UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

MÁRIO TONIZZA PEREIRA

Metaheurísticas Híbridas para a Programação Detalhada e Integrada de
Operações de Produção e Distribuição em Sistemas *No-Wait Flow Shop*

São Carlos
2021

MÁRIO TONIZZA PEREIRA

Metaheurísticas Híbridas para a Programação Detalhada e Integrada de
Operações de Produção e Distribuição em Sistemas *No-Wait Flow Shop*

Versão Corrigida

Tese apresentada à Escola de Engenharia
de São Carlos da Universidade de São
Paulo para obtenção do título de Doutor em
Ciências

Área de Concentração:
Processos e Gestão de Operações

Orientador:
Prof. Dr. Marcelo Seido Nagano

São Carlos
2021

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR
QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO
E PESQUISA, DESDE QUE CITADA A FONTE

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

P436m Pereira, Mario Tonizza
Metaheurísticas Híbridas para a Programação
Detalhada e Integrada de Operações de Produção e
Distribuição em Sistemas No-Wait Flow Shop / Mario
Tonizza Pereira; orientador Marcelo Seido Nagano.
São Carlos, 2021.

Tese (Doutorado) - Programa de Pós-Graduação em
Engenharia de Produção e Área de Concentração em
Processos e Gestão de Operações -- Escola de
Engenharia de São Carlos da Universidade de São
Paulo, 2021.

Versão corrigida.

1. Programação Flow Shop 2. Roteamento de
Veículos 3. Sistemas Integrados 4. Otimização
Combinatória 5. Metaheurísticas I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

FOLHA DE JULGAMENTO

Candidato: Engenheiro **MARIO TONIZZA PEREIRA.**

Título da tese: "Metaheurísticas Híbridas para a programação detalhada e integrada de operações de produção e distribuição em *Sistemas No- Wait Flow Shop*".

Data da defesa: 15/07/2021

Comissão Julgadora

Resultado

Prof. Dr. **Marcelo Seido Nagano** (Orientador)
(Escola de Engenharia de São Carlos/EESC-USP)

Aprovado

Prof. Associado **José Francisco Ferreira Ribeiro**
(Faculdade de Economia, Administração e Contabilidade-FEARP/USP)

Aprovado

Prof. Dr. **Roberto Fernandes Tavares Neto**
(Universidade Federal de São Carlos/UFSCar)

Aprovado

Prof. Dr. **José Luís Garcia Hermosilla**
(Universidade de Araraquara/UNIARA)

Aprovado

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

Aprovado

Coordenadora do Programa de Pós-Graduação em Engenharia de Produção:

Profa. Dra. **Janaina Mascarenhas Hornos da Costa**

Presidente da Comissão de Pós-Graduação:

Prof. Titular **Murilo Araujo Romero**

À minha família, Fabiana, Nicolás, Gustavo e Helena,
minhas eternas fontes de inspiração.
Aos meus pais, Eduardo (em memória) e Celina,
pela dedicação de uma vida.

AGRADECIMENTOS

Meu especial agradecimento ao Professor Dr. Marcelo Seido Nagano pela preciosa orientação, pelo incentivo, paciência e compreensão durante a execução desta pesquisa.

Ao meu diretor na empresa Oracle do Brasil Alberto Pereira pelo apoio e confiança ao longo desta jornada.

À minha querida esposa Fabiana, pelo incansável estímulo, apoio, ajuda, paciência e compreensão.

Aos meus queridos filhos, Nicolas, Gustavo e Helena, pela compreensão quanto aos momentos que foram subtraídos de nossa convivência diária para que eu pudesse desenvolver este trabalho.

Aos meus pais, Eduardo (em memória) e Celina, por terem me preparado para a vida e por terem me ensinado a perseverar e a valorizar tanto a educação e o conhecimento.

A todos aqueles que, direta ou indiretamente, contribuíram para a realização deste trabalho.

Dubito, ergo cogito, ergo sum.
(Questiono, logo penso, logo existo.)
(René Descartes)

RESUMO

PEREIRA, M. T. **Metaheurísticas Híbridas para a Programação Detalhada e Integrada de Operações de Produção e Distribuição em Sistemas *No-Wait Flow Shop***, 2021. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2021.

Nos complexos ambientes de negócio atuais, a otimização de processos operacionais é fator essencial à sobrevivência das organizações. Muitas empresas, esforçando-se cada vez mais para manterem-se competitivas, buscam por melhores métodos e modelos operacionais que possibilitem minimizar seus custos logísticos e melhorar seus níveis de serviço ao mercado. Nestas empresas, a coordenação das atividades de manufatura e logística é crítica. A integração e sincronização destas atividades é a alternativa que estas empresas necessitam para melhorar seus desempenhos operacionais e, conseqüentemente, aumentar o nível de serviço ao cliente. Esta Tese contribui para esta abordagem, propondo e avaliando novos métodos heurísticos para a resolução do problema de programação de operações de produção e distribuição em sistemas *no-wait flow shop* integrados a sistemas de transporte que operam estratégias distintas de distribuição. Os métodos propostos (ILS_{SA-VNS} , IG_{SA}^1 , IG_{SA}^2) implementam algoritmos baseados nas metaheurísticas IG, ILS, SA e VNS que programam de forma conjunta as operações de produção e transporte. O mínimo tempo total de fluxo é o principal critério de otimização adotado. O problema aqui estudado foi motivado por um caso real da indústria de produtos eletrônicos. A análise dos resultados experimentais demonstrou que os métodos propostos têm potencial para resolver o problema. Especificamente, o método IG_{SA}^2 obteve os melhores resultados, melhorando a qualidade da solução inicial em até 16% para instâncias menores e 11% para instâncias maiores. O estudo também verificou uma hipótese proposta para melhorar a qualidade das soluções geradas pelos métodos heurísticos que mostrou ser verdadeira para a maior parte dos experimentos.

Palavras-Chave: Programação *Flow Shop*. Roteamento de Veículos. Sistemas Integrados. Otimização Combinatória. Metaheurísticas.

ABSTRACT

PEREIRA, M. T. **Hybrid Metaheuristics for Integrated and Detailed Scheduling of Production and Distribution Operations in No-Wait Flow Shop Systems**, 2021. Tese (Doutorado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2021.

In today's complex business environments, the optimization of operational processes is an essential factor for the survival of organizations. Many companies struggling to keep competitiveness, search for better operational methods and models to minimize costs and improve market service level. In those companies, the coordination of manufacturing and logistics activities is critical. The integration and synchronization of these activities is the alternative that those companies need to improve their operational performance and, consequently, increase customer service level. This Thesis contributes to that approach, proposing and evaluating new heuristic methods to solve the problem of scheduling production and distribution operations in no-wait flow shop systems integrated to transport systems which operate different distribution strategies. The proposed methods (ILS_{SA-VNS} , IG_{SA}^1 , IG_{SA}^2) implement algorithms based on IG, ILS, SA and VNS metaheuristics which schedule manufacturing and transportation tasks jointly. Minimum total flow time is the main optimization criterion. The problem studied herein is based on a real case from electronic goods industry. The analysis of experimental results demonstrates the potential of the methods for solving the approached problem. Specifically, the IG_{SA}^2 method obtained the best results, improving the quality of the initial solution by up to 16% for small instances and 11% for larger instances. The study also verified a hypothesis proposed to improve the quality of the solutions generated by the heuristic methods which has proved to be true for most of the experiments.

Keywords: Flow Shop Scheduling. Vehicle Routing. Integrated Systems.
Combinatorial Optimization. Metaheuristics.

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Linhas de produção do caso real	8
Figura 2.2 – Sistema integrado produção-distribuição do caso real.....	10
Figura 2.3 – Programa de produção para o sistema <i>no-wait flow shop</i>	14
Figura 2.4 – Roteamento e programação de veículos.....	16
Figura 2.5 – Sistema integrado de produção e distribuição	20
Figura 2.6 – Programação NWFS no modelo DFC	24
Figura 2.7 – Sequências de produção e entrega.....	25
Figura 2.8 – Programação NWFS no modelo DVC	28
Figura 3.1 - Algoritmo para a metaheurística SA.....	32
Figura 3.2 - Algoritmo para a metaheurística VNS	35
Figura 3.3 - Algoritmo para a metaheurística ILS	37
Figura 3.4 - Algoritmo para a metaheurística IG.....	39
Figura 4.1 – Estruturas gerais dos métodos heurísticos propostos.....	62
Figura 4.2 – Procedimento de inserção de ordens.....	72
Figura 4.3 – Procedimentos de geração de vizinhança.....	77

LISTA DE TABELAS

Tabela 4.1 – Tempos de processamento	80
Tabela 4.2 – Distâncias	81
Tabela 4.3 – Atributos da ordem	81
Tabela 4.4 – Programa de produção (LPT)	82
Tabela 4.5 – Programa de produção (NEH)	83
Tabela 4.6 – Programa de produção final (IG _{SA})	83
Tabela 4.7 – Programa de entrega antes da perturbação de rota (LPT)	84
Tabela 4.8 – Programa de entrega antes da perturbação de rota (NEH)	84
Tabela 4.9 – Programa de entrega antes da perturbação de rota (IG _{SA})	85
Tabela 4.10 – Programa de entrega após a perturbação de rota (IG _{SA})	86
Tabela 4.11 – Resumo dos tempos de fluxo totais (TFT)	86
Tabela 4.12 – Programa de produção usando C_{max}	87
Tabela 4.13 – Programa de entrega usando C_{max}	88
Tabela 4.14 – Programa de entrega após a perturbação de rota (IG _{SA})	88
Tabela 5.1 – Níveis do fator 1	92
Tabela 5.2 – Fatores e níveis experimentais	93
Tabela 5.3 – Plano experimental	94
Tabela 5.4 – Tamanhos dos subconjuntos de inserção	96
Tabela 6.1 – DRP1 e percentual de melhoria usando NEH (DFC2)	108
Tabela 6.2 – Estatísticas complementares usando NEH (DFC2)	108
Tabela 6.3 – DRP1 e percentual de melhoria usando XNEH (DFC2)	109
Tabela 6.4 – Estatísticas complementares usando XNEH (DFC2)	109
Tabela 6.5 – DRP1 e percentual de melhoria usando CHINS (DFC2)	110
Tabela 6.6 – Estatísticas complementares usando CHINS (DFC2)	110
Tabela 6.7 – DRP1 e percentual de melhoria usando XCHINS (DFC2)	111
Tabela 6.8 – Estatísticas complementares usando XCHINS (DFC2)	112
Tabela 6.9 – DRP1 e percentual de melhoria usando HC1 (DFC2)	113

Tabela 6.10 – Estatísticas complementares usando HC1 (DFC2)	114
Tabela 6.11 – DRP1 e percentual de melhoria usando HC2 (DFC2)	114
Tabela 6.12 – Estatísticas complementares usando HC2 (DFC2)	115
Tabela 6.13 – Resumo NRV usando DRP1 (DFC2).....	116
Tabela 6.13B – Resumo NRV por tamanho de instância (DFC2)	116
Tabela 6.14 – DRP2 e DRP3 para as três metaheurísticas (DFC2).....	117
Tabela 6.15 – Resultados ANOVA (DFC2)	118
Tabela 6.16 – DRP1 e percentual de melhoria usando NEH (DVC1)	119
Tabela 6.17 – Estatísticas complementares usando NEH (DVC1).....	119
Tabela 6.18 – DRP1 e percentual de melhoria usando XNEH (DVC1)	120
Tabela 6.19 – Estatísticas complementares usando XNEH (DVC1)	120
Tabela 6.20 – DRP1 e percentual de melhoria usando CHINS (DVC1).....	121
Tabela 6.21 – Estatísticas complementares usando CHINS (DVC1)	122
Tabela 6.22 – DRP1 e percentual de melhoria usando XCHINS (DVC1)	122
Tabela 6.23 – Estatísticas complementares usando XCHINS (DVC1).....	123
Tabela 6.24 – DRP1 e percentual de melhoria usando HC1 (DVC1).....	124
Tabela 6.25 – Estatísticas complementares usando HC1 (DVC1)	124
Tabela 6.26 – DRP1 e percentual de melhoria usando HC2 (DVC1).....	125
Tabela 6.27 – Estatísticas complementares usando HC2 (DVC1)	125
Tabela 6.28 – Resumo NRV usando DRP1 (DVC1)	126
Tabela 6.28B – Resumo NRV por tamanho de instância (DVC1)	126
Tabela 6.29 – DRP2 e DRP3 para as três metaheurísticas (DVC1)	127
Tabela 6.30 – Resultados ANOVA (DVC1)	128
Tabela 6.31 – Resultados onde DVC1 foi melhor que DVC2.....	129
Tabela A.1 – DRP1 e percentual de melhoria usando NEH (DVC2)	153
Tabela A.2 – Estatísticas complementares usando NEH (DVC2)	153
Tabela A.3 – DRP1 e percentual de melhoria usando XNEH (DVC2).....	154
Tabela A.4 – Estatísticas complementares usando XNEH (DVC2).....	154
Tabela A.5 – DRP1 e percentual de melhoria usando CHINS (DVC2)	155
Tabela A.6 – Estatísticas complementares usando CHINS (DVC2).....	155
Tabela A.7 – DRP1 e percentual de melhoria usando XCHINS (DVC2)	156
Tabela A.8 – Estatísticas complementares usando XCHINS (DVC2)	156
Tabela A.9 – DRP1 e percentual de melhoria usando HC1 (DVC2)	157
Tabela A.10 – Estatísticas complementares usando HC1 (DVC2).....	157

Tabela A.11 – DRP1 e percentual de melhoria usando HC2 (DVC2)	158
Tabela A.12 – Estatísticas complementares usando HC2 (DVC2).....	158
Tabela A.13 – Resumo NRV usando DRP1 (DVC2)	159
Tabela A.13B – Resumo NRV por tamanho de instância (DVC2).....	159
Tabela A.14 – DRP2 e DRP3 para as três metaheurísticas (DVC2)	160
Tabela A.15 – Resultados ANOVA (DVC2).....	161
Tabela A.16 – Variações percentuais entre as FO de DVC1 e DVC2.....	163
Tabela B.1 – Resultado teste Tukey instâncias N50-M5 (DFC2)	166
Tabela B.2 – Resultado teste Fisher instâncias N50-M5 (DFC2).....	167
Tabela B.3 – Resultado teste Tukey instâncias N50-M10 (DFC2)	168
Tabela B.4 – Resultado teste Fisher instâncias N50-M10 (DFC2).....	169
Tabela B.5 – Resultado teste Tukey instâncias N50-M20 (DFC2)	170
Tabela B.6 – Resultado teste Fisher instâncias N50-M20 (DFC2).....	171
Tabela B.7 – Resultado teste Tukey instâncias N100-M5 (DFC2)	172
Tabela B.8 – Resultado teste Fisher instâncias N100-M5 (DFC2)	173
Tabela B.9 – Resultado teste Tukey instâncias N100-M10 (DFC2)	174
Tabela B.10 – Resultado teste Fisher instâncias N100-M10 (DFC2)	175
Tabela B.11 – Resultado teste Tukey instâncias N100-M20 (DFC2)	176
Tabela B.12 – Resultado teste Fisher instâncias N100-M20 (DFC2)	177
Tabela B.13 – Resultado teste Tukey instâncias N200-M10 (DFC2)	178
Tabela B.14 – Resultado teste Fisher instâncias N200-M10 (DFC2)	179
Tabela B.15 – Resultado teste Tukey instâncias N200-M20 (DFC2)	180
Tabela B.16 – Resultado teste Fisher instâncias N200-M20 (DFC2)	181
Tabela B.17 – Resultado teste Tukey instâncias N100-M20 (DVC1)	182
Tabela B.18 – Resultado teste Fisher instâncias N100-M20 (DVC1).....	183
Tabela B.19 – Resultado teste Tukey instâncias N200-M10 (DVC1)	184
Tabela B.20 – Resultado teste Fisher instâncias N200-M10 (DVC1).....	185
Tabela B.21 – Resultado teste Tukey instâncias N200-M20 (DVC1)	186
Tabela B.22 – Resultado teste Fisher instâncias N200-M20 (DVC1).....	187
Tabela B.23 – Resultado teste Tukey instâncias N20-M20 (DVC2).....	188
Tabela B.24 – Resultado teste Fisher instâncias N20-M20 (DVC2).....	189
Tabela B.25 – Resultado teste Tukey instâncias N50-M5 (DVC2)	190
Tabela B.26 – Resultado teste Fisher instâncias N50-M5 (DVC2).....	191
Tabela B.27 – Resultado teste Tukey instâncias N50-M10 (DVC2).....	192

Tabela B.28 – Resultado teste Fisher instâncias N50-M10 (DVC2).....	193
Tabela B.29 – Resultado teste Tukey instâncias N50-M20 (DVC2).....	194
Tabela B.30 – Resultado teste Fisher instâncias N50-M20 (DVC2).....	195
Tabela B.31 – Resultado teste Tukey instâncias N100-M5 (DVC2).....	196
Tabela B.32 – Resultado teste Fisher instâncias N100-M5 (DVC2).....	197
Tabela B.33 – Resultado teste Tukey instâncias N100-M10 (DVC2).....	198
Tabela B.34 – Resultado teste Fisher instâncias N100-M10 (DVC2).....	199
Tabela B.35 – Resultado teste Tukey instâncias N100-M20 (DVC2).....	200
Tabela B.36 – Resultado teste Fisher instâncias N100-M20 (DVC2).....	201
Tabela B.37 – Resultado teste Tukey instâncias N200-M10 (DVC2).....	202
Tabela B.38 – Resultado teste Fisher instâncias N200-M10 (DVC2).....	203
Tabela B.39 – Resultado teste Tukey instâncias N200-M20 (DVC2).....	204
Tabela B.40 – Resultado teste Fisher instâncias N200-M20 (DVC2).....	205
Tabela C.1 – Tempos de processamento	207
Tabela C.2 – Atributos da ordem.....	208
Tabela C.3 – Tempos de transporte (DFC2)	209
Tabela C.4 – Programa de produção final (DFC2)	210
Tabela C.5 – Programa de entrega antes da perturbação de rota (DFC2)	211
Tabela C.6 – Programa de entrega depois da perturbação de rota (DFC2).....	212
Tabela C.7 – Tempos de transporte (DVC1)	213
Tabela C.8 – Programa de produção final (DVC1).....	214
Tabela C.9 – Programa de entrega antes da perturbação de rota (DVC1)	215
Tabela C.10 – Programa de entrega depois da perturbação de rota (DVC1)	216
Tabela C.11 – Tempos de transporte (DVC2)	217
Tabela C.12 – Programa de produção final (DVC2)	218
Tabela C.13 – Programa de entrega antes da perturbação de rota (DVC2)	219
Tabela C.14 – Programa de entrega depois da perturbação de rota (DVC2)	220

LISTA DE ABREVIATURAS E SIGLAS

3PL	Serviço de transporte terceirizado
AAT	Tempo médio de chegada de veículos
AIA	Algoritmo imunológico artificial
CD	Centro de distribuição
CHINS	Algoritmo de ‘inserção mais barata’
CPU	Unidade de processamento central
DE	Algoritmo evolutivo diferencial
DFC	Data fixa de embarque, partida ou coleta
DRP	Desvio relativo percentual
DVC	Data variável de embarque, partida ou coleta
ERP	Sistemas de planejamento de recursos empresariais
FO	Função objetivo
FS	<i>Flow shop</i>
GA	Algoritmo genético
GVNS	Busca em vizinhança variável geral
HC1	Heurística construtiva 1
HC2	Heurística construtiva 2
ICA	Algoritmo competitivo imperialista
IDE	Ambiente integrado de desenvolvimento
IG	Algoritmo iterativo guloso
IG1, IG_{SA}^1	Variante 1 do algoritmo IG
IG2, IG_{SA}^2	Variante 2 do algoritmo IG
ILS	Busca local iterativa
JIT	Just in Time
LFE	Efeito de aprendizado e esquecimento
LPT	Tempo mais longo de processamento
LSD	Diferença mínima significativa

MCT	Tempo médio de conclusão das ordens
MOSA	Recozimento simulado multiobjectivo
MTO	Produção por encomenda
MTS	Produção para estoque
NEH	Heurística Nawaz-Enscore-Ham
NRV	Número de resultados vencedores
NWFS	<i>No-wait flow shop</i>
OBL	Aprendizagem baseada em oposição
PLIM	Programação linear inteira mista
PSO	Otimização por enxame de partículas
PPFS	Problema de programação <i>flow shop</i>
PPNWFS	Problema de programação <i>no-wait flow shop</i>
PPIPD	Problema de programação integrada da produção e distribuição
PPIPD-NWFS	PPIPD em ambientes <i>no-wait flow shop</i>
PRV	Problema de roteamento de veículos
PRVC	Problema de roteamento de veículos com capacidade
NSSG	Geração de estruturas de busca em vizinhança
SA	Recozimento simulado
SIPD	Sistema integrado de produção e distribuição
SPT	Tempo mais curto de processamento
TCT	Tempo total de conclusão das ordens
TS	Busca tabu
TT	Atraso total das ordens
TFT	Tempo total de fluxo
VND	Busca descendente em vizinhança variável
VNS	Busca em vizinhança variável
XCHINS	Variante 'X' da heurística CHINS
XNEH	Variante 'X' da heurística NEH

LISTA DE SÍMBOLOS

α	Taxa de resfriamento
C_{max}	Data máxima de conclusão das ordens
C_j	Data de conclusão da ordem
$C(.)$	Critério de otimização
d_n	Conjunto de ordens extraídas e não sequenciadas
$D_{j-1,j}$	Distância mínima entre ordens consecutivas
DT_j	Data de entrega
δ	Fator de filtragem
F_j	Tempo de fluxo da ordem
$F(.), FO$	Função objetivo
$f(.)$	Função matemática
f_n, f	Fator de ajuste
φ	Fábrica
I, I_t	Intervalo entre coletas ou saídas de veículos
i	Índice da máquina ou da iteração
j	Índice da ordem
J_j	Ordem de produção
J_{π_h}	Ordem de produção sequenciada
k	Índice do veículo
l	Índice do cliente
L, L_D	Lista de ordens não sequenciadas
L_{SA}	Número de iterações até a mudança de temperatura
M_i	Máquina
M, m	Conjunto e número de máquinas
N, n	Conjunto e número de ordens
o_{ji}	Operação

π_R, π_{D_i}	Sequência parcial de ordens
π, π^*, π'	Sequência completa de ordens
p_{ji}	Tempo de processamento
$FP_t, FP_{\tau k}$	Data fixa de coleta
$P(.)$	Probabilidade
PS_i	Programa parcial
Q, Q_k	Capacidade veicular
q_j	Tamanho da ordem
s_j	Tempo de serviço
S_j	Tempo total (acumulado) de serviço
SO_i	Ordens programáveis
ST_j	Data de embarque
$t_{l,l'}, t_{ij}$	Tempo de transporte
T_j, T_{jk}	Tempo total (acumulado) de transporte
t, t_{SA}	Tempo de execução do algoritmo SA
T_0	Temperatura inicial
$T(.)$	Temperatura de recozimento
VP_j, VP_k	Data variável de coleta
W_j	Tempo de espera para embarque

