

UNIVERSIDADE FEDERAL DE MINAS GERAIS

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO / ESCOLA DE ENGENHARIA

**Modelos e Algoritmos de Otimização para
Agrupamento de Pedidos e Sequenciamento de
Painéis em Lingotamento Contínuo**

Aloísio de Castro Gomes Júnior

Orientador(UFMG): Prof. Dr. Maurício Cardoso de Souza

Co-Orientador(UFMG): Prof. Dr. Martín Gómez Ravetti

Belo Horizonte

Agosto, 2015

UNIVERSIDADE FEDERAL DE MINAS GERAIS

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO / ESCOLA DE ENGENHARIA

Modelos e Algoritmos de Otimização para Agrupamento de Pedidos e Sequenciamento de Painéis em Lingotamento Contínuo

Aloísio de Castro Gomes Júnior

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais para a obtenção do título de Doutor em Engenharia de Produção.

Área de Concentração: Pesquisa Operacional e Engenharia de Manufatura

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

Orientador(UFMG): Prof. Dr. Maurício Cardoso de Souza

Co-Orientador(UFMG): Prof. Dr. Martín Gómez Ravetti

Belo Horizonte

Agosto, 2015

G633m

Gomes Júnior, Aloísio de Castro.

Modelos e algoritmos de otimização para agrupamentos de pedidos e sequenciamento de painéis em lingotamento contínuo [manuscrito] / Aloísio de Castro Gomes Júnior. – 2015.
xv, 137 f., enc.: il.

Orientador: Maurício Cardoso de Souza.

Coorientador: Martín Gómez Ravetti.

Tese (doutorado) - Universidade Federal de Minas Gerais,
Escola de Engenharia.

Anexos: f.131-137.

Inclui bibliografia.

1. Engenharia de produção - Teses. 2. Aço - Fundição - Teses. 3. Fundição contínua - Teses. I. Souza, Maurício Cardoso de. II. Gómez Ravetti, Martín. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 658.5(043)



FOLHA DE APROVAÇÃO

Modelos e Algoritmos de Otimização para Agrupamento de Pedidos e Sequenciamento de Painelas em Lingotamento Contínuo

ALOISIO DE CASTRO GOMES JUNIOR

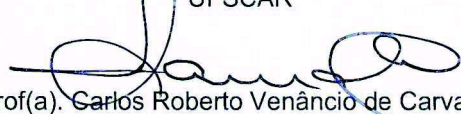
Tese submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA DE PRODUÇÃO, como requisito para obtenção do grau de Doutor em ENGENHARIA DE PRODUÇÃO, área de concentração PESQUISA OPERACIONAL E ENGENHARIA DE MANUFATURA, linha de pesquisa Modelos e

Aprovada em 07 de agosto de 2015, pela banca constituída pelos membros:

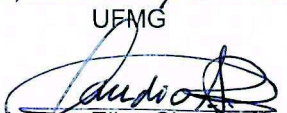

Prof(a). Mauricio Cardoso de Souza - Orientador
UFMG


Prof(a). Martin Gomez Ravetti
UFMG


Prof(a). Reinaldo Morabito Neto
UFSCAR


Prof(a). Carlos Roberto Venâncio de Carvalho
UFMG


Prof(a). Geraldo Robson Mateus
UFMG


Prof(a). Jose Elias Claudio Arroyo
UFV

Belo Horizonte, 7 de agosto de 2015.



ATA DA DEFESA DE TESE DO ALUNO ALOISIO DE CASTRO GOMES JUNIOR

Realizou-se, no dia 07 de agosto de 2015, às 10:00 horas, a definir, da Universidade Federal de Minas Gerais, a 12ª defesa de tese, intitulada *Modelos e Algoritmos de Otimização para Agrupamento de Pedidos e Sequenciamento de Panelas em Lingotamento Contínuo*, apresentada por ALOISIO DE CASTRO GOMES JUNIOR, número de registro 2009751935, graduado no curso de ENGENHARIA DE PRODUÇÃO, como requisito parcial para a obtenção do grau de Doutor em ENGENHARIA DE PRODUÇÃO, à seguinte Comissão Examinadora: Prof(a). Mauricio Cardoso de Souza - Orientador (UFMG), Prof(a). Martin Gomez Ravetti (UFMG), Prof(a). Reinaldo Morabito Neto (UFSCAR), Prof(a). Carlos Roberto Venâncio de Carvalho (UFMG), Prof(a). Geraldo Robson Mateus (UFMG), Prof(a). Jose Elias Claudio Arroyo (UFV).

A Comissão considerou a tese:

Aprovada

Reprovada

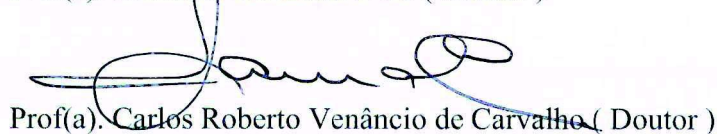
Finalizados os trabalhos, lavrei a presente ata que, lida e aprovada, vai assinada por mim e pelos membros da Comissão.

Belo Horizonte, 07 de agosto de 2015.


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


Prof(a). Martin Gomez Ravetti (Doutor)


Prof(a). Reinaldo Morabito Neto (Doutor)


Prof(a). Carlos Roberto Venâncio de Carvalho (Doutor)


Prof(a). Geraldo Robson Mateus (Doutor)


Prof(a). Jose Elias Claudio Arroyo (Doutor)

“À minha filha, eternamente em meu coração, e à minha esposa por estar sempre ao meu lado.”

Agradecimentos

Gostaria de agradecer, primeiramente, à Deus pelo dom da vida.

À UFMG pela oportunidade.

Aos professores Maurício e Martin pela orientação e pelas preciosas dicas, imprescindíveis para a realização deste trabalho.

À minha esposa Kívia pelo incentivo, confiança, compreensão e carinho.

À minha família pelo apoio e incentivos constantes.

Ao Allan pela preciosa ajuda com a obtenção das informações junto à ESB.

À ESB pela bolsa de auxílio e pelos dados fornecidos.

Aos meus amigos, colegas da faculdade, professores do DEP, colegas do Unileste e à todos que de uma forma ou de outra contribuíram para a realização deste trabalho.

Aos membros da banca examinadora, Prof. Dr. Reinaldo Morabito, Prof. Dr. Geraldo Robson Mateus e Prof. Dr. José Elias Cláudio Arroyo e Prof. Dr. Carlos Roberto Venâncio de Carvalho, por aceitarem participar da banca e pelas contribuições para a melhoria do texto final.

Resumo

A aciaria é o setor responsável pela transformação do ferro-gusa (oriundo do alto-forno) em aço e é constituída de três etapas: refino primário, refino secundário e lingotamento contínuo (responsável pela solidificação do aço líquido e sua transformação em placas). O aço líquido é transportado de uma etapa para a outra por painéis de grande capacidade. Um dos principais problemas encontrados na aciaria consiste na definição dos pedidos que irão compor uma panela e na programação destas painéis na máquina de lingotamento contínuo, de forma a minimizar os custos de produção da empresa. Entretanto, observou-se, na literatura pesquisada, que há pouquíssimos trabalhos que tratam o problema de agrupamento de pedidos e o problema de sequenciamento de painéis de forma integrada. A maioria procura tratar os problemas separadamente. Além disso, poucos trabalhos lidam com a formação de placas de misturas versus operações de *setup*, um dos principais problemas encontrados na maioria das siderúrgicas. Por isto, neste trabalho, propõe-se a utilização de modelos de programação linear inteira mista e heurísticas para resolver de forma eficiente os problemas de sequenciamento de painéis e agrupamento de pedidos em uma aciaria, separadamente e de forma integrada. Os modelos e heurísticas foram implementados em linguagem AMPL e resolvidos com o auxílio do pacote de otimização CPLEX. Para testar a eficiência dos modelos e das heurísticas, foram gerados dois conjuntos de instâncias-teste baseadas em relatórios gerenciais fornecidos por uma empresa siderúrgica brasileira. Os resultados se mostraram satisfatórios, viabilizando e encorajando a aplicação destes métodos em cenários reais.

Palavras-Chave: aciaria; lingotamento contínuo; sequenciamento de painéis; agrupamento de pedidos.

Abstract

The melt shop is the sector responsible for the transformation of pig iron (from blast furnace) in steel and consists of three stages: primary refining, secondary refining and continuous casting (responsible for liquid steel solidification and its transformation into plates). The liquid steel is transported from one stage to the other by large capacity ladles. One of the main problems encountered in the melt shop is the definition of orders that will compose a ladle and in the programming of these ladles in the caster, in order to minimize the costs of production of this company. However, it was observed, in literature researched, there are few papers that deal with the batching orders problem and sequencing charges (ladles) problem in an integrated manner. Most existing papers seeks to deal with the problems separately. In addition, few papers deal with the formation of intermix slabs versus perform a setup operation, one of the main problems found in most steel plants. So, in this thesis, it was proposed the use of mixed integer linear programming models and heuristics to solve efficiently the sequencing charges and batching orders problem in a steel plant, separately and in an integrated manner. The models and heuristics were implemented in language AMPL and solved with the optimization package CPLEX. To test the efficiency of the models and heuristics, it was created two sets of instances based on reports provided by a Brazilian steel plant. The results were satisfactory, enabling and encouraging the application of these methods in real-world scenarios.

Keywords: melt shop; continuous casting; sequencing charges; batching orders.

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
Lista de Siglas	xiv
1 Introdução	1
1.1 Objetivos e Contribuições	2
1.2 Estrutura do Trabalho	3
2 Contexto Siderúrgico Estudado	5
3 Revisão Bibliográfica	12
3.1 Problema de Sequenciamento da Produção	12
3.1.1 Problema de Sequenciamento de Máquina Única	13
3.1.2 Problema de Sequenciamento em Máquinas Paralelas	14
3.2 Problema Integrado de Dimensionamento e Sequenciamento de Lotes .	15
3.3 Problema de Sequenciamento na Aciaria	18
3.3.1 Decisões de Agrupamento de Corridas e Séries	19
3.3.2 Decisões de Sequenciamento	22
3.3.3 Conclusões	28
4 Problema de Sequenciamento de Painelas	29

4.1	Modelos Matemáticos para o Problema de Sequenciamento de Painelas	30
4.1.1	Modelo M	31
4.1.2	Modelo W	33
4.1.3	Modelo I	35
4.1.4	Modelo P	37
4.2	Instâncias e Resultados	39
4.2.1	Geração das Instâncias Baseadas no Caso Real	39
4.2.2	Resultados e Análise	42
4.3	Heurísticas para o PSP	49
4.3.1	Heurística de Esau-Williams para o PSP - HEW	52
4.3.2	Heurística de Esau-Williams com Múltiplos Reinícios	54
4.3.2.1	Determinação dos Parâmetros da HEWMR	57
4.3.2.2	Resultados e Análise - HEWMR	59
4.4	Conclusões	62
5	Problema de Agrupamento de Pedidos	63
5.1	Problema de Agrupamento de Painelas	63
5.1.1	Modelo PED	65
5.1.2	Modelo PAD-N	67
5.1.3	Modelo PAD-Bk	67
5.1.4	Modelo PAD-Bd	69
5.1.5	Modelo PAD-B2	70
5.2	Heurística RF-FOPt para o PAP	71
5.3	Resultados e Análise	75
5.3.1	Geração das Instâncias	75
5.3.2	Resultados - Modelos para o PAP	77
5.3.3	Resultados - Heurísticas RF e FOPt	86
5.4	Conclusões	90

6 Problema Integrado de Agrupamento de Pedidos e Sequenciamento de Painelas	92
6.1 Modelos Matemáticos	93
6.2 Heurísticas para o PIAPSP	98
6.3 Resultados e Análise	102
6.3.1 Resultados - Modelos para o PIAPSP	102
6.3.2 Resultados - Heurísticas para o PIAPSP	105
6.4 Conclusões	118
7 Conclusões	121
Referências Bibliográficas	124
A Resultados da Heurística RF-FOpt para o PAP	131

Lista de Figuras

2.1	Fluxo de Material em uma Usina Siderúrgica Integrada. Fonte: Tang e Wang [54]	6
2.2	Processo de Produção na Aciaria	6
2.3	Processo de Lingotamento Contínuo. Fonte: Bretas [8]. (1) panela contendo o aço líquido; (2) distribuidor; (3) moldes; (4) rolos	8
2.4	Possíveis ações quando duas panelas são processadas consecutivamente: (a) Geração de Placa de Mistura - sem interrupção no processo; (b) Operação de <i>setup</i> - interrupção do processo para realizar o <i>setup</i>	9
2.5	Processo de Elaboração do Sequenciamento na ESB	10
4.1	Perda de material por mudança na largura das placas	41
4.2	Junção de <i>jobs</i> pode levar à uma solução sub-ótima	43
4.3	Diagrama de Gantt da Instância IRM42	46
4.4	Procedimento para a Escolha da Maior Economia	56
4.5	Heurística EW com múltiplos reinícios para o PSP	56
4.6	Gráfico para determinação do valor de α	58
4.7	Gráfico para determinação do número máximo de iterações (<i>iterMAX</i>)	59
5.1	Pseudo-código da heurística RF-FOpt para resolução do PAP	74
6.1	Pseudo-código da heurística agrupa primeiro e sequencia depois (HAPSD)100	100
6.2	Pseudo-código da heurística PIRF-FOpt para resolução do PIAPSP	102

Lista de Tabelas

4.1	Tempo de processamento para cada tipo de aço	41
4.2	Resultados das instâncias reais com o modelo M	44
4.3	Resultados das instâncias reais com o modelo M fixando o número de <i>setups</i>	45
4.4	Resultados das instâncias reais com o modelo W	47
4.5	Resultados das instâncias reais menores com o modelo P	48
4.6	Resultados das instâncias reais maiores com o modelo P	50
4.7	Comparativo do Número de Variáveis e do Número de Restrições dos Modelos M, W e P	50
4.8	Resultados das instâncias reais menores com a Heurística EW	54
4.9	Resultados das instâncias reais maiores com a Heurística EW	55
4.10	Tabela auxiliar usada para geração de instâncias artificiais	57
4.11	Resultados da resolução da HEWMR para as instâncias reais	60
4.12	Resultados da resolução da HEWMR para as instâncias reais	61
5.1	Expressões para os cálculos dos custos	76
5.2	Resultados das instâncias artificiais com o modelo PED	78
5.3	Resultados das instâncias reais com o modelo PED	79
5.4	Resultados das instâncias artificiais com o modelo PAD-N	80
5.5	Resultados das instâncias reais com o modelo PAD-N	81
5.6	Resultados das instâncias artificiais com o modelo PAD-Bk	82

5.7	Resultados das instâncias reais com o modelo PAD-Bk	83
5.8	Resultados das instâncias artificiais com o modelo PAD-Bd	84
5.9	Resultados das instâncias reais com o modelo PAD-Bd	85
5.10	Resultados das instâncias artificiais com o modelo PAD-B2	86
5.11	Resultados das instâncias reais com o modelo PAD-B2	87
5.12	Resumo dos resultados obtidos com a aplicação dos modelos PED, PAD-N, PAD-Bk, PAD-Bd e Pad-B2 à resolução do PAP - Instâncias Artificiais (IAP001 a IAP024)	87
5.13	Resumo dos resultados obtidos com a aplicação dos modelos PED, PAD-N, PAD-Bk, PAD-Bd e Pad-B2 à resolução do PAP - Instâncias Reais(IAP025 a IAP056)	88
5.14	Resumo dos Resultados da Heurística RF-FOpt	89
5.15	Resultados da Heurística RF-FOpt usando as combinações de estratégias RF-zt1b/FOpt-ztpf e RF-zp1f/FOpt-ztpf	90
6.1	Resultados para as instâncias Artificiais com o modelo PIPED	103
6.2	Resultados para as instâncias Reais com o modelo PIPED	104
6.3	Resultados para as instâncias Artificiais com o modelo PIPAD-Bk	105
6.4	Resultados para as instâncias Artificiais com a heurística HAPSD	107
6.5	Resultados para as instâncias Reais com a heurística HAPSD	108
6.6	Resultados para as instâncias Artificiais com a heurística PIRF-FOpt - estratégia PIRF-xt1f	109
6.7	Resultados para as instâncias Artificiais com a heurística PIRF-FOpt - estratégia PIRF-xt2f	110
6.8	Resultados para as instâncias Artificiais com a heurística PIRF-FOpt - estratégia PIRF-xt1b	111
6.9	Resultados para as instâncias Artificiais com a heurística PIRF-FOpt - estratégia PIRF-xt2b	112
6.10	Resultados para as instâncias Reais com a heurística PIRF-FOpt - estratégia PIRF-xt1f	113

6.11	Resultados para as instâncias Reais com a heurística PIRF-FOpt - estratégia PIRF-xt2f	114
6.12	Resultados para as instâncias Reais com a heurística PIRF-FOpt - estratégia PIRF-xt1b	115
6.13	Resultados para as instâncias Reais com a heurística PIRF-FOpt - estratégia PIRF-xt2b	117
6.14	Melhores resultados para as instâncias testadas	118
6.15	Comparativo entre os modelos PIPED e PIPAD-Bk	119
A.1	Resultados da Heurística usando a estratégia RF-zt1f	132
A.2	Resultados da Heurística usando a estratégia RF-zt2f	133
A.3	Resultados da Heurística usando a estratégia RF-zt1b	134
A.4	Resultados da Heurística usando a estratégia RF- τ t1f	135
A.5	Resultados da Heurística usando a estratégia RF- τ t1f	136
A.6	Resultados da Heurística usando a estratégia RF-zp1f	137

Lista de Siglas

ACO: *Ant Colony Optimization* - Colônia de Formigas

ATSP: *Asymmetric Travelling Salesman Problem* - Problema do caixeiro viajante assimétrico

CSMT: *Capacited Minimum Spanning Tree* - árvore geradora mínima capacitada

EAF: *Electric Arc Furnace* - Forno Elétrico a Arco

FOpt: *fix-and-optimize* - fixa e otimiza

GVNS: *General Variable Neighborhood Search* - Busca em vizinhança variável geral

HEW: Heurística de Esau-Williams

HCW: Heurística de Clarke & Wright

GRASP: *Greedy Randomized Adaptive Search Procedure* - Procedimento de busca adaptativa gulosa e randomizada

LC: lingotamento contínuo

LF: *ladle furnace* - Forno Panela

MLC: Máquina de Lingotamento Contínuo

PAP: Problema de Agrupamento de Pedidos

PGDSL: Problema geral de dimensionamento e sequenciamento de lotes

PDSLPP: Problema de dimensionamento de sequenciamento de lotes com linhas em paralelo

PIAPSP: Problema Integrado de Agrupamento de Pedidos e Sequenciamento de Painelas

PSA: Problema de Sequenciamento na Aciaria

PSP: Problema de Sequenciamento de Painéis

PLIM: Programação Linear Inteira Mista

PMP: Plano Mestre de Produção

PRVAC: Problema de Roteamento de Veículos Assimétrico Capacitado

PSMP: Problema de Sequenciamento de Máquinas Paralelas

PSMU: Problema de Sequenciamento de Máquina Única

RF: *relax-and-fix* - relaxa e fixa

VNDS: Variable Neighborhood Decomposition Search - Busca de decomposição de vizinhança variável

Capítulo 1

Introdução

A indústria de ferro e aço é uma indústria básica importante para qualquer setor da economia industrial, fornecendo matérias-primas para diversas outras indústrias, tais como, indústria de construção civil, indústria automobilística, indústria de utensílios domésticos, entre outras. Segundo o site *www.pwc.com.br*, em 2012, o Brasil era o 9º produtor de aço no mundo, sendo Minas Gerais, o principal produtor de aço no país.

Pode-se dividir o processo de produção de aço em três grandes etapas: produção do ferro-gusa (alto-fornos), produção do aço (aciaria) e laminação.

A aciaria é o setor responsável pela transformação do ferro-gusa obtido da redução do minério de ferro nos alto-fornos em aço e pela posterior solidificação do mesmo e segundo diversos autores é o gargalo no processo de produção de aço (Tang e Wang [54], Bellabdaoui e Teghem [7], Missbauer et al. [36], Li *et al.* [29], Tang *et al.* [50], Tang *et al.* [51]). A transformação de ferro-gusa em aço ocorre por meio de refinamentos primário e secundário, que têm a finalidade de garantir ao aço as características físicas e químicas desejadas. Já a solidificação e a transformação em placas é feita através de um processo de lingotamento contínuo.

É neste cenário que se enquadra o problema abordado neste trabalho. O problema consiste em determinar o melhor agrupamento dos pedidos em painéis e o melhor sequenciamento destas painéis no setor de lingotamento contínuo de forma que os custos de produção da empresa estudada sejam minimizados. Considerou-se no trabalho apenas o setor de lingotamento contínuo, pois este setor é o que mais agrega valor ao produto final e é um processo que não pode ser interrompido, em virtude dos altos custos e tempos de *setup*. O processo de lingotamento contínuo é considerado

crítico para a aciaria, pois ele é sucedido de processos não-contínuos e é este processo quem dita o ritmo da aciaria, sendo que as datas de início e término das atividades nos estágios anteriores são calculadas à partir das datas de início das atividades no processo de lingotamento contínuo.

Quando dois aços de composições químicas diferentes são sequenciados um imediatamente após o outro no lingotamento gera-se a chamada placa de mistura. Em determinadas situações, estas placas só podem ser aproveitadas como sucata, o que implica em perdas para a empresa. Em outras situações, esta mistura é aproveitada como sub-produto e vendida por um preço menor. Por outro lado, para que não ocorra a geração desta placa, pode-se fazer o *setup*, preparando o equipamento para receber o novo tipo de aço. O problema, neste caso, é que o tempo de *setup* é muito alto e implica em custos com troca de equipamentos. Este tempo depende apenas da máquina e não da sequência de produção. Desta forma, o problema aqui estudado traz uma particularidade interessante: o *trade-off* gerar placa de mistura \times fazer *setup*.

Conforme afirma Tang *et al.* [51], excetuando o trabalho realizado por eles, não há outros que tratem do problema de agrupamento de pedidos em painéis e a formação de séries de painéis de forma integrada. Daí a importância do trabalho aqui proposto, pois também procura tratar estes dois problemas de forma integrada.

Além disso, em uma pesquisa feita em quatro siderúrgicas do estado de Minas Gerais, verificou-se que nenhuma delas possuía um *software* capaz de gerar e avaliar soluções para o problema aqui estudado, o que justifica o desenvolvimento dos modelos e dos métodos aqui propostos. Observou-se, também, através desta pesquisa, que a maioria das empresas preferem não fazer a mistura, desta forma não gerando a placa de mistura e preferindo fazer, na maioria das vezes, mais operações de *setup*. Esta situação acontece, principalmente, quando não consegue-se processar várias painéis com a mesma composição química sem interrupção.

1.1 Objetivos e Contribuições

O principal objetivo deste trabalho é desenvolver modelos matemáticos e algoritmos capazes de resolver de forma eficiente o problema integrado de agrupamento de pedidos e sequenciamento de painéis no setor de lingotamento contínuo de uma empresa siderúrgica brasileira, minimizando os custos de estoque, atraso, *setup* e geração de placas de mistura.

Para alcançar este objetivo, alguns objetivos secundários devem ser alcançados: (i) realizar um estudo bibliográfico que aponte as principais tendências na resolução do problema em estudo; (ii) formular modelos matemáticos que contemplem as principais características do problema estudado e (iii) desenvolver algoritmos que possam resolver de forma eficiente o problema em estudo.

Como contribuições acadêmicas, pode-se citar o trabalho de Gomes Júnior *et al.* [21], que apresenta os modelos de programação linear inteira mista para o problema de agrupamento de pedidos e para o problema integrado de agrupamento de pedidos e sequenciamento de painéis. Outra contribuição é o trabalho de De Souza *et al.* [11], que apresenta dois modelos de programação linear inteira mista para o problema de sequenciamento de painéis no lingotamento contínuo. Além destes trabalhos, foram feitas duas apresentações internacionais sobre os modelos de programação linear inteira mista para o problema de sequenciamento de painéis (De Souza *et al.* [13] e De Souza *et al.* [12]).

1.2 Estrutura do Trabalho

A estrutura do trabalho é definida da forma como se segue.

Inicialmente, no Capítulo 2 é apresentado o contexto siderúrgico estudado e suas principais características.

No Capítulo 3 são apresentados os principais trabalhos que tratam de assuntos relacionados ao problema estudado.

Por sua vez, o Capítulo 4 apresenta os quatro modelos e os algoritmos desenvolvidos para resolver o problema de sequenciamento de painéis no processo de lingotamento contínuo. Ainda no Capítulo 4, é apresentado o primeiro grupo de instâncias-teste gerado e os resultados da aplicação dos modelos e dos algoritmos.

Já no Capítulo 5 são apresentados os modelos e os algoritmos para a resolução do problema de agrupamento de pedidos em painéis. Apresenta-se, também, nesse capítulo como o segundo grupo de instâncias-teste foi gerado e os resultados da aplicação dos modelos e das heurísticas à estas instâncias.

O Capítulo 6 apresenta os modelos matemáticos e os algoritmos para a resolução do problema integrado de agrupamento de pedidos e sequenciamento de painéis. Além disso, o Capítulo 6 apresenta os resultados da aplicação dos modelos matemáticos

e algoritmos ao problema integrado, utilizando as instâncias apresentadas no Capítulo 5.

O Capítulo 7 conclui o trabalho e apresenta perspectivas para trabalhos futuros.

Capítulo 2

Contexto Siderúrgico Estudado

A Empresa Siderúrgica Brasileira (ESB), assim chamada para manter confidencialidade sobre as suas informações, é uma usina integrada, pois adquire a matéria-prima bruta e executa todas as fases do processo industrial. Seu processo é voltado para produção de aços planos em formas de placas ou bobinas.

A usina siderúrgica integrada é um típico processo, onde os produtos finais são produzidos através de uma sequência de etapas, ou seja, produção do ferro, produção do aço, laminação a quente e laminação a frio. A Figura 2.1 ilustra um fluxo de materiais genéricos para uma usina siderúrgica integrada. No estágio de produção do ferro, minério de ferro, coque e outras matérias primas são convertidas em ferro-gusa no alto-forno. No estágio de produção do aço, denominado Aciaria, o ferro-gusa é transformado em placas sólidas de aço, passando por três processos: refino primário (transformação do ferro-gusa em aço), refino secundário (acerto da composição química do aço) e lingotamento contínuo. No estágio de laminação à quente, as placas são laminadas e transformadas em bobinas em um laminador à quente. No setor de laminação à frio, bobinas de tiras à quente são transformadas em bobinas à frio em laminadores à frio, e são finalizadas através de diferentes linhas de produção, tais como, recozimento contínuo, galvanização contínua, eletro galvanização, entre outras. (TANG e WANG, [54])

A ESB possui duas aciarias em diferentes ambientes, mas este trabalho foca-se em apenas uma delas. A aciaria é o setor responsável pela transformação do ferro-gusa em aço e pela posterior solidificação do mesmo. O processo de produção da aciaria da ESB é apresentado na Figura 2.2.

Na aciaria analisada existem dois convertedores LD (nome dado em home-

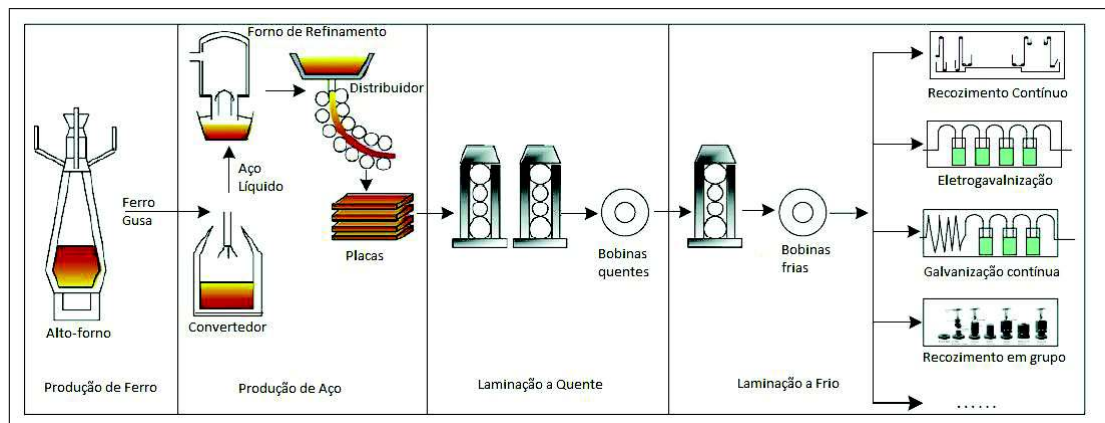


Figura 2.1: Fluxo de Material em uma Usina Siderúrgica Integrada. Fonte: Tang e Wang [54]

nagem às cidades onde o processo foi desenvolvido - Linz e Donawitz) para realizar o refino primário. Esta etapa do processo tem como finalidade transformar o gusa líquido em aço, através do sopro de oxigênio sobre a sua superfície por meio de uma lança refrigerada.

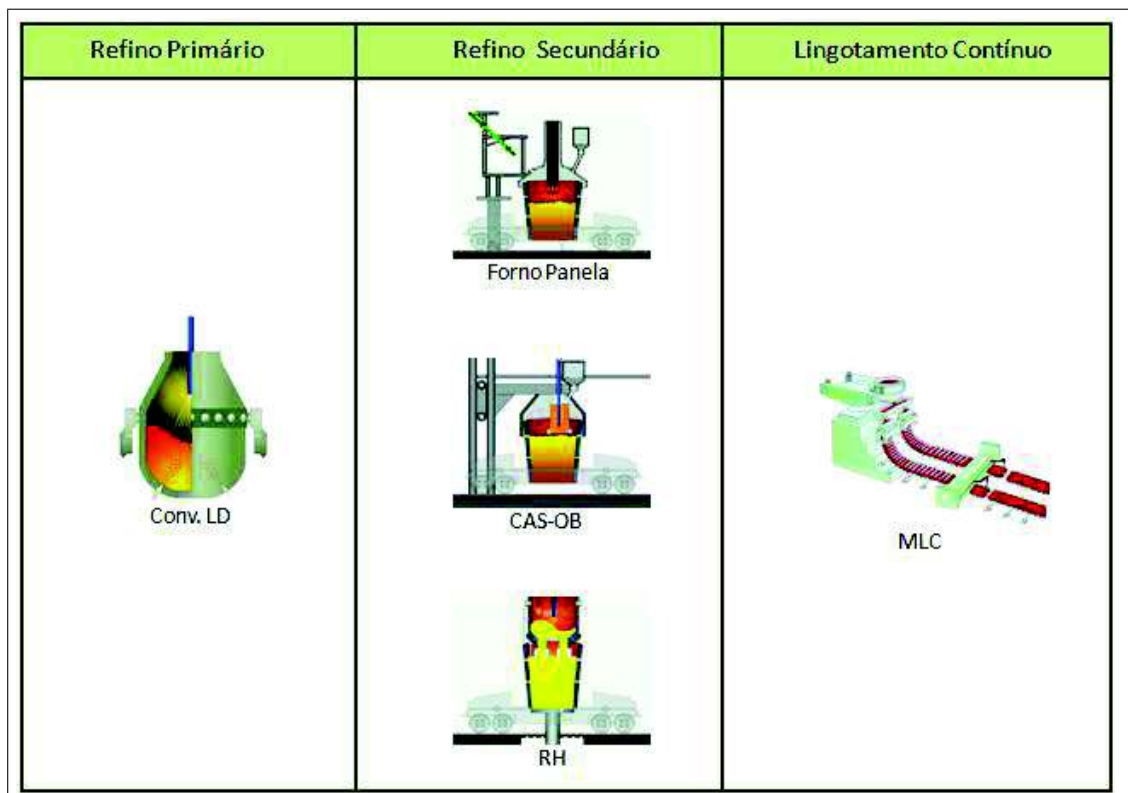


Figura 2.2: Processo de Produção na Aciaria

Segundo Rizzo [42], compõem a carga do LD, além do ferro gusa: gases industriais, carga metálica (sucata, ferro gusa sólido, etc.), fundentes e/ou refrigerantes,

ferro-ligas e ligas nobres, desoxidantes, entre outros. Segundo este mesmo autor, a proporção entre os diversos tipos de carga metálica, depende de uma série de fatores, como por exemplo, disponibilidade e preço das matérias-primas.

Assim que o refino no convertedor LD é finalizado, o aço é então vazado em panelas de grande capacidade. Em seguida, estas panelas são encaminhadas para o refino secundário. Na aciaria da ESB, o refino secundário pode ser feito em três tipos de equipamentos: CAS-OB (processo de aquecimento químico), Forno Panela (processo de aquecimento através de energia elétrica) e RH (processo de desgaseificação à vácuo).

A ESB possui dois equipamentos do tipo CAS-OB (*Composition Adjustment by Sealed Argon Bubbling and Oxygen Blowing*). Segundo Rizzo [43], o CAS-OB é um processo de aquecimento químico e é utilizado principalmente para ajustes de composição química, temperatura e melhoria dos aços. Já o forno panela é um processo de aquecimento através de energia elétrica e tem as mesmas finalidades do processo CAS-OB.

Os processos de aquecimento por reações químicas e os processos de aquecimento através de energia elétrica (forno panela) são tecnologias ou processos considerados concorrentes na etapa de refino secundário. A escolha de uma tecnologia ou outra depende de uma série de fatores como custo da energia elétrica e dos desoxidantes, tipo de instalação do refino primário, entre outros. (Rizzo, [43])

Outro processo usado para o refino secundário dos aços é o processo de desgaseificação à vácuo. A ESB trabalha especificamente com o processo denominado RH (Ruhrstahl e Heraeus). Basicamente, o processo RH é caracterizado pela circulação contínua de aço através de um recipiente revestido de refratários, no qual promove-se o vácuo. (Rizzo, [43])

A definição de qual equipamento será usado no refino secundário depende das características de cada tipo de aço. Inclusive, alguns tipos de aço necessitam de um duplo refino, então, primeiro eles passam pelo forno panela e em seguida pelo RH.

Após o refino secundário, a panela contendo o aço líquido é então encaminhada ao processo de lingotamento contínuo. Nesse processo, o aço é vazado por meio de uma válvula colocada no fundo da panela, que proporcionará seu escoamento até o distribuidor. O distribuidor, por sua vez, proporciona o vazamento do aço até o molde, parte constituinte da máquina de lingotamento contínuo (MLC). No lingotamento, o aço líquido é solidificado através de um sistema de refrigeração. Ao fim do processo, o aço em barras é cortado, dando origem às placas, produto final da aciaria. A Figura

2.3 ilustra o processo de lingotamento contínuo.

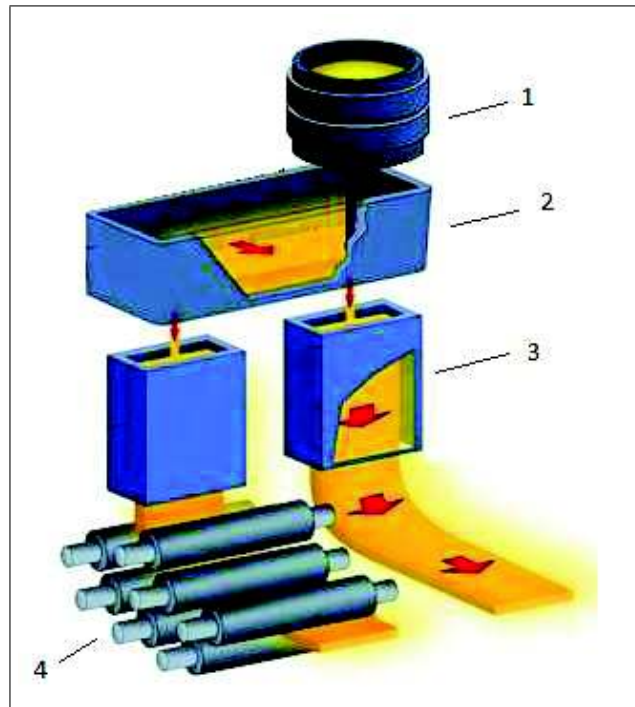


Figura 2.3: Processo de Lingotamento Contínuo. Fonte: Bretas [8]. (1) panela contendo o aço líquido; (2) distribuidor; (3) moldes; (4) rolos

No processo de lingotamento contínuo, entre o vazamento de duas panelas consecutivas de dois tipos de aços diferentes, duas situações podem ocorrer: (i) os dois tipos de aço são misturados gerando-se a chamada placa de mistura e (ii) a máquina de lingotamento tem seu processo interrompido para troca do material refratário do distribuidor (*setup*), estas duas situações são descritas na Figura 2.4.

Quando há geração da placa de mistura, a placa é vendida como sub-produto por um preço bem inferior ao original. Quando é necessário fazer o *setup*, há um custo associado ao material refratário que reveste o distribuidor da máquina de lingotamento contínuo. Como o material refratário tem uma vida útil curta, obrigatoriamente deve haver uma troca deste revestimento antes que a sua vida útil chegue ao fim. Outro ponto importante é que dois aços com larguras muito diferentes não podem ser vazados um após o outro sem haver uma troca do distribuidor. Neste caso, existe uma variação máxima de largura permitida entre o vazamento de dois aços consecutivos, se a variação da largura entre os dois aços for menor que a variação máxima permitida, os dois aços podem ser misturados, caso contrário há necessidade de se fazer o *setup*. Quando os aços são misturados e há variação na largura entre estes é gerada uma rebarba, que é cortada e aproveitada como sucata.

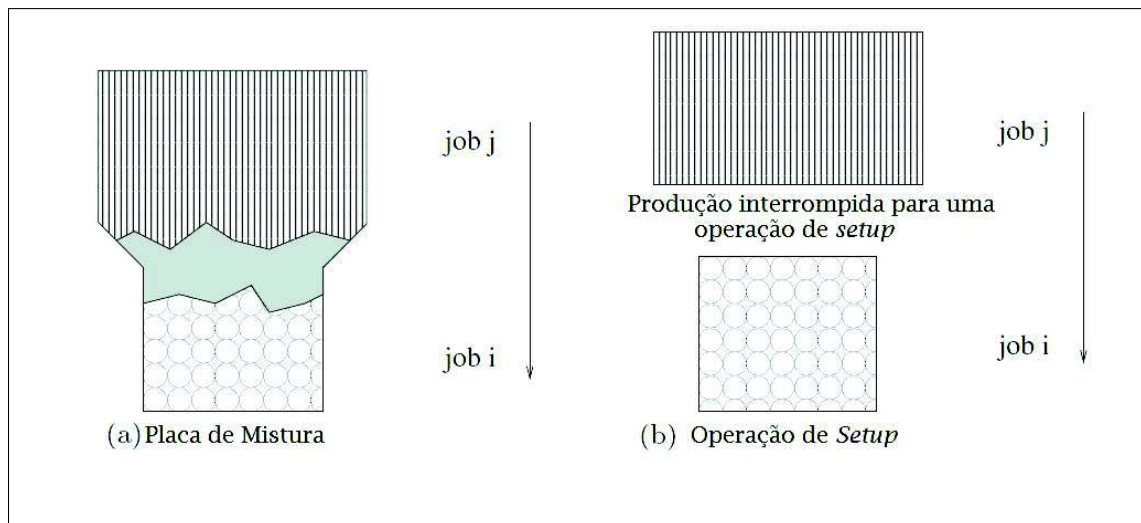


Figura 2.4: Possíveis ações quando duas placas são processadas consecutivamente: (a) Geração de Placa de Mistura - sem interrupção no processo; (b) Operação de *setup* - interrupção do processo para realizar o *setup*

Em média, na ESB, são realizadas 24 corridas por dia na máquina de lingotamento contínuo. Uma corrida corresponde à uma panela vazada para a produção de placas de aço. Todos os equipamentos exigem um tempo de preparação de máquina (*setup*) antes de iniciar o processo, que acontece antes de cada corrida, com exceção da máquina de lingotamento, que deve receber o aço ininterruptamente, panela após panela. Caso demore mais do que certo tempo máximo entre o término do vazamento de uma panela e o início do vazamento de outra, o processo deverá ser interrompido e o *setup* da máquina deverá ser realizado antes de reiniciar o lingotamento. Como um dos artifícios para evitar essa parada, ocorre a redução da velocidade da máquina de lingotamento enquanto a troca da panela acontece.

Dentro deste contexto, um dos problemas encontrados pela ESB é definir a melhor sequência de produção na aciaria, principalmente no setor de lingotamento contínuo. Atualmente, a programação da produção na aciaria é feita manualmente com auxílio de um *software*. O *software* utilizado permite ao programador escolher a máquina, inserir as corridas e montar a série da forma como quiser. Os programadores buscam montar as séries contendo a maior quantidade de corridas possível, evitando misturas de determinados tipos de aço. Mas o *software* utilizado pela ESB não possui funcionalidades de comparação ou avaliação da programação proposta.

Atualmente, o processo de elaboração do sequenciamento da ESB é realizado conforme a Figura 2.5. Primeiramente, o setor de vendas recebe os pedidos dos clientes. Os pedidos contêm a quantidade de aço desejada pelo cliente, a composição química

desejada e a largura das placas ao término do processo de lingotamento contínuo. Estes pedidos são transformados em ordens de venda e encaminhados ao setor de aceite. Somente são aceitos pedidos que possam formar panelas. Os pedidos que não puderem formar panelas são colocados em uma cesta de pedidos e aguardam outros pedidos para que conjuntamente possam formar uma panela. Os pedidos aceitos são encaminhados ao setor de programação da aciaria, que com o auxílio de um *software* determinam a sequência de produção. O *software* apenas auxilia na determinação da sequência, permitindo que o operador selecione e aloque as panelas em uma determinada sequência. O *software* permite ao operador visualizar a sequência gerada, mas não é capaz de avaliar a qualidade desta sequência. Após a sequência ser aprovada, a mesma é enviada ao responsável pelo sequenciamento na aciaria que verifica se é possível executar esta sequência. Caso positivo, a sequência é realizada e, em caso negativo, a mesma é reenviada ao setor de programação da aciaria.

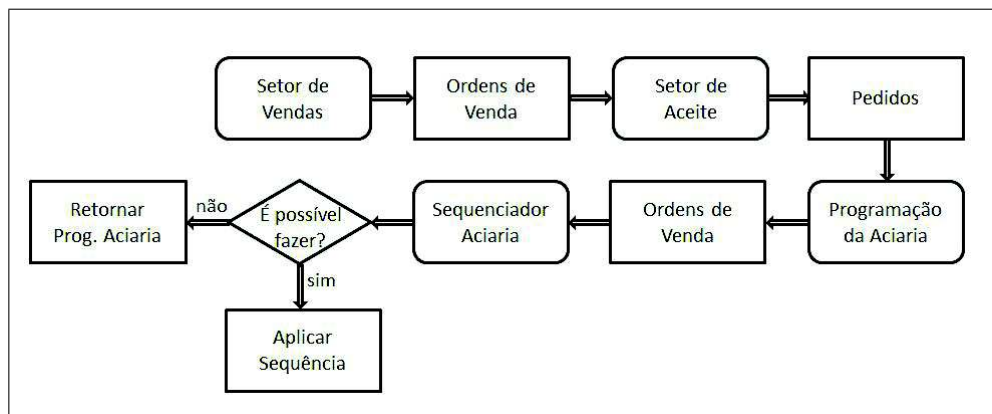


Figura 2.5: Processo de Elaboração do Sequenciamento na ESB

A cada dez dias o setor de Planejamento e Controle da Produção da ESB recebe do setor de aceite as ordens que devem ser produzidas com prazos definidos por quinquídios. Entretanto, o horizonte programado e enviado para a aciaria corresponde a 24 horas de produção aproximadamente.

As panelas a serem produzidas são determinadas priorizando o prazo de entrega e então forma-se a sequência de produção.

Resumindo-se, neste trabalho pretende-se, inicialmente, determinar os pedidos que irão compor uma panela e em que período do horizonte de planejamento estas panelas serão processadas. Neste primeiro problema, objetiva-se minimizar os custos de atraso na entrega dos pedidos, custos de manutenção de estoque e os custos de agrupamento dos pedidos em uma mesma panela. Somente pedidos com composições

químicas semelhantes e larguras próximas deverão ser agrupados em uma mesma panela. Após a geração das panelas, deve-se decidir, dentro de cada período, qual a sequência de processamento das panelas no setor de lingotamento contínuo. Busca-se, aqui, minimizar os custos de geração de placas de mistura e os custos de realização de operações de *setup*. A principal restrição operacional deste segundo problema é o desgaste do material refratário do distribuidor da MLC após o processamento de algumas panelas, por isto, deve-se respeitar um intervalo de tempo máximo entre dois *setups* consecutivos. Os problemas descritos anteriormente são resolvidos, primeiramente, de forma separada e logo em seguida de forma conjunta.

Capítulo 3

Revisão Bibliográfica

3.1 Problema de Sequenciamento da Produção

Os problemas de sequenciamento da produção consistem em designar um conjunto de operações a recursos e determinar a sequência de processamento destas operações em cada recurso, bem como o instante de início e término do processamento de cada operação. Cada tipo de problema possui uma característica que o diferencia dos demais. Várias modelagens foram propostas para estes problemas e vários métodos foram criados para resolvê-los.

Os problemas de sequenciamento de produção podem ser classificados de acordo com o ambiente operacional em: (i) máquina única; (ii) máquinas paralelas; (iii) *flow-shop* e (iv) *jobshop*. Para mais detalhes sobre estes tipos de problema ver Pinedo [41] e Baker [5].

Os problemas podem, ainda, ser classificados de acordo com o número de etapas de produção em: mono-estágio e multi-estágio. O problema é considerado mono-estágio quando o processo de produção é realizado em uma única etapa ou quando, no processo de produção, existe um recurso que é considerado o gargalo do sistema e o sequenciamento é feito baseado neste recurso. O problema é considerado multi-estágio quando o processo de produção é constituído de duas ou mais etapas, onde em cada etapa tem-se um certo número m de recursos disponíveis.

Problemas mono-estágio podem ser encontrados em alguns cenários, tais como, indústria de fundição de materiais metálicos ou plásticos, produção de tijolos refratários (ver Mateus *et al.* [33]) e indústria siderúrgica (processo de lingotamento contínuo, pro-

cesso de trefilação, entre outros). Já os problemas multi-estágio podem ser encontrados em cenários que o produto final é montado a partir de componentes.

3.1.1 Problema de Sequenciamento de Máquina Única

No problema de sequenciamento de máquina única (PSMU), todos os produtos são processados em uma única máquina especializada. Segundo Pinedo [41], na prática, problemas de sequenciamento em ambientes de produção mais complexos são frequentemente decompostos em subproblemas que lidam com máquinas únicas. O autor cita, como exemplo, um ambiente de produção complexo com um único gargalo que pode dar origem a um modelo de máquina única. Desta forma, além dos ambientes produtivos que operam com uma única máquina/recurso, ambientes produtivos onde as estações de trabalho são dependentes uma das outras ou ambientes produtivos que possuem um único gargalo podem ser representados como um PSMU.

Os critérios de otimização normalmente adotados para o PSMU são: minimização do tempo necessário para processar todos os *jobs* (*makespan*); minimização do atraso total; minimização do tempo de fluxo (*lead time*) médio; minimização dos custos com antecipação e atraso da produção; entre outros.

Um exemplo de aplicação real do PSMU é apresentado por Bustamante [9]. Neste trabalho, o autor resolve o problema de sequenciamento de ordens em um processo de laminação a quente em uma siderúrgica brasileira como um PSMU com penalidades por antecipação e atraso. Bustamante [9] considera, em seu trabalho, uma linha de laminadores em sequência como uma máquina única e busca minimizar o somatório das penalidades de antecipação e atraso da produção. Outro exemplo de aplicação real bem similar é apresentada por Jin *et al.* [24]. Jin *et al.* [24] resolvem um problema de sequenciamento em uma fábrica de fios de aço na China. Os autores caracterizam o problema como um PSMU e buscam minimizar o atraso máximo.

Keha *et al.* [26] apresentam e avaliam a performance computacional de quatro diferentes modelos de programação linear inteira mista para o PSMU. As variáveis de decisão que distinguem os quatro diferentes tipos de modelo no trabalho de Keha *et al.* [26] são: (i) variáveis de data de conclusão; (ii) variáveis indexadas no tempo; (iii) variáveis de ordenação linear e (iv) variáveis indexadas pela posição/alocação.

As quatro formulações anteriores foram testadas por Keha *et al.* [26] com instâncias teste geradas aleatoriamente e os autores chegaram às seguintes conclusões

sobre as quatro formulações: (a) a formulação (iv) encontra frequentemente a solução ótima, mas o limite inferior obtido pela relaxação linear é ruim; (b) as formulações (ii) e (iii) apresentaram os melhores limites inferiores, mas a resolução da relaxação linear era mais difícil para estas formulações; (c) as formulações (i) e (ii) não produziram soluções ótimas com tanta frequência quanto as formulações (iii) e (iv).

