	20-m-NP-1	9
	50	15-25-35	15-25-35	U(1,5)	50-m-NP-1	9
	60	18-30-42	18-30-42	U(1,5)	50-m-NP-1	9
2	5	2-3-4	3-4	U(1,50)	5-m-NP-2	8
	10	3-5-7	5-7	U(1,50)	10-m-NP-2	8
	15	5-8-11	5-8-11	U(1,50)	15-m-NP-2	9
	20	6-10-14	6-10-14	U(1,50)	20-m-NP-2	9
	50	15-25-35	15-25-35	U(1,50)	50-m-NP-2	9
	60	18-30-42	18-30-42	U(1,50)	50-m-NP-2	9

É importante destacar que o SB é o responsável por identificar se há possibilidade de melhora da solução do PMR. A possibilidade de melhora ocorre quando o resultado da função objetivo do SB é inferior a zero, ou seja, o custo reduzido é negativo. Dessa forma, pode-se dizer que não é necessário que o SB seja resolvido no ótimo, pois caso o limite inferior calculado durante o processo de otimização seja não negativo, é possível afirmar que a solução ótima do SB também não será negativa e, conseqüentemente, o custo reduzido será maior ou igual a zero. Esse recurso é válido, pois o tempo de processamento do SB pode ser muito longo para instâncias maiores, logo, o tempo para se obter uma conclusão válida pode ser reduzido, melhorando o desempenho do método.

### 6.2.1 Resultados Testes 1 - Geração de Colunas

Os resultados do teste envolvendo a geração de colunas, sem a introdução da heurística, são expostos em paralelo com os resultados do modelo original e de sua relaxação linear, o objetivo é facilitar a comparação entre os resultados e avaliação do método descrito. A tabela 6.3 expõem alguns dos resultados do teste citado, as mesmas instâncias analisadas no capítulo 4. As demais instâncias listadas na tabela

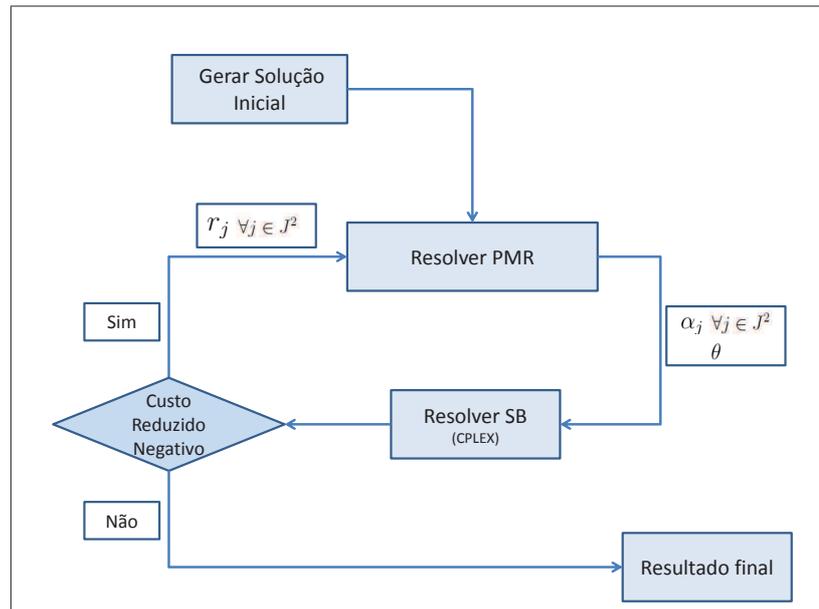


Figura 6.1: Dinâmica Teste Geração de Colunas - Teste 1

6.2 são disponibilizados no Anexo A.

O valor do limitante inferior calculado na tabela 6.3 é feito com base nos valores da solução do SB e do PMR. Assim, quando a FO do SB é igual ou maior que zero, ou seja, não há colunas para serem acrescentadas para melhorar a solução do PMR, é definido que  $LI = LS$ . Por outro lado, quando a FO do SB é negativa, esse valor é somado a FO do PMR para obter o valor do LI.

## 6.2.2 Análise Resultados Teste 1 - Geração de Colunas

No teste realizado utilizando o método de geração de colunas sem a heurística, percebe-se na tabela 6.3 que os resultados das instâncias do grupo 1 são iguais ou próximos da solução ótima, para os casos nos quais o valor de  $n$  é igual ou menor do que 15, ou seja, 15 jobs sequenciados na máquina 1. O mesmo ocorre para as instâncias do grupo 2, porém para aqueles casos nos quais o valor de  $n$  é igual a 5. Nas demais instâncias é perceptível a dificuldade do modelo original e do método de geração de colunas identificarem as respectivas soluções ótimas no intervalo de tempo estabelecido de 3600 segundos. A análise por meio do limitante inferior do SB não permite chegar a conclusões diferentes daquelas relatadas acima, pois o valor do limite inferior do SB, assim como a função objetivo do mesmo, também é negativo. Outro ponto importante é relativo as instâncias para as quais a geração de colunas obteve o valor do GAP maior do que zero. Quando essa situação ocorre significa que não é possível chegar a uma conclusão válida, pois a função objetivo do SB ainda é negativa e, portanto, o problema não foi resolvido na otimalidade. Assim, apenas 11 das 30 instâncias apresentam o valor do GAP igual a zero, nenhuma com  $n$  superior a 15.

Com relação aos testes realizados com o modelo completo, constata-se que não foi identificada a solução ótima para as maiores instâncias do grupo 1. No grupo 2, apenas as menores instâncias tiveram sua solução ótima identificada. Esse fato expõe a dificuldade de resolução de modelos com indexação no tempo, uma vez que os parâmetros entre os dois grupos são os mesmos, com exceção do tempo de processamento.

## 6.3 Descrição Teste 2 - Heurística e Geração de Colunas com garantia de Otimalidade

Nessa seção é descrita a interação entre a heurística e o método de geração de colunas, que terá o SB resolvido pelo CPLEX ao longo do processo, a fim de garantir a otimalidade da solução fornecida. O objetivo é avaliar os valores das soluções fornecidas pelo modelo original e compará-las com valores obtidos pelo método que associa a geração de colunas e a heurística. A figura 6.2 ilustra a interação entre o método de geração de colunas e a heurística. O experimento descrito a seguir será denominado *Teste 2*.

Durante o Teste 2 descrito pela figura 6.2, a heurística é responsável por inserir

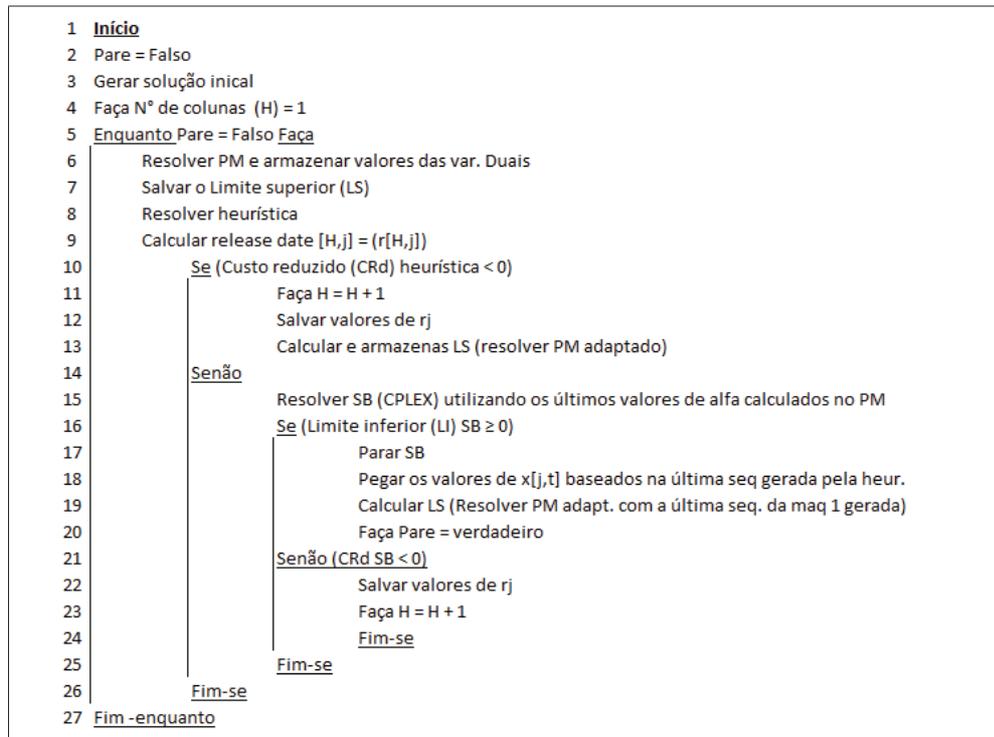


Figura 6.2: Pseudo-Algoritmo Teste 2

colunas para o modelo PMR. A cada coluna inserida, o subproblema de precificação é resolvido, caso o valor do custo reduzido seja negativo, ou seja, ainda há possibilidade de melhora da solução, mais colunas são introduzidas por esse método. É ressaltado que até o momento, o problema de precificação não é solucionado com garantia de otimalidade, ou seja, os valores das colunas do PMR são inseridos por meio da heurística. Dessa forma, não há garantia de otimalidade nessa etapa. Por outro lado, quando o custo reduzido desse conjunto de colunas é maior ou igual a zero, o SB é resolvido com o objetivo de avaliar se o atual conjunto é suficiente para fornecer a solução ótima, ou seja, comprovar a otimalidade da solução corrente. Em seguida, o SB é resolvido pelo CPLEX e caso a solução seja maior ou igual a zero, a mesma pode ser considerada como solução ótima, caso contrário, uma nova coluna é introduzida ao conjunto de colunas e inicia-se o processo.