SUMÁRIO

1. Introdução.....	1
2. Definição do Problema.....	7
2.1. O caso real na indústria de produtos eletrônicos.....	7
2.2. Programação <i>no-wait flow shop</i>	12
2.3. Roteamento de veículos	15
2.4. A programação integrada da produção e distribuição	17
2.5. O PPIPD-NWFS	21
2.5.1. O sistema de produção <i>no-wait flow shop</i>	21
2.5.2. O sistema de distribuição	22
2.5.2.1. Data fixa de coleta (DFC)	22
2.5.2.2. Data variável de coleta (DVC).....	26
3. Revisão Bibliográfica	29
3.1. Complexidade computacional.....	29
3.2. PPNWFS não integrado ao sistema de transporte	31
3.2.1. Recozimento simulado.....	31
3.2.2. Busca em vizinhança variável.....	34
3.2.3. Busca local iterativa	36
3.2.4. Busca iterativa gulosa	38
3.2.5. Heurísticas construtivas	42
3.3. Sistemas integrados de produção e distribuição (SIPD).....	43
3.3.1. Data fixa de embarque ou coleta (DFC).....	43
3.3.2. Data variável de embarque ou coleta (DVC).....	46
4. Métodos de Solução Propostos	51
4.1. Modelagem matemática	51
4.2. A hipótese $\min \{ N \times C_{max} + \sum_{j \in N} T_j \} \leq \min \sum_{j \in N} (C_j + T_j)$	58
4.3. Estrutura algorítmica geral.....	59
4.4. Pré-processamento	64

4.5. Fase 1 – Iniciação	64
4.5.1. Regras de prioridade.....	64
4.5.2. Heurística NEH	65
4.5.3. Heurística XNEH	66
4.5.4. Heurística CHINS.....	67
4.5.5. Heurística XCHINS	68
4.5.6. Heurística HC1	69
4.5.7. Heurística HC2.....	70
4.6. Fase 2 – Busca local e destruição	71
4.6.1. Busca local.....	71
4.6.2. Destruição	72
4.7. Fase 3 – Perturbação e construção.....	73
4.7.1. Perturbação	73
4.7.2. Construção.....	75
4.8. Fase 4 – Busca local e aceitação	76
4.8.1. Busca Local	76
4.8.2. Aceitação	78
4.9. Fase 5 – Perturbação da rota	79
4.10. Um exemplo de aplicação.....	79
4.11. Verificação da hipótese	86
5. Delineamento dos Experimentos Computacionais	90
5.1. Plano experimental	91
5.2. Parâmetros de execução	94
5.2.1. Metaheurística IG _{SA}	95
5.2.2. Metaheurística ILS _{SA-VNS}	98
5.3. Dados complementares das instâncias	99
5.4. Codificação e execução dos algoritmos computacionais.....	101
6. Análise Estatística dos Resultados	102
6.1. Considerações iniciais.....	102
6.1.1. Desvio relativo percentual.....	103
6.1.2. ANOVA e testes de Tukey e Fisher	106
6.2. Análise dos resultados experimentais para o modelo DFC2	107
6.2.1. Análise DRP1 usando a heurística NEH na iniciação	107

6.2.2. Análise DRP1 usando a heurística XNEH na iniciação.....	108
6.2.3. Análise DRP1 usando a heurística CHINS na iniciação.....	109
6.2.4. Análise DRP1 usando a heurística XCHINS na iniciação	111
6.2.5. Análise DRP1 usando a heurística HC1 na iniciação.....	112
6.2.6. Análise DRP1 usando a heurística HC2 na iniciação.....	114
6.2.7. Resumo da análise DRP1	115
6.2.8. Análise DRP independente da heurística de iniciação.....	116
6.2.9. Análise de variância (ANOVA)	117
6.3. Análise dos resultados experimentais para o modelo DVC1	118
6.3.1. Análise DRP1 usando a heurística NEH na iniciação	118
6.3.2. Análise DRP1 usando a heurística XNEH na iniciação.....	120
6.3.3. Análise DRP1 usando a heurística CHINS na iniciação.....	121
6.3.4. Análise DRP1 usando a heurística XCHINS na iniciação	122
6.3.5. Análise DRP1 usando a heurística HC1 na iniciação.....	123
6.3.6. Análise DRP1 usando a heurística HC2 na iniciação.....	124
6.3.7. Resumo da análise DRP1	125
6.3.8. Análise DRP independente da heurística de iniciação.....	127
6.3.9. Análise de variância (ANOVA)	128
6.3.10. Verificação da hipótese – DVC1 versus DVC2	129
6.4. Conclusões da análise estatística	130
7. Conclusões e Proposições.....	133
Referências Bibliográficas	136
Apêndice A – Análise dos Resultados Experimentais (DVC2)	153
Apêndice B – Resultados das Análises de Tukey e Fisher	164
Apêndice C – Exemplos de Resultados de Execução	206
Apêndice D – Modelo Matemático em GAMS e <i>Gurobi</i> py	221

1. Introdução

Nos complexos ambientes de negócio atuais, a maioria das empresas manufatureiras tem enfrentado grandes desafios pelo aumento da competição trazida pela globalização, encurtamento dos ciclos de vida dos produtos, pressões por redução de estoques e aumento do nível de serviço, mercados financeiros voláteis, mercados consumidores dinâmicos etc. Para se manterem competitivas e obterem ganhos superiores de desempenho na gestão na sua cadeia de suprimentos, elas têm buscado por novos e melhores métodos, modelos operacionais e tecnologias que possibilitem minimizar seus custos logísticos e aumentar seus níveis de serviço (SCHOLZ-REITER; FRAZZON; MAKUSCHEWITZ, 2010; REIMANN; NETO; BOGENDORFER, 2014; ZOU *et al.*, 2018).

As operações de produção e logística de distribuição são duas funções de negócio fundamentais da cadeia de suprimentos. A gestão eficiente dos processos logísticos e de produção requer a programação detalhada das atividades inerentes a cada área em conjunto. Isto significa alocar eficientemente os recursos produtivos (máquinas, equipamentos, ferramentas e operadores) e de transporte (veículos e tripulação) às ordens e operações de modo a maximizar a utilização dos recursos e minimizar os custos totais (EHM; FREITAG, 2016).

Tradicionalmente dentro das empresas, as decisões de programação de produção e de transporte são realizadas de forma sequencial, com pouca ou nenhuma integração entre as áreas funcionais de manufatura e logística, com resultados claramente subótimos (CHEN, 2010; EHM; FREITAG, 2016).

Muitos autores têm salientado que integrar a programação da produção e distribuição é a abordagem que muitas empresas necessitam para melhorar seus

desempenhos operacionais, reduzir custos e simultaneamente atender aos níveis de serviço contratados (CHEN; VAIRAKTARAKIS, 2005; PUNDOOR; CHEN, 2005; CHEN, 2010; SCHOLZ-REITER; FRAZZON; MAKUSCHEWITZ, 2010; EHM; FREITAG, 2016).

Scholz-Reiter, Frazzon e Makuschewitz (2010) ponderam que as novas demandas e necessidades para os sistemas de produção e logística trazidos pelo processo de globalização têm levado os participantes das cadeias de suprimentos a buscarem por reduções contínuas de custos e lead times para continuarem competitivos. Para alcançarem este fim, os processos de produção e logística (cruciais para a gestão de operações) deveriam ser sincronizados e coordenados conjuntamente dentro da empresa e desta com outros níveis da cadeia de suprimentos. Estes pesquisadores defendem que os processos de programação da produção e de transportes, por serem bastante relacionados, deveriam ser tratados como um único problema de otimização, destacando ainda a importância do intercâmbio robusto de informações entre estas duas funções de negócio.

Ehm e Freitag (2016) destacam que para melhorar o processo de tomada de decisão em um ambiente global bastante dinâmico e competitivo, a utilização dos recursos logísticos deveria ser mais bem considerada no âmbito do planejamento e controle da produção, destacando a programação detalhada e integrada da produção e transporte como uma das frentes de investigação a ser desenvolvida pela comunidade acadêmica.

No contexto dos sistemas integrados de produção e distribuição (SIPD), mais especificamente na otimização do planejamento e programação de operações, diferentes configurações e modelos tem sido estudados para diferentes objetivos, níveis de decisão e critérios de otimização (CHEN, 2010; FAHIMNIA *et al.*, 2013; ADULYASAK; CORDEAU; JANS, 2015; DIAS-MADROÑERO; PEIDRO; MULA, 2015; LI; CHEN; TANG, 2017; MOONS *et al.*, 2017; KUMAR *et al.*, 2020).

Conjuntamente com trabalhos teóricos, muitos artigos acadêmicos têm destacado a relevância prática do problema de otimização da programação em SIPD, dentre os quais podem ser citados: Chen e Vairaktarakis (2005), Garcia e Lozano (2005), Li,

Ganesan e Sivakumar (2005), Chen e Pundoor (2006), Li, Sivakumar e Ganesan (2008), Chen, Hsueh e Chang (2009) Liu, Zhang e Li (2014) e Ma, Wang e Bai (2019).

Chen (2010) argumenta que os modelos de programação detalhada e integrada buscam otimizar conjuntamente a produção e a entrega, levando em consideração as receitas, os custos e os níveis de serviço pedido a pedido (ou ordem a ordem) individualmente. Tais modelos são adequados (ou necessários) a empresas que produzem sob encomenda. Neste modelo de negócio, os produtos são fabricados para atenderem exclusivamente aos pedidos, em muitos casos, em prazos reduzidos. Consequentemente, há pouco ou nenhum estoque de produto acabado, as atividades de manufatura e distribuição são intimamente ligadas e devem ser programadas em conjunto para atenderem os prazos predefinidos a um custo total mínimo.

Não obstante o grande número de trabalhos publicados nos últimos 40 anos estudando problemas de otimização de planejamento e programação de operações em SIPD, a pesquisa acadêmica tem focado sobremaneira na resolução de problemas de planejamento integrado de operações, seja no nível estratégico, tático ou operacional. Relativamente pouca pesquisa tem sido realizada para o nível da programação detalhada e integrada de operações (CHEN, 2010; MOONS *et al.*, 2017; KUMAR *et al.*, 2020).

Mesmo para o nível de programação detalhada de SIPD, poucos são os estudos que solucionam o problema para sistemas de produção complexos (ex.: *flow shop* e *job shop*) simultaneamente ao roteamento de veículos para diferentes configurações do sistema de transporte. O que se concluiu da revisão bibliográfica foi que a produção acadêmica para este tema específico é recente (SCHOLZ-REITER *et al.*, 2011; MARANDI; ZEGORDI, 2017; RAMEZANIAN; MOHAMMADI; CHERAGHALIKHANI, 2017; MOHAMMADI; CHERAGHALIKHANI; RAMEZANIAN, 2018; HAN *et al.*, 2019; MARANDI; GHOMI, 2019).

O tema central desta Tese é o **problema de programação detalhada e integrada de operações de produção e distribuição em sistemas *no-wait flow shop* (PPIPD-NWFS)** que operam estratégias distintas de distribuição.

No PPIPD-NWFS, os recursos de produção e transporte devem ser programados simultaneamente, respeitando restrições relacionadas a ordem (ex.: data de liberação, data máxima de entrega etc.) e aos sistemas de produção (ex.: layout, número e disponibilidade de máquinas etc.) e de transporte (ex.: tamanho de frota, capacidade veicular, frequência de coleta etc.) (CHEN, 2010; EHM; FREITAG, 2016).

Este tema foi motivado principalmente pelo caso real de uma empresa multinacional que produz componentes e produtos eletrônicos para clientes de diferentes setores econômicos. A empresa opera com diferentes layouts de produção, dentre os quais destacam-se linhas e células de manufatura nas quais o PPIPD-NWFS ocorre.

O PPIPD-NWFS é a junção de dois problemas clássicos da pesquisa operacional: o problema de programação *no-wait flow shop* (PPNWFS) e o problema de roteamento de veículos (PRV) (CHEN, 2010).

O PPNWFS e o PRV são problemas de otimização combinatória cujos casos gerais são muito difíceis de serem solucionados até a otimalidade, representando um grande desafio de pesquisa (GAREY; JOHNSON; SETHI, 1976; LENSTRA; RINNOOY-KHAN, 1981; RÖCK, 1984; SRISKANDARAJAH; LADET, 1986).

Tal desafio estimulou a comunidade acadêmica a gerar uma grande quantidade de pesquisa nos últimos 60 anos sobre modelos e métodos matemáticos e computacionais para solucionar ambos os problemas isoladamente (GUPTA; STAFFORD JR., 2006; ALLAHVERDI *et al.*, 2008; LAPORTE, 2009; RIBAS; LEISTEN; FRAMINAN, 2010; PAN; RUIZ, 2013; YENISEY; YAGMAHAN, 2014; NAGANO; MIYATA, 2016b; ALLAHVERDI, 2016; FERNANDEZ-VIAGAS; RUIZ; FRAMINAN, 2017; MONTOYA-TORRES *et al.*, 2015).

Apesar da grande quantidade de literatura criada para o PPNWFS e o PRV, na revisão bibliográfica, não foram encontrados estudos para o PPIPD-NWFS.

A relevância prática do PIPD-NWFS, sua natureza combinatória complexa e a falta de estudos acadêmicos publicados para ele motivaram o desenvolvimento da pesquisa.

O objetivo desta Tese é propor, desenvolver e avaliar novos métodos heurísticos de programação de operações de produção e distribuição em sistemas NWFS integrados a sistemas de transporte que operam de acordo com duas estratégias logísticas distintas, denominadas 'data fixa de coleta' e 'data variável de coleta'.

Fundamentados na pressuposição de que a melhoria na coordenação e sincronização das operações de produção e distribuição leva à minimização dos tempos de entrega e atrasos, e conseqüentemente ao aumento do nível de serviço, os métodos heurísticos propostos implementam algoritmos de programação híbridos baseados nas metaheurísticas IG, ILS, SA e VNS.

O principal critério de otimização usado pelos métodos heurísticos propostos é a minimização do tempo total de fluxo (*total flow time* - TFT) definido como o somatório dos tempos de fluxo de todas as ordens produzidas e entregues aos clientes.

Os métodos heurísticos foram codificados na linguagem C++ e seus desempenhos foram avaliados por meio de experimentos computacionais e análises estatísticas. Os resultados experimentais foram analisados extensamente por meio da medida DRP (desvio relativo percentual) e pelos testes estatísticos ANOVA, Tukey e Fisher (LSD).

Dentre as contribuições desta Tese para a pesquisa acadêmica destacam-se:

- 1) O estudo do problema de programação detalhada de operações em sistemas de produção e transporte integrados, para os quais a pesquisa acadêmica ainda é recente.
- 2) A proposição e resolução de um modelo em programação matemática para o problema.
- 3) A proposição de novos métodos heurísticos de solução com potencial aplicação prática e que implementam procedimentos de busca também propostos no estudo.

- 4) A proposição e análise de uma hipótese formulada para melhorar a qualidade das soluções geradas pelos métodos heurísticos que se mostrou verdadeira para a maior parte dos casos simulados.

Esta Tese está estruturada da seguinte maneira: no Capítulo 2, o caso real é apresentado e o problema da pesquisa é definido. No Capítulo 3, é apresentada a revisão bibliográfica relevante para a condução do estudo. No Capítulo 4, é desenvolvido um modelo matemático para o problema e são descritos os métodos de solução propostos. Nos Capítulos 5, 6 e 7 são descritos, respectivamente, o planejamento experimental e a parametrização dos métodos de solução para execução computacional, a análise estatística dos resultados e, por fim, as conclusões do estudo e proposições para pesquisa futura.

2. Definição do Problema

Conforme já mencionado anteriormente, o PPIPD-NWFS é composto por dois problemas de otimização: o problema de programação *no-wait flow shop* (PPNWFS) e o problema de roteamento de veículos (PRV).

Este capítulo inicia-se na Seção 2.1 com a descrição do caso real que motivou esta pesquisa. O PPNWFS e o PRV são descritos, em seguida, nas Seções 2.2 e 2.3 respectivamente. As Seções 2.4 e 2.5 são dedicadas ao detalhamento do PPIPD no ambiente NWFS.

2.1. O caso real na indústria de produtos eletrônicos

O problema estudado nesta Tese foi motivado principalmente por um caso real de um fabricante multinacional de produtos eletrônicos com sede nos EUA que atende aos mercados norte americano, latino-americano, europeu e asiático.

Com faturamento de aproximadamente treze bilhões de dólares anuais, este fabricante possui mais de 100 fábricas automatizadas e semiautomatizadas distribuídas em 17 países – principalmente nos EUA, Alemanha, Inglaterra e Romênia – que produzem um variado portfólio de produtos (cabos, terminais, filtros, conectores, sensores, sistemas óticos e elétricos, microprocessadores, componentes eletrônicos diversos, ferramentas manuais, dentre muitos outros produtos), atendendo a grandes fabricantes de diferentes setores (automotivo, equipamentos industriais, aeroespacial, defesa, telecomunicações, computadores, equipamentos médicos, petróleo e gás, bens de consumo e energia) e diretamente para diversas pequenas, médias e grandes redes varejistas e distribuidores em 150 países.

A fábrica do estudo está localizada nos EUA e opera um sistema de produção caracterizado por conjuntos de máquinas dispostas em linhas ou células dedicadas à produção de diversos itens com maior ou menor nível de automação dependendo da família de produtos, havendo linhas totalmente automatizadas (sem operadores) e linhas com baixo nível de automação (com mão de obra intensiva).

Esta fábrica em particular atende tanto a fabricantes (de diferentes setores econômicos) quanto aos mercados distribuidor e varejista de componentes dos EUA.

As linhas de produção foco desta Tese são semiautomatizadas e possuem vários (sub)processos executados por equipamentos isolados e/ou trabalhadores que realizam diferentes operações, tais como separação de materiais (scrap), furação, corte, roscamento, aquecimento, dobramento, soldagem, estampagem, polimento, montagens simples, inspeção e embalagem, trabalhando em diferentes turnos diários.

Cada linha ou célula de produção possui de seis a quinze estações de trabalho semi ou totalmente automatizadas. Na Figura 2.1, é apresentado um esquema das linhas de produção.

Figura 2.1 – Linhas de produção do caso real

Fonte: o autor

O sistema de produção atende diretamente a: a) contratos de venda anuais e programas de entrega semanais firmados com os fabricantes e b) pedidos de venda captados diariamente (parte deles via comércio eletrônico) de distribuidores e varejistas.

Algumas famílias de produtos são produzidas para estoque (*make to stock* – MTS) e outras somente por encomenda (*make to order* – MTO). Em geral, os contratos com os grandes fabricantes são atendidos pelo modelo MTO.

Independente do modelo de produção (MTS ou MTO), diferentes famílias de produtos (e respectivas ordens de produção) alternam-se nas linhas ou células de produção.

Como regra geral, cada ordem deve ser processada integralmente na linha ou célula de produção e sem interrupção. Isto porque além de haver pouco ou nenhum espaço para o acúmulo de materiais entre as estações, os diferentes produtos exigem ferramental próprio e materiais distintos.

Um dos determinantes do fluxo de produção em cada linha ou célula é o tamanho de lote de processamento que é definido pelo fracionamento das ordens em lotes menores de produção que são então processados integralmente em cada estação.

Uma vez concluído em uma estação, o lote é movido para a estação seguinte e assim sucessivamente até que toda a ordem seja processado na linha ou célula. A determinação dos tamanhos de lote de processamento segue a regras empíricas que dependem do *mix* de produtos da linha ou célula, do tamanho da ordem, dentre outros fatores.

Em cada estação de trabalho, os tempos ociosos são usados para limpeza, troca de gabaritos e ferramentas, preparação do material, dentre outras atividades.

Devido às características descritas anteriormente, o sistema de produção pode ser modelado como um *flow shop* contínuo ou *no-wait*. Na Figura 2.2, é apresentado um esquema do sistema integrado de produção e distribuição.

A fábrica não mantém frota própria. A distribuição dos produtos é feita utilizando dois modelos ou estratégias de transporte. Em ambos os modelos, o transporte é realizado por transportadoras terceirizadas (3PL).

Na primeira estratégia, os transportadores realizam coletas em intervalos regulares que variam de quatro a sete dias. Neste modelo, o serviço é contratado pelo cliente que estabelece a periodicidade das coletas.

Na segunda estratégia, a fábrica contrata diretamente o serviço de transportadores homologados. A fábrica faz uso desta estratégia quando: a) não consegue atender integralmente às programações de entrega (dos fabricantes) dentro dos intervalos de coleta, gerando um *backlog* de produção ou estoque pós coleta que deve ser executado (entregue) o mais rapidamente possível; b) os clientes (em geral varejistas) não contratam o serviço de transporte, deixando para a fábrica a responsabilidade pela contratação.

Figura 2.2 – Sistema integrado produção-distribuição do caso real

Fonte: o autor

Buscando por economias de escala, a fábrica sempre tenta atender o máximo de pedidos de varejistas conjuntamente com os pedidos de grandes clientes, procurando criar um *mix* de produção e entrega que deve ser atendido pelos dois modelos de distribuição.

Diariamente, os planejadores realizam diferentes simulações de programação de produção considerando diferentes *mixes* de produtos (ordens) e de forma separada os analistas de logística executam o planejamento do transporte. As decisões nestas duas áreas são claramente subotimizadas e sempre foi um objetivo da gerência de operações integrar as respectivas atividades.

A implementação de um ERP atendeu parcialmente a este requisito de negócio. A disciplina imposta pelo sistema ERP na execução das atividades de controle da manufatura e do transporte aumentou a integração entre as áreas com o aumento do compartilhamento de informações, mas as atividades de planejamento e programação em ambas as áreas continuaram sendo executadas de forma independente, em parte pelo fato do sistema informatizado não possuir nenhuma funcionalidade de planejamento integrado de produção e transporte, criando oportunidades para a proposição e implementação de melhorias para o planejamento e programação conjuntos.

Alguns dos requisitos de negócio (anseios) pendentes reportados pelos gestores de operações após a fase de amadurecimento do uso do ERP foram:

- 1) Efetivamente sincronizar os fluxos de produtos e informação entre as áreas de produção e logística.
- 2) Obter visibilidade total do *status* da produção e logística, antecipando as necessidades de recurso de manufatura (máquinas e mão de obra) e transporte (veículos).
- 3) Melhorar a acurácia das datas estimadas de entrega, principalmente para os clientes 'premium' (grandes fabricantes).
- 4) Reduzir os estoques de produtos aguardando o embarque.
- 5) Melhorar a sinergia entre as áreas de produção e logística na resolução de problemas de operações que afetem conjuntamente ambas as áreas, tais como as reprogramações de produção que ocasionam má consolidação de carga

(fracionamento), perda de janela de coleta e contração urgente (*spot*) de transporte.