3.1.2 Problema de Sequenciamento em Máquinas Paralelas

Existem três tipos de ambientes de máquinas paralelas: idênticas, uniformes e não-relacionadas. Em máquinas idênticas, os *jobs* têm o mesmo tempo de processamento e o mesmo tempo de preparação. Em máquinas uniformes, o tempo de processamento de um *job* e o tempo de preparação de máquinas mais modernas são proporcionais aos tempos correspondentes às máquinas mais antigas. Em máquinas não-relacionadas, não existe uma relação entre os tempos de processamento e preparação de máquinas distintas. Um modelo básico deste tipo de problema considera que todos os *jobs* estão disponíveis para processamento na data 0 e uma vez iniciado o processamento de um *job*, o mesmo deve ser finalizado (Arenales *et al.* [3]).

Assim como no PSMU, os critérios de otimização frequentemente adotados para o problema de sequenciamento de máquinas paralelas (PSMP) são: minimização do *makespan* (tempo necessário para processar todos os *jobs*); minimização do atraso total; minimização do tempo de fluxo (*lead time*) médio; minimização dos custos com antecipação e atraso da produção; minimização do *makespan* acrescido do atraso médio ponderado (ver Paula *et al.* [40] e Rocha *et al.* [44]); entre outros.

Há vários trabalhos na literatura que tratam do PSMP. Alguns destes trabalhos são apresentados a seguir.

Rocha *et al.* [44] resolvem o problema de sequenciamento de máquinas paralelas não-relacionadas com tempos de *setup* dependente da máquina e da sequência. Estes autores apresentam dois modelos matemáticos baseados nos modelos de Manne [30] e Wagner [60]. O critério de otimização adotado é a minimização do atraso ponderado acrescido do *makespan*. Para resolver o problema, os autores desenvolvem um método *Branch-and-Bound* e usam uma solução fornecida pela meta-heurística GRASP como limitante superior. Os autores conseguem resolver problemas com até 30 *jobs*.

O mesmo problema é resolvido por Paula *et al.* [40]. Estes autores propõem um algoritmo *non-delayed relax-and-cut*, baseado na relaxação lagrangiana de uma

formulação indexada no tempo deste problema. Usando o método proposto, foi possível obter soluções ótimas com razoáveis tempos para instâncias com até 180 *jobs* e 6 máquinas.

Huang *et al.* [23] abordam o problema de sequenciamento de máquinas paralelas em que o tempo de *setup* depende da sequência e as operações de *setup* são realizadas por um único servidor. Estes autores tratam de um problema real encontrado na indústria de impressão. Para resolver o problema, os autores apresentam um modelo matemático para resolver instâncias com poucos *jobs* e poucas máquinas e um algoritmo genético híbrido para resolver instâncias com um número maior de *jobs*.

Kim *et al.* [27] tratam do problema de máquinas paralelas não-relacionadas com tempos de *setup* dependentes da sequência, com o objetivo de minimizar o atraso total priorizado. Os autores utilizam 4 heurísticas de buscas: data de entrega mais cedo priorizada; o menor tempo de processamento priorizado; heurística de sequenciamento de dois níveis e o método *Simulated Annealing*. As heurísticas são testadas com dados oriundos de um processo de corte em uma fábrica de semi-condutores compostos.

3.2 Problema Integrado de Dimensionamento e Sequenciamento de Lotes

O Plano Mestre de Produção (PMP) atua entre a previsão de demanda e as decisões de manufatura. Ele considera como entrada as previsões de venda e os dados aproximados de capacidade instalada com a finalidade de decidir quantos itens finais serão produzidos dentro do período especificado. Isto produz o problema de otimização de dimensionamento de lotes sobre um horizonte de planejamento de médio prazo dividido em períodos discretos de tempo. (Mateus *et al.* [33])

O problema de dimensionamento de lotes consiste em decidir o tamanho dos lotes de produção em cada período para minimizar custos de produção e de atendimento ao consumidor sujeito às restrições de capacidade. Custos de produção são incorridos em operações de manufatura - tais como preparação de máquinas e manuseio de material - e em manutenção de estoques entre períodos consecutivos - tais como capital amarrado e seguros. Custos de atendimento ao consumidor são incorridos quando há demandas insatisfeitas, e elas são modeladas como penalidades de custo dependendo do ambiente de negócios. (Mateus *et al.* [33])

A literatura do problema de dimensionamento de lotes lida com o tempo de duas maneiras distintas, *big* ou *small buckets*, dependendo do período de tempo. Modelos *big buckets* são caracterizados por grandes períodos de tempo, tipicamente semanas ou meses, permitindo a produção de vários produtos por período. Modelos *small buckets*, por outro lado, são caracterizados por períodos curtos de tempo, turnos ou dias, por exemplo, no qual é permitido, no máximo, um *setup* por período.

Uma vez que as decisões de tamanho do lote tenham sido tomadas, lotes de produção constituem ordens de produção (*jobs*) a serem processadas. Operações de sequenciamento assinalam *jobs* à máquinas a curto prazo.

Segundo Drexel e Kimms [14] o planejamento e sequenciamento da produção é um dos temas mais desafiadores para o gerenciamento de uma fábrica. A seguir, são apresentados alguns trabalhos encontrados na literatura que abordam o problema integrado de dimensionamento e sequenciamento de lotes.

Drexel e Kimms [14] fazem uma revisão extensa da literatura sobre o problema de dimensionamento e sequenciamento de lotes, apresentando diversos modelos tipo *small buckets* para este problema e suas extensões.

Lasserre [28] apresenta um modelo integrado para o dimensionamento e sequenciamento em um ambiente *Job-Shop*. Além do modelo, o autor apresenta um método que alterna entre a resolução de um problema de planejamento para uma sequência conhecida de produtos nas máquinas e a resolução de um problema padrão de *job-shop* para um plano de produção fixado.

Meyr e Mann [35] e Meyr [34] apresentam um modelo geral para o problema de dimensionamento e sequenciamento de lotes com linhas em paralelo (PDSLPP). Neste modelo, o horizonte de planejamento finito T consiste de macroperíodos discretos $t = 1, \dots, T$ (por exemplo, meses ou semanas) com um dado tamanho. Para cada linha de produção l , cada macroperíodo é adicionalmente representado por um conjunto S_{lt} de microperíodos sem sobreposição s , onde o número $|S_{lt}|$ de microperíodos é conhecido a priori. Somente um único produto pode ser produzido em um único microperíodo. Se em uma certa linha o mesmo item é produzido dentro de um ou vários microperíodos, estes microperíodos constituem um lote. A quantidade total produzida durante estes microperíodos definem o tamanho do lote.

Kawamura e Ronconi [25] apresentam um modelo análogo ao apresentado por Meyr e Mann [35] e Meyr [34], mas levam em consideração que pode haver atrasos na entrega e existe uma restrição de capacidade de armazenamento.

Supithak *et al.* [47] apresentam três heurísticas para resolver o problema de dimensionamento e sequenciamento de lotes com penalidades por antecipação e atraso da produção, sendo um destes baseado na meta-heurística Algoritmos Genéticos (AG). A heurística baseada no AG se mostrou superior as outras duas, apresentando pequenos desvios da solução ótima do problema.

Mateus *et al.* [33] propõem um modelo de programação linear inteira para abordar o problema de dimensionamento de lotes e sequenciamento de forma integrada. Estes autores também propõem um método iterativo que resolve o problema de dimensionamento de lote em sua otimalidade através de um modelo matemático e o problema de sequenciamento através da heurística GRASP.

Baldo *et al.* [6] apresentam um modelo de PLIM para resolver um problema de dimensionamento e sequenciamento de lotes em uma cervejaria. Além do modelo matemático, Baldo *et al.* [6] propõem heurísticas que procuram utilizar a estrutura do modelo de PLIM apresentado. As heurísticas desenvolvidas foram baseadas nas técnicas “*relax-and-fix*” e “*fix-and-optimize*” e aplicadas em instâncias-teste geradas artificialmente. Essas instâncias são baseadas em um caso real de uma cervejaria em Portugal.

Toledo *et al.* [56], Ferreira *et al.* [17], Ferreira *et al.* [18] e Toledo *et al.* [57] apresentam modelos matemáticos e/ou métodos heurísticos a resolução de um problema de dimensionamento de lotes e programação da produção aplicado a uma indústria de refrigerantes.

Kawamura e Ronconi [25] utilizam o modelo para o PDSLLP permitindo-se atrasos na entrega e restrições de capacidade de armazenagem para resolver o problema real de uma empresa de bens de higiene pessoal, tais como, fraldas, absorventes, lenços, entre outros. Os autores utilizam a heurística “*relax-and-fix*” com onze diferentes tipos de estratégias para fixar as variáveis.

Toso *et al.* [58] apresentam um modelo matemático que busca integrar o problema geral de dimensionamento e sequenciamento de lotes com o problema do caixeiro viajante assimétrico (ATSP - *Asymmetric Travelling Salesman Problem*) na resolução de um problema na indústria de nutrição animal. Clark *et al.* [10] também utilizam um modelo baseado no ATSP para resolver este mesmo problema. Neste trabalho, os autores utilizam um procedimento iterativo de eliminação de sub-rotas reforçados pela inclusão de um procedimento de correções de sub-rotas. Assim como Toso *et al.* [58], os autores utilizam instâncias-teste baseadas em um problema real de

uma indústria de nutrição animal.

Araújo *et al.* [2] resolvem um problema de dimensionamento e sequenciamento de lotes em uma fundição. Os autores apresentam um modelo de programação inteira mista para resolver o problema, mas ressaltam que não conseguem resolver o problema em sua otimalidade. Então os autores propõem alguns métodos heurísticos, como o *Simulated Annealing*, desta maneira conseguindo gerar boas soluções para o problema.

Seeanner *et al.* [45] apresentam um modelo de PLIM para o problema geral de dimensionamento e sequenciamento de lotes (PGDSL) em um ambiente de produção multi-estágio. O ambiente estudado pelos autores é constituído de dois estágios e encontra-se em uma empresa europeia de produção de folhas de plástico. Para resolver o problema, os autores propõem um algoritmo que combina as características da meta-heurística VNDS (*Variable Neighborhood Decomposition Search*) com a heurística “*fix-and-optimize*” baseada no modelo de PLIM. Utilizou-se instâncias-teste geradas artificialmente e instâncias-teste baseadas no problema real.

Figueira *et al.* [20] propõem um algoritmo que combina métodos heurísticos e métodos exatos em uma estrutura denominada GVNS (*General Variable Neighborhood Search*) para resolver o problema de planejamento e sequenciamento da produção. O trabalho é motivado por um estudo de caso real em indústria de papel e celulose. Também na indústria de papel e celulose, Shim *et al.* [46] propõem uma heurística de dois estágios para resolver um problema de dimensionamento e sequenciamento de lotes capacitado com máquina única e custo de *setup* dependente da sequência em uma empresa que produz papelões ondulados.

3.3 Problema de Sequenciamento na Aciaria

Como dito anteriormente, a aciaria é o setor responsável pela transformação do ferro-gusa em aço e sua posterior solidificação em forma de placas ou tarugos. O processo na aciaria é constituído de três estágios: refino primário (transformação do ferro-gusa em aço), refino secundário (acerto da composição química do aço) e lingotamento contínuo (solidificação do aço).

Vários autores (Tang e Wang [54], Bellabdaoui e Teghem [7], Missbauer *et al.* [36], Li *et al.* [29], Tang *et al.* [50], Tang *et al.* [51], Mao *et al.* [31]) ressaltam que a aciaria é reconhecida como o gargalo na produção do aço. Segundo Tang e Wang [54], tal fato se deve a capacidade de produção da aciaria ser geralmente menor que os

processos de laminação à quente e laminação à frio.

A maioria dos trabalhos encontrados na literatura utilizam as seguintes terminologias:

- **Corrida (*charge*):** corresponde a uma panela no processo de obtenção do aço;
- **Série (*cast*):** conjunto de corridas processadas sem interrupção.

Tang e Wang [54] afirmam que o problema de planejamento e programação na aciaria incluem dois tipos de decisões:

- (i) Decisões de agrupamento, isto é, transformar os pedidos dos clientes (ordens primárias de produção) em lotes de produção. Segundo os autores, há dois tipos de lote de produção: corridas e séries.
- (ii) Decisões de sequenciamento, ou seja, alocar e sequenciar as corridas e séries nas correspondentes facilidades, da produção do aço ao lingotamento contínuo.

Na Seção 3.3.1 são apresentados trabalhos que lidam com as decisões de agrupamento. Já na Seção 3.3.2, apresentam-se trabalhos que têm seu foco nas decisões de sequenciamento.

3.3.1 Decisões de Agrupamento de Corridas e Séries

Tang e Wang [54] focam seu trabalho nas decisões de agrupamento, que incluem problemas de dois níveis de agrupamento: (i) problema de agrupamento das corridas (*Charge Batching Problem*) e (ii) problema de agrupamento de séries (*Cast Batching Problem*). Segundo os autores estes problemas podem ser definidos da forma como se segue:

- **Problema de Agrupamento de Corridas:** uma corrida consiste de um único tipo de aço com uma composição química única e esta corrida pode produzir várias placas com tamanhos diversos. Estas placas podem pertencer a um único cliente ou a diversos clientes. O problema de agrupamento de corridas consiste em tomar decisões que respondam como consolidar placas dentro das corridas, ou seja, como alocar as placas oriundas dos diversos pedidos dentro de uma corrida até que a capacidade da panela seja alcançada.

- **Problema de Agrupamento de Séries:** uma série consiste de uma sequência de corridas, que são consecutivamente produzidas utilizando um mesmo distribuidor. O problema de agrupamento de séries consiste em tomar decisões que respondam como agrupar corridas dentro de uma série, satisfazendo as seguintes restrições: (i) agrupar tantas corridas quanto possível para formar uma série tal que o distribuidor possa ser eficientemente usado; (ii) somente corridas com propriedades metalúrgicas similares podem ser agrupadas em uma série. Procura-se desta forma maximizar a utilização dos distribuidores e minimizar o custo de mudanças de composição química.

Tang e Wang [54] propõem modelos de Programação Linear Inteira Mista e métodos heurísticos para resolver estes dois tipos de problemas de agrupamento. Os testes são feitos com instâncias reais de uma grande siderúrgica chinesa e após os testes, gerou-se um sistema integrado para resolver estes dois tipos de problemas. Os autores ressaltam que estes problemas eram resolvidos manualmente pela empresa estudada, processo que gastava, em média, 2 horas e com a implantação do sistema, este tempo caiu para menos de 5 minutos.

Tang *et al.* [55] também tratam dos problemas de agrupamento da aciaria, mais especificamente do problema do agrupamento de corridas. Tang *et al.* [55] criam um novo tipo de agrupamento: lote de corridas (*charge-lot*). Lotes de corridas são corridas consecutivas com a mesma composição química e a mesma largura das placas geradas ao final do processo de lingotamento contínuo. Segundo os autores, agrupar placas diretamente em lotes de corridas pode ser mais benéfico para a redução de resíduos. Além disso, usar lotes de corridas ao invés de corridas como uma unidade básica pode simplificar as decisões de agrupamento na aciaria e, conseqüentemente, reduzir o tamanho do problema de agrupamento. O foco do trabalho destes autores é especificar conjuntamente as placas necessárias para satisfazer cada ordem dos consumidores e agrupar estas placas de diferentes ordens em lotes de corridas.

Primeiramente, Tang *et al.* [55] apresentam um modelo de PLIM para determinar: (a) o número de placas de cada ordem i assinalada (designada) a cada um dos admissíveis lotes de corridas j ; (b) o peso das placas da ordem i assinalada ao lote de corridas j e (c) o número de corridas em cada lote de corridas j . Em seguida, os autores propõem um método que combina as técnicas de Relaxação Lagrangiana e Geração de Colunas para resolver o problema proposto. O método foi testado usando instâncias geradas a partir de uma semana de produção de uma grande siderúrgica

chinesa (Baosteel) e instâncias geradas aleatoriamente. Os testes mostraram que o algoritmo combinado é altamente eficiente e gera soluções muito próximas do ótimo.

Outro trabalho que trata do problema de agrupamento de séries é apresentado por Tang e Luo [53]. Assim como Tang e Wang [54] e Tang *et al.* [55], este trabalho é um estudo de caso da siderúrgica Baosteel, na China. Para resolver o problema, os autores desenvolvem uma nova busca ILS (*Iterated Local Search*) para determinar soluções próximas à ótima. O algoritmo ILS proposto pelos autores se destaca em três pontos: (1) duas heurísticas são propostas para gerar uma solução inicial para o ILS; (2) uma estrutura de vizinhança muito ampla é adotada (vizinhança de transferência cíclica) na busca local e (3) uma nova estratégia de perturbação é desenvolvida com a ideia de alocação de corridas para diferentes séries de acordo com os custos de dissimilaridade entre elas e as corridas das séries centrais identificadas pela solução corrente. Com a finalidade de melhorar a produtividade da MLC, tantas corridas quanto possível podem ser agrupadas em uma série, uma vez que menos séries para um fixado número de corridas significa menos tempo de *setup* para a MLC.

Tang e Luo [53] propõem, também, uma formulação de programação matemática inteira quadrática para o problema agrupamento de séries. A função objetivo, deste modelo, inclui o somatório dos custos das dissimilaridades entre corridas, o custo de *setup* das séries e as penalidades para corridas não selecionadas. Em seguida, os autores apresentam um modelo de PLIM equivalente ao modelo de programação matemática inteira quadrática.

Os testes do algoritmo ILS apresentado por Tang e Luo [53] são feitos com instâncias geradas a partir de dados obtidos na siderúrgica chinesa em estudo. Os resultados são comparados com resultados obtidos pelo modelo de PLIM, das heurísticas para geração da solução inicial e com resultados obtidos junto à empresa siderúrgica.

Tang *et al.* [51] tratam do problema integrado de agrupamento de corridas e séries com seleção de larguras em uma grande siderúrgica chinesa. O problema, resolvido por estes autores, tem como objetivo encontrar m séries viáveis, uma para cada distribuidor, tal que o número de corridas alocadas às séries é maximizado e o custo total de mistura e de trocas de largura destas séries é minimizado. Os autores ressaltam que o distribuidor atua como um pulmão entre a panela e a MLC, tal que uma panela vazia pode ser removida e uma nova, preenchida, pode ser posicionada sem a interrupção do processo de LC. Segundo os autores, é crítico para qualquer empresa siderúrgica aumentar a utilização do distribuidor, representado pelo número

médio de corridas em cada distribuidor, tal que a produtividade é aumentada e o custo de produção é reduzido. E, segundo os autores, isto pode ser alcançado através do planejamento e programação efetivos da produção.

Tang *et al.* [51] afirmam que o problema integrado de agrupamento de corridas e séries com seleção de larguras tem recebido pouca atenção da literatura acadêmica. Segundo os autores, não há trabalhos que tratem de forma integrada o agrupamento de corridas e séries com seleção de larguras. Tang *et al.* [51], ainda, classificam este problema como NP-Difícil, ou seja, não existe um algoritmo que consiga resolvê-lo em tempo polinomial. Para resolver o problema integrado de agrupamento de corridas e séries com seleção de larguras, os autores propõem um algoritmo baseado em geração de colunas e utilizam um algoritmo *branch-and-price* para resolver o problema principal. Os testes do algoritmo foram feitos em instâncias-teste baseadas no cenário real da empresa siderúrgica estudada. Com a utilização do algoritmo, os autores constataram que os distribuidores estavam sendo melhor utilizados em comparação com os resultados obtidos pelo sistema manual usado pela empresa.

A empresa siderúrgica estudada por Tang *et al.* [51] utilizava um sistema no qual as informações sobre as corridas e séries eram passadas manualmente pelo operador. Daí, o algoritmo desenvolvido foi embutido nesse sistema de forma a aumentar a produtividade da empresa. Segundo os autores, o novo sistema desenvolvido propiciará uma economia anual de US\$ 1,6 milhões e um aumento anual na receita em torno de US\$ 3,25 milhões à empresa estudada.

3.3.2 Decisões de Sequenciamento

Tang *et al.* [49] apresentam uma revisão bibliográfica sobre trabalhos que apresentam sistemas e métodos de planejamento e sequenciamento para a produção integrada de aço. Estes autores afirmam que somente com um sistema racional de planejamento e sequenciamento os benefícios potenciais de um novo processo de produção podem ser alcançados.

Inicialmente, Tang *et al.* [49] apresentam algumas empresas (a maioria delas japonesas) que adotam um sistema de gestão da produção para a aciaria, lingotamento contínuo e laminação à quente. Estes autores destacam os principais benefícios econômicos para as empresas que adotaram este sistema de gestão integrada da produção. Entre estes benefícios destacam-se: (i) Economia de energia; (ii) Aumento da eficiência e eficácia na criação do plano de atividades; (iii) Redução do estoque de

placas e sucata; (iv) Aumento da taxa de pedidos entregues no prazo; (v) Melhora dos lucros da produção.

Em seguida, Tang *et al.* [49] apresentam uma revisão sobre métodos integrados de planejamento e sequenciamento da produção. Os autores classificam os métodos em quatro tipos: (i) Métodos baseados em Pesquisa Operacional; (ii) Métodos baseados em inteligência artificial; (iii) Métodos de coordenação homem-máquina; (iv) Métodos multi-agentes.

Harjunoski e Grossmann [22] apresentam uma estratégia de decomposição para determinar o sequenciamento de ordens de produção em uma aciaria. Os autores justificam a aplicação desta estratégia no fato de problemas de programação das aciarias envolvem a manipulação de um grande número de complicadas regras químicas, geométricas e de sequenciamento. Devido a este fato, utilizar modelos de programação linear inteira mista (PLIM) pode requerer um excessivo tempo computacional. Os autores dizem, ainda, que os modelos de PLIM podem ser complementados com heurísticas e, neste caso, o fato que os *jobs* (corridas) necessitam ser agrupados em sequências para o último estágio de processamento, o lingotamento contínuo, motivam a estratégia de decomposição.

A estratégia de decomposição proposta por Harjunoski e Grossmann [22] consiste em primeiramente particionar as ordens dos consumidores em grupos de corridas com propriedades similares, cada qual é otimizado como um problema de *jobshop*. Em seguida, cada um dos grupos são otimamente sequenciados como um problema de *flow-shop* e, finalmente, um método de programação linear e/ou linear inteira mista é usado para, adequadamente, contabilizar os tempos de *setup* e para otimizar a alocação à algum equipamento em paralelo. Esta estratégia não garantia que a solução ótima fosse encontrada. A estratégia de decomposição proposta pelos autores pôde ser aplicada com sucesso às programações próximas a 100 produtos, que é quase igual a 1 semana de produção.

O problema resolvido por Tang *et al.* [48] consiste em determinar qual a sequência, em que momento do tempo e em que recurso, o aço líquido deve ser arranjado dentro dos vários processos, da produção do aço ao lingotamento contínuo. Para resolver este problema, primeiramente, Tang *et al.* [48] propõem um modelo de programação não-linear. Este modelo busca garantir a continuidade da produção e a entrega no prazo certo dos produtos finais. Em seguida, os autores convertem este modelo em um modelo de programação linear, o qual é resolvido por pacotes de otimização

existentes no mercado. Daí, desenvolveu-se um sistema computacional que integrava homem-computador. Este sistema foi implementado e testado na empresa estudada no fim de 1995. O tempo médio de espera foi reduzido de 20 para 14 minutos por corrida. Isto traduziu-se em uma economia em torno de 0,5 milhões de dólares por ano em custos de produção e um aumento em torno de 5% em rendimento.

No processo estudado por Tang *et al.* [50], corridas na mesma série necessitam satisfazer as seguintes restrições tecnológicas: (1) composições químicas (*steel grades*) para corridas adjacentes devem ser idênticas ou similares; (2) as medidas da placa de diferentes corridas devem ser idênticas; (3) as corridas devem ser sequenciadas tal que as larguras da placa estão em ordem decrescente; (4) as diferenças na largura da placa das corridas devem estar dentro de um certo limite e a mudança de largura entre corridas adjacentes não podem exceder um dado valor máximo; (5) o número total de corridas em uma série deve estar entre um dado limite inferior e um dado limite superior que é determinado pela vida útil do distribuidor; (6) datas de entrega de diferentes corridas na mesma série devem ser as mais próximas possíveis. Tempo de *setup* na MLC não é necessário entre corridas adjacentes na mesma série. Entretanto, um tempo relativamente longo de *setup* é requerido entre duas séries na mesma MLC para troca do distribuidor. Corridas e séries são definidas no nível de planejamento tático. Após o planejamento tático, as ordens em cada corrida e sua sequência são fixadas. As corridas em cada série e sua sequência também são fixadas. A programação da aciaria é então decidir a programação destas séries nas máquinas.

O modelo proposto por Tang *et al.* [50] divide o horizonte de planejamento em períodos e as variáveis deste modelo são indexadas no tempo. A metodologia proposta pelos autores para resolver o problema de programação da aciaria é uma combinação de relaxação lagrangiana, programação dinâmica e heurísticas. O método proposto é testado com instâncias geradas aleatoriamente que foram desenvolvidas para refletir situações práticas na indústria de ferro e aço. Os autores procuram determinar a programação para um turno de 8 horas.

Pacciarelli e Pranzo [38] utilizam um algoritmo do tipo *Beam Search* para resolver um problema de programação da produção em uma aciaria. O processo da aciaria resolvido pelos autores é constituído de 4 etapas: forno elétrico a arco, descarburização por sopro combinado de oxigênio e argônio, forno panela e lingotamento contínuo. O algoritmo *Beam Search* é uma técnica de enumeração parcial, ela pode ser vista como um *branch-and-bound* truncado, onde somente as ramificações que são mais prováveis de gerar soluções ótimas são exploradas. Os autores usam um grafo alternativo para

modelar o problema e resolvem instâncias com até 30 *jobs*.

Zanoni e Zavanella [61] propõem um modelo de PLIM para resolver o problema de programação em uma pequena siderúrgica. No modelo proposto por Zanoni e Zavanella [61], a função objetivo busca minimizar os custos de produção (matéria-prima, energia, mão-de-obra e manutenção) e custos devido à pedidos atrasados. Neste modelo, levam-se em consideração o espaço para armazenamento e a capacidade de produção. Apesar dos autores citarem a incompatibilidade entre grupos de aço, isto não é levado em consideração na construção do modelo. Os autores propõem também um algoritmo para resolver problemas onde não foi possível encontrar a solução ótima. Fez-se testes com problemas gerados aleatoriamente baseados no contexto industrial estudado.

Ferretti *et al.* [19] resolvem o mesmo problema de Zanoni e Zavanella [61], usando a meta-heurística Colônia de Formigas (ACO). Os autores fazem um paralelo entre o problema estudado e o problema do Caixeiro Viajante (PCV) para aplicar o algoritmo ACO. Nesta comparação, enquanto os arcos representam as distâncias entre as cidades no PCV, no problema de programação ele representa o tempo de *setup* necessário para processar a ordem j após a ordem i . Os resultados do método proposto foram comparados com o modelo de PLIM proposto por Zanoni e Zavanella [61]. Para isto, gerou-se instâncias teste baseadas em dados históricos do caso industrial real estudado.

Bellabdaoui e Teghem [7] desenvolvem um modelo de programação linear inteira mista para resolver o problema de programação de uma aciaria. O processo da aciaria estudado pelos autores consistia de três estágios principais: dois convertedores idênticos, duas estações de refino com mesmo tempo de processamento para cada máquina, e duas máquinas de lingotamento contínuo. As corridas são reunidas de acordo com o pedido, a qualidade do aço e as dimensões da placa na saída do lingotamento contínuo. A função objetivo usada pelos autores procurava minimizar o tempo total para a conclusão de todas as sequências. Este trabalho é baseado em um estudo de caso em uma grande siderúrgica na Bélgica. Para resolver o modelo, os autores utilizaram pacotes de otimização disponíveis no mercado. Os autores resolvem problemas de 6 a 8 *jobs* (corridas).

Atighehchian *et al.* [4] propõem um método baseado na meta-heurística Colônia de Formigas (ACO) e em métodos de otimização não-linear. Este trabalho apresenta um estudo de caso em uma grande companhia siderúrgica no Irã, Oriente Médio. O

método proposto pelos autores, denominado HANO, é dividido em duas fases: (i) A primeira fase determina a sequência dos *jobs* (corridas) através do algoritmo colônia de formigas (ACO); (ii) a segunda fase determina os instantes de tempo que cada *job* será processado através da resolução de um modelo de programação não-linear. Nota-se no trabalho de Atighehchian *et al.* [4] que, apesar de falar que é possível fazer a mistura entre dois tipos de aços, o custo desta mistura não é computado na função de avaliação do método.

Missbauer *et al.* [36] apresentam o estudo de caso da implantação e desenvolvimento de um sistema integrado de planejamento e sequenciamento da produção de uma grande siderúrgica na Áustria. O processo estudado por estes autores consistia de quatro fases: (1) preparação do metal aquecido; (2) fabricação do aço (processo LD); (3) refino (metalurgia secundária); (4) lingotamento contínuo.

A empresa estudada por Missbauer *et al.* [36] possui três convertedores LD para a produção do aço, dois fornos panela e dois fornos de desgaseificação à vácuo (RH) para o refino secundário e quatro máquinas de lingotamento contínuo (três máquinas simples com um único veio, e uma dupla com dois veios). A produção média da empresa é de 100 corridas por dia. O nível de planejamento que define as metas para o planejamento da aciaria é um planejamento integrado para as MLC e laminadores. Este nível de planejamento determina as seguintes informações: (i) as séries; (ii) a alocação das séries nas MLC; (iii) a sequência das séries nas MLC; (iv) as datas preliminares de início e fim de cada operação. O principal objetivo é maximizar o número de corridas por dia, que segundo os autores, corresponde a minimizar o *makespan*. Os transportes entre estágios aparecem como operações no processo e as pontes rolantes são consideradas como facilidades. Missbauer *et al.* [36] apresentam um modelo de PLIM para resolver o problema. O problema foi considerado como um problema de *flowshop* híbrido.

Li *et al.* [29] propõem um modelo de programação linear para resolver o problema de sequenciamento na aciaria. O modelo proposto por estes autores divide o horizonte de planejamento em cada facilidade em pontos de evento (*events point*) e a partir daí procura alocar as séries nestes pontos de evento e obter a sequência de séries em cada facilidade. A função objetivo deste modelo procura minimizar o *makespan*.

O processo estudado por Li *et al.* [29] possui quatro estágios, cada um deles possuindo várias máquinas em paralelo. Os quatro estágios são: (1) fornos de arco elétrico (EAF); (2) descarburização, redução e dessulfuração; (3) fornos panela (LF);

(4) máquina de lingotamento contínuo (MLC). Existem estoques intermediários entre alguns estágios. A produção é organizada em séries contendo uma ou várias corridas com similar composição química. As corridas pertencentes à mesma série devem ser lingotadas sem interrupção, caso contrário, o distribuidor deteriora e um *setup* é necessário para substituí-lo. O distribuidor deve ser trocado sempre quando há mudança de uma série para outra ou para um dado número de corridas. Uma suposição assumida pelos autores é que a sequência de corridas pertencentes à mesma série são determinadas no estágio de planejamento e, por isso, são conhecidas à priori.

Tang *et al.* [52] abordam o problema em 3 estágios: produção do aço, refinamento e lingotamento contínuo. A diferença deste trabalho para os descritos anteriormente é que os autores procuram incorporar eventos dinâmicos que podem acontecer durante um dia de produção ao problema estudado. Os autores classificam estes eventos dinâmicos em: eventos relacionados ao recurso (tal como quebra de máquina); eventos relacionados ao *job* (tal como cancelamento de um *job*). Os autores apresentam um modelo de PLIM e desenvolvem um algoritmo de evolução diferencial para resolver o problema. Os autores testaram o algoritmo desenvolvido com duas categorias de instâncias-teste: geradas aleatoriamente à partir de dados práticos e geradas à partir de uma aplicação prática de 10 dias de produção em uma grande siderúrgica chinesa. Os autores constataram que o algoritmo de Evolução Diferencial foi mais efetivo e conseguiu obter melhores soluções em comparação ao método manual usado pela empresa.

Mao *et al.* [31] e Pan *et al.* [39] tratam o problema de sequenciamento na aciaria como um problema de *flowshop* híbrido, onde cada etapa é constituída de um certo conjunto de máquinas disponíveis. O objetivo dos trabalhos de ambos autores é determinar a sequência em cada uma das etapas do processo de forma a minimizar os custos de antecipação e atraso da produção. Além disso, ambos autores utilizam dados oriundos da siderúrgica Baosteel, na China. Nestes trabalhos, o setor de LC é composto de três máquinas e não pode haver ociosidade entre dois *jobs* processados consecutivamente dentro de uma mesma série.

Mao *et al.* [31] apresentam um modelo de PLIM para representar o problema de sequenciamento na aciaria. Para resolução do modelo de PLIM, os autores propõem uma nova aproximação de relaxação lagrangiana, onde o problema lagrangiano dual é resolvido por um algoritmo sub-gradiente de melhora, que garante a convergência sem requerer a solução ótima dual à priori. Já Pan *et al.* [39] aplicam um algoritmo de colônia de abelhas artificial (*Artificial Bee Colony Algorithm*) para resolver o problema.

3.3.3 Conclusões

Dos problemas apresentados anteriormente, o problema apresentado por Tang *et al.* [51] é o que mais se assemelha com o problema tratado neste trabalho. Uma das diferenças entre o trabalho aqui proposto e o trabalho proposto por Tang *et al.* [51] é que no problema de agrupamento de corridas (panelas), considera-se o pedido do cliente em toneladas, ao invés do número de placas. Além disso, no agrupamento das panelas considera-se também os custos de atraso e de estoque, sendo que no trabalho de Tang *et al.* [51] consideram-se apenas os custos de mistura, *setup* e trocas de largura entre duas corridas consecutivas. Outro ponto que cabe salientar é que o modelo apresentado por Tang *et al.* [51] resolve apenas um período por vez, sendo que os modelos aqui propostos procuram resolver o sequenciamento para diversos períodos.

A maioria dos trabalhos citados anteriormente tratam o problema de sequenciamento na aciaria como um problema de *flowshop* híbrido, procurando determinar a sequência de produção em todas as etapas do processo de transformação do ferro-gusa em aço. Neste trabalho, decidiu-se determinar o sequenciamento apenas no processo de lingotamento contínuo, pois, como foi dito anteriormente, é o setor que dita o ritmo da aciaria, não podendo haver ociosidade entre duas corridas (panelas) processadas consecutivamente.

Evidencia-se, também, que a maior parte dos trabalhos que tratam do problema de sequenciamento na aciaria lidam apenas com o problema de sequenciamento das panelas nos diversos equipamentos. E, além disso, pouquíssimos trabalhos tratam de forma integrada o problema de sequenciamento de panelas e o problema de agrupamento de pedidos.

Capítulo 4

Problema de Sequenciamento de Painéis

Inicialmente, neste capítulo, apresentam-se quatro modelos de PLIM para resolver o problema de sequenciamento de painéis (PSP) na máquina de lingotamento contínuo. O objetivo destes modelos é determinar a sequência das painéis que serão processadas na MLC, minimizando os custos de *setup* e de geração de placas de mistura, respeitando as restrições operacionais do problema. Os quatro modelos diferenciam-se na maneira como as variáveis de decisão são definidas. Tais modelos são descritos de forma detalhada na seção 4.1. Em seguida, apresentam-se como foram geradas as instâncias usadas nos testes e os resultados da resolução destas instâncias com os modelos propostos.

Finalizando o capítulo, apresentam-se duas heurísticas para resolver o problema de sequenciamento de painéis e os resultados da aplicação destas heurísticas às instâncias geradas na seção 4.2.1. A primeira heurística desenvolvida faz uma analogia entre o PSP e o problema de árvore geradora mínima capacitada. Esta primeira heurística é baseada na heurística de Esau-Williams (ver Esau e Williams [16]), utilizada para resolver o problema de árvore geradora mínima capacitada. A segunda heurística busca melhorar os resultados obtidos pela primeira heurística. Esta heurística é baseada na meta-heurística *multi-start* e utiliza um procedimento, análogo ao processo de geração de uma solução inicial do método GRASP, para escolher a combinação que será feita a cada iteração da heurística de Esau-Williams. Este processo é repetido por certo número de iterações.

4.1 Modelos Matemáticos para o Problema de Sequenciamento de Paneas

Nesta seção são apresentados os modelos de programação linear inteira mista para resolver o problema de sequenciamento de paneas no setor de lingotamento contínuo da ESB. Escolheu-se por determinar o sequenciamento apenas no setor de lingotamento contínuo.

Outras características que são levadas em consideração para este problema são:

- Um *job* corresponde a uma panela a ser vazada (corrida);
- Deve-se respeitar um intervalo de tempo máximo admissível entre dois *setups* consecutivos, pois depois de algumas corridas ocorre desgaste do material refratário que compõe o distribuidor e o mesmo deve ser substituído;
- Deve-se respeitar o limite de variação máxima de largura entre duas corridas consecutivas;
- Diferentes tipos de aço exigem velocidades de lingotamento diferentes;
- Caso haja o lingotamento seguido de dois tipos de aço diferentes, haverá a produção de placa de mistura, que pode não ser aproveitada para aqueles tipos de aço, para que esta mistura não aconteça pode ser realizado um *setup*.

Desenvolveu-se quatro modelos para resolver o problema descrito anteriormente. No primeiro modelo, as variáveis binárias são usadas para identificar se dois *jobs* são processados um após o outro. Já no segundo modelo, as variáveis binárias são usadas para identificar a posição de um *job* na sequência. O terceiro utiliza variáveis indexadas no tempo e o último modelo faz uma analogia entre o problema de sequenciamento na aciaria e o problema de roteamento de veículos assimétrico. Todos os modelos utilizam os seguintes parâmetros de entrada:

n = número total de *jobs* (paneas) a serem processados;

$N = \{1, 2, \dots, n\}$ - conjunto de *jobs* (paneas) a serem processados;

l = intervalo de tempo máximo entre dois *setups* consecutivos;

$$b_{ij} = \begin{cases} 1, & \text{se o } job\ j \text{ pode ser processado imediatamente após o } job\ i \\ 0, & \text{caso contrário} \end{cases}$$

p_i = tempo de processamento do *job* i ;

s = tempo de *setup*;

f_{ij} = custo da placa de mistura formada quando o *job* j é processado imediatamente após o *job* i ;

h = custo de uma operação de *setup*;

4.1.1 Modelo M

O modelo M, assim como Manne [30], usa variáveis binárias para: (i) indicar se dois *jobs* são processados um imediatamente após o outro, independentemente se eles formam placa de mistura ou não; e (ii) indicar se é feita uma operação de *setup* após o processamento de um *job*. Além disso, usam-se variáveis contínuas para a data de início do processamento de um *job*, para a data de início de um *setup*, e também para indicar se há formação de placa de mistura entre dois *jobs*. Estas variáveis são descritas a seguir:

$$\beta_{ij} = \begin{cases} 1, & \text{se o } job\ j \text{ é o próximo a ser processado após o } job\ i \\ 0, & \text{caso contrário} \end{cases}$$

$$w_{ik} = \begin{cases} 1, & \text{se o } k\text{-ésimo } setup \text{ é realizado após o } job\ i \text{ ter sido processado} \\ 0, & \text{caso contrário} \end{cases}$$

t_i = data de início do processamento do *job* i ;

θ_k = data de início da k -ésima operação de *setup*;

q_{ij} = indica se o *job* j é processado imediatamente após o *job* i e formam uma placa de mistura.

Neste modelo, é necessário a utilização de dois *jobs* artificiais. Os *jobs* artificiais 0 e $n+1$ abrem e fecham, respectivamente, a sequência. A variável t_0 e os parâmetros p_0 e p_{n+1} são fixados em zero. Utiliza-se, ainda, um limite superior $M = (n-1)s + \sum_{i \in N} p_i$ para o tempo de finalização de todos os *jobs*. O modelo matemático é descrito como se segue.

$$\min \sum_{i \in N} \sum_{j \in N} f_{ij} q_{ij} + \sum_{k=1}^{n-1} \sum_{i \in N} h w_{ik} \quad (4.1)$$

$$\text{s.â: } t_j - t_i - s \sum_{k=1}^{n-1} w_{ik} - (1 - \beta_{ij})M \leq p_i \quad \forall i \in N \cup \{0\}, \forall j \in N \cup \{n+1\} \quad (4.2)$$

$$t_j - t_i - s \sum_{k=1}^{n-1} w_{ik} + (1 - \beta_{ij})M \geq p_i \quad \forall i \in N \cup \{0\}, \forall j \in N \cup \{n+1\} \quad (4.3)$$

$$\sum_{j \in N \cup \{n+1\}} \beta_{ij} = 1 \quad \forall i \in N \cup \{0\} \quad (4.4)$$

$$\sum_{i \in N \cup \{0\}} \beta_{ij} = 1 \quad \forall j \in N \cup \{n+1\} \quad (4.5)$$

$$q_{ij} - \beta_{ij} + \sum_{k=1}^{n-1} w_{ik} \geq 0 \quad \forall i, j \in N \quad (4.6)$$

$$q_{ij} \leq b_{ij} \quad \forall i, j \in N \quad (4.7)$$

$$\sum_{i \in N} w_{i1} \leq 1 \quad (4.8)$$

$$\sum_{i \in N} w_{ik} - \sum_{i \in N} w_{i(k-1)} \leq 0 \quad k = 2, \dots, n-1 \quad (4.9)$$

$$\sum_{k=1}^{n-1} w_{ik} \leq 1 \quad \forall i \in N \quad (4.10)$$

$$\theta_k - t_i - (1 - w_{ik})M \leq p_i \quad \forall i \in N, k = 1, \dots, n-1 \quad (4.11)$$

$$\theta_k - t_i + (1 - w_{ik})M \geq p_i \quad \forall i \in N, k = 1, \dots, n-1 \quad (4.12)$$

$$\theta_1 - l \sum_{i \in N} w_{i1} \leq 0 \quad (4.13)$$

$$\theta_k - \theta_{(k-1)} - (l + s) \sum_{i \in N} w_{ik} \leq 0 \quad k = 2, \dots, n-1 \quad (4.14)$$

$$\theta_k - \theta_{(k-1)} + (l + s) \sum_{i \in N} w_{ik} \geq 0 \quad k = 2, \dots, n-1 \quad (4.15)$$

$$t_{n+1} - \theta_{n-1} - s \sum_{i \in N} w_{i1} \leq l \quad (4.16)$$

$$t_0 = 0 \quad (4.17)$$

$$w_{0k} = 0 \quad k = 1, \dots, n-1 \quad (4.18)$$

$$\beta_{ij} \in \{0, 1\} \quad \forall i \in N \cup \{0\}, \forall j \in N \cup \{n+1\} \quad (4.19)$$

$$w_{ik} \in \{0, 1\} \quad \forall i \in N, k = 1, \dots, n-1 \quad (4.20)$$

$$t_i \geq 0 \quad \forall i \in N \cup \{n+1\} \quad (4.21)$$

$$\theta_k \geq 0 \quad k = 1, \dots, n-1 \quad (4.22)$$

$$q_{ij} \geq 0 \quad \forall i, j \in N \quad (4.23)$$

A função objetivo (4.1) é o custo total obtido pelo somatório das perdas monetárias devido à geração de placas de mistura e valores monetários dos distribuidores substituídos. As restrições disjuntivas de precedência (4.2) e (4.3) requerem que há somente um *job* sendo processado ou uma operação de *setup* ocorrendo na máquina por vez. Pelas restrições (4.4) e (4.5) cada *job* em N tem somente um *job* imediatamente predecessor e um *job* sucessor na sequência. As duas ações possíveis enfrentadas pelos tomadores de decisão quando processam um *job* após o outro são modeladas nas restrições (4.6), uma vez que uma placa de mistura ou uma operação de *setup* devem ser feitas sempre que o *job* j é processado após o *job* i . Restrições (4.7) limitam a possibilidade de formar placas de mistura para os pares de *jobs* para os quais isto é permitido. Pelas restrições (4.8) a (4.10) o k -ésimo *setup* somente pode ocorrer se o $(k - 1)$ -ésimo *setup* é realizado e para somente um *job*. Se o k -ésimo *setup* é realizado após o *job* i ser processado, as restrições (4.11) e (4.12) fazem com que a data de início da realização deste *setup* seja igual a data de conclusão de i . Restrições (4.13) a (4.16) garantem que no mínimo um *setup* é realizado em um período de tempo l . Note que se $\sum_{i \in N} p_i \leq l$, então pode ser viável processar todos os *jobs* sem qualquer *setup*. Neste caso, as restrições (4.13) a (4.15) fazem $\theta_k = 0$, $k = 1, \dots, n - 1$, e pelas restrições (4.16) a sequência não deve consumir um tempo maior que l . Caso contrário, $1 \leq m \leq n - 1$ *setups* podem ser feitos, então as restrições (4.14) e (4.15) fazem θ_u , $u = m + 1, \dots, n - 1$, igual a θ_m e as restrições (4.16) garantem que a última série da sequência não gaste um tempo maior que l . Finalmente, restrições (4.17) a (4.23) definem o domínio das variáveis.

4.1.2 Modelo W

Neste modelo, assim como Wagner [60], usam-se variáveis binárias para: (i) indicar em qual posição um *job* é processado na sequência; e (ii) indicar se uma operação de *setup* é realizada após o processamento de um *job*. Além disso, usam-se variáveis contínuas para a data de início de processamento de um *job* e para indicar se uma placa de mistura é formada entre dois *jobs*. As variáveis para o Modelo W são apresentadas a seguir:

$$\beta_{ik} = \begin{cases} 1, & \text{se o } job \ i \text{ ocupa a } k\text{-ésima posição na sequência} \\ 0, & \text{caso contrário} \end{cases}$$

$$w_k = \begin{cases} 1, & \text{se um } \textit{setup} \text{ é realizado após o job ocupando a } k\text{-ésima posição} \\ 0, & \text{caso contrário} \end{cases}$$

t_k = data de início de processamento do *job* ocupando a k -ésima posição

q_{ij} = indica se o *job* j é processado imediatamente após o *job* i e forma placa de mistura

Desta forma, o modelo W é definido como se segue.

$$\min \sum_{i \in N} \sum_{j \in N} f_{ij} q_{ij} + \sum_{k \in N - \{n\}} h w_k \quad (4.24)$$

$$\text{s.à:} \quad \sum_{k \in N} \beta_{ik} = 1 \quad \forall i \in N \quad (4.25)$$

$$\sum_{i \in N} \beta_{ik} = 1 \quad \forall k \in N \quad (4.26)$$

$$t_k - t_{(k-1)} - \sum_{i \in N} \beta_{i(k-1)} p_i - w_{(k-1)} s = 0 \quad \forall k \in N, k \geq 2 \quad (4.27)$$

$$\beta_{jk} + \beta_{i(k-1)} - w_{(k-1)} - q_{ij} \leq 1 \quad \forall i, j, k \in N, k \geq 2, i \neq j \quad (4.28)$$

$$q_{ij} \leq b_{ij} \quad \forall i, j \in N \quad (4.29)$$

$$-(l + s) \sum_{v=k}^{r-1} w_v + t_r + \sum_{i \in N} \beta_{ir} p_i - t_k \leq l \quad \forall k, r \in N, r > k \quad (4.30)$$

$$t_1 = 0 \quad (4.31)$$

$$\beta_{ik} \in \{0, 1\} \quad \forall i, k \in N \quad (4.32)$$

$$w_k \in \{0, 1\} \quad \forall k \in N - \{n\} \quad (4.33)$$

$$t_k, q_{ik} \geq 0 \quad \forall i, k \in N \quad (4.34)$$

A função objetivo (4.24) é o custo total obtido pelo somatório das perdas monetárias devido às placas de mistura geradas e o valor monetário de substituição de um distribuidor. Restrições (4.25) e (4.26) definem a sequência de *jobs*. Pelas restrições (4.27), o *job* na k -ésima posição inicia seu processamento após a finalização, na $(k - 1)$ -ésima posição, de um *job* mais o *setup* se o último for realizado. Restrições (4.28) modelam as duas possíveis ações de fazer uma placa de mistura ou realizar uma operação de *setup* quando o *job* j é processado após o *job* i . Restrições (4.29) limitam a possibilidade de formar placas de mistura aos pares de *jobs* para os quais isto é permitido. Restrições (4.30) garantem no mínimo um *setup* seja realizado em um

período de tempo l . Além disso, este conjunto de restrições procura determinar após qual posição a operação de *setup* deverá ser realizada. Nestas restrições, pelo menos algum w_v deverá ser igual a 1 quando $l - t_r - \sum_{i \in N} \beta_{ir} p_i + t_k < 0$, ou seja, quando o tempo entre a data de início de um *job* na posição k e a data de conclusão de um *job* na posição r for superior a l . Finalmente, as restrições (4.31) a (4.34) definem o domínio das variáveis.