A dinâmica descrita pode ser observada na figura 6.3.

Destaca-se que sempre que uma nova coluna é inserida, o modelo descrito abaixo (6.1 - 6.4) é solucionado de forma que seja obtida uma nova solução viável ou um novo limitante superior. Esse modelo foi definido por meio da adaptação do modelo original ( 3.2 - 3.8 ), pois os valores das variáveis  $x_{jt}$  foram introduzidos como

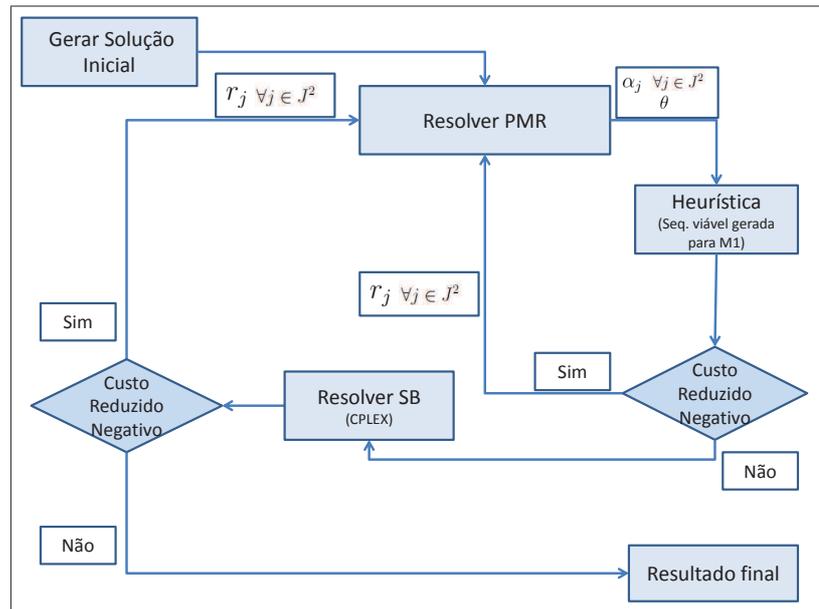


Figura 6.3: Dinâmica Teste 2 - Heurística e Geração de Colunas com garantia de Otimalidade

parâmetros, ou seja, os valores de  $x_{jt}$  são fixados de acordo com a sequência definida pela heurística ou pelo SB. Assim, quando nenhuma coluna for mais acrescentada, o modelo adaptado é o responsável por calcular os valores viáveis para as variáveis  $y_{jt}$ , que podem ser os valores da solução ótima, para o caso no qual o SB garantir a otimalidade ou o custo reduzido não negativo. Outra possibilidade é o cálculo de apenas uma solução viável, para o caso no qual não há a garantia dada pelo SB citada anteriormente. O modelo adaptado é definido abaixo:

$$\min Z = \sum_{j \in J^2} \sum_{t=0}^{T-p_{2j}} (t + p_{2j}) y_{jt}$$

sujeito a

$$\sum_{t=0}^{T-p_{2j}} y_{jt} = 1 \quad \forall j \in J^2 \quad (6.1)$$

$$\sum_{j \in J^2} \sum_{s=\max(0; t-p_{2j}+1)}^t y_{js} \leq 1 \quad \forall t \in T \quad (6.2)$$

$$\sum_{t=0}^{T-p_{2j}} ty_{jt} - \sum_{t=0}^{T-p_{1k}} (t+p_{1k})x_{kt}^* \geq 0 \quad \forall j \in J^2 \wedge \forall k \in S_j \quad (6.3)$$

$$y_{jt} \in \{0, 1\} \quad \forall j \in J^2 \wedge \forall t \in T \quad (6.4)$$

### 6.3.1 Resultados Teste 2

Os resultados associados ao Teste 2 são expostos na tabela 6.4.

### 6.3.2 Análise Resultados Teste 2

Percebe-se que os resultados associados as instâncias do grupo 1 foram melhores do que aqueles descritos na subseção 6.2.2, pois as soluções fornecidas pela heurística são mais próximas das respectivas soluções calculadas pelo modelo original. Além disso, observa-se na tabela 6.4 que o GAP estimado ao final do processamento do Teste 2 é igual a zero em 25 das 30 instâncias avaliadas. A avaliação do limitante inferior relativo somente ao SB desempenhou um papel muito importante nesses resultados, pois a partir desse valor foi possível estabelecer se a função objetivo do SB é não negativa e, assim, garantir que  $LI = LS$ .

Além disso, o teste obteve, em 9 oportunidades no total de 11, o limite obtido igual ao valor da solução ótima do problema. Pois foi considerado apenas as 11 instâncias resolvidas na otimalidade pelo modelo original em até 3600 segundos, ou seja, valor do GAP igual a zero.

## 6.4 Descrição Teste 3 - Heurística e Geração de Colunas sem garantia de Otimalidade

O próximo teste é ilustrado pela figura 6.4 será denominado de *Teste 3*.

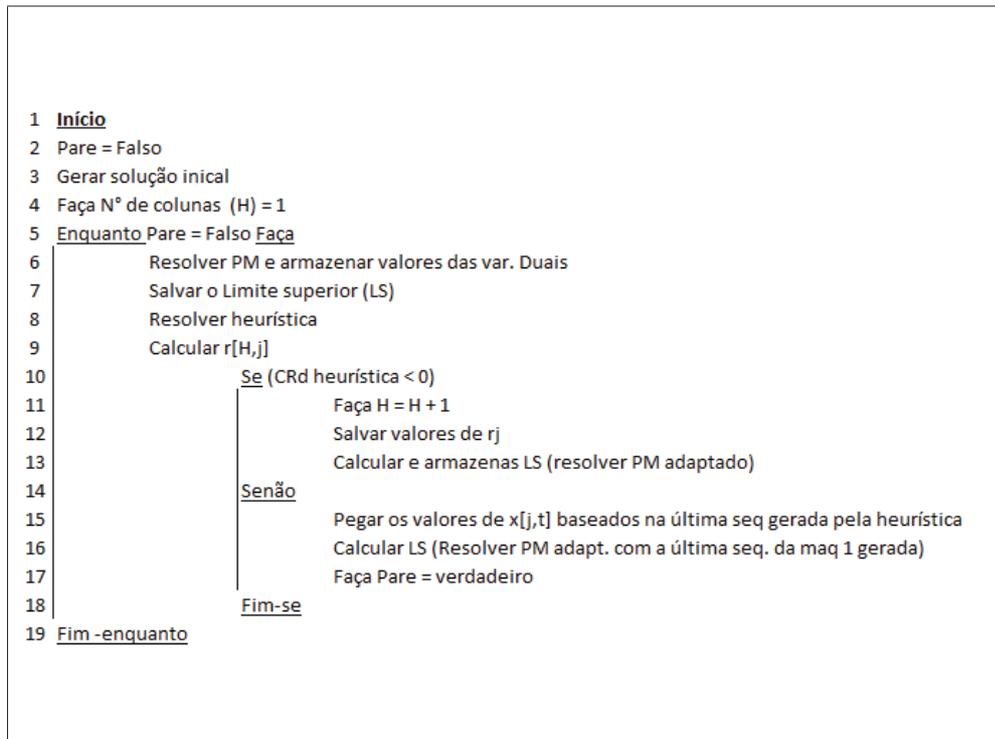


Figura 6.4: Pseudo-Algoritmo Teste 3

É proposto solucionar o problema, porém não é abordada a resolução do SB com o solver *CPLEX*, mas por meio da heurística. Consequentemente, a otimalidade da solução não pode ser garantida. O objetivo do teste proposto é avaliar o quão próximo a heurística é capaz de chegar da solução obtida no Teste 2 e do modelo original, com isso, é possível avaliar se o critério adotado para seleção de colunas é adequado. Caso a diferença entre as soluções dos Testes 2 e 3 for elevada, pode significar que o SB contribui significativamente para a obtenção de boas colunas e não a heurística. O método encerra-se quando o valor do custo reduzido, calculado com base no conjunto de colunas fornecidas pela heurística, for igual a zero ou positivo. Em seguida o modelo 6.1 - 6.4 é resolvido e a solução ótima desse modelo é armazenada.