- 6) Avaliar qual estratégia de transporte é melhor para uma determinada situação (veículo disponível, data de coleta mais próxima, *mix* de produção, status da produção).

Os métodos de programação propostos nesta Tese procuram apoiar em maior ou menor grau os planejadores no atendimento destes requisitos.

Vale mencionar que nenhum estudo foi encontrado na literatura acadêmica que descreva um problema real com características similares a este e, portanto, uma abordagem de solução similar à adotada nesta Tese.

2.2. Programação *no-wait flow shop*

Um dos ambientes de produção mais estudados em âmbito acadêmico, o *flow shop* clássico é um layout comum nos sistemas fabris onde um número fixo de máquinas continuamente disponíveis são dispostas em série. Cada ordem deve ser processada nas máquinas na sequência em que elas estão dispostas formando um fluxo unidirecional (GUPTA; STAFFORD JR., 2006).

Devido à complexidade e relevância prática dos sistemas *flow shop*, diversas configurações e restrições operacionais têm sido modeladas. Uma das possíveis restrições está relacionada a não espera entre as execuções consecutivas das operações de uma mesma ordem, o que resulta na não formação de estoques intermediários entre as máquinas. As restrições *no-wait* ocorrem devido principalmente: a) a necessidades tecnológicas de alguns ambientes de produção; b) a esforços para a redução do estoque em processo; c) a implementação da filosofia de ‘produção enxuta’ ou JIT (MACCHIAROLI; MOLÈ; RIEMMA, 1999).

Estudos seminais do problema de programação *no-wait flow shop* (PPNWFS) surgiram no início dos anos de 1970s (REDDI; RAMAMOORTHY, 1972; WISMER,

1972). Desde a sua proposição, o PPNWFS vem sendo muito estudado pela comunidade acadêmica. Os artigos de (HALL; SRISKANDARAJAH, 1996), (ALLAHVERDI, 2016) e (NAGANO; MIYATA, 2016b) apresentam revisões abrangentes sobre métodos de solução para PPNWFS.

PPNWFS ocorre em diversos tipos de indústria, tais como química, metalúrgica, alimentos, eletrônica e saúde (HALL; SRISKANDARAJAH, 1996; SAPKAL; LAHA, 2013; DONG *et al.* 2021). Formalmente, o PPNWFS clássico é definido a seguir (adaptado de LIN; YING, 2016).

Considere um conjunto $\vartheta = \{J_1, J_2, \dots, J_n\}$ de n ordens (*jobs*) a serem processadas em m máquinas do conjunto $\mathcal{M} = \{M_1, M, \dots, M_m\}$. Todas as ordens estão simultaneamente disponíveis para execução no início do período de programação e todas as máquinas estão continuamente disponíveis para processamento das ordens. Cada ordem J_j ($j = 1, 2, \dots, n$) compreende uma sequência estritamente ordenada de m operações $(o_{j1}, o_{j2}, \dots, o_{jm})$, onde o_{ji} é a operação da ordem J_j pré designada para a máquina i ($i = 1, 2, \dots, m$) e deve ser processada no tempo $p_{ji} \geq 0$.

É importante observar, sem perda de generalidade, que nesta Tese os termos ‘ordem’ (*work order*) e ‘trabalho’ (*job*) têm o mesmo significado e correspondem ao conjunto de operações que deve ser executado em uma sequência de máquinas, seguindo o roteiro de produção estabelecido para a ordem (trabalho).

Em qualquer instante, uma ordem só pode ser processada em uma única máquina e uma máquina pode executar apenas uma ordem por vez. Nenhuma fila é permitida em qualquer máquina, exceto na primeira máquina. Uma vez iniciado o processamento de uma ordem na primeira máquina, esta ordem deve ser processada continuamente em todas as máquinas até a sua conclusão. Para garantir que cada ordem seja processada sem interrupção, todas as ordens devem ser processadas na mesma sequência em todas as máquinas.

Para satisfazer a restrição *no-wait*, o início do processamento de uma ordem J_j na primeira máquina ($i = 1$) deve ser atrasado em uma porção de tempo $D_{j-1,j} =$

$f(J_{j-1}, J_j)$ estritamente suficiente para garantir que a ordem J_j seja processada sem ter que aguardar o final do processamento de uma ordem J_{j-1} iniciada anteriormente em qualquer uma das demais máquinas ($i \geq 2$). Para que isto aconteça, uma das formulações propostas define $D_{j-1,j} = \max_{1 \leq i \leq m} (\sum_{k=i}^m p_{j,k} - \sum_{k=i+1}^m p_{j-1,k})$, o que corresponde ao intervalo de tempo ('distância') mínimo(a) entre as datas de conclusão das duas ordens J_{j-1} e J_j (LI; WANG; WU, 2008; YE; LI; MIAO, 2016). O valor de $D_{j-1,j}$ é calculado para cada par de ordens de ϑ antes da programação. Esta característica permite formular o tempo de conclusão de cada ordem J_j na posição $h+1$ da sequência π como $C_{\pi[h+1]}^j = \sum_{i=0}^h D_{\pi[i], \pi[i+1]}$.

Uma sequência $\pi = (J_{\pi_1}, J_{\pi_2}, \dots, J_{\pi_h}, \dots, J_{\pi_n})$, onde π_h corresponde a posição de uma ordem J_j na sequência, com $h = (1, 2, \dots, n)$, é dita viável se e somente se atender a todas as restrições. A sequência $\pi^* = (J_{\pi_1^*}, J_{\pi_2^*}, \dots, J_{\pi_n^*})$ é dita ótima para a minimização do critério de otimização $C(\pi)$ se e somente se $C(\pi^*) < C(\pi), \forall \pi^* \neq \pi$.

Na Figura 2.3, é apresentado o gráfico de Gantt de um exemplo de programa de produção gerado para um ambiente *no-wait flow shop* onde cinco ordens devem ser processadas em cinco máquinas.

Figura 2.3 – Programa de produção para o sistema *no-wait flow shop*

Fonte: o autor

Na Figura 2.3, cada bloco contém o número da ordem e a máquina entre parênteses. O tamanho do bloco corresponde ao tempo de processamento em cada máquina. As ‘distâncias’ $D_{j-1,j}$ também são apresentadas.

2.3. Roteamento de veículos

Desde a sua proposição por Dantzig e Ramser (1959) e Clarke e Wright (1964) o problema de roteamento de veículos (PRV) e suas variantes tornaram-se uma das classes de problemas mais estudadas no âmbito da Pesquisa Operacional.

Devido a sua complexidade e relevância prática, diversas configurações e restrições operacionais tem sido modeladas (BRÄYSY; GENDREAU, 2005a, 2005b; PARRAGH; DOERNER; HARTL, 2008a, 2008b; LAPORTE, 2009; PILLAC *et al.*, 2013; CAMPBELL; WILSON, 2014; LIN *et al.*, 2014; MONTOYA-TORRES *et al.*, 2015).

O PRV clássico (também denominado PRVC – problema de roteamento de veículos com capacidade) é um problema de gestão da distribuição onde um conjunto de clientes (pontos de entrega) geograficamente dispersos deve ser atendido a partir de um único depósito por meio de uma frota homogênea de veículos de capacidade limitada (BRAEKERS; RAMAEKERS; NIEUWENHUYSE, 2016).

A solução deve prover um conjunto ótimo de rotas de entrega onde cada veículo trafegue apenas em uma rota. Cada rota representa a sequência de pontos de parada que cada veículo deve realizar. O objetivo é encontrar o conjunto de rotas de menor custo de maneira que cada cliente seja visitado exatamente uma vez, cada veículo inicie e termine a viagem no depósito e a capacidade do veículo não seja excedida (BRAEKERS; RAMAEKERS; NIEUWENHUYSE, 2016).

Na Figura 2.4, é apresentado um exemplo do PRV contendo a programação de transporte de três veículos e respectivas rotas de entrega.

Os três veículos ($V_k \in \{1...3\}$) devem atender a sete clientes ($C_l \in \{1...7\}$). As rotas de V_1 e V_3 atendem a dois clientes cada. A rota de V_2 atende a três clientes.

No gráfico de Gantt de cada veículo, são apresentados os tempos de transporte, serviço (ex.: carregamento, espera, manobra e descarregamento) e de retorno do veículo ao centro de distribuição (CD).

Para os tempos de transporte, a identificação do veículo e clientes de origem e destino é feita em cada bloco por $[V_k, (C_l, C_{l+1})]$ e para os tempos de serviço ela é feita por $[V_k, C_{l+1}]$.

Figura 2.4 – Roteamento e programação de veículos

Fonte: o autor

Esta Tese estuda o PRV no contexto do PPIPD. Na próxima seção o PRV é formalmente apresentado conjuntamente com o PPIPD.

2.4. A programação integrada da produção e distribuição

O PPIPD ocorre em empresas que produzem sob encomenda (MTO) ou com restrição de tempo de produção e entrega (ex.: produtos perecíveis, produtos com alta taxa de obsolescência tecnológica ou planejada, ordens *rush*, contratos de pronta entrega ou *just-in-time* etc.). Neste contexto, os produtos devem ser embarcados para os clientes imediatamente após serem produzidos ou dentro de um curto intervalo de tempo após a sua produção. Consequentemente, pouco ou nenhum estoque é mantido no sistema, a produção e distribuição são diretamente relacionadas e devem ser programadas conjuntamente para que seja atingido o desempenho contratual almejado a um mínimo custo logístico total e para se reduzir o risco de perda de valor ou de venda por deterioração ou obsolescência (PUNDOOR; CHEN, 2005; CHEN; PUNDOOR, 2006; STECKE; ZHAO, 2007; CHEN, 2010).

O PPIPD ocorre na cadeia de produção de muitos produtos, tais como alimentos em natura (ex.: hortifrutigranjeiros, carnes e flores), alimentos processados para consumo imediato (ex.: refeições e laticínios), produtos sanguíneos (ex.: sangue e derivados), vacinas, concreto pronto para uso (*ready-mix*) e produtos radioativos para uso hospitalar que deterioram em taxas controladas ao longo de sua produção e, logo após, em taxas aceleradas durante a armazenagem e transporte. Outros produtos, tais como jornais e impressos diários, vestuário de moda, componentes e aparelhos eletrônicos, dentre muitos outros, tornam-se obsoletos rapidamente assim que são produzidos. (GARCIA; LOZANO, 2005; STECKE; ZHAO, 2007; GEISMAR *et al.*, 2008; CHEN; HSUEH; CHANG, 2009; CHIANG *et al.*, 2009; LEE *et al.*, 2014; DEVAPRIYA; FERRELL; GEISMAR, 2017; MOONS *et al.*, 2017; NOROOZI *et al.*, 2019).

O caso geral do PPIPD (independente do sistema de produção) é definido a seguir (adaptado de CHEN, 2010).

Um fabricante, que possui uma ou mais fábricas ou sistemas de produção, simbolizados por φ , recebe um conjunto $N = \{1, 2, \dots, j, \dots, n\}$ de n ordens (*jobs*) provenientes de $C = \{1, 2, \dots, l, \dots, c\}$ clientes dispersos geograficamente. Sem perda de generalidade, considere uma ordem por cliente, então $c = n$. Primeiramente, o

fabricante deve processar cada ordem por uma ou mais máquinas do conjunto de máquinas $M_\varphi = \{1, 2, \dots, i, \dots, m_\varphi\}$ até a sua conclusão em C_j . O número e sequência de máquinas depende da configuração do sistema de produção em cada fábrica e os tempos de processamento $p_{ji} \geq 0$ são predefinidos. Logo após a produção, cada ordem deve ser enviada individualmente ou em lote para cada cliente l o mais rápido possível por meio de um conjunto disponível de veículos (frota) $V_\varphi = \{1, 2, \dots, k \dots v_\varphi\}$.

Além dos parâmetros já mencionados, cada ordem do conjunto N pode estar associada a um ou mais parâmetros descritos a seguir:

- 1) Data de liberação r_j a qual define o instante em que a ordem j chega ao sistema e está disponível para processamento.
- 2) Data fixa de entrega FD_j na qual obrigatoriamente a ordem j deve ser entregue para o cliente.
- 3) Data fixa de embarque (ou partida) FP_j na qual obrigatoriamente a ordem j deve ser embarcada para o cliente.
- 4) Data de entrega d_j a qual corresponde à data solicitada pelo cliente para o recebimento da ordem j .
- 5) Data limite (deadline) d_j^u até a qual a ordem j deve ser entregue para o cliente.
- 6) Janela de entrega $[a_l, b_l]$ dentro da qual a ordem deve ser entregue ao cliente l , podendo ser violada com penalidade.
- 7) Tamanho q_j o qual representa o tamanho da ordem j para produção ou transporte.
- 8) Importância relativa w_j de cada ordem j .
- 9) A receita R_j (ou custo c_j) realizada (ou incorrido) se a ordem j for processada e entregue para o cliente.

Em geral, as seguintes configurações do sistema de transporte são consideradas:

- 1) Todos os veículos do conjunto V_φ estão disponíveis no início do horizonte de programação.
- 2) Ao final das entregas, os veículos podem ou não retornar à fábrica na qual estavam alocados inicialmente.

- 3) Os veículos podem ser homogêneos, com a mesma capacidade (Q), ou não homogêneos (Q_k).
- 4) O número de veículos da frota pode ser limitado, ou seja, uma vez concluída uma entrega, o veículo pode ser alocado em outra viagem, ou ilimitado, ou seja, o veículo é utilizado uma única vez.
- 5) Os tempos de transporte $t_{l,l'} > 0$ e serviço $s_j > 0$.

A solução para o PPIPD consiste em um *programa integrado de produção-distribuição* o qual é dividido em: a) um *programa de produção* que especifica quando (data de início e de término) e onde (em qual máquina) cada ordem será processada e b) um *programa de entrega* que especifica a data de embarque de cada ordem e veículo, a rota realizada por cada veículo, quais ordens são transportadas por qual veículo e a data de entrega de cada ordem em cada cliente.

Neste contexto, as seguintes variáveis de decisão são definidas para a ordem j :

- 1) Data de conclusão C_j , a qual define a data de finalização ou conclusão da produção da ordem j .
- 2) Data de embarque (ou partida) $ST_j \geq C_j$, a qual define em que data o embarque será realizado, ou seja, em que data o veículo transportando a ordem j deixará a fábrica.
- 3) Data de entrega $DT_j \geq ST_j$, a qual define a data em que a ordem j será entregue para o cliente.
- 4) Tempo de fluxo (da produção à entrega) $F_j = DT_j - r_j$.
- 5) Retardação (Lateness) ou desvio da entrega $L_j = DT_j - d_j$.
- 6) Atraso de entrega (Tardiness) $T_j = \max\{0; L_j\}$.
- 7) Adiantamento de entrega (Earliness) $E_j = \max\{0; -L_j\}$.
- 8) Número de ordens atrasadas $U_j = \{1 \text{ se } L_j > 0; 0 \text{ se } L_j \leq 0\}$.

Tais variáveis constituem medidas individuais de desempenho de cada ordem. De modo a caracterizar o desempenho global do sistema, estes valores individuais devem ser somados para formar alguma medida agregada de desempenho. O objetivo da maioria dos modelos existentes consiste em otimizar (minimizar ou maximizar) uma ou mais medidas de desempenho agregadas relacionadas a tempo, custo ou receita.

As medidas de desempenho agregadas mais comumente usadas são:

- 1) Data máxima de entrega $DT_{max} = \max\{DT_j | j \in N\}$.
- 2) Soma total das datas de entrega $TDT = \sum_{j \in N} DT_j$.
- 3) Tempo total de fluxo $TFT = \sum_{j \in N} F_j$.
- 4) Atraso médio $\bar{T} = \frac{1}{N} \sum_{j \in N} T_j$.
- 5) Atraso total $TT = \sum_{j \in N} T_j$.
- 6) Retardação máxima $L_{max} = \max\{L_j | j \in N\}$.
- 7) Antecipação total $E = \sum_{j \in N} E_j$.

Outras considerações relacionadas a parâmetros de entrada, variáveis de decisão, critérios de otimização e diferentes configurações de sistema de produção podem ser obtidas em (CHEN, 2010).

Na Figura 2.5, é apresentado um esquema de um sistema integrado de produção e distribuição que processa ordens no conjunto de máquinas ($M_1, \dots, M_m$) e as embarca em lotes em diferentes veículos que, em seguida, fazem entregas aos clientes.

Figura 2.5 – Sistema integrado de produção e distribuição

Fonte: o autor

Vale observar que o sistema não mantém estoques. O tempo de armazenagem de produtos é o necessário para montar os lotes para envio aos clientes.

2.5. O PIPD-NWFS

Na seção anterior, o caso geral do problema de programação integrado da produção e distribuição foi definido. Nesta seção, são estabelecidas as características, configurações e restrições do PIPD-NWFS, foco desta pesquisa, que representa um caso particular do PIPD.

Para melhor entendimento do PIPD-NWFS, as características, configurações e restrições são descritas separadamente para ambos os sistemas de produção e distribuição.

2.5.1. O sistema de produção *no-wait flow shop*

As seguintes características, configurações e restrições descrevem o sistema NWFS estudado nesta Tese:

- 1) O sistema de produção é o *no-wait flow shop* simples, com as características descritas na seção correspondente deste capítulo.
- 2) O sistema está restrito a apenas uma fábrica ($\varphi = 1$).
- 3) Não existem operações de montagem que gerem interdependência entre ordens.
- 4) Todas as máquinas estão continuamente disponíveis.
- 5) Não existem máquinas alternativas ou em paralelo.
- 6) Todas as ordens estão disponíveis para processamento no instante inicial ($r_j = 0$).
- 7) Os tempos de processamento são sempre positivos ($p_{ji} > 0$).
- 8) Todos os demais tempos que não o de processamento (ex.: *setup* e movimentações) são irrelevantes.
- 9) O sistema não opera com restrições de data fixa de entrega (FD_j) ou data limite de entrega (d_j^u).
- 10) As datas de entrega das ordens (d_j) não são relevantes para a análise do desempenho do sistema.
- 11) Todas as ordens têm a mesma importância relativa ($w_j = w = 1$).

- 12) As ordens não são fracionadas em lotes menores para processamento.
- 13) O sistema produz por encomenda (MTO) e opera apenas para atender às ordens. Portanto, não é objetivo do sistema a formação de estoques. Tão logo as ordens são concluídas já estão disponíveis para embarque, transporte e entrega para os respectivos clientes.

2.5.2. O sistema de distribuição

Nesta Tese, são estudadas duas estratégias ou modelos de distribuição: a data fixa de embarque, saída ou coleta de ordens (DFC) e a data variável de embarque, saída ou coleta de ordens (DVC). Portanto, para o mesmo modelo de produção NWFS descrito anteriormente existem dois modelos de distribuição. As características de cada um deles são descritas a seguir.

Cabe observar que não é objetivo desta pesquisa avaliar qual dos dois modelos de distribuição (DFC ou DVC) é o melhor. Ambos são cabíveis dentro das situações e condições de sua seleção ou ativação. Também não é objetivo resolver o PPIPD-NWFS para ambos os modelos operando conjuntamente.

2.5.2.1. Data fixa de coleta (DFC)

Neste modelo de distribuição, as datas (ou intervalos) de coleta de ordens são previamente definidas pelos clientes ou pelo transportador.

A fábrica (sistema NWFS) deve respeitar as datas, buscando embarcar em lotes todas as ordens disponíveis em cada evento de coleta.

Apesar de não controlar as datas de coleta, a fábrica deve rotear e programar os veículos para obter melhores estimativas das datas de entrega; deve também carregar

o máximo de ordens possível em cada evento de coleta, respeitando a restrição de capacidade veicular.

As seguintes características, configurações e restrições descrevem o modelo DFC:

- 1) Em um instante t , um conjunto $J_t = \{j_1, j_2, \dots, j_j\}$ de ordens concluídas está disponível para embarque.
- 2) Em intervalos predefinidos I_t ou datas fixas $FP_t \in T$, com $I_t = FP_t - FP_{t-1}$, veículos realizam a coleta das ordens disponíveis J_t para envio aos clientes.
- 3) Toda ordem $j_j \in J_t$ deve aguardar a data de coleta para ser embarcada ($ST_j = FP_t \geq C_j$).
- 4) O tempo de espera para coleta é definido como $W_j = FP_t - C_j$.
- 5) Não existem janelas de entrega para os clientes.
- 6) A capacidade do sistema de transporte é limitada por uma quantidade $Q_k = Q_t$ definida por data fixa de coleta FP_t .
- 7) Em cada coleta, o número de ordens embarcadas é restringido por $Q_t \geq \sum_j q_j$.
- 8) Não há fracionamento de ordens, portanto, aquelas ordens disponíveis, mas não coletadas em uma determinada data devido à restrição de capacidade, devem aguardar a(s) próxima(s) data(s) de coleta para serem embarcadas.
- 9) Não há restrição de armazenagem no depósito para as ordens recém produzidas aguardando a coleta.
- 10) A rede de distribuição é formada pela fábrica (ponto de coleta) e pelos locais de entrega representando clientes finais dispersos geograficamente.
- 11) Não existe incompatibilidade entre as ordens, ou seja, qualquer ordem pode ser transportada conjuntamente com qualquer outra.
- 12) Os tempos de transporte da fábrica ('0') ao cliente l diretamente $t_{0,l} > 0$ e entre clientes $t_{l,l'} > 0$ são predefinidos.
- 13) Os tempos de serviço $s_j > 0$ incluem todos os demais tempos que não o de transporte (ex.: carregamento, descarregamento, manobra, descanso etc.).

A solução para o PPIPD-NWFS no modelo DFC consiste em um *programa integrado de produção-distribuição* dividido em: a) um *programa de produção no-wait flow shop* que especifica as datas de início e de término das ordens nas máquinas e b) um *programa de entrega* que especifica as datas de coleta das ordens, os veículos

que transportam as ordens e as respectivas datas de entrega das ordens para os clientes.

Na Figura 2.6, é apresentado o gráfico de Gantt de um exemplo de programa integrado contendo cinco ordens ($J_1, \dots, J_5$), três máquinas ($M_1, \dots, M_3$) e quatro veículos ($V_1, \dots, V_4$) para o modelo de distribuição DFC. Os blocos correspondentes à programação NWFS contêm o número da ordem e da máquina entre parênteses. Os blocos correspondentes à programação DFC contêm o número do veículo entre colchetes e as ordens coletadas entre parênteses.

Observa-se também na Figura 2.6 as datas de coleta (FP_i), definidas por intervalos equidistantes de tempo, nas quais os veículos chegarão na fábrica para realizarem as coletas das ordens. A data t_0 corresponde ao início do horizonte de programação.

Figura 2.6 – Programação NWFS no modelo DFC

Fonte: o autor

Na data FP_1 , não estão programadas ordens para serem coletadas, portanto, o veículo V_1 seguirá sem coletar nada. Já em FP_2 , as ordens J_1 e J_2 , programadas para serem concluídas anteriormente, serão coletadas pelo veículo V_2 . Em FP_3 e FP_4 , as demais ordens concluídas e aguardando embarque (J_3, J_4, J_5) serão coletadas pelos veículos V_3 e V_4 .