4.1.3 Modelo I

O terceiro modelo, denominado Modelo I, utiliza variáveis binárias para indicar qual é o instante de tempo que se inicia o processamento de um determinado *job* e para indicar se é realizada uma operação de *setup* após o processamento de um *job* que inicia em um determinado instante de tempo (ver Dyer e Wolsey [15] e Van den Akker *et al.* [59]). Além disso, este modelo utiliza variáveis contínuas para indicar se há formação de placa de mistura entre dois *jobs* e para indicar se um *job* é o último na sequência. Dado um conjunto de períodos $T = \{1, \dots, nt\}$, as variáveis do Modelo I são definidas como se segue:

$$x_i^t = \begin{cases} 1, & \text{se o } job \ i \text{ iniciar seu processamento no período } t \\ 0, & \text{caso contrário} \end{cases}$$

$$w_i^t = \begin{cases} 1, & \text{se é realizada uma operação de } setup \text{ no período } t \text{ após o } job \ i \\ 0, & \text{caso contrário} \end{cases}$$

q_{ij}^t = indica se o *job* i que começa no período t gera placa de mistura com o *job* j ;

γ_i^t = indica se o *job* i que começa no período t é o último processado.

E o Modelo I é definido como se segue:

$$\min \sum_{i \in N} \sum_{j \in N} \sum_{t=1}^{nt} q_{ij}^t f_{ij} + \sum_{i \in N} \sum_{t=1}^{nt} w_i^t h \quad (4.35)$$

$$\text{s.à: } \sum_{t=1}^{nt-p_i+1} x_i^t = 1 \quad \forall i \in N \quad (4.36)$$

$$\sum_{i \in N} x_i^1 = 1 \quad (4.37)$$

$$x_i^t + \sum_{u=t}^{t+p_i-1} x_j^u \leq 1 \quad \forall i, j \in N, i \neq j, t \in [1, nt - p_i + 1] \quad (4.38)$$

$$w_i^t + \sum_{u=t}^{t+s-1} x_j^u \leq 1 \quad \forall i, j \in N, i \neq j, t \in [1, nt - s + 1] \quad (4.39)$$

$$w_i^t - \sum_{j \in N - \{i\}} x_j^{t+s} \leq 0 \quad \forall i \in N, t \in [1, nt - s] \quad (4.40)$$

$$x_i^t - \sum_{j \in N - \{i\}} q_{ij}^t - w_i^{t+p_i} - \gamma_i^t = 0 \quad \forall i \in N, t \in [1, nt - p_i + 1] \quad (4.41)$$

$$x_i^t + x_j^{t+p_i} - q_{ij}^t \leq 1 \quad \forall i, j \in N, i \neq j, t \in [1, nt - p_i] \quad (4.42)$$

$$x_i^t - x_j^{t+p_i} + q_{ij}^t \leq 1 \quad \forall i, j \in N, i \neq j, t \in [1, nt - p_i] \quad (4.43)$$

$$\sum_{i \in N} \sum_{t=1}^{nt} \gamma_i^t = 1 \quad (4.44)$$

$$\gamma_i^t + \sum_{u=t}^{nt} x_j^u \leq 1 \quad \forall i, j \in N, i \neq j, t \in T \quad (4.45)$$

$$x_i^t - \gamma_i^t - \sum_{u=t+p_i}^{t+l} \sum_{j \in N} w_j^u \leq 0 \quad \forall i \in N, t \in [1, nt - l + 1] \quad (4.46)$$

$$\sum_{t=nt-s+1}^{nt} w_i^t = 0 \quad \forall i \in N \quad (4.47)$$

$$\sum_{t=nt-p_i+1}^{nt} x_i^t = 0 \quad \forall i \in N \quad (4.48)$$

$$q_{ij}^t \leq b_{ij} \quad \forall i, j \in N, i \neq j, t \in T \quad (4.49)$$

$$q_{ij}^t \geq 0 \quad \forall i, j \in N, i \neq j, t \in T \quad (4.50)$$

$$\gamma_i^t \geq 0 \quad \forall i \in N, t \in T \quad (4.51)$$

$$x_i^t, w_i^t \in \{0, 1\} \quad \forall i \in N, t \in T \quad (4.52)$$

A função objetivo (4.35) tem o mesmo significado dos dois modelos anteriores. As restrições (4.36) asseguram que cada *job* i somente será alocado a um único instante de tempo t . A restrição (4.37) garante que o primeiro *job* na sequência iniciará seu processamento no instante de tempo $t = 1$. As restrições (4.38) asseguram que se um *job* i é programado para iniciar no instante t nenhum outro *job* pode ser programado nos próximos $t + p_i - 1$ períodos. As restrições (4.39) garantem que se um *setup* foi programado para iniciar no instante t , após o processamento do *job* i , nenhum outro *job* pode ser programado nos próximos $t + s - 1$ períodos. As restrições (4.40) garantem que após a realização de um *setup*, um *job* j deve ser programado para iniciar no instante $t + s$. Depois do processamento de um *job* três situações podem ocorrer: (i) um novo *job* pode ser programado, desta forma, gerando a mistura; (ii) é realizado um *setup*; ou (iii) a sequência de *jobs* é finalizada. As restrições (4.41) procuram

garantir que somente uma destas três situações aconteça. As restrições (4.42) e (4.43) procuram garantir que q_{ij}^t será igual a 1, somente se os *jobs* i e j forem processados um imediatamente após o outro, sem realização de *setups*. As restrições (4.44) garantem que somente uma variável γ_i^t finalizará a sequência e as restrições (4.45) garantem que estas variáveis estão associadas a uma única variável x_i^t , que finaliza a sequência. As restrições (4.46) garantem que entre dois *setups* consecutivos há no máximo um tempo l . As restrições (4.47) e (4.48) determinam que as variáveis x_i^t e w_i^t associadas às últimas posições do horizonte de planejamento sejam iguais a zero. As restrições (4.49) a (4.52) definem o domínio das variáveis.

4.1.4 Modelo P

Como o problema de sequenciamento no lingotamento contínuo considera apenas os custos de mistura e *setup*, pode-se fazer uma analogia entre o PSP e o Problema de Roteamento de Veículos Assimétrico Capacitado (PRVAC). Para fazer tal analogia, primeiramente, não são levadas em consideração as datas de início e conclusão dos *jobs*, mas somente a posição do *job* dentro da sequência. Nesta analogia, os nós da rede representam os *jobs* a serem sequenciados e os arcos indicam se dois *jobs* podem ser sequenciados um após o outro dentro da programação. As demandas dos nós da rede são representadas pelo tempo de processamento dos *jobs*. Além disso, a capacidade do veículo corresponde ao tempo máximo entre dois *setups* consecutivos (l). Baseado nestas informações tem-se então a seguinte variável de decisão:

$$x_{ij} = \begin{cases} 1, & \text{se o } job\ j \text{ é sequenciado logo após o } job\ i \\ 0, & \text{caso contrário} \end{cases}$$

O centro de distribuição, representado por 0, corresponde à realização de um *setup*, ou seja, toda vez que um *job* for ligado ao nó central ($x_{i0} = 1$ e $x_{0j} = 1$) indica que um *setup* está sendo realizado e uma nova série está sendo iniciada. Neste modelo não há controle do número de veículos, sendo utilizado o que for estritamente necessário.

Além dos parâmetros de entrada definidos anteriormente, o Modelo P considera, também, o seguinte parâmetro:

c_{ij} = custos nos arcos, que correspondem ao custo de oportunidade se o *job* j é sequenciado logo após o *job* i ou ao custo de *setup* quando o *job* i ou j são

ligados ao nó central. Resumidamente, os custos nos arcos podem ser definidos da seguinte forma:

$$c_{ij} = \begin{cases} f_{ij}, & \text{se } i \text{ e } j \neq 0 \text{ e } b_{ij} = 1; \\ \infty, & \text{se } i \text{ e } j \neq 0 \text{ e } b_{ij} = 0; \\ h, & \text{se } i = 0; \\ 0, & \text{se } j = 0. \end{cases}$$

Além destas informações tem-se também a variável auxiliar Φ_{ij} que corresponde ao fluxo de tempo enviado do nó i para o nó j . Desta forma, o modelo é definido como se segue.

$$\min \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij} \quad (4.53)$$

$$\text{s.à:} \quad \sum_{i=0}^n x_{ik} = 1 \quad \forall k = 1, \dots, n \quad (4.54)$$

$$\sum_{j=0}^n x_{kj} = 1 \quad \forall k = 1, \dots, n \quad (4.55)$$

$$\sum_{j=1}^n x_{0j} - \sum_{i=1}^n x_{i0} = 0 \quad (4.56)$$

$$\sum_{i=0}^n \Phi_{ik} - \sum_{j=0}^n \Phi_{kj} = p_k \quad \forall k = 1, \dots, n \quad (4.57)$$

$$\Phi_{ij} - l \cdot x_{ij} \leq 0 \quad \forall i, j = 0, \dots, n \quad (4.58)$$

$$x_{ij} \leq b_{ij} \quad \forall i, j = 1, \dots, n \quad (4.59)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j = 0, \dots, n \quad (4.60)$$

$$\Phi_{ij} \geq 0 \quad \forall i, j = 0, \dots, n \quad (4.61)$$

A equação (4.53) procura minimizar os custos com geração de placas de mistura e os custos de *setups*. O número de *setups* é sempre igual ao número de viagens ao nó central menos uma unidade, por isso, ao apresentar os resultados, desconta-se o custo de uma operação de *setup* na função objetivo. As restrições (4.54) e (4.55) garantem que cada *job* tenham somente um *job* sucessor e somente um *job* antecessor, respectivamente. As restrições (4.56) garantem que o número de arcos que saem do nó central e que chegam no nó central sejam os mesmos. As restrições (4.57) garantem que o fluxo de tempo que chega no nó k menos o fluxo de tempo que sai deste mesmo

nó é igual ao tempo de processamento do *job* k . As restrições (4.58) garantem que o fluxo de tempo nos arcos não seja superior ao tempo máximo permitido entre dois *setups* consecutivos, ou seja, assim como as restrições (4.30), procuram garantir que ocorra no mínimo uma operação de *setup* em um período de tempo l . Além disso, as restrições (4.57) e (4.58) garantem que não sejam gerados sub-ciclos na solução do problema. As restrições (4.59) a (4.61) definem o domínio das variáveis.

4.2 Instâncias e Resultados

Nesta seção é apresentado como as instâncias-teste são geradas. Explica-se detalhadamente como, a partir do cenário real, elas foram criadas. Estas instâncias representam o problema de sequenciamento encontrado pela ESB. Criou-se um total de 96 instâncias, as quais se tentou resolver com os modelos de programação linear inteira mista.

Em seguida, apresentam-se os resultados obtidos com os modelos para estas instâncias. As soluções obtidas são avaliadas através da qualidade da solução gerada e pelo tempo computacional necessário para a resolução das instâncias. Para medir a qualidade da solução, os resultados obtidos com os modelos são comparados com os resultados apresentados pela ESB.

4.2.1 Geração das Instâncias Baseadas no Caso Real

As instâncias utilizadas neste trabalho foram criadas a partir de um relatório de produção da aciaria da empresa estudada. Este relatório contém informações referentes a um ano de produção, tais como, a sequência de produção, a máquina de lingotamento contínuo utilizada, o tipo de aço, a família a qual este tipo de aço pertence, a largura da placa gerada ao final do processo de lingotamento, o tipo de produto gerado e seu valor de mercado. Além deste relatório, utilizou-se, também, uma planilha contendo informações sobre a composição química de cada tipo de aço e um documento da empresa que possibilitava calcular a quantidade de mistura gerada para cada combinação de tipos de aço.

A partir do relatório de produção extraiu-se informações pertinentes a alguns dias do ano e criou-se um conjunto de 48 instâncias contendo de 11 a 24 painelas a serem processadas. Criou-se também um conjunto de 48 instâncias com um número

maior de painéis, variando de 54 a 72 painéis a serem processadas. Lembrando-se que cada painel equivale a um *job* no problema de sequenciamento. As instâncias maiores correspondem de dois a três dias de produção e as instâncias menores correspondem de meio dia a um dia de produção. Fez-se necessário criar estes dois conjuntos de instâncias, pois os modelos M, W e I não eram capazes de gerar uma solução viável para os problemas maiores dentro do tempo estipulado.

Identificou-se 236 tipos de aços nos dados que foram extraídos do relatório de produção. Estes tipos de aço foram agrupados em 7 famílias de acordo com similaridades na composição química.

Para determinar o custo de oportunidade devido à placa de mistura gerada quando a painel j é processada após a painel i (f_{ij}), levou-se em consideração as seguintes informações:

- (i) o peso (em toneladas) qm_i e qm_j das painéis que fica na mistura;
- (ii) o valor comercial de cada produto vm_i e vm_j em unidades monetárias por tonelada;
- (iii) o valor comercial da placa de mistura gerada sm_{ij} em unidades monetárias por tonelada.

Os valores de qm_i e qm_j são determinados a partir de relatório fornecido pela empresa. Determinam-se estes valores de acordo com a composição química das painéis i e j . Os valores comerciais, também, são obtidos a partir de relatórios da empresa.

Além disso, quando há variação de largura entre duas painéis processadas consecutivamente sem *setup* gera-se uma rebarba, que é aproveitada apenas como sucata. A Figura 4.1 ilustra a situação quando ocorre variação de largura entre duas painéis processadas consecutivamente. Nesta figura, ΔL representa a diferença entre as larguras das placas geradas pelas painéis i e j e D representa a distância necessária para que a largura da placa gerada pela painel j alcance a largura desejada para a placa desta painel.

A variação de largura é computada levando-se em consideração as seguintes informações:

- (i) a perda lateral da painel j (lm_j) de passar da largura da painel i para a largura da painel j ;
- (ii) o valor de vm_j ;

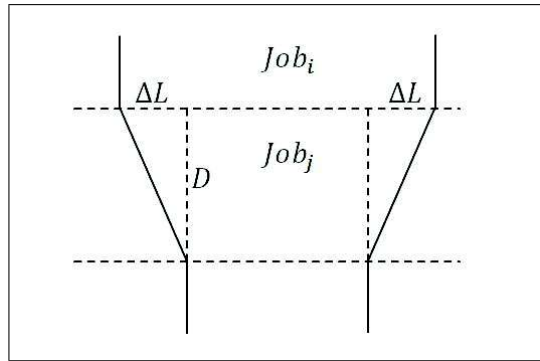


Figura 4.1: Perda de material por mudança na largura das placas

(iii) o valor comercial da sucata rm (a perda lateral só pode ser aproveitada como sucata).

E o custo de oportunidade devido à placa de mistura gerada quando a panela j é processada após a panela i (f_{ij}) é calculado pela equação 4.62.

$$f_{ij} = qm_i(vm_i - sm_{ij}) + qm_j(vm_j - sm_{ij}) + lm_j(vm_j - rm) \quad (4.62)$$

Já os custos de *setup* incluem os custos operacionais e materiais na substituição do distribuidor.

Os tempos de processamento de cada panela são determinados de acordo com o tipo de aço e a família a qual ele pertence. Estas informações são apresentadas na Tabela 4.1.

Tabela 4.1: Tempo de processamento para cada tipo de aço

Família do Aço	Tempo de Processamento (em minutos)
1	40
2	42
3	44
4	44
5	42
6	44
7	44

O tempo de *setup* é constante e igual a 90 minutos. Já o tempo entre dois *setups* consecutivos (l) é igual a 528 minutos, que corresponde ao processamento de 12

panelas consecutivas com tempo de processamento igual a 44 minutos. Utilizou-se este valor, pois a vida útil do distribuidor, segundo a empresa, é, em média, 12 panelas.

O parâmetro b_{ij} é definido da seguinte forma: $b_{ij} = 0$, se a variação de largura entre as panelas i e j é maior que a variação máxima permitida, desta forma, não permitindo que estes tipos de aço sejam sequenciados um após o outro, e $b_{ij} = 1$, caso contrário. Adotou-se nas instâncias geradas a variação máxima permitida igual a maior variação permitida pela empresa, que, neste caso, foi igual a 150 mm.

4.2.2 Resultados e Análise

Apresenta-se nesta seção os resultados obtidos com os modelos de PLIM propostos em instâncias reais derivadas da ESB. Permitiu-se, através de relatórios de produção da empresa, validar os modelos com dados reais e comparar os resultados dos modelos com a programação atual planejada pela ESB.

O problema de sequenciamento de panelas (PSP) no setor de lingotamento contínuo da aciaria da ESB é resolvido com o uso de *softwares* de otimização disponíveis no mercado. Os modelos foram implementados e resolvidos através do *software* de otimização CPLEX 12.6, utilizando-se a linguagem AMPL. Utilizou-se a configuração padrão do CPLEX para a resolução das instâncias. O computador usado para os testes é um Intel Xeon X5690@3.47GHz, com 24 núcleos e 132 GB de memória RAM. Utilizou-se um tempo limite de 7200 segundos para obtenção da solução ótima pelo modelo.

A Tabela 4.2 apresenta os resultados computacionais do Modelo M. A primeira coluna desta tabela identifica as instâncias. A segunda coluna mostra o número de *jobs* de cada instância. A terceira coluna mostra o número de *jobs* diferentes. Por exemplo, na instância IRM01 todos os *jobs* são diferentes um dos outros, já na instância IRM02 há pares de *jobs* completamente compatíveis. Observa-se que, devido a restrição de tempo máximo permitido entre dois *setups*, a menos que $\sum_{i \in J} p_i \leq l$, não pode-se reduzir o número de *jobs* de uma instância pela junção de *jobs* completamente compatíveis em um único *job*. A Figura 4.2 ilustra um exemplo onde uma solução sub-ótima é obtida se *jobs* compatíveis são unidos. Supondo a instância na parte (a) com 8 *jobs*, todos com tempo de processamento \bar{p} , $l = 4\bar{p}$, $h = 3$ e $f_{ij} = 1$ para todos os pares de *jobs* formados por diferentes *jobs* ($f_{ij} = 0$, caso contrário). Pela redução da instância para 3 *jobs*, obtém-se uma solução sub-ótima na parte (b) com custo de 6 devido aos dois *setups*. Entretanto, por produzir placas de mistura na instância

original, obtém-se uma solução ótima na parte (c) com custo de 5 devido ao único *setup*. A quarta coluna mostra o percentual de pares de *jobs* que são permitidos serem processados sequencialmente, devido a variação de largura que não pode ser ajustada sem interromper a produção. Na quinta e sexta colunas reportam-se os resultados da programação atual na prática: o número w de *setups* realizados, e o custo. Nas colunas seguintes são reportados os resultados obtidos com o Modelo M: o número w de *setups* realizados, o custo da solução viável obtida, a redução percentual do custo r com relação à prática da empresa, o tempo computacional TC em segundos, e o *gap* de otimalidade apresentado pelo CPLEX em termos percentuais.

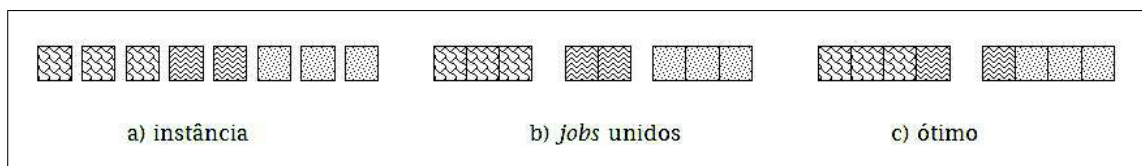


Figura 4.2: Junção de *jobs* pode levar à uma solução sub-ótima

Primeiramente, na Tabela 4.2, nota-se que há muito espaço para melhorias quando considera-se o processo manual da ESB para construir a programação na aciaria. Reduções de custo de mais de 10% são alcançadas em 42 das 48 instâncias. Além disso, significantes reduções de custo de mais de 25% são alcançadas em 36 das 48 instâncias, e reduções maiores que 50% para metade das instâncias testadas. Na maioria das instâncias testadas, soluções obtidas pelo Modelo M indicam mais *setups* que as soluções geradas pela empresa. Isto significa que altos custos de placas de mistura estão sendo gerados na prática. Mesmo assim, grandes reduções de custo foram alcançadas em 13 das 48 instâncias sem aumentar o número de *setups*. De fato, na média, alcançou-se uma redução de custo de 47% nestas instâncias, em 10 delas, reduções de mais de 25% são obtidas. Isto mostra que, de fato, melhorias são possíveis de serem alcançadas pela recombinação da sequência de *jobs*. O potencial para redução de custos não deve ser associada ao percentual de *jobs* permitidos de serem processados sequencialmente (coluna b_{ij}), uma vez que significantes reduções de custo são alcançadas por instâncias com alto ou baixo percentual.

Das 48 instâncias, 19 não foram resolvidas otimamente e após 2 horas de tempo computacional verifica-se que alguns altos *gaps* de otimalidade persistem. Considerando estas instâncias, a média dos *gaps* de otimalidade é de 41,6% com desvio padrão de 27,0%. Considerando apenas as instâncias que foram resolvidas no ótimo, o tempo computacional médio foi de 732,8 segundos com desvio padrão de 1620,6 segundos.

Tabela 4.2: Resultados das instâncias reais com o modelo M

INST.	n	$ndif$	$b_{ij}(\%)$	Prática		Modelo M				
				w	Custo	w	Custo	$r(\%)$	TC(s)	$g(\%)$
IRM01	12	12	56,1	0	6.280	0	6.280	0,0	1	0,0
IRM02	12	6	59,1	0	6.554	0	6.554	0,0	2	0,0
IRM03	12	7	63,6	0	27.708	1	6.306	77,2	6	0,0
IRM04	12	5	66,7	0	27.178	2	12.336	54,6	8	0,0
IRM05	11	4	100,0	0	37.026	2	16.691	54,9	1	0,0
IRM06	11	6	85,5	0	46.500	1	18.464	60,3	2	0,0
IRM07	12	9	100,0	0	50.929	3	18.587	63,5	31	0,0
IRM08	11	8	100,0	0	20.486	0	9.324	54,5	5	0,0
IRM09	12	7	100,0	0	15.260	0	12.711	16,7	1	0,0
IRM10	12	10	100,0	0	6.218	0	162	97,4	3	0,0
IRM11	12	12	63,6	0	13.020	0	9.684	25,6	7	0,0
IRM12	12	7	50,0	0	11.051	1	6.290	43,1	5	0,0
IRM13	20	14	55,3	1	39.262	2	18.438	53,0	7205	22,0
IRM14	24	12	68,1	1	61.018	2	15.368	74,8	7205	65,2
IRM15	22	16	52,4	1	17.392	1	13.331	23,3	7202	29,7
IRM16	21	12	100,0	1	7.714	1	6.192	19,7	7201	98,7
IRM17	18	15	71,2	1	38.152	2	16.635	56,4	95	0,0
IRM18	20	15	41,6	1	55.467	3	23.832	57,0	7204	47,5
IRM19	11	5	100,0	0	4.199	0	1.804	57,0	2	0,0
IRM20	19	13	46,8	1	22.297	1	12.057	45,9	7201	30,8
IRM21	11	6	92,7	0	37.870	2	24.536	35,2	1	0,0
IRM22	23	15	36,8	1	37.669	2	21.746	42,3	7202	38,2
IRM23	17	7	51,5	1	43.209	3	22.874	47,1	203	0,0
IRM24	12	6	59,1	0	6.554	0	6.554	0,0	3	0,0
IRM25	24	17	29,4	1	14.439	1	14.439	0,0	7201	22,5
IRM26	16	6	54,2	2	28.703	3	24.199	15,7	2258	0,0
IRM27	19	15	75,4	2	20.024	2	19.906	0,6	7202	58,4
IRM28	17	13	91,9	1	32.164	1	17.433	45,8	134	0,0
IRM29	21	11	78,1	1	33.956	1	12.438	63,4	7222	96,9
IRM30	18	10	36,6	1	23.662	2	22.533	4,8	544	0,0
IRM31	21	17	46,2	1	72.003	4	29.028	59,7	7205	35,5
IRM32	19	18	38,0	2	29.474	4	25.015	15,1	7201	23,6
IRM33	19	14	60,8	1	33.297	3	22.300	33,0	7201	23,7
IRM34	22	20	50,7	1	54.465	3	26.376	51,6	59	0,0
IRM35	21	19	53,8	1	28.900	2	22.192	23,2	7204	30,8
IRM36	21	19	40,5	1	70.425	4	31.791	54,9	6843	0,0
IRM37	20	18	30,5	1	30.562	3	21.680	29,1	7202	20,7
IRM38	23	22	52,6	1	103.968	5	36.589	64,8	7205	4,6
IRM39	20	19	59,0	1	99.229	5	47.901	51,7	18	0,0
IRM40	24	16	48,6	1	32.051	1	18.716	41,6	7201	24,0
IRM41	24	15	61,2	1	51.986	3	24.399	53,1	7206	32,3
IRM42	20	15	62,6	1	60.527	3	25.507	57,9	382	0,0
IRM43	21	14	50,0	1	20.136	3	14.459	28,2	4097	0,0
IRM44	21	16	42,9	1	92.011	4	38.020	58,7	2860	0,0
IRM45	19	13	49,7	1	43.386	2	23.474	45,9	3414	0,0
IRM46	22	18	39,4	1	41.130	1	16.880	59,0	201	0,0
IRM47	17	12	94,1	1	23.981	1	9.206	61,6	7202	84,8
IRM48	14	12	74,7	1	56.808	3	24.683	56,6	64	0,0

Para impulsionar ainda mais a análise onde reduções de custo podem ser alcançadas se o número de *setups* forem mantidos baixos, por exemplo, em situações de intenso fluxo de material, conduziu-se novos experimentos impondo-se um limite superior para o número de *setups* permitidos. O Modelo M foi rodado restringindo-se o número máximo de *setups* ao número w de *setups* usado na prática. A Tabela 4.3 apresenta os resultados obtidos com a versão restringida do modelo de todas as instâncias para as quais são usados mais *setups* que na prática. Apesar das restrições para o número de *setups*, pela recombinação da sequência de *jobs* é possível evitar placas

de mistura altamente dispendiosas e alcançar significantes reduções de custo. Das 30 instâncias testadas, 19 apresentaram reduções de custo acima de 20% em relação à prática. A redução média dos custos em relação à prática foi de 25,6% com desvio padrão de 19,1%.

Tabela 4.3: Resultados das instâncias reais com o modelo M fixando o número de *setups*

INST.	n	$ndif$	$b_{ij}(\%)$	Prática		Modelo M fixando w				
				w	Custo	w	Custo	$r(\%)$	TC(s)	$g(\%)$
IRM03	12	7	63,6	0	27.708	0	7.755	72,0	1	0,0
IRM04	12	5	66,7	0	27.178	0	26.929	0,9	2	0,0
IRM05	11	4	100,0	0	37.026	0	22.986	37,9	0	0,0
IRM06	11	6	85,5	0	46.500	0	26.748	42,5	1	0,0
IRM07	12	9	100,0	0	50.929	0	32.926	35,3	0	0,0
IRM12	12	7	50,0	0	11.051	0	11.051	0,0	1	0,0
IRM13	20	14	55,3	1	39.262	1	27.488	30,0	7203	38,1
IRM14	24	12	68,1	1	61.018	1	20.089	67,1	7201	65,3
IRM17	18	15	71,2	1	38.152	1	24.956	34,6	3520	0,0
IRM18	20	15	41,6	1	55.467	1	54.399	1,9	7203	33
IRM21	11	6	92,7	0	37.870	0	31.088	17,9	2	0,0
IRM22	23	15	36,8	1	37.669	1	26.507	29,6	7205	10,5
IRM23	17	7	51,5	1	43.209	1	29.169	32,5	101	0,0
IRM26	16	6	54,2	2	28.703	2	27.878	2,9	337	0,0
IRM30	18	10	36,6	1	23.662	1	23.116	2,3	1085	0,0
IRM31	21	17	46,2	1	72.003	1	43.596	39,5	4373	0,0
IRM32	19	18	38,0	2	29.474	2	28.083	4,7	7216	20
IRM33	19	14	60,8	1	33.297	1	31.176	6,4	6939	0,0
IRM34	22	20	50,7	1	54.465	1	42.624	21,7	7204	19,7
IRM35	21	19	53,8	1	28.900	1	25.492	11,8	4372	0,0
IRM36	21	19	40,5	1	70.425	1	54.010	23,3	7205	18,7
IRM37	20	18	30,5	1	30.562	1	27.894	8,7	856	0,0
IRM38	23	22	52,6	1	103.968	1	59.534	42,7	7204	25,6
IRM39	20	19	59,0	1	99.229	1	70.235	29,2	17	0,0
IRM41	24	15	61,2	1	51.986	1	37.831	27,2	7206	51,4
IRM42	20	15	62,6	1	60.527	1	34.803	42,5	1477	0,0
IRM43	21	14	50,0	1	20.136	1	18.780	6,7	7205	21,6
IRM44	21	16	42,9	1	92.011	1	46.783	49,2	3459	0,0
IRM45	19	13	49,7	1	43.386	1	32.694	24,6	1428	0,0
IRM48	14	12	74,7	1	56.808	1	44.267	22,1	255	0,0

Observando-se, ainda, a Tabela 4.3, das 30 instâncias testadas, 10 não foram resolvidas no ótimo, apresentando um *gap* médio de otimalidade de 30,4% com desvio padrão de 16,9%. Para as instâncias que foram resolvidas no ótimo, gastou-se um tempo computacional médio de 1411,3 segundos com desvio padrão 2016,1 segundos.

A Figura 4.3 mostra a solução da instância IRM42 através do Diagrama de Gantt para três situações: (a) solução aplicada pela empresa; (b) solução encontrada pelo Modelo M; e (c) solução encontrada pelo Modelo M fixando-se o número de *setups*. Como pode-se observar na figura, o Modelo M procurou isolar os *jobs* 14 e 20, pois, neste caso, é menos custoso para o modelo realizar a operação de *setup* do que combinar estes *jobs* com os demais. Isto faz com que o tempo para completar o processamento de todos os *jobs* seja maior do que nas situações apresentadas nas letras (a) e (c) da Figura 4.3. Na prática, tal solução seria inviável, pois para a empresa tal situação

é impraticável, mas com este resultado, percebe-se que os *jobs* 14 e 20 não deveriam ter sido incluídos nestas séries. Observando-se as soluções apresentadas nas letras (a) e (c), pode-se observar que somente recombinação dos *jobs* consegue-se uma redução percentual significativa dos custos (42,5%).

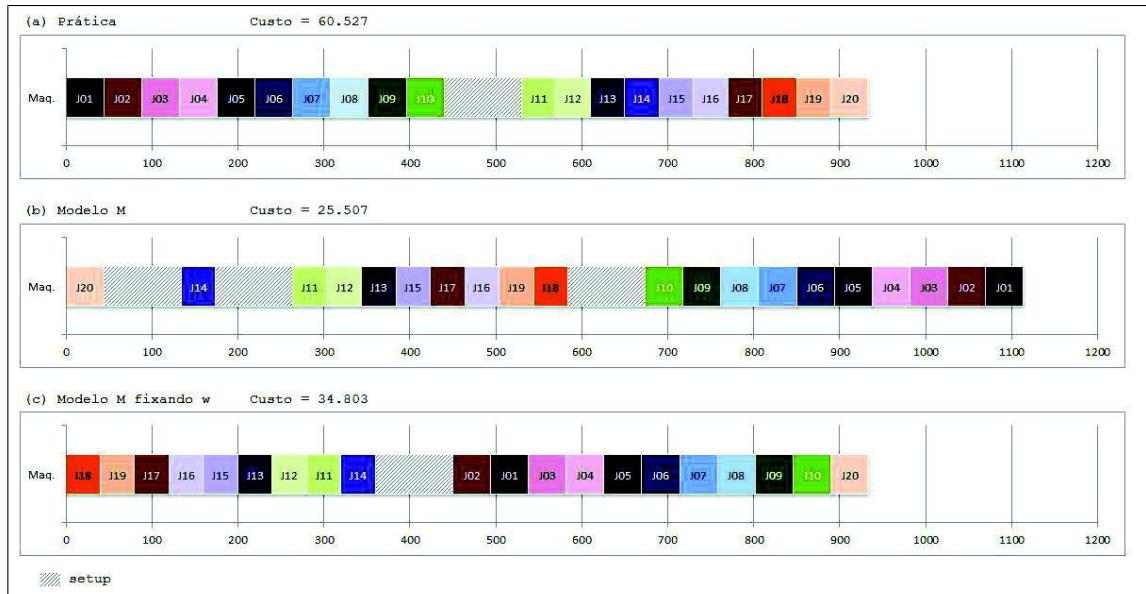


Figura 4.3: Diagrama de Gantt da Instância IRM42

A Tabela 4.4 apresenta os resultados computacionais do Modelo W. Sua estrutura é idêntica à da Tabela 4.2. Assim como na tabela de resultados para o Modelo M, verifica-se que há grande potencial de melhoria na programação manual feita pela ESB. Metade das instâncias testadas apresentou um percentual de melhora em relação à prática em mais de 50%. Cabe nesta situação a mesma análise feita para o Modelo M.

Das 48 instâncias testadas 23 não foram resolvidas no ótimo dentro do prazo de duas horas. Este número foi um pouco superior ao apresentado pelo Modelo M, aonde 19 instâncias não foram resolvidas no ótimo. Considerando apenas estas instâncias, o *gap* médio de otimalidade apresentado pelo CPLEX foi de 43,7% com desvio padrão de 24,2%. Assim como no Modelo M, após duas horas de tempo computacional, altos *gaps* de otimalidade persistem. Considerando apenas as instâncias que foram resolvidas no ótimo, o tempo computacional médio foi de 785,6 segundos com desvio padrão de 1620,3 segundos.

Considerando-se os resultados para ambos os modelos, pode-se observar que existem 28 instâncias que pelo menos um dos modelos não foi capaz de resolver. Considerando apenas estas instâncias, o *gap* médio de otimalidade apresentado pelo CPLEX

Tabela 4.4: Resultados das instâncias reais com o modelo W

INST.	n	$ndif$	$b_{ij}(\%)$	Prática		Modelo W				
				w	Custo	w	Custo	$r(\%)$	TC(s)	$g(\%)$
IRM01	12	12	56,1	0	6.280	0	6.280	0,0	6	0,0
IRM02	12	6	59,1	0	6.554	0	6.554	0,0	1	0,0
IRM03	12	7	63,6	0	27.708	1	6.306	77,2	1	0,0
IRM04	12	5	66,7	0	27.178	2	12.336	54,6	1	0,0
IRM05	11	4	100,0	0	37.026	2	16.691	54,9	0	0,0
IRM06	11	6	85,5	0	46.500	1	18.464	60,3	8	0,0
IRM07	12	9	100,0	0	50.929	3	18.587	63,5	63	0,0
IRM08	11	8	100,0	0	20.486	0	9.324	54,5	3	0,0
IRM09	12	7	100,0	0	15.260	0	12.711	16,7	7	0,0
IRM10	12	10	100,0	0	6.218	0	162	97,4	0	0,0
IRM11	12	12	63,6	0	13.020	0	9.684	25,6	8	0,0
IRM12	12	7	50,0	0	11.051	1	6.290	43,1	1	0,0
IRM13	20	14	55,3	1	39.262	2	18.438	53,0	7201	33,5
IRM14	24	12	68,1	1	61.018	2	15.368	74,8	7204	47,3
IRM15	22	16	52,4	1	17.392	1	13.331	23,3	6205	0,0
IRM16	21	12	100,0	1	7.714	1	6.192	19,7	1090	0,0
IRM17	18	15	71,2	1	38.152	2	16.635	56,4	2747	0,0
IRM18	20	15	41,6	1	55.467	3	23.832	57,0	7201	35,5
IRM19	11	5	100,0	0	4.199	0	1.804	57,0	1	0,0
IRM20	19	13	46,8	1	22.297	1	12.057	45,9	567	0,0
IRM21	11	6	92,7	0	37.870	2	24.536	35,2	6	0,0
IRM22	23	15	36,8	1	37.669	2	21.746	42,3	7201	43,3
IRM23	17	7	51,5	1	43.209	3	22.874	47,1	899	0,0
IRM24	12	6	59,1	0	6.554	0	6.554	0,0	1	0,0
IRM25	24	17	29,4	1	14.439	1	14.439	0,0	7201	14,4
IRM26	16	6	54,2	2	28.703	3	24.199	15,7	54	0,0
IRM27	19	15	75,4	2	20.024	2	19.906	0,6	3291	0,0
IRM28	17	13	91,9	1	32.164	1	17.433	45,8	4380	0,0
IRM29	21	11	78,1	1	33.956	1	12.438	63,4	7201	49,4
IRM30	18	10	36,6	1	23.662	2	22.533	4,8	7200	17,1
IRM31	21	17	46,2	1	72.003	4	29.028	59,7	7201	57,3
IRM32	19	18	38,0	2	29.474	4	25.015	15,1	7202	50,0
IRM33	19	14	60,8	1	33.297	3	22.300	33,0	7200	29,2
IRM34	22	20	50,7	1	54.465	3	26.376	51,6	7201	64,5
IRM35	21	19	53,8	1	28.900	2	22.377	22,6	7202	44,9
IRM36	21	19	40,5	1	70.425	4	31.791	54,9	7201	72,6
IRM37	20	18	30,5	1	30.562	3	21.680	29,1	7201	42,8
IRM38	23	22	52,6	1	103.968	5	36.589	64,8	7203	66,4
IRM39	20	19	59,0	1	99.229	5	47.901	51,7	7201	62,3
IRM40	24	16	48,6	1	32.051	1	18.716	41,6	7201	49,3
IRM41	24	15	61,2	1	51.986	3	24.414	53,0	7201	74,0
IRM42	20	15	62,6	1	60.527	3	25.507	57,9	7200	27,0
IRM43	21	14	50,0	1	20.136	2	14.459	28,2	7202	13,4
IRM44	21	16	42,9	1	92.011	4	38.020	58,7	7201	64,1
IRM45	19	13	49,7	1	43.386	2	23.474	45,9	7201	20,0
IRM46	22	18	39,4	1	41.130	1	16.880	59,0	7201	27,9
IRM47	17	12	94,1	1	23.981	1	9.206	61,6	181	0,0
IRM48	14	12	74,7	1	56.808	3	24.683	56,6	118	0,0

foi de 28,2% e 35,9% para os modelos M e W, respectivamente.

Na resolução das instâncias pelo modelo I, cada unidade de tempo t corresponde a 2 minutos, pois todos os parâmetros de entrada eram múltiplos de 2. Não apresentam-se os resultados para o modelo I, pois este modelo não foi capaz de gerar soluções inteiras viáveis para quase todas as instâncias dentro do tempo estipulado. Além disso, a relaxação linear destes problemas não foi tão boa quanto a esperada, apresentando um *gap* médio de 92,1% e um desvio-padrão de 12,1%. O tempo computacional para obtenção da solução de relaxação linear também foi relativamente alto

(média de 1882 segundos e desvio-padrão de 3899,5 segundos).

A Tabela 4.5 apresenta os resultados do modelo P para as instâncias de 11 a 24 *jobs*. A estrutura desta tabela é idêntica às tabelas anteriores. O percentual de melhora é similar aos demais modelos, percentual médio de redução do custo em relação à prática de 42,4%, com desvio padrão de 23,0%. Diferentemente dos outros três modelos, todas as instâncias foram resolvidas no ótimo dentro do prazo de duas horas de tempo computacional. O tempo computacional médio para rodar as instâncias foi de 1,6 segundos com desvio padrão de 1,2 segundos.

Tabela 4.5: Resultados das instâncias reais menores com o modelo P

INST.	n	$ndif$	b_{ij} (%)	Prática		Modelo P				
				w	Custo	w	Custo	r (%)	TC(s)	g (%)
IRM01	12	12	56,06	0	6.280	0	6.280	0,0	2	0,0
IRM02	12	6	59,09	0	6.554	0	6.554	0,0	2	0,0
IRM03	12	7	63,64	0	27.708	1	6.306	77,2	2	0,0
IRM04	12	5	66,67	0	27.178	2	12.336	54,6	1	0,0
IRM05	11	4	100	0	37.026	2	16.691	54,9	1	0,0
IRM06	11	6	85,45	0	46.500	1	18.464	60,3	2	0,0
IRM07	12	9	100	0	50.929	3	18.587	63,5	1	0,0
IRM08	11	8	100	0	20.486	0	9.324	54,5	0	0,0
IRM09	12	7	100	0	15.260	0	12.711	16,7	1	0,0
IRM10	12	10	100	0	6.218	0	162	97,4	0	0,0
IRM11	12	12	63,64	0	13.020	0	9.684	25,6	1	0,0
IRM12	12	7	50	0	11.051	1	6.290	43,1	1	0,0
IRM13	20	14	55,26	1	39.262	2	18.438	53,0	2	0,0
IRM14	24	12	68,12	1	61.018	2	15.368	74,8	1	0,0
IRM15	22	16	52,38	1	17.392	1	13.331	23,3	2	0,0
IRM16	21	12	100	1	7.714	1	6.192	19,7	1	0,0
IRM17	18	15	71,24	1	38.152	2	16.635	56,4	4	0,0
IRM18	20	15	41,58	1	55.467	3	23.832	57,0	4	0,0
IRM19	11	5	100	0	4.199	0	1.804	57,0	0	0,0
IRM20	19	13	46,78	1	22.297	1	12.057	45,9	1	0,0
IRM21	11	6	92,73	0	37.870	2	24.536	35,2	0	0,0
IRM22	23	15	36,76	1	37.669	2	21.746	42,3	1	0,0
IRM23	17	7	51,47	1	43.209	3	22.874	47,1	2	0,0
IRM24	12	6	59,09	0	6.554	0	6.554	0,0	2	0,0
IRM25	24	17	29,35	1	14.439	1	14.439	0,0	3	0,0
IRM26	16	6	54,17	2	28.703	3	24.199	15,7	1	0,0
IRM27	19	15	75,44	2	20.024	2	19.906	0,6	2	0,0
IRM28	17	13	91,91	1	32.164	1	17.433	45,8	4	0,0
IRM29	21	11	78,1	1	33.956	1	12.438	63,4	1	0,0
IRM30	18	10	36,6	1	23.662	2	22.533	4,8	1	0,0
IRM31	21	17	46,19	1	72.003	4	29.028	59,7	2	0,0
IRM32	19	18	38,01	2	29.474	4	25.015	15,1	0	0,0
IRM33	19	14	60,82	1	33.297	3	22.300	33,0	2	0,0
IRM34	22	20	50,65	1	54.465	3	26.376	51,6	1	0,0
IRM35	21	19	53,81	1	28.900	2	22.192	23,2	1	0,0
IRM36	21	19	40,48	1	70.425	4	31.791	54,9	3	0,0
IRM37	20	18	30,53	1	30.562	3	21.680	29,1	1	0,0
IRM38	23	22	52,57	1	103.968	5	36.589	64,8	1	0,0
IRM39	20	19	58,95	1	99.229	5	47.901	51,7	5	0,0
IRM40	24	16	48,55	1	32.051	1	18.716	41,6	1	0,0
IRM41	24	15	61,23	1	51.986	3	24.399	53,1	4	0,0
IRM42	20	15	62,63	1	60.527	3	25.507	57,9	1	0,0
IRM43	21	14	50	1	20.136	2	14.459	28,2	3	0,0
IRM44	21	16	42,86	1	92.011	4	38.020	58,7	1	0,0
IRM45	19	13	49,71	1	43.386	2	23.474	45,9	2	0,0
IRM46	22	18	39,39	1	41.130	1	16.880	59,0	1	0,0
IRM47	17	12	94,12	1	23.981	1	9.206	61,6	1	0,0
IRM48	14	12	74,73	1	56.808	3	24.683	56,6	1	0,0

Baseado nos resultados anteriores, resolveu-se realizar experimentos com o Modelo P para as instâncias de 54 a 72 corridas, que correspondem a dois dias e meio de produção. Os resultados destes experimentos são apresentados na Tabela 4.6. Novamente, observa-se um grande potencial para melhoria na programação manual feita pela empresa. Das 48 instâncias testadas, 47 apresentaram percentual de reduções de custos em relação à prática acima de 10%. Além disso, 38 instâncias testadas (79,2% das instâncias) apresentaram reduções acima de 25%. O percentual de redução médio dos custos em relação à prática foi de 39,7% com desvio padrão de 15,1%. Outro ponto interessante que cabe destacar é que em 4 das 48 instâncias testadas houve uma redução do número de *setups*. Já na maioria das instâncias (aproximadamente 70% delas), o número de *setups* foi superior ao número usado na prática.

Apenas 3 instâncias não foram resolvidas no ótimo, e além disso, apresentaram *gaps* de otimalidade bem pequenos, mínimo de 1,3% e máximo de 4,3%. Levando-se em consideração apenas as instâncias que foram resolvidas otimamente, o tempo computacional médio para resolver estas instâncias foi de 106,4 segundos com desvio padrão de 447,1 segundos.

Entre os quatro modelos testados, o modelo P foi o modelo que apresentou o melhor desempenho. Um dos fatores que contribuiu para este melhor desempenho do modelo P pode ser evidenciado na Tabela 4.7. Nesta tabela, pode-se verificar que o número de restrições do modelo P é bem inferior ao número de restrições dos modelos M e W. Além disso, o número de variáveis inteiras do modelo P é quase igual ao número de variáveis inteiras do modelo W, mas bem inferior ao número de variáveis inteiras do modelo M. Outro fator importante para o bom desempenho do modelo P é que não houve necessidade de linearizar restrições do modelo, como foi feito no modelo M.

4.3 Heurísticas para o PSP

Nesta seção apresentam-se duas heurísticas para resolução do problema de sequenciamento de painelas no setor de lingotamento contínuo. Resolveu-se criar estas heurísticas, porque o modelo P, para os problemas maiores, não conseguiu resolver otimamente 3 instâncias e apresentou tempos computacionais altos para algumas instâncias.