A dinâmica descrita pode ser observada na figura 6.5.

Destaca-se dois pontos quanto ao Teste 3, o primeiro é com relação a sequência de jobs abordada pela mesma, pois é idêntica a aquela utilizada no Teste 2, anteriormente a entrada do SB. O Segundo ponto é que esse experimento fornece ao final de seu processamento uma solução viável para o problema e, portanto, um valor para o limite superior. Valor que não possui garantia de otimalidade, como já discutido.

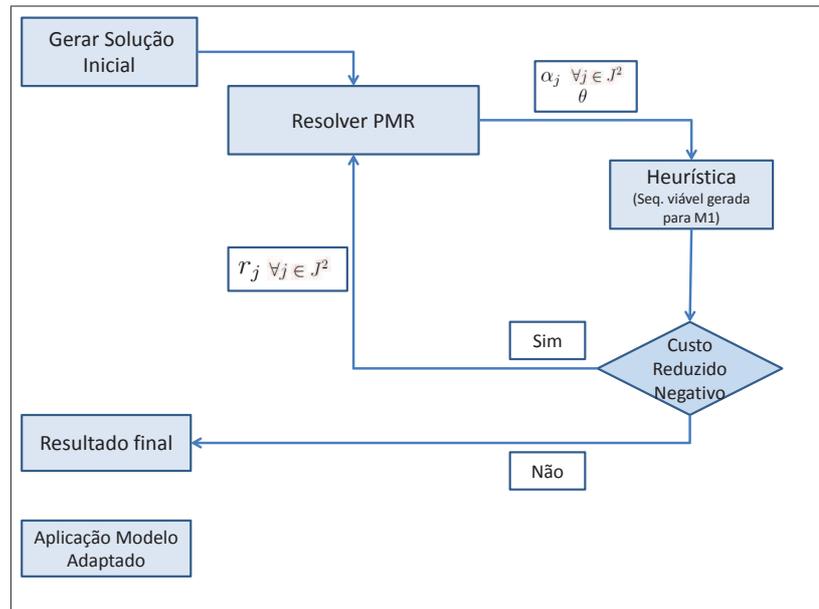


Figura 6.5: Dinâmica Teste 3 - Heurística e Geração de Colunas sems garantia de Otimalidade

## 6.5 Resultados Teste 3

Os resultados do Teste 3 são expostos na tabela 6.5.

### 6.5.1 Análise Resultados Teste 3

O último teste realizado envolve a aplicação da Heurística e parte do mesmo princípio aplicado para o Teste 2, porém sem a utilização do SB resolvido pelo *solver* CPLEX. O objetivo do Teste 3 é avaliar a qualidade da solução fornecida utilizando a heurística para solucionar o SB e, dessa forma, sem garantir de otimalidade da solução.

Os resultados obtidos através desse método se mostraram mais rápidos de serem calculados do que nos outros experimentos, como mostra a tabela 6.5. Com relação a solução obtida, esta foi bem próxima a solução calculada pelo Teste 2, sendo que em 10 das 30 instâncias o valor foi o mesmo. Outro dado importante, é a variação percentual entre as soluções fornecidas pelos Testes 2 e 3, pois a maior diferença encontrada foi de 11%, ou seja, o valor obtido pelo Teste 3 foi até 11% maior do que aquele encontrado pelo Teste 2. Vale destacar que o valor obtido no Teste 3 será sempre maior ou igual do que no experimento anterior, pois enquanto um calcula uma solução viável o outro procurar solucionar uma relaxação.

A tabela 6.6 expõem os resultados obtidos com os Testes 2 e 3 e, em paralelo, a solução do modelo original para facilitar a comparação.

Tabela 6.3: Comparação entre Formulação Original, Relaxação Linear e Geração de Colunas. A coluna (*Código*) refere-se a instância trabalhada, os itens (*RL*) e (*GC*) significam Relaxação Linear e geração de colunas, respectivamente. Já as siglas (*FO*), (*LI*), (*LS*), (*T*) e (*GAP*) representam a Função Objetivo, valor do Limitante Inferior, valor do Limitante Superior, Tempo de processamento em segundos e o GAP de otimalidade do método ( $\frac{LS - LI}{LS}$ ) respectivamente. Foi apresentada a solução ótima para relaxação linear.

Código	Modelo original			RL		GC			
	FO	GAP	T(seg)	FO	T(seg)	LI	LS	GAP	T(seg)
5-2-3-1	34	0,0%	0	28	0	34	34	0,0%	2
5-3-3-1	28	0,0%	0	26	0	28	28	0,0%	37
5-4-3-1	50	0,0%	0	44	0	48	48	0,0%	29
10-3-5-1	61	0,0%	3	50	0	61	61	0,0%	11
10-5-5-1	113	0,0%	45	98	0	113	113	0,0%	497
10-7-5-1	157	0,0%	50	142	0	157	157	0,0%	471
15-5-8-1	149	0,0%	3600	133	0	149	149	0,0%	3402
15-8-8-1	240	3,3%	3600	209	0	206	238	13,5%	5682
15-11-8-1	347	10,6%	3600	293	0	338	342	1,1%	3780
20-6-10-1	190	0,0%	3350	169	0	53	190	72,0%	3602
20-10-10-1	390	20,1%	3600	383	3	354	480	26,3%	3602
20-14-10-1	481	15,4%	3600	477	1	430	559	23,2%	3602
50-15-15-1	1383	12,7%	3600	1207	1	695	1351	48,6%	3602
50-25-15-1	3093	33,0%	3600	2071	7	2139	2749	22,2%	3602
50-35-15-1	4725	35,7%	3600	3037	8	3480	4071	14,5%	3603
5-2-3-2	226	0,0%	2	186	0	226	226	0,0%	32
5-3-3-2	436	0,0%	4	402	0	436	436	0,0%	43
5-4-3-2	327	0,0%	0	323	0	325	325	0,0%	6
10-3-5-2	361	6,0%	3600	320	1	342	342	0,0%	855
10-5-5-2	882	9,0%	3600	789	3	645	862	25,2%	4019
10-7-5-2	1000	5,7%	3600	926	2	861	996	13,6%	3602
15-5-8-2	1705	17,2%	3600	1403	14	1044	1682	37,9%	3605
15-8-8-2	2055	28,9%	3600	1454	9	1239	1731	28,4%	3602
15-11-8-2	3907	24,3%	3600	2955	8	2766	3794	27,1%	3604
20-6-10-2	1859	24,6%	3600	1397	5	582	1586	63,3%	3602
20-10-10-2	3642	22,2%	3600	2832	9	2521	3327	24,2%	3604
20-14-10-2	4866	27,8%	3600	3510	3	3361	4190	19,8%	3619
50-15-15-2	7950	24,5%	3600	6002	41	3331	7082	53,0%	3618
50-25-15-2	18699	47,8%	3600	9749	33	11828	14187	16,6%	3613
50-35-15-2	22456	49,0%	3600	12196	115	12998	15873	2,0%	3611

Tabela 6.4: Comparação entre Formulação Original, Relaxação Linear e Teste 2. A coluna (*Código*) refere-se a instância trabalhada, os itens (*RL*) e (*GC*) significam Relaxação Linear e Geração de Colunas, respectivamente. Já as siglas (*FO*), (*LI*), (*LS*), (*T*) e (*GAP*) representam a Função Objetivo, valor do Limitante Inferior, valor do Limitante Superior, Tempo de processamento em segundos e o GAP de otimalidade do método ( $\frac{LS - LI}{LS}$ ) respectivamente. Foi apresentada a solução ótima para relaxação linear.