Os tempos de espera (W_j) por veículo são também apresentados no diagrama. Assume-se que todos os veículos têm capacidade suficiente para transportar as ordens programadas.

Na Figura 2.7, é apresentado outro exemplo de programação integrada no modelo DFC para oito ordens e três veículos (e as respectivas datas de coleta) no qual os veículos possuem restrições de capacidade.

Figura 2.7 – Sequências de produção e entrega

Fonte: o autor

No exemplo, as sequências de entrega das ordens não são alteradas após a alocação da carga em cada caminhão. A ordem J_6 não pôde ser programada para a entrega pelo veículo V_1 disponível em FP_1 por restrição de capacidade de transporte, sendo alocada no veículo V_2 que somente estará disponível em FP_2 .

Vale observar que as decisões de alocação são tomadas no início (t_0) do horizonte de programação, portanto, anteriormente às datas de coleta.

2.5.2.2. Data variável de coleta (DVC)

Neste modelo de distribuição, as datas de embarque, saída ou coleta de ordens não são predeterminadas.

Cabe à fábrica estabelecer as datas, buscando embarcar em lotes todas as ordens disponíveis em cada evento de embarque ou coleta, respeitando a restrição de capacidade veicular.

Neste caso, a fábrica coordena o transporte executado por uma frota homogênea de veículos pertencente ou não a ela que deve ser roteada e programada para realizar as entregas.

A principal característica que distingue o modelo DVC do DFC é que não existe nenhuma restrição quanto à escolha da data de coleta, ou seja, a qualquer instante, existindo capacidade de transporte disponível, a ordem $j_j \in J_t$ pode ser embarcada e despachada (coletada).

Como a subcontratação de serviço terceirizado de transporte (3PL) é uma alternativa, o número de veículos da frota é ilimitado. Apesar disso, o fabricante, com o objetivo de reduzir custos logísticos, aloca o máximo possível da capacidade a cada veículo e cada veículo executa apenas uma rota.

Quase todas as características, configurações e restrições definidas para o modelo DFC listadas na Subseção 2.5.2.1 (de 1 a 13) são também comuns ao modelo DVC com exceção do item 2 que pertence apenas ao modelo DFC. Além disso, a data FP_t (fixa) é substituída pela data VP_j (variável).

No modelo DVC, a data de coleta é formulada como $ST_j = VP_j = VP_k = VP_{j,k} = C_{j^u,k} = \max_j \{C_{j,k}\} = C_j + W_j$, ou seja, a data de coleta da ordem j é igual a data de conclusão da última ordem j^u , todas embarcada no veículo k , considerando o limite de capacidade veicular. Na fórmula, W_j é o tempo de espera da ordem j a ser embarcada no mesmo veículo, sendo determinado a partir de $C_{j^u,k}$.

No modelo DVC, parte-se do pressuposto que a fábrica aciona os veículos anteriormente às datas programadas de coleta para que estejam disponíveis no instante da coleta e que é carregado o máximo de ordens possível em cada veículo.

Tal qual o modelo DFC, a solução para o PIPD-NWFS no modelo DVC consiste também de um programa integrado de produção-distribuição com as mesmas informações descritas para o modelo DFC.

Na Figura 2.8, é apresentado o gráfico de Gantt de um exemplo de programa integrado contendo cinco ordens ($J_1, \dots, J_5$), três máquinas ($M_1, \dots, M_3$) e três veículos ($V_1, \dots, V_3$) para o modelo de distribuição DVC. Os blocos correspondentes à programação NWFS contêm o número da ordem e a máquina entre parênteses. Os blocos correspondentes à programação DVC contêm o número do veículo entre colchetes e as ordens embarcadas entre parênteses. A data t_0 é o início do horizonte de programação.

Observa-se também as datas VP_i nas quais os veículos realizarão as coletas. O veículo V_1 está programado para chegar na data VP_1 para a coleta das ordens J_1 e J_2 . A ordem J_1 deverá aguardar até que a ordem J_2 seja concluída para serem embarcadas juntas em V_1 . O veículo V_2 está programado para chegar em VP_2 para a coleta da ordem J_3 . O mesmo ocorre para o veículo V_3 programado para chegar em VP_3 para coletar J_4 e J_5 . Os tempos de espera (W_j) das ordens são também apresentados no diagrama. A data t_0 corresponde ao início do horizonte de programação.

Figura 2.8 – Programação NWFS no modelo DVC

Fonte: o autor

Neste exemplo, o programa de entrega prevê o uso de três veículos para o transporte das cinco ordens. Estes veículos são previamente alocados para cada ordem e serão acionados (notificados) anteriormente a data da coleta.

As datas programadas de coleta e o número de veículos são definidos com base na capacidade veicular, e, portanto, a programação garante que todos os veículos tenham capacidade suficiente para o transporte das ordens vinculadas a eles.

3. Revisão Bibliográfica

Este capítulo apresenta uma revisão bibliográfica de estudos que tiveram influência direta na modelagem do problema de otimização e na escolha e desenvolvimento dos métodos heurísticos propostos na pesquisa. Ele é dividido em três partes:

- 1) A primeira parte discorre brevemente sobre a complexidade computacional do PPNWFS e do PRV.
- 2) Na segunda parte, a ênfase é dada ao sistema *flow shop* tomado isoladamente, sendo apresentados estudos relevantes relacionados apenas ao PPNWFS.
- 3) Na terceira parte, a ênfase é dada ao PPIPD, sendo apresentados estudos relevantes que solucionam o problema por diferentes métodos, considerando diferentes configurações e restrições dos (sub)sistemas de produção e distribuição.

3.1. Complexidade computacional

Apesar de serem muito estudados, o PPNWFS e o PRV são problemas de otimização combinatória muito difíceis de serem solucionados até a otimalidade. Mesmo com avanços na pesquisa de métodos analíticos exatos e crescente aumento da capacidade computacional, vários pesquisadores reportam que apenas para poucos casos especiais (instâncias) de ambos os problemas (tratados isoladamente) existem algoritmos eficientes, ou seja, algoritmos que resolvem cada um deles em tempo polinomial ou pseudo polinomial, encontrando a solução exata ou o ótimo global. A solução torna-se ainda mais árdua de ser encontrada quando o PPNWFS e o PRV são considerados de forma integrada (GUPTA; STAFFORD JR., 2006; GEISMAR *et al.*, 2008; NADERI; RUIZ; ZANDIEH, 2010; MONTOYA-TORRES *et al.*, 2015).

Infelizmente para a Indústria, os casos especiais são simples e estão longe de retratarem a magnitude dos problemas reais. Os respectivos casos gerais são insolúveis do ponto de vista computacional independente do critério de otimização adotado. Até o presente momento nenhum algoritmo eficiente foi desenvolvido para solucionar exatamente os casos gerais dos dois problemas tanto isoladamente quanto de forma integrada (GUPTA; STAFFORD JR., 2006; NADERI; RUIZ; ZANDIEH, 2010; MONTOYA-TORRES *et al.*, 2015).

No âmbito da complexidade computacional, o PPNWFS e o PRV são ambos classificados como *NP-hard* (GAREY; JOHNSON; SETHI, 1976; LENSTRA; RINNOOY-KHAN, 1981; RÖCK, 1984; SRISKANDARAJAH; LADET, 1986).

A complexidade computacional do PPIPD para ambientes *flow shop* é estudada por Soukhal, Oulamara e MARTINEAU (2005). Os pesquisadores concluem que o problema é fortemente *NP-hard* para ambientes *flow shop* de duas máquinas e veículo único com capacidade limitada.

Consequentemente, nas últimas três décadas, o direcionamento da pesquisa acadêmica têm privilegiado bastante as abordagens (meta)heurística e híbrida, como o descrito em várias revisões bibliográficas para o PPFS, incluindo o PPNWFS (ALLAHVERDI *et al.*, 2008; RIBAS; LEISTEN; FRAMIÑAN, 2010; PAN; RUIZ, 2013; ALLAHVERDI, 2016; NAGANO; MIYATA, 2016b; NEUFELD; GUPTA; BUSCHER, 2016), e para o PRV (BRÄYSY; GENDREAU, 2005a, 2005b; PARRAGH *et al.*, 2008a, 2008b; LAPORTE, 2009; KUMAR; PANNEERSELVAM, 2012; PILLAC *et al.*, 2013; CAMPBELL; WILSON, 2014; MONTOYA-TORRES *et al.*, 2015). Em todas as revisões citadas, ambos os problemas são solucionados separadamente.

Dentre as metaheurísticas, as estratégias de busca em vizinhança variável (*variable neighborhood search* – VNS), recozimento simulado (*simulated annealing* – SA), busca local iterativa (*iterated local search* – ILS) e busca iterativa gulosa (*iterated greedy algorithm* – IG), assim como estratégias conjugadas (híbridas) destes métodos, têm sido muito utilizadas para a solução de ambos os problemas. Os artigos de Allahverdi *et al.* (2008), Allahverdi (2016), Laporte (2009) e Montoya-Torres *et al.*

(2015) descrevem o uso das principais metaheurísticas para a solução de ambos os problemas isoladamente.

3.2. PPNWFS não integrado ao sistema de transporte

Nesta seção, são apresentados estudos onde o PPNWFS ‘clássico’ (não integrado ao sistema de transporte) é resolvido por meio de heurísticas construtivas e metaheurísticas adotadas nesta Tese – ILS, IG, SA e VNS. Nos estudos revisados, as metaheurísticas são usadas isoladamente ou em conjunto (híbridas).

3.2.1. Recozimento simulado

Inspirado em processos termodinâmicos, o recozimento simulado – *simulated annealing* (SA) – (KIRKPATRICK; VECCHI, 1983 e CERNY, 1985) é uma metaheurística para construção de algoritmos de busca local que utilizam procedimentos específicos para movimentos ascendentes (hill-climbing) os quais objetivam escapar do ótimo local. Em cada iteração, o algoritmo SA compara os valores de três soluções, a ‘atual’ (s'), a anterior (s) e a ‘melhor’ (s^*) até a iteração atual. Se a solução ‘atual’ for superior às outras duas, ela será imediatamente selecionada para ser a ‘melhor’ e a próxima a gerar uma nova solução. Caso contrário, a solução ‘atual’ (mesmo sendo a pior) tem a chance de ser selecionada para a próxima iteração mediante a aplicação de um critério de aceitação baseado em um modelo de distribuição probabilística originalmente proposto por Metropolis *et al.* (1953). Tal modelo probabilístico é inserido no algoritmo SA o qual seleciona (probabilisticamente) soluções ruins para as próximas iterações na esperança que estas levem a alguma solução melhor que s^* (EGLESE, 1990; NIKOLAEV; JACOBSON, 2010; BOUSSAÏD; LEPAGNOT; SIARRY, 2013).

Na Figura 3.1, é apresentado um algoritmo para SA (baseado em EGGLESE, 1990 e BOUSSAÏD; LEPAGNOT; SIARRY, 2013).

Fink e Voß (2003) avaliam a aplicação de diferentes algoritmos SA conjuntamente com algoritmos de busca tabu (TS) para o PPNWFS, objetivando minimizar o tempo total de fluxo (TFT). Por meio de experimentos computacionais os autores concluem que a eficácia de ambas metaheurísticas depende crucialmente do tamanho da instância, da qualidade da solução inicial e da capacidade computacional (CPU) disponível.

Figura 3.1 - Algoritmo para a metaheurística SA

```

1  Defina a função objetivo  $F(.)$  a ser minimizada
2  Escolha aleatoriamente uma solução inicial  $s$  do espaço de soluções
3  Defina a função de temperatura de recozimento  $T(.)$ 
4  Inicie a temperatura  $T(0) > 0$  e o contador da temperatura  $k$ 
5      Repita
6          Busca local: selecione aleatoriamente  $s'$  na vizinhança  $N(s)$ 
7          Aceitação:
8              Se  $F(s') < F(s)$  então
9                   $s \leftarrow s'$ 
10             Fim
11             Se  $F(s') < F(s^*)$  então
12                  $s^* \leftarrow s'$ 
13             Fim
14             Senão
15                 Selecione  $p = \text{random}[0, 1]$ 
16                  $s \leftarrow s'$  se  $p < \exp(-[f(s') - f(s)]/T)$ 
17             Fim
18         Até que o equilíbrio termodinâmico seja atingido:
19         Se  $T(k) > 0$  então
20             Resfriamento:
21                  $k \leftarrow k+1$ 
22                 Diminua a temperatura  $T=T(k)$ 
23         Fim
24     Até que o critério de parada seja satisfeito
25     Fim
26     Retorne a melhor solução  $s^*$  encontrada

```

Fonte: (EGLESE, 1990) e (BOUSSAÏD; LEPAGNOT; SIARRY, 2013)

Wang, Yang e Lin (2006) desenvolvem e comparam os desempenhos de algoritmos GA e SA para o PPNWFS de duas e três máquinas, objetivando a minimização do *makespan* (C_{max}). Os pesquisadores reportam o desempenho

superior do algoritmo SA frente ao algoritmo GA tanto na qualidade das soluções obtidas quanto na performance computacional para os mesmos critérios de parada.

Em um estudo similar, Adressi, Hassanpour e Azizi (2016) propõem e analisam os desempenhos de algoritmos SA e GA para o PPNWFS de duas máquinas para o critério C_{max} em um sistema NWFS sujeito a quebras aleatórias de máquinas. Os autores reportam a superioridade do algoritmo SA em comparação ao GA.

Aldowaisan e Allahverdi (2012) desenvolvem e avaliam novas regras de prioridade (*dispatching rules*) para o PPNWFS, objetivando a minimização do critério de atraso total (TT). Em seguida, desenvolvem algoritmos SA e GA que utilizaram as regras propostas para gerarem as soluções iniciais em vez de aplicarem procedimentos de geração aleatória, obtendo assim ganhos na qualidade das soluções finais. Por fim, avaliam os desempenhos dos dois algoritmos propostos e concluem que o algoritmo SA é superior ao GA em tempo computacional e empata na qualidade das soluções geradas.

Aydilek e Allahverdi (2012) propõem quatro algoritmos SA (variantes HA, mSA, HH1 e HH2) para o PPNWFS, objetivando a minimização do critério C_{max} sujeito a limites do tempo médio de conclusão de ordens (MCT) e conduzem um estudo experimental sobre seus desempenhos. Os autores reportam que a heurística HH1 é significativamente superior às demais na qualidade das soluções geradas, mas com maior uso de tempo de processamento computacional.

Aydilek e Allahverdi (2013) propõem melhorias em dois dos algoritmos reportados anteriormente (AYDILEK; ALLAHVERDI, 2012), gerando novas heurísticas SA (variantes eSA e eHH). Uma das heurísticas propostas (eSA) obtém desempenho superior à melhor heurística (HH1) avaliada no estudo anterior.

Em um extenso estudo, Allahverdi e Aydilek (2014) desenvolvem e avaliam onze algoritmos baseados nas metaheurísticas GA, SA e evolução diferencial (DE), conjuntamente com outras heurísticas especializadas para resolver o PPNWFS com restrições de tempo de *setup*, objetivando minimizar o critério de tempo total de conclusão de ordens (TCT) sujeito a limites do C_{max} . Os resultados comprovam a

superioridade da metaheurística SA, mais especificamente o algoritmo ISA-2 (uma variante do SA), sobre os demais algoritmos.

3.2.2. Busca em vizinhança variável

Busca em vizinhança variável – *variable neighborhood search* (VNS) – (MLADENOVIC; HANSEN, 1997) é uma metaheurística para construção de algoritmos de busca local cuja estratégia básica consiste em explorar sistematicamente a dinâmica de mudança de estruturas de vizinhança de uma dada solução, buscando tanto por um valor ótimo na fase de busca local quanto por um valor de escape na fase de perturbação. Em uma dada iteração, o algoritmo VNS gera uma solução ‘intermediária’ por meio de um procedimento de perturbação a partir da solução ‘atual’ e para uma dada estrutura de vizinhança. A partir desta solução, uma busca local é realizada. Caso uma das soluções ‘candidatas’ obtidas na busca local seja melhor que a solução ‘atual’, a ‘candidata’ se torna a solução ‘atual’ e o ciclo inicia-se novamente com a mesma estrutura de vizinhança. Caso a solução ‘intermediária’ seja pior que a ‘atual’, o algoritmo seleciona outro procedimento de geração de estrutura de vizinhança e o ciclo recomeça a partir da mesma solução ‘atual’ (HANSEN; MLADENOVIC, 2001; BOUSSAÏD; LEPAGNOT; SIARRY, 2013).

Na Figura 3.2, é apresentado um algoritmo para VNS (baseado em BOUSSAÏD; LEPAGNOT; SIARRY, 2013).

Grabowski e Pempera (2005) propõem e avaliam cinco algoritmos baseados nas metaheurísticas VND (uma variante do VNS) e TS para o PPNWFS, objetivando a minimização do C_{max} . Os algoritmos são testados para diferentes instâncias de *benchmark* (HELLER, 1960; CARLIER, 1978; REEVES, 1995), sendo comparados com outros três algoritmos da literatura acadêmica baseados em GA, SA e VNS, além de uma heurística específica (HAJ). Os resultados experimentais demonstram que os métodos propostos são superiores aos demais.

Figura 3.2 - Algoritmo para a metaheurística VNS

```

1  Defina a função objetivo  $F(.)$  a ser minimizada
2  Selecione o conjunto de estruturas de vizinhança  $N_n, n = 1, \dots, n_{\max}$ 
3  Escolha aleatoriamente uma solução inicial  $s$  do espaço de soluções
4  Enquanto o critério de parada não for satisfeito faça
5       $n \leftarrow 1$ 
6      Enquanto  $n \leq n_{\max}$  faça
7          Perturbação: selecione aleatoriamente  $s'$  na  $n$ -ésima vizinhança  $N_n(s)$ 
8          Busca local: obtenha  $s''$  a partir de  $s'$ 
9          Se  $F(s'') < F(s)$  então
10              $s \leftarrow s''$ 
11          Fim
12          Senão
13              $n \leftarrow n + 1$  (uma nova estrutura de vizinhança será adotada)
14          Fim
15      Fim
16       $s^* \leftarrow s$ 
17  Fim
18  Retorne a melhor solução  $s^*$  encontrada

```

Fonte: (BOUSSAÏD; LEPAGNOT, SIARRY, 2013)

Pan, Wang e Zhu (2006) desenvolvem um algoritmo VNS para o PPNWFS, objetivando a minimização do $C_{\max}$. Os autores comparam o desempenho do algoritmo com outro método (GRABOWSKI; PEMPERA, 2005), demonstrando experimentalmente a sua superioridade – tanto em eficácia (qualidade das soluções) quanto em eficiência (tempo computacional) – para encontrar soluções exatas ou próximas do ótimo global.

Jaboui, Eddaly e Siarry (2009) propõem um algoritmo híbrido GA-VNS para o PPNWFS com objetivo de minimizar os critérios $C_{\max}$ e TFT. O método VNS é utilizado para melhorar o resultado do algoritmo GA pela exploração de diferentes estruturas de vizinhança. Os autores demonstram experimentalmente que o método desenvolvido é competitivo para ambos os critérios de otimização quando comparado a outros dez algoritmos da literatura acadêmica, usando diferentes instâncias de *benchmark* (HELLER, 1960; TAILLARD, 1993; REEVES, 1995).

Samarghandi e Elmekawy (2011) desenvolvem um algoritmo híbrido VNS-TS para o PPNWFS de duas máquinas e dependente de tempo de *setup*, com o objetivo

de minimizar o C_{max} . Os resultados experimentais demonstram que o algoritmo gera soluções ótimas para instâncias de teste de pequeno porte. Para instâncias maiores, o algoritmo proposto é comparado com outro algoritmo de busca local (2-opt), mostrando-se eficiente e gerando boas soluções.

Komaki e Malakooti (2017) propõem um algoritmo GVNS (uma variante do VNS) para o PPNWFS distribuído (para múltiplas fábricas), objetivando minimizar C_{max} . O algoritmo implementa duas regras de alocação de ordens nas fábricas. Em um estudo experimental, os autores comparam o método proposto com diferentes heurísticas da literatura acadêmica. Os resultados demonstram que a variante que usa uma determinada regra de alocação obtém desempenho superior à outra variante que não usa.

3.2.3. Busca local iterativa

Busca local iterativa – *Iterated Local Search* (ILS) – (STÜTZLE, 1999; LOURENÇO; MARTIN; STÜTZLE, 2016) é uma metaheurística baseada na estratégia simples de não se tentar explorar todo o espaço de soluções candidatas, mas focar em regiões adjacentes a soluções ‘promissoras’ geradas por algum subalgoritmo, em geral uma heurística de busca local (ex.: um algoritmo SA), inserida em uma estrutura algorítmica iterativa maior. Tais regiões são geradas e exploradas por meio do processo de perturbação da solução ‘atual’ com a esperança de que a solução intermediária obtida seja uma solução ‘promissora’, ou seja, esteja dentro da ‘região de atração’ de alguma solução ‘candidata’ (a ser encontrada) que seja melhor que a solução ‘atual’. A solução intermediária (‘promissora’) não necessariamente é melhor que a solução ‘atual’, basta estar dentro de uma região geradora de alguma solução ‘candidata’ cujo valor seja (esperançosamente) melhor que a ‘atual’, região esta que será explorada em seguida à perturbação pelo mesmo ou outro subalgoritmo de busca local inserido na estrutura algorítmica maior (LOURENÇO; MARTIN; STÜTZLE, 2010; BOUSSAÏD; LEPAGNOT; SIARRY, 2013).

Na Figura 3.3, é apresentado um algoritmo para a metaheurística ILS (baseado em BOUSSAÏD; LEPAGNOT; SIARRY, 2013).

Figura 3.3 - Algoritmo para a metaheurística ILS

```

1  Defina a função objetivo  $F(.)$  a ser minimizada
2  Escolha aleatoriamente uma solução inicial  $s$  do espaço de soluções
3  Busca local: obtenha  $s'$  a partir de  $s$ 
4      Repita até que o critério de parada seja satisfeito
5          Perturbação: selecione aleatoriamente  $s''$  na vizinhança  $N(s')$ 
6          Busca local: obtenha  $s'''$  a partir de  $s''$ 
7          Se  $F(s''') < F(s')$  então
8               $s' \leftarrow s'''$ 
9          Fim
10         Se critério de parada for satisfeito então
11              $s^* \leftarrow s'$ 
12         Fim
13     Fim
14  Retorne a melhor solução  $s^*$  encontrada

```

Fonte: (BOUSSAÏD; LEPAGNOT; SIARRY, 2013)

Ruiz e Allahverdi (2007) desenvolvem cinco heurísticas simples e duas heurísticas estocásticas para solucionar PPNWFS, com o objetivo de minimizar o critério TCT. Uma das heurísticas estocásticas propostas, denominada ILS_RA, baseia-se na metaheurística ILS. O algoritmo faz uso de uma das heurísticas simples (H5) também proposta no estudo para gerar a solução inicial. Por meio de uma extensa análise estatística experimental, os autores comprovam a superioridade do algoritmo ILS_RA frente às heurísticas simples e a outros dois métodos (ACO e TRIPS) selecionados da literatura acadêmica.