As heurísticas desenvolvidas fazem uma analogia entre o PSP e o problema de árvore geradora mínima capacitada (CMST - *Capacited Minimum Spanning Tree*). O

Tabela 4.6: Resultados das instâncias reais maiores com o modelo P

INST.	n	ndif	b_{ij} (%)	Prática		Modelo P				
				w	Custo	w	Custo	r (%)	TC(s)	g (%)
IR01	56	37	33,8	5	68.254	5	44.079	35,4	18	0,0
IR02	56	38	43,8	4	75.018	4	27.771	63,0	6	0,0
IR03	67	38	36,1	4	31.707	4	28.283	10,8	7203	1,3
IR04	57	34	47,3	4	98.422	5	54.950	44,2	20	0,0
IR05	54	26	30,5	3	39.647	4	31.495	20,6	6	0,0
IR06	57	37	49,1	4	72.525	6	48.115	33,7	22	0,0
IR07	66	33	49,7	4	66.043	6	46.163	30,1	15	0,0
IR08	65	44	38,3	5	94.119	7	66.019	29,9	79	0,0
IR09	65	39	39,9	5	76.538	6	48.967	36,0	10	0,0
IR10	65	28	32,3	7	78.880	7	64.378	18,4	7	0,0
IR11	59	44	55,0	5	39.444	4	32.738	17,0	830	0,0
IR12	65	42	53,9	5	104.264	5	55.174	47,1	27	0,0
IR13	63	34	31,9	4	58.885	6	40.895	30,6	5	0,0
IR14	60	34	42,0	5	87.026	6	42.270	51,4	6	0,0
IR15	68	51	41,7	5	65.315	7	43.793	33,0	10	0,0
IR16	62	45	47,5	5	58.844	5	54.917	6,7	40	0,0
IR17	59	37	36,5	6	169.641	9	64.401	62,0	3	0,0
IR18	56	41	46,0	5	69.196	6	37.413	45,9	6	0,0
IR19	66	53	45,6	6	89.619	8	53.233	40,6	14	0,0
IR20	68	42	38,5	6	110.386	8	66.899	39,4	10	0,0
IR21	66	45	27,7	5	102.335	6	48.965	52,2	154	0,0
IR22	66	46	29,8	5	85.356	7	62.171	27,2	111	0,0
IR23	65	58	35,4	5	228.973	9	99.339	56,6	6	0,0
IR24	63	49	33,3	5	73.715	7	51.398	30,3	4	0,0
IR25	59	44	34,6	4	194.953	11	89.718	54,0	7	0,0
IR26	59	40	25,0	5	144.340	8	89.750	37,8	3	0,0
IR27	63	36	45,9	5	74.543	5	42.117	43,5	10	0,0
IR28	58	28	35,9	5	117.023	6	48.752	58,3	6	0,0
IR29	64	35	40,1	4	75.232	5	39.370	47,7	7205	1,6
IR30	72	39	51,1	5	132.626	7	48.621	63,3	112	0,0
IR31	67	44	39,3	6	67.331	8	51.732	23,2	38	0,0
IR32	70	43	27,9	6	183.176	12	91.316	50,1	5	0,0
IR33	63	32	32,8	5	156.766	7	81.093	48,3	4	0,0
IR34	55	23	55,0	5	54.315	6	48.076	11,5	5	0,0
IR35	64	32	64,0	6	82.450	5	50.621	38,6	22	0,0
IR36	64	41	35,7	5	117.992	6	57.051	51,6	10	0,0
IR37	68	49	30,5	6	118.610	8	62.833	47,0	6	0,0
IR38	60	31	28,9	7	117.113	6	61.039	47,9	3	0,0
IR39	68	48	37,4	6	164.466	8	83.750	49,1	6	0,0
IR40	66	36	46,0	5	162.261	9	89.378	44,9	164	0,0
IR41	63	46	36,6	5	79.541	7	60.228	24,3	5	0,0
IR42	66	49	49,0	7	77.821	5	45.076	42,1	7202	4,3
IR43	66	31	31,3	5	97.360	5	42.716	56,1	14	0,0
IR44	61	39	32,7	5	138.622	6	50.131	63,8	7	0,0
IR45	61	35	70,5	4	70.412	4	32.634	53,7	19	0,0
IR46	60	34	48,1	5	53.000	7	45.384	14,4	8	0,0
IR47	54	26	54,0	3	47.522	3	36.930	22,3	2920	0,0
IR48	60	47	32,2	5	99.794	6	50.889	49,0	4	0,0

Tabela 4.7: Comparativo do Número de Variáveis e do Número de Restrições dos Modelos M, W e P

n	Modelo M			Modelo W			Modelo P		
	NTV	NVI	NR	NTV	NVI	NR	NTV	NVI	NR
12	473	314	817	299	155	1673	312	156	206
16	821	546	1409	527	271	3991	544	272	338
20	1265	842	2161	819	419	7829	840	420	502
24	1805	1202	3073	1175	599	13571	1200	600	698

NTV = Número Total de Variáveis;

NVI = Número de Variáveis Inteiras;

NR = Número total de restrições.

problema de árvore geradora mínima capacitada é definido da forma como se segue. Seja $G = (N, E)$ um grafo não direcionado, onde N é o conjunto de nós e E é o conjunto de arestas. Custos c_{ij} e pesos b_i não-negativos estão associados respectivamente a cada arco $(i, j) \in E$ e a cada nó $i \in N$. Dado um valor inteiro Q e um nó central $r \in N$, o problema CSMT consiste em encontrar uma árvore geradora de custo mínimo T de G tal que o somatório dos pesos dos nós de cada componente conexo gerado em T por $V - r$ é menor ou igual a Q (para mais detalhes ver Amberg *et al.* [1] e Martins *et al.* [32]).

A analogia foi feita por causa de algumas semelhanças entre os dois problemas. O PSP busca definir as séries que geram menor custo e normalmente isto acontece quando painéis contendo peças de composição química semelhante são agrupadas dentro de uma mesma série. Baseado neste contexto é que originou-se a ideia de usar o problema de CMST para gerar as séries. O problema de CSMT agrupa as painéis com maior aptidão dentro de uma mesma série, e depois a sequência das painéis dentro da série é definida usando-se um dos modelos apresentados na seção anterior. Como o modelo P faz uma analogia entre o PSP e o PRVAC, tentou-se também utilizar a heurística de economias de Clark & Wright para resolver o PSP, mas os resultados não foram bons, por isso estes resultados não são apresentados neste trabalho.

Para fazer a analogia entre os dois problemas, considerou-se que cada *job* representa um nó da rede. O custo de ligação entre dois nós da rede ($CA[i, j]$) é igual ao custo da placa de mistura formada quando o *job* j é processado imediatamente após o *job* i (f_{ij}) e o custo de ligação dos nós ao nó central é igual ao custo de realização de um *setup* (h). O peso $wt(i)$ de cada nó da árvore corresponde ao tempo necessário para processar o *job* que gerou aquele nó (p_j) e a capacidade máxima Q de cada sub-árvore gerada corresponde ao tempo máximo entre dois *setups* consecutivos (l).

As heurísticas são baseadas na heurística de Esau-Williams (ver Esau e Williams [16]), utilizada para resolver o problema de árvore geradora mínima capacitada, e utiliza o Modelo P para gerar a sequência de painéis a serem processadas na MLC.

A segunda heurística busca melhorar os resultados obtidos pela primeira heurística. Para isto, a heurística utiliza um procedimento, análogo ao processo de geração de uma solução inicial do método GRASP, para escolher a combinação que será feita a cada iteração da heurística de Esau-Williams. Este processo é repetido por certo número de iterações.

As duas heurísticas são testadas com as instâncias reais apresentadas na seção

4.2. Os resultados são comparados com a prática da empresa e com as soluções geradas pelo Modelo P.

4.3.1 Heurística de Esau-Williams para o PSP - HEW

A heurística de Esau-Williams (EW) é um procedimento de economia. Os procedimentos de economia para o CSMT usualmente começam com a árvore estrela, ou seja, todos os nós conectados ao nó central. Neste caso, a melhor troca viável, isto é, a troca que produz a maior economia, é realizada. Este processo é repetido até que nenhuma economia possa ser obtida. Estes métodos também podem ser classificados como procedimentos de melhora (AMBERG *et al.* [1]).

Segundo Amberg *et al.* [1], a heurística EW procura juntar dois componentes (sub-árvores) que produzem a maior economia em termos de custo. A economia s_{ij} de juntar dois componentes C_i e C_j é definida através da equação (4.63).

$$s_{ij} = \begin{cases} \max\{\chi_i, \chi_j\} - c_{ij}^*, & \text{se a junção dos componentes } C_i \text{ e } C_j \text{ é viável} \\ -\infty, & \text{caso contrário} \end{cases} \quad (4.63)$$

Na Equação (4.63), χ_i representa o custo mínimo de conexão do nó central aos nós de C_i e c_{ij}^* é o custo mínimo de um arco conectando C_i e C_j . Então todas as economias relativas ao novo componente devem ser recomputadas e novamente, a maior economia deve ser escolhida. O processo é finalizado se nenhuma economia positiva é avaliada.

Após a execução do algoritmo, cada sub-árvore gerada representa uma série. Para determinar o sequenciamento dos *jobs* dentro desta série, aplicou-se a cada sub-árvore gerada o Modelo P apresentado na seção 4.1. A necessidade da aplicação do modelo à cada sub-árvore gerada se deve ao fato de que cada sub-árvore gerada pela heurística EW não representa uma sequência de *jobs* a serem processados. Permitiu-se ao modelo, caso necessário, a introdução de novos *setups* à solução do problema.

A heurística EW proposta foi implementada na linguagem AMPL. Os testes foram feitos com as 96 instâncias baseadas no caso real apresentadas na seção 4.2.1. Utilizou-se as 96 instâncias para poder avaliar o desempenho das heurísticas. O computador usado para os testes é o mesmo usado para resolução dos modelos matemáticos. O modelo matemático foi resolvido pelo pacote de otimização CPLEX 12.6, utilizando-

se a linguagem AMPL e a configuração padrão do mesmo. Por ter apresentados melhores resultados, utilizou-se o Modelo P para a obtenção das sequências de painéis. Estipulou-se um tempo limite de 3600 segundos para a resolução dos modelos.

A Tabela 4.8 apresenta os resultados computacionais para as instâncias reais menores (11 a 24 *jobs*). A estrutura desta tabela é idêntica às tabelas apresentadas na Seção 4.2.2. Inclui-se apenas a coluna $d(\%)$ que representa a diferença percentual entre a solução obtida com o Modelo P e a solução obtida com heurística EW, calculada pela equação (4.64). Nesta equação, $Custo$ representa o custo total obtido com a resolução da instância e MR representa a solução obtida pela resolução da instância pelo modelo P.

$$d(\%) = \frac{Custo - MR}{MR} \quad (4.64)$$

Na Tabela 4.8, observa-se que 46 das 48 instâncias foram resolvidas no ótimo com a HEW. Além disso, a diferença percentual entre a solução ótima e a solução da HEW para as duas instâncias que não foram resolvidas otimamente é pequena (0,2% e 4,1%). O tempo computacional para a resolução de todas as instâncias foi inferior ou igual a 1 segundo.

A Tabela 4.9 apresenta os resultados computacionais para as instâncias reais maiores (54 a 72 *jobs*). A estrutura desta tabela é idêntica à Tabela 4.8.

Nesta tabela, observa-se que 26 das 48 instâncias foram resolvidas no ótimo com a aplicação da HEW. Observando-se as demais 22 instâncias e considerando apenas as instâncias que foram resolvidas otimamente pelo Modelo P (19 instâncias), o desvio percentual médio entre a solução ótima obtida pelo modelo P e a solução da heurística EW foi de 5,0% com um desvio padrão de 5,95%. Destas 19 instâncias, 14 apresentaram um desvio percentual abaixo de 10%. A HEW não conseguiu melhorar a solução da empresa somente na instância IR03, apresentando uma diferença percentual entre a solução da empresa e a solução da heurística de 8,2%.

O tempo computacional médio para a resolução das instâncias foi de 1,21 segundos com desvio padrão de 0,65 segundos. A instância que foi resolvida com maior tempo computacional foi a instância IR30, onde gastou-se 3 segundos.

Tabela 4.8: Resultados das instâncias reais menores com a Heurística EW

INST.	n	modelo P			Heurística EW			
		w	custo	TC(s)	w	Custo	d(%)	TC(s)
IRM01	12	0	6.280	2	0	6.280	0,0	0
IRM02	12	0	6.554	2	0	6.554	0,0	1
IRM03	12	1	6.306	2	1	6.306	0,0	0
IRM04	12	2	12.336	1	2	12.336	0,0	0
IRM05	11	2	16.691	1	1	16.691	0,0	0
IRM06	11	1	18.464	2	1	18.464	0,0	0
IRM07	12	3	18.587	1	3	18.587	0,0	0
IRM08	11	0	9.324	0	0	9.324	0,0	0
IRM09	12	0	12.711	1	0	12.711	0,0	1
IRM10	12	0	162	0	0	162	0,0	0
IRM11	12	0	9.684	1	0	9.684	0,0	0
IRM12	12	1	6.290	1	1	6.290	0,0	0
IRM13	20	2	18.438	2	2	18.438	0,0	0
IRM14	24	2	15.368	1	2	15.368	0,0	0
IRM15	22	1	13.331	2	1	13.331	0,0	1
IRM16	21	1	6.192	1	1	6.192	0,0	0
IRM17	18	2	16.635	4	2	16.635	0,0	0
IRM18	20	3	23.832	4	3	23.832	0,0	1
IRM19	11	0	1.804	0	0	1.804	0,0	0
IRM20	19	1	12.057	1	1	12.057	0,0	1
IRM21	11	2	24.536	0	2	24.536	0,0	0
IRM22	23	2	21.746	1	2	21.746	0,0	0
IRM23	17	3	22.874	2	2	22.874	0,0	1
IRM24	12	0	6.554	2	0	6.554	0,0	0
IRM25	24	1	14.439	3	1	14.439	0,0	0
IRM26	16	3	24.199	1	3	24.199	0,0	1
IRM27	19	2	19.906	2	2	19.906	0,0	0
IRM28	17	1	17.433	4	1	17.433	0,0	0
IRM29	21	1	12.438	1	2	12.469	0,2	0
IRM30	18	2	22.533	1	2	22.533	0,0	1
IRM31	21	4	29.028	2	3	29.028	0,0	0
IRM32	19	4	25.015	0	4	25.015	0,0	0
IRM33	19	3	22.300	2	2	22.300	0,0	1
IRM34	22	3	26.376	1	3	26.376	0,0	0
IRM35	21	2	22.192	1	2	22.192	0,0	0
IRM36	21	4	31.791	3	4	31.791	0,0	0
IRM37	20	3	21.680	1	3	21.680	0,0	1
IRM38	23	5	36.589	1	5	36.589	0,0	0
IRM39	20	5	47.901	5	5	47.901	0,0	0
IRM40	24	1	18.716	1	1	18.716	0,0	0
IRM41	24	3	24.399	4	3	24.399	0,0	1
IRM42	20	3	25.507	1	3	25.507	0,0	0
IRM43	21	2	14.459	3	1	14.459	0,0	0
IRM44	21	4	38.020	1	4	38.020	0,0	0
IRM45	19	2	23.474	2	2	23.474	0,0	0
IRM46	22	1	16.880	1	1	16.880	0,0	0
IRM47	17	1	9.206	1	1	9.583	4,1	1
IRM48	14	3	24.683	1	3	24.683	0,0	0

4.3.2 Heurística de Esau-Williams com Múltiplos Reinícios

Com a finalidade de melhorar a qualidade das soluções geradas pela HEW, propôs-se uma modificação nesta heurística de tal forma que ela fosse executada certo número de vezes e a cada iteração uma nova solução fosse gerada. Para que fossem geradas soluções diferentes a cada iteração, o processo de escolha da combinação de maior economia utiliza uma estrutura muito semelhante ao processo de construção de uma solução inicial da meta-heurística GRASP. Este procedimento é descrito na Figura 4.4.

Tabela 4.9: Resultados das instâncias reais maiores com a Heurística EW

INST.	n	modelo P			Heurística EW			
		w	custo	TC(s)	w	Custo	d(%)	TC(s)
IR01	56	5	44.079	18	5	44.079	0,0	1
IR02	56	4	27.771	6	4	27.771	0,0	2
IR03	67	4	28.283	7203	5	34.292	21,2	2
IR04	57	5	54.950	20	6	55.772	1,5	2
IR05	54	4	31.495	6	4	31.495	0,0	0
IR06	57	6	48.115	22	6	48.115	0,0	2
IR07	66	6	46.163	15	6	46.163	0,0	1
IR08	65	7	66.019	79	7	66.069	0,1	1
IR09	65	6	48.967	10	6	48.967	0,0	1
IR10	65	7	64.378	7	7	64.378	0,0	1
IR11	59	4	32.738	830	5	38.785	18,5	2
IR12	65	5	55.174	27	6	56.034	1,6	1
IR13	63	6	40.895	5	6	40.895	0,0	2
IR14	60	6	42.270	6	7	48.252	14,2	1
IR15	68	7	43.793	10	7	43.793	0,0	1
IR16	62	5	54.917	40	5	55.057	0,3	1
IR17	59	9	64.401	3	7	64.401	0,0	1
IR18	56	6	37.413	6	6	37.477	0,2	2
IR19	66	8	53.233	14	8	53.233	0,0	2
IR20	68	8	66.899	10	8	67.097	0,3	1
IR21	66	6	48.965	154	7	54.989	12,3	1
IR22	66	7	62.171	111	7	62.171	0,0	0
IR23	65	9	99.339	6	10	102.025	2,7	2
IR24	63	7	51.398	4	7	51.398	0,0	1
IR25	59	11	89.718	7	11	89.718	0,0	0
IR26	59	8	89.750	3	8	89.750	0,0	1
IR27	63	5	42.117	10	5	42.307	0,5	2
IR28	58	6	48.752	6	6	48.752	0,0	1
IR29	64	5	39.370	7205	6	41.718	6,0	1
IR30	72	7	48.621	112	8	54.650	12,4	3
IR31	67	8	51.732	38	7	51.732	0,0	1
IR32	70	12	91.316	5	12	91.316	0,0	1
IR33	63	7	81.093	4	6	81.093	0,0	1
IR34	55	6	48.076	5	6	48.076	0,0	0
IR35	64	5	50.621	22	5	50.621	0,0	1
IR36	64	6	57.051	10	7	59.252	3,9	1
IR37	68	8	62.833	6	8	63.210	0,6	2
IR38	60	6	61.039	3	6	61.039	0,0	0
IR39	68	8	83.750	6	9	85.367	1,9	2
IR40	66	9	89.378	164	9	92.325	3,3	1
IR41	63	7	60.228	5	7	60.228	0,0	1
IR42	66	5	45.076	7202	5	45.091	0,0	1
IR43	66	5	42.716	14	5	42.716	0,0	2
IR44	61	6	50.131	7	7	56.460	12,6	1
IR45	61	4	32.634	19	4	32.634	0,0	1
IR46	60	7	45.384	8	7	45.394	0,0	1
IR47	54	3	36.930	2920	4	39.669	7,4	1
IR48	60	6	50.889	4	6	50.889	0,0	1

* Instâncias que não foram resolvidas no ótimo

Na linha 1 do procedimento apresentado na Figura 4.4 inicia-se a lista de candidatos. Considera-se que nesta lista encontram-se todas as combinações possíveis. Nas linhas 2 e 3 procura-se determinar, respectivamente, a menor economia e a maior economia possível gerada por alguma combinação. À partir das combinações pertencentes à lista de candidatos (LC), selecionam-se aquelas cuja combinação satisfaça a condição imposta na linha 4 para formar a lista restrita de candidatos (LRC). Na condição apresentada na linha 4, $\alpha \in \{0, 1\}$. Dentre as combinações pertencentes a LRC, seleciona-se aleatoriamente uma combinação e faz-se esta combinação.

```

procedimento Determinar_Maior_Economia()
1  Inicialize o conjunto  $LC$  de candidatos;
2   $E_{min} = \min_{(i,j) \in LC} \{s_{ij}\}$ ;
3   $E_{max} = \max_{(i,j) \in LC} \{s_{ij}\}$ ;
4   $LRC = \{(i,j) \in LC \mid s_{ij} \leq E_{min} + \alpha(E_{max} - E_{min})\}$ ;
5  Selecione, aleatoriamente, uma combinação  $(i,j) \in LRC$ ;
fim Determinar_Maior_Economia();

```

Figura 4.4: Procedimento para a Escolha da Maior Economia

A heurística de Esau-Williams com Múltiplos Reinícios (HEWMR) para o problema de sequenciamento de painéis (PSP) é apresentada na Figura 4.5. Na linha 1 do código apresentado nesta figura, inicia-se o processo iterativo que é executado $iterMAX$ vezes, onde $iterMAX$ corresponde ao número máximo de iterações. Na linha 2 gera-se uma solução para o PSP com a heurística EW modificada que inclui o procedimento descrito na Figura 4.4 para determinar a maior economia. Se o custo da solução gerada for maior que o custo da melhor solução gerada até então, faz-se $f(S^*)$ receber o valor de $f(S)$ na linha 4 e faz-se com que a solução gerada S seja a melhor solução encontrada até o momento (S^*) na linha 5. Na linha 8, retorna-se, então, a melhor solução encontrada dentro do número máximo de iterações.

```

Procedimento HEWMR()
1  para  $i = 1$  até  $iterMAX$  faça
2       $S \leftarrow HEW\_modificado()$ ;
3      se  $(f(S) < f(S^*))$  então
4           $S^* \leftarrow S$ ;
5           $f(S^*) \leftarrow f(S)$ ;
6      fim-se;
7  fim-para;
8  Retorne  $S^*$ 
fim-Procedimento

```

Figura 4.5: Heurística EW com múltiplos reinícios para o PSP

A seguir, na Seção 4.3.2.1 é apresentado como determinou-se os valores dos parâmetros α e $iterMAX$ da HEWMR e na Seção 4.3.2.2 são apresentados os resultados obtidos com a HEWMR para as instâncias reais maiores que apresentaram diferença percentual acima de 0% entre a solução encontrada pelo Modelo P e a solução gerada pela HEW.

4.3.2.1 Determinação dos Parâmetros da HEWMR

Para determinar os valores dos parâmetros α e $iterMAX$ da HEWMR, criou-se um conjunto de 16 instâncias geradas artificialmente. Gerou-se 8 instâncias com 50 *jobs* e 8 instâncias com 100 *jobs*. Para gerar estas instâncias foram selecionados 148 tipos de aço entre os 230 existentes. Escolheu-se aqueles com maior expressão, ou seja, os tipos de aço que apareciam com maior frequência no relatório fornecido pela ESB. A Tabela 4.10 apresenta algumas das informações utilizadas para gerar as instâncias artificiais. A coluna “Inst.” desta tabela apresenta o nome das instâncias, a coluna “nJ” o número de *jobs* da instância, a coluna “nTA” o número de tipos de aço existentes na instância e a coluna “Larg. (mm)” a distribuição de probabilidade usada para gerar as larguras desejadas das placas na instância. Para gerar os tipos de aço, considerou-se o relatório cedido pela ESB, desta forma aquele tipo de aço mais frequente tinha uma maior probabilidade de ser escolhido, por exemplo, se o tipo de aço 1 aparecesse no relatório em 20% das placas processadas, este teria 20% de chance de ser sorteado. Os cálculos dos custos de mistura e *setup* foram feitos conforme descrito na Seção 4.2.

Tabela 4.10: Tabela auxiliar usada para geração de instâncias artificiais

Inst.	nJ	nTA	Larg. (mm)
IA05001	50	1	U(850, 1850)
IA05002	50	1	U(1275, 1425)
IA05003	50	3	U(850, 1850)
IA05004	50	3	U(1275, 1425)
IA05005	50	5	U(850, 1850)
IA05006	50	5	U(1275, 1425)
IA05007	50	50	U(850, 1850)
IA05008	50	50	U(1275, 1425)
IA10001	100	1	U(850, 1850)
IA10002	100	1	U(1275, 1425)
IA10003	100	5	U(850, 1850)
IA10004	100	5	U(1275, 1425)
IA10005	100	10	U(850, 1850)
IA10006	100	10	U(1275, 1425)
IA10007	100	100	U(850, 1850)
IA10008	100	100	U(1275, 1425)

Inicialmente, na determinação dos valores dos parâmetros da HEWMR, resolveu-se as instâncias com a HEW descrita na seção 4.3.1 e com o Modelo P. Fez-se isto para determinar em quais instâncias seria interessante aplicar a HEWMR, pois não valeria a pena aplicar esta heurística em instâncias que a solução ótima tivesse sido encontrada pela HEW. Desta forma, utilizou-se, então, as instâncias IA05001, IA05008,

IA10003, IA10005, IA10007 e IA10008. Os valores considerados para *iterMAX* foram: 10, 15, 20, 25, 30, 40 e 50. Já os valores testados para α foram: 0,9; 0,8; 0,7; 0,6; 0,5 e 0,4. Rodou-se a heurística 20 vezes com cada uma destas instâncias e com cada combinação possível dos valores de *iterMAX* e α .

Para determinar o valor de α calculou-se o percentual médio de melhora para todas as instâncias consideradas. Este percentual médio era calculado individualmente para cada instância e em seguida calculava-se a média para todas as instâncias. O percentual de melhora era calculado a partir da Equação (4.65), onde $f(S^*)$ é o custo da melhor solução encontrada pelo modelo P e $f(S)$ é o custo encontrado pela solução da HEWMR.

$$\%Melhora = \frac{f(S^*) - f(S)}{f(S^*)} \times 100 \quad (4.65)$$

Os percentuais médios de melhora para cada valor de α são apresentados na Figura 4.6. Observa-se nesta figura que o melhor percentual médio de melhora aconteceu quando $\alpha = 0,5$, portando adotou-se este valor para α .



Figura 4.6: Gráfico para determinação do valor de α

Determinado o valor de α , o próximo passo era determinar o valor de *iterMAX*. Usando-se o valor de $\alpha = 0,5$ e utilizando um valor inicial *iterMAX* = 50, determinou-

se, inicialmente, em qual iteração a melhor solução era encontrada. A heurística HEWMR foi executada vinte vezes para as seis instâncias consideradas. Dentro das 120 execuções determinou-se quantas vezes a melhor solução era encontrada em uma dada iteração. A partir destes valores, construiu-se o gráfico de percentual acumulado apresentado na Figura 4.7. Analisando-se o gráfico, resolveu-se adotar o valor de $iterMAX = 40$, pois a HEWMR em 82,5% das execuções encontrava a melhor solução em um número de iterações inferior ou igual a 40.

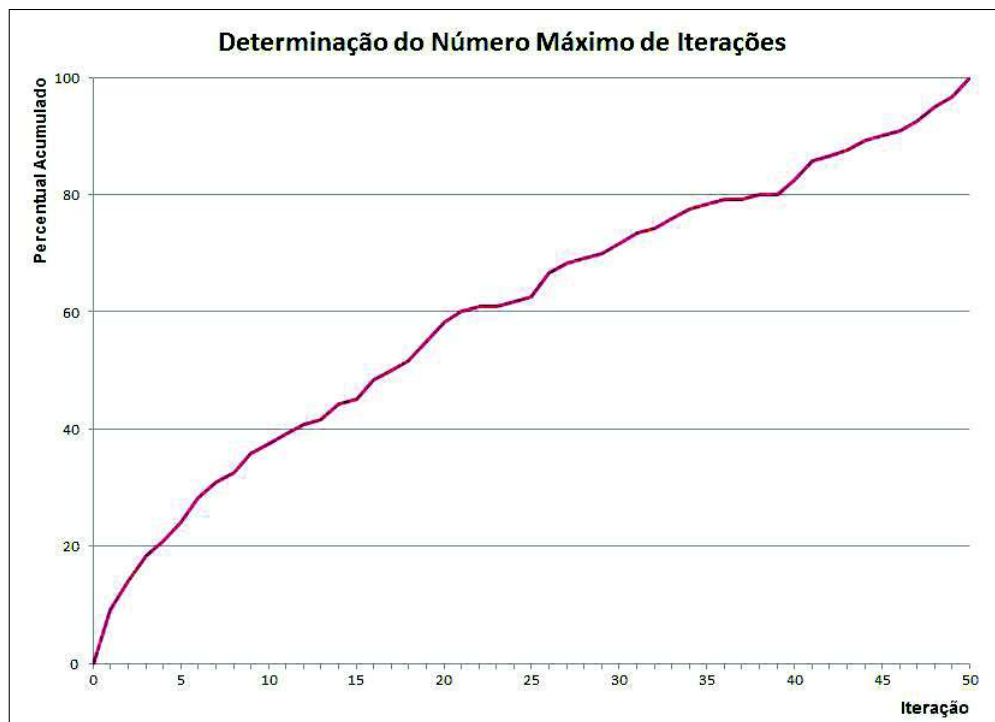


Figura 4.7: Gráfico para determinação do número máximo de iterações ($iterMAX$)

4.3.2.2 Resultados e Análise - HEWMR

Os testes com a HEWMR foram feitos com as instâncias descritas na seção 4.2.1. A heurística foi implementada em linguagem AMPL e utilizou-se o Modelo P para resolver os problemas de sequenciamento em cada sub-árvore gerada pela HEW. O modelo matemático foi resolvido pelo pacote de otimização CPLEX 12.6 com sua configuração padrão. Estipulou-se um tempo computacional mínimo de 3600 segundos para resolução do modelo. Os testes foram feitos com 20 das 48 instâncias reais com maior número de *jobs* e rodados em um computador Intel Xeon X5690@3.47GHz, com 24 núcleos e 132 GB de memória RAM.

Os resultados dos testes com a HEWMR são apresentados nas Tabelas 4.11 e

4.12. Na Tabela 4.11, a coluna “*custo**” apresenta o custo associado à melhor solução encontrada dentro das 20 execuções da heurística, a coluna “*custo_m*” apresenta o custo médio das melhores soluções encontradas dentro das 20 execuções desta heurística e a coluna “*dp_c*” apresenta o desvio padrão desses custos. Observa-se, nesta tabela, que 9 das 20 instâncias testadas apresentaram desvio padrão igual a zero, ou seja, elas encontraram a melhor solução em todas as 20 execuções da heurística. As colunas “*it_{MC}*” e “*it_{PC}*” apresentam a iteração em que a melhor solução foi encontrada no melhor caso e no pior caso, respectivamente. Nota-se, observando a Tabela 4.11, que apenas duas instâncias encontraram a melhor solução na 40^a iteração.

Tabela 4.11: Resultados da resolução da HEWMR para as instâncias reais

INST.	<i>custo*</i>	<i>custo_m</i>	<i>dp_c</i>	<i>it_{MC}</i>	<i>it_{PC}</i>
IR03	28.287	28.297,1	13,4	1	39
IR04	54.950	55.002,3	12,3	1	32
IR11	32.738	32.738,0	0,0	1	16
IR12	55.174	55.174,0	0,0	1	15
IR14	42.270	42.569,1	1.337,6	1	14
IR16	54.917	54.917,0	0,0	1	22
IR18	37.413	37.418,4	9,9	1	38
IR20	67.097	67.097,0	0,0	1	1
IR21	48.965	48.965,5	2,0	2	37
IR23	99.339	99.714,6	794,1	1	40
IR27	42.117	42.117,0	0,0	1	15
IR29	39.411	39.857,1	703,0	1	37
IR30	48.621	48.625,2	14,4	1	35
IR36	57.078	57.078,0	0,0	1	18
IR37	62.833	62.833,0	0,0	1	34
IR39	83.750	84.643,7	763,7	1	33
IR40	89.403	89.932,6	238,2	1	40
IR44	50.131	50.131,0	0,0	2	29
IR46	45.384	45.384,0	0,0	1	15
IR47	36.930	37.052,9	89,4	2	35

A Tabela 4.12 faz um comparativo entre as soluções obtidas pelo Modelo P e pela HEW para as 20 instâncias testadas. Nesta tabela, as colunas “*custo(w)*” apresentam o custo da melhor solução gerada pelo método e entre parêntesis o número de *setups* realizados. As colunas “*d(%)*” e “*d_H(%)*” apresentam a diferença percentual entre a solução obtida pela HEWMR e a solução obtida pelo Modelo P e a diferença percentual entre a solução obtida pela HEWMR e a solução obtida pela HEW, respectivamente. Os valores obtidos na coluna “*d(%)*” são obtidos pela equação (4.64) e os valores da coluna “*d_H(%)*” são obtidos pela equação (4.66), onde $f(HEW)$ é o custo obtido pela HEW na resolução da instância. E a coluna “*T_m(s)*” apresenta o tempo computacional médio (em segundos) gasto pela HEWMR.

$$d_H(\%) = \frac{Custo - f(HEW)}{f(HEW)} \quad (4.66)$$

Tabela 4.12: Resultados da resolução da HEWMR para as instâncias reais

INST.	Modelo P		*	HEW	HEWMR			
	custo(w)	TC(s)		custo(w)	custo(w)	d(%)	d _H (%)	T _m (s)
IR03	28.283(4)	7203	*	34.292(5)	28.287(4)	0,01	-17,51	85,4
IR04	54.950(5)	20		55.772(6)	54.950(5)	0,00	-1,47	50,2
IR11	32.738(4)	830		38.785(5)	32.738(4)	0,00	-15,59	72,7
IR12	55.174(5)	27		56.034(6)	55.174(5)	0,00	-1,53	63,3
IR14	42.270(6)	6		48.252(7)	42.270(6)	0,00	-12,40	49,7
IR16	54.917(5)	40		55.057(5)	54.917(5)	0,00	-0,25	45,2
IR18	37.413(6)	6		37.477(6)	37.413(6)	0,00	-0,17	58,6
IR20	66.899(8)	10		67.097(8)	67.097(8)	0,30	0,00	32,7
IR21	48.965(6)	154		54.989(7)	48.965(6)	0,00	-10,95	51,9
IR23	99.339(9)	6		102.025(10)	99.339(9)	0,00	-2,63	45,6
IR27	42.117(5)	10		42.307(5)	42.117(5)	0,00	-0,45	60,1
IR29	39.370(5)	7205	*	41.718(6)	39.411(5)	0,10	-5,53	56,8
IR30	48.621(7)	112		54.650(8)	48.621(7)	0,00	-11,03	87,0
IR36	57.051(6)	10		59.252(7)	57.078(6)	0,05	-3,67	34,7
IR37	62.833(8)	6		63.210(8)	62.833(8)	0,00	-0,60	41,5
IR39	83.750(8)	6		85.367(9)	83.750(8)	0,00	-1,89	48,1
IR40	89.378(9)	164		92.325(9)	89.403(9)	0,03	-3,16	49,7
IR44	50.131(6)	7		56.460(7)	50.131(6)	0,00	-11,21	42,7
IR46	45.384(7)	8		45.394(7)	45.384(7)	0,00	-0,02	46,9
IR47	36.930(3)	2920		39.669(4)	36.930(3)	0,00	-6,90	39,0

* Instâncias que não foram resolvidas no ótimo

Analisando-se a Tabela 4.12, nota-se que em apenas 5 das 20 instâncias testadas não foi encontrada a solução ótima ou a melhor solução (quando a solução ótima não era conhecida). A maior diferença percentual aconteceu na instância IR20 com 0,3%. Pode-se observar, também, que em 19 das 20 instâncias a HEWMR melhorou a solução da HEW. Apenas a instância IR20 não foi melhorada, apresentando o mesmo resultado da HEW. O percentual médio de melhora em relação a HEW foi de 5,3%, sendo que a maior melhora foi encontrada na resolução da instância IR03 com 17,51%.

Assim como na resolução dos modelos matemáticos, e na resolução da HEW, vê-se uma significativa melhora entre a solução aplicada na prática e solução gerada pela HEWMR. Em 6 das 20 instâncias, a melhora foi superior a 50%. O tempo computacional médio para resolução da HEWMR foi de 53,1 segundos com desvio padrão de 14,5 segundos.

4.4 Conclusões

Neste capítulo, apresentou-se, inicialmente, quatro modelos de PLIM para a resolução do Problema de Sequenciamento de Panelas no setor de lingotamento contínuo de uma aciaria. Para testar o desempenho dos modelos foram geradas, à partir de relatórios gerenciais da ESB, 96 instâncias-teste. Os modelos se mostraram mais eficazes do que a política adotada pela ESB, apresentando melhores resultados em quase todas as instâncias. Isto demonstra que estes modelos podem ser adotados na prática pela empresa.

Dos quatro modelos testados, o modelo P, que faz analogia entre o PSP e o problema de roteamento de veículos assimétrico capacitado, foi o que apresentou melhor desempenho, conseguindo resolver instâncias com até 72 *jobs*.

Além dos modelos, são apresentadas duas heurísticas para a resolução do PSP. Ambas heurísticas são baseadas na Heurística de Esau-Williams. As heurísticas foram criadas porque algumas instâncias não foram resolvidas otimamente pelo modelo P, e também levando-se em consideração uma possível ampliação do horizonte de planejamento. As heurísticas também apresentaram um bom desempenho, com resultados bem próximos do ótimo ou da melhor solução conhecida.

Capítulo 5

Problema de Agrupamento de Pedidos

Neste capítulo, apresentam-se inicialmente, na seção 5.1, os modelos matemáticos para resolução do Problema de Agrupamento de Pedidos (PAP). O PAP consiste em agrupar os pedidos em painéis, as quais são sequenciadas em seguida na MLC de tal forma que os custos de produção da ESB sejam minimizados. São propostos cinco modelos, os quais são diferenciados pela maneira como as variáveis de decisão são definidas.

Já na seção 5.2, apresentam-se as estratégias para a heurística *relax-and-fix* e *fix-and-optimize*, usada para resolver o PAP.

Logo após, na seção 5.3, apresenta-se como as instâncias-teste foram geradas e os resultados obtidos para estas instâncias com os modelos matemáticos e com as heurísticas.

5.1 Problema de Agrupamento de Painéis

Antes dos pedidos serem enviados à aciaria, estes devem ser analisados para definir o conjunto de painéis a serem processadas neste setor. Este tipo de problema é conhecido como Problema de Agrupamento de Pedidos (PAP). O PAP consiste em transformar os pedidos dos clientes em painéis, as quais devem ser processadas em um determinado período de tempo t dentro do horizonte de planejamento. Ao contrário do trabalho de Tang e Wang [54], que procuram alocar pedidos (em forma de placas) às painéis, neste trabalho além de alocar os pedidos às painéis, procura-se definir, também, em qual período cada painel será processada.

Algumas características do PAP são descritas a seguir:

- **Entrada:** Pedido de um cliente para um determinado tipo de aço, o qual deverá ser entregue em um período t . Cada pedido contém as seguintes características:
 - (i) Cliente que define o pedido;
 - (ii) Composição química do pedido (tipo de aço);
 - (iii) Largura desejada da placa ao final do processo de LC;
 - (iv) Período em que o pedido deve ser entregue;
 - (v) Quantidade de aço desejada pelo cliente (em toneladas).

Saída: para cada período serão decididos as frações de cada pedido a serem produzidas, a composição química do aço e a largura da placa para o processamento de cada panela no processo de lingotamento contínuo da ESB.

Antes de apresentar os modelos para o PAP faz-se necessário apresentar os seguintes conceitos:

- **Composição Química Compatível:** Um pedido i tem composição química compatível com um pedido p se pode haver um *upgrade* na composição química deste pedido para a composição química do pedido p ou se ambos os pedidos possuem a mesma composição química.
- **Largura Compatível:** Um pedido i tem largura compatível com um pedido p se o pedido i tem largura menor e pode ser cortada uma rebarba da placa ou se ambos os pedidos tiverem a mesma largura.

Para resolução do PAP em painéis, criou-se cinco modelos de PLIM, os quais são descritos nas seções a seguir.

Seja $T = \{1, \dots, nt\}$ o conjunto de períodos e $I = \{1, \dots, np\}$ o conjunto de pedidos. Sejam, ainda, w_i a quantidade (em toneladas) de aço do pedido i , q^t o tempo máximo disponível para o processamento das painéis no período t , G_{max} a quantidade máxima de aço (em toneladas) que pode ser alocada a uma panela, G_{min} a quantidade mínima de aço (em toneladas) que deve ser alocada a uma panela. Os demais dados para todos os modelos são:

l_i^t = custo de atrasar a entrega do pedido i para o período t , por exemplo, se o pedido i deve ser entregue no período 2, l_i^3 corresponderá ao custo de atrasar a entrega do pedido em um período. Se o pedido for finalizado antes ou no período em que ele deve ser entregue l_i^t é igual a zero. Observação: O custo l_i^{m+1} corresponde ao custo de não entregar o pedido i completo dentro do horizonte de planejamento.

e_i^t = custo de manutenção de estoque do pedido i no período t . Nesta situação, se o pedido i deve ser entregue no período 2, e_i^1 corresponderá ao custo de manter o pedido i em estoque por um período. Se o pedido for finalizado depois ou no período em que ele deve ser entregue e_i^t é igual a zero. O custo de estoque é proporcional à quantidade de aço que foi produzida até então.

E as variáveis comuns a todos os modelos são:

$$\tau_i^t = \begin{cases} 1, & \text{se o pedido } i \text{ não foi completado até o período } t, \text{ inclusive} \\ 0, & \text{caso contrário} \end{cases}$$

L_i = custo incorrido pelo atraso na entrega completa do pedido i .

Adotou-se duas premissas na definição dos modelos descritos nas seções a seguir: (i) Todos os pedidos que forem agrupados para formar uma panela devem ter composição química e largura compatíveis; (ii) Considera-se que o estoque tem capacidade infinita.

5.1.1 Modelo PED

Neste primeiro modelo, as panelas serão definidas com base nos pedidos. Uma panela a ser processada no LC possuirá a composição química e a largura de um pedido p , ou seja, predominará as características do pedido p .

Além dos dados descritos anteriormente, o modelo PED possui os seguintes parâmetros:

pr_i = tempo necessário para o processamento de uma panela com a composição química predominante do pedido i .

v_{pi} = custo de *upgrade* quando os pedidos p e i são agrupados em uma mesma panela, sendo que o pedido i passará a ter todas as características do pedido p , pois este tem qualidade superior;

$$a_{pi} = \begin{cases} 1, & \text{se os pedidos } i \text{ e } p \text{ possuem composi\c{c}\tilde{a}o qu\imica e largura} \\ & \text{compat\i{v}eis} \\ 0, & \text{caso contr\c{a}rio} \end{cases}$$

Al\em das vari\aveis descritas anteriormente, o modelo PED possui as seguintes vari\aveis de decis\~ao:

y_{pi}^t = fra\c{c}\~ao do pedido i que ser\c{a} processado na panela com caracter\xedsticas predominantes do pedido p , no per\xedodo t .

$$z_p^t = \begin{cases} 1, & \text{se \e{e} formada uma panela com as mesmas caracter\xedsticas do pedido } p \text{ no} \\ & \text{per\xedodo } t \\ 0, & \text{caso contr\c{a}rio} \end{cases}$$

Neste modelo deve-se dividir os pedidos maiores que a capacidade da panela em pedidos menores (no caso em estudo, 170 toneladas). Isto deve ser feito para que a vari\avel z_p^t seja bin\c{a}ria.

Desta forma, o Modelo PED \e{e} definido como se segue:

$$\min \sum_{p \in I} \sum_{i \in I | a_{pi}=1} \sum_{t \in T} (v_{pi} y_{pi}^t + e_i^t y_{pi}^t) + \sum_{i \in I} L_i \quad (5.1)$$

$$\text{s. \c{a}:} \quad \sum_{p \in I | a_{pi}=1} \sum_{t \in T} y_{pi}^t \leq 1 \quad \forall i \in I \quad (5.2)$$

$$\sum_{i \in I | a_{pi}=1} w_i y_{pi}^t - G_{max} z_p^t \leq 0 \quad \forall p \in I, \forall t \in T \quad (5.3)$$

$$\sum_{i \in I | a_{pi}=1} w_i y_{pi}^t - G_{min} z_p^t \geq 0 \quad \forall p \in I, \forall t \in T \quad (5.4)$$

$$\sum_{p \in I} pr_p z_p^t \leq q^t \quad \forall t \in T \quad (5.5)$$

$$\tau_i^t + \sum_{u=1}^t \sum_{p \in I | a_{pi}=1} y_{pi}^u \geq 1 \quad \forall i \in I, t \in T \quad (5.6)$$

$$L_i - \tau_i^t l_i^{t+1} \geq 0 \quad \forall i \in I, t \in T \quad (5.7)$$

$$y_{pi}^t \geq 0 \quad \forall p, i \in I, \forall t \in T \quad (5.8)$$

$$z_p^t \in \{0, 1\} \quad \forall p \in I, \forall t \in T \quad (5.9)$$

$$\tau_i^t \in \{0, 1\} \quad \forall i \in I, \forall t \in T \quad (5.10)$$

$$L_i \geq 0 \quad \forall i \in P \quad (5.11)$$

A função objetivo é representada pela equação (5.1) e busca minimizar os custos de agrupamentos de pedidos em uma mesma panela e os custos de atraso e de estoque. As restrições (5.2) garantem que o somatório das parcelas do pedido i entregues não seja superior a 1. As restrições (5.3) garantem que a quantidade de aço gerado pelo agrupamento dos pedidos em uma mesma panela não ultrapasse a capacidade máxima desta panela. Já as restrições (5.4) garantem que cada panela formada deve conter uma quantidade mínima de aço. As restrições (5.5) garantem que o tempo necessário para processar todas as painelas no período t não ultrapasse a quantidade de tempo disponível para aquele período. As restrições (5.6) definem em qual período o pedido i foi completamente entregue. As restrições (5.7) definem o valor do custo de atraso associado a entrega completa do pedido i . As restrições (5.8) a (5.11) definem o domínio das variáveis.

5.1.2 Modelo PAD-N

Neste segundo modelo utilizam-se variáveis indexadas pelos padrões de panela para encontrar a solução para o PAP, sendo a variável z definida como inteira. Busca-se, desta forma, reduzir o número de variáveis do modelo PED. Padrão de Panela é a composição química do aço e a largura da placa ao final do processamento da panela no LC. Serão identificados tantos padrões de panela quanto as diferentes combinações de composição química e largura que aparecem no conjunto de pedidos. Assim, pedidos de clientes com a mesma combinação de composição química e largura correspondem a um único padrão. Seja $P = \{1, \dots, npad\}$ o conjunto de padrões, onde $npad \leq np$ e pr_p é o tempo necessário para o processamento de uma panela com o padrão p .

O modelo PAD-N é idêntico ao modelo PED, mudando apenas a interpretação relacionada às variáveis de decisão. Aqui a variável z_p^t corresponde ao número de painelas com o padrão p a serem processadas no período t . Além disso, o índice p diz respeito a um padrão de panela.

5.1.3 Modelo PAD-Bk

Nesse modelo, assim como no PAD-N, as painelas são definidas com base nos padrões. Mas, ao contrário do modelo PAD-N, o modelo PAD-Bk utiliza variáveis z binárias. Para isto, as variáveis z são definidas da seguinte forma:

$$z_{kp}^t = \begin{cases} 1, & \text{se a } k\text{-ésima panela com o padrão } p \text{ é processada no período } t \\ 0, & \text{caso contrário} \end{cases}$$

No modelo apresentado a seguir, $K[p]$ representa o número máximo de painéis com o padrão p por período. Neste trabalho definiu-se que o valor de $K[p]$ é dado pela equação 5.12.

$$K[p] = \min \left\{ \left\lceil \frac{Q_{max}}{pr_p} \right\rceil, \sum_{i \in I} a_{pi} \right\}, \quad \forall p \in P \quad (5.12)$$

$$\text{onde } Q_{max} = \max_{1 <= t <= nt} \{q^t\}$$

Daí, o modelo PAD-Bk é definido da seguinte forma:

$$\min \sum_{p \in P} \sum_{i \in I | a_{pi}=1} \sum_{t \in T} (v_{pi} y_{pi}^t + e_i^t y_{pi}^t) + \sum_{i \in I} L_i \quad (5.13)$$

$$\text{s. à:} \quad \sum_{p \in P | a_{pi}=1} \sum_{t \in T} y_{pi}^t \leq 1 \quad \forall i \in I \quad (5.14)$$

$$\sum_{i \in I | a_{pi}=1} w_i y_{pi}^t - G_{max} \sum_{k=1}^{K[p]} z_{kp}^t \leq 0 \quad \forall p \in P, \forall t \in T \quad (5.15)$$

$$\sum_{i \in I | a_{pi}=1} w_i y_{pi}^t - G_{min} \sum_{k=1}^{K[p]} z_{kp}^t \geq 0 \quad \forall p \in P, \forall t \in T \quad (5.16)$$

$$\sum_{p \in P} pr_p \left(\sum_{k=1}^{K[p]} z_{kp}^t \right) \leq q^t \quad \forall t \in T \quad (5.17)$$

$$\tau_i^t + \sum_{u=1}^t \sum_{p \in P | a_{pi}=1} y_{pi}^u \geq 1 \quad \forall i \in I, t \in T \quad (5.18)$$

$$L_i - \tau_i^t l_i^{t+1} \geq 0 \quad \forall i \in I, t \in T \quad (5.19)$$

$$z_{kp}^t - z_{k-1,p}^t \leq 0 \quad \forall p \in P, t \in T, k = 2 \dots K[p] \quad (5.20)$$

$$y_{pi}^t \geq 0 \quad \forall p \in P, i \in I, \forall t \in T \quad (5.21)$$

$$z_{kp}^t \in \{0, 1\} \quad \forall p \in P, \forall t \in T \quad (5.22)$$

$$\tau_i^t \in \{0, 1\} \quad \forall i \in I, \forall t \in T \quad (5.23)$$

$$L_i \geq 0 \quad \forall i \in I \quad (5.24)$$

A função objetivo é representada pela equação (5.13) e busca minimizar os custos de agrupamentos de pedidos em uma mesma panela e os custos de atraso e

de estoque. As restrições (5.14) garantem que o somatório das parcelas do pedido i entregues não seja superior a 1. As restrições (5.15) garantem que a quantidade de aço gerado pelo agrupamento dos pedidos em uma mesma panela não ultrapasse a capacidade máxima desta panela. Já as restrições (5.16) garantem que cada panela formada deve conter uma quantidade mínima de aço. As restrições (5.17) garantem que tempo necessário para processar todas as painelas no período t não ultrapasse a quantidade de tempo disponível para aquele período. As restrições (5.18) definem em qual período o pedido i foi completamente entregue. As restrições (5.19) definem o valor do custo de atraso associado a entrega completa do pedido i . As restrições (5.20) garantem que a k -ésima panela com o padrão p no período t , só será ativada se $(k - 1)$ -ésima panela com o padrão p no período t for processada. As restrições (5.21) a (5.24) definem o domínio das variáveis.