Código	Modelo original			RL		Teste 2			
	FO	GAP	T(seg)	FO	T(seg)	LI	LS	GAP	T(seg)
5-2-3-1	34	0,0%	0	28	0	34	34	0,00%	0
5-3-3-1	28	0,0%	0	26	0	28	28	0,00%	0
5-4-3-1	50	0,0%	0	44	0	48	48	0,00%	3
10-3-5-1	61	0,0%	3	50	0	61	61	0,00%	6
10-5-5-1	113	0,0%	45	98	0	113	113	0,00%	15
10-7-5-1	157	0,0%	50	142	0	157	157	0,00%	12
15-5-8-1	149	0,0%	3600	133	0	149	149	0,00%	128
15-8-8-1	240	3,3%	3600	209	0	238	238	0,00%	663
15-11-8-1	347	10,6%	3600	293	0	342	342	0,00%	1143
20-6-10-1	190	0,0%	3350	169	0	190	190	0,00%	3600
20-10-10-1	390	20,1%	3600	383	3	483	483	0,03%	3611
20-14-10-1	481	15,4%	3600	477	1	566	566	0,06%	3607
50-15-15-1	1383	12,7%	3600	1207	1	1352	1352	0,00%	3607
50-25-15-1	3093	33,0%	3600	2071	7	2761	2781	0,71%	3600
50-35-15-1	4725	35,7%	3600	3037	8	4047	4049	0,04%	3610
5-2-3-2	226	0,0%	2	186	0	226	226	0,00%	2
5-3-3-2	436	0,0%	4	402	0	436	436	0,00%	2
5-4-3-2	327	0,0%	0	323	0	325	325	0,00%	1
10-3-5-2	361	6,0%	3600	320	1	361	361	0,00%	158
10-5-5-2	882	9,0%	3600	789	3	862	862	0,00%	943
10-7-5-2	1000	5,7%	3600	926	2	996	996	0,00%	3561
15-5-8-2	1705	17,2%	3600	1403	14	1682	1682	0,00%	3607
15-8-8-2	2055	28,9%	3600	1454	9	1977	1977	0,00%	3600
15-11-8-2	3907	24,3%	3600	2955	8	3603	3615	0,34%	3607
20-6-10-2	1859	24,6%	3600	1397	5	1575	1575	0,00%	3600
20-10-10-2	3642	22,2%	3600	2832	9	3279	3279	0,00%	3609
20-14-10-2	4866	27,8%	3600	3510	3	4077	4077	0,00%	3609
50-15-15-2	7950	24,5%	3600	6002	41	6816	6816	0,00%	3619
50-25-15-2	18699	47,8%	3600	9749	33	12458	12458	0,00%	3614
50-35-15-2	22456	49,0%	3600	12196	115	14762	14762	0,00%	3727

Tabela 6.5: Comparação entre Formulação Original, Relaxação Linear e Teste 3. A coluna (*Código*) refere-se a instância trabalhada, os itens (*RL*) e (*GC*) significam relaxação linear, geração de colunas e GAP de otimalidade do modelo original, respectivamente. Já as siglas (*FO*), (*T*) e (*GAP*) representam a função objetivo e o tempo de processamento em segundos, respectivamente. Foi apresentada a solução ótima para relaxação linear.

Código	Modelo Original			RL		Teste 3	
	FO	GAP	T(seg)	FO	T(seg)	FO	T(seg)
5-2-3-1	34	0,0%	0	28	0	34	0
5-3-3-1	28	0,0%	0	26	0	28	0
5-4-3-1	50	0,0%	0	44	0	52	0
10-3-5-1	61	0,0%	3	50	0	61	0
10-5-5-1	113	0,0%	45	98	0	116	0
10-7-5-1	157	0,0%	50	142	0	159	0
15-5-8-1	149	0,0%	3600	133	0	149	0
15-8-8-1	240	3,3%	3600	209	0	244	0
15-11-8-1	347	10,6%	3600	293	0	355	0
20-6-10-1	190	0,0%	3350	169	0	190	0
20-10-10-1	390	20,1%	3600	383	3	510	1
20-14-10-1	481	15,4%	3600	477	1	582	1
50-15-15-1	1383	12,7%	3600	1207	1	1391	1
50-25-15-1	3093	33,0%	3600	2071	7	2897	1
50-35-15-1	4725	35,7%	3600	3037	8	4333	1
5-2-3-2	226	0,0%	2	186	0	226	0
5-3-3-2	436	0,0%	4	402	0	436	0
5-4-3-2	327	0,0%	0	323	0	327	0
10-3-5-2	361	6,0%	3600	320	1	361	1
10-5-5-2	882	9,0%	3600	789	3	890	4
10-7-5-2	1000	5,7%	3600	926	2	1022	0
15-5-8-2	1705	17,2%	3600	1403	14	1682	1
15-8-8-2	2055	28,9%	3600	1454	9	2037	1
15-11-8-2	3907	24,3%	3600	2955	8	3900	5
20-6-10-2	1859	24,6%	3600	1397	5	1575	1
20-10-10-2	3642	22,2%	3600	2832	9	3374	1
20-14-10-2	4866	27,8%	3600	3510	3	4524	1
50-15-15-2	7950	24,5%	3600	6002	41	7001	2
50-25-15-2	18699	47,8%	3600	9749	33	12746	8
50-35-15-2	22456	49,0%	3600	12196	115	15580	6

Tabela 6.6: Comparação entre Formulação original, Testes 2 e Teste 3. A coluna (Código) refere-se a instância trabalhada. Já as siglas (FO), (LI), (LS), (T), (GAP) e VAR T2xT3 representam a função objetivo, valor do Limitante Inferior, valor do Limitante Superior, tempo de processamento em segundos, GAP de otimalidade ( $\frac{LS - LI}{LS}$ ) e VAR T2xT3 indica a diferença entre as soluções dos Testes 2 e Teste 3, respectivamente. Foi apresentada a solução ótima para relaxação linear.

Código	Modelo Original			Teste 2				Teste 3		
	FO	GAP	T (seg)	LI	LS	GAP	T (seg)	FO	T (seg)	VAR T2xT3
5-2-3-1	34	0,0%	0	34	34	0,00%	0	34	0	0,00%
5-3-3-1	28	0,0%	0	28	28	0,00%	0	28	0	0,00%
5-4-3-1	50	0,0%	0	48	48	0,00%	3	52	0	8,33%
10-3-5-1	61	0,0%	3	61	61	0,00%	6	61	0	0,00%
10-5-5-1	113	0,0%	45	113	113	0,00%	15	116	0	2,65%
10-7-5-1	157	0,0%	50	157	157	0,00%	12	159	0	1,27%
15-5-8-1	149	0,0%	3600	149	149	0,00%	128	149	0	0,00%
15-8-8-1	240	3,3%	3600	238	238	0,00%	663	244	0	2,52%
15-11-8-1	347	10,6%	3600	342	342	0,00%	1143	355	0	3,80%
20-6-10-1	190	0,0%	3350	190	190	0,00%	3600	190	0	0,00%
20-10-10-1	390	20,1%	3600	483	483	0,03%	3611	510	1	5,59%
20-14-10-1	481	15,4%	3600	566	566	0,06%	3607	582	1	2,83%
50-15-15-1	1383	12,7%	3600	1352	1352	0,00%	3607	1391	1	2,88%
50-25-15-1	3093	33,0%	3600	2761	2781	0,71%	3600	2897	1	4,17%
50-35-15-1	4725	35,7%	3600	4047	4049	0,04%	3610	4333	1	7,01%
5-2-3-2	226	0,0%	2	226	226	0,00%	2	226	0	0,00%
5-3-3-2	436	0,0%	4	436	436	0,00%	2	436	0	0,00%
5-4-3-2	327	0,0%	0	325	325	0,00%	1	327	0	0,62%
10-3-5-2	361	6,0%	3600	361	361	0,00%	158	361	1	0,00%
10-5-5-2	882	9,0%	3600	862	862	0,00%	943	890	4	3,25%
10-7-5-2	1000	5,7%	3600	996	996	0,00%	3561	1022	0	2,61%
15-5-8-2	1705	17,2%	3600	1682	1682	0,00%	3607	1682	1	0,00%
15-8-8-2	2055	28,9%	3600	1977	1977	0,00%	3600	2037	1	3,03%
15-11-8-2	3907	24,3%	3600	3603	3615	0,34%	3607	3900	5	7,88%
20-6-10-2	1859	24,6%	3600	1575	1575	0,00%	3600	1575	1	0,00%
20-10-10-2	3642	22,2%	3600	3279	3279	0,00%	3609	3374	1	2,90%
20-14-10-2	4866	27,8%	3600	4077	4077	0,00%	3609	4524	1	10,96%
50-15-15-2	7950	24,5%	3600	6816	6816	0,00%	3619	7001	2	2,71%
50-25-15-2	18699	47,8%	3600	12458	12458	0,00%	3614	12746	8	2,31%
50-35-15-2	22456	49,0%	3600	14762	14762	0,00%	3727	15580	6	5,54%

# Capítulo 7

## Conclusão e Trabalhos Futuros

Esse trabalho contribui com a aplicação dos métodos de relaxação Lagrangeana, Geração de Colunas e desenvolvimento de dois métodos heurísticos para a resolução do problema de sequenciamento denotado por  $F2|CD|\sum C_j^2$ . Esse problema é interpretado como um problema de *flowshop* com duas máquinas, com restrições de *crossdocking*, no qual a função objetivo busca minimizar a soma das datas de conclusão de todos os jobs que serão trabalhados pela máquina 2. Os métodos heurísticos buscam auxiliar a resolução da relaxação Lagrangeana e Geração de Colunas obtendo uma solução viável, que será utilizada como limite superior, e calculando boas sequências de processamento de jobs para a máquina 1, respectivamente.