Naderi *et al.* (2012) propõem três algoritmos baseados nas metaheurísticas ILS e IG para o PPNWFS para a minimização bi critério das funções C_{max} e TT. Dois dos algoritmos são baseados na metaheurística IG com dois procedimentos de busca local, o 'guloso' e o 'moderado'. O terceiro algoritmo é baseado na metaheurística ILS cujo procedimento é denominado 'reduzido'. Os pesquisadores conduzem experimentos computacionais comparando os três algoritmos propostos com outros dois (MOSA e

AIA) e comprovam estatisticamente que aquele que utilizou o procedimento ‘moderado’ é superior a todos os demais.

3.2.4. Busca iterativa gulosa

Busca iterativa gulosa – *Iterated Greedy Algorithm* (IG) – (JACOBS; BRUSCO, 1995; MARCHIAROLI; STEENBEEK, 2000) gera uma sequência de soluções por meio de heurísticas construtivas gulosas, usando duas fases principais: uma destrutiva e outra construtiva. Durante a fase destrutiva, um conjunto de elementos ou componentes são removidos de uma solução candidata completa s_i gerada previamente. Na fase seguinte de construção, uma heurística construtiva é aplicada para gerar uma solução completa s_{i+1} a partir da solução parcial anterior, portanto, reconstruindo a solução parcialmente destruída com os elementos anteriormente removidos até que se torne completa novamente, mas desta vez aplicando-se um procedimento para a (re)inserção que pode ser diferente daquele utilizado para a geração da solução completa anterior s_i . Uma vez gerada a solução s_{i+1} , um critério de aceitação é aplicado para decidir se ela substituirá a solução s_i (RUIZ; STÜTZLE, 2007).

Os métodos IG e ILS diferem em um aspecto principal: em vez de executar as fases de busca local e perturbação presentes nos algoritmos ILS, os algoritmos IG executam, respectivamente, as fases de destruição e reconstrução (RUIZ; STÜTZLE, 2007).

Na Figura 3.4, é apresentado um algoritmo para a metaheurística IG (baseado em RUIZ; STÜTZLE, 2007).

Ruiz e Stützle (2007) propõem e avaliam o primeiro algoritmo IG para o PPNWFS objetivando a minimização do critério C_{max} . O algoritmo proposto utiliza as estratégias de destruição e construção conjuntamente com busca local para melhorar a solução inicial gerada pela heurística NEH. Na fase de destruição, a solução completa atual é parcialmente destruída pela eliminação aleatória de um número predefinido de ordens

(d). Na fase de construção, as ordens eliminadas são reinseridas na solução parcial uma a uma utilizando novamente o procedimento de inserção da heurística NEH. Na fase seguinte, a solução completa é melhorada por meio de um algoritmo de busca local baseado na estrutura de inserção simples na qual ordens aleatoriamente selecionadas uma a uma são reinseridas em posições aleatoriamente selecionadas.

Figura 3.4 - Algoritmo para a metaheurística IG

```

1  Defina a função objetivo  $F(.)$  a ser minimizada
2  Escolha aleatoriamente uma solução inicial  $s$  do espaço de soluções
3  Defina  $d$ : o número de elementos a serem retirados de  $s$ 
4   $n \leftarrow 1$ 
5  Repita até que o critério de parada seja satisfeito
6      Enquanto  $n \leq d$  faça
7          Destruição:
8          i. retire aleatoriamente um elemento de  $s$ 
9          ii. insira o elemento eliminado em  $L(s)$ 
10          $s' \leftarrow s$ 
11          $n \leftarrow n+1$ 
12      Fim
13       $n \leftarrow 1$ 
14      Enquanto  $n \leq d$  faça
15          Construção:
16          i. retire sem reposição um elemento  $j$  de  $L(s)$ 
17          ii. obtenha  $s''$  pela melhor inserção de  $j$  em  $s'$ 
18           $s \leftarrow s''$ 
19           $n \leftarrow n+1$ 
20      Fim
21      Busca local: obtenha  $s'''$  a partir de  $s''$ 
22      Se  $F(s''') < F(s)$  então
23           $s \leftarrow s'''$ 
24      Fim
25      Se critério de parada for satisfeito então
26           $s^* \leftarrow s$ 
27      Fim
28  Fim
29  Retorne a melhor solução  $s^*$  encontrada

```

Fonte: (RUIZ; STÜTZLE, 2007)

Ainda em Ruiz e Stützle (2007), dois critérios de aceitação de solução são utilizados: um critério simples – o qual compara a solução gerada com a melhor

solução encontrada na iteração atual e seleciona a melhor dentre elas – e um segundo critério baseado na regra de aceitação clássica de algoritmos SA. O desempenho do algoritmo proposto é comparado com outros treze métodos. Os resultados experimentais comprovam a superioridade do algoritmo proposto frente aos demais.

Em um estudo similar, Ruiz e Allahverdi (2009) propõem um algoritmo IG para o PPNWFS objetivando a otimização bi critério do C_{max} e atraso máximo (L_{max}). O que diferencia este algoritmo do anterior é que na fase pós-construção, uma nova solução é gerada por meio de um algoritmo GA e esta é selecionada ou não para substituir a solução anterior com base na regra de aceitação dos algoritmos SA. O desempenho do algoritmo IG proposto é comparado com outros onze métodos. Os resultados experimentais comprovam a sua superioridade frente aos demais.

Shao, Pi e Shao (2017) propõem três variantes IG para resolver o PPNWFS em múltiplas plantas fabris, objetivando a otimização do critério C_{max} . Cada variante utiliza uma estratégia distinta de geração de estruturas de vizinhança (NSSG). Semelhante ao que é descrito em outros trabalhos (RUIZ; STÜTZLE, 2007 e RUIZ; ALLAHVERDI, 2009), os algoritmos propostos utilizam as estratégias de: a) destruição, por seleção aleatória de ordens e b) construção, por reinserção de ordens por meio do método NEH, para gerar uma nova solução candidata. Uma distinção entre os trabalhos anteriores é que neste os sistemas *flow shop* são distribuídos por diferentes locais (fábricas) e o método proposto realiza a programação centralizada de todas as operações, considerando que uma ordem pode ser programada em qualquer local (fábrica). Os métodos IG propostos são testados para quatro estratégias distintas de geração de estruturas de vizinhança: duas utilizando *swap* (simples e multiplanta) e duas utilizando inserção (simples e multiplanta). As estruturas de vizinhança são ainda modificadas de forma aleatória ou determinística. A solução gerada na fase de construção é melhorada por meio de busca local. Os critérios de aceitação são iguais aos utilizados em (RUIZ; STÜTZLE, 2007). Os algoritmos IG propostos são comparados entre si para diferentes métodos de geração de soluções iniciais.

Li *et al.* (2018) desenvolvem e avaliam um algoritmo baseado nas metaheurísticas IG e VND para a resolução do PPNWFS dependente de *setup*, objetivando minimizar o critério TFT. O modelo de otimização considera os efeitos de aprendizado e

esquecimento (LFE) na determinação dos tempos de processamento das ordens nas máquinas. O algoritmo proposto é comparado com quatro heurísticas (BIH, TRIPS, TRIPS_M, QUARTS) e duas metaheurísticas (IL_RSLs E ICX). Os resultados experimentais concluem que o algoritmo proposto é superior aos demais tanto na qualidade das soluções geradas quanto na performance computacional.

Ding *et al.* (2015a) desenvolvem um algoritmo baseado nas metaheurísticas IG e TS para o PPNWFS para minimização do critério C_{max} . O algoritmo proposto (TMIG) utiliza uma estratégia TS de reconstrução da solução para melhorar a exploração do espaço de soluções. Em um experimento computacional com instâncias de *benchmark* (TAILLARD, 1993), o desempenho do algoritmo TMIG é comparado a outros seis algoritmos da literatura acadêmica. A análise estatística dos resultados demonstra que o algoritmo proposto é bastante competitivo, alcançando desempenho superior na maior parte das instâncias testadas.

Ding *et al.* (2015b) propõem e avaliam três algoritmos, dois deles baseados na metaheurística IG e o terceiro baseado no procedimento NEH, para resolver o PPNWFS, minimizando o critério TT. Os autores investigam ‘propriedades incrementais’ de inserção de ordens no programa de produção que levam à redução da complexidade computacional e desenvolvem os três algoritmos (ANEH, AIG1, AIG2) considerando estas propriedades. Em seguida, em um experimento computacional, os três algoritmos propostos são comparados com outros dois algoritmos da literatura acadêmica (MNEH e FISA), sendo executados em instâncias de *benchmark* (TAILLARD, 1993). Os resultados confirmam a melhora do desempenho computacional dos algoritmos propostos se comparados às execuções que não levam em consideração as ‘propriedades incrementais’ e demonstram estatisticamente que eles também são mais eficientes (tempo de processamento computacional) e eficazes (qualidade da solução) que os selecionados da literatura.

3.2.5. Heurísticas construtivas

Heurísticas construtivas são definidas como procedimentos algorítmicos que constroem uma ou mais soluções a partir de uma solução inicial nula (programa ‘vazio’) pela inserção sucessiva (dentro do programa) de ordens de uma lista de ordens programáveis (NAGANO; MIYATA, 2016b).

Em geral, uma heurística construtiva parte de uma lista ou conjunto de ordens disponíveis SO_i (programáveis) em determinada iteração i ($1 \leq i \leq n$) do algoritmo e por meio de alguma regra ou critério de seleção retira-se as ordens de SO_i e as insere em determinadas posições do conjunto PS_i (programa parcial) por meio de alguma regra ou critério de inserção que define uma sequência ordenada (permutação). Uma vez retirada de SO_i e inserida em PS_i a ordem não é mais reavaliada até que uma solução completa seja gerada, ou seja, até que todas as ordens de SO_i tenham sido inseridas em PS_i , com $SO_{i=n} = \{\emptyset\}$ e $|PS_{i=n}| = n$ (FRAMINAN; GUPTA; LEISTEN, 2004).

Em geral, uma estratégia de construção implementa duas decisões principais (FRAMINAN; GUPTA; LEISTEN, 2004):

- 1) Seleção. Uma ou mais ordens são selecionadas de SO_i . Deve-se decidir qual(is) ordem(ns) selecionar.
- 2) Inserção. Cada ordem selecionada deve ser inserida em uma única posição (slot) de PS_i . Uma ou mais posições são avaliadas. Deve-se decidir em qual posição cada ordem deve ser inserida.

As regras de seleção e inserção da estratégia de construção são, por definição, ‘míopes’ e determinísticas, ao utilizarem alguma propriedade instantânea da ordem (índice) ou do programa parcial (FRAMINAN; GUPTA; LEISTEN, 2004).

Algumas heurísticas construtivas adotam o procedimento de reinserção de ordens. Na reinserção, uma ou mais ordens são selecionadas a partir do programa (parcial ou completo) e são reinseridas em outras posições após a avaliação do efeito da

mudança no critério de otimização. A regra (estratégia) de reinserção pode seguir a mesma regra adotada para a inserção ou pode ser diferente (NAGANO; MIYATA, 2016b).

Ao longo das últimas quatro décadas diferentes estratégias de construção (regras de seleção e inserção) foram propostas para a construção de programas por meio de heurísticas construtivas de programação. NAGANO e MIYATA (2016b) descrevem diferentes estratégias e heurísticas para o PPNWFS.

Nagano, Miyata e Araújo (2015) e Nagano e Miyata (2016a) desenvolvem e avaliam heurísticas construtivas para o PPNWFS. Os resultados experimentais destes estudos demonstram que as heurísticas propostas são bastante competitivas frente a outras heurísticas também avaliadas nos estudos.

Esta Tese adota três estratégias construtivas para a criação das heurísticas de geração da solução inicial dos métodos propostos: BAH e BIH (BIANCO; GIORDANI, 1999) e NEH (NAWAZ *et al.*, 1983).

3.3. Sistemas integrados de produção e distribuição (SIPD)

Nesta seção, são apresentados estudos relevantes relacionados ao PPIPD onde as duas estratégias DFC e DVC são usadas.

3.3.1. Data fixa de embarque ou coleta (DFC)

Em diversas situações práticas da gestão de operações, as datas de embarque ou coleta de produtos são predeterminadas ou fixas. Isto é particularmente relevante quando (HALL; LESAOANA; POTTS, 2001; LI, S.; LI, M., 2014; LIU; HSU, 2015):

- 1) A operação de entrega é um processo complexo e custoso que requer a determinação antecipada da sequência de produção e de entrega de cada componente ou (sub)conjunto (ex.: maquinário pesado; obras de infraestrutura).
- 2) O cliente considera os atrasos de entrega como disruptivos de sua operação e requer que as entregas sejam feitas pelo fornecedor em intervalos repetitivos (e muitas vezes fixos) definidos pelo cliente.
- 3) O ciclo financeiro do fornecedor efetivamente define as datas fixas de entrega.
- 4) Na gestão da cadeia de suprimentos, economias de escala relacionadas ao custo de entrega ou preferências definidas pelo cliente (ou qualquer tomador de decisão a jusante da cadeia, tal como o transportador contratado) determinam as datas (ou intervalos) de entrega possíveis.

Vários estudos acadêmicos foram publicados para o planejamento e programação de SIPD com datas de embarque ou coleta fixas ou predeterminadas. Mas na quase totalidade dos casos, os sistemas de produção estudados são simples, sendo modelados como:

- 1) Máquina única (STECKE; ZHAO, 2007; LU; YUAN; ZHANG, 2008; ZHONG; CHEN; CHEN, 2010; FU; HOU; ZHAO, 2012; LEUNG; CHEN, 2013; SEDDIK; GONZALES; KEDAD-SIDHOUM, 2013; LI, S.; LI, M., 2014; LI, S.; LI, M., 2015; LI *et al.*, 2016; AGNETIS; ALOULOU; KOVALYOV, 2017; LI; CHEN; TANG, 2017).
- 2) Máquinas em paralelo (LI; SIVAKUMAR; GANESAN, 2008; LI; JIA; LEUNG, 2015; MENSENDIEK; GUPTA; HERRMANN, 2015).

Nos estudos com máquina única ou em paralelo, os pesquisadores modelam os SIPD considerando diferentes configurações do sistema de distribuição. Em geral, para cada data de embarque ou coleta é associada uma capacidade de transporte:

- 1) Ilimitada (STECKE; ZHAO, 2007; ZHONG; CHEN; CHEN, 2010; SEDDIK; GONZALES; KEDAD-SIDHOUM, 2013; MENSENDIEK; GUPTA; HERRMANN, 2015; LI; CHEN; TANG, 2017).
- 2) Limitada e definida de forma agregada, ou seja, sem a consideração explícita do número de veículos disponíveis e, portanto, sem roteamento (LI; SIVAKUMAR; GANESAN, 2008; FU; HOU; ZHAO, 2012).

- 3) Limitada e definida por número de veículos disponíveis e capacidade individual de cada veículo (LU; YUAN; ZHANG, 2008; LEUNG; CHEN, 2013; LI, S.; LI, M., 2014; LI, S.; LI, M., 2015; LI; JIA; LEUNG, 2015; LI *et al.*, 2016; AGNETIS; ALOULOU; KOVALYOV, 2017). Destes, apenas o estudo de Li *et al.* (2016) considera o roteamento de veículos.

Raros são os estudos que consideram sistemas de produção mais complexos (ex.: *job shop*, *flow shop* e *open shop*) com datas fixas de embarque ou coleta e com enfoque operacional. São exemplos os artigos de Hall, Lesaoana e Potts (2001), Liu e Hsu (2015) e Han *et al.* (2019). Destes, apenas Han *et al.* (2019) realiza o roteamento de veículos.

Hall, Lesaoana e Potts (2001) analisam a complexidade e a solvibilidade de diferentes problemas de programação de produção e desenvolvem algoritmos polinomiais e pseudo polinomiais para diferentes configurações do sistema de produção, incluindo o *flow shop* de duas e três máquinas, para diferentes critérios de otimização, considerando datas de partida fixas e entregas diretas. Nenhuma restrição de capacidade de entrega é considerada na modelagem.

Liu e Hsu (2015) propõem nove regras de prioridade e comparam seus desempenhos entre si e com outras três regras da literatura acadêmica para solucionar o PPIPD em sistemas *job shop* com entregas diretas em intervalos fixos (predefinidos), objetivando minimizar a soma dos custos de atraso e antecipação de entrega. Nenhuma restrição de capacidade de entrega é considerada na modelagem. Os resultados experimentais concluem estatisticamente que as versões das regras que incorporam explicitamente as informações de custos de atraso e antecipação são superiores às regras originais.

Han et al, (2019) estudam o PPIPD de um sistema integrado onde operam um fornecedor, um fabricante e diversos clientes dispersos geograficamente. Os autores modelam o sistema como um *flow shop* de duas máquinas (fornecedor-fabricante). Para cada cliente, um conjunto de ordens deve ser processado primeiramente no fornecedor. Em seguida, as ordens são enviadas diretamente para o fabricante para processamento final. O fornecedor e o fabricante operam próximos um do outro. Não

há restrição de capacidade de transporte entre eles e cada ordem concluída é enviada imediatamente para o fabricante com o mesmo tempo de transporte. Uma vez processadas no fabricante, as ordens concluídas são enviadas aos clientes. Para atender aos clientes, o fabricante conta com o serviço de entrega de transportadores contratados (3PL) que operam com datas fixas de coleta e capacidade de transporte restrita. Os autores desenvolvem algoritmos em programação dinâmica para a solução exata do problema e discutem a viabilidade de uso destes algoritmos para casos reais.

3.3.2. Data variável de embarque ou coleta (DVC)

Na grande maioria dos modelos de planejamento e programação de SIPD, os veículos de entrega estão livres para partirem a qualquer tempo e os produtos estão disponíveis para serem enviados a qualquer instante após a sua produção (LIU; HSU, 2015).

Tal qual para o modelo DFC, vários estudos acadêmicos foram publicados na última década para o PPIPD no modelo DVC e em sua maioria foram desenvolvidos para sistemas de produção e distribuição simples (MOONS *et al.*, 2017; KUMAR *et al.*, 2020), com a maior parte dos estudos sendo conduzidos para sistemas de máquina única ou em paralelo. São alguns exemplos os trabalhos de Chang e Lee (2004), Huo, Leung e Wang (2010), Ullrich (2013), Low (2014) e Tavares-Neto e Nagano (2019).

Alguns pesquisadores desenvolveram métodos de solução para o PPIDP em sistemas *flow shop* com diferentes configurações do sistema de distribuição:

- 1) Número ilimitado de veículos com capacidade ilimitada (WANG *et al.*, 2016; PORSEVI; BALAJI; JAWAHAR, 2018; WANG *et al.*, 2018). Nenhum estudo deste grupo considera explicitamente o roteamento de veículos.
- 2) Veículo único com capacidade limitada (PAN *et al.*, 2009; GU, J.; GU, M.; GU, X., 2015; YAGMUR; KESSEN, 2020). Apenas o estudo de Yagmur e Kesen (2020) considera explicitamente o roteamento de veículos.

- 3) Número ilimitado de veículos com capacidade limitada por veículo (BASIR; MAZDEH; NAMAKSHENAS, 2018; NOROOZI *et al.*, 2019). Nenhum estudo deste grupo considera explicitamente o roteamento de veículos.
- 4) Número limitado de veículos com capacidade limitada por veículo (SCHOLZ-REITER *et al.*, 2011; MARANDI; ZEGORDI, 2017; RAMEZANIAN; MOHAMMADI; CHERAGHALIKHANI, 2017; MOHAMMADI; CHERAGHALIKHANI; RAMEZANIAN, 2018; MARANDI; GHOMI, 2019; MARTINS *et al.*, 2021). Todos os estudos deste grupo consideram explicitamente o roteamento de veículos.

A pesquisa bibliográfica revelou que são raros os estudos que modelam o PPIPD conjuntamente com o roteamento de veículos em ambientes *flow shop*. Nenhum estudo foi encontrado que modelasse e resolvesse o PPIPD-NWFS, seja no modelo DFC ou no modelo DVC.

A seguir são descritos os estudos classificados nos grupos 1 a 4 que foram considerados relevantes para esta Tese.

Pan *et al.* (2009) estudam o PPFS de duas máquinas integrado ao transporte das ordens (produtos) para um único cliente, considerando explicitamente o limite de capacidade e os tempos de transporte de um único veículo, objetivando minimizar os critérios C_{max} e TCT. Análises de complexidade computacional do problema são conduzidas para ambos os critérios. Os resultados demonstram a natureza não-determinística do problema (fortemente *NP-hard*) para ambos os critérios de otimização. Baseada nesta conclusão, os autores desenvolvem e avaliam algoritmos aproximados para resolverem o problema de otimização.

Gu, J., Gu, M. e Gu, X. (2015) desenvolvem e analisam o desempenho de um algoritmo baseado na metaheurística GA para o PPIPD em ambiente *flow shop* simples com o objetivo de encontrar um programa coordenado de produção-distribuição que minimize o critério C_{max} definido no estudo como sendo o tempo (máximo) de chegada da última ordem (produto) no cliente. O problema considera um sistema produtivo que atende apenas a um único cliente, utilizando-se de um único veículo com capacidade limitada. O algoritmo GA faz uso de uma 'estratégia

mutualista evolucionária' para gerar as populações de soluções. Um experimento computacional é conduzido para verificar a adequação de três variações do algoritmo (MQGA, MGA, QGA). Os resultados demonstram que o algoritmo atende satisfatoriamente ao seu propósito.

Wang *et al.* (2016) propõem e comparam os desempenhos de três algoritmos para solucionar o PPFS flexível de dois estágios, objetivando minimizar o tempo médio de chegada dos veículos no cliente (AAT), em um sistema *flow shop* com máquinas em paralelo no primeiro estágio, uma operação de montagem no segundo estágio, capacidade ilimitada por veículo, número ilimitado de veículos para carga imediata pós-produção e um único cliente. Dois dos algoritmos propostos são heurísticas simples (SPT e LPT) adaptadas para o contexto integrado. O terceiro algoritmo (HGA-OVNS) é baseado nas metaheurísticas GA, VNS e aprendizagem baseada em oposição (OBL) e usa as outras duas heurísticas simples. O desempenho do algoritmo HGA-OVNS é comparado com outros quatro métodos (GA, GA-VNS, LPT e SPT). Os resultados experimentais concluem estatisticamente que o algoritmo híbrido é superior aos demais em qualidade das soluções.