5.1.4 Modelo PAD-Bd

No modelo PAD-Bd as painelas também são definidas de acordo com os padrões. Aqui as variáveis do tipo z são definidas como binárias e um índice d definirá o número de painelas do padrão p a serem processadas. Desta forma, a variável z do modelo PAD-Bd é definida como se segue:

$$z_{dp}^t = \begin{cases} 1, & \text{se são processadas } d \text{ painelas com o padrão } p \text{ no período } t \\ 0, & \text{caso contrário} \end{cases}$$

O modelo PAD-Bd é apresentado a seguir:

$$\min \sum_{p \in P} \sum_{i \in I | a_{pi}=1} \sum_{t \in T} (v_{pi} y_{pi}^t + e_i^t y_{pi}^t) + \sum_{i \in I} L_i \quad (5.25)$$

$$\text{s. à: (5.14), (5.18), (5.19), (5.21), (5.23), (5.24)}$$

$$\sum_{i \in I | a_{pi}=1} w_i y_{pi}^t - G_{max} \sum_{d=1}^{K[p]} d z_{dp}^t \leq 0 \quad \forall p \in P, \forall t \in T \quad (5.26)$$

$$\sum_{i \in I | a_{pi}=1} w_i y_{pi}^t - G_{min} \sum_{d=1}^{K[p]} d z_{dp}^t \geq 0 \quad \forall p \in P, \forall t \in T \quad (5.27)$$

$$\sum_{p \in P} pr_p \left(\sum_{d=1}^{K[p]} d z_{dp}^t \right) \leq q^t \quad \forall t \in T \quad (5.28)$$

$$\sum_{d=1}^{K[p]} z_{dp}^t \leq 1 \quad \forall p \in P, t \in T \quad (5.29)$$

$$z_{dp}^t \in \{0, 1\} \quad \forall p \in P, \forall t \in T \quad (5.30)$$

Neste modelo, a função objetivo é representada pela equação (5.25) e, assim como nos modelos anteriores, busca minimizar os custos de agrupamentos de pedidos em uma mesma panela e os custos de atraso e de estoque.

As restrições (5.26) garantem que a quantidade de aço gerado pelo agrupamento dos pedidos em uma mesma panela não ultrapasse a capacidade máxima desta panela. E as restrições (5.27) garantem que cada panela formada deve conter uma quantidade mínima de aço. As restrições (5.28) garantem que tempo necessário para processar todas as panelas no período t não ultrapasse a quantidade de tempo disponível para aquele período. As restrições (5.29) garantem que a variável z_{dp}^t será igual a 1 para no máximo um valor de d . Finalmente, as restrições (5.30) definem o domínio das variáveis.

5.1.5 Modelo PAD-B2

Este quarto modelo com panelas definidas de acordo com os padrões foi inspirado no trabalho de Owen e Mehrotra[37] que busca reformular modelos de PLIM usando variáveis binárias. Neste modelo, as variáveis binárias u_{ip}^t são definidas de forma que $z_p^t = \sum_{i=0}^{M_p} 2^i u_{ip}^t$, sendo que as variáveis z_p^t são definidas como contínuas e representam o número de panelas com padrão p processadas no período t . O parâmetro M_p é definido por $\lceil \log_2 K[p] \rceil$. Por exemplo, se $K[p] = 7$ então $M_p = 2$, logo se forem produzidas 5 panelas com o padrão p no período t , as variáveis u_{2p}^t e u_{0p}^t devem ser iguais a 1, daí $z_p^t = 2^0 u_{0p}^t + 2^1 u_{1p}^t + 2^2 u_{2p}^t = 1(1) + 2(0) + 4(1) = 5$. O modelo PAD-B2 é apresentado a seguir:

$$\begin{aligned} \min \quad & \sum_{p \in P} \sum_{i \in I | a_{pi}=1} \sum_{t \in T} (v_{pi} y_{pi}^t + e_i^t y_{pi}^t) + \sum_{i \in I} L_i \\ \text{s. à:} \quad & (5.14), (5.19), (5.21), (5.23), (5.24) \end{aligned} \quad (5.31)$$

$$\sum_{i \in I|a_{pi}=1} w_i y_{pi}^t - G_{max} z_p^t \leq 0 \quad \forall p \in P, \forall t \in T \quad (5.32)$$

$$\sum_{i \in I|a_{pi}=1} w_i y_{pi}^t - G_{min} z_p^t \geq 0 \quad \forall p \in P, \forall t \in T \quad (5.33)$$

$$\sum_{p \in P} pr_p z_p^t \leq q^t \quad \forall t \in T \quad (5.34)$$

$$\tau_i^t + \sum_{t'=1}^t \sum_{p \in P|a_{pi}=1} y_{pi}^{t'} \geq 1 \quad \forall i \in I, t \in T \quad (5.35)$$

$$z_p^t = \sum_{i=0}^{M_p} 2^i u_{ip}^t \quad \forall p \in P, \forall t \in T \quad (5.36)$$

$$z_p^t \geq 0 \quad \forall p \in P, \forall t \in T \quad (5.37)$$

$$u_{ip}^t \in \{0, 1\} \quad \forall i \in \{0, \dots, K[p]\}, \forall p \in P, \forall t \in T \quad (5.38)$$

Neste modelo as equações (5.31) a (5.35), (5.37) são definidas exatamente como nos modelos PED e PAD-N. No modelo PAD-B2 introduzem-se as restrições (5.36) que garantem que as variáveis z_p^t , definidas como contínuas (equação (5.37)), assumam apenas valores inteiros. Estas restrições foram definidas de acordo com o trabalho de Owen e Mehrotra[37]. As restrições (5.38) definem o domínio das variáveis u_{ip}^t .

5.2 Heurística RF-FOPT para o PAP

Com a finalidade de encontrar boas soluções em um menor tempo computacional, desenvolveu-se uma heurística baseada nas técnicas *relax-and-fix* e *fix-and-optimize* para a resolução do problema de agrupamento de pedidos. Resolveu-se adotar estas técnicas, pois vários trabalhos recentes vêm adotando estas técnicas para resolução do Problema Integrado de Dimensionamento e Sequenciamento de Lotes (Baldo *et al.* [6], Ferreira *et al.* [18], entre outros).

Segundo Baldo *et al.* [6], nas heurísticas *relax-and-fix*(RF), todas as variáveis inteiras são relaxadas e particionadas em subconjuntos disjuntos. Em cada iteração, as variáveis de um destes subconjuntos são definidas como inteiras, obtendo um subproblema de PLIM menor. Este submodelo é resolvido e as variáveis inteiras fixadas para seu valor corrente. Esse processo iterativo é repetido até todas as variáveis dos subconjuntos serem fixadas, ou um dos subproblemas ser inviável.

Ainda, segundo Baldo *et al.* [6], as heurísticas *fix-and-optimize* (FOpt) iniciam

com uma solução viável e tentam iterativamente melhorá-la. Em cada iteração da heurística há dois passos básicos: (a) um subconjunto de variáveis é selecionado para ser livre para a re-otimização e (b) as variáveis restantes mantêm seus valores da melhor solução encontrada até então.

Partindo das definições anteriores, combinaram-se 6 estratégias *relax-and-fix* e 4 estratégias *fix-and-optimize* para resolver o PAP. Na heurística desenvolvida, as soluções iniciais para a heurística *fix-and-optimize* são determinadas a partir da heurística *relax-and-fix*.

As estratégias adotadas para a heurística *relax-and-fix* são definidas a seguir:

- **RF-zt1f:** Inicialmente mantêm-se a restrição de integralidade das variáveis z_{kp}^t no período $t = 1$ e relaxam-se as demais variáveis z_{kp}^t . Resolve-se o modelo. Atualiza-se o período t para $t + 1$. Nos passos seguintes, fixam-se as variáveis z_{kp}^t no período $t - 1$ a 1 em seus valores atuais, mantêm-se a restrição de integralidade das variáveis z_{kp}^t no período t corrente e relaxam-se as demais variáveis z_{kp}^t . Daí, resolve-se novamente o modelo. O passo iterativo é feito até que o último período (nt) seja resolvido. As demais variáveis (τ_i^t , L_i e y_{pi}^t) ficam livres para re-otimização em todas as iterações, não sendo fixadas em nenhum momento. Além disso, as variáveis τ_i^t não são relaxadas em nenhuma iteração do método.
- **RF-zt2f:** Inicialmente mantêm-se a restrição de integralidade das variáveis z_{kp}^t no período $t = 1$ e $t = 2$ e relaxam-se as demais variáveis z_{kp}^t . Resolve-se o modelo. Atualiza-se o período t para $t + 1$. Nos passos seguintes, fixam-se as variáveis z_{kp}^t no período $t - 1$ a 1 em seus valores atuais, mantêm-se a restrição de integralidade das variáveis z_{kp}^t no período t corrente e no período $t + 1$ e relaxam-se as demais variáveis z_{kp}^t . Daí, resolve-se novamente o modelo. O passo iterativo é feito até que o penúltimo período ($nt - 1$) seja resolvido. Assim como na estratégia anterior, as demais variáveis (τ_i^t , L_i e y_{pi}^t) ficam livres para re-otimização em todas as iterações, não sendo fixadas em nenhum momento.
- **RF-zt1b:** Inicialmente mantêm-se a restrição de integralidade das variáveis z_{kp}^t no período $t = nt$ e relaxam-se as demais variáveis z_{kp}^t . Resolve-se o modelo. Atualiza-se o período t para $t - 1$. Nos passos seguintes, fixam-se as variáveis z_{kp}^t no período $t + 1$ a nt em seus valores atuais, mantêm-se a restrição de integralidade das variáveis z_{kp}^t no período t corrente e relaxam-se as demais variáveis z_{kp}^t . Daí, resolve-se novamente o modelo. O passo iterativo é feito até que o primeiro

período seja resolvido. E, assim como nas estratégias anteriores, as demais variáveis (τ_i^t , L_i e y_{pi}^t) ficam livres para re-otimização em todas as iterações, não sendo fixadas em nenhum momento.

- **RF- $\tau t 1f$:** Inicialmente mantêm-se a restrição de integralidade das variáveis z_{kp}^t e τ_i^t no período $t = 1$ e relaxam-se as demais variáveis z_{kp}^t e τ_i^t . Resolve-se o modelo. Atualiza-se o período t para $t + 1$. Nos passos seguintes, fixam-se as variáveis z_{kp}^t e τ_i^t nos períodos $t - 1$ a 1 em seus valores atuais, mantêm-se a restrição de integralidade das variáveis z_{kp}^t e τ_i^t no período t corrente e relaxam-se as demais variáveis z_{kp}^t e τ_i^t . Daí, resolve-se novamente o modelo. O passo iterativo é feito até que o último período (nt) seja resolvido. As variáveis (L_i e y_{pi}^t) ficam livres para re-otimização em todas as iterações, não sendo fixadas em nenhum momento.
- **RF- $\tau t 2f$:** Inicialmente mantêm-se a restrição de integralidade das variáveis z_{kp}^t e τ_i^t no período $t = 1$ e $t = 2$ e relaxam-se as demais variáveis z_{kp}^t e τ_i^t . Resolve-se o modelo. Atualiza-se o período t para $t + 1$. Nos passos seguintes, fixam-se as variáveis z_{kp}^t e τ_i^t no período $t - 1$ a 1 em seus valores atuais, mantêm-se a restrição de integralidade das variáveis z_{kp}^t e τ_i^t no período t corrente e no período $t + 1$, relaxam-se as demais variáveis z_{kp}^t e τ_i^t . Daí, resolve-se novamente o modelo. O passo iterativo é feito até que o penúltimo ($nt - 1$) período seja resolvido. Assim como na estratégia RF- $\tau t 1f$, as variáveis (L_i e y_{pi}^t) ficam livres para re-otimização em todas as iterações, não sendo fixadas em nenhum momento.
- **RF- $z p 1f$:** Inicialmente mantêm-se a restrição de integralidade das variáveis z_{kp}^t para o padrão $p = 1$ e relaxam-se as demais variáveis z_{kp}^t . Resolve-se o modelo. Atualiza-se o período p para $p + 1$. Nos passos seguintes, fixam-se as variáveis z_{kp}^t para o padrão $p - 1$ a 1 em seus valores atuais, mantêm-se a restrição de integralidade das variáveis z_{kp}^t para o padrão p corrente e relaxam-se as demais variáveis z_{kp}^t . Daí, resolve-se novamente o modelo. O passo iterativo é feito até que o último padrão (np) seja resolvido. As demais variáveis (τ_i^t , L_i e y_{pi}^t) ficam livres para re-otimização em todas as iterações, não sendo fixadas em nenhum momento. E, as variáveis τ_i^t não são relaxadas em nenhuma iteração do método.

E as estratégias da heurística FOpt são:

- **FOpt- $z t 1f$:** A partir de uma solução corrente S , fixam-se as variáveis z_{kp}^t em todos os períodos exceto o período corrente t e re-otimizam-se as variáveis do

período t . Se a solução gerada for melhor que a solução corrente, a solução corrente deverá ser atualizada. O período t é atualizado para $t+1$ a cada iteração e o processo é repetido até que as variáveis do último período nt sejam re-otimizadas.

- **FOpt-zp1f:** A partir de uma solução corrente S , fixam-se as variáveis z_{kp}^t para todos os padrões exceto o padrão corrente p e re-otimizam-se as variáveis do padrão p . Se a solução gerada for melhor que a solução corrente, a solução corrente deverá ser atualizada. O padrão p é atualizado para $p+1$ a cada iteração e o processo é repetido até que as variáveis do último padrão np sejam re-otimizadas.
- **FOpt-ztpf:** Esta estratégia combina as estratégias FOpt-zt1f e FOpt-zp1f. Aqui, inicialmente, aplica-se a estratégia FOpt-zt1f e logo em seguida a estratégia FOpt-zp1f.
- **FOpt-zptf:** Esta estratégia, assim como a estratégia FOpt-ztpf, combina as estratégias FOpt-zt1f e FOpt-zp1f. Mas, inicialmente, aplica-se a estratégia FOpt-zp1f e logo em seguida a estratégia FOpt-zt1f.

Nas estratégias FOpt-zt1f a FOpt-zptf, as variáveis τ_i^t são mantidas inteiras e, assim como as variáveis contínuas L_i e y_{pi}^t , não são fixadas a cada iteração, ou seja, são submetidas a nova re-otimização a cada iteração do método.

A Figura 5.1 apresenta o pseudo-código da heurística RF-FOpt desenvolvida para resolver o PAP. Nesta heurística, TC representa o tempo gasto pela heurística em algum determinado instante de tempo e RT_{max} o tempo máximo para a execução da heurística.

<p>Procedimento <i>RF-FOpt()</i></p> <p>1 $S \leftarrow$ Gerar uma solução inicial usando estratégia RF;</p> <p>2 $TC \leftarrow 0$;</p> <p>3 Enquanto Houver melhora ou $TC < RT_{max}$ faça</p> <p>4 $S \leftarrow$ Re-otimiza usando estratégia FOpt;</p> <p>5 Atualiza TC;</p> <p>6 fim-enquanto</p> <p>7 Retorna S;</p> <p>fim-Procedimento</p>

Figura 5.1: Pseudo-código da heurística RF-FOpt para resolução do PAP

5.3 Resultados e Análise

Nesta seção são apresentados os resultados dos cinco modelos descritos anteriormente para a resolução do PAP e os resultados obtidos com as heurísticas propostas.

Apresenta-se também como foram criadas as instâncias-teste utilizadas. Foram criadas um total de 56 instâncias-teste, sendo que 32 delas foram baseadas em relatórios gerenciais da ESB.

Os testes computacionais foram feitos em um computador Intel Xeon X5690@3.47GHz, com 24 núcleos e 132 GB de memória RAM. Definiu-se um tempo limite de 7200 segundos para obtenção da solução ótima pelo modelo. Os modelos foram implementados e resolvidos através do *software* de otimização CPLEX 12.6, utilizando-se a linguagem AMPL. Para os testes dos modelos, utilizou-se a configuração padrão do CPLEX.

As heurísticas foram desenvolvidas em linguagem AMPL e utilizou-se as configurações padrões do CPLEX 12.6. Estipulou-se um tempo limite de 300 segundos para obtenção da solução ótima pelo modelo matemático e um tempo de execução máximo da heurística de 7200 segundos. Os testes das heurísticas foram feitos no mesmo computador usado para resolver os modelos.

5.3.1 Geração das Instâncias

Os testes computacionais foram feitos com 24 instâncias geradas artificialmente (IAP001 a IAP024) e 32 instâncias geradas à partir de relatórios gerenciais da ESB (IAP025 a IAP056).

Nas instâncias geradas aleatoriamente, definiu-se primeiramente número de períodos que variou de 2 a 10 períodos e, em seguida, definiu-se quantos pedidos deveriam ter suas datas de conclusão definidas para um determinado período. Além disso, definiu-se o número de grupos e quantos pedidos fariam parte de cada grupo. Cada grupo gerado formaria uma panela e o peso final dos pedidos pertencentes ao grupo deveria variar de 160 a 170 toneladas. A ideia era criar pedidos que pudessem ser processados dentro do horizonte de planejamento, mesmo que fosse permitido ao pedido não ser totalmente finalizado dentro deste horizonte. Os grupos gerados possuíam de 1 a 3 pedidos.

Em seguida, gerou-se a quantidade de aço (em toneladas) de cada pedido.

A quantidade de aço de cada pedido foi gerada da seguinte maneira: (i) grupo de 1 pedido: U(160,170) toneladas, (ii) grupo com 2 pedidos: U(60,120) toneladas e (iii) grupo com 3 pedidos: U(40,70) toneladas. Para grupos com 2 ou 3 pedidos, o último pedido era gerado com a expressão U(160,170) menos a soma dos pesos dos demais pedidos do grupo.

Após isto, definiu-se o tipo de aço de cada pedido. O tipo de aço era definido pela composição química e pela largura da placa desejada ao final do processo de LC. Gerou-se 100 tipos de aços. As larguras eram definidas de acordo com a seguinte expressão: $5 \times U(180, 360)$. Os tipos de aço são divididos em 7 famílias e o valor de mercado de cada tipo de aço depende da sua família. O tempo de processamento depende da família do aço e varia de 40 a 44 minutos. O tempo de *setup* é igual a 90 minutos e o custo de realização do *setup* é de \$6000.

Sejam CA o custo de atraso por tonelada por período, $PAtr$ a quantidade de períodos que o pedido atrasou para ser entregue, CE o custo de manutenção de estoque por tonelada por período, $PMEst$ a quantidade de períodos em que o pedido foi mantido em estoque, vm_j o valor de mercado do tipo de aço do pedido j , vm_s o valor de mercado da sucata, PL a perda (em ton) por diferença de largura entre pedidos, qm_j a quantidade de material perdido (em ton) na panela j por causa da mistura dos tipos de aço e plm_j a perda lateral (em ton) da panela j por causa da diferença de largura entre as painéis processadas consecutivamente. Os custos das instâncias são calculados a partir das expressões apresentadas na Tabela 5.1. Considerou-se aqui que $CE = \$85$ e $CA = \$100$.

Tabela 5.1: Expressões para os cálculos dos custos

Custo	Expressão
Atraso (l_i^t)	$CA \times w_i \times PAtr$
Manutenção Estoque (e_i^t)	$CE \times w_i \times PMEst$
Upgrade (v_{pi})	$(vm_p - vm_i) \times w_i + (PL \times vm_s)$
Mistura e Setup (c_{ij})	$qm_i(vm_i - sm_{ij}) + qm_j(vm_j - sm_{ij}) + plm_j(vm_j - vm_s)$

Como foi dito anteriormente, para gerar as instâncias baseadas em casos reais utilizou-se relatórios gerenciais gentilmente cedidos pela ESB. Nestes relatórios não constavam a largura desejada da placa e nem os custos de manutenção de estoque e atraso, desta forma os dados para estas instâncias foram gerados aleatoriamente e calculados conforme as instâncias artificiais. Por isto, não fez-se uma comparação com os resultados apresentados pela ESB. Utilizou-se para criar estas instâncias 230 tipos de aço. Assim como para as instâncias artificiais, os aços foram divididos em sete famílias e os custos foram calculados conforme a Tabela 5.1. Para as instâncias reais utilizou-

se 5, 7 ou 10 períodos como horizonte de planejamento. Para as instâncias IAP024 a IAP039 considerou-se que $CE = \$35$ e $CA = \$55$ e para as instâncias IAP040 a IAP056 considerou-se que $CE = \$4$ e $CA = \$15$.

5.3.2 Resultados - Modelos para o PAP

As Tabelas 5.2 e 5.3 apresentam os resultados das instâncias artificiais e reais do modelo PED, respectivamente. Nestas tabelas a coluna “Inst.” apresenta a instância resolvida, as colunas “NPd” e “NPe” apresentam o número de pedidos e o número de períodos relativos à cada instância, respectivamente. A coluna “T(s)” apresenta o tempo computacional gasto para resolução do modelo pelo CPLEX. Por sua vez, se a instância não era resolvida dentro dos 7200 segundos, a coluna “G(%)” apresenta o *gap* apresentado pelo CPLEX. Já a coluna “Custo” apresenta o custo total apresentado ao final da resolução do modelo. E as colunas “R(%)” e “TR(s)” apresentam, respectivamente, o *gap* de relaxação linear e o tempo computacional para obter a solução de relaxação linear.

Observa-se na Tabela 5.2 que o modelo PED foi capaz de resolver otimamente 22 das 24 instâncias artificiais geradas. O tempo computacional médio para a resolução destas instâncias foi de 835 segundos com um desvio-padrão de 2031,9 segundos. Das 8 instâncias com 100 pedidos, três foram resolvidas com tempo inferior a 20 segundos. As duas instâncias que não foram resolvidas otimamente, apresentaram um *gap* médio do CPLEX de 5,75%. O percentual médio de relaxação linear foi de 60,5% com desvio-padrão de 21,4%. O tempo computacional para obter a relaxação linear das instâncias foi inferior a 1 segundo.

Observando-se a Tabela 5.3, nota-se que 23 das 32 instâncias são resolvidas otimamente, ou seja, 72% das instâncias. O tempo computacional médio para a resolução das instâncias foi de 2480,2 segundos com desvio padrão de 3226,1 segundos. Das 32 instâncias, seis foram resolvidas com tempo computacional inferior a 10 segundos, todas com 5 períodos. As instâncias que não foram resolvidas otimamente, apresentaram um *gap* médio do CPLEX de 17,9% com desvio-padrão de 25,5%. Apenas a instância IAP036 apresentou um *gap* do CPLEX muito alto (87,4%), sendo que as demais instâncias apresentaram um *gap* do CPLEX inferior a 20%. Três instâncias apresentaram *gap* inferior a 5%. O percentual médio de relaxação linear foi de 70,6% com desvio-padrão de 33,1%, sendo que 15 das 32 instâncias apresentaram um percentual de relaxação linear igual a 100%. O tempo computacional médio para obtenção da

Tabela 5.2: Resultados das instâncias artificiais com o modelo PED

Inst.	NPd	NPe	T(s)	G(%)	Custo	R(%)	TR(s)
IAP001	20	2	1,0		62.154,7	75,1	0,0
IAP002	25	3	1,0		69.119,2	62,0	0,0
IAP003	30	3	1,0		16.340,0	89,3	0,0
IAP004	30	4	2,0		37.514,8	64,6	1,0
IAP005	40	3	3,0		30.204,0	86,8	0,0
IAP006	40	4	1,0		116.658,2	12,3	0,0
IAP007	40	5	1,0		293.785,1	38,7	0,0
IAP008	40	6	6,0		168.688,9	77,9	1,0
IAP009	45	3	1,0		32.591,4	73,2	0,0
IAP010	45	4	2,0		38.822,9	63,0	0,0
IAP011	45	5	0,0		49.773,0	22,9	0,0
IAP012	45	6	1,0		22.097,7	62,6	0,0
IAP013	50	3	2,0		38.980,6	59,4	0,0
IAP014	50	4	2,0		63.343,0	49,8	0,0
IAP015	50	5	1,0		79.951,9	18,3	0,0
IAP016	50	6	3,0		53.178,3	66,7	0,0
IAP017	100	3	6,0		70.096,8	58,3	0,0
IAP018	100	4	11,0		172.981,2	27,4	0,0
IAP019	100	5	7206,0	5,8	94.951,9	76,3	0,0
IAP020	100	6	19,0		46.931,9	73,9	1,0
IAP021	100	7	843,0		132.980,6	71,3	0,0
IAP022	100	8	2641,0		122.660,1	59,6	0,0
IAP023	100	9	2086,0		126.614,9	75,9	0,0
IAP024	100	10	7201,0	5,7	144.081,4	86,8	0,0

relaxação linear foi de 2,6 segundos com desvio-padrão de 2,7 segundos. A relaxação linear de 68% das instâncias foi obtida com tempo inferior a 2 segundos.

As Tabelas 5.4 e 5.5 apresentam os resultados das instâncias artificiais e reais do modelo PAD-N, respectivamente. A estrutura destas tabelas é idêntica à da tabela 5.2, incluindo-se a coluna “NPa” que representa o número de padrões encontrados na instância.

Observando-se a Tabela 5.4, constata-se que 2 instâncias (8,3%) não foram resolvidas otimamente. O tempo computacional médio para resolução do PAP foi de 706,7 segundos com desvio-padrão de 1991,0 segundos, sendo que 79,1% das instâncias foram resolvidas com tempo inferior a 10 segundos. As duas instâncias que não foram resolvidas otimamente apresentaram um *gap* médio do CPLEX de 17,3%. O *gap* médio de relaxação linear foi de 57,0% com desvio-padrão de 21,7%. O tempo computacional para obtenção da relaxação linear foi inferior a 1 segundo para todas as instâncias.

Na Tabela 5.5, observa-se que 15,6% das instâncias (5 instâncias) não foram resolvidas otimamente. O tempo computacional médio para resolução destas instâncias foi de 1158,7 segundos com desvio-padrão de 2603,8 segundos. Das 32 instâncias, 18 (56,3%) foram resolvidas com tempo inferior a 10 segundos. As instâncias que não

Tabela 5.3: Resultados das instâncias reais com o modelo PED

Inst.	NPd	NPe	T(s)	G(%)	Custo	R(%)	TR(s)
IAP025	212	5	153,0		49.148,8	100,0	1,0
IAP026	220	5	7205,0	5,6	382.614,9	100,0	0,0
IAP027	219	5	2347,0		21.509,9	100,0	0,0
IAP028	228	5	4,0		388,0	100,0	0,0
IAP029	226	5	1,0		9.481,0	62,8	0,0
IAP030	260	5	5,0		97,9	100,0	1,0
IAP031	290	7	7214,0	16,9	104.113,1	100,0	1,0
IAP032	311	7	7209,0	16,8	100.486,4	100,0	2,0
IAP033	336	7	315,0		25.151,9	91,7	2,0
IAP034	320	7	6,0		345,8	30,3	2,0
IAP035	360	7	15,0		0,0	0,0	3,0
IAP036	432	10	7228,0	87,4	319.794,0	100,0	6,0
IAP037	445	10	2521,0		19.244,8	100,0	6,0
IAP038	486	10	58,0		9.578,9	63,1	6,0
IAP039	460	10	519,0		1.536,3	100,0	8,0
IAP040	258	5	10,0		3.887,9	100,0	1,0
IAP041	220	5	7210,0	1,4	23.034,2	17,2	1,0
IAP042	256	5	47,0		3.324,1	30,2	1,0
IAP043	194	5	4,0		1.368,2	100,0	1,0
IAP044	264	5	590,0		3.848,3	25,5	1,0
IAP045	217	5	4,0		1.067,9	100,0	0,0
IAP046	240	5	12,0		2.008,3	80,0	0,0
IAP047	256	5	49,0		2.590,0	46,1	1,0
IAP048	356	7	464,0		11.042,4	100,0	2,0
IAP049	314	7	130,0		21.783,9	12,1	2,0
IAP050	306	7	180,0		11.406,0	26,4	1,0
IAP051	343	7	7211,0	2,0	4.891,6	41,4	2,0
IAP052	332	7	33,0		2.032,8	80,2	3,0
IAP053	478	10	6965,0		11.216,1	100,0	7,0
IAP054	452	10	7213,0	0,1	4.692,3	50,5	6,0
IAP055	483	10	7225,0	19,3	4.916,2	41,7	7,0
IAP056	496	10	7220,0	11,5	4.598,3	60,9	10,0

foram resolvidas otimamente apresentaram um *gap* médio do CPLEX de 60,0% com desvio-padrão de 22,3%. Assim como no modelo PED, a instância IAP036 apresentou um *gap* do CPLEX superior a 80%. O *gap* médio de relaxação linear foi de 67,5% com desvio-padrão de 32,4%, sendo que 43,8% das instâncias apresentaram um *gap* de relaxação linear superior a 90%. O tempo computacional para obtenção da relaxação linear foi inferior a 2 segundos para todas as instâncias.

As Tabelas 5.6 e 5.7 apresentam os resultados das instâncias artificiais e reais do modelo PAD-Bk, respectivamente. A estrutura destas tabelas é idêntica à da tabela 5.4.

Observa-se na Tabela 5.6 que todas as instâncias artificiais foram resolvidas otimamente, sendo que 80% das instâncias (19 instâncias) foram resolvidas com um tempo computacional inferior a 10 segundos. O *gap* médio de relaxação linear foi de

Tabela 5.4: Resultados das instâncias artificiais com o modelo PAD-N

Inst.	NPd	NPe	NPa	T(s)	G(%)	Custo	R(%)	TR(s)
IAP001	20	2	7	0,0		62.154,7	58,4	0,0
IAP002	25	3	14	0,0		69.119,2	34,3	0,0
IAP003	30	3	15	1,0		16.340,0	89,3	0,0
IAP004	30	4	13	1,0		37.514,8	60,2	0,0
IAP005	40	3	17	1,0		30.204,0	85,3	1,0
IAP006	40	4	28	0,0		116.658,2	9,1	0,0
IAP007	40	5	21	1,0		293.785,1	38,0	0,0
IAP008	40	6	27	4,0		168.688,9	77,7	0,0
IAP009	45	3	23	1,0		32.591,4	71,8	0,0
IAP010	45	4	20	0,0		38.822,9	62,6	1,0
IAP011	45	5	28	0,0		49.773,0	22,9	0,0
IAP012	45	6	21	2,0		22.097,7	60,3	0,0
IAP013	50	3	21	1,0		38.980,6	56,4	0,0
IAP014	50	4	25	1,0		63.343,0	49,8	0,0
IAP015	50	5	27	1,0		79.951,9	18,3	1,0
IAP016	50	6	25	1,0		53.178,3	65,7	0,0
IAP017	100	3	26	2,0		70.096,8	52,4	0,0
IAP018	100	4	37	6,0		172.981,2	25,5	0,0
IAP019	100	5	34	520,0		94.951,9	76,4	1,0
IAP020	100	6	32	3,0		46.931,9	60,4	0,0
IAP021	100	7	49	7208,0	13,4	132.980,6	71,1	0,0
IAP022	100	8	46	1676,0		122.660,1	59,6	1,0
IAP023	100	9	49	320,0		126.614,9	75,8	0,0
IAP024	100	10	47	7210,0	21,1	144.081,4	86,8	0,0

61,2% com desvio-padrão de 21,0%. O tempo computacional para obtenção da solução de relaxação linear foi inferior a 1 segundo para todas as instâncias.

Constata-se através da Tabela 5.7 que apenas 4 das 32 instâncias reais (12,5%) não foram resolvidas otimamente. Destas 4 instâncias, 3 apresentaram *gap* do CPLEX inferior a 6%. Assim como nos modelos anteriores, a instância IAP036 apresentou um *gap* do CPLEX acima de 80%, mas a solução obtida com o modelo PAD-Bk foi melhor que as soluções obtidas com os outros dois modelos. O tempo computacional médio para resolução do PAP com o modelo PAD-Bk foi de 945,8 segundos com desvio-padrão de 2368,7 segundos. Das 32 instâncias reais, 14 (43,8%) foram resolvidas com um tempo computacional inferior a 10 segundos. O *gap* médio de relaxação linear foi de 70,6% com desvio-padrão de 33,1%, sendo que 16 instâncias (50%) apresentaram um *gap* de relaxação linear acima de 90%. O tempo computacional para obtenção da solução de relaxação linear foi inferior a 3 segundos para todas as instâncias.

As Tabelas 5.8 e 5.9 apresentam os resultados das instâncias artificiais e reais do modelo PAD-Bd, respectivamente. A estrutura destas tabelas é idêntica à da tabela 5.4.

Na Tabela 5.8, observa-se que 22 das 24 instâncias foram resolvidas otimamente

Tabela 5.5: Resultados das instâncias reais com o modelo PAD-N

Inst.	NPd	NPe	NPa	T(s)	G(%)	Custo	R(%)	TR(s)
IAP025	212	5	51	7206,0	55,7	49.046,2	99,5	1,0
IAP026	220	5	59	7206,0	63,5	382.614,9	99,4	0,0
IAP027	219	5	50	22,0		21.509,9	98,5	0,0
IAP028	228	5	51	0,0		388,0	100,0	0,0
IAP029	226	5	75	1,0		9.481,0	62,8	0,0
IAP030	260	5	58	1,0		97,9	100,0	1,0
IAP031	290	7	70	7205,0	77,9	104.010,5	99,1	0,0
IAP032	311	7	77	7210,0	20,0	100.486,4	99,4	0,0
IAP033	336	7	84	33,0		25.151,9	91,3	1,0
IAP034	320	7	90	0,0		345,8	30,3	1,0
IAP035	360	7	74	1,0		0,0	0,0	0,0
IAP036	432	10	109	7205,0	83,1	316.984,3	99,6	1,0
IAP037	445	10	98	313,0		19.244,8	99,3	1,0
IAP038	486	10	133	3,0		9.578,9	63,1	1,0
IAP039	460	10	95	15,0		1.536,3	95,5	1,0
IAP040	258	5	52	2,0		3.887,9	85,0	1,0
IAP041	220	5	39	3,0		23.034,2	14,8	0,0
IAP042	256	5	70	0,0		3.324,1	30,2	0,0
IAP043	194	5	44	0,0		1.368,2	100,0	0,0
IAP044	264	5	66	5,0		3.848,3	25,5	0,0
IAP045	217	5	67	0,0		1.067,9	40,7	0,0
IAP046	240	5	68	2,0		2.008,3	80,0	1,0
IAP047	256	5	47	3,0		2.590,0	46,1	0,0
IAP048	356	7	77	39,0		11.042,4	94,7	1,0
IAP049	314	7	66	4,0		21.783,9	9,5	1,0
IAP050	306	7	79	4,0		11.406,0	26,4	0,0
IAP051	343	7	96	18,0		4.891,6	41,4	0,0
IAP052	332	7	92	4,0		2.032,8	80,2	0,0
IAP053	478	10	91	346,0		11.216,1	94,8	1,0
IAP054	452	10	115	3,0		4.692,3	50,5	2,0
IAP055	483	10	133	118,0		4.916,2	41,7	2,0
IAP056	496	10	112	106,0		4.598,3	60,9	2,0

com o modelo PAD-Bd. O *gap* médio do CPLEX para as duas instâncias não resolvidas otimamente foi de 9,4%. O tempo computacional médio para a resolução das instâncias artificiais foi de 651,2 segundos com um desvio-padrão de 1981,6 segundos. O *gap* médio de relaxação linear foi de 57,0% com desvio-padrão de 21,7%. O tempo computacional para obtenção da solução de relaxação linear foi inferior a 1 segundo para todas as instâncias.

Nota-se pela Tabela 5.9 que 15,6% das instâncias reais (5 instâncias) não foram resolvidas otimamente. O tempo computacional médio para a resolução das instâncias reais pelo modelo PAD-Bd foi de 1307,2 segundos com desvio-padrão de 2640,2 segundos. Das 32 instâncias, 18 (56,3%) foram resolvidas com um tempo inferior a 10 segundos. As cinco instâncias que não foram resolvidas otimamente apresentaram um *gap* médio do CPLEX de 61,5% com desvio padrão de 18,2%. O *gap* médio de rela-

Tabela 5.6: Resultados das instâncias artificiais com o modelo PAD-Bk

Inst.	NPd	NPe	NPa	T(s)	G(%)	Custo	R(%)	TR(s)
IAP001	20	2	7	1,0		62.154,7	75,1	0,0
IAP002	25	3	14	0,0		69.119,2	62,0	0,0
IAP003	30	3	15	0,0		16.340,0	89,3	0,0
IAP004	30	4	13	1,0		37.514,8	67,0	0,0
IAP005	40	3	17	2,0		30.204,0	86,8	1,0
IAP006	40	4	28	0,0		116.658,2	15,6	0,0
IAP007	40	5	21	1,0		293.785,1	38,7	0,0
IAP008	40	6	27	5,0		168.688,9	77,9	0,0
IAP009	45	3	23	1,0		32.591,4	73,2	1,0
IAP010	45	4	20	1,0		38.822,9	63,0	0,0
IAP011	45	5	28	1,0		49.773,0	22,9	0,0
IAP012	45	6	21	1,0		22.097,7	62,6	0,0
IAP013	50	3	21	1,0		38.980,6	64,6	1,0
IAP014	50	4	25	1,0		63.343,0	51,8	0,0
IAP015	50	5	27	1,0		79.951,9	18,5	0,0
IAP016	50	6	25	1,0		53.178,3	66,9	0,0
IAP017	100	3	26	2,0		70.096,8	58,3	1,0
IAP018	100	4	37	5,0		172.981,2	28,4	0,0
IAP019	100	5	34	90,0		94.951,9	76,4	0,0
IAP020	100	6	32	4,0		46.931,9	73,9	1,0
IAP021	100	7	49	75,0		132.980,6	72,0	0,0
IAP022	100	8	46	136,0		122.660,1	61,2	0,0
IAP023	100	9	49	55,0		126.614,9	76,0	1,0
IAP024	100	10	47	2480,0		144.081,4	86,8	0,0

xação linear foi de 67,5% com desvio-padrão de 32,4%, sendo que 14 das 32 instâncias apresentaram um *gap* de relaxação linear acima de 90%. O tempo computacional para obtenção da relaxação linear foi inferior a 2 segundos para todas as instâncias.

As Tabelas 5.10 e 5.11 apresentam os resultados das instâncias artificiais e reais do modelo PAD-B2, respectivamente. A estrutura destas tabelas também é idêntica à da tabela 5.4.

Na Tabela 5.10, observa-se que 22 das 24 instâncias artificiais foram resolvidas otimamente. O tempo computacional médio para resolução das instâncias artificiais pelo modelo PAD-B2 foi de 623,8 segundos com desvio-padrão de 1985,7 segundos. Das 24 instâncias, 80% (19 instâncias) foram resolvidas com um tempo inferior a 10 segundos. O *gap* médio do CPLEX foi de 8,4% para as duas instâncias que não foram resolvidas otimamente. O *gap* médio de relaxação linear foi de 95,3% com desvio padrão de 7,3%. O tempo computacional para obtenção da solução de relaxação linear foi inferior a 1 segundo para todas as instâncias.

Constata-se através da tabela 5.11 que 84,4% das instâncias reais foram resolvidas otimamente (27 instâncias) pelo modelo PAD-B2. O tempo computacional médio para resolução das instâncias pelo modelo foi de 1164,3 segundos com desvio

Tabela 5.7: Resultados das instâncias reais com o modelo PAD-Bk

Inst.	NPd	NPe	NPa	T(s)	G(%)	Custo	R(%)	TR(s)
IAP025	212	5	51	96,0		49.046,2	100,0	1,0
IAP026	220	5	59	7206,0	2,0	382.614,9	100,0	0,0
IAP027	219	5	50	39,0		21.509,9	100,0	0,0
IAP028	228	5	51	1,0		388,0	100,0	0,0
IAP029	226	5	75	1,0		9.481,0	62,8	0,0
IAP030	260	5	58	1,0		97,9	100,0	0,0
IAP031	290	7	70	7214,0	5,4	104.010,5	100,0	0,0
IAP032	311	7	77	7208,0	5,2	100.486,4	100,0	1,0
IAP033	336	7	84	67,0		25.151,9	91,7	1,0
IAP034	320	7	90	2,0		345,8	30,3	1,0
IAP035	360	7	74	5,0		0,0	0,0	1,0
IAP036	432	10	109	7210,0	85,8	316.921,6	100,0	2,0
IAP037	445	10	98	360,0		19.244,8	100,0	2,0
IAP038	486	10	133	10,0		9.578,9	63,1	2,0
IAP039	460	10	95	80,0		1.536,3	100,0	2,0
IAP040	258	5	52	5,0		3.887,9	100,0	1,0
IAP041	220	5	39	5,0		23.034,2	17,2	0,0
IAP042	256	5	70	1,0		3.324,1	30,2	0,0
IAP043	194	5	44	1,0		1.368,2	100,0	0,0
IAP044	264	5	66	4,0		3.848,3	25,5	0,0
IAP045	217	5	67	2,0		1.067,9	100,0	0,0
IAP046	240	5	68	5,0		2.008,3	80,0	0,0
IAP047	256	5	47	6,0		2.590,0	46,1	1,0
IAP048	356	7	77	98,0		11.042,4	100,0	1,0
IAP049	314	7	66	8,0		21.783,9	12,1	1,0
IAP050	306	7	79	14,0		11.406,0	26,4	1,0
IAP051	343	7	96	60,0		4.891,6	41,4	2,0
IAP052	332	7	92	13,0		2.032,8	80,2	1,0
IAP053	478	10	91	150,0		11.216,1	100,0	1,0
IAP054	452	10	115	13,0		4.692,3	50,5	2,0
IAP055	483	10	133	202,0		4.916,2	41,7	2,0
IAP056	496	10	112	180,0		4.598,3	60,9	3,0

padrão de 2601,4 segundos, sendo que 13 das 32 instâncias (40,6%) foram resolvidas em um tempo computacional inferior a 10 segundos. As cinco instâncias que não foram resolvidas otimamente apresentaram um *gap* médio do CPLEX de 51,8% com desvio padrão de 24,1%. Assim como nos outros quatro modelos a instância IAP036 teve um *gap* do CPLEX acima de 80%. O *gap* médio de relaxação linear foi de 78,4% com desvio padrão de 29,3%. O tempo computacional para obtenção da solução de relaxação linear foi inferior a 2 segundos para todas as instâncias.

Observou-se, ainda, através das tabelas anteriores que o modelo PED não é necessariamente igual aos demais modelos. Tal fato, pode ser observado na instância IAP025. A solução ótima desta instância para o modelo PED foi de \$49.148,8, enquanto a solução ótima dos modelos com painéis indexados nos padrões (PAD-N a PAD-B2) tem como solução ótima \$49.046,2. Analisando esta instância observou-se que o modelo

Tabela 5.8: Resultados das instâncias artificiais com o modelo PAD-Bd

Inst.	NPd	NPe	NPa	T(s)	G(%)	Custo	R(%)	TR(s)
IAP001	20	2	7	0,0		62.154,7	58,4	1,0
IAP002	25	3	14	1,0		69.119,2	34,3	0,0
IAP003	30	3	15	0,0		16.340,0	89,3	0,0
IAP004	30	4	13	1,0		37.514,8	60,2	0,0
IAP005	40	3	17	2,0		30.204,0	85,3	1,0
IAP006	40	4	28	1,0		116.658,2	9,1	0,0
IAP007	40	5	21	1,0		293.785,1	38,0	0,0
IAP008	40	6	27	6,0		168.688,9	77,7	0,0
IAP009	45	3	23	1,0		32.591,4	71,8	0,0
IAP010	45	4	20	1,0		38.822,9	62,6	1,0
IAP011	45	5	28	1,0		49.773,0	22,9	0,0
IAP012	45	6	21	1,0		22.097,7	60,3	0,0
IAP013	50	3	21	2,0		38.980,6	56,4	0,0
IAP014	50	4	25	1,0		63.343,0	49,8	0,0
IAP015	50	5	27	2,0		79.951,9	18,3	1,0
IAP016	50	6	25	1,0		53.178,3	65,7	0,0
IAP017	100	3	26	4,0		70.096,8	52,4	0,0
IAP018	100	4	37	9,0		172.981,2	25,5	0,0
IAP019	100	5	34	244,0		94.951,9	76,4	0,0
IAP020	100	6	32	8,0		46.931,9	60,4	0,0
IAP021	100	7	49	7206,0	13,7	132.980,6	71,1	0,0
IAP022	100	8	46	638,0		122.660,1	59,6	0,0
IAP023	100	9	49	291,0		126.614,9	75,8	0,0
IAP024	100	10	47	7207,0	5,2	144.081,4	86,8	0,0

com painéis indexados nos pedidos podem formar apenas 6 painéis com o padrão 1, sendo este o número máximo de painéis com este tipo de padrão, pois há nesta instância somente seis pedidos com esta composição química. Mas a solução ótima desta instância com os modelos com painéis indexados nos padrões utilizam 7 painéis com o padrão 1. Então as soluções com os modelos com painéis indexados nos padrões serão melhores ou iguais às soluções obtidas pelo modelo com painéis indexados nos pedidos. Um ponto interessante que cabe salientar aqui é que tal fato aconteceu apenas com a instância IAP025.

As Tabelas 5.12 e 5.13 apresentam o resumo dos resultados da aplicação dos cinco modelos à resolução do Problema de Agrupamento de Pedidos (PAP). Nestas tabelas, a linha “ $T_m(s)$ ” apresenta o tempo computacional médio para resolução das 56 instâncias e na linha “DP(T(s))” apresenta o desvio-padrão do tempo computacional. Já as linhas “ $R_m(\%)$ ” e “DP(R(%))” apresentam o *gap* médio de relaxação linear para as 56 instâncias e o desvio-padrão deste *gap*, respectivamente. As linhas “ $G_m(\%)$ ” e “DP(G(%))” apresentam o *gap* médio apresentado pelo CPLEX para as instâncias que não foram resolvidas otimamente e o desvio padrão deste *gap*, respectivamente. A linha “NMS” apresenta o número de melhores soluções que foram encontradas e na

Tabela 5.9: Resultados das instâncias reais com o modelo PAD-Bd

Inst.	NPd	NPe	NPa	T(s)	G(%)	Custo	R(%)	TR(s)
IAP025	212	5	51	7206,0	50,9	49.046,2	99,5	0,0
IAP026	220	5	59	7213,0	70,2	382.614,9	99,4	0,0
IAP027	219	5	50	87,0		21.509,9	98,5	1,0
IAP028	228	5	51	0,0		388,0	100,0	0,0
IAP029	226	5	75	1,0		9.481,0	62,8	1,0
IAP030	260	5	58	1,0		97,9	100,0	0,0
IAP031	290	7	70	7204,0	49,2	104.010,5	99,1	1,0
IAP032	311	7	77	7212,0	43,8	100.486,4	99,4	1,0
IAP033	336	7	84	253,0		25.151,9	91,3	1,0
IAP034	320	7	90	2,0		345,8	30,3	1,0
IAP035	360	7	74	2,0		0,0	0,0	1,0
IAP036	432	10	109	7206,0	93,2	317.106,9	99,6	2,0
IAP037	445	10	98	4182,0		19.244,8	99,3	2,0
IAP038	486	10	133	5,0		9.578,9	63,1	2,0
IAP039	460	10	95	224,0		1.536,3	95,5	1,0
IAP040	258	5	52	4,0		3.887,9	85,0	0,0
IAP041	220	5	39	5,0		23.034,2	14,8	0,0
IAP042	256	5	70	2,0		3.324,1	30,2	0,0
IAP043	194	5	44	0,0		1.368,2	100,0	0,0
IAP044	264	5	66	5,0		3.848,3	25,5	1,0
IAP045	217	5	67	0,0		1.067,9	40,7	0,0
IAP046	240	5	68	2,0		2.008,3	80,0	1,0
IAP047	256	5	47	8,0		2.590,0	46,1	1,0
IAP048	356	7	77	218,0		11.042,4	94,7	0,0
IAP049	314	7	66	9,0		21.783,9	9,5	0,0
IAP050	306	7	79	4,0		11.406,0	26,4	0,0
IAP051	343	7	96	32,0		4.891,6	41,4	1,0
IAP052	332	7	92	6,0		2.032,8	80,2	1,0
IAP053	478	10	91	500,0		11.216,1	94,8	1,0
IAP054	452	10	115	7,0		4.692,3	50,5	2,0
IAP055	483	10	133	67,0		4.916,2	41,7	2,0
IAP056	496	10	112	162,0		4.598,3	60,9	2,0

linha abaixo seu respectivo valor em termos percentuais. E a linha “*NIRO*” apresenta o número de instâncias resolvidas otimamente pelo modelo e na linha seguinte o seu respectivo valor em termos percentuais.