Os resultados obtidos com os métodos se mostraram satisfatórios. A relaxação Lagrangeana, por meio da heurística Lagrangeana, obteve limites superiores próximos aos valores obtidos pelo modelo original em um curto intervalo de tempo, não ultrapassando 2 segundos para as maiores instâncias avaliadas. O método de geração de colunas associado a heurística também obteve resultados interessantes, pois o experimento denominado por Heurística 1 conseguiu obter um limite válido para 25 das 30 instâncias em um intervalo de tempo de até 3600 segundos. Destaca-se que o GAP das demais 5 instâncias não ultrapassou 1%. Além disso, o experimento obteve em 9 oportunidades o limite obtido igual ao valor da solução ótima do problema, considerando apenas as 11 instâncias que foram resolvidas pelo modelo original em até 3600 segundos, ou seja, GAP igual a zero. O tempo de processamento para esse experimento varia de acordo com a complexidade da instância, ou seja, quanto maiores o número de jobs processados, o tempo de processamento dos mesmos e o número de precedentes, maior será a complexidade e, conseqüentemente, o tempo de processamento para resolução

do problema.

Com relação a qualidade da sequência gerada pela heurística, observa-se pelo experimento Heurística 2 que a mesma auxilia a geração de colunas a encontrar boas colunas, acelerando o método. Esse fato pode ser comprovado ao comparar o experimento realizado sem a heurística, que obteve um limite válido em apenas 11 das 30 instâncias, contra 25 da Heurística 1. Além disso, ao comparar os valores dos experimentos Heurística 1 e 2, observa-se que os valores são semelhantes, apesar do SB ser resolvido de forma heurística por um dos experimentos. Os valores dos dois experimentos foram idênticos em 10 oportunidades e a maior diferença encontrada foi de 11%. Assim, pode-se avaliar como positiva a qualidade das sequências geradas, assim como a introdução da heurística no método de geração de colunas.

Comparando a relaxação Lagrangeana com a Heurística 1 por meio das tabelas 4.2 e 6.4, é observado que ambas possuem bons valores para os respectivos limites superiores, pois nos dois casos esses valores são próximos a solução ótima calculada pelo modelo original. Porém, ao avaliarmos os resultados relativos as maiores instâncias do grupo 1 e 2, percebe-se que a heurística 1 calcula um limitante superior, cujo valor está abaixo do valor do limite superior da relaxação Lagrangeana, este por sua vez, está acima ou muito próximo da solução do modelo original. No entanto, destaca-se que a solução fornecida pelo modelo original não é ótima e possui GAP elevado, acima de 20%, indicando que a mesma pode ser melhorada. Além disso, comparando os GAP's das soluções calculadas pelos dois métodos, é visível que ambos possuem dificuldade para convergirem. Por outro lado, a geração de colunas foi favorecida pela aplicação da heurística, resultando em resultados com GAP's nulos ou muito baixos, diferente daqueles obtidos pela relaxação Lagrangeana. Assim, o uso da heurística 1 pode trazer valores mais próximos da solução do problema.

Como trabalhos futuros, aponta-se a exploração do método de relaxação Lagrangeana, focando na melhora de seu desempenho. Assim, é proposto a aplicação de uma meta-heurística que calcule o limite superior de maneira alternativa ao que foi feita na seção 4.4. Uma possibilidade para esse fim é a heurística ILS (*Iterated – Local – Search*).

Com relação ao método de geração de colunas, é sugerido a introdução de outra heurística para trabalhar em conjunto com aquela descrita no trabalho. O objetivo é melhorar as sequências de jobs geradas para a máquina 1 e que são introduzidas no PMR. Uma opção é a heurística de busca local, que pode auxiliar nesse contexto.

## Referências Bibliográficas

- [1] *Cross Docking: How to use the EAN - UCC Standards.*
- [2] Savelsbergh M.W.P. Akker J.M., Hurkens C.A.J. Time-Indexed Formulations for Machine Scheduling Problems: Column Generation. *Inform Journal on Computing*, 12:111–124, 2000.
- [3] Viswanathan S. Apte. U.M. Effective cross docking for improving distribution efficiencies. *International Journal of Logistics: Research and Applications*, 3:291–302, 2000.
- [4] Gue K.R. Bartholdi JJ. The Best Shape for Crossdock. *Transportation Science*, 38 n.2:235–244, 2004.
- [5] Valckenaers P. Cattrysse D. Belle, J.V. Cross-Docking: State of the art. *Omega: The International Journal of Management Science*, 40:827–846, 2012.
- [6] Flidner M. Boysen, N. Cross Docking Sheduling: Classification, Literature Review and Research Agenda. *The International Journal of Management Science*, 38:413–422, 2010.
- [7] Carlo HJ. Bozer YA. Optimizing inbound and outbound door assignments in less-than-truckload crossdock. *IIE Transactions*, 40(11):1007–1018, 2008.
- [8] J.F. Campbell. A survey of network hub location. *Studies in Locational Analysis*, 6:31–39, 1994.
- [9] Tarantilis. C.D. Adaptive multi-restart Tabu Search algorithm for the vehicle routing problem with cross-docking. *Optimization Letters*, 7:1583–1596, 2013.
- [10] Lee C-Y. Chen, F. Minimizing the makespan in two-machine cross-docking flow shop problem. *European Journal of Operational Research*, 193:59–72, 2009.

- [11] Song K. Chen, F. Minimizing makespan in two-stage hybrid cross-docking scheduling problem. *Computers & Operations Research*, 36:2066–2073, 2009.
- [12] Santos-F. A. Mateus G. R. Cunha, A. S. A Branch-and-price algorithm for a Vehicle Routing Problem with Cross-Docking. *Electronic Notes in Discrete Mathematics*, 37:249–254, 2011.
- [13] Santos-F. A. Mateus G. R. Cunha, A. S. The Pickup and Delivery Problem with Cross-Docking. *Computers & Operations Research*, 40:1085–1093, 2013.
- [14] Kang K. Gue KR. Staging queues in material handling and transportation systems. In *Conference on winter simulation; 2001*. p. 1104–8.
- [15] BookBinder J.H. Gümüs, M. Cross-docking and its implications in Location-Distribution Systems. *Journal of Business Logistics*, 5:64–79, 2004.
- [16] Ross A. Jayaraman V. A simulated annealing methodology to distribution network design and management. *Computers & Industrial Engineering*, 55(1):629–645, 2008.
- [17] Drexl A. Klose, A. Facility location models for distribution system design. *European Journal of Operational Research*, 162:4–29, 2005.
- [18] Gue KR. The effects of trailer scheduling on the layout of freight terminals. *Transportation Science*, 33(4):419–428, 1999.
- [19] Lee KM. Lee YH., Jung JW. Vehicle routing scheduling for cross-docking distribution networks with setup costs and time window constraints. *Omega*, 39(1):64–72, 2011.
- [20] Rodrigues B. Li Y., Lim A. Crossdocking - JIT scheduling with time windows. *Journal of the Operational Research Society*, 55(12):1342–1351, 2004.
- [21] Shakeri M. Lim YG. Li ZP., Low MYH. Crossdocking planning and scheduling: problems and algorithms. Technical report, Singapore Institute of Manufacturing Technology, 2009.
- [22] Rodrigues B. Xu Z. Lim A., Miao Z. Transshipment through crossdock with inventory and time windows. *Naval Research Logistics*, 52(8):724–733, 2005.
- [23] Pinedo. M. *Scheduling: Theory, Algorithms and Systems*. Hall, Englewood Cliffs, 2001.