Ramezani, Mohammadi e Cheraghalikhani (2017) desenvolvem e avaliam um novo algoritmo baseado na estratégia competitiva imperialista (ICA) para o PPIPD em um ambiente *flow shop* simples com o objetivo de minimizar os custos de produção e transporte. Dois modelos de entrega são avaliados separadamente: entrega direta e com roteamento. Após a produção, as ordens devem ser entregues ou diretamente para cada cliente ou devem ser roteadas em lotes para os clientes, usando uma frota homogênea de veículos com restrição de capacidade. O método proposto (IIICA) é avaliado conjuntamente com outros seis métodos (Palmer, CDS, ODE, NEH, PGA e HDDE) usando instâncias de *benchmark*. Para o sistema de transporte, os pesquisadores propõem formas de geração dos dados complementares das instâncias (custos, distâncias, tempos, capacidades, número de veículos, dentre outros). Os pesquisadores conduzem experimentos para avaliar a performance dos algoritmos para os dois modelos de entrega. Os resultados experimentais concluem que o método proposto (IIICA) é competitivo e eficiente para a resolução de ambos os modelos e que o modelo integrado com roteamento de veículos é mais eficiente (menos custoso) do que o modelo de entrega direta.

Noroozi *et al.* (2019) propõem e analisam os desempenhos de algoritmos baseados na metaheurística GA para o PIPD em um ambiente *flow shop* flexível. Os autores se inspiraram em um caso real de um fabricante de laticínios que opera uma frota de veículos híbrida, parte própria e parte terceirizada (3PL). A empresa recebe grupos de ordens (pedidos) que podem ser aceitos ou não, dependendo do esforço de produção (custos) e lucro total projetado. As ordens aceitas possuem diferentes tamanhos e quando concluídas são transportadas em lotes para diferentes clientes por veículos com limitação de capacidade. A possibilidade de contratação de transportadores terceirizados determina a capacidade ilimitada da frota. O objetivo do problema é a maximização de lucros. Quatro variações do algoritmo GA são avaliadas. Os resultados experimentais concluem que a variante que incorpora um subalgoritmo de busca local e uma regra inicial supera as demais variantes.

Marandi e Ghomi (2019) desenvolvem e comparam os desempenhos de cinco algoritmos baseados nas metaheurísticas GA, ICA e otimização por enxame de partículas (*particle swarm optimization* – PSO) para o PIPD com o objetivo de minimizar a soma dos custos de atraso e transporte. O modelo integra múltiplas fábricas (modeladas como um sistema *flow shop* flexível) com um sistema de transporte onde uma frota de veículos de capacidade limitada é operada para entregar as ordens concluídas para os clientes. O último estágio do sistema é um depósito central que recebe as ordens concluídas das fábricas para serem enviadas aos clientes. As ordens ficam aguardando no depósito apenas o tempo necessário para que um veículo da frota esteja disponível para carregamento. Uma vez embarcadas, as ordens são roteadas para os clientes. Duas variantes do algoritmo ICA são estudadas, sendo uma delas (IICA) proposta na pesquisa. Outras duas variantes do algoritmo PSO são estudadas, sendo uma delas (IPSO) também proposta na pesquisa. Os quatro algoritmos são comparados entre si e com um algoritmo GA usando instâncias de testes criadas pelos pesquisadores. Os resultados experimentais concluem que a variante IICA (que inclui um subalgoritmo SA) supera as demais em qualidade do resultado, mas as custas do aumento significativo do tempo de processamento.

Yagmur e Kesen (2020) propõem e avaliam um algoritmo memético (evolucionário) para o PIPD em um ambiente *flow shop* simples onde uma fábrica produz e distribui

para clientes dispersos geograficamente. As ordens concluídas são entregues em lotes por apenas um único veículo de capacidade limitada que deve ser roteado. As ordens disponíveis para entrega, mas não embarcadas, devem aguardar até o retorno do veículo à fábrica para um novo carregamento e roteamento. O objetivo é minimizar a soma dos tempos de transporte e dos atrasos de entrega. O desempenho do algoritmo memético é comparado com o desempenho de um *solver* de mercado (IBM CPLEX). Os resultados experimentais demonstram que o algoritmo proposto obtém soluções próximas ou exatamente iguais às aquelas obtidas pelo *software* comercial para as instâncias e parâmetros de execução definidos pelos pesquisadores.

Martins *et al.* (2021) propõem e avaliam métodos heurísticos de solução baseados na metaheurística VND para a resolução do PPIPD objetivando minimizar a data máxima de entrega em um ambiente *flow shop* híbrido onde uma fábrica produz para atender, por meio de uma frota de veículos de capacidade limitada, um conjunto de clientes dispersos geograficamente. Os métodos são avaliados para diferentes estruturas de vizinhança e heurísticas de iniciação cujas variantes são proposta na pesquisa. Os pesquisadores propõem um modelo em PLIM para o problema e definem *lower bounds* que usados na análise da qualidade das soluções obtidas pelos métodos heurísticos. Experimentos computacionais são delineados para avaliar a solução do problema para diferentes instâncias de pequeno e grande portes geradas na pesquisa. O estudo ainda avalia diferentes estratégias de carregamento dos veículos e formação de lotes (sequências de entrega). Os experimentos demonstram que a estrutura de vizinhança, a heurística de iniciação, a estratégia de carregamento e a regra de formação do lote afetam conjuntamente a qualidade das soluções. A análise estatística destaca qual é a melhor combinação encontrada para estes fatores.

4. Métodos de Solução Propostos

Este capítulo inicia-se com a apresentação do modelo matemático proposto para a solução do PIPD-NWFS por métodos exatos. Em seguida, são descritos os métodos heurísticos que são o foco da pesquisa.

4.1. Modelagem matemática

Ao longo das últimas duas décadas, muitos pesquisadores propuseram diferentes modelos em programação matemática, notadamente a programação linear inteira mista (PLIM), para o PIPD em ambientes *flow shop*, considerando diferentes restrições e critérios de otimização. Os trabalhos de Scholz-Reiter *et al.* (2011), Marandi e Zegordi (2017), Ramezani, Mohammadi e Cheraghalikhani (2017), Mohammadi, Cheraghalikhani e Ramezani (2018) e Marandi e Ghomi (2019) são exemplos desta abordagem.

O modelo em PLIM proposto neste capítulo para o PIPD-NWFS tem como principal objetivo caracterizar os dois subproblemas estudados (PPNWFS e PRV) e identificar o vínculo que os integra em um único problema de otimização. Não é objetivo desta pesquisa comparar sistematicamente os resultados gerados por métodos exatos na resolução do modelo proposto em PLIM com aqueles gerados pelos métodos propostos.

Instâncias de pequeno porte do PIPD são costumeiramente resolvidas por métodos exatos tais como *branch-and-bound*, *branch-and-cut* e *branch-price-and-cut*, sendo muitos deles implementados por *solvers* de mercado como o Gurobi (GUROBI, 2020) e o CPLEX (IBM, 2020). Os artigos de Ehm e Freitag (2016), Marandi e Ghomi (2019) e Yagmur e Kesen (2020) descrevem o uso destes *solvers* na resolução do

PPIPD. Instâncias de médio e grande portes são resolvidas pelos métodos descritos na revisão bibliográfica e pelos métodos propostos neste capítulo.

A seguir, é apresentado o modelo matemático conjuntamente com a nomenclatura de seus índices, parâmetros de entrada e variáveis de decisão. O modelo considera em sua formulação ambas as estratégias de transporte (DFC e DVC).

Com o intuito de demonstrar a exequibilidade do modelo proposto, no Apêndice D, é apresentado o modelo codificado em *Gurobipy* (*Python* para Gurobi), a interface de programação de aplicações (*Application Programming Interface* – API) da Gurobi para modelagem matemática na linguagem *Python*. O modelo também foi codificado na linguagem GAMS e resolvido para a instância do exemplo do final deste capítulo e para a instância N20-M20 do Apêndice C. Os resultados das resoluções também são apresentados no Apêndice D.

Índices e conjuntos do modelo:

$j \in N = \{0, \dots, n\}$: ordem de produção do cliente j .

$k \in V = \{1, \dots, v\}$: veículo.

$\tau \in T = \{1, \dots, \rho\}$: evento de coleta.

Parâmetros de entrada do modelo:

D_{ij} : distância (atraso) mínima(o) entre as ordens i e j na sequência.

q_j : tamanho (ex.: peso, volume, quantidade) da ordem j .

s_j : tempo de serviço da ordem j .

Q_k : capacidade de transporte do veículo k .

t_{ij} : tempo de transporte entre os clientes i e j .

$FP_{\tau k}$: data fixa de coleta (ou saída) τ do veículo k .

B : um número inteiro suficientemente grande.

Variáveis de decisão do modelo:

y_{ij} : variável binária, define se a ordem i precede a ordem j na sequência.

z_{jk} : variável binária, define se o veículo k transporta a ordem j .

x_{ijk} : variável binária, define se o trecho (i, j) é atendido pelo veículo k .

$a_{\tau k}$: variável binária, define qual evento de coleta τ é escolhido pelo veículo k .

C_j : tempo de conclusão da ordem j .

VP_k : data variável de coleta (ou saída) do veículo k .

F_{jk} : tempo (de fluxo) até a entrega da ordem j pelo veículo k .

DT_{jk} : data de entrega da ordem j coletada pelo veículo k .

O principal critério de otimização usado pelo modelo é o TFT. Sua definição mais geral é dada como a soma dos tempos de fluxo F_j ou pela soma das datas de entrega DT_j subtraídas dos instantes de liberação r_j .

O critério TFT também é definido com a soma dos tempos de conclusão de produção C_j , espera por veículo W_j , transporte T_j e serviço S_j subtraídos dos instantes de liberação r_j . O valor de TFT deve ser minimizado. A Equação (4.1) apresenta a expressão matemática geral da função objetivo:

$$\min TFT = \sum_{j \in N} F_j = \sum_{j \in N} (DT_j - r_j) = \sum_{j \in N} (C_j + W_j + T_j + S_j - r_j) \quad (4.1)$$

O critério TFT foi escolhido como critério de otimização nesta pesquisa (LIU; REEVES, 2001; FERNANDEZ-VIAGAS; FRAMINAN, 2017):

- 1) Por ser considerado um dos critérios mais usados na análise de desempenho de sistemas *flow shop*.
- 2) Por ser bastante relevante e significativo nos ambientes de produção dinâmicos atuais.
- 3) Por estar diretamente relacionado a importantes indicadores de desempenho de logística e manufatura, tais como o nível de atendimento e o nível de estoque em processo.
- 4) Por fundamentar a hipótese formulada e verificada nesta pesquisa.

No modelo a seguir, a Equação (4.2) define a função objetivo como o somatório dos tempos de fluxo calculados para cada ordem j entregue pelo veículo k . Vale observar que nesta pesquisa os tempos de liberação $r_j = 0$.

$$\min TFT = \sum_{j \in N} \sum_{k \in V} F_{jk} = \sum_{j \in N} \sum_{k \in V} DT_{jk} \quad (4.2)$$

O conjunto de restrições (4.3) a (4.9) modelam o sistema produtivo *no-wait flow shop*. As restrições (4.3) a (4.5) definem o início e o término da sequência de produção usando a ordem fictícia '0'.

$$y_{0,0} = 0 \quad (4.3)$$

$$\sum_{i=1}^n y_{i0} \leq 1 \quad (4.4)$$

$$\sum_{j=1}^n y_{0j} \leq 1 \quad (4.5)$$

As restrições (4.6) e (4.7) estabelecem o controle do fluxo de produção. As restrições (4.6) definem que apenas uma ordem deve preceder a outra. As restrições de balanço de fluxo (4.7) definem que toda ordem iniciada deve ser concluída.

$$\sum_{i=1}^n y_{ij} = 1 \quad \forall j \in N \setminus \{0\} \quad (4.6)$$

$$\sum_{i=1}^n y_{ij} = \sum_{i=1}^n y_{ji} \quad \forall j \in N \setminus \{0\} \quad (4.7)$$

As restrições (4.8) e (4.9) calculam o tempo de conclusão de cada ordem. A ordem fictícia '0' tem tempo de conclusão igual a zero. Os tempos de conclusão das demais ordens são calculados pelas restrições (4.8) que também garantem que não ocorram *subtours* de produção, ou seja, que apenas um único programa (ou sequência) de produção seja criado(a) para o conjunto de ordens.

$$C_0 = 0 \quad (4.8)$$

$$C_i + D_{ij} - B \times (1 - y_{ij}) \leq C_j \quad \forall i, j \in N \setminus \{0\} \quad (4.9)$$

O conjunto de restrições (4.10) a (4.21) modelam o sistema de transporte para as estratégias DFC e DVC. O vínculo entre os modelos de produção e distribuição é feito pelas restrições (4.18) (modelo DVC) e (4.18B) (modelo DFC).

As restrições (4.10) garantem que apenas um veículo transporte uma determinada ordem.

$$\sum_{k=1}^v z_{jk} \leq 1 \quad \forall j \in N \setminus \{0\} \quad (4.10)$$

As restrições (4.11) garantem que o conjunto de ordens transportado por um veículo não ultrapasse sua capacidade.

$$\sum_{j=1}^n q_j \cdot z_{jk} \leq Q_k \quad \forall k \in V \quad (4.11)$$

As restrições (4.12) a (4.17) definem a rota de cada veículo. As restrições (4.12) a (4.15) garantem que cada veículo parta da fábrica '0' e retorne a ela ao término das entregas.

$$x_{ijk} = 0 \quad \forall i = j; i, j \in N; k \in V \quad (4.12)$$

$$\sum_{j=1}^n x_{0jk} \leq 1 \quad \forall k \in V \quad (4.13)$$

$$\sum_{i=1}^n x_{i0k} \leq 1 \quad \forall k \in V \quad (4.14)$$

$$\sum_{i=1}^n x_{ijk} = 1 \quad \forall j \in N \setminus \{0\}; k \in V \quad (4.15)$$

As restrições (4.16) garantem que cada cliente j seja visitado por um veículo.

$$\sum_{i=1}^n x_{ijk} \leq z_{jk} \quad \forall j \in N; k \in V \quad (4.16)$$

As restrições de balanço de fluxo (4.17) garantem que cada veículo que chegue em um cliente parta dele após a entrega.

$$\sum_{i=0}^n x_{ijk} = \sum_{i=1}^n x_{jik} \quad \forall j \in N \setminus \{0\}; k \in V \quad (4.17)$$

As restrições (4.18) a (4.21) calculam os tempos de fluxo de cada veículo. As restrições (4.20) e (4.21) garantem ainda que não ocorram *subtours* na rota de cada veículo.

$$C_j - B \times (1 - z_{jk}) \leq VP_k \quad \forall j \in N; k \in V \quad (4.18)$$

$$F_{0k} = 0 \quad \forall k \in V \quad (4.19)$$

$$F_{0k} + VP_k + t_{0j} + s_j - B \times (1 - x_{0jk}) \leq F_{jk} \quad \forall j \in N \setminus \{0\}; k \in V \quad (4.20)$$

$$F_{lk} + t_{ij} + s_j - B \times (1 - x_{ijk}) \leq F_{jk} \quad \forall i, j \in N \setminus \{0\}; k \in V \quad (4.21)$$

As restrições (4.22) a (4.26) definem os respectivos domínios das variáveis de decisão binárias e contínuas e o valor do parâmetro B .

$$\sum_{i=0}^n \sum_{j=0}^n D_{ij} + \sum_{i=0}^n \sum_{j=0}^n t_{ij} = B \quad (4.22)$$

$$x_{ijk} \in \{0,1\} \quad \forall i, j \in N; k \in V \quad (4.23)$$

$$y_{ij} \in \{0,1\} \quad \forall i, j \in N \quad (4.24)$$

$$z_{jk} \in \{0,1\} \quad \forall j \in N; k \in V \quad (4.25)$$

$$C_j, VP_k, F_{jk}, \in \mathbb{R}^+ \quad \forall j \in N; k \in V \quad (4.26)$$

O modelo matemático apresentado anteriormente atende exclusivamente à estratégia de transporte DVC.

Para atender à estratégia de transporte DFC, as restrições (4.18) e (4.20) devem ser substituídas, respectivamente, pelas restrições (4.18B) e (4.20B), e as restrições (4.17B) e (4.24B) devem ser incluídas.

As restrições (4.17B) garantem que apenas uma data de coleta seja selecionada para cada veículo.

$$\sum_{\tau=1}^{\rho} a_{\tau k} \leq 1 \quad \forall k \in V \quad (4.17B)$$

As restrições (4.18B) e (4.20B) calculam os tempos de fluxo de cada veículo, considerando agora as datas fixas de coleta.

$$C_j - B \times (1 - z_{jk}) \leq \sum_{\tau=1}^{\rho} (FP_{\tau k} \times a_{\tau k}) \quad \forall j \in N; k \in V \quad (4.18B)$$

$$F_{0k} + \sum_{\tau=1}^{\rho} (FP_{\tau k} \times a_{\tau k}) + t_{0j} + s_j - B \times (1 - x_{0jk}) \leq F_{jk} \quad \forall j \in N \setminus \{0\}; k \in V \quad (4.20B)$$

As restrições (4.24B) definem o domínio da variável binária de decisão.

$$a_{\tau k} \in \{0,1\} \quad \forall \tau \in T; k \in V \quad (4.24B)$$

4.2. A hipótese $\min \{ |N| \times C_{max} + \sum_{j \in N} T_j \} \leq \min \sum_{j \in N} (C_j + T_j)$

No modelo DFC, a data de coleta FP_j da ordem j é fixa (predeterminada) e selecionada com base no conjunto de valores discretos $FP_t \in T$, sendo a capacidade de transporte disponível em t a principal restrição para a seleção de FP_j . A data de entrega neste modelo é dada por $DT_j = FP_j + T_j + s_j$, onde T_j é o tempo de transporte e s_j é o tempo de serviço. Como FP_j e s_j não são variáveis de decisão (são parâmetros) e, por definição, $FP_j \geq \min (C_j + W_j)$, onde C_j é a data de conclusão e W_j é a espera para embarque, a função objetivo TFT pode ser formulada como $\min TFT = \min \sum_j DT_j = \min \sum_j T_j$. Portanto, a minimização ocorre efetivamente ‘pelo lado da distribuição’. A minimização ‘pelo lado da produção’ tem menor (ou quase nenhum) efeito devido a restrição da data fixa de coleta FP_j .

No modelo DVC, a data de coleta VP_j da ordem j é variável, sendo determinada pelo processo de otimização e fortemente dependente da capacidade de transporte, com $VP_j \in \mathbb{R}^+$. A data de coleta é formulada como $VP_j = C_j + W_j$ e a data de entrega é dada por $DT_j = VP_j + T_j + s_j = C_j + W_j + T_j + s_j$. Como s_j e W_j não são variáveis de decisão (s_j é um parâmetro e W_j é completamente determinada a partir de C_j), a função objetivo TFT pode ser formulada como:

$$\min TFT = \min \sum_j DT_j = \min \sum_j (VP_j + T_j) = \min \sum_j C_j + \min \sum_j T_j \quad (4.27)$$

Na Equação (4.27), a minimização ‘pelo lado da produção’ é dada por $\min \sum_j C_j$ e a minimização ‘pelo lado da distribuição’ é dada por $\min \sum_j T_j$. Isto permite afirmar que o instante de embarque de um lote de ordens em um veículo (determinado pela

variável $VP_j = VP_{j,k} = VP_k$) estabelece uma divisão no modelo integrado e cria dois submodelos que podem ser resolvidos de forma quase independente (veja as restrições (17) do modelo em PLIM).

O contexto dado pelo modelo DVC abre a possibilidade de conjecturar se para atender a o objetivo principal de minimização do tempo total de fluxo ($\min TFT$) a solução final deva ser obtida pela minimização do *makespan* ($\min C_{max}$), em vez de $\min \sum_j C_j$, ‘pelo lado da produção’, mantendo-se a minimização do tempo total de transporte ($\min \sum_j T_j$) ‘pelo lado da distribuição’.

Formalmente, para o objetivo $\min TFT$ em um SIPD e para um conjunto N de ordens embarcadas na mesma data VP_k , a hipótese pode ser formulada como:

$$\min \left\{ |N| \times C_{max} + \sum_{j \in N} T_j \right\} \leq \min \left\{ \sum_{j \in N} (C_j + T_j) \right\} \quad (4.28)$$

Os métodos heurísticos propostos nesta pesquisa foram modificados para considerarem na otimização ambas as expressões matemáticas da Inequação (4.28) e, com isto, possibilitar a verificação da hipótese pela comparação de resultados.

No Capítulo 5, são delineados os experimentos para a verificação da hipótese. A análise dos resultados descritos no Capítulo 6 e Apêndice A evidenciou que a hipótese mostrou ser verdadeira para a quase totalidade dos casos (instâncias) simulados. Uma confirmação da hipótese foi obtida também pelos resultados apresentados no Apêndice D para o modelo matemático proposto. No exemplo de aplicação do final do capítulo, a hipótese é verificada.

4.3. Estrutura algorítmica geral

Os métodos heurísticos de programação desenvolvidos nesta pesquisa são fundamentados no seguinte raciocínio: a melhoria na coordenação e sincronização

das operações de produção e distribuição leva à minimização dos tempos de fluxo e atrasos de entrega e, conseqüentemente, ao aumento do nível de serviço ao mercado.

Este raciocínio pressupõe que se conseguisse concluir a produção das ordens no exato momento em que os veículos (de uma frota bem dimensionada) se encontrassem disponíveis para carregamento e pronta entrega, sem a espera por falta de produto para embarcar e sem a geração de estoques por ausência de veículo, o sistema atingiria um nível de fluência de materiais que ocasionaria o atendimento otimizado dos pedidos e a redução de custos logísticos.

Para alcançarem este objetivo, os métodos heurísticos propostos implementam, em uma única estrutura algorítmica, procedimentos iterativos de programação baseados nas metaheurísticas IG, ILS, SA e VNS para solucionar o PPIPD-NWFS, considerando as restrições relacionadas ao sistema de produção NWFS, à ordem e aos modelos (estratégias) de distribuição DFC e DVC.

Dois métodos de solução são propostos e avaliados. O primeiro método é baseado na metaheurística híbrida ILS-SA-VNS ou ILS_{SA-VNS} . O segundo método é baseado na metaheurística híbrida IG-SA ou IG_{SA} . Ainda, para o método IG_{SA} , duas variantes são avaliadas: IG_{SA}^1 e IG_{SA}^2 , definindo cada qual um método distinto.

Os métodos heurísticos seguem a estrutura geral (*framework*) proposta por Framinan, Gupta e Leisten (2004) para a construção de heurísticas.

As estruturas algorítmicas gerais dos métodos possuem cinco fases principais. Na Figura 4.1, são apresentadas as estruturas gerais.

Os métodos realizam as seguintes tarefas agrupadas em duas macrofases de execução:

- 1) **Geração da solução inicial.** Execução das heurísticas construtivas (fase 1), realizando as seguintes tarefas:
 - a) programação da produção NWFS;
 - b) alocação de carga (ordens) aos veículos;

c) roteamento e programação dos veículos.