Nota-se, pela análise das Tabelas 5.12 e 5.13, que o modelo PAD-Bk teve um desempenho superior aos demais modelos. Dos modelos testados, apenas o modelo PAD-Bk conseguiu resolver otimamente todas as instâncias artificiais. Além disso, o modelo PAD-Bk foi o modelo que resolveu as instâncias com o menor tempo computacional médio (591,6 segundos). O modelo PAD-Bk foi o que obteve o menor custo para a instância IAP036, única instância que todos os modelos apresentaram um *gap* do CPLEX superior a 80%. Todos os modelos apresentaram *gaps* de relaxação linear muito ruins, sendo que o modelo PAD-B2 foi o que obteve pior resultado médio (85,7% de *gap* de relaxação linear). As soluções com menores custos foram todas apresentadas

Tabela 5.10: Resultados das instâncias artificiais com o modelo PAD-B2

Inst.	NPd	NPe	NPa	T(s)	G(%)	Custo	R(%)	TR(s)
IAP001	20	2	7	0,0		62.154,7	100,0	0,0
IAP002	25	3	14	0,0		69.119,2	100,0	0,0
IAP003	30	3	15	0,0		16.340,0	100,0	0,0
IAP004	30	4	13	1,0		37.514,8	95,6	0,0
IAP005	40	3	17	1,0		30.204,0	90,3	0,0
IAP006	40	4	28	1,0		116.658,2	100,0	0,0
IAP007	40	5	21	0,0		293.785,1	100,0	1,0
IAP008	40	6	27	5,0		168.688,9	94,4	0,0
IAP009	45	3	23	1,0		32.591,4	100,0	0,0
IAP010	45	4	20	1,0		38.822,9	100,0	0,0
IAP011	45	5	28	1,0		49.773,0	100,0	1,0
IAP012	45	6	21	1,0		22.097,7	68,1	0,0
IAP013	50	3	21	1,0		38.980,6	100,0	0,0
IAP014	50	4	25	0,0		63.343,0	100,0	0,0
IAP015	50	5	27	1,0		79.951,9	100,0	0,0
IAP016	50	6	25	1,0		53.178,3	100,0	0,0
IAP017	100	3	26	2,0		70.096,8	100,0	0,0
IAP018	100	4	37	8,0		172.981,2	98,9	0,0
IAP019	100	5	34	108,0		94.951,9	90,4	1,0
IAP020	100	6	32	5,0		46.931,9	92,0	0,0
IAP021	100	7	49	7206,0	12,0	132.980,6	91,8	0,0
IAP022	100	8	46	292,0		122.660,1	92,8	0,0
IAP023	100	9	49	128,0		126.614,9	84,3	0,0
IAP024	100	10	47	7206,0	4,8	144.081,4	89,1	0,0

pele modelo PAD-Bk. Além disso, o modelo PAD-Bk foi o que resolveu o maior número de instâncias otimamente (92,9%).

5.3.3 Resultados - Heurísticas RF e FOpt

Para os testes com a heurística RF-FOpt utilizaram-se 10 instâncias-teste. As 10 instâncias foram escolhidas de acordo com o desempenho do modelo PAD-Bk na resolução destas instâncias. Utilizaram-se as instâncias que não foram resolvidas otimamente (IAP026, IAP031, IAP032 e IAP036) e as instâncias cujo tempo necessário para obtenção da solução foi superior a 98 segundos (IAP024, IAP037, IAP048, IAP053, IAP055 e IAP056). Como o modelo PAD-Bk apresentou melhor desempenho que os demais, adotou-se este modelo como modelo padrão para a heurística desenvolvida.

As Tabelas que apresentam os resultados para a heurística RF-FOpt com as diferentes estratégias *relax-and-fix* adotadas se encontram no Apêndice A.

A Tabela 5.14 apresenta um resumo dos resultados da aplicação da heurística RF-FOpt à resolução do PAP, usando-se os diferentes tipos de estratégias descritas anteriormente. Nesta tabela, a coluna “Est. RF” apresenta a estratégia *relax-and-*

Tabela 5.11: Resultados das instâncias reais com o modelo PAD-B2

Inst.	NPd	NPe	NPa	T(s)	G(%)	Custo	R(%)	TR(s)
IAP025	212	5	51	7207,0	33,4	49.046,2	100,0	1,0
IAP026	220	5	59	7205,0	67,2	382.614,9	100,0	0,0
IAP027	219	5	50	21,0		21.509,9	100,0	0,0
IAP028	228	5	51	1,0		388,0	100,0	0,0
IAP029	226	5	75	2,0		9.481,0	65,3	0,0
IAP030	260	5	58	1,0		97,9	100,0	0,0
IAP031	290	7	70	7206,0	54,2	104.010,5	100,0	1,0
IAP032	311	7	77	7204,0	18,0	101.301,8	100,0	1,0
IAP033	336	7	84	40,0		25.151,9	91,7	0,0
IAP034	320	7	90	2,0		345,8	100,0	0,0
IAP035	360	7	74	3,0		0,0	0,0	0,0
IAP036	432	10	109	7208,0	86,1	317.106,9	100,0	1,0
IAP037	445	10	98	228,0		19.244,8	100,0	1,0
IAP038	486	10	133	6,0		9.578,9	65,7	2,0
IAP039	460	10	95	68,0		1.536,3	100,0	2,0
IAP040	258	5	52	5,0		3.887,9	100,0	1,0
IAP041	220	5	39	5,0		23.034,2	100,0	0,0
IAP042	256	5	70	2,0		3.324,1	32,6	0,0
IAP043	194	5	44	1,0		1.368,2	100,0	0,0
IAP044	264	5	66	15,0		3.848,3	25,5	0,0
IAP045	217	5	67	1,0		1.067,9	100,0	0,0
IAP046	240	5	68	6,0		2.008,3	80,0	0,0
IAP047	256	5	47	6,0		2.590,0	46,1	0,0
IAP048	356	7	77	68,0		11.042,4	100,0	1,0
IAP049	314	7	66	12,0		21.783,9	100,0	0,0
IAP050	306	7	79	12,0		11.406,0	26,4	0,0
IAP051	343	7	96	35,0		4.891,6	41,4	0,0
IAP052	332	7	92	12,0		2.032,8	80,2	0,0
IAP053	478	10	91	468,0		11.216,1	100,0	1,0
IAP054	452	10	115	24,0		4.692,3	52,3	1,0
IAP055	483	10	133	97,0		4.916,2	41,7	2,0
IAP056	496	10	112	88,0		4.598,3	60,9	2,0

Tabela 5.12: Resumo dos resultados obtidos com a aplicação dos modelos PED, PAD-N, PAD-Bk, PAD-Bd e Pad-B2 à resolução do PAP - Instâncias Artificiais (IAP001 a IAP024)

Medida de Desempenho	Modelo				
	PED	PAD-N	PAD-Bk	PAD-Bd	PAD-B2
$T_m(s)$	835,0	706,7	119,4	651,2	623,8
$DP(T(s))$	2.031,9	1.991,0	493,4	1.981,6	1.985,7
$R_m(\%)$	60,5	57,0	61,2	57,0	95,3
$DP(R(\%))$	21,4	21,7	21,0	21,7	7,3
$G_m(\%)$	0,5	1,4	0,0	0,8	0,7
$DP(G(\%))$	1,6	4,9	0,0	2,9	2,5
NMS	24	24	24	24	24
	100,0	100,0	100,0	100,0	100,0
$NIRO$	22	22	24	22	22
	91,7	91,7	100,0	91,7	91,7

Tabela 5.13: Resumo dos resultados obtidos com a aplicação dos modelos PED, PAD-N, PAD-Bk, PAD-Bd e Pad-B2 à resolução do PAP - Instâncias Reais(IAP025 a IAP056)

Medida de Desempenho	Modelo				
	PED	PAD-N	PAD-Bk	PAD-Bd	PAD-B2
$T_m(s)$	2.480,2	1.158,7	945,8	1.307,2	1.164,3
$DP(T(s))$	3.226,1	2.603,8	2.368,7	2.640,2	2.601,4
$R_m(\%)$	70,6	67,5	70,6	67,5	78,4
$DP(R(\%))$	33,1	32,4	33,1	32,4	29,3
$G_m(\%)$	5,0	9,4	3,1	9,6	8,1
$DP(G(\%))$	15,7	23,5	14,9	23,5	21,1
NMS	29	31	32	31	30
	90,6	96,9	100,0	96,9	93,8
$NIRO$	23	27	28	27	27
	71,9	84,4	87,5	84,4	84,4

fix utilizada e a coluna “Est. FOpt” apresenta a estratégia fix -and-optimize utilizada. As colunas “ D_m ” e “ $DP(D)$ ” apresentam, respectivamente, o desvio médio em relação à melhor solução encontrada pelos modelos para a instância e o desvio-padrão deste desvio. Já as colunas “ $T_m(s)$ ” e “ $DP(T)$ ” apresentam o tempo computacional médio para obtenção da solução e o desvio-padrão deste tempo computacional, respectivamente. E a coluna “ NMS ” apresenta o número de vezes que encontrou-se a melhor solução com a heurística. O desvio do custo total da solução obtida pela heurística em relação ao custo da melhor solução obtida para aquela instância (D) é calculado conforme a equação 5.39.

$$D = \frac{CT - CT^*}{CT^*} \times 100 \quad (5.39)$$

Analisando-se a Tabela 5.14, verifica-se que os melhores resultados foram obtidos combinando-se as estratégias RF-zt1b/FOpt-ztpf, RF-zp1f/FOpt-ztpf e RF-zp1f/FOpt-zptf. O maior número de vezes que a melhor solução foi encontrada aconteceu quando combinou-se as estratégias RF-zt1b/FOpt-ztpf (6 de 10 instâncias). Mas as combinações que apresentaram o menor desvio médio em relação a melhor solução encontrada foram as combinações RF-zp1f/FOpt-ztpf e RF-zp1f/FOpt-zptf.

Analisando-se, ainda, a Tabela 5.14 verifica-se que usando-se as estratégias FOpt-zt1f e FOpt-zp1f, o tempo computacional foi inferior que o tempo computacional obtido usando-se as estratégias FOpt-ztpf e FOpt-zptf. Este resultado era esperado, pois as estratégias FOpt-ztpf e FOpt-zptf são combinação das estratégias FOpt-zt1f e FOpt-zp1f.

Por obterem os melhores resultados para a heurística RF-FOpt, apresenta-

Tabela 5.14: Resumo dos Resultados da Heurística RF-FOpt

Est. RF	Est. FOpt	D_m	DP(D)	$T_m(s)$	DP(T)	NMS
RF-zt1f	FOpt-zt1f	64,4	64,4	730,8	1511,2	0
	FOpt-zp1f	12,5	24,9	1.035,6	1.362,3	1
	FOpt-ztpf	5,8	9,3	2.786,1	2.518,9	3
	FOpt-zptf	5,2	8,8	1.708,5	2.133,3	3
RF-zt2f	FOpt-zt1f	171,3	473,2	1.220,4	1.038,5	1
	FOpt-zp1f	173,0	470,8	1.001,1	831,3	1
	FOpt-ztpf	13,5	16,2	1.814,1	1.377,2	3
	FOpt-zptf	17,3	15,7	1.709,5	1.101,9	2
RF-zt1b	FOpt-zt1f	25,4	19,2	366,3	623,5	0
	FOpt-zp1f	7,8	11,5	481,1	496,6	2
	FOpt-ztpf	3,2	8,6	1.386,5	1.130,4	6
	FOpt-zptf	3,3	8,6	1.123,6	1.131,6	5
RF- τ t1f	FOpt-zt1f	108,6	103,8	149,9	125,4	0
	FOpt-zp1f	96,0	88,4	935,1	909,0	0
	FOpt-ztpf	92,0	88,6	950,5	701,4	0
	FOpt-zptf	91,5	88,6	774,7	843,5	0
RF- τ t2f	FOpt-zt1f	395,4	1.096,5	504,2	335,4	0
	FOpt-zp1f	426,0	1.088,4	721,6	715,6	0
	FOpt-ztpf	389,0	1.098,7	986,8	519,9	1
	FOpt-zptf	425,8	1.088,5	947,1	500,1	0
RF-zp1f	FOpt-zt1f	5,4	5,1	1.155,9	989,0	2
	FOpt-zp1f	26,9	26,5	737,7	692,1	1
	FOpt-ztpf	2,8	4,4	1.807,9	1.986,9	4
	FOpt-zptf	3,0	4,2	2.222,4	2.149,8	3

se na Tabela 5.15, os resultados para as combinações de estratégias RF-zt1b/FOpt-ztpf e RF-zp1f/FOpt-ztpf. A estrutura desta tabela é definida como se segue. A coluna “Inst.” apresenta a instância resolvida. A coluna “ C_{MP} ” apresenta o custo obtido com a resolução da instância pelo modelo PAD-Bk e a coluna “ $T_{MP}(s)$ ” o tempo computacional necessário para a obtenção desta solução. A coluna “ C_{RF} ” apresenta o custo total da solução obtida pela aplicação apenas da heurística RF e a coluna “ $T_{RF}(s)$ ” o tempo computacional para obtenção desta solução. As colunas “ CT ” e “ $T(s)$ ” apresentam, respectivamente, o custo total da solução obtida pela heurística e o tempo computacional gasto para obtenção desta solução por este modelo. E a coluna “ $D(\%)$ ” apresenta o desvio do custo total da solução obtida pela heurística em relação ao custo da melhor solução obtida para aquela instância. O valor de D é calculado conforme a equação 5.39.

Observando-se a Tabela 5.15, pode-se notar que o desempenho da heurística usando a combinação de estratégias RF-zt1b/FOpt-ztpf tem um desempenho levemente superior que a combinação de estratégias RF-zp1f/FOpt-ztpf para a maioria das instâncias testadas. O desempenho da combinação de estratégias RF-zt1b/FOpt-ztpf só é inferior para as instâncias IAP024 e IAP056. Além disso, a combinação de estraté-

Tabela 5.15: Resultados da Heurística RF-FOpt usando as combinações de estratégias RF-zt1b/FOpt-ztpf e RF-zp1f/FOpt-ztpf

Combinação das Estratégias RF-zt1b/FOpt-ztpf							
Inst	C_{MP}	$T_{MP}(s)$	C_{RF}	$T_{RF}(s)$	CT	$T(s)$	$D(\%)$
IAP024	144.081,4	2480	186.017,6	41	186.017,6	549	29,1
IAP026	382.614,9	7206	561.272,2	61	382.614,9	2358	0,0
IAP031	104.010,5	7214	115.210,5	74	104.010,5	657	0,0
IAP032	100.486,4	7208	147.190,6	33	100.486,4	577	0,0
IAP036	316.921,6	7210	461.458,9	575	320.444,2	3688	1,1
IAP037	19.244,8	360	32.844,1	133	19.244,8	1145	0,0
IAP048	11.042,4	98	17.212,1	33	11.042,4	148	0,0
IAP053	11.216,1	150	15.056,1	64	11.216,1	394	0,0
IAP055	4.916,2	202	5.592,2	111	4.952,2	2818	0,7
IAP056	4.598,3	180	4.654,3	119	4.654,3	1531	1,2
Combinação das Estratégias RF-zp1f/FOpt-ztpf							
Inst	C_{MP}	$T_{MP}(s)$	C_{RF}	$T_{RF}(s)$	CT	$T(s)$	$D(\%)$
IAP024	144.081,4	2480	176.292,4	120	165.095,1	388	14,6
IAP026	382.614,9	7206	611.875,0	223	385.563,0	1116	0,8
IAP031	104.010,5	7214	197.149,7	178	104.010,5	1044	0,0
IAP032	100.486,4	7208	122.019,3	463	107.350,4	1056	6,8
IAP036	316.921,6	7210	558.658,8	1274	320.453,8	7326	1,1
IAP037	19.244,8	360	29.273,3	749	19.244,8	1213	0,0
IAP048	11.042,4	98	13.372,1	72	11.042,4	188	0,0
IAP053	11.216,1	150	11.216,1	554	11.216,1	847	0,0
IAP055	4.916,2	202	5.175,4	1143	5.095,3	3024	3,6
IAP056	4.598,3	180	4.650,3	1214	4.650,3	1877	1,1

gias RF-zt1b/FOpt-ztpf consegue encontrar a solução ótima em 6 instâncias testadas contra 4 da combinação de estratégias RF-zp1f/FOpt-ztpf. Ambas combinações de estratégias conseguem encontrar boas soluções em tempos computacionais inferiores ao do modelo PAD-Bk em metade das instâncias testadas. Com relação às quatro instâncias que não foram resolvidas otimamente pelo modelo PAD-Bk, ambas estratégias conseguiram obter boas soluções em tempos computacionais inferiores à resolução do modelo. Tal fato só não aconteceu na resolução da instância IAP036 pela heurística RF-Opt, usando a estratégia RF-zp1f/FOpt-ztpf, onde o tempo computacional foi de 7326 segundos.

5.4 Conclusões

Neste capítulo, inicialmente, apresentou-se quatro modelos matemáticos para a resolução do problema de agrupamento de pedidos(PAP). Os quatro modelos apresentaram bom desempenho nos testes com as instâncias geradas à partir de relatórios gerenciais da ESB, comprovando que podem ser utilizados na prática. Mas, entre os quatro modelos, o modelo PAD-Bk foi o que apresentou o melhor desempenho.

Além dos modelos matemáticos, apresentaram-se, também, heurísticas baseadas nas técnicas *relax-and-fix* e *fix-and-optimize* para resolver o PAP. Utilizaram-se diferentes estratégias para o desenvolvimento destas heurísticas. Dentro das estratégias utilizadas, a estratégia que apresentou melhor desempenho foi a estratégia que combinava a estratégia *relax-and-fix* ztlb e a estratégia *fix-and-optimize* ztpf. Assim como os modelos matemáticos, as heurísticas também podem ser utilizadas na prática, visto que as heurísticas conseguem produzir boas soluções dentro de um tempo hábil para a tomada de decisões.

Capítulo 6

Problema Integrado de Agrupamento de Pedidos e Sequenciamento de Painelas

Neste capítulo são apresentados os modelos matemáticos que integram o problema de agrupamento de pedidos em painelas e o problema de sequenciamento destas painelas na MLC da ESB. Para fazer a integração utilizam-se os modelos PED e PAD-Bk apresentados no capítulo 5 para o problema de agrupamento de pedidos e o modelo P, apresentado no capítulo 4, para o problema de sequenciamento de painelas. Esses modelos são apresentados na seção 6.1.

Além dos modelos de PLIM para o Problema Integrado de Agrupamento de Pedidos e Sequenciamento de Painelas (PIAPSP), também são apresentadas duas heurísticas para resolver este mesmo problema. Essas heurísticas são apresentadas na seção 6.2.

A primeira heurística procura trabalhar com os modelos PAD-Bk para resolver o PAP e o modelo P para resolver o PSP. Nesta primeira heurística, busca-se resolver primeiramente o PAP e em seguida aplicar o modelo P aos pedidos gerados para obtenção da sequência de produção.

A segunda heurística é baseada nas técnicas *relax-and-fix* e *fix-and-optimize*. São adotadas quatro estratégias *relax-and-fix* e duas estratégias *fix-and-optimize* para resolução do PIAPSP.

6.1 Modelos Matemáticos

Elaborou-se dois modelos de programação linear inteira mista para resolver o PIAPSP. Adotaram-se como premissas na elaboração desses modelos que ao final de cada período t era realizada uma operação de *setup*.

No primeiro modelo, as variáveis z e x são indexadas nos pedidos. Este primeiro modelo incorpora ao modelo PED, definido no capítulo anterior, as restrições do modelo P para determinar o sequenciamento. Esse modelo é denominado aqui de modelo PIPED.

Os dados do modelo PIPED são os mesmos definidos para o modelo PED (seção 5.1.1) e para o modelo P (seção 4.1.4), incluindo-se o parâmetro r_t que representa a quantidade máxima de tempo que um período t pode utilizar de algum outro período $t + 1$.

Além das variáveis apresentadas para o modelo PED, o modelo PIPED inclui as seguintes variáveis de decisão:

- $x_{ij}^t = \begin{cases} 1, & \text{se o pedido } j \text{ é sequenciado logo após o pedido } i \text{ no período } t \\ 0, & \text{caso contrário} \end{cases}$
- $\Phi_{ij}^t =$ fluxo de tempo enviado do nó que representa o pedido i para o nó que representa o pedido j no período t (variável auxiliar).
- $F^t =$ variável livre que representa a quantidade de tempo que sobrou ou que foi utilizada a mais no período t . Se $F^t > 0$ implica que sobrou tempo no período t e este pode ser aproveitado no período $t + 1$. Por outro lado, se $F^t < 0$ implica que foi utilizado uma quantidade de tempo a mais e esta quantidade deve ser descontada no período $t + 1$.

O objetivo de se utilizar a variável F^t é poder controlar a quantidade de tempo que pode ser transferida de um período para o outro. Desta forma, quando sobra-se tempo na MLC em um determinado período, este tempo pode ser aproveitado no período seguinte. E quando não há tempo para processar todos os pedidos dentro de algum determinado período, poderá então ser utilizada uma quantidade limitada de tempo do período seguinte. Considerou-se, neste trabalho, que $F^0 = 0$. Mas, F^0 poderia assumir a quantidade de tempo que sobrou ou a quantidade de tempo que foi utilizada a mais no último período do planejamento anterior.

O modelo PIPED é apresentado a seguir:

$$\min \sum_{p \in P} \sum_{i \in I | a_{pi}=1} \sum_{t \in T} (v_{pi} y_{pi}^t + e_i^t y_{pi}^t) + \sum_{i \in I} L_i + \sum_{t \in T} \sum_{i=0}^n \sum_{i=0}^n c_{ij} x_{ij}^t \quad (6.1)$$

$$\text{s. à:} \quad \sum_{p \in P | a_{pi}=1} \sum_{t \in T} y_{pi}^t \leq 1 \quad \forall i \in P \quad (6.2)$$

$$\sum_{i \in I | a_{pi}=1} w_i y_{pi}^t - G_{max} z_p^t \leq 0 \quad \forall p \in P, \forall t \in T \quad (6.3)$$

$$\sum_{i \in P | a_{pi}=1} w_i y_{pi}^t - G_{min} z_p^t \geq 0 \quad \forall p \in P, \forall t \in T \quad (6.4)$$

$$\sum_{p \in P} (pr_p) z_p^t + \left(\sum_{p \in P} x_{0p}^t \right) s + F^t - F^{t-1} = q^t \quad \forall t \in T \quad (6.5)$$

$$\tau_i^t + \sum_{u=1}^t \sum_{p \in P | a_{pi}=1} y_{pi}^u \geq 1 \quad \forall i \in P, t \in T \quad (6.6)$$

$$L_i - \tau_i^t l_i^{t+1} \geq 0 \quad \forall i \in P, t \in T \quad (6.7)$$

$$\sum_{i=0}^n x_{ik}^t - z_k^t = 0 \quad \forall k \in P, \forall t \in T \quad (6.8)$$

$$\sum_{j=0}^n x_{kj}^t - z_k^t = 0 \quad \forall k \in P, \forall t \in T \quad (6.9)$$

$$\sum_{j=1}^n \sum_{t \in T} x_{0j}^t - \sum_{i=1}^n \sum_{t \in T} x_{i0}^t = 0 \quad (6.10)$$

$$\sum_{i=0}^n \Phi_{ik}^t - \sum_{j=0}^n \Phi_{kj}^t - pr_k(z_k^t) = 0 \quad \forall k \in P, \forall t \in T \quad (6.11)$$

$$\Phi_{ij}^t - l \cdot x_{ij}^t \leq 0 \quad \forall i, j = 0, \dots, n, \forall t \in T \quad (6.12)$$

$$x_{ij}^t \leq b_{ij} \quad \forall i, j = 1, \dots, n, \forall t \in T \quad (6.13)$$

$$y_{pi}^t, L_i \geq 0 \quad \forall p, i \in P, \forall t \in T \quad (6.14)$$

$$z_p^t \geq 0 \quad \forall p \in P, \forall t \in T \quad (6.15)$$

$$\tau_i^t \in \{0, 1\} \quad \forall i \in P, \forall t \in T \quad (6.16)$$

$$x_{ij}^t \in \{0, 1\} \quad \forall i, j = 0, \dots, n, \forall t \in T \quad (6.17)$$

$$\Phi_{ij}^t \geq 0 \quad \forall i, j = 0, \dots, n, \forall t \in T \quad (6.18)$$

$$F^t \in \mathbb{R} \quad \forall t \in T \quad (6.19)$$

$$F^t \geq -r_t \quad \forall t \in T \quad (6.20)$$

A função objetivo é representada pela equação (6.1) e busca minimizar os

custos de agrupamentos de pedidos em uma mesma panela, os custos de atraso e de estoque, os custos com geração de placas de mistura e os custos de *setups*. As restrições (6.2) garantem que o somatório das parcelas do pedido i entregues não seja superior a 1. As restrições (6.3) garantem que a quantidade de aço gerado pelo agrupamento dos pedidos em uma mesma panela não ultrapasse a capacidade máxima desta panela. Já as restrições (6.4) garantem que cada panela formada deve conter uma quantidade mínima de aço. As restrições (6.5) garantem que o tempo de processamento das panelas geradas em um período t mais o tempo utilizado para realização de *setups* não ultrapasse a quantidade máxima de tempo definida para aquele período. As restrições (6.6) definem em qual período o pedido i foi completamente entregue. As restrições (6.7) definem o valor do custo de atraso associado a entrega completa do pedido i .

As restrições (6.8) e (6.9) garantem que, em cada período, cada pedido tenha somente um pedido sucessor e somente um pedido antecessor, respectivamente. Tais restrições somente são válidas se a panela com a composição química predominante do pedido k tenha sido definida para o período. As restrições (6.10) garantem que o número de arcos que saem do nó central e que chegam no nó central sejam os mesmos. As restrições (6.11) garantem que, em cada período, o fluxo de tempo que chega no nó k menos o fluxo de tempo que sai deste mesmo nó é igual ao tempo de processamento da panela k . Tais restrições somente são válidas se o pedido k for definido para o período, caso contrário, o fluxo de tempo que chega no nó k menos o fluxo de tempo que sai deste mesmo nó deve ser igual 0. As restrições (6.12) garantem que o fluxo de tempo nos arcos não seja superior ao tempo máximo permitido entre dois *setups* consecutivos.

As restrições (6.13) a (6.20) definem o domínio das variáveis.

No segundo modelo, denominado aqui de PIPAD-Bk, as variáveis x e z são indexadas nos padrões de panela. Os dados deste modelo são os mesmos apresentados para o modelo PAD-Bk (seção 5.1.3) e para o modelo P (seção 4.1.4). Nesse modelo, incorpora-se ao modelo PAD-Bk as restrições do modelo P para determinar o sequenciamento das panelas em cada período.

Além das variáveis de decisão definidas para o modelo PAD-Bk, as demais variáveis de decisão do modelo PIPAD-Bk são definidas como se segue:

$$\bullet x_{kp,k'p'}^t = \begin{cases} 1, & \text{se a } k'\text{-ésima panela com padrão } p' \text{ é sequenciada logo após a} \\ & k\text{-ésima panela com padrão } p \text{ no período } t \\ 0, & \text{caso contrário} \end{cases}$$

- $\Phi_{kp,k'p'}^t$ = fluxo de tempo enviado do nó representado a k -ésima panela com padrão p para o nó representando a k' -ésima panela com padrão p' no período t (variável auxiliar).
- F^t = variável livre que representa a quantidade de tempo que sobrou ou que foi utilizada a mais no período t . Se $F^t > 0$ implica que sobrou tempo no período t e este pode ser aproveitado no período $t + 1$. Por outro lado, se $F^t < 0$ implica que foi utilizado uma quantidade de tempo a mais e esta quantidade deve ser descontada no período $t + 1$.

Seja $E = \{(k, p)\}$ o conjunto do pares ordenados (k, p) , onde (k, p) representa a k -ésima panela com padrão p e seja $A = E \cup \{(0, 0)\}$, onde o par ordenado $(0, 0)$ representa o nó central da rede. O modelo PIPAD-Bk é apresentado a seguir:

$$\min \sum_{p \in P} \sum_{i \in I | a_{pi}=1} \sum_{t \in T} (v_{pi} y_{pi}^t + e_i^t y_{pi}^t) + \sum_{i \in I} L_i + \sum_{t \in T} \sum_{(k,p) \in A} \sum_{k'p' \in A} c_{pp'} x_{kp,k'p'}^t \quad (6.21)$$

$$\text{s. à:} \quad \sum_{p \in P | a_{pi}=1} \sum_{t \in T} y_{pi}^t \leq 1 \quad \forall i \in I \quad (6.22)$$

$$\sum_{i \in I | a_{pi}=1} w_i y_{pi}^t - G_{max} \sum_{k=1}^{K[p]} z_{kp}^t \leq 0 \quad \forall p \in P, \forall t \in T \quad (6.23)$$

$$\sum_{i \in I | a_{pi}=1} w_i y_{pi}^t - G_{min} \sum_{k=1}^{K[p]} z_{kp}^t \geq 0 \quad \forall p \in P, \forall t \in T \quad (6.24)$$

$$\sum_{p \in P} pr_p \left(\sum_{k=1}^{k_p} z_{kp}^t \right) + \left(\sum_{(k,p) \in E} x_{00,kp}^t \right) s + F^t - F^{t-1} = q^t \quad t \in T \quad (6.25)$$

$$\tau_i^t + \sum_{u=1}^t \sum_{p \in P | a_{pi}=1} y_{pi}^u \geq 1 \quad \forall i \in P, t \in T \quad (6.26)$$

$$L_i - \tau_i^t \leq 0 \quad \forall i \in P, t \in T \quad (6.27)$$

$$\sum_{(k,p) \in A} x_{kp,k'p'}^t - z_{k'p'}^t = 0 \quad (k', p') \in E, t \in T \quad (6.28)$$

$$\sum_{(k',p') \in A} x_{kp,k'p'}^t - z_{kp}^t = 0 \quad (k, p) \in E, t \in T \quad (6.29)$$

$$\sum_{(k',p') \in E} \sum_{t \in T} x_{00,k'p'}^t - \sum_{(k,p) \in E} \sum_{t \in T} x_{kp,00}^t = 0 \quad (6.30)$$

$$\sum_{(k,p) \in A} \Phi_{kp,k''p''}^t - \sum_{(k',p') \in A} \Phi_{k''p'',k'p'}^t - pr_{k''} (z_{k''p''}^t) = 0 \quad (k'', p'') \in E \quad (6.31)$$

$$\forall t \in T$$

$$\Phi_{kp,k'p'}^t - l \cdot x_{kp,k'p'}^t \leq 0 \quad (k,p), (k',p') \in A, \quad (6.32)$$

$$\forall t \in T$$

$$z_{kp}^t - z_{k-1,p}^t \leq 0 \quad \forall p \in P, k = 2 \dots K[p] \quad (6.33)$$

$$\forall t \in T$$

$$x_{kp,k'p'}^t \leq b_{pp'} \quad (k,p), (k',p') \in E, \quad (6.34)$$

$$\forall t \in T$$

$$y_{pi}^t, L_i \geq 0 \quad \forall p, i \in P, \forall t \in T \quad (6.35)$$

$$z_p^t \geq 0 \quad \forall p \in P, \forall t \in T \quad (6.36)$$

$$\tau_i^t \in \{0, 1\} \quad \forall i \in P, \forall t \in T \quad (6.37)$$

$$x_{kp,k'p'}^t \in \{0, 1\} \quad (k,p), (k',p') \in A, \quad (6.38)$$

$$\forall t \in T$$

$$\Phi_{kp,k'p'}^t \geq 0 \quad (k,p), (k',p') \in A, \quad (6.39)$$

$$\forall t \in T$$

$$F^t \in \mathbb{R} \quad t \in T \quad (6.40)$$

$$F^t \geq -r_t \quad t \in T \quad (6.41)$$

A função objetivo é representada pela equação (6.21) e busca minimizar os custos de agrupamentos de pedidos em uma mesma panela, os custos de atraso e de estoque, os custos com geração de placas de mistura e os custos de *setups*. As restrições (6.22) garantem que o somatório das parcelas do pedido i entregues não seja superior a 1. As restrições (6.23) garantem que a quantidade de aço gerado pelo agrupamento dos pedidos em uma mesma panela não ultrapasse a capacidade máxima desta panela. Já as restrições (6.24) garantem que cada panela formada deve conter uma quantidade mínima de aço.

As restrições (6.25) garantem que o tempo de processamento das panelas geradas em um período t mais o tempo utilizado para realização de *setups* não ultrapasse a quantidade máxima de tempo definida para aquele período. As restrições (6.26) definem em qual período o pedido i foi completamente entregue. As restrições (6.27) definem o valor do custo de atraso associado a entrega completa do pedido i . As restrições (6.28) e (6.29) garantem que, em cada período, cada panela tenha somente uma panela sucessora e somente uma panela antecessora, respectivamente. Tais restrições somente são válidas se a k -ésima panela com padrão p tenha sido definida para o período. As restrições (6.30) garantem que o número de arcos que saem do nó central e que chegam no nó central sejam os mesmos.

As restrições (6.31) garantem que, em cada período, o fluxo de tempo que chega no nó (k, p) menos o fluxo de tempo que sai deste mesmo nó é igual ao tempo de processamento da k -ésima panela com o padrão p . Tais restrições somente são válidas se a k -ésima panela com padrão p for definida para o período, caso contrário, o fluxo de tempo que chega no nó (k, p) menos o fluxo de tempo que sai deste mesmo nó deve ser igual 0. As restrições (6.32) garantem que o fluxo de tempo nos arcos não seja superior ao tempo máximo permitido entre dois *setups* consecutivos. Por sua vez, as restrições (6.41) garantem que a k -ésima panela com o padrão p no período t , só será ativada se $(k - 1)$ -ésima panela com o padrão p no período t for processada. As restrições (6.34) a (6.41) definem o domínio das variáveis.

6.2 Heurísticas para o PIAPSP

Além dos modelos de PLIM, desenvolveram-se duas heurísticas para a resolução do PIAPSP. A primeira heurística consiste em fazer primeiramente o agrupamento dos pedidos em painéis e em seguida determinar o sequenciamento. Já a segunda heurística utiliza estratégias *relax-and-fix* e *fix-and-optimize* para resolver o problema. As duas heurísticas são detalhadas a seguir.

Como foi dito anteriormente, a primeira heurística busca primeiramente resolver o problema de agrupamento de pedidos e em seguida determinar o sequenciamento destes pedidos na MLC. A heurística realiza esta tarefa de forma iterativa, resolvendo o sequenciamento para cada período individualmente e atualizando a quantidade de tempo disponível do período corrente para o período seguinte. Para garantir a viabilidade da solução gerada, a cada iteração verifica-se o tempo total gasto no período. Este tempo não pode superar a quantidade de tempo disponível no período em mais de r_t unidades de tempo. Caso isto aconteça, é reduzido uma quantidade de tempo da disponibilidade de tempo do período atual e resolve-se novamente o PAP. Este processo é repetido enquanto uma solução viável não fosse gerada. A heurística termina quando é definido o sequenciamento do último período. Na heurística desenvolvida, o PAP é resolvido pelo modelo PAD-Bk descrito na seção 5.1.1 e o PSP é resolvido pelo modelo P descrito na seção 4.1.4.

A heurística, denominada aqui de agrupa primeiro e sequencia depois (HAPSD), é descrita de forma detalhada na Figura 6.1. Nesta heurística, $TG[tl]$ representa o tempo total gasto para processar todas as painéis no período tl e TTR representa o

tempo total reduzido do período corrente.

O algoritmo descrito na Figura 6.1, inicia na linha 1 resolvendo o PAP. Nas linhas 2 e 3, são atribuídos os valores iniciais para o período corrente tl e para o tempo total reduzido da disponibilidade de tempo $Q[tl]$ do período corrente. O processo iterativo começa na linha 4 e termina na linha 30. Na linha 5, o algoritmo monta o modelo PSP para o período corrente tl , na linha 6 este modelo é resolvido e na linha 7 calcula-se o tempo total gasto para processamento das tarefas naquele período. Na linha 8, verifica-se se o tempo total gasto é menor ou igual ao tempo disponível para o período. Em caso positivo, nas linhas 9 e 10 atualizam-se as disponibilidades de tempo nos períodos tl e $tl + 1$, na linha 11 fixam-se os valores da variável z_{kp}^t até o período corrente tl em seu valor corrente, na linha 12 resolve o PAP, caso não seja o último período, e na linha 13 atualiza-se o período corrente. Em caso negativo, na linha 15 calcula-se a diferença de tempo entre o tempo gasto e o tempo disponível do período corrente tl . Na linha 16, é verificado se a diferença menos o tempo total reduzido do período é inferior a r_t . Em caso afirmativo, na linha 17 retorna-se a disponibilidade de tempo no período corrente tl ao seu valor original, na linha 18 atualiza-se o valor do tempo total reduzido e nas linhas 19 a 23 é realizado o mesmo procedimento definido nas linhas 9 a 13. Em caso negativo, o tempo disponível no período corrente é reduzido em r_t unidades de tempo na linha 25 e este valor é armazenado no tempo total reduzido na linha 26. Atualizado estes valores, na linha 27 resolve-se novamente o PAP. O algoritmo é finalizado quando o último período é alcançado, retornando na linha 31 a solução encontrada.

A segunda heurística desenvolvida utiliza estratégias *relax-and-fix* e *fix-and-optimize* para resolver o PIAPSP. Adotaram-se aqui quatro estratégias *relax-and-fix* e duas estratégias *fix-and-optimize*. As soluções iniciais para a heurística FOpt são determinadas a partir da heurística RF. Esta heurística é denominada de PIRF-FOpt.

Em todas as estratégias utilizadas, as variáveis inteiras τ_i^t não são relaxadas e, juntamente, com as variáveis contínuas z_p^t , L_i , y_{pi}^t e Φ_{ij}^t , não têm seu valor fixado, ou seja, ficam livres para serem submetidas a nova re-otimização em cada iteração do método. As quatro estratégias *relax-and-fix* utilizadas na heurística PIRF-FOpt são definidas a seguir:

- **PIRF-xt1f:** Inicialmente mantêm-se a restrição de integralidade das variáveis x_{ij}^t no período $t = 1$ e relaxam-se as demais variáveis x_{ij}^t . Resolve-se o modelo. Atualiza-se o período t para $t + 1$. Nos passos seguintes, fixam-se as variáveis x_{ij}^t

```

Procedimento HAPSD()
1  Resolver PAP;
2   $tl \leftarrow 1$ ;
3   $TTR \leftarrow 0$ ;
4  Enquanto ( $tl \leq nt$ ) faça
5      Montar PSP para o período corrente  $tl$ ;
6      Resolver PSP para o período corrente  $tl$ ;
7      Calcula  $TG[tl]$ ;
8      Se ( $TG[tl] \leq Q[tl]$ ) então
9          Se ( $tl \neq nt$ ) então  $Q[tl + 1] \leftarrow Q[tl + 1] + Q[tl] - TG[tl]$ ;
10          $Q[tl] \leftarrow Q[tl] + Q[tl] - TG[tl]$ ;
11         Fixar os valores da variável  $z_{kp}^t$  até o período  $tl$  em seu valor corrente;
12         Se ( $tl \neq nt$ ) então Resolver PAP;
13          $tl \leftarrow tl + 1$ ;
14     Senão
15          $\Delta \leftarrow TG[tl] - Q[tl]$ ;
16         Se ( $\Delta - TTR < r_t$ ) então
17              $Q[tl] \leftarrow Q[tl] + TTR$ ;
18              $TTR \leftarrow 0$ ;
19             Se ( $tl \neq nt$ ) então  $Q[tl + 1] \leftarrow Q[tl + 1] + Q[tl] - TG[tl]$ ;
20              $Q[tl] \leftarrow Q[tl] + Q[tl] - TG[tl]$ ;
21             Fixar os valores da variável  $z_{kp}^t$  até o período  $tl$  em seu valor corrente;
22             Se ( $tl \neq nt$ ) então Resolver PAP;
23              $tl \leftarrow tl + 1$ ;
24     Senão
25          $Q[tl] \leftarrow Q[tl] - r_t$ ;
26          $TTR \leftarrow TTR + r_t$ ;
27         Resolver PAP;
28     fim-se
29 fim-se
30 fim-enquanto
31 Retornar solução;
fim-Procedimento

```

Figura 6.1: Pseudo-código da heurística agrupa primeiro e sequencia depois (HAPSD)

nos períodos $t - 1$ a 1 em seus valores atuais, mantêm-se a restrição de integralidade das variáveis x_{ij}^t no período t corrente e relaxam-se as demais variáveis x_{ij}^t . Daí, resolve-se novamente o modelo. O passo iterativo é feito até que o último período (nt) seja resolvido.

- **PIRF-xt2f:** Inicialmente mantêm-se a restrição de integralidade das variáveis x_{ij}^t no período $t = 1$ e $t = 2$ e relaxam-se as demais variáveis x_{ij}^t . Resolve-se o modelo. Atualiza-se o período t para $t + 1$. Nos passos seguintes, fixam-se as variáveis x_{ij}^t nos períodos $t - 1$ a 1 em seus valores atuais, mantêm-se a restrição de integralidade das variáveis x_{ij}^t no período t corrente e no período $t + 1$ e relaxam-se as demais variáveis x_{ij}^t . Daí, resolve-se novamente o modelo. O passo iterativo é feito até que o penúltimo período ($nt - 1$) seja resolvido.

- **PIRF-xt1b:** Inicialmente mantêm-se a restrição de integralidade das variáveis x_{ij}^t no período $t = nt$ e relaxam-se as demais variáveis x_{ij}^t . Resolve-se o modelo. Atualiza-se o período t para $t - 1$. Nos passos seguintes, fixam-se as variáveis x_{ij}^t nos períodos $t + 1$ a nt em seus valores atuais, mantêm-se a restrição de integralidade das variáveis x_{ij}^t no período t corrente e relaxam-se as demais variáveis x_{ij}^t . Daí, resolve-se novamente o modelo. O passo iterativo é feito até que o primeiro período seja resolvido.
- **PIRF-xt2b:** Inicialmente mantêm-se a restrição de integralidade das variáveis x_{ij}^t no período $t = nt$ e $t = nt - 1$ e relaxam-se as demais variáveis x_{ij}^t . Resolve-se o modelo. Atualiza-se o período t para $t - 1$. Nos passos seguintes, fixam-se as variáveis x_{ij}^t nos períodos $t + 1$ a nt em seus valores atuais, mantêm-se a restrição de integralidade das variáveis x_{ij}^t no período t corrente e no período $t - 1$ e relaxam-se as demais variáveis x_{ij}^t . Daí, resolve-se novamente o modelo. O passo iterativo é feito até que o período $t = 2$ seja resolvido.

E as duas estratégias *fix-and-optimize* utilizadas são:

- **PIFOpt-xt1f:** A partir da solução corrente S , fixam-se as variáveis x_{ij}^t em todos os períodos exceto o período corrente t (iniciando-se no período $t = 1$) e re-otimizam-se as variáveis do período t . Se a solução gerada for melhor que a solução corrente, a solução corrente deverá ser atualizada. O período t é atualizado para $t + 1$ a cada iteração e o processo é repetido até que as variáveis do último período nt sejam re-otimizadas.
- **PIFOpt-xt1b:** A partir da solução corrente S , fixam-se as variáveis x_{ij}^t em todos os períodos exceto o período corrente t (iniciando-se no período $t = nt$) e re-otimizam-se as variáveis do período t . Se a solução gerada for melhor que a solução corrente, a solução corrente deverá ser atualizada. O período t é atualizado para $t - 1$ a cada iteração e o processo é repetido até que as variáveis do primeiro período sejam re-otimizadas.

A heurística PIRF-FOpt é similar à heurística RF-FOpt apresentada no capítulo anterior para resolver o PAP. Isto pode ser evidenciado na Figura 6.2 que apresenta o pseudo-código da heurística PIRF-FOpt desenvolvida para resolver o PIAPSP. Igualmente a heurística RF-FOpt, a variável TC representa o tempo gasto pela heurística em algum determinado instante de tempo e o parâmetro RT_{max} representa o tempo máximo para a execução da heurística.

<p>Procedimento <i>PIRF-FOpt</i>()</p> <ol style="list-style-type: none"> 1 $S \leftarrow$ Gerar uma solução inicial usando estratégia PIRF; 2 $TC \leftarrow 0$; 3 Enquanto Houver melhora ou $TC < RT_{max}$ faça <li style="padding-left: 2em;">4 $S \leftarrow$ Re-otimiza usando estratégia PIFOpt; <li style="padding-left: 2em;">5 Atualiza TC; 6 fim-enquanto 7 Retorna S; <p>fim-Procedimento</p>
--

Figura 6.2: Pseudo-código da heurística PIRF-FOpt para resolução do PIAPSP

6.3 Resultados e Análise

Os testes com os dois modelos e com as heurísticas foram feitos com as 56 instâncias definidas no capítulo anterior.

Os modelos foram implementados em linguagem AMPL e resolvidos pelo pacote de otimização CPLEX 12.6. Utilizou-se a configuração padrão do CPLEX e estipulou-se um tempo de 7200 segundos para obtenção da solução ótima. Os testes foram feitos em um computador Intel Xeon X5690@3.47GHz, com 24 núcleos e 132 GB de memória RAM. Para ambos os modelos adotou-se $r_t = 134$ (maior tempo de processamento de uma tarefa mais o tempo de realização de uma operação de *setup*), ou seja, o período t não deve usar mais que 134 minutos do período $t + 1$.

Os testes com as heurísticas também foram feitos em um computador Intel Xeon X5690@3.47GHz, com 24 núcleos e 132 GB de memória RAM. Para a heurística HAPSD, definiu-se um tempo limite de 300 segundos para obtenção da solução pelos modelos de PLIM. Já para a heurística PIRF-FOpt, estipulou-se um tempo limite de 540 segundos para obtenção da solução ótima pelo modelo matemático e um tempo de execução máximo (RT_{max}) da heurística de 7200 segundos. Utilizou-se para a resolução do modelo o *software* de otimização CPLEX 12.6 com sua configuração padrão. Ambas as heurísticas foram desenvolvidas em linguagem AMPL.

6.3.1 Resultados - Modelos para o PIAPSP

As tabelas 6.1 e 6.2 apresentam os resultados para o modelo PIPED. As colunas dessas tabelas indicam a instância a ser resolvida (“Inst.”), o tempo computacional necessário para a obtenção da solução (“T(s)”), o *gap* apresentado pelo CPLEX ao final da resolução do modelo (“G(%)”), o custo total da solução obtida (“CT”), o custo da solução de relaxação linear do problema (“CTR”), o *gap* de relaxação linear (“R(%)”) e

o tempo necessário para a obtenção da solução relaxada (“TR(s)”).

Observando-se as tabelas 6.1 e 6.2, nota-se que apenas 16 das 56 instâncias (28,6%) foram resolvidas otimamente, sendo que o CPLEX apresentou um *gap* superior a 90% para 29 instâncias (51,8%). Este fato corrobora para que estas 29 instâncias apresentem custos relativamente altos. A maior parte dos custos dessas 29 instâncias vêm do atraso dos pedidos ou de pedidos que não foram entregues dentro do horizonte de planejamento.

Para as 16 instâncias que foram resolvidas otimamente, o CPLEX gastou em média 298,1 segundos com um desvio-padrão de 731,3 segundos para obtenção da solução ótima. O *gap* médio de relaxação linear para essas instâncias foi de 56,3% com um desvio-padrão de 13,6%.

O tempo computacional médio para obtenção da solução de relaxação linear foi de 423,0 segundos com um desvio-padrão de 845,1 segundos. Em 9 das 56 instâncias o tempo computacional para obtenção da solução de relaxação linear foi superior a 1000 segundos. Considerando-se apenas as instâncias artificiais, o tempo computacional médio para gerar a solução de relaxação linear é de 2,1 segundos com desvio-padrão de 2,7 segundos.