- [24] Geiger CD. McWilliams DL., Stanfield PM. The parcel hub scheduling problem: a simulation based solution approach. *Computers & Industrial Engineering*, 49(3):393–412, 2005.
- [25] Fei Q. Wang F. Miao Z., Fu K. Meta-heuristic algorithm for the transshipment problem with fixed transportation schedules. *New frontiers in applied artificial intelligence, Lecture Notes in Artificial Intelligence*, 2008.
- [26] V. Morais. Master's thesis.
- [27] Boysen N. Truck scheduling at zero-inventory cross docking terminals. *Computers & Operations Research*, 37(1):32–41, 2010.
- [28] Luh P.B.; and Hoiomt D.J. Scheduling of Manufacturing Systems Using the Lagrangean Relaxation Technique. *IEEE transactions*, 38(7):1066–1079, 1993.
- [29] Wein J. Philips C., Stein C. Minimizing Average Completion Time in the Presence of Release Dates. *Mathematical Programming*, 82:199–223, 1995.
- [30] Curi R. O problema de flow shop cross-docking com duas máquinas: Formulação indexada no tempo e relaxação lagrangeana. Trabalho de Graduação. Universidade Federal de Minas Gerais. 2011.
- [31] Johnson. S.M. Optimal two and three-stage production schedules with setup times included. *Naval Research Logistic Quarterly*, 1:61–68, 1954.
- [32] Song SH. Sung, CS. Integrated sevice network design for a cross-docking supply chain network. *Journal of the Operational Research Society*, 54(12):1283–1295, 2003.
- [33] Wolsey L.A. Vanderbeck F. Reformulation and decomposition of integer programs. In *Reformulation and Decomposition of Integer Programs, International Association for Research and Teaching. 2009*.
- [34] Clausen J. Cordeau JF. Laporte G. Wen M., Larsen J. Vehicle routing scheduling with cross-docking. *Journal of the Operational Research Society*, 60(12):1708–1718, 2009.

ANEXO A

Tabela 1: Resultados dos Testes Envolvendo o Modelo Original e a Relaxação Linear - Grupo 1. A coluna (*Código*) refere-se a instância trabalhada, os itens (*RL*) e (*GC*) significam Relaxação Linear e geração de colunas, respectivamente. Já as siglas (*FO*), (*T*) e (*GAP*) representam a Função Objetivo, Tempo de processamento em seg. e o GAP de otimalidade do método ( $\frac{LS - LI}{LS}$ ) respectivamente. Foi apresentada a solução ótima para relaxação linear. Os campos preenchido com (-) significam que nenhuma solução viável foi encontrada no intervalo de tempo de 3600 seg.

Código	Modelo original			RL	
	FO	GAP	T(seg)	FO	T(seg)
5-2-4-1	37	0%	9	30,69	0
5-3-2-1	28	0%	0	24,78	0
5-3-4-1	27	0%	0	24,40	0
5-4-2-1	48	0%	0	45,75	0
5-4-4-1	58	0%	1	49,86	0
10-3-7-1	74	0%	16	61,50	0
10-5-3-1	94	0%	90	82,91	0
10-5-7-1	105	0%	73	92,43	0
10-7-3-1	115	0%	4	106,72	0
10-7-7-1	188	0%	854	146,34	0
15-5-11-1	157	3%	3614	133,19	0
15-5-5-1	133	0%	466	117,44	0
15-11-11-1	395	19%	3613	304,04	0
15-11-5-1	248	0%	3150	220,22	0
15-8-11-1	242	4%	3602	211,75	0
15-8-5-1	220	0%	1815	207,00	0
20-6-14-1	195	2%	3607	175,40	0
20-6-6-1	190	5%	3600	170,30	0
20-10-14-1	493	28%	3620	337,27	0
20-10-6-1	352	10%	3604	310,88	0
20-14-14-1	651	30%	3602	451,80	3
20-14-6-1	572	7%	3602	521,00	3
50-15-25-1	1287	17%	3600	1062,00	6
50-15-35-1	1762	33%	3600	1179,7	10
50-25-25-1	3407	40%	3609	2022,00	8
50-25-35-1	-	-	3600	1900,00	11
50-35-25-1	-	-	3600	2661,00	26
50-35-35-1	4502	40%	3600	2671,00	16
60-18-18-1	1986	26%	3600	1457,00	10
60-18-30-1	2534	36%	3600	1611,00	20
60-18-42-1	2715	35%	3600	1763,00	13
60-30-18-1	3730	23%	3600	2859,00	12
60-30-30-1	3865	37%	3600	2411,00	14
60-30-42-1	5290	43%	3600	2996,00	73
60-42-18-1	7037	41%	3600	4141,00	21
60-42-30-1	7424	45%	3600	4071,00	53
60-42-42-1	7316	49%	3600	3665,00	29

Tabela 2: Resultados dos Testes Envolvendo o Modelo Original e a Relaxação Linear - Grupo 2. A coluna (*Código*) refere-se a instância trabalhada, os itens (*RL*) e (*GC*) significam Relaxação Linear e geração de colunas, respectivamente. Já as siglas (*FO*), (*T*) e (*GAP*) representam a Função Objetivo, Tempo de processamento em seg. e o GAP de otimalidade do método ( $\frac{LS - LI}{LS}$ ) respectivamente. Foi apresentada a solução ótima para relaxação linear. Os campos preenchido com (-) significam que nenhuma solução viável foi encontrada no intervalo de tempo de 3600 seg.

Código	Modelo original			RL	
	FO	GAP	T(seg)	FO	T(seg)
5-2-4-2	226	0%	89	185,95	0
5-3-2-2	286	0%	6	260,01	0
5-3-4-2	436	0%	28	401,60	0
5-4-2-2	540	0%	33	509,36	0
5-4-4-2	327	0%	1	323,20	0
10-3-7-2	431	8%	3604	354,10	1
10-5-3-2	608	0%	1740	544,18	4
10-5-7-2	999	20%	3602	788,76	6
10-7-3-2	1436	7%	3604	1314,60	6
10-7-7-2	1185	16%	3617	966,70	2
15-11-11-2	3700	20%	3603	2954,50	9
15-11-5-2	2971	19,5%	3604	2389,00	10
15-5-11-2	1947	23%	3600	1479,00	3
15-5-5-2	1226	11%	3606	1085,00	10
15-8-11-2	2600	43%	3606	1453,00	6
15-8-5-2	1312	12%	3606	1141,00	7
20-6-14-2	1724	14%	3629	1467,00	7
20-6-6-2	2055	25%	3603	1528,00	9
20-10-14-2	3955	27%	3604	2858,86	7
20-10-6-2	3963	27%	3621	2867,26	33
20-14-14-2	5802	39%	3606	3509,70	13
20-14-6-2	3534	10%	3615	3153,00	11
50-15-25-2	-	-	3600	10896,50	241
50-15-35-2	-	-	3600	10943,44	169
50-25-25-2	-	-	3600	19038,00	270
50-25-35-2	-	-	3600	15649,00	619
50-35-25-2	-	-	3600	24878,00	627
50-35-35-2	-	-	3600	24251,00	1710
60-18-18-2	-	-	3600	14448,6	368
60-18-30-2	-	-	3614	14503,50	900
60-18-42-2	-	-	3600	12628,08	556
60-30-18-2	-	-	3600	22300,00	692
60-30-30-2	-	-	3600	23726,16	1523
60-30-42-2	-	-	3600	26009,50	1108
60-42-18-2	-	-	3600	31549,70	1239
60-42-30-2	-	-	3600	32399,98	1576
60-42-42-2	-	-	3600	30315,00	2513

Tabela 3: Resultados dos Testes Envolvendo a Geração de Colunas - Grupo 1. A coluna (*Código*) refere-se a instância trabalhada, o item (*GC*) significa Geração de Colunas. Já as siglas (*LI*), (*LS*), (*T*) e (*GAP*) representam o valor do Limitante Inferior, valor do Limitante Superior, Tempo de processamento em segundos e o GAP de otimalidade do método ( $\frac{LS - LI}{LS}$ ) respectivamente.