2) **Melhoria da solução.** Execução iterativa das metaheurísticas ILS_{SA-VNS} e IG_{SA} (fases 2 a 5), realizando as seguintes tarefas:

- a) reprogramação da produção NWFS;
- b) realocação de carga (ordens) aos veículos;
- c) roteamento e reprogramação dos veículos;
- d) perturbação da rota e reprogramação dos veículos.

É importante observar que as tarefas das macrofases 1 e 2 são executadas dentro da mesma estrutura algorítmica que resolve de forma integrada ambos os (sub)problemas de otimização (PPNWFS e PRV). As cinco fases de execução dos métodos são descritas resumidamente a seguir.

Pré-processamento. Esta ‘pré fase’ é realizada da mesma maneira por ambos os métodos. Anteriormente à execução da primeira fase é realizado o cálculo da matriz de ‘distâncias’ (ou atrasos) mínimas(os) entre ordens para atender a restrição *no-wait*.

Fase 1. Geração da solução inicial. Esta fase é realizada da mesma maneira por ambos os métodos. As soluções iniciais são geradas por meio de diferentes heurísticas construtivas: NEH, XNEH, CHINS, XCHINS, HC1 e HC2.

Fase 2. Busca Local (ILS_{SA-VNS}) e Destruição (IG_{SA}). Nesta fase, o método ILS_{SA-VNS} implementa um subalgoritmo de busca local simples para melhorar a solução inicial. O método IG_{SA} implementa um subalgoritmo de destruição, selecionando aleatoriamente parte das ordens da solução inicial para serem retiradas. O subconjunto de ordens retiradas será reinserido posteriormente à solução parcial para melhorar a solução.

Fase 3. Perturbação (ILS_{SA-VNS}) e Construção (IG_{SA}). Nesta fase, o método ILS_{SA-VNS} implementa um subalgoritmo SA. O método IG_{SA} implementa duas estratégias de inserção de ordens baseadas na heurística NEH para reconstruir a solução parcial. Cada estratégia é executada por uma versão (variante) do algoritmo IG_{SA} (IG_{SA}^1 e IG_{SA}^2) cujo desempenho é avaliado nesta pesquisa.

Figura 4.1 – Estruturas gerais dos métodos heurísticos propostos

Fonte: o autor

Fase 4. Busca Local (ILS_{SA-VNS}) e Aceitação (IG_{SA}). Nesta fase, o método ILS_{SA-VNS} implementa um subalgoritmo VNS de busca local, explorando três estruturas de vizinhança: troca de pares distantes (*swap*), troca de pares adjacentes (*pairwise exchange*) e inserção. O método IG_{SA} implementa um subalgoritmo SA.

Fase 5. Perturbação (ILS_{SA-VNS} e IG_{SA}). Nesta fase, ambos os métodos implementam a estratégia de inserção da heurística NEH para reavaliar as rotas criadas na fase anterior. As rotas são reavaliadas a partir dos veículos já carregados e ‘fechados’. Portanto, apenas a sequência de entrega das ordens já embarcadas em cada veículo é refeita na tentativa de melhorar a solução.

As expressões matemáticas (4.29) a (4.36) são utilizadas em todas as fases para o cálculo iterativo dos critérios de otimização TFT e C_{max} , bem como para a verificação da restrição de capacidade veicular (veja nomenclatura na Seção 4.1).

Nas (in)equações, $\pi_{[i]} \in \pi = \{\pi_{[1]}, \dots, \pi_{[n]}\}$ é a posição i da ordem $\pi_{[i]}$ no programa π ; π^p e π^k correspondem aos programas de produção p e do veículo k :

$$\min TFT = \sum_{j \in N} \sum_{k \in V} F_{jk} = \sum_{j \in N} \sum_{k \in V} DT_{jk} = \sum_{j \in N} \sum_{k \in V} (C_j + W_j + T_{jk} + S_{jk}) \quad (4.29)$$

$$\min C_{max} = \max_{j \in N} (C_j) \quad (4.30)$$

$$C_j = C_{\pi_{[h+1]}^p = j} = \sum_{i=0}^h D_{\pi_{[i]}^p, \pi_{[i+1]}^p} \quad \forall j \in N \quad (4.31)$$

$$W_j = VP_k - C_j \quad \forall j \in N; k \in V \quad (4.32)$$

$$W_j = FP_{\tau k} - C_j \quad \forall j \in N; k \in V \quad (4.33)$$

$$T_{jk} = T_{\pi_{[h+1]}^k = j} = \sum_{i=0}^h t_{\pi_{[i]}^k, \pi_{[i+1]}^k} \quad \forall j \in N; k \in V \quad (4.34)$$

$$S_{jk} = S_{\pi_{[h]}^k = j} = \sum_{i=1}^h s_{\pi_{[i]}^k} \quad \forall j \in N; k \in V \quad (4.35)$$

$$\sum_{i=1}^h q_{\pi_{[i]}^k} \leq Q_k \quad \forall k \in V \quad (4.36)$$

A descrição detalhada de cada uma das fases é feita a seguir.

4.4. Pré-processamento

Anteriormente à fase de iniciação de cada método, é gerada a matriz D de ‘distâncias’ (ou atrasos) mínimas(os) entre ordens a qual define as restrições de dependência temporal entre o término de execução da ordem J_{j-1} e o término de execução da ordem J_j , garantindo que cada ordem uma vez iniciada seja processada até seu término sem nenhuma espera em fila entre as máquinas. A matriz é construída utilizando a Equação (4.37):

$$(D_{j-1,j})_{(n+1) \times n} = \left[\max_i \left(\sum_{k=i}^m p_{j,k} - \sum_{k=i+1}^m p_{j-1,k} \right) \right]_{(n+1) \times n} \quad (4.37)$$

Li, Wang e Wu (2008) e Ye, Li e Miao (2016) descrevem a derivação da Equação (4.37) e apresentam exemplos de cálculo da matriz.

4.5. Fase 1 – Iniciação

Nesta fase, seis heurísticas construtivas são selecionadas para gerar a solução inicial: NEH, XNEH, CHINS, XCHINS, HC1 e HC2. Os métodos NEH e XNEH usam regras de prioridade na sua iniciação. As demais heurísticas não necessitam ser iniciadas. As regras de prioridade e heurísticas construtivas são descritas a seguir.

4.5.1. Regras de prioridade

As seguintes regras de prioridade (*sequencing rules*) são utilizadas na fase de iniciação dos métodos NEH e XNEH:

- 1) Tempo de processamento mais curto (*shortest processing time* - SPT). Sequencia as ordens com base no tempo de processamento na máquina, ordenando do menor para o maior valor e executando a programação nesta ordem. Regra formulada pela equação $\min p_j = \sum_{i \in M} p_{ji}$.
- 2) Tempo de processamento mais longo (*longest processing time* - LPT). Sequencia as ordens com base nos tempos de processamento na máquina, ordenando do maior para o menor valor e executando a programação nesta ordem. Regra formulada pela equação $\max p_j = \sum_{i \in M} p_{ji}$.

As regras SPT e LPT foram escolhidas porque (NAWAZ *et al.*, 1984; HAUPT, 1989; FRAMINAN; LEISTEN; RAJENDRAN, 2003; BAKER; TRIETSCH, 2009):

- 1) Elas são consideradas regras clássicas da área de sequenciamento e programação de operações.
- 2) Suas expressões matemáticas são simples.
- 3) Elas são costumeiramente usadas em estudos de análise de desempenho de sistemas *flow shop*.
- 4) A regra LPT é usada na iniciação da versão original da heurística NEH que provou ser uma das melhores heurísticas para a resolução do PPFS com o objetivo de minimizar o C_{max} .
- 5) A regra SPT gera a sequência ótima de ordens para o problema de programação em máquina única ($r_j = 0$) com o objetivo de minimizar o TFT.
- 6) A hipótese analisada nesta pesquisa propõe a minimização dos critérios C_{max} e TFT.

4.5.2. Heurística NEH

A heurística NEH (NAWAZ *et al.*, 1984) é usada para a geração da solução inicial. A sequência de passos da heurística NEH é descrita resumidamente a seguir.

- I. Para cada ordem $j = \{1, 2, \dots, n\}$ calcule o valor da regra de prioridade específica (SPT ou LPT)

- II. Determine a sequência de ordens pelos valores calculados (crescente para SPT e decrescente para LPT), criando a lista $L = \{J_{L_1}, J_{L_2}, \dots, J_{L_n}\}$.
- III. Selecione as duas primeiras ordens da sequência definida em $L (J_{L_1}, J_{L_2})$. Em seguida, encontre a melhor sequência parcial $\pi^*(i|i = 2) = (J_{\pi_1^*}, J_{\pi_2^*})$ para elas considerando o melhor (menor) valor do critério de otimização $C(\pi)$. Neste passo, podem existir apenas duas permutações ou sequências possíveis: as permutações (J_{L_1}, J_{L_2}) ou (J_{L_2}, J_{L_1}) . Uma vez encontrado o valor mínimo, atualize o programa parcial atual $\pi^*(2)$. Não altere a posição relativa das duas ordens nas próximas iterações do algoritmo. Retire da lista L as ordens programadas em $\pi^*(2)$.
- IV. Tomando como partida o programa parcial $\pi^*(2) = (J_{\pi_1^*}, J_{\pi_2^*})$ gerado em III, proceda iterativamente selecionando e programando a próxima ordem j da lista L (posição J_{L_3}) e assim sucessivamente até a última ordem J_{L_n} , buscando encontrar a melhor posição para a ordem J_{L_i} no programa parcial $\pi(i|i = 3, \dots)$, de maneira a obter a posição $p^* = \arg \min_{p=1, \dots, i} \pi(i)$, ou seja, buscando encontrar uma sequência parcial $\pi^*(i) = (J_{\pi_1^*}, J_{\pi_2^*}, \dots, J_{\pi_i^*})$ que minimize o critério de otimização adotado, isto é $C(\pi^*(i)) < C(\pi(i)), \forall \pi^* \neq \pi$, mantendo sempre a posição relativa entre as ordens já programadas na(s) iteração(ões) anterior(es).
- V. Execute o passo IV até que todas as ordens da lista L estejam programadas ($|L| = \{\phi\}$) e $\pi^* = \pi^*(i|i = n) = (J_{\pi_1^*}, J_{\pi_2^*}, \dots, J_{\pi_n^*})$.

4.5.3. Heurística XNEH

A heurística XNEH (*eXtended* NEH), proposta nesta pesquisa, é uma extensão da heurística NEH (NAWAZ *et al.*, 1984) na qual é adicionada uma fase de reinserção de ordens após a construção da solução completa. Ela é usada para a geração da solução inicial. A sequência de passos da heurística XNEH é descrita resumidamente a seguir.

A. Fase 1 – Inserção. Os passos realizados nesta fase são idênticos aos realizados pela heurística NEH.

B. Fase 2 – Reinserção.

- I. Tomando como partida o programa completo $\pi^* = (J_{\pi_1^*}, J_{\pi_2^*}, \dots, J_{\pi_n^*})$, selecione a ordem π_1^* da posição original $p = 1$ e a reinsira em todas as posições i à direita de π_1^* na sequência de π^* , com $p < i \leq n$, buscando encontrar uma posição i^* que otimize $C(\pi(n, i))$, ou seja, buscando encontrar uma nova sequência $\pi'(n) = (J_{\pi_2'}, \dots, J_{\pi_{i^*}'}, \dots, J_{\pi_n'})$ que minimize o critério de otimização adotado, isto é $C(\pi^*(n)) = \min_{i=2, \dots, n} C(\pi(n, i))$. Fixe π_1^* na posição i^* . Caso não encontre a posição i^* , retorne a ordem π_1^* para a sua posição original.
- II. Repita o passo I iterativamente para as demais posições originais, com $1 < p \leq n$, reinserindo a ordem π_p nas $n - p$ posições à direita, buscando encontrar uma posição i^* que resulte em $C(\pi^*(n)) = \min_{i=p, \dots, n} C(\pi(n, i))$.

4.5.4. Heurística CHINS

A heurística CHINS (*CHeapest INSsertion*) (BIANCO; GIORDANI, 1999; FINK; VOß, 2003) é usada para a geração da solução inicial. A sequência de passos da heurística CHINS é descrita resumidamente a seguir.

- I. Defina a lista inicial de ordens a ser programada $L = \{J_{L_1}, J_{L_2}, \dots, J_{L_n}\}$, com $|L| = N$. Esta lista não necessita ser ordenada por nenhum critério.
- II. Selecione a primeira ordem de L . Insira J_{L_1} na primeira posição π_1 do programa parcial $\pi(p = 1) = \pi(J_{\pi_1}) = \pi(J_{L_1})$. Calcule o valor do critério de otimização $C(\pi(p = 1, i = 1))$. Substitua J_{L_1} por J_{L_i} ($i = 2, \dots, n$) e calcule o valor de $C(\pi(p, i))$ para J_{L_i} . Selecione a ordem com o melhor (menor) valor $C(\pi^*(1)) = \min_{i=1, \dots, n} C(\pi(1, i))$ e fixe-a na posição π_1^* . Retire da lista L a ordem $J_{L_i} = J_{\pi_1^*}$.

- III. Tomando como partida o programa parcial $\pi^*(p = 1) = (J_{\pi_1^*})$, repita o passo II iterativamente para $1 < p \leq n$. Em cada iteração p , insira nas posições π_1 a π_p as $n - p$ ordens remanescentes da lista L , uma por vez, buscando encontrar uma ordem J_{L_i} que uma vez inserida em alguma posição π_s ($s = 1, \dots, p$) otimize $C(\pi(p, i))$, ou seja, buscando encontrar uma sequência $\pi^*(p) = (J_{\pi_1^*}, \dots, J_{\pi_s^*}, \dots, J_{\pi_p^*})$ que minimize o critério de otimização adotado, isto é $C(\pi^*(p)) = \min_{i=1, \dots, n} \{ \min_{s=1, \dots, p} [C(\pi(s, i))] \}$, mantendo as posições relativas entre as ordens programadas na(s) iteração(ões) anterior(es). Fixe a ordem J_{L_i} na posição π_s e retire-a da lista L .
- IV. Execute o passo III até que todas as ordens da lista L estejam programadas ($|L| = \{\phi\}$) e $\pi^* = \pi^*(p|p = n) = (J_{\pi_1^*}, J_{\pi_2^*}, \dots, J_{\pi_n^*})$.

4.5.5. Heurística XCHINS

A heurística XCHINS (*eXtended CHepeast INSsertion*), proposta nesta pesquisa, e uma extensão da heurística CHINS (BIANCO; GIORDANI, 1999; FINK; VOß, 2003) na qual é adicionada uma fase de reinserção de ordens após o término da construção da solução completa. A sequência de passos da heurística XCHINS é descrita resumidamente a seguir.

A. Fase 1 – Inserção. Os passos realizados nesta fase são idênticos aos realizados pela heurística CHINS.

B. Fase 2 – Reinserção.

- I. Tomando como partida o programa completo $\pi^* = (J_{\pi_1^*}, J_{\pi_2^*}, \dots, J_{\pi_n^*})$, selecione a ordem π_1^* da posição original $p = 1$ e a reinsira em todas as posições i à direita de π_1^* na sequência de π^* , com $p < i \leq n$, buscando encontrar uma posição i^* que otimize $C(\pi(n, i))$, ou seja, buscando encontrar uma nova sequência $\pi'(n) = (J_{\pi_2'}, \dots, J_{\pi_{i^*}'}, \dots, J_{\pi_n'})$ que minimize o critério de

otimização adotado, isto é $C(\pi^*(n)) = \min_{i=2,\dots,n} C(\pi(n, i))$. Fixe π_1^* na posição i^* .

Caso não encontre a posição i^* , retorne a ordem π_1^* para a sua posição original.

- II. Repita o passo I iterativamente para as demais posições originais, com $1 < p \leq n$, reinserindo a ordem π_p nas $n - p$ posições à direita, buscando encontrar uma posição i^* que resulte em $C(\pi^*(n)) = \min_{i=p,\dots,n} C(\pi(n, i))$.

4.5.6. Heurística HC1

A heurística construtiva HC1, proposta nesta pesquisa, usa a estratégia de inserção da heurística BIH (BIANCO; GIORDANI, 1999) associada a uma estratégia de reinserção de ordens após o término da construção da solução completa. Ela é usada para a geração da solução inicial. HC1 é semelhante a heurística XCHINS, diferindo desta no passo III da fase de inserção. A sequência de passos da HC1 é descrita resumidamente a seguir.

A. Fase 1 – Inserção.

- I. Defina a lista inicial $L = \{J_{L_1}, J_{L_2}, \dots, J_{L_n}\}$ de ordens a serem programadas, com $|L| = N$. Esta lista não necessita ser ordenada por nenhum critério.
- II. Selecione a primeira ordem de L . Insira J_{L_1} na primeira posição π_1 do programa parcial $\pi(p = 1) = \pi(J_{\pi_1}) = \pi(J_{L_1})$. Calcule o valor do critério de otimização $C(\pi(p = 1, i = 1))$. Substitua J_{L_1} por J_{L_i} ($i = 2, \dots, n$) em π_1 e calcule o valor de $C(\pi(p, i))$ para J_{L_i} . Selecione a ordem com o melhor (menor) valor $C(\pi^*(p)) = \min_{i=1,\dots,n} C(\pi(p, i))$ e fixe-a na posição π_1^* . Retire da lista L a ordem $J_{L_i} = J_{\pi_1^*}$.
- III. Tomando como partida o programa parcial $\pi^*(p = 1) = (J_{\pi_1^*})$, repita o passo II iterativamente para $1 < p \leq n$. Em cada iteração p , inserir na posição π_p e somente nesta posição, as $n - p$ ordens remanescentes da lista L , uma por vez, buscando encontrar uma ordem J_{L_i} que otimize $C(\pi(p, i))$, ou seja,

buscando encontrar uma sequência $\pi^*(p) = (J_{\pi_1^*}, \dots, J_{\pi_p^*}) = (J_{\pi_1^*}, \dots, J_{L_i})$ que minimize o critério de otimização adotado, isto é $C(\pi^*(p)) = \min_{i=1, \dots, n} C(\pi(p, i))$, mantendo sempre fixas as posições das ordens já programadas na(s) iteração(ões) anterior(es). Fixe a ordem J_{L_i} na posição π_p e retire-a da lista L .

IV. Execute o passo III até que todas as ordens da lista L estejam programadas ($|L| = \{\phi\}$) e $\pi^* = \pi^*(p|p = n) = (J_{\pi_1^*}, J_{\pi_2^*}, \dots, J_{\pi_n^*})$.

B. Fase 2 – Reinserção. Os passos realizados nesta fase são idênticos aos realizados pela heurística XCHINS.

4.5.7. Heurística HC2

A heurística construtiva HC2, proposta nesta pesquisa, também usa a estratégia de inserção da heurística BAH (BIANCO; GIORDANI, 1999) associada a uma estratégia de reinserção de ordens após o término da construção da solução completa. Ela é usada para a geração da solução inicial. Esta heurística é similar a HC1 e XCHINS. Mas difere das demais nos passos III e IV. A sequência de passos da HC2 é descrita resumidamente a seguir.

A. Fase 1 – Inserção.

- I. Defina a lista inicial de ordens a ser programada $L = \{J_{L_1}, J_{L_2}, \dots, J_{L_n}\}$, com $|L| = N$. Esta lista não necessita ser ordenada por nenhum critério.
- II. Selecione a primeira ordem de L . Insira J_{L_1} na primeira posição π_1 do programa parcial $\pi(p = 1) = \pi(J_{\pi_1}) = \pi(J_{L_1})$. Calcule o valor do critério de otimização $C(\pi(p = 1, i = 1))$. Substitua J_{L_1} por J_{L_i} ($i = 2, \dots, n$) e calcule o valor de $C(\pi(p, i))$ para J_{L_i} . Selecione a ordem com o melhor (menor) valor $C(\pi^*(1)) = \min_{i=1, \dots, n} C(\pi(1, i))$ e fixe-a na posição π_1^* . Retire da lista L a ordem $J_{L_k} = J_{\pi_1^*}$.

- III. Tomando como partida o programa parcial $\pi^*(p = 1) = (J_{\pi_1^*})$, repita o passo II iterativamente para $1 < p \leq n$. Em cada iteração p , insira na posição π_p e somente nesta posição, as $n - p$ ordens remanescentes da lista L , uma por vez, buscando encontrar uma ordem J_{L_i} que otimize $C(\pi(p, i))$, ou seja, buscando encontrar uma sequência $\pi^*(p) = (J_{\pi_1^*}, \dots, J_{\pi_p^*})$ que minimize o critério de otimização adotado, isto é $C(\pi^*(p)) = \min_{i=1, \dots, n} C(\pi(p, i))$, mantendo sempre fixas as posições das ordens já programadas na(s) iteração(ões) anterior(es).
- IV. Insira a ordem vencedora $J_{L_i} = J_{\pi_p^*}$ nas demais posições π_s ($s = 1, \dots, p - 1$), buscando encontrar uma sequência $\pi^*(p) = (J_{\pi_1^*}, \dots, J_{\pi_s^*}, \dots, J_{\pi_p^*})$ que minimize o critério de otimização adotado, isto é $C(\pi^*(p)) = \min_{s=1, \dots, p} C(\pi(p, s))$, mantendo as posições relativas entre as ordens programadas na(s) iteração(ões) anterior(es). Fixe a ordem J_{L_i} na posição π_s e retire-a da lista L .
- V. Execute os passos III e IV até que todas as ordens da lista L estejam programadas ($|L| = \{\phi\}$) e $\pi^* = \pi^*(p|p = n) = (J_{\pi_1^*}, J_{\pi_2^*}, \dots, J_{\pi_n^*})$.

B. Fase 2 – Reinserção. Os passos realizados nesta fase são idênticos aos realizados pelas heurísticas XCHINS e HC1.

4.6. Fase 2 – Busca local e destruição

4.6.1. Busca local

Este procedimento somente é realizado pelo método ILS_{SA-VNS} . A melhor solução π^* encontrada na fase anterior é utilizada como solução inicial desta fase.

O método implementa um subalgoritmo de busca local simples que realiza a seleção aleatória – baseada em uma distribuição uniforme de probabilidades (d.u.p.)

no intervalo $[0,1]$ – de um ou mais pares de ordens distintas para mudarem de posição por meio do procedimento de inserção de ordens. Para cada par selecionado, a ordem com maior valor atribuído pela d.u.p. é inserida na posição ocupada pela ordem com o segundo maior valor. O número de pares selecionados é definido por meio de um parâmetro de execução do algoritmo. A escolha do valor do parâmetro é discutida no Capítulo 5.

Figura 4.2 – Procedimento de inserção de ordens

Na Figura 4.2, é apresentado um exemplo de inserção para as ordens dos pares (2,4) e (6,8). As ordens 4 e 8 são selecionadas para serem inseridas nas posições ocupadas pelas ordens 2 e 6 respectivamente.