Tabela 6.1: Resultados para as instâncias Artificiais com o modelo PIPED

Inst.	T(s)	G(%)	CT	CTR	R(%)	TR(s)
IAP001	2		88.987,7	19.415,2	78,2	0
IAP002	22		110.329,2	31.893,8	71,1	0
IAP003	26		60.210,0	37.566,4	37,6	0
IAP004	202		87.670,8	47.957,7	45,3	1
IAP005	331		73.711,0	31.480,8	57,3	0
IAP006	20		173.000,2	69.678,6	59,7	0
IAP007	138		361.220,1	47.091,3	87,0	1
IAP008	95		135.258,4	72.731,6	46,2	0
IAP009	113		79.096,6	45.200,4	42,9	1
IAP010	162		98.350,5	49.330,0	49,8	1
IAP011	32		119.894,0	59.626,5	50,3	1
IAP012	3113		95.762,9	58.963,8	38,4	0
IAP013	128		84.719,3	36.734,2	56,6	1
IAP014	97		121.432,0	55.448,2	54,3	1
IAP015	144		150.891,9	47.042,1	68,8	1
IAP016	145		134.084,3	57.618,4	57,0	1
IAP017	7203	4,1	122.259,3	35.577,1		6
IAP018	7220	5,1	243.705,6	63.451,0		3
IAP019	7204	38,1	216.473,8	62.462,6		3
IAP020	7205	15,3	155.479,1	71.086,6		5
IAP021	7250	97,0	7.491.599,2	111.871,8		9
IAP022	7205	19,2	249.688,6	116.910,3		10
IAP023	7206	17,2	265.922,3	112.883,2		3
IAP024	7205	34,1	297.025,0	119.153,2		2

Tabela 6.2: Resultados para as instâncias Reais com o modelo PIPED

Inst.	T(s)	G(%)	CT	CTR	TR(s)
IAP025	7224	92,0	1.032.325,1	70.588,2	74
IAP026	7208	85,4	767.478,1	55.996,3	79
IAP027	7208	99,2	7.614.767,3	53.954,7	55
IAP028	7214	46,1	172.212,0	54.836,3	53
IAP029	7207	99,4	11.157.129,7	55.777,8	53
IAP030	7209	99,1	8.420.032,5	61.974,1	106
IAP031	7247	97,9	4.792.389,1	90.043,1	365
IAP032	7221	99,8	39.030.124,2	77.732,5	187
IAP033	7235	99,6	25.402.195,7	82.094,1	374
IAP034	7217	94,9	2.109.089,2	78.138,5	153
IAP035	8971	99,8	30.698.528,7	83.538,9	2531
IAP036	7258	99,9	125.103.550,0	126.584,5	2429
IAP037	7283	99,9	115.225.550,0	108.313,7	2045
IAP038	7307	99,9	122.504.250,0	117.751,9	1315
IAP039	7257	99,9	121.064.350,0	108.023,3	2826
IAP040	7225	99,0	6.038.777,7	60.090,8	96
IAP041	7243	53,8	132.741,1	33.761,3	222
IAP042	7210	93,2	1.101.922,2	61.786,6	54
IAP043	7225	45,5	105.279,2	46.151,5	65
IAP044	7249	99,0	6.137.756,2	58.549,5	79
IAP045	7223	91,6	944.353,8	54.644,8	81
IAP046	7247	94,7	1.188.732,5	54.140,2	46
IAP047	7228	97,7	2.092.885,4	53.968,7	254
IAP048	7269	99,3	9.831.078,7	78.075,4	279
IAP049	7220	98,7	6.967.683,0	61.598,7	199
IAP050	7265	99,0	7.117.630,1	77.402,3	252
IAP051	7250	98,1	4.858.872,9	77.814,2	252
IAP052	7288	99,2	8.831.276,0	74.925,5	380
IAP053	7359	99,7	31.886.700,0	93.852,1	2376
IAP054	7354	99,6	33.193.950,0	108.438,1	1037
IAP055	7274	99,6	35.645.400,0	109.455,8	1646
IAP056	7256	99,7	34.497.600,0	60.629,3	3673

A tabela 6.3 apresenta os resultados do modelo PIPAD-Bk. A estrutura dessa tabela é idêntica a da tabela 6.1. Para as instâncias reais (IAP025 a IAP056), o modelo PIPAD-Bk não foi capaz de apresentar uma solução viável dentro dos 7200 segundos. Por isso, apresentam-se aqui apenas os resultados para as instâncias artificiais (IAP001 a IAP024). Além disso, o modelo PIPAD-Bk também não foi capaz de gerar a solução de relaxação linear para as instâncias reais dentro dos 7200 segundos. Um dos fatores que contribuiu para que o modelo PIPAD-Bk não conseguisse um bom desempenho com as instâncias reais é o número de variáveis e restrições do problema para estas instâncias. Pode-se citar como exemplo a instância IAP024 (100 pedidos) que possui 316.180 variáveis, sendo que 155.470 variáveis são inteiras, e 168351 restrições, enquanto que a instância IAP025 (212 pedidos) possui 7.919.794 variáveis, sendo 3.953.140 variáveis são inteiras, e 3.982.624 restrições. O número de variáveis inteiras da instância IAP025

é 25 vezes maior que o número de variáveis inteira da instância IAP024.

Analisando-se a tabela 6.3, nota-se que 17 das 56 instâncias (30,3%) foram resolvidas otimamente, gastando-se um tempo computacional médio para obtenção da solução ótima de 650,1 segundos com um desvio-padrão de 1230,5 segundos. Para essas instâncias resolvidas otimamente, o *gap* médio de relaxação linear foi de 59,3% com um desvio-padrão de 12,8%. O tempo computacional médio para gerar a solução de relaxação linear das instâncias artificiais foi de 18,3 segundos com desvio-padrão de 29,6 segundos.

Das 24 instâncias artificiais, somente a instância IAP021 apresentou um *gap* do CPLEX superior a 90%, conseqüentemente apresentando um custo relativamente alto.

Tabela 6.3: Resultados para as instâncias Artificiais com o modelo PIPAD-Bk

Inst.	T(s)	G(%)	CT	CTR	R(%)	TR(s)
IAP001	3		88.987,7	19.217,5	78,4	0
IAP002	35		110.329,2	30.738,9	72,1	1
IAP003	11		59.962,0	35.971,1	40,0	0
IAP004	87		87.670,8	46.382,3	47,1	1
IAP005	1212		73.394,5	30.195,5	58,9	2
IAP006	6		171.983,3	53.140,2	69,1	1
IAP007	210		360.938,7	45.503,2	87,4	1
IAP008	184		133.737,4	67.291,4	49,7	2
IAP009	15		78.754,6	42.611,2	45,9	2
IAP010	5		98.350,5	46.914,5	52,3	1
IAP011	23		118.896,0	54.313,6	54,3	5
IAP012	906		94.701,9	58.037,6	38,7	2
IAP013	92		84.431,1	35.702,8	57,7	3
IAP014	177		121.222,8	54.875,4	54,7	2
IAP015	3276		150.253,9	44.026,1	70,7	4
IAP016	379		134.084,3	54.499,5	59,4	3
IAP017	4431		122.240,3	35.058,4	71,3	22
IAP018	7213	9,1	243.705,6	48.860,8		72
IAP019	7211	35,8	213.838,0	60.616,8		33
IAP020	7240	19,1	155.720,2	69.102,4		48
IAP021	7220	98,7	15.708.876,8	103.626,9		125
IAP022	7233	18,8	249.471,4	92.544,6		50
IAP023	7219	24,5	261.728,0	104.743,9		28
IAP024	7212	33,6	294.059,7	114.061,8		31

6.3.2 Resultados - Heurísticas para o PIAPSP

As tabelas 6.4 e 6.5 apresentam os resultados da heurística HAPSD para as instâncias-teste artificiais e reais, respectivamente. Nessas tabelas, a coluna “Inst.” apresenta o nome da instância resolvida, a coluna “T(s)” apresenta o tempo com-

putacional gasto pela heurística, a coluna “CT” o custo total da solução obtida pela heurística. As colunas “D(%)” e “DR(%)” apresentam, respectivamente, o desvio em relação à melhor solução encontrada pelos modelos e o desvio em relação à solução de relaxação linear. Os valores da coluna “D(%)” são calculados pela equação 6.42, onde $ResHeu$ representa o resultado obtido pela heurística e MR representa o melhor resultado encontrado pelos modelos PIPED ou PIPAD-Bk.

$$D = \frac{ResHeu - MR}{MR} \times 100 \quad (6.42)$$

Já os valores da coluna “DR(%)” são determinados pela equação 6.43, onde SR representa a solução encontrada pela relaxação linear do modelo PIPED.

$$DR = \frac{ResHeu - SR}{SR} \times 100 \quad (6.43)$$

Analisando-se as tabelas 6.4 e 6.5, verifica-se que em duas instâncias encontrou-se a solução ótima do problema. Considerando-se apenas as 17 instâncias (IAP001 a IAP017) que foram resolvidas otimamente pelos modelos PIPED e PIPAD-Bk, o desvio médio em relação à solução ótima foi de 0,8% com desvio-padrão de 0,7%. Em 36 das 56 instâncias (64,3%) encontrou-se uma solução melhor que a melhor solução encontrada pelos modelos.

O tempo computacional médio gasto pela heurística HAPSD foi de 681,5 segundos com um desvio-padrão de 1164,7 segundos, sendo que em apenas 23,2% delas (13 instâncias) o tempo computacional foi superior a 1000 segundos. Por sua vez, o desvio médio em relação à solução de relaxação linear foi de 56,2% com desvio padrão de 11,7%.

O modelo utilizado no desenvolvimento da heurística PIRF-FOpt foi o modelo PIPED, já que o modelo PIPAD-Bk não conseguiu resolver o problema relaxado dentro dos 7200 segundos definidos para a resolução do problema.

As tabelas 6.6 a 6.9 apresentam os resultados da heurística PIRF-FOpt, usando as estratégias de *relax-and-fix* PIRF-xt1f a PIRF-xt2b, para as instâncias artificiais do PIAPSP. Nestas tabelas têm-se dois grupos de resultados, um usando a estratégia PIFOpt-xt1f e outro usando a estratégia PIFOpt-xt1b. A primeira coluna destas tabelas apresenta a instância resolvida, a segunda coluna (NPd) o número de pedidos constantes na instância. Já a coluna “CT” apresenta o custo total da solução obtida pela heurística PIRF-FOpt, a coluna “TC(s)” apresenta o tempo computacional necessário

Tabela 6.4: Resultados para as instâncias Artificiais com a heurística HAPSD

Inst.	T(s)	CT	D(%)	DR(%)
IAP001	2,0	89.101,7	0,1	78,2
IAP002	3,0	110.443,2	0,1	71,1
IAP003	2,0	60.210,0	0,4	37,6
IAP004	6,0	87.670,8	0,0	45,3
IAP005	5,0	74.455,0	1,4	57,7
IAP006	4,0	173.196,2	0,7	59,8
IAP007	7,0	361.941,1	0,3	87,0
IAP008	10,0	136.836,9	2,3	46,8
IAP009	4,0	79.579,4	1,0	43,2
IAP010	6,0	99.112,9	0,8	50,2
IAP011	5,0	119.894,0	0,8	50,3
IAP012	9,0	96.935,7	2,4	39,2
IAP013	7,0	85.209,6	0,9	56,9
IAP014	10,0	122.107,0	0,7	54,6
IAP015	14,0	150.891,9	0,4	68,8
IAP016	10,0	134.084,3	0,0	57,0
IAP017	9,0	124.101,8	1,5	71,3
IAP018	36,0	250.875,2	2,9	74,7
IAP019	835,0	186.030,5	-13,0	66,4
IAP020	60,0	156.138,9	0,4	54,5
IAP021	3763,0	263.798,6	-96,5	57,6
IAP022	2340,0	246.181,9	-1,3	52,5
IAP023	411,0	255.229,4	-2,5	55,8
IAP024	4430,0	296.849,6	0,9	59,9

para obtenção desta solução e a coluna “D(%)” apresenta o desvio em relação à melhor solução obtida pelos modelos ou pela heurística HAPSD. Os valores da coluna “D(%)” são calculados conforme a equação (6.42).

Observando-se a tabela 6.6, nota-se que as duas estratégias *fix-and-optimize* tiveram desempenho semelhante quando utilizou-se a estratégia *ralax-and-fix* PIRF-xt1f. O tempo computacional médio para a estratégia PIFOpt-xt1f foi de 1095,3 segundos com desvio-padrão de 2154,9 segundos. Já o tempo computacional médio da heurística usando a estratégia PIFOpt-xt1b foi de 1064,5 segundos com desvio-padrão de 1867,5 segundos. Usando-se a estratégia PIFOpt-xt1f, 10 das 24 instâncias (41,7%) apresentaram um desvio em relação à melhor solução inferior a 10%, sendo que em 5 destas instâncias o desvio foi inferior a 1% e encontrou-se a solução ótima de uma instância (IAP001). E, utilizando-se a estratégia PIFOpt-xt1b, 9 das 24 instâncias (37,5%) apresentaram um desvio em relação à melhor solução inferior a 10%, sendo que em 4 destas instâncias o desvio foi inferior a 1%. Por sua vez, usando-se a estratégia PIFOpt-xt1f, 6 das 24 instâncias (25,0%) apresentaram desvio em relação à melhor solução superior a 100% e utilizando-se a estratégia PIFOpt-xt1b, 5 das 24 instâncias (20,8%) apresentaram desvio em relação à melhor solução superior a 100%.

Tabela 6.5: Resultados para as instâncias Reais com a heurística HAPSD

Inst.	T(s)	CT	D(%)	DR(%)
IAP025	1083,0	182.048,2	-82,4	61,2
IAP026	1843,0	500.649,4	-34,8	88,8
IAP027	493,0	123.096,9	-98,4	56,2
IAP028	56,0	120.672,0	-29,9	54,6
IAP029	23,0	120.311,0	-98,9	53,6
IAP030	56,0	109.774,9	-98,7	43,5
IAP031	2573,0	276.498,5	-94,2	67,4
IAP032	2585,0	247.754,4	-99,4	68,6
IAP033	1192,0	197.996,9	-99,2	58,5
IAP034	46,0	164.764,8	-92,2	52,6
IAP035	95,0	130.479,0	-99,6	36,0
IAP036	3680,0	697.055,0	-99,4	81,8
IAP037	3691,0	241.048,8	-99,8	55,1
IAP038	196,0	230.085,9	-99,8	48,8
IAP039	1007,0	188.076,3	-99,8	42,6
IAP040	83,0	112.118,9	-98,1	46,4
IAP041	45,0	88.076,2	-33,6	61,7
IAP042	65,0	121.683,1	-89,0	49,2
IAP043	21,0	92.981,2	-11,7	50,4
IAP044	81,0	115.804,4	-98,1	49,4
IAP045	21,0	117.641,9	-87,5	53,5
IAP046	38,0	119.909,3	-89,9	54,8
IAP047	97,0	88.767,0	-95,8	39,2
IAP048	1954,0	154.149,4	-98,4	49,4
IAP049	204,0	151.388,9	-97,8	59,3
IAP050	99,0	140.755,2	-98,0	45,0
IAP051	274,0	146.716,7	-97,0	47,0
IAP052	157,0	169.397,8	-98,1	55,8
IAP053	3076,0	185.663,1	-99,4	49,5
IAP054	398,0	214.751,3	-99,4	49,5
IAP055	505,0	233.446,3	-99,3	53,1
IAP056	441,0	208.676,3	-99,4	70,9

Observando-se agora a tabela 6.7, nota-se que a estratégia *fix-and-optimize* PIFOpt-xt1f teve uma pequena vantagem em relação a estratégia PIFOpt-xt1b. Além disso, nota-se um desempenho melhor da heurística usando a estratégia *relax-and-fix* PIRF-xt2f em relação a estratégia PIRF-xt1f. Usando-se a estratégia PIFOpt-xt1f, 16 das 24 instâncias (66,7%) apresentaram um desvio em relação à melhor solução inferior à 10%, sendo que em 13 destas instâncias o desvio foi inferior a 1% e em 5 instâncias a solução ótima foi encontrada. Já utilizando-se a estratégia PIFOpt-xt1b, 13 das 24 instâncias (54,2%) apresentaram um desvio em relação à melhor solução inferior à 10%, sendo que 10 destas instâncias o desvio foi inferior a 1% e em 4 instâncias a solução ótima foi encontrada. Por sua vez, usando-se a estratégia PIFOpt-xt1f, 3 das 24 instâncias (12,5%) apresentaram desvio em relação à melhor solução superior a 100% e utilizando-se a estratégia PIFOpt-xt1b, 4 das 24 instâncias (16,7%) apresentaram

Tabela 6.6: Resultados para as instâncias Artificiais com a heurística PIRF-FOpt - estratégia PIRF-xt1f

Inst.	NPd	PIFOpt-xt1f			PIFOpt-xt1b		
		CT	TC(s)	D(%)	CT	TC(s)	D(%)
IAP001	20	88.987,7	2,0	0,0	98.083,8	0,0	10,2
IAP002	25	1.043.806,1	8,0	846,1	1.047.206,1	1,0	849,2
IAP003	30	60.210,0	2,0	0,4	60.210,0	1,0	0,4
IAP004	30	93.095,8	4,0	6,2	92.550,5	4,0	5,6
IAP005	40	74.353,6	27,0	1,3	74.353,6	30,0	1,3
IAP006	40	173.000,2	1,0	0,6	173.000,2	1,0	0,6
IAP007	40	361.220,1	3,0	0,1	361.220,1	3,0	0,1
IAP008	40	151.002,6	6,0	12,9	213.276,4	4,0	59,5
IAP009	45	90.686,6	9,0	15,2	90.078,7	8,0	14,4
IAP010	45	108.418,6	8,0	10,2	108.418,6	7,0	10,2
IAP011	45	134.492,0	3,0	13,1	134.492,0	2,0	13,1
IAP012	45	95.762,9	7,0	1,1	95.762,9	6,0	1,1
IAP013	50	84.719,3	7,0	0,3	84.719,3	5,0	0,3
IAP014	50	133.616,2	7,0	10,2	145.893,3	4,0	20,4
IAP015	50	175.366,9	8,0	16,7	196.816,9	4,0	31,0
IAP016	50	187.340,9	12,0	39,7	1.084.374,4	7,0	708,7
IAP017	100	133.992,5	2430,0	9,6	133.995,5	4014,0	9,6
IAP018	100	1.029.149,7	3149,0	322,3	286.632,3	3291,0	17,6
IAP019	100	224.039,7	1494,0	20,4	224.039,7	1686,0	20,4
IAP020	100	1.655.261,7	9488,0	964,6	1.655.360,1	5897,0	964,7
IAP021	100	6.594.253,6	4341,0	>1000	6.594.253,6	4869,0	>1000
IAP022	100	254.929,5	225,0	3,6	254.929,5	308,0	3,6
IAP023	100	1.052.174,4	2005,0	312,2	312.925,5	702,0	22,6
IAP024	100	4.114.823,1	3040,0	>1000	3.252.759,9	4695,0	>1000

desvio em relação à melhor solução superior a 100%. O tempo computacional médio para a estratégia PIFOpt-xt1f foi de 1782,9 segundos com desvio-padrão de 2873,6 segundos. E o tempo computacional médio da heurística usando a estratégia PIFOpt-xt1b foi de 1251,3 segundos com desvio-padrão de 2042,7 segundos.

Por sua vez, observando-se a tabela 6.8, nota-se que ambas estratégias *fix-and-optimize*, PIFOpt-xt1f e PIFOpt-xt1b, tiveram um desempenho semelhante. Nota-se, também, um desempenho melhor da heurística usando a estratégia *relax-and-fix* PIRF-xt2f em relação a estratégia PIRF-xt1b. Usando-se a estratégia PIFOpt-xt1f, 13 das 24 instâncias (54,2%) apresentaram um desvio em relação à melhor solução inferior à 10%, sendo que em 4 destas instâncias o desvio foi inferior a 1% e em 1 instância (IAP001) a solução ótima foi encontrada. Já utilizando-se a estratégia PIFOpt-xt1b, 12 das 24 instâncias (50,0%) apresentaram um desvio em relação à melhor solução inferior à 10%, sendo que 4 destas instâncias o desvio foi inferior a 1% e em 1 instância (IAP001) a solução ótima foi encontrada. Por sua vez, em ambas estratégias, 3 das 24 instâncias (12,5%) apresentaram desvio em relação à melhor solução superior a 100%. O tempo computacional médio para a estratégia PIFOpt-xt1f foi de 782,3 segundos

Tabela 6.7: Resultados para as instâncias Artificiais com a heurística PIRF-FOpt - estratégia PIRF-xt2f

Inst.	NPd	PIFOpt-xt1f			PIFOpt-xt1b		
		CT	TC(s)	D(%)	CT	TC(s)	D(%)
IAP001	20	88.987,7	6,0	0,0	88.987,7	3,0	0,0
IAP002	25	110.329,2	12,0	0,0	110.329,2	5,0	0,0
IAP003	30	60.210,0	14,0	0,4	60.210,0	3,0	0,4
IAP004	30	91.899,9	34,0	4,8	91.899,9	13,0	4,8
IAP005	40	73.711,0	552,0	0,4	73.711,0	85,0	0,4
IAP006	40	173.000,2	13,0	0,6	173.000,2	6,0	0,6
IAP007	40	361.220,1	36,0	0,1	361.220,1	13,0	0,1
IAP008	40	151.002,6	29,0	12,9	158.017,4	11,0	18,2
IAP009	45	79.096,6	145,0	0,4	90.078,7	55,0	14,4
IAP010	45	98.350,5	95,0	0,0	98.350,5	16,0	0,0
IAP011	45	119.894,0	21,0	0,8	119.894,0	10,0	0,8
IAP012	45	95.762,9	87,0	1,1	95.762,9	24,0	1,1
IAP013	50	84.719,3	168,0	0,3	84.719,3	42,0	0,3
IAP014	50	121.432,0	54,0	0,2	131.779,5	13,0	8,7
IAP015	50	161.108,9	95,0	7,2	173.483,9	17,0	15,5
IAP016	50	134.084,3	41,0	0,0	134.084,3	18,0	0,0
IAP017	100	122.240,3	6908,0	0,0	136.994,2	2546,0	12,1
IAP018	100	277.758,5	4396,0	14,0	1.019.295,9	3698,0	318,2
IAP019	100	206.740,5	4106,0	11,1	206.740,5	2819,0	11,1
IAP020	100	181.035,8	2194,0	16,4	181.035,8	1078,0	16,4
IAP021	100	8.578.955,2	11632,0	>1000	6.567.129,6	7927,0	>1000
IAP022	100	406.151,6	4029,0	65,0	406.151,6	3638,0	65,0
IAP023	100	1.961.268,2	3414,0	668,4	1.961.268,2	3525,0	668,4
IAP024	100	2.165.407,0	4708,0	636,4	2.165.407,0	4465,0	636,4

com desvio-padrão de 1446,0 segundos. E o tempo computacional médio da heurística usando a estratégia PIFOpt-xt1b foi de 594,2 segundos com desvio-padrão de 991,2 segundos.

Por fim, observando-se a tabela 6.9, nota-se que ambas estratégias *fix-and-optimize*, PIFOpt-xt1f e PIFOpt-xt1b, tiveram um desempenho semelhante, com uma pequena vantagem para a estratégia PIFOpt-xt1b. Nota-se, também, um desempenho melhor da heurística usando a estratégia *relax-and-fix* PIRF-xt2b em relação às outras três estratégias. Usando-se a estratégia PIFOpt-xt1f, 15 das 24 instâncias (62,5%) apresentaram um desvio em relação à melhor solução inferior à 10%, sendo que em 10 destas instâncias o desvio foi inferior a 1% e em 4 instâncias a solução ótima foi encontrada. Já utilizando-se a estratégia PIFOpt-xt1b, 19 das 24 instâncias (79,2%) apresentaram um desvio em relação à melhor solução inferior à 10%, sendo que 14 destas instâncias o desvio foi inferior a 1% e em 6 instâncias a melhor solução foi encontrada. Usando-se a estratégia PIFOpt-xt1f, 3 das 24 instâncias (12,5%) apresentaram desvio em relação à melhor solução superior a 100% e utilizando-se a estratégia PIFOpt-xt1b, 2 das 24 instâncias (8,3%) apresentaram desvio em relação à melhor solução superior

Tabela 6.8: Resultados para as instâncias Artificiais com a heurística PIRF-FOpt - estratégia PIRF-xt1b

Inst.	NPd	PIFOpt-xt1f			PIFOpt-xt1b		
		CT	TC(s)	D(%)	CT	TC(s)	D(%)
IAP001	20	88.987,7	2,0	0,0	88.987,7	2,0	0,0
IAP002	25	121.217,2	2,0	9,9	121.217,2	1,0	9,9
IAP003	30	60.210,0	1,0	0,4	60.210,0	2,0	0,4
IAP004	30	93.445,8	2,0	6,6	93.445,8	3,0	6,6
IAP005	40	74.353,6	6,0	1,3	74.353,6	50,0	1,3
IAP006	40	212.250,2	3,0	23,4	212.250,2	2,0	23,4
IAP007	40	371.810,1	4,0	3,0	371.810,1	5,0	3,0
IAP008	40	162.279,1	3,0	21,3	166.352,6	4,0	24,4
IAP009	45	79.096,6	3,0	0,4	79.096,6	5,0	0,4
IAP010	45	115.232,6	12,0	17,2	108.418,6	9,0	10,2
IAP011	45	178.459,0	6,0	50,1	162.102,0	5,0	36,3
IAP012	45	95.762,9	11,0	1,1	95.762,9	11,0	1,1
IAP013	50	91.927,8	15,0	8,9	91.927,8	23,0	8,9
IAP014	50	143.264,6	3,0	18,2	121.432,0	5,0	0,2
IAP015	50	165.149,9	11,0	9,9	175.675,9	8,0	16,9
IAP016	50	191.206,4	12,0	42,6	183.949,3	15,0	37,2
IAP017	100	133.992,5	1537,0	9,6	133.992,5	1437,0	9,6
IAP018	100	1.016.465,7	2489,0	317,1	255.305,6	1554,0	4,8
IAP019	100	231.767,8	1923,0	24,6	254.913,4	1352,0	37,0
IAP020	100	163.424,1	686,0	5,1	1.669.806,4	3426,0	974,0
IAP021	100	8.687.402,3	5949,0	>1000	6.649.987,6	3022,0	>1000
IAP022	100	248.590,9	1187,0	1,0	272.316,7	216,0	10,6
IAP023	100	469.177,7	3738,0	83,8	405.620,3	1619,0	58,9
IAP024	100	1.361.740,0	1170,0	363,1	2.249.109,7	1485,0	664,8

a 100%. O tempo computacional médio para a estratégia PIFOpt-xt1f foi de 1188,2 segundos com desvio-padrão de 1991,0 segundos. E o tempo computacional médio da heurística usando a estratégia PIFOpt-xt1b foi de 1234,5 segundos com desvio-padrão de 2325,1 segundos.

As instâncias IAP021 e IAP024 foram as duas instâncias em que todas as estratégias apresentaram desvio em relação à melhor solução muito alto (superior a 100%). Além disso, pode-se constatar, pela análise das tabelas 6.6 a 6.9, que a combinação das estratégias PIRF-xt2b e PIFOpt-xt1b foi a que apresentou melhores resultados em comparação às demais combinações.

As tabelas 6.10 a 6.13 apresentam os resultados da heurística PIRF-FOpt, usando as estratégias de *relax-and-fix* PIRF-xt1f a PIRF-xt2b, para as 32 instâncias reais do PIAPSP. A estrutura destas tabelas é idêntica à estrutura das tabelas 6.6 a 6.9.

Observando-se a tabela 6.10, pode-se notar que em 46,9% das instâncias (15 de 32 instâncias), em ambas situações (estratégia PIFOpt-xt1f e PIFOpt-xt1b), encontrou-

Tabela 6.9: Resultados para as instâncias Artificiais com a heurística PIRF-FOpt - estratégia PIRF-xt2b

Inst.	NPd	PIFOpt-xt1f			PIFOpt-xt1b		
		CT	TC(s)	D(%)	CT	TC(s)	D(%)
IAP001	20	88.987,7	4,0	0,0	88.987,7	4,0	0,0
IAP002	25	122.104,2	6,0	10,7	110.329,2	6,0	0,0
IAP003	30	60.210,0	9,0	0,4	60.210,0	8,0	0,4
IAP004	30	87.670,8	12,0	0,0	87.670,8	14,0	0,0
IAP005	40	73.711,0	111,0	0,4	73.711,0	223,0	0,4
IAP006	40	190.900,2	8,0	11,0	173.000,2	9,0	0,6
IAP007	40	361.220,1	12,0	0,1	361.220,1	14,0	0,1
IAP008	40	140.925,6	13,0	5,4	140.925,6	12,0	5,4
IAP009	45	79.096,6	21,0	0,4	79.096,6	29,0	0,4
IAP010	45	98.350,5	20,0	0,0	98.350,5	48,0	0,0
IAP011	45	142.251,6	18,0	19,6	134.492,0	22,0	13,1
IAP012	45	95.762,9	34,0	1,1	95.762,9	33,0	1,1
IAP013	50	84.719,3	68,0	0,3	84.719,3	79,0	0,3
IAP014	50	137.664,6	11,0	13,6	121.432,0	18,0	0,2
IAP015	50	167.624,9	18,0	11,6	165.149,9	28,0	9,9
IAP016	50	145.684,3	38,0	8,7	145.684,3	39,0	8,7
IAP017	100	122.240,3	2207,0	0,0	122.240,3	1748,0	0,0
IAP018	100	248.451,9	2546,0	1,9	243.705,6	2200,0	0,0
IAP019	100	257.132,0	4711,0	38,2	257.132,0	3626,0	38,2
IAP020	100	163.374,1	1933,0	5,1	163.374,1	2452,0	5,1
IAP021	100	8.683.449,2	8089,0	>1000	7.529.428,1	10591,0	>1000
IAP022	100	248.590,9	1893,0	1,0	248.590,9	2088,0	1,0
IAP023	100	1.026.997,8	3464,0	302,4	435.937,5	2030,0	70,8
IAP024	100	1.325.498,6	3271,0	350,8	1.251.938,0	4308,0	325,7

se um desvio em relação à melhor solução inferior a 10%, sendo que em 10 instâncias (31,3%) este desvio foi inferior a 1%. Utilizando-se a estratégia PIFOpt-xt1f, em 2 instâncias a melhor solução foi encontrada e em outras 6 instâncias (18,8%) a melhor solução foi superada. Utilizando-se a estratégia PIFOpt-xt1b, em 3 instâncias a melhor solução foi encontrada e em outras 6 instâncias (18,8%) a melhor solução foi superada. O tempo computacional médio foi de 11819,5 segundos com desvio padrão de 3854,2 segundos, usando a estratégia PIFOpt-xt1f. Já o tempo computacional médio, usando-se a estratégia PIFOpt-xt1b, foi de 11524,0 segundos com desvio-padrão de 3865,0 segundos. Em ambas estratégias, 31,3% (10 de 32 instâncias) das instâncias apresentaram um desvio em relação à melhor solução superior a 100%.

Analisando-se a tabela 6.11, nota-se que em 34,4% das instâncias (11 de 32 instâncias) resolvidas usando-se a estratégia PIFOpt-xt1f, encontrou-se um desvio em relação à melhor solução inferior a 10%, sendo que em 6 instâncias (18,8%) este desvio foi inferior a 1%. E em 28,1% das instâncias (9 de 32 instâncias) resolvidas usando-se a estratégia PIFOpt-xt1b, encontrou-se um desvio em relação à melhor solução inferior a 10%, sendo que em 5 instâncias (15,6%) este desvio foi inferior a 1%. Utilizando-se

Tabela 6.10: Resultados para as instâncias Reais com a heurística PIRF-FOpt - estratégia PIRF-xt1f

Inst.	NPd	PIFOpt-xt1f			PIFOpt-xt1b		
		CT	TC(s)	D(%)	CT	TC(s)	D(%)
IAP025	212	182.150,8	9552,0	0,1	182.180,8	9574,0	0,1
IAP026	220	522.188,4	7736,0	4,3	526.355,6	9560,0	5,1
IAP027	219	121.080,8	10844,0	-1,6	120.809,1	8132,0	-1,9
IAP028	228	120.672,0	5844,0	0,0	120.672,0	6213,0	0,0
IAP029	226	120.317,0	3895,0	0,0	120.311,0	6111,0	0,0
IAP030	260	124.916,9	7932,0	13,8	124.916,9	7107,0	13,8
IAP031	290	281.993,0	10926,0	2,0	284.098,1	10899,0	2,7
IAP032	311	279.945,9	11290,0	13,0	277.035,0	11347,0	11,8
IAP033	336	196.867,7	11540,0	-0,6	196.976,7	11445,0	-0,5
IAP034	320	164.735,8	10768,0	0,0	164.735,8	8633,0	0,0
IAP035	360	3.595.651,0	12307,0	>1000	3.812.205,8	11897,0	>1000
IAP036	432	9.670.559,8	17969,0	>1000	23.162.613,6	17927,0	>1000
IAP037	445	24.182.614,1	18530,0	>1000	26.025.863,2	18437,0	>1000
IAP038	486	29.983.453,8	17689,0	>1000	29.036.017,5	17648,0	>1000
IAP039	460	57.157.466,5	14328,0	>1000	69.834.166,8	14346,0	>1000
IAP040	258	127.061,4	10869,0	13,3	125.704,4	10859,0	12,1
IAP041	220	84.007,9	8187,0	-4,6	85.800,0	10617,0	-2,6
IAP042	256	159.598,1	10369,0	31,2	158.633,1	6515,0	30,4
IAP043	194	92.981,2	4318,0	0,0	92.981,2	3835,0	0,0
IAP044	264	110.294,4	10865,0	-4,8	111.193,4	10856,0	-4,0
IAP045	217	125.861,9	11763,0	7,0	125.796,9	8954,0	6,9
IAP046	240	122.637,3	10410,0	2,3	122.592,3	10790,0	2,2
IAP047	256	91.085,2	10997,0	2,6	91.895,3	10985,0	3,5
IAP048	356	188.970,0	11562,0	22,6	188.641,1	11538,0	22,4
IAP049	314	146.533,4	13047,0	-3,2	146.533,4	11196,0	-3,2
IAP050	306	155.664,2	11485,0	10,6	155.408,2	11247,0	10,4
IAP051	343	398.579,0	12480,0	171,7	424.902,5	11607,0	189,6
IAP052	332	203.872,7	11549,0	20,4	205.085,4	11534,0	21,1
IAP053	478	2.157.126,5	19047,0	>1000	1.876.056,3	18959,0	910,5
IAP054	452	6.152.405,3	17784,0	>1000	2.038.587,0	17729,0	849,3
IAP055	483	4.029.939,5	18287,0	>1000	7.083.614,1	18232,0	>1000
IAP056	496	8.313.263,7	14054,0	>1000	8.313.360,5	14040,0	>1000

a estratégia PIFOpt-xt1f, em 1 instância a melhor solução foi encontrada e em outras 3 instâncias (9,4%) a melhor solução foi superada. Utilizando-se a estratégia PIFOpt-xt1b, também em 1 instância a melhor solução foi encontrada e em outras 2 instâncias (6,3%) a melhor solução foi superada. Utilizando-se a estratégia PIFOpt-xt1f, 40,6% (13 de 32 instâncias) das instâncias apresentaram um desvio em relação à melhor solução superior a 100%. Já utilizando-se a estratégia PIFOpt-xt1b, 37,5% (12 de 32 instâncias) das instâncias apresentaram um desvio em relação à melhor solução superior a 100%. O tempo computacional médio foi de 12447,1 segundos com desvio padrão de 3349,5 segundos, usando a estratégia PIFOpt-xt1f. Já o tempo computacional médio, usando-se a estratégia PIFOpt-xt1b, foi de 12432,2 segundos com desvio-padrão de 3192,0 segundos.

Tabela 6.11: Resultados para as instâncias Reais com a heurística PIRF-FOpt - estratégia PIRF-xt2f

Inst.	NPd	PIFOpt-xt1f			PIFOpt-xt1b		
		CT	TC(s)	D(%)	CT	TC(s)	D(%)
IAP025	212	284.102,7	10423,0	56,1	192.674,0	10326,0	5,8
IAP026	220	516.088,7	10378,0	3,1	534.947,9	9785,0	6,9
IAP027	219	120.222,0	10409,0	-2,3	121.687,7	10332,0	-1,1
IAP028	228	120.672,0	10809,0	0,0	120.682,0	8857,0	0,0
IAP029	226	120.459,0	8552,0	0,1	120.311,0	11288,0	0,0
IAP030	260	119.146,9	10357,0	8,5	142.491,9	10378,0	29,8
IAP031	290	345.470,9	11192,0	24,9	344.534,5	11015,0	24,6
IAP032	311	569.303,1	11027,0	129,8	480.835,1	11072,0	94,1
IAP033	336	761.117,8	11114,0	284,4	854.555,7	14350,0	331,6
IAP034	320	164.876,8	13018,0	0,1	164.783,8	13322,0	0,0
IAP035	360	883.356,2	11899,0	577,0	2.681.912,4	11374,0	>1000
IAP036	432	2.164.661,6	17782,0	210,5	16.234.305,5	17663,0	>1000
IAP037	445	4.445.694,8	19953,0	>1000	7.695.142,6	18563,0	>1000
IAP038	486	1.559.934,1	17785,0	578,0	9.338.068,7	15710,0	>1000
IAP039	460	5.783.859,3	14129,0	>1000	43.534.132,2	14101,0	>1000
IAP040	258	145.487,1	10376,0	29,8	160.393,2	10426,0	43,1
IAP041	220	106.245,8	8916,0	20,6	105.534,2	6911,0	19,8
IAP042	256	171.102,8	10179,0	40,6	162.870,1	9677,0	33,8
IAP043	194	88.894,2	8283,0	-4,4	88.894,2	8489,0	-4,4
IAP044	264	315.121,1	9404,0	172,1	420.729,3	10348,0	263,3
IAP045	217	128.710,1	9746,0	9,4	128.596,9	10279,0	9,3
IAP046	240	124.455,3	10304,0	3,8	171.392,1	10378,0	42,9
IAP047	256	97.275,6	10597,0	9,6	129.853,2	10724,0	46,3
IAP048	356	251.501,2	12997,0	63,2	273.732,4	11106,0	77,6
IAP049	314	148.793,4	13495,0	-1,7	157.206,3	11080,0	3,8
IAP050	306	189.196,2	11036,0	34,4	255.863,2	13418,0	81,8
IAP051	343	608.059,0	11172,0	314,4	855.108,9	14329,0	482,8
IAP052	332	211.542,5	11232,0	24,9	204.184,6	11131,0	20,5
IAP053	478	2.140.597,5	19029,0	>1000	2.652.122,6	17152,0	>1000
IAP054	452	1.582.955,0	15110,0	637,1	18.286.329,3	17650,0	>1000
IAP055	483	529.292,7	18226,0	126,7	3.116.150,4	17294,0	>1000
IAP056	496	850.774,0	19377,0	307,7	2.139.288,9	19303,0	925,2

Observando-se a tabela 6.12, nota-se que em 31,3% das instâncias (10 de 32 instâncias) resolvidas usando-se a estratégia PIFOpt-xt1f, encontrou-se um desvio em relação à melhor solução inferior a 10%, sendo que em 6 instâncias (18,8%) este desvio foi inferior a 1%. E em 25,0% das instâncias (8 de 32 instâncias) resolvidas usando-se a estratégia PIFOpt-xt1b, encontrou-se um desvio em relação à melhor solução inferior a 10%, sendo que em 5 instâncias (15,6%) este desvio foi inferior a 1%. Utilizando-se a estratégia PIFOpt-xt1f, em 2 instâncias a melhor solução foi encontrada e em outras 3 instâncias (9,4%) a melhor solução foi superada. Utilizando-se a estratégia PIFOpt-xt1b, também em 2 instâncias a melhor solução foi encontrada e em 1 instância (3,1%) a melhor solução foi superada. Em ambas estratégias, 28,1% das instâncias (9 de 32 instâncias) apresentaram um desvio em relação à melhor solução superior a 100%. Além

disso, usando-se a estratégia PIFOpt-xt1f, 1 instância (IAP035) não foi resolvida, pois o processo foi interrompido por falta de memória e usando-se a estratégia PIFOpt-xt1b, 2 instâncias (IAP035 e IAP036) não foram resolvidas devido ao mesmo problema de falta de memória. O tempo computacional médio das instâncias que foram resolvidas usando a estratégia PIFOpt-xt1f, foi de 11882,6 segundos com desvio padrão de 3598,1 segundos. Já o tempo computacional médio para as instâncias que foram resolvidas usando-se a estratégia PIFOpt-xt1b, foi de 11840,9 segundos com desvio-padrão de 3102,1 segundos.

Tabela 6.12: Resultados para as instâncias Reais com a heurística PIRF-FOpt - estratégia PIRF-xt1b

Inst.	NPd	PIFOpt-xt1f			PIFOpt-xt1b		
		CT	TC(s)	D(%)	CT	TC(s)	D(%)
IAP025	212	182.159,8	7365,0	0,1	182.154,8	9569,0	0,1
IAP026	220	529.900,9	10178,0	5,8	536.720,5	9900,0	7,2
IAP027	219	120.200,0	10841,0	-2,4	120.200,0	10641,0	-2,4
IAP028	228	120.672,0	4014,0	0,0	120.672,0	4345,0	0,0
IAP029	226	120.311,0	7395,0	0,0	120.311,0	8837,0	0,0
IAP030	260	133.418,0	10871,0	21,5	153.224,9	11544,0	39,6
IAP031	290	299.210,2	11270,0	8,2	299.349,2	11200,0	8,3
IAP032	311	290.680,1	11463,0	17,3	277.974,5	11482,0	12,2
IAP033	336	197.398,9	11518,0	-0,3	198.859,9	11571,0	0,4
IAP034	320	164.735,8	9032,0	0,0	387.949,8	11639,0	135,5
IAP035	360	-	-	-	-	-	-
IAP036	432	10.871.653,0	19563,0	>1000	-	-	-
IAP037	445	6.844.984,2	17890,0	>1000	7.431.494,7	18460,0	>1000
IAP038	486	30.226.656,2	15963,0	>1000	13.518.368,5	14533,0	>1000
IAP039	460	40.032.860,2	15685,0	>1000	40.616.851,9	13943,0	>1000
IAP040	258	150.998,5	10854,0	34,7	169.584,0	10859,0	51,3
IAP041	220	104.468,1	9851,0	18,6	109.075,7	11537,0	23,8
IAP042	256	208.963,3	9992,0	71,7	213.745,3	10098,0	75,7
IAP043	194	96.945,1	8179,0	4,3	96.986,1	6978,0	4,3
IAP044	264	125.157,6	10859,0	8,1	140.293,4	10860,0	21,1
IAP045	217	162.573,9	7097,0	38,2	167.166,4	10626,0	42,1
IAP046	240	145.196,9	10426,0	21,1	161.236,6	10409,0	34,5
IAP047	256	126.849,5	10984,0	42,9	143.466,0	10982,0	61,6
IAP048	356	220.373,6	10965,0	43,0	499.142,8	11401,0	223,8
IAP049	314	199.023,4	13428,0	31,5	204.328,1	11203,0	35,0
IAP050	306	197.853,1	12698,0	40,6	244.654,9	11402,0	73,8
IAP051	343	310.174,6	11604,0	111,4	258.820,1	11616,0	76,4
IAP052	332	230.183,9	11532,0	35,9	275.171,4	11540,0	62,4
IAP053	478	3.633.424,3	17339,0	>1000	6.779.854,1	18716,0	>1000
IAP054	452	582.395,3	17722,0	171,2	1.381.940,0	17713,0	543,5
IAP055	483	3.301.551,1	18059,0	>1000	2.195.342,6	18016,0	840,4
IAP056	496	634.369,3	13723,0	204,0	4.055.047,2	13606,0	>1000

Analisando-se a tabela 6.13, nota-se que em 31,3% das instâncias (10 de 32 instâncias) resolvidas usando-se a estratégia PIFOpt-xt1f, encontrou-se um desvio em relação à melhor solução inferior a 10%, sendo que em 5 instâncias (15,6%) este desvio

foi inferior a 1%. E em 28,1% das instâncias (9 de 32 instâncias) resolvidas usando-se a estratégia PIFOpt-xt1b, encontrou-se um desvio em relação à melhor solução inferior a 10%, sendo que em 5 instâncias (15,6%) este desvio foi inferior a 1%. Utilizando-se a estratégia PIFOpt-xt1f, em 1 instância a melhor solução foi encontrada e em outra instância (IAP043) a melhor solução foi superada. Utilizando-se a estratégia PIFOpt-xt1b, em 2 instâncias a melhor solução foi encontrada e em outra instância (IAP043) a melhor solução foi superada. Utilizando-se a estratégia PIFOpt-xt1f, 34,4% das instâncias (11 de 32 instâncias) apresentaram um desvio em relação à melhor solução superior a 100%. Já utilizando-se a estratégia PIFOpt-xt1b, 50,0% das instâncias (16 de 32 instâncias) apresentaram um desvio em relação à melhor solução superior a 100%. O tempo computacional médio foi de 12166,6 segundos com desvio padrão de 3220,0 segundos, usando a estratégia PIFOpt-xt1f. Já o tempo computacional médio, usando-se a estratégia PIFOpt-xt1b, foi de 12378,6 segundos com desvio-padrão de 2956,6 segundos.

Constata-se que a combinação de estratégias que teve melhor desempenho foi a combinação da estratégia *relax-and-fix* PIRF-xt1f com a estratégia *fix-and-optimize* PIFOpt-xt1f. Esta combinação foi a que produziu maior quantidade de novas melhores soluções para as instâncias, sendo 5 novas melhores soluções no total.

Na resolução das instâncias IAP035 a IAP039 e IAP053 a IAP056 pela heurística HPIRF-FOpt, encontraram-se soluções com desvio médio superior a 100% em relação à melhor solução para todas as combinações de estratégias.

A tabela 6.14 apresenta as melhores soluções obtidas com os três métodos propostos: (i) os modelos de programação linear inteira mista (MPLIM) PIPED e PIPAD-Bk; (ii) a heurística agrupa primeiro sequencia depois (HAPSD) e (iii) a heurística que combina as técnicas de *relax-and-fix* e *fix-and-optimize* (HPIRF-FOpt). Esta tabela apresenta as melhores soluções tanto para as instâncias artificiais quanto para as instâncias reais. A coluna “Inst.” desta tabela apresenta a instância resolvida, a coluna “NPd” representa o número de pedidos contantes na instância, a coluna “NPa” apresenta o número de padrões de aço existente na instância e a coluna “NPe” apresenta o número de períodos. Já a coluna “MR” apresenta o melhor resultado para aquela instância e a coluna “Método” apresenta o método que obteve esta melhor solução.

Observando-se a tabela 6.14, nota-se que 83,3% das melhores soluções para as instâncias artificiais (IAP001 a IAP024) foram obtidas com os modelos de PLIM, sendo que 70,8% das instâncias foram resolvidas otimamente. As demais 16,7% das

Tabela 6.13: Resultados para as instâncias Reais com a heurística PIRF-FOpt - estratégia PIRF-xt2b

Inst.	NPd	PIFOpt-xt1f			PIFOpt-xt1b		
		CT	TC(s)	D(%)	CT	TC(s)	D(%)
IAP025	212	182.243,8	10125,0	0,1	199.168,4	10340,0	9,4
IAP026	220	506.607,9	10364,0	1,2	517.215,1	11476,0	3,3
IAP027	219	129.036,4	10336,0	4,8	131.610,8	10361,0	6,9
IAP028	228	120.672,0	11113,0	0,0	120.672,0	8665,0	0,0
IAP029	226	120.332,0	9890,0	0,0	120.311,0	9403,0	0,0
IAP030	260	133.428,0	10347,0	21,5	143.477,9	10355,0	30,7
IAP031	290	536.408,6	10987,0	94,0	1.105.718,7	11030,0	299,9
IAP032	311	273.448,7	11007,0	10,4	288.951,5	11012,0	16,6
IAP033	336	615.121,8	11050,0	210,7	1.614.606,4	10704,0	715,5
IAP034	320	164.864,8	13016,0	0,1	164.888,8	13611,0	0,1
IAP035	360	808.746,1	11863,0	519,8	871.597,3	11424,0	568,0
IAP036	432	1.640.892,7	17835,0	135,4	3.092.330,7	17664,0	343,6
IAP037	445	1.879.767,9	18338,0	679,8	3.605.994,4	15422,0	>1000
IAP038	486	24.409.715,2	17159,0	>1000	7.467.798,4	15856,0	>1000
IAP039	460	1.264.283,2	13607,0	572,2	5.794.849,0	18057,0	>1000
IAP040	258	159.041,9	10147,0	41,9	241.100,7	10437,0	115,0
IAP041	220	90.312,8	10464,0	2,5	101.717,8	10504,0	15,5
IAP042	256	185.866,2	10341,0	52,7	162.551,0	10144,0	33,6
IAP043	194	88.894,2	5682,0	-4,4	91.461,5	10311,0	-1,6
IAP044	264	131.275,1	10351,0	13,4	166.242,0	10412,0	43,6
IAP045	217	122.765,9	7588,0	4,4	118.414,9	9557,0	0,7
IAP046	240	142.981,4	10372,0	19,2	308.617,6	11352,0	157,4
IAP047	256	90.822,0	10585,0	2,3	108.869,7	10617,0	22,6
IAP048	356	186.863,8	11062,0	21,2	377.194,5	11114,0	144,7
IAP049	314	172.957,9	13578,0	14,2	157.867,8	11031,0	4,3
IAP050	306	217.257,2	10859,0	54,4	223.648,2	11029,0	58,9
IAP051	343	438.633,6	11224,0	199,0	833.673,3	14222,0	468,2
IAP052	332	332.287,3	11132,0	96,2	374.327,9	11148,0	121,0
IAP053	478	902.570,4	18848,0	386,1	1.173.910,4	19028,0	532,3
IAP054	452	3.664.423,5	18000,0	>1000	5.737.607,2	17891,0	>1000
IAP055	483	1.086.708,0	18444,0	365,5	4.974.400,0	18251,0	>1000
IAP056	496	890.378,8	13618,0	326,7	10.546.164,3	13688,0	>1000

melhores soluções foram obtidas pela heurística HAPSD. Nas instâncias reais(IAP025 a IAP056), 78,1% das melhores soluções foram obtidas com a heurística HAPSD. As demais melhores soluções (21,9%) foram obtidas com a heurística HPIRF-FOpt. No âmbito geral, considerando instâncias artificiais e reais, obteve-se um maior número de melhores soluções com a heurística HAPSD (51,8%). Já 35,7% das melhores soluções foram obtidas com os modelos de PLIM e 12,5% com a heurística HPIRF-FOpt.