Código	GC			T(seg)
	LI	LS	GAP	
5-2-4-1	37	37	0,00%	5
5-3-2-1	27	27	0,00%	9
5-3-4-1	26	26	0,00%	20
5-4-2-1	48	48	0,00%	29
5-4-4-1	54	54	0,00%	14
10-3-7-1	74	74	0,00%	99
10-5-3-1	94	94	0,00%	162
10-5-7-1	104	104	0,00%	503
10-7-3-1	111	111	0,00%	69
10-7-7-1	181	181	0,00%	1225
15-5-11-1	157	157	0,00%	1585
15-5-5-1	133	133	0,00%	3641
15-11-11-1	314	380	17,48%	3602
15-11-5-1	234	234	0,00%	3944
15-8-11-1	239	239	0,00%	3638
15-8-5-1	219	219	0,00%	3901
20-6-14-1	193	195	1,11%	3751
20-6-6-1	171	190	10,01%	3902
20-10-14-1	466	480	2,82%	3889
20-10-6-1	223	348	35,92%	3601
20-14-14-1	469	559	16,12%	3603
20-14-6-1	334	566	41,01%	3602
50-15-25-1	755	1264	40,29%	3603
50-15-35-1	1516	1721	11,91%	3602
50-25-25-1	2138	2665	19,79%	3603
50-25-35-1	2514	2815	10,70%	3602
50-35-25-1	3357	3740	10,24%	3605
50-35-35-1	3241	3728	13,07%	3602
60-18-18-1	1442	1871	22,92%	3602
60-18-30-1	1852	2141	13,51%	3601
60-18-42-1	1848	2326	20,53%	3601
60-30-18-1	2677	3451	22,43%	3601
60-30-30-1	2606	3302	21,08%	3602
60-30-42-1	4155	4521	8,09%	3601
60-42-18-1	4882	5654	13,65%	3605
60-42-30-1	5085	5832	12,81%	3602
60-42-42-1	5416	5683	4,69%	3602

Tabela 4: Resultados dos Testes Envolvendo a Geração de Colunas - Grupo 2. A coluna (*Código*) refere-se a instância trabalhada, o item (*GC*) significa Geração de Colunas. Já as siglas (*LI*), (*LS*), (*T*) e (*GAP*) representam o valor do Limitante Inferior, valor do Limitante Superior, Tempo de processamento em segundos e o GAP de otimalidade do método ( $\frac{LS - LI}{LS}$ ) respectivamente. O símbolo (-) significa que nenhum valor para o limite inferior do SB foi calculado.

Código	GC			
	LI	LS	GAP	T(seg)
5-2-4-2	226	226	0,00%	143
5-3-2-2	286	286	0,00%	96
5-3-4-2	436	436	0,00%	124
5-4-2-2	540	540	0,00%	139
5-4-4-2	325	325	0,00%	9
10-3-7-2	334	431	22,52%	4086
10-5-3-2	577	577	0,00%	1132
10-5-7-2	758	961	21,16%	3602
10-7-3-2	1210	1420	14,81%	3603
10-7-7-2	1084	1167	7,09%	3912
15-11-11-2	3271	3802	13,98%	3606
15-11-5-2	1867	2642	29,34%	3606
15-5-11-2	1623	1879	13,62%	3604
15-5-5-2	642	1218	47,31%	3603
15-8-11-2	1982	2090	5,16%	3603
15-8-5-2	606	1209	49,86%	3608
20-6-14-2	1046	1703	38,61%	3602
20-6-6-2	942	1709	44,89%	3602
20-10-14-2	3085	3717	17,01%	3604
20-10-6-2	3302	3637	9,22%	3607
20-14-14-2	4052	4585	11,62%	3620
20-14-6-2	1856	3296	43,70%	3608
50-15-25-2	19922	19963	0,21%	3627
50-15-35-2	20052	20105	0,27%	3638
50-25-25-2	32859	32859	0,00%	3670
50-25-35-2	28353	28365	0,04%	3645
50-35-25-2	42807	42826	0,05%	3728
50-35-35-2	43327	43344	0,04%	3683
60-18-18-2	-	26035	-	3619
60-18-30-2	-	24371	-	3624
60-18-42-2	-	21220	-	3620
60-30-18-2	-	40293	-	3643
60-30-30-2	-	41491	-	3642
60-30-42-2	-	47816	-	3667
60-42-18-2	-	53701	-	3710
60-42-30-2	-	57747	-	3632
60-42-42-2	-	55713	-	3630

Tabela 5: Resultados dos Testes Envolvendo os Teste 2 e 3 - Grupo 1. A coluna (*Código*) refere-se a instância trabalhada. As siglas (*FO*), (*LI*), (*LS*), (*T*) e (*GAP*) representam a Função Objetivo, valor do Limitante Inferior, valor do Limitante Superior, Tempo de processamento em segundos e o GAP de otimalidade do método( $\frac{LS - LI}{LS}$ ) respectivamente.

Código	Teste 2				Teste 3	
	LI	LS	GAP	T(seg)	FO	T(seg)
5-2-4-1	37	37	0%	7	37	0
5-3-2-1	27	27	0%	7	28	0
5-3-4-1	26	26	0%	1	27	0
5-4-2-1	48	48	0%	1	48	0
5-4-4-1	54	54	0%	1	59	0
10-3-7-1	74	74	0%	3	74	0
10-5-3-1	94	94	0%	11	95	0
10-5-7-1	104	104	0%	24	116	0
10-7-3-1	111	111	0%	1	115	0
10-7-7-1	181	181	0%	24	203	0
15-5-11-1	157	157	0%	71	157	0
15-5-5-1	133	133	0%	40	133	0
15-11-11-1	380	380	0%	1880	413	0
15-11-5-1	234	234	0%	574	259	0
15-8-11-1	239	239	0%	1347	243	0
15-8-5-1	219	219	0%	252	223	0
20-6-14-1	195	195	0%	328	195	0
20-6-6-1	190	190	0%	463	190	0
20-10-14-1	480	480	0%	3690	500	0
20-10-6-1	350	350	0%	3603	371	0
20-14-14-1	567	568	0%	3604	592	0
20-14-6-1	561	580	3%	3602	632	1
50-15-25-1	1244	1244	0%	3603	1274	0
50-15-35-1	1726	1726	0%	3607	1726	1
50-25-25-1	2604	2610	0%	3611	2772	1
50-25-35-1	2736	2741	0%	3625	2986	1
50-35-25-1	3724	3726	0%	3617	4234	6
50-35-35-1	3736	3736	0%	3611	4041	1
60-18-18-1	1870	1870	0%	3605	1923	1
60-18-30-1	2137	2137	0%	3606	2182	2
60-18-42-1	2315	2315	0%	3607	2428	1
60-30-18-1	3448	3448	0%	3633	3595	1
60-30-30-1	3305	3305	0%	3623	3444	1
60-30-42-1	4454	4454	0%	3623	4764	2
60-42-18-1	5500	5500	0%	3624	5717	5
60-42-30-1	5821	5821	0%	3631	6306	3
60-42-42-1	5574	5574	0%	3603	6101	3

Tabela 6: Resultados dos Testes Envolvendo os Testes 2 e 3 - Grupo 2. A coluna (*Código*) refere-se a instância trabalhada. As siglas (*FO*), (*LI*), (*LS*), (*T*) e (*GAP*) representam a Função Objetivo, valor do Limitante Inferior, valor do Limitante Superior, Tempo de processamento em segundos e o GAP de otimalidade do método( $\frac{LS - LI}{LS}$ ) respectivamente. O símbolo (-) significa que nenhum valor para o limite inferior do SB foi calculado.

Código	Teste 2				Teste 3	
	LI	LS	GAP	T(seg)	FO	T(seg)
5-2-4-2	226	226	0%	61	226	0
5-3-2-2	286	286	0%	15	286	1
5-3-4-2	436	436	0%	12	436	1
5-4-2-2	540	540	0%	540	540	0
5-4-4-2	325	325	0%	327	346	1
10-3-7-2	431	431	0%	1218	431	0
10-5-3-2	577	577	0%	26	630	1
10-5-7-2	961	961	0%	3610	1002	1
10-7-3-2	1426	1427	0%	3606	1462	1
10-7-7-2	1167	1167	0%	3781	1204	1
15-11-11-2	3601	3601	0%	3618	3806	1
15-11-5-2	2650	2650	0%	3615	2887	2
15-5-11-2	1863	1863	0%	3611	1863	0
15-5-5-2	1218	1218	0%	3606	1229	1
15-8-11-2	2083	2083	0%	3613	2226	1
15-8-5-2	1209	1209	0%	3703	1300	4
20-6-14-2	1692	1692	0%	3605	1692	1
20-6-6-2	1709	1709	0%	3616	1773	2
20-10-14-2	3639	3639	0%	3698	3702	1
20-10-6-2	3470	3470	0%	3692	3561	3
20-14-14-2	4509	4509	0%	3616	5048	4
20-14-6-2	3369	3428	2%	3630	3529	2
50-15-25-2	14467	14467	0%	3690	14738	8
50-15-35-2	16750	16750	0%	3692	16961	6
50-25-25-2	22188	22188	0%	3857	23003	14
50-25-35-2	22808	22808	0%	3608	24231	157
50-35-25-2	33716	33716	0%	4169	35694	70
50-35-35-2	36110	36110	0%	4102	41058	143
60-18-18-2	17369	17369	0%	3733	17934	7
60-18-30-2	17467	17467	0%	3690	17905	12
60-18-42-2	16556	16556	0%	3765	17124	9
60-30-18-2	-	28243	-	3781	41304	419
60-30-30-2	-	33575	-	3871	35745	184
60-30-42-2	-	39731	-	3890	41305	419
60-42-18-2	-	40566	-	3789	43822	569
60-42-30-2	-	49480	-	3799	54018	256
60-42-42-2	-	48174	-	3699	56760	625

Tabela 7: Comparação entre Modelo Original, Relaxação Linear e Relaxação Lagrangeana. A coluna (*Código*) refere-se a instância trabalhada, os itens (*RL*) e (*R.Lag*) significam Relaxação Linear e Relaxação Lagrangeana, respectivamente. Já as siglas (*FO*), (*LI*), (*LS*), (*T*) e (*GAP*) representam a função objetivo, limite inferior, limite superior, tempo de processamento em segundos e o GAP de otimalidade ( $\frac{LS - LI}{LS}$ ) respectivamente. Foi apresentada a solução ótima para relaxação linear.

Código	Modelo Original			RL		R. LAG			
	FO	GAP	T(seg)	FO	T(seg)	LI	LS	GAP	T(seg)
5-2-4-1	37	0%	9	30,69	0	30,69	37	17%	0
5-3-2-1	28	0%	0	24,78	0	24,78	28	12%	0
5-3-4-1	27	0%	0	24,40	0	24,40	27	10%	0
5-4-2-1	48	0%	0	45,75	0	45,75	48	5%	0
5-4-4-1	58	0%	1	49,86	0	49,86	58	14%	0
10-3-7-1	74	0%	16	61,50	0	61,50	74	17%	0
10-5-3-1	94	0%	90	82,91	0	82,91	94	12%	0
10-5-7-1	105	0%	73	92,43	0	92,43	105	12%	0
10-7-3-1	115	0%	4	106,72	0	106,72	116	8%	0
10-7-7-1	188	0%	854	146,34	0	146,34	197	26%	0
15-5-11-1	157	3%	3614	133,19	0	133,19	167	20%	0
15-5-5-1	133	0%	466	117,44	0	117,44	133	12%	0
15-11-11-1	395	19%	3613	304,04	0	304,04	416	27%	0
15-11-5-1	248	0%	3150	220,22	0	220,22	283	22%	0
15-8-11-1	242	4%	3602	211,75	0	211,75	260	19%	0
15-8-5-1	220	0%	1815	207,00	0	207,00	221	6%	0
20-6-14-1	195	2%	3607	175,40	0	175,40	195	10%	0
20-6-6-1	190	5%	3600	170,30	0	170,30	190	10%	0
20-10-14-1	493	28%	3620	337,27	0	337,27	544	38%	0
20-10-6-1	352	10%	3604	310,88	0	310,88	368	16%	0
20-14-14-1	651	30%	3602	451,80	3	451,80	647	30%	0
20-14-6-1	572	7%	3602	521,00	3	521,00	626	17%	0
50-15-25-1	1287	17%	3600	1062,00	6	1062,00	1558	32%	0
50-15-35-1	1762	33%	3600	1179,7	10	1179,70	1917	38%	0
50-25-25-1	3407	40%	3609	2022,00	8	2022,00	3064	34%	0
50-25-35-1	-	-	3600	1900,00	11	1900,00	3314	43%	0
50-35-25-1	-	-	3600	2661,00	26	2661,00	4671	43%	0
50-35-35-1	4502	40%	3600	2671,00	16	2671,00	4537	41%	0
60-18-18-1	1986	26%	3600	1457,00	10	1457,00	2196	34%	0
60-18-30-1	2534	36%	3600	1611,00	20	1611,00	2588	38%	0
60-18-42-1	2715	35%	3600	1763,00	13	1763,00	2760	36%	0
60-30-18-1	3730	23%	3600	2859,00	12	2859,00	4130	31%	0
60-30-30-1	3865	37%	3600	2411,00	14	2411,00	3857	37%	0
60-30-42-1	5290	43%	3600	2996,00	73	2996,00	5342	44%	1
60-42-18-1	7037	41%	3600	4141,00	21	4141,00	6708	38%	1
60-42-30-1	7424	45%	3600	4071,00	53	4071,00	6984	42%	1
60-42-42-1	7316	49%	3600	3665,00	29	3665,00	6946	47%	1

Tabela 8: Comparação entre Modelo Original, Relaxação Linear e Relaxação Lagrangeana. A coluna (*Código*) refere-se a instância trabalhada, os itens (*RL*) e (*R.Lag*) significam Relaxação Linear e Relaxação Lagrangeana, respectivamente. Já as siglas (*FO*), (*LI*), (*LS*), (*T*) e (*GAP*) representam a função objetivo, limite inferior, limite superior, tempo de processamento em segundos e o GAP de otimalidade ( $\frac{LS - LI}{LS}$ ) respectivamente. Foi apresentada a solução ótima para relaxação linear.

Código	Modelo Original			RL		R. LAG			
	FO	GAP	T(seg)	FO	T(seg)	LI	LS	GAP	T(seg)
5-2-4-2	226	0%	89	185,95	0	185,95	226	18%	0
5-3-2-2	286	0%	6	260,01	0	260,01	286	9%	0
5-3-4-2	436	0%	28	401,60	0	401,60	436	8%	0
5-4-2-2	540	0%	33	509,36	0	509,36	540	6%	0
5-4-4-2	327	0%	1	323,20	0	323,20	327	1%	0
10-3-7-2	431	8%	3604	354,10	1	354,10	458	23%	0
10-5-3-2	608	0%	1740	544,18	4	544,18	617	12%	0
10-5-7-2	999	20%	3602	788,76	6	788,76	995	21%	0
10-7-3-2	1436	7%	3604	1314,60	6	1314,60	1500	12%	0
10-7-7-2	1185	16%	3617	966,70	2	966,70	1204	20%	0
15-11-11-2	3700	20%	3603	2954,50	9	2954,50	3895	24%	0
15-11-5-2	2971	19,5%	3604	2389,00	10	2389,00	2976	20%	0
15-5-11-2	1947	23%	3600	1479,00	3	1479,00	1980	25%	0
15-5-5-2	1226	11%	3606	1085,00	10	1085,00	1226	12%	0
15-8-11-2	2600	43%	3606	1453,00	6	1453,00	2356	38%	0
15-8-5-2	1312	12%	3606	1141,00	7	1141,00	1474	23%	0
20-6-14-2	1724	14%	3629	1467,00	7	1467,00	1692	13%	0
20-6-6-2	2055	25%	3603	1528,00	9	1528,00	1804	15%	0
20-10-14-2	3955	27%	3604	2858,86	7	2858,86	3951	28%	0
20-10-6-2	3963	27%	3621	2867,26	33	2867,26	3770	24%	0
20-14-14-2	5802	39%	3606	3509,70	13	3509,70	5324	34%	0
20-14-6-2	3534	10%	3615	3153,00	11	3153,00	3551	11%	0
50-15-25-2	-	-	3600	10896,50	241	10896,50	16834	35%	1
50-15-35-2	-	-	3600	10943,44	169	10943,44	18708	42%	2
50-25-25-2	-	-	3600	19038,00	270	19038,00	26000	27%	2
50-25-35-2	-	-	3600	15649,00	619	15649,00	26645	41%	2
50-35-25-2	-	-	3600	24878,00	627	24878,00	39269	37%	3
50-35-35-2	-	-	3600	24251,00	1710	24251,00	44306	45%	4
60-18-18-2	-	-	3600	14448,6	368	14448,60	20691	30%	1
60-18-30-2	-	-	3614	14503,50	900	14503,50	21396	32%	2
60-18-42-2	-	-	3600	12628,08	556	12628,08	19507	35%	2
60-30-18-2	-	-	3600	22300,00	692	22300,00	34721	36%	3
60-30-30-2	-	-	3600	23726,16	1523	23726,16	39835	40%	3
60-30-42-2	-	-	3600	26009,50	1108	26009,50	46036	44%	5
60-42-18-2	-	-	3600	31549,70	1239	31549,70	50360	37%	4
60-42-30-2	-	-	3600	32399,98	1576	32399,98	61095	47%	6
60-42-42-2	-	-	3600	30315,00	2513	30315,00	59063	49%	7