Tabela 6.14: Melhores resultados para as instâncias testadas

Inst.	NPd	NPa	NPe	MR	Método
IAP001	20	7	2	88.987,7	MPLIM (*)
IAP002	25	14	3	110.329,2	MPLIM (*)
IAP003	30	15	3	59.962,0	MPLIM (*)
IAP004	30	13	4	87.670,8	MPLIM (*)
IAP005	40	17	3	73.394,5	MPLIM (*)
IAP006	40	28	4	171.983,3	MPLIM (*)
IAP007	40	21	5	360.938,7	MPLIM (*)
IAP008	40	27	6	133.737,4	MPLIM (*)
IAP009	45	23	3	78.754,6	MPLIM (*)
IAP010	45	20	4	98.350,5	MPLIM (*)
IAP011	45	28	5	118.896,0	MPLIM (*)
IAP012	45	21	6	94.701,9	MPLIM (*)
IAP013	50	21	3	84.431,1	MPLIM (*)
IAP014	50	25	4	121.222,8	MPLIM (*)
IAP015	50	27	5	150.253,9	MPLIM (*)
IAP016	50	25	6	134.084,3	MPLIM (*)
IAP017	100	26	3	122.240,3	MPLIM (*)
IAP018	100	37	4	243.705,6	MPLIM
IAP019	100	34	5	186.030,5	HAPSD
IAP020	100	32	6	155.479,1	MPLIM
IAP021	100	49	7	263.798,6	HAPSD
IAP022	100	46	8	246.181,9	HAPSD
IAP023	100	49	9	255.229,4	HAPSD
IAP024	100	47	10	294.059,7	MPLIM
IAP025	212	51	5	182.048,2	HAPSD
IAP026	220	59	5	500.649,4	HAPSD
IAP027	219	50	5	120.200,0	HPIRF-FOpt
IAP028	228	51	5	120.672,0	HAPSD
IAP029	226	75	5	120.311,0	HAPSD
IAP030	260	58	5	109.774,9	HAPSD
IAP031	290	70	7	276.498,5	HAPSD
IAP032	311	77	7	247.754,4	HAPSD
IAP033	336	84	7	196.867,7	HPIRF-FOpt
IAP034	320	90	7	164.735,8	HPIRF-FOpt
IAP035	360	74	7	130.479,0	HAPSD
IAP036	432	109	10	697.055,0	HAPSD
IAP037	445	98	10	241.048,8	HAPSD
IAP038	486	133	10	230.085,9	HAPSD
IAP039	460	95	10	188.076,3	HAPSD
IAP040	258	52	5	112.118,9	HAPSD
IAP041	220	39	5	84.007,9	HPIRF-FOpt
IAP042	256	70	5	121.683,1	HAPSD
IAP043	194	44	5	88.894,2	HPIRF-FOpt
IAP044	264	66	5	110.294,4	HPIRF-FOpt
IAP045	217	67	5	117.641,9	HAPSD
IAP046	240	68	5	119.909,3	HAPSD
IAP047	256	47	5	88.767,0	HAPSD
IAP048	356	77	7	154.149,4	HAPSD
IAP049	314	66	7	146.533,4	HPIRF-FOpt
IAP050	306	79	7	140.755,2	HAPSD
IAP051	343	96	7	146.716,7	HAPSD
IAP052	332	92	7	169.397,8	HAPSD
IAP053	478	91	10	185.663,1	HAPSD
IAP054	452	115	10	214.751,3	HAPSD
IAP055	483	133	10	233.446,3	HAPSD
IAP056	496	112	10	208.676,3	HAPSD

(*) Instâncias que a solução ótima foi encontrada

6.4 Conclusões

Neste capítulo foram apresentados, inicialmente, dois modelos matemáticos para resolver o problema integrado de agrupamento de pedidos e sequenciamento de

panelas (PIAPSP). O objetivo de resolver este problema é determinar os pedidos que formarão uma panela e depois determinar o sequenciamento destas panelas na máquina de lingotamento contínuo.

Comparando-se o desempenho dos dois modelos (PIPED e PIPAD-Bk), o modelo PIPED obteve o melhor desempenho, conseguindo encontrar, dentro do tempo estipulado, uma solução para todas as instâncias testadas. Um dos fatores que contribuiu para o melhor desempenho do modelo PIPED está associado ao número de variáveis inteiras. Pode-se observar pela Tabela 6.15 que o número de variáveis inteiras do modelo PIPED é bem menor que o número de variáveis inteiras do modelo PIPAD-Bk. O número de variáveis inteiras do modelo PIPAD-Bk é 56,1 vezes maior que o número de variáveis inteiras do modelo PIPED na instância IAP025 e 8,6 vezes maior na instância IAP024. Apesar disto, o modelo PIPAD-Bk encontrou 19 das melhores soluções para as instâncias artificiais (IAP001 a IAP024), enquanto que o modelo PIPED encontrou apenas 7 das melhores soluções para estas mesmas instâncias.

Tabela 6.15: Comparativo entre os modelos PIPED e PIPAD-Bk

Instância	NPd	NPa	Modelo PIPED			Modelo PIPAD-Bk		
			NVT	NVI	NR	NVT	NVI	NR
IAP024	100	47	43.209	18.140	23.341	316.180	155.470	168.351
IAP025	212	51	171.638	70.410	75.919	7.919.794	3.953.140	3.982.624

Onde:

NPd = Número de Pedidos; NPa = Número de padrões; NVT = Número total de variáveis; NVI = Número de Variáveis Inteiras; e NR = número de restrições.

Ainda neste capítulo, foram apresentadas duas heurísticas para resolver o PIAPSP. A primeira heurística, denominada HAPSD, procura, primeiramente, agrupar os pedidos em panelas e em seguida determinar o sequenciamento dessas panelas na máquina de lingotamento contínuo. Este procedimento é feito período por período. A segunda heurística baseia-se nas técnicas *relax-and-fix* e *fix-and-optimize*. A heurística HAPSD apresentou melhor desempenho que as heurísticas baseadas nas técnicas *relax-and-fix* e *fix-and-optimize*, encontrando 29 melhores soluções para todas as instâncias, enquanto que as heurísticas baseadas nas técnicas *relax-and-fix* e *fix-and-optimize* encontraram 7 melhores soluções.

Além disso, pôde-se observar que é vantajoso fazer a integração dos problemas, uma vez que situações como a encontrada na instância IRM42, onde a solução ótima deixa dois *jobs* isolados (seção 4.2.2), podem ser evitadas. Com a integração, panelas com composição química e largura compatíveis podem ser agrupadas, dentro de um período, em uma mesma série, evitando que situações como a da instância IRM42

possam acontecer, diminuindo desta forma, o número de operações de *setup* realizadas.

Capítulo 7

Conclusões

Este trabalho teve foco no problema integrado de agrupamento de pedidos e sequenciamento de painéis em um processo de LC. O estudo baseou-se em um caso real de empresa siderúrgica brasileira.

Apresentou-se, inicialmente, quatro modelos e uma heurística para a resolução do problema de sequenciamento de painéis. Os modelos desenvolvidos configuraram-se como uma forte contribuição do trabalho. O modelo P foi o modelo que apresentou melhores resultados, sendo capaz de resolver rapidamente problemas com até 72 painéis. Ao conhecimento do autor, é a primeira vez que tal problema é resolvido fazendo-se analogia com o problema de roteamento de veículos assimétrico. A heurística desenvolvida baseou-se em uma analogia do PSP com o problema de árvore geradora mínima capacitada. Os resultados obtidos com a heurística também foram satisfatórios. A heurística foi capaz de encontrar boas soluções em um tempo computacional reduzido. Além disso, tanto a heurística quanto os modelos provaram gerar soluções melhores que a ESB, mesmo quando o número de *setups* era fixado.

Em seguida, apresentou-se cinco modelos de PLIM e uma heurística para a resolução do PAP. Os cinco modelos apresentaram desempenhos semelhantes, com uma certa vantagem para o modelo PAD-Bk, que foi capaz de resolver um maior número de instâncias no ótimo. Os modelos encontrados na literatura para resolver o PAP consideravam o pedido do cliente em número de placas e neste trabalho utilizou-se o pedido do cliente em toneladas. A heurística desenvolvida incorporava as estratégias *relax-and-fix* e *fix-and-optimize*, apesar das técnicas não serem novas, ao conhecimento do autor, nenhuma delas tinha sido aplicado ao PAP. Resolveu-se utilizar as duas estratégias, pois verificou-se que os resultados obtidos utilizando-se apenas com a técnica

relax-and-fix não eram bons. Os resultados obtidos com a heurística, assim como os resultados obtidos pelos modelos foram muito satisfatórios, apenas em poucas situações a solução ótima não foi encontrada.

E, finalmente, apresentou-se dois modelos de PLIM e duas heurísticas para resolver o problema integrado de agrupamento de pedidos e sequenciamento de painelas. Os dois modelos não apresentaram bons resultados, sendo capazes de resolver de forma satisfatória apenas instâncias com 100 ou menos pedidos. Conforme afirmou Tang *et al.* [51], não há trabalhos que tratem dos problemas PAP e PSP de forma integrada, portanto estes modelos são uma boa contribuição para a literatura existente, assim como as heurísticas desenvolvidas.

As heurísticas desenvolvidas para resolver o PIAPSP, ao contrário dos modelos, já foram capazes de encontrar boas soluções para instâncias com mais de 200 pedidos, que correspondem ao problema real da ESB. A heurística HAPSD foi a que encontrou o maior número de melhores soluções para as 56 instâncias testadas.

Em entrevista realizada com quatro siderúrgicas, verificou-se que todas possuem características muito semelhantes, tornando este trabalho uma importante contribuição para o setor industrial. Nenhuma das empresas entrevistadas possuía um *software* específico para gerar e avaliar os planos de produção gerados. Além disso, encontrou-se na literatura diversos casos de sucesso de implantação de sistemas de suporte à decisão em empresas do setor siderúrgico, o que evidencia um enorme potencial de ganho para a ESB. Contudo, para a aplicação dos métodos descritos neste trabalho, é necessário o comprometimento total da empresa e de todos os seus colaboradores. Com a implantação dos métodos descritos neste trabalho, a empresa ganha: (i) agilidade na determinação do seu plano de produção, (ii) redução dos atrasos nas entregas dos pedidos e (iii) melhor aproveitamento dos seus recursos produtivos.

Como sugestão para trabalhos futuros, sugere-se o desenvolvimento de métodos de resolução para o problema integrado de agrupamento de pedidos e sequenciamento de painelas, que procurem reduzir o tempo computacional para a obtenção das soluções. Um exemplo seria a aplicação de meta-heurísticas (por exemplo, algoritmos genéticos) para determinar as séries e a sequência das painelas em cada período e em seguida uma heurística definiria a quantidade de cada pedido seria alocada a cada painela.

É interessante, também, obter mais informações junto à empresa para uma melhor avaliação do desempenho dos métodos propostos nos Capítulos 5 e 6, de tal forma que se possa medir o ganho para a empresa com a aplicação dos métodos descritos

neste trabalho.

Referências Bibliográficas

- [1] AMBERG, A.; DOMSCHKE, W.; VOSS, S. “Capacited minimum spanning tree: Algorithms using intelligent search”. *Combinatorial Optimization: Theory and Practice*, v. 1, p. 9–40, 1996.
- [2] ARAUJO, S. A.; ARENALES, M. N.; CLARK, A. R. “Lot sizing and furnace scheduling in small foundries”. *Computers & Operations Research*, v. 35, p. 916–932, 2008.
- [3] ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. *Pesquisa Operacional para Cursos de Engenharia*. Editora Campus - Elsevier, Rio de Janeiro-RJ, 1 ed., 2006.
- [4] ATIGHEHCHIAN, A.; BIJARI, M.; TARKESH, H. “A novel hybrid algorithm for scheduling steel-making continuous casting production”. *Computers & Operations Research*, v. 36(8), p. 2450–2461, 2009.
- [5] BAKER, K. R. *Introduction to Sequencing and Scheduling*. John Wiley, New York-NY, 1974.
- [6] BALDO, T. A.; SANTOS, M. O.; ALMADA-LOBO, B.; MORABITO, R. “An optimization approach for the lot sizing and scheduling problem in the brewery industry”. *Computers & Industrial Engineering*, v. 72, p. 58–71, 2014.
- [7] BELLABDAOUI, A.; TEGHEM, J. “A mixed-integer linear programming model for the continuous casting planning”. *International Journal of Production Economics*, v. 104, p. 260–270, 2006.
- [8] BRETAS, A. M. C. *Modelagem do Sequenciamento no Processo de Lingotamento Contínuo por Métodos Exatos e Heurísticos*. Dissertação de mestrado, Programa de Pós Graduação em Engenharia de Produção, UFMG, Belo Horizonte, 2013.

- [9] BUSTAMANTE, L. M. *Minimização do Custo de Antecipação e Atraso para o Problema de Seqüenciamento de uma Máquina com Tempo de Preparação Dependente da Seqüência: Aplicação em uma Usina Siderúrgica*. Dissertação de mestrado, Programa de Pós Graduação em Engenharia de Produção, UFMG, Belo Horizonte, 2006.
- [10] CLARK, A. R.; MORABITO, R.; TOSO, E. A. V. “Production setup-sequencing and lot-sizing at an animal nutrition plant through ATSP subtour elimination and patching”. *Journal of Scheduling*, v. 13, p. 111–121, 2010.
- [11] DE SOUZA, M. C.; GOMES JÚNIOR, A. C.; BRETAS, A. M. C.; RAVETTI, M. G. “Models for scheduling charges in continuous casting: application to a Brazilian steel plant”. *Optimization Letters*, 2015, DOI 10.1007/s11590-015-0901-8.
- [12] DE SOUZA, M. C.; RAVETTI, M. G.; GOMES JÚNIOR, A. C.; BRETAS, A. M. C. “Models for Scheduling in Continuous Ingot Casting with Parallel Machine”. *In: 9th International Conference on Computational Management Science, London-GBR*, v. 1, p. 85–86, 2012.
- [13] DE SOUZA, M. C.; RAVETTI, M. G.; GOMES JÚNIOR, A. C.; BRETAS, A. M. C. “Scheduling in Continuous Ingot Casting”. *In: 14ème Conférence ROADEF de la société Française de Recherche Opérationnelle et Aide à la Décision, Troyes-FR*, p. 86, 2013.
- [14] DREXL, A.; KIMMS, A. “Lot sizing and scheduling - Survey and extensions”. *European Journal of Operational Research*, v. 99, p. 221–235, 1997.
- [15] DYER, M. E.; WOLSEY, L. A. “Formulating the single machine sequencing problem with release dates as a mixed integer program”. *Discrete Applied Mathematics*, v. 26, p. 255–270, 1990.
- [16] ESAU, L. R.; WILLIAMS, K. C. “On teleprocessing system design”. *IBM Systems Journal*, v. 5, p. 142–147, 1966.
- [17] FERREIRA, D.; MORABITO, R.; RANGEL, S. “Solution approaches for the soft drink integrated production lot sizing and scheduling problem”. *European Journal of Operational Research*, v. 196, p. 697–706, 2009.

- [18] FERREIRA, D.; MORABITO, R.; RANGEL, S. “Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants”. *Computers & Operations Research*, v. 37, p. 684–691, 2010.
- [19] FERRETTI, I.; ZANONI, S.; ZAVANELLA, L. “Production-inventory scheduling using Ant System metaheuristic”. *International Journal of Production Economics*, v. 104, p. 317–326, 2006.
- [20] FIGUEIRA, G.; SANTOS, M. O.; ALMADA LOBO, B. “A hybrid VNS approach for the short-term production planning and scheduling”. *Computers & Operations Research*, v. 40, p. 1804–1818, 2013.
- [21] GOMES JÚNIOR, A. C.; DE SOUZA, M. C.; BRETAS, A. M. C.; RAVETTI, M. G. “Modelos de Programação Linear Inteira Mista para o problema de agrupamento de pedidos e sequenciamento de painéis na aciaria: um estudo de caso em uma siderúrgica brasileira”. *Anais do XLVI Simpósio Brasileiro de Pesquisa Operacional, Salvador-BA*, p. 1472–1483, 2014.
- [22] HARJUNKOSKI, I.; GROSSMANN, I. E. “A decomposition approach for the scheduling of a steel plant production”. *Computers and Chemical Engineering*, v. 25, p. 1647–1660, 2001.
- [23] HUANG, S.; CAI, L.; ZHANG, X. “Parallel dedicated machine scheduling problem with sequence-dependent setups and a single server”. *Computers & Industrial Engineering*, v. 58, p. 165–174, 2010.
- [24] JIN, F.; SONG, S.; WU, C. “A simulated annealing algorithm for single machine scheduling problems with family setups”. *Computers & Operations Research*, v. 36, p. 2133–2138, 2009.
- [25] KAWAMURA, M. S.; RONCONI, D. P. “Aplicação da Heurística Relax-and-Fix no Problema de Dimensionamento e Sequenciamento de Lotes de Produção em Máquinas Distintas em Paralelo”. In *Anais do XLII Simpósio Brasileiro de Pesquisa Operacional*, p. 1141–1152, 2010.
- [26] KEHA, A. B.; KHOWALA, K.; FOWLER, J. W. “Mixed integer programming formulations for single machine scheduling problems”. *Computers & Industrial Engineering*, v. 56, p. 357–367, 2009.

- [27] KIM, D.; NA, D.; CHENB, F. F. “Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective”. *Robotics and Computer Integrated Manufacturing*, v. 19, p. 173–181, 2003.
- [28] LASSERRE, J. B. “An integrated model for job-shop planning and scheduling”. *Management Science*, v. 38, p. 1201–1211, 1992.
- [29] LI, J.; XIAO, X.; TANG, Q.; FLOUDAS, C. A. “Production Scheduling of a Large-Scale Steelmaking Continuous Casting Process via Unit-Specific Event-Based Continuous-Time Models: Short-Term and Medium-Term Scheduling”. *Industrial & Engineering Chemistry Research*, v. 51, p. 7300–7319, 2012.
- [30] MANNE, A. S. “On the Job-shop Scheduling Problem”. *Operations Research*, v. 8, p. 219–223, 1960.
- [31] MAO, K.; PAN, Q.; PANG, X.; CHAI, T. “A novel Lagrangian relaxation approach for a hybrid flowshop scheduling problem in the steelmaking-continuous casting process”. *European Journal of Operational Research*, v. 236, p. 51–60, 2014.
- [32] MARTINS, A. X.; DE SOUZA, M. C.; SOUZA, M. J. F.; TOFFOLO, T. A. M. “GRASP with hybrid heuristic-subproblem optimization for the multi-level capacitated minimum spanning tree problem”. *Journal of Heuristics*, v. 15, p. 133–151, 2009.
- [33] MATEUS, G. R.; RAVETTI, M. G.; DE SOUZA, M. C.; VALERIANO, T. M. “Capacitated lot sizing and sequence dependent setup scheduling: an iterative approach for integration”. *Journal of Scheduling*, v. 13, p. 245–259, 2010.
- [34] MEYR, H. “Simultaneous lotsizing and scheduling on parallel machines”. *European Journal of Operational Research*, v. 139, p. 277–292, 2002.
- [35] MEYR, H.; MANN, M. “A decomposition approach for the General Lotsizing and Scheduling Problem for Parallel production Lines”. *European Journal of Operational Research*, v. 229, p. 718–731, 2013.
- [36] MISSBAUER, H.; HAUBER, W.; STADLER, W. “A scheduling system for the steelmaking continuous casting process: a case study from the steel-making industry”. *International Journal of Production Research*, v. 47, p. 4147–4172, 2009.

- [37] OWEN, J. H.; MEHROTRA, S. “On the value of binary expansions for general mixed-integer linear programs”. *Operations Research*, v. 50, p. 810–819, 2002.
- [38] PACCIARELLI, D.; PRANZO, M. “Production scheduling in a steelmaking-continuous casting plant”. *Computers and Chemical Engineering*, v. 28, p. 2823–2835, 2004.
- [39] PAN, Q.; WANG, L.; MAO, K.; ZHAO, J.; ZHANG, M. “An effective artificial bee colony algorithm for a real-world hybrid flowshop problem in steelmaking process”. *IEEE Transactions on Automation Science and Engineering*, v. 10, p. 307–322, 2013.
- [40] PAULA, M. R.; MATEUS, G. R.; RAVETTI, M. G. “A non-delayed relax-and-cut algorithm for scheduling problems with parallel machines, due dates and sequence-dependent setup times”. *Computers & Operations Research*, v. 37, p. 938–949, 2010.
- [41] PINEDO, M. L. *Scheduling: Theory, Algorithms, and Systems*. Springer-Science, 3 ed., 2008.
- [42] RIZZO, E. M. S. *Introdução aos Processos de Refino Primário dos Aços nos Convertedores a Oxigênio*. Associação Brasileira de Metalurgia e Materiais, 2006.
- [43] RIZZO, E. M. S. *Introdução aos Processos de Refino Secundário dos Aços*. Associação Brasileira de Metalurgia e Materiais, São Paulo-SP, 2006.
- [44] ROCHA, P. L.; RAVETTI, M. G.; MATEUS, G. R.; PARDALOS, P. M. “Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times”. *Computers & Operations Research*, v. 35, p. 1250–1264, 2008.
- [45] SEEANNER, F.; ALMADA-LOBO, B.; MEYR, H. “Lot sizing and scheduling problem with earliness tardiness and setup penalties”. *Computers & Operations Research*, v. 40, p. 303–317, 2013.
- [46] SHIM, I.; KIM, H.; DOH, H.; LEE, D. “A two-stage heuristic for single machine capacitated lot-sizing and scheduling with sequence-dependent setup costs”. *Computers & Industrial Engineering*, v. 61, p. 920–929, 2011.

- [47] SUPITHAK, W.; LIMAN, S. D.; MONTES, E. J. “Combining the principles of VND search and the fix-optimize heuristic to solve multi-level lot-sizing and scheduling problems”. *Computers & Operations Research*, v. 58, p. 363–372, 2013.
- [48] TANG, L.; LIU, J.; RONG, A.; YANG, Z. “A mathematical programming model for scheduling steelmaking-continuous casting production”. *European Journal of Operational Research*, v. 120, p. 423–435, 2000.
- [49] TANG, L.; LIU, J.; RONG, A.; YANG, Z. “A review of planning and scheduling systems and methods for integrated steel production”. *European Journal of Operational Research*, v. 133, p. 1–20, 2001.
- [50] TANG, L.; LUH, P. B.; LIU, J.; FANG, L. “Steel-Making process scheduling using lagrangian relaxation”. *International Journal of Production Research*, v. 40, p. 55–70, 2002.
- [51] TANG, L.; WANG, G.; CHEN, Z. L. “Integrated charge batching and casting width selection at Baosteel”. *Operations Research*, v. 62, p. 772–787, 2014.
- [52] TANG, L.; ZHAO, Y.; LIU, J. “An improved differential evolution algorithm for practical dynamic scheduling in steelmaking continuous casting production”. *IEEE Transactions on Evolutionary Computation*, v. 18, p. 209–225, 2014.
- [53] TANG, L. X.; LUO, J. “A new ILS algorithm for cast planning problem in steel industry”. *ISIJ International*, v. 47(3), p. 443–452, 2007.
- [54] TANG, L. X.; WANG, G. S. “Decision support system for the batching problems of steelmaking and continuous-casting production”. *Omega*, v. 36, p. 976–991, 2008.
- [55] TANG, L. X.; WANG, G. S.; LIU, J. Y.; LIU, J. “A combination of lagrangian relaxation and column generation for order batching in steelmaking and continuous-casting production”. *Naval Research Logistics*, v. 58, p. 370–388, 2011.
- [56] TOLEDO, C. F.; FRANÇA, P. M.; MORABITO, R.; KIMMS, A. “Um modelo de otimização para o problema integrado de dimensionamento de lotes e programação da produção em fábricas de refrigerantes”. *Pesquisa Operacional*, v. 27, p. 155–186, 2007.
- [57] TOLEDO, C. F.; OLIVEIRA, L.; PEREIRA, R. F.; FRANÇA, P. M.; MORABITO, R. “A genetic algorithm/mathematical programming approach to solve

- a two-level soft drink production problem”. *Computers & Operations Research*, v. 48, p. 40–52, 2014.
- [58] TOSO, E. A. V.; MORABITO, R.; CLARK, A. “Combinação de abordagens GLSP e ATSP para o problema de dimensionamento e sequenciamento de lotes de produção de suplementos para nutrição animal”. *Pesquisa Operacional*, v. 28, p. 423–450, 2008.
- [59] VAN-DEN-AKKER, J.; HURKENS, C.; SAVELSBERGH, M. “Time-indexed formulation for machine scheduling problems: column generation”. *INFORMS J. on Computing*, v. 12, p. 111–124, 2000.
- [60] WAGNER, H. M. “An Integer Programming Model for Machine Scheduling”. *Naval Research Logistics Quarterly*, v. 6, p. 131–140, 1959.
- [61] ZANONI, S.; ZAVANELLA, L. “Model and analysis of integrated production-inventory system: The case of steel production”. *International Journal of Production Economics*, v. 93-94, p. 197–205, 2005.

Apêndice A

Resultados da Heurística RF-FOpt para o PAP

As tabelas A.1 a A.6 apresentam os resultados para a heurística RF-FOpt com as diferentes estratégias “*relax-and-fix*” adotadas. A estrutura destas tabelas é definida como se segue. A coluna “Inst.” apresenta a instância resolvida, a coluna “ CT^* ” apresenta o melhor resultado encontrado para aquela instância através da resolução dos modelos para o PAP. Já a coluna “FOpt” apresenta a estratégia “*fix-and-optimize*” usada na heurística. A coluna “ C_{RF} ” apresenta o custo total da solução obtida pela aplicação apenas da heurística RF e a coluna “ $T_{RF}(s)$ ” o tempo computacional para obtenção desta solução. As colunas “ CT ” e “ $T(s)$ ” apresentam, respectivamente, o custo total da solução obtida pela heurística e o tempo computacional gasto para obtenção desta solução. E a coluna “ $D(\%)$ ” apresenta o desvio do custo total da solução obtida pela heurística em relação ao custo da melhor solução obtida para aquela instância. O valor de D é calculado conforme a equação A.1.

$$D = \frac{CT - CT^*}{CT^*} \times 100 \quad (\text{A.1})$$

Tabela A.1: Resultados da Heurística usando a estratégia RF-zt1f

Inst.	CT^*	FOpt	C_{RF}	$T_{RF}(s)$	CT	$T(s)$	$D(\%)$
IAP024	144.081,4	zt1f	1.126.510,3	38	246.274,7	102	70,9
IAP026	382.614,9	zt1f	481.749,7	66	414.820,0	212	8,4
IAP031	104.010,5	zt1f	114.148,0	89	113.255,2	194	8,9
IAP032	100.486,4	zt1f	165.862,5	172	118.632,6	386	18,1
IAP036	316.921,6	zt1f	451.374,6	982	378.234,4	5254	19,3
IAP037	19.244,8	zt1f	47.464,6	95	47.464,6	181	146,6
IAP048	11.042,4	zt1f	24.240,5	46	24.240,5	86	119,5
IAP053	11.216,1	zt1f	33.618,9	124	33.618,9	267	199,7
IAP055	4.916,2	zt1f	6.184,0	312	6.184,0	424	25,8
IAP056	4.598,3	zt1f	5.835,2	131	5.835,2	202	26,9
IAP024	144.081,4	zp1f	1.126.510,3	38	267.522,9	174	85,7
IAP026	382.614,9	zp1f	481.749,7	57	452.643,5	272	18,3
IAP031	104.010,5	zp1f	114.148,0	61	107.652,4	301	3,5
IAP032	100.486,4	zp1f	165.862,5	105	107.977,1	469	7,5
IAP036	316.921,6	zp1f	451.374,6	884	328.239,2	4866	3,6
IAP037	19.244,8	zp1f	47.464,6	91	19.271,9	564	0,1
IAP048	11.042,4	zp1f	24.240,5	41	11.155,9	145	1,0
IAP053	11.216,1	zp1f	33.618,9	127	11.216,1	555	0,0
IAP055	4.916,2	zp1f	6.184,0	282	5.071,3	1583	3,2
IAP056	4.598,3	zp1f	5.835,2	136	4.715,2	1427	2,5
IAP024	144.081,4	ztpf	1.126.510,3	97	191.039,2	1519	32,6
IAP026	382.614,9	ztpf	481.749,7	122	391.847,8	3437	2,4
IAP031	104.010,5	ztpf	114.148,0	200	107.652,4	1858	3,5
IAP032	100.486,4	ztpf	165.862,5	400	108.346,6	1709	7,8
IAP036	316.921,6	ztpf	451.360,3	1445	336.279,9	10009	6,1
IAP037	19.244,8	ztpf	47.464,6	134	19.244,8	2879	0,0
IAP048	11.042,4	ztpf	24.240,5	177	11.042,4	723	0,0
IAP053	11.216,1	ztpf	33.618,9	281	11.216,1	1256	0,0
IAP055	4.916,2	ztpf	6.184,0	435	5.071,3	2321	3,2
IAP056	4.598,3	ztpf	5.835,2	170	4.715,2	2150	2,5
IAP024	144.081,4	zptf	1.126.510,3	49	188.937,5	641	31,1
IAP026	382.614,9	zptf	481.749,7	64	391.847,8	1187	2,4
IAP031	104.010,5	zptf	114.148,0	118	107.652,4	635	3,5
IAP032	100.486,4	zptf	165.862,5	177	106.323,5	914	5,8
IAP036	316.921,6	zptf	451.374,6	1137	326.382,8	7873	3,0
IAP037	19.244,8	zptf	47.464,6	119	19.244,8	826	0,0
IAP048	11.042,4	zptf	24.240,5	50	11.042,4	217	0,0
IAP053	11.216,1	zptf	33.618,9	135	11.216,1	680	0,0
IAP055	4.916,2	zptf	6.184,0	313	5.071,3	1985	3,2
IAP056	4.598,3	zptf	5.835,2	209	4.715,2	2127	2,5

Tabela A.2: Resultados da Heurística usando a estratégia RF-zt2f

Inst.	CT^*	FOpt	C_{RF}	$T_{RF}(s)$	CT	$T(s)$	$D(\%)$
IAP024	144.081,4	zt1f	2.436.096,6	160	2.436.096,6	196	1590,8
IAP026	382.614,9	zt1f	405.212,5	524	401.277,8	672	4,9
IAP031	104.010,5	zt1f	114.075,2	167	114.075,2	1378	9,7
IAP032	100.486,4	zt1f	131.223,9	518	116.171,4	625	15,6
IAP036	316.921,6	zt1f	416.370,7	1671	366.576,8	3840	15,7
IAP037	19.244,8	zt1f	19.271,9	253	19.244,8	1737	0,0
IAP048	11.042,4	zt1f	11.671,3	31	11.671,3	72	5,7
IAP053	11.216,1	zt1f	13.422,4	70	13.422,4	713	19,7
IAP055	4.916,2	zt1f	6.160,0	819	6.160,0	1809	25,3
IAP056	4.598,3	zt1f	5.798,3	268	5.798,3	1162	26,1
IAP024	144.081,4	zp1f	2.436.096,6	143	2.427.574,1	321	1584,9
IAP026	382.614,9	zp1f	405.212,5	478	393.111,5	722	2,7
IAP031	104.010,5	zp1f	114.075,2	203	114.075,2	390	9,7
IAP032	100.486,4	zp1f	131.223,9	386	131.223,9	680	30,6
IAP036	316.921,6	zp1f	416.308,0	1779	416.308,0	3117	31,4
IAP037	19.244,8	zp1f	19.271,9	230	19.271,9	657	0,1
IAP048	11.042,4	zp1f	11.671,3	32	11.042,4	114	0,0
IAP053	11.216,1	zp1f	13.422,4	70	13.422,4	1587	19,7
IAP055	4.916,2	zp1f	6.160,0	573	6.160,0	1411	25,3
IAP056	4.598,3	zp1f	5.798,3	229	5.798,3	1012	26,1
IAP024	144.081,4	ztpf	2.436.096,6	155	218.249,1	957	51,5
IAP026	382.614,9	ztpf	405.212,5	562	389.727,8	1097	1,9
IAP031	104.010,5	ztpf	114.075,2	184	114.075,2	1659	9,7
IAP032	100.486,4	ztpf	131.223,9	426	100.486,4	773	0,0
IAP036	316.921,6	ztpf	405.436,1	1741	319.085,0	5497	0,7
IAP037	19.244,8	ztpf	19.271,9	177	19.244,8	2238	0,0
IAP048	11.042,4	ztpf	11.671,3	27	11.042,4	150	0,0
IAP053	11.216,1	ztpf	13.422,4	68	13.422,4	1949	19,7
IAP055	4.916,2	ztpf	6.160,0	608	6.160,0	2085	25,3
IAP056	4.598,3	ztpf	5.798,3	229	5.798,3	1736	26,1
IAP024	144.081,4	zptf	2.436.096,6	138	218.249,1	569	51,5
IAP026	382.614,9	zptf	405.212,5	545	389.727,8	981	1,9
IAP031	104.010,5	zptf	114.075,2	221	114.075,2	1604	9,7
IAP032	100.486,4	zptf	131.223,9	424	131.223,9	2140	30,6
IAP036	316.921,6	zptf	416.308,0	1842	342.737,6	4172	8,1
IAP037	19.244,8	zptf	19.271,9	228	19.244,8	882	0,0
IAP048	11.042,4	zptf	11.671,3	29	11.042,4	128	0,0
IAP053	11.216,1	zptf	13.422,4	70	13.422,4	2482	19,7
IAP055	4.916,2	zptf	6.160,0	589	6.160,0	1998	25,3
IAP056	4.598,3	zptf	5.798,3	287	5.798,3	2139	26,1

Tabela A.3: Resultados da Heurística usando a estratégia RF-zt1b

Inst.	CT^*	FOpt	C_{RF}	$T_{RF}(s)$	CT	$T(s)$	$D(\%)$
IAP024	144.081,4	zt1f	186.017,6	43	186.017,6	101	29,1
IAP026	382.614,9	zt1f	561.272,2	24	404.195,1	227	5,6
IAP031	104.010,5	zt1f	115.210,5	139	115.210,5	179	10,8
IAP032	100.486,4	zt1f	147.190,6	53	125.218,9	488	24,6
IAP036	316.921,6	zt1f	461.458,9	530	389.798,1	2201	23,0
IAP037	19.244,8	zt1f	32.844,1	53	32.844,1	83	70,7
IAP048	11.042,4	zt1f	17.212,1	16	15.522,4	36	40,6
IAP053	11.216,1	zt1f	15.056,1	41	15.056,1	68	34,2
IAP055	4.916,2	zt1f	5.592,2	70	5.592,2	120	13,8
IAP056	4.598,3	zt1f	4.654,3	83	4.654,3	160	1,2
IAP024	144.081,4	zp1f	186.017,6	21	186.017,6	185	29,1
IAP026	382.614,9	zp1f	561.272,2	22	502.734,5	437	31,4
IAP031	104.010,5	zp1f	115.210,5	23	104.010,5	224	0,0
IAP032	100.486,4	zp1f	147.190,6	34	106.609,4	168	6,1
IAP036	316.921,6	zp1f	461.458,9	205	326.419,4	1856	3,0
IAP037	19.244,8	zp1f	32.844,1	46	19.271,9	355	0,1
IAP048	11.042,4	zp1f	17.212,1	14	11.706,2	92	6,0
IAP053	11.216,1	zp1f	15.056,1	35	11.216,1	235	0,0
IAP055	4.916,2	zp1f	5.592,2	71	4.952,2	788	0,7
IAP056	4.598,3	zp1f	4.654,3	70	4.654,3	471	1,2
IAP024	144.081,4	ztpf	186.017,6	41	186.017,6	549	29,1
IAP026	382.614,9	ztpf	561.272,2	61	382.614,9	2358	0,0
IAP031	104.010,5	ztpf	115.210,5	74	104.010,5	657	0,0
IAP032	100.486,4	ztpf	147.190,6	33	100.486,4	577	0,0
IAP036	316.921,6	ztpf	461.458,9	575	320.444,2	3688	1,1
IAP037	19.244,8	ztpf	32.844,1	133	19.244,8	1145	0,0
IAP048	11.042,4	ztpf	17.212,1	33	11.042,4	148	0,0
IAP053	11.216,1	ztpf	15.056,1	64	11.216,1	394	0,0
IAP055	4.916,2	ztpf	5.592,2	111	4.952,2	2818	0,7
IAP056	4.598,3	ztpf	4.654,3	119	4.654,3	1531	1,2
IAP024	144.081,4	zptf	186.017,6	30	186.017,6	137	29,1
IAP026	382.614,9	zptf	561.272,2	17	385.318,5	533	0,7
IAP031	104.010,5	zptf	115.210,5	22	104.010,5	497	0,0
IAP032	100.486,4	zptf	147.190,6	65	100.486,4	345	0,0
IAP036	316.921,6	zptf	461.458,9	255	321.316,7	2602	1,4
IAP037	19.244,8	zptf	32.844,1	131	19.244,8	748	0,0
IAP048	11.042,4	zptf	17.212,1	14	11.042,4	144	0,0
IAP053	11.216,1	zptf	15.056,1	62	11.216,1	937	0,0
IAP055	4.916,2	zptf	5.592,2	201	4.952,2	3770	0,7
IAP056	4.598,3	zptf	4.654,3	99	4.654,3	1523	1,2

Tabela A.4: Resultados da Heurística usando a estratégia RF- τ 1f

Inst.	CT^*	FOpt	C_{RF}	$T_{RF}(s)$	CT	$T(s)$	$D(\%)$
IAP024	144.081,4	zt1f	6.114.790,1	13	521.727,5	23	262,1
IAP026	382.614,9	zt1f	482.242,4	21	456.852,1	33	19,4
IAP031	104.010,5	zt1f	114.148,0	16	114.148,0	33	9,7
IAP032	100.486,4	zt1f	304.906,0	56	213.827,6	81	112,8
IAP036	316.921,6	zt1f	591.891,7	178	398.149,9	331	25,6
IAP037	19.244,8	zt1f	321.036,2	300	80.419,1	366	317,9
IAP048	11.042,4	zt1f	130.213,8	20	22.372,4	39	102,6
IAP053	11.216,1	zt1f	204.063,9	72	30.496,1	114	171,9
IAP055	4.916,2	zt1f	6.216,0	196	6.216,0	235	26,4
IAP056	4.598,3	zt1f	6.328,9	109	6.328,9	244	37,6
IAP024	144.081,4	zp1f	6.114.790,1	11	518.619,6	64	259,9
IAP026	382.614,9	zp1f	482.242,4	20	482.189,7	150	26,0
IAP031	104.010,5	zp1f	114.148,0	23	114.148,0	134	9,7
IAP032	100.486,4	zp1f	304.906,0	106	194.024,8	307	93,1
IAP036	316.921,6	zp1f	591.891,7	423	358.954,6	2735	13,3
IAP037	19.244,8	zp1f	321.036,2	144	64.809,9	1026	236,8
IAP048	11.042,4	zp1f	130.213,8	46	21.869,6	461	98,1
IAP053	11.216,1	zp1f	204.063,9	63	29.054,2	753	159,0
IAP055	4.916,2	zp1f	6.216,0	333	6.216,0	1298	26,4
IAP056	4.598,3	zp1f	6.328,9	407	6.328,9	2423	37,6
IAP024	144.081,4	ztpf	6.114.790,1	26	517.813,8	156	259,4
IAP026	382.614,9	ztpf	482.242,4	79	424.784,8	1251	11,0
IAP031	104.010,5	ztpf	114.148,0	36	114.148,0	250	9,7
IAP032	100.486,4	ztpf	304.906,0	109	192.487,2	946	91,6
IAP036	316.921,6	ztpf	591.891,7	393	356.909,1	2156	12,6
IAP037	19.244,8	ztpf	321.036,2	149	64.809,9	666	236,8
IAP048	11.042,4	ztpf	130.213,8	19	21.092,4	125	91,0
IAP053	11.216,1	ztpf	204.063,9	73	27.296,1	843	143,4
IAP055	4.916,2	ztpf	6.216,0	199	6.216,0	945	26,4
IAP056	4.598,3	ztpf	6.328,9	120	6.328,9	2167	37,6
IAP024	144.081,4	zptf	6.114.790,1	13	517.813,8	84	259,4
IAP026	382.614,9	zptf	482.242,4	24	424.784,8	143	11,0
IAP031	104.010,5	zptf	114.148,0	13	114.148,0	61	9,7
IAP032	100.486,4	zptf	304.906,0	49	188.091,6	230	87,2
IAP036	316.921,6	zptf	591.891,7	372	356.909,1	1810	12,6
IAP037	19.244,8	zptf	321.036,2	88	64.809,9	1230	236,8
IAP048	11.042,4	zptf	130.213,8	77	21.092,4	306	91,0
IAP053	11.216,1	zptf	204.063,9	48	27.296,1	337	143,4
IAP055	4.916,2	zptf	6.216,0	199	6.216,0	838	26,4
IAP056	4.598,3	zptf	6.328,9	203	6.328,9	2708	37,6

Tabela A.5: Resultados da Heurística usando a estratégia RF- τ 1f

Inst.	CT^*	FOpt	C_{RF}	$T_{RF}(s)$	CT	$T(s)$	$D(\%)$
IAP024	144.081,4	zt1f	5.452.059,0	36	5.452.059,0	48	3684,0
IAP026	382.614,9	zt1f	405.485,2	399	401.776,0	449	5,0
IAP031	104.010,5	zt1f	114.075,2	100	114.075,2	873	9,7
IAP032	100.486,4	zt1f	342.226,6	339	146.701,4	365	46,0
IAP036	316.921,6	zt1f	523.350,7	910	370.668,1	1030	17,0
IAP037	19.244,8	zt1f	38.191,9	198	38.191,9	642	98,5
IAP048	11.042,4	zt1f	13.372,1	22	13.372,1	242	21,1
IAP053	11.216,1	zt1f	26.922,1	59	13.776,1	103	22,8
IAP055	4.916,2	zt1f	6.160,0	563	6.160,0	956	25,3
IAP056	4.598,3	zt1f	5.749,1	165	5.749,1	334	25,0
IAP024	144.081,4	zp1f	5.452.059,0	34	5.452.059,0	186	3684,0
IAP026	382.614,9	zp1f	405.485,2	302	405.432,5	414	6,0
IAP031	104.010,5	zp1f	114.075,2	80	114.075,2	268	9,7
IAP032	100.486,4	zp1f	342.226,6	417	342.226,6	810	240,6
IAP036	316.921,6	zp1f	523.350,7	1226	348.502,6	2727	10,0
IAP037	19.244,8	zp1f	38.191,9	201	38.191,9	457	98,5
IAP048	11.042,4	zp1f	13.372,1	32	13.372,1	87	21,1
IAP053	11.216,1	zp1f	26.922,1	67	26.922,1	706	140,0
IAP055	4.916,2	zp1f	6.160,0	526	6.160,0	889	25,3
IAP056	4.598,3	zp1f	5.749,1	147	5.749,1	672	25,0
IAP024	144.081,4	ztpf	5.452.059,0	65	5.452.059,0	434	3684,0
IAP026	382.614,9	ztpf	405.485,2	431	399.416,6	1531	4,4
IAP031	104.010,5	ztpf	114.075,2	133	114.075,2	1264	9,7
IAP032	100.486,4	ztpf	342.226,6	339	120.031,4	485	19,5
IAP036	316.921,6	ztpf	523.350,7	996	324.495,2	1919	2,4
IAP037	19.244,8	ztpf	38.191,9	304	38.191,9	1064	98,5
IAP048	11.042,4	ztpf	13.372,1	29	13.372,1	407	21,1
IAP053	11.216,1	ztpf	26.922,1	60	11.216,1	368	0,0
IAP055	4.916,2	ztpf	6.160,0	529	6.160,0	1409	25,3
IAP056	4.598,3	ztpf	5.749,1	159	5.749,1	987	25,0
IAP024	144.081,4	zptf	5.452.059,0	34	5.452.059,0	120	3684,0
IAP026	382.614,9	zptf	405.485,2	352	398.662,2	528	4,2
IAP031	104.010,5	zptf	114.075,2	83	114.075,2	988	9,7
IAP032	100.486,4	zptf	342.226,6	381	342.226,6	875	240,6
IAP036	316.921,6	zptf	523.350,7	864	348.182,6	1732	9,9
IAP037	19.244,8	zptf	38.191,9	205	38.191,9	1010	98,5
IAP048	11.042,4	zptf	13.372,1	26	13.372,1	262	21,1
IAP053	11.216,1	zptf	26.922,1	171	26.922,1	1114	140,0
IAP055	4.916,2	zptf	6.160,0	614	6.160,0	1605	25,3
IAP056	4.598,3	zptf	5.749,1	166	5.749,1	1237	25,0

Tabela A.6: Resultados da Heurística usando a estratégia RF-zp1f

Inst.	CT^*	FOpt	C_{RF}	$T_{RF}(s)$	CT	$T(s)$	$D(\%)$
IAP024	144.081,4	zt1f	176.292,4	345	165.095,1	662	14,6
IAP026	382.614,9	zt1f	611.875,0	582	385.563,0	761	0,8
IAP031	104.010,5	zt1f	197.149,7	168	114.077,9	409	9,7
IAP032	100.486,4	zt1f	122.019,3	321	113.300,4	559	12,8
IAP036	316.921,6	zt1f	558.658,8	1387	334.574,7	3560	5,6
IAP037	19.244,8	zt1f	29.273,3	732	19.244,8	1069	0,0
IAP048	11.042,4	zt1f	13.372,1	114	11.682,4	154	5,8
IAP053	11.216,1	zt1f	11.216,1	478	11.216,1	621	0,0
IAP055	4.916,2	zt1f	5.175,4	1319	5.095,3	1472	3,6
IAP056	4.598,3	zt1f	4.650,3	2193	4.650,3	2292	1,1
IAP024	144.081,4	zp1f	176.292,4	82	176.292,4	156	22,4
IAP026	382.614,9	zp1f	611.875,0	106	611.875,0	300	59,9
IAP031	104.010,5	zp1f	197.149,7	83	164.514,5	403	58,2
IAP032	100.486,4	zp1f	122.019,3	234	113.136,7	616	12,6
IAP036	316.921,6	zp1f	558.658,8	755	558.580,7	2596	76,3
IAP037	19.244,8	zp1f	29.273,3	282	24.404,0	763	26,8
IAP048	11.042,4	zp1f	13.372,1	66	11.706,2	136	6,0
IAP053	11.216,1	zp1f	11.216,1	230	11.216,1	353	0,0
IAP055	4.916,2	zp1f	5.175,4	650	5.175,4	1060	5,3
IAP056	4.598,3	zp1f	4.650,3	577	4.650,3	994	1,1
IAP024	144.081,4	ztpf	176.292,4	120	165.095,1	388	14,6
IAP026	382.614,9	ztpf	611.875,0	223	385.563,0	1116	0,8
IAP031	104.010,5	ztpf	197.149,7	178	104.010,5	1044	0,0
IAP032	100.486,4	ztpf	122.019,3	463	107.350,4	1056	6,8
IAP036	316.921,6	ztpf	558.658,8	1274	320.453,8	7326	1,1
IAP037	19.244,8	ztpf	29.273,3	749	19.244,8	1213	0,0
IAP048	11.042,4	ztpf	13.372,1	72	11.042,4	188	0,0
IAP053	11.216,1	ztpf	11.216,1	554	11.216,1	847	0,0
IAP055	4.916,2	ztpf	5.175,4	1143	5.095,3	3024	3,6
IAP056	4.598,3	ztpf	4.650,3	1214	4.650,3	1877	1,1
IAP024	144.081,4	zptf	176.292,4	56	165.095,1	264	14,6
IAP026	382.614,9	zptf	611.875,0	168	385.563,0	971	0,8
IAP031	104.010,5	zptf	197.149,7	101	107.652,4	1599	3,5
IAP032	100.486,4	zptf	122.019,3	444	105.328,4	1191	4,8
IAP036	316.921,6	zptf	558.658,8	1802	320.453,8	7064	1,1
IAP037	19.244,8	zptf	29.273,3	543	19.244,8	1908	0,0
IAP048	11.042,4	zptf	13.372,1	248	11.042,4	574	0,0
IAP053	11.216,1	zptf	11.216,1	265	11.216,1	414	0,0
IAP055	4.916,2	zptf	5.175,4	1064	5.095,3	2971	3,6
IAP056	4.598,3	zptf	4.650,3	2147	4.650,3	5268	1,1