4.6.2. Destruição

Este procedimento somente é realizado pelo método IG_{SA} . A melhor solução π^* encontrada na fase anterior é utilizada como solução inicial desta fase.

O procedimento de destruição seleciona $d_n < n$ (com $d = f(n)$) ordens da sequência inicial para remoção, inserindo-as na lista L_D .

A seleção das ordens é aleatória baseada em uma distribuição uniforme de probabilidades (d.u.p.) no intervalo $[0,1]$. A sequência resultante é dada por $\pi_R(n - d_n) = \pi^* - L_D(d_n)$. O valor do parâmetro d_n depende do número de ordens de π^* e sua escolha é discutida no Capítulo 5.

4.7. Fase 3 – Perturbação e construção

4.7.1. Perturbação

Este procedimento somente é realizado pelo método ILS_{SA-VNS} . A melhor solução π^* encontrada na fase anterior é utilizada como solução inicial desta fase.

Nesta fase, o método implementa um subalgoritmo baseado na metaheurística SA. O subalgoritmo SA não seleciona necessariamente a melhor solução na vizinhança $N(\pi'(i - 1))$ da melhor solução obtida na iteração anterior $(i - 1)$. Se o valor do critério de otimização obtido na iteração i atual $C(\pi(i))$ for menor do que aquele obtido na iteração anterior $i-1$, ou seja, $C(\pi(i)) < C(\pi'(i - 1))$, a solução $\pi(i)$ se torna a solução atual $\pi'(i)$. Caso contrário, se $C(\pi(i)) \geq C(\pi'(i - 1))$, uma das duas alternativas seguintes é selecionada de acordo com uma distribuição de probabilidades $P(i)$: a) a pior solução $\pi(i)$ torna-se a solução atual ($\pi'(i)$) com probabilidade $P(i)$ ou b) $\pi'(i - 1)$ torna-se a solução atual com probabilidade $1 - P(i)$.

O subalgoritmo SA é descrito resumidamente a seguir (baseado em PIRLOT, 1996):

- I. Para a solução inicial π' e correspondente $C(\pi')$ gerada na fase anterior, inicie a melhor solução fazendo $\pi^* \leftarrow \pi'$ e o melhor valor do critério de otimização fazendo $C(\pi^*) \leftarrow C(\pi')$.
- II. Para $i = \{1, 2, \dots\}$, $\pi'(i - 1)$ e $\pi'(i)$ denotam as soluções nas iterações $i-1$ e i respectivamente. Selecione aleatoriamente (via d.u.p. no intervalo $[0,1]$) ordens para mudança de posição. Usando a heurística NEH-SPT, reinsira as ordens

selecionadas e gere uma solução $\pi(i)$ na vizinhança $N(\pi'(i-1))$. Calcule o valor $C(\pi(i))$ e avalie-o.

a. Se $C(\pi(i)) < C(\pi'(i-1))$, então, $\pi'(i) \leftarrow \pi(i)$.

b. Se $C(\pi(i)) < C(\pi^*)$, então, $C(\pi^*) \leftarrow C(\pi(i))$ e $\pi^*(i) \leftarrow \pi(i)$.

III. Caso contrário, se $[C(\pi(i)) - C(\pi'(i-1))]/C(\pi'(i-1)) \leq \delta$ então:

a. Selecione aleatoriamente um número p de $[0,1]$.

b. Se $p \leq P(i)$, então $\pi'(i) \leftarrow \pi(i)$.

c. Caso contrário, $\pi'(i) \leftarrow \pi'(i-1)$.

IV. Atualize a iteração para $i+1$. Se o critério de parada (tempo de execução) for alcançado, então finalize o subalgoritmo. Caso contrário, execute os passos II e III novamente.

Para tornar o subalgoritmo SA operacional, os seguintes parâmetros e funções são definidos (a escolha dos valores de cada parâmetro é discutida no Capítulo 5):

- 1) O fator de filtragem δ , com $0 < \delta < 1$;
- 2) A distribuição de probabilidades $P(i)$ é dada por $P(i) = e^{\left(\frac{-\Delta C(\pi(i))}{T(t)}\right)}$, com $\Delta C(\pi(i)) = C(\pi(i)) - C(\pi'(i-1))$ e $C(\pi(i)) > C(\pi'(i-1))$;
- 3) A temperatura de recozimento $T(t)$ é dada pela função de decaimento exponencial $T(t) = T_0 e^{-\alpha t}$, onde:
 - a. t é o instante atual de execução do algoritmo SA;
 - b. L_{SA} é o intervalo de tempo até a mudança de temperatura;
 - c. $T(t) < T(t-1)$, se $t \bmod L_{SA} = 0$, caso contrário $T(t) = T(t-1)$;
 - d. T_0 é a temperatura inicial de recozimento;
 - e. α é a taxa de resfriamento, com $\alpha = -\ln\left(1/T_0\right) \cdot \frac{1}{t_{SA}}$;
 - f. t_{SA} é o instante final (tempo total) de execução do algoritmo SA;
 - g. $T(t_{SA})$ é a temperatura final de recozimento, com $T(t_{SA}) = 1$.

4.7.2. Construção

Este procedimento somente é realizado pelo algoritmo IG_{SA} . A sequência $\pi_R(n - d_n)$ gerada na fase de destruição é utilizada como solução parcial inicial desta fase e π^* é usada como solução completa inicial.

A (re)construção da solução completa é realizada por meio de dois procedimentos que diferenciam as versões (variantes) IG_{SA}^1 e IG_{SA}^2 avaliadas nesta pesquisa.

1) Procedimento 1 (apenas IG_{SA}^1):

- I. Ordene $L_D(d_n)$ usando a regra LPT.
- II. Reinsira as ordens em $\pi_R(n - d_n)$ usando a heurística NEH:
 - a. $\pi' = \pi_R \leftarrow L_D$.
- III. Se $C(\pi') < C(\pi^*)$, então $\pi^* \leftarrow \pi'$.
- IV. Se o critério de parada (tempo de execução) for alcançado então pare, caso contrário, execute a Fase 2 (destruição).

2) Procedimento 2 (apenas IG_{SA}^2):

- I. Separe o conjunto $L_D(d_n)$ em três subconjuntos $L_{D_i}(d_n)$, com $i = \{1,2,3\}$ e $L_{D_1} < L_{D_2} < L_{D_3}$.
- II. Ordene $L_{D_i}(d_n)$ usando a regra SPT, gerando as sequencias π_{D_i} .
- III. Inicie o programa $\pi_{D_0} = \{\emptyset\}$.
- IV. Reinsira as ordens em quatro etapas usando a heurística NEH da seguinte maneira:
 - a. $\pi_{D_0} \leftarrow \pi_{D_1}$;
 - b. $\pi_{D_{12}} = \pi_{D_0} \leftarrow \pi_{D_2}$;
 - c. $\pi_{D_{123}} = \pi_{D_{12}} \leftarrow \pi_{D_3}$;
 - d. $\pi' = \pi_{D_{123}} \leftarrow \pi_R$.
- V. Se $C(\pi') < C(\pi^*)$, então $\pi^* \leftarrow \pi'$.
- VI. Se o critério de parada (tempo de execução) for alcançado, então pare, caso contrário, execute a Fase 2 (destruição).

A escolha dos valores dos parâmetros L_{D_i} e do critério de parada (tempo de execução) são discutidos no Capítulo 5.

O procedimento realizado por IG_{SA}^1 foi proposto inicialmente por Ruiz e Stützle (2007) e utilizado por outros pesquisadores em diversos trabalhos subsequentes. O procedimento de inserções sucessivas realizado por IG_{SA}^2 é proposto nesta Tese e resultou, na maior parte das vezes, em soluções melhores que aquelas geradas pelo procedimento realizado por IG_{SA}^1 .

4.8. Fase 4 – Busca local e aceitação

4.8.1. Busca Local

Este procedimento somente é realizado pelo método ILS_{SA-VNS} . A melhor solução π^* encontrada na fase anterior é usada como solução inicial desta fase.

Nesta fase, um subalgoritmo VNS realiza busca local diversificada usando estruturas de vizinhas distintas.

Diferente da Fase 2 na qual é usado apenas o procedimento de inserção simples de ordens, nesta fase, o subalgoritmo implementa três procedimentos de geração de estruturas de vizinhança N_v : inserção ($v = 1$), troca de pares distantes ou *swap* ($v = 2$) e troca de pares adjacentes ou *pairwise exchange* ($v = 3$).

A ordem de seleção de cada estrutura de vizinhança e a regra de mudança de uma estrutura para outra são predefinidas por parametrização e são discutidas no Capítulo 5.

Na geração das estruturas de vizinhança N_v , o subalgoritmo VNS realiza a seleção aleatória – baseada em uma distribuição uniforme de probabilidades (d.u.p.) no

intervalo $[0,1]$ – de um ou mais pares de ordens distintas para serem mudados de posição.

Para cada par selecionado, a ordem com maior valor atribuído pela d.u.p. troca a sua posição (seja por inserção, *swap* ou troca adjacente) com a ordem com o segundo maior valor. As ordens selecionadas são fixadas na sequência e não poderão ser selecionadas novamente durante a construção da vizinhança (e, em seguida, pelo procedimento de busca local).

O número de pares selecionados para a criação da vizinhança é definido por parametrização. A escolha do valor deste parâmetro é discutida no Capítulo 5. Na Figura 4.3, são apresentadas ilustrações dos três procedimentos de geração das estruturas de vizinhança.

Figura 4.3 – Procedimentos de geração de vizinhança

Fonte: o autor

Após a geração da vizinhança, o subalgoritmo VNS implementa uma busca local simples para melhorar a solução. A sequência de passos do subalgoritmo VNS é descrita resumidamente a seguir (baseado em MLADENOVIC; HANSEN, 1997):

- I. Selecione os procedimentos de geração das vizinhanças N_v , com $v \in \{1,2,3\}$.
- II. Iniciação: faça $\pi' \leftarrow \pi^*$, $v \leftarrow 1$ e gere a vizinhança $N_1(\pi')$.
- III. Para $i = \{1, 2, \dots\}$, $\pi'(i-1)$ e $\pi'(i)$ denotam as soluções nas iterações $i-1$ e i respectivamente. Gere uma solução $\pi(i)$ na vizinhança $N_v(\pi'(i-1))$.
- IV. Execute busca local simples para melhorar a solução $\pi(i)$; calcule $C(\pi(i))$.
- V. Se $C(\pi(i)) < C(\pi'(i-1))$, então, $\pi'(i) \leftarrow \pi(i)$.
- VI. Caso contrário, faça $v \leftarrow v + 1$ e gere uma nova vizinhança $N_v(\pi'(i-1))$.
- VII. Atualize $i \leftarrow i + 1$. Se o critério de parada (tempo de execução) for alcançado, faça $\pi^* \leftarrow \pi'(i)$ e finalize o subalgoritmo VNS. Caso contrário, execute os passos III a VI novamente.

4.8.2. Aceitação

Este procedimento somente é realizado pelo algoritmo IG_{SA} . A melhor solução π^* encontrada na fase anterior é utilizada como solução inicial desta fase.

Nesta fase, o método IG_{SA} implementa um subalgoritmo SA para perturbar π^* ($i = 1$) ou $\pi'(i-1 > 0)$ e gerar uma solução candidata $\pi(i)$ que será avaliada pelo procedimento de aceitação. Caso seja melhor que $\pi'(i-1)$, ela será aceita como a nova solução $\pi'(i)$. Caso contrário, será aceita de maneira probabilística.

O subalgoritmo SA executado nesta fase tem as mesmas características daquele executado na Fase 3 (perturbação) do método ILS_{SA-VNS} . Os parâmetros de execução são discutidos no Capítulo 5.

4.9. Fase 5 – Perturbação da rota

Este procedimento é realizado por ambos os métodos ILS_{SA-VNS} e IG_{SA} . A melhor solução π^* encontrada na fase anterior é utilizada como solução inicial desta fase.

Na tentativa de melhorar a solução obtida até então, os métodos propostos executam a heurística NEH para reavaliar apenas as rotas criadas na fase anterior.

Inicialmente, a cada ordem é atribuído um valor gerado por uma distribuição uniforme de probabilidades (d.u.p.) no intervalo $[0,1]$. Em seguida, as ordens são ordenadas de maneira decrescente de valores e um programa de entrega $\pi(i)$ é criado para cada veículo usando a estratégia de inserção da heurística NEH. Caso a solução $\pi(i)$ seja melhor que aquela da iteração anterior $\pi(i-1)$, ou seja $C(\pi(i)) < C(\pi'(i-1))$, ela passa a ser a solução atual $\pi'(i)$. Caso contrário, uma nova iteração é executada, e assim sucessivamente, até que o critério de parada (tempo de execução) seja alcançado.

O subalgoritmo de perturbação de rota é executado para cada veículo tomado isoladamente sem alterar as ordens já alocadas ao veículo, isto é, apenas as rotas são reavaliadas, partindo-se dos veículos já carregados e ‘fechados’. Portanto, as ordens já carregadas em um veículo não são transferidas para outro e o programa de produção criado na fase anterior é mantido inalterado.

4.10. Um exemplo de aplicação

Nesta seção, é apresentado um exemplo de aplicação dos métodos propostos, usando os dados da instância do exemplo de Li, Wang e Wu (2008). Além de ilustrar o funcionamento dos métodos propostos, o exemplo também serve para demonstrar a hipótese proposta no início deste capítulo.

No exemplo original, seis ordens devem ser programadas para produção em cinco máquinas em um ambiente NWFS. No exemplo desta seção, as seis ordens devem ser programadas para a produção em cinco máquinas e para o transporte subsequente para os respectivos clientes. Assume-se que cada ordem se originou de um cliente específico, portanto, seis clientes devem ser atendidos. O modelo de distribuição é o DVC e o critério de otimização usado é (apenas) o TFT.

Na Tabela 4.1, são apresentados os dados da instância do exemplo. Eles correspondem aos tempos de execução das ordens nas máquinas. A linha 'Total' mostra o tempo de processamento de cada ordem.

Tabela 4.1 – Tempos de processamento

	J ₁	J ₂	J ₃	J ₄	J ₅	J ₆
M ₁	72	83	11	89	44	5
M ₂	68	83	90	7	62	27
M ₃	9	31	74	57	41	82
M ₄	1	66	72	37	13	24
M ₅	48	90	36	31	22	13
Total	198	353	283	221	182	151

Na Tabela 4.2, é apresentada a matriz de 'distâncias' calculada usando a Equação (37). A linha '0' mostra as 'distâncias' $D_{0,j}$. Li, Wang e Wu (2008) apresentam o cálculo da matriz do exemplo.

Neste exemplo, os tempos de transporte entre os clientes foram determinados transpondo-se a matriz de 'distâncias' (atrasos), ou seja, $D_{i,j} = t_{j,i}$, $i \neq j$ e $i \neq 0$. Os tempos de transporte da fábrica diretamente para os clientes imediatos da rota foram definidos como $t_{0,j} = D_{0,j}$, correspondendo a linha '0' da Tabela 4.2.

Tabela 4.2 – Distâncias

	J ₁	J ₂	J ₃	J ₄	J ₅	J ₆
'0'	198	353	283	221	182	151
J ₁	-	227	214	95	80	88
J ₂	48	-	85	31	22	13
J ₃	48	120	-	32	22	13
J ₄	66	221	151	-	50	51
J ₅	60	215	196	90	-	84
J ₆	52	207	153	88	39	-

Por exemplo, com base na Tabela 4.2, os valores $t_{3,2} = D_{2,3} = 85$, $t_{4,5} = D_{5,4} = 90$, $t_{2,1} = D_{1,2} = 227$ e $t_{0,3} = D_{0,3} = 283$.

Na Tabela 4.3, são apresentados os tamanhos de ordem (necessários à verificação de capacidade do veículo) e tempos de serviço de carregamento e descarregamento das ordens (necessários aos cálculos dos tempos de atendimento das ordens).

A capacidade do veículo foi calculada para ser igual a 1778. Os dados da Tabela 4.3 e a capacidade do veículo foram calculados (com arredondamentos) usando as fórmulas (5.1) a (5.8) descritas no Capítulo 5.

Tabela 4.3 – Atributos da ordem

J _i	Tamanho	Tempo Serviço
1	217	39
2	388	70
3	311	56
4	243	44
5	200	36
6	166	30

Para evidenciar o processo de melhoria contínua das soluções geradas pelos métodos propostos, são apresentadas as soluções obtidas nas macrofases de geração da solução inicial, usando a heurística NEH-LPT, e de melhoria, usando a metaheurística IG_{SA} . O mesmo resultado foi obtido pela metaheurística ILS_{SA-VNS} .

Para um melhor entendimento da aplicação dos métodos, os resultados são apresentados separadamente na forma de um ‘programa de produção’ e de um ‘programa de entrega’. Mas como ficará evidenciado ao longo da explicação do exemplo, ambos os programas são gerados simultaneamente.

Nas Tabelas 4.4, 4.5 e 4.6, são apresentados os programas de produção gerados, respectivamente, pela regra LPT, heurística NEH e metaheurística IG_{SA} .

Tabela 4.4 – Programa de produção (LPT)

Seq	J_j	Tempo Proc	Distância	Data Conclusão	Espera	Data Coleta
1	2	353	353	353	347	700
2	3	283	85	438	262	700
3	4	221	32	470	230	700
4	1	198	66	536	164	700
5	5	182	80	616	84	700
6	6	151	84	<u>700</u>	0	700

Nestas três tabelas, na coluna ‘Seq’, é apresentada a sequência de execução da ordem J_j . Na coluna ‘Tempo Proc’, é apresentado o tempo de processamento da ordem. Na coluna ‘Distância’, é apresentado o valor $D_{i,j}$.

Na coluna ‘Data Conclusão’, é apresentada a data programada de término da ordem. Esta coluna é calculada somando-se a ‘distância’ de uma determinada linha da tabela à data de conclusão da linha anterior. O valor do *makespan* é mostrado em negrito e grifado.

Na coluna ‘Espera’, é mostrado o tempo de espera até a coleta ou saída do veículo. O valor desta coluna é calculado subtraindo-se do *makespan* a data de conclusão de cada ordem. No modelo DVC, a ‘Data de Coleta’ é coincidente com o *makespan*.

Tabela 4.5 – Programa de produção (NEH)

Seq	J _j	Tempo Proc	Distância	Data Conclusão	Espera	Data Coleta
1	6	151	151	151	465	616
2	5	182	39	190	426	616
3	1	198	60	250	366	616
4	4	221	95	345	271	616
5	3	283	151	496	120	616
6	2	353	120	616	0	616

Nas Tabelas 4.7, 4.8 e 4.9, são apresentados os programas de entrega gerados, respectivamente, pela regra LPT, heurística NEH e metaheurística IG_{SA} antes da execução do subalgoritmo de perturbação da rota.

Tabela 4.6 – Programa de produção final (IG_{SA})

Seq	J _j	Tempo Proc	Distância	Data Conclusão	Espera	Data Coleta
1	6	151	151	151	448	599
2	5	182	39	190	409	599
3	4	221	90	280	319	599
4	3	283	151	431	168	599
5	2	353	120	551	48	599
6	1	198	48	599	0	599

Nestas três tabelas, na coluna ‘Veículo’, é apresentado o veículo alocado à ordem. Por ter capacidade suficiente para atender a todo o programa de entrega, apenas um

veículo é alocado. Na coluna 'Cap Liq', é mostrada a capacidade líquida (residual) após cada ordem ser embarcada no veículo.

Tabela 4.7 – Programa de entrega antes da perturbação de rota (LPT)

Seq	J _j	Veículo	Data Coleta	Serviço Entrega	Tempo Viagem	Data Entrega	Tempo Fluxo	Cap Liq
1	2	1	700	70	353	1123	1123	1390
2	3	1	700	126	473	1299	2422	1079
3	4	1	700	170	624	1494	3916	836
4	1	1	700	209	719	1628	5544	619
5	5	1	700	245	779	1724	7268	419
6	6	1	700	275	818	1793	<u>9061</u>	253

Na coluna 'Tempo Viagem', é apresentado o tempo estimado (acumulado a partir da saída do veículo) para cada entrega programada. Os valores desta coluna são calculados com base na matriz de tempos de transporte (a matriz da Tabela 4.2 transposta). O tempo de viagem de uma determinada linha da tabela é calculado somando-se o tempo de transporte ao valor do tempo de viagem da linha anterior.

Tabela 4.8 – Programa de entrega antes da perturbação de rota (NEH)

Seq	J _j	Veículo	Data Coleta	Serviço Entrega	Tempo Viagem	Data Entrega	Tempo Fluxo	Cap Liq
1	6	1	616	30	151	797	797	1612
2	5	1	616	66	235	917	1714	1412
3	1	1	616	105	315	1036	2750	1195
4	4	1	616	149	381	1146	3896	952
5	3	1	616	205	413	1234	5130	641
6	2	1	616	275	498	1389	<u>6519</u>	253

Na coluna ‘Data Coleta’, é apresentada a data de saída do veículo. Na coluna ‘Serviço Entrega’, é mostrado o tempo estimado (acumulado a partir da saída do veículo) dos serviços (embarque e desembarque) a serem realizados para cada entrega. O tempo de serviço é calculado com base nos dados da Tabela 4.3.

Tabela 4.9 – Programa de entrega antes da perturbação de rota (IG_{SA})

Seq	J _j	Veículo	Data Coleta	Serviço Entrega	Tempo Viagem	Data Entrega	Tempo Fluxo	Cap Liq
1	6	1	599	30	151	780	780	1612
2	5	1	599	66	235	900	1680	1412
3	4	1	599	110	285	994	2674	1169
4	3	1	599	166	317	1082	3756	858
5	2	1	599	236	402	1237	4993	470
6	1	1	599	275	629	1503	6496	253

Na coluna ‘Data Entrega’, é mostrada a data de entrega estimada. Ela é calculada pela soma das colunas ‘Data Coleta’, ‘Serviço Entrega’ e ‘Tempo Viagem’.

Na coluna ‘Tempo Fluxo’, é apresentado o valor acumulado (estimado) do tempo de fluxo em cada entrega. Esta coluna é calculada somando-se a data da entrega de uma determinada linha ao valor do tempo de fluxo da linha anterior. O valor total do tempo de fluxo para este programa de entrega é dado em negrito e grifado.

Na Tabela 4.10, é apresentado o programa de entrega após a execução do subalgoritmo de perturbação da rota o que levou a uma redução do tempo total de fluxo de 3,17% para 6290 (em negrito e grifado na tabela).

A perturbação da rota não alterou a programação da produção, apenas redefiniu a rota, alterando a sequência {6,5,4,3,2,1} para {6,4,1,5, 3,2} na entrega.

Tabela 4.10 – Programa de entrega após a perturbação de rota (IG_{SA})

Seq	J _j	Veículo	Data Coleta	Serviço Entrega	Tempo Viagem	Data Entrega	Tempo Fluxo	Cap Liq
1	6	1	599	30	151	780	780	1612
2	4	1	599	74	202	875	1655	1369
3	1	1	599	113	297	1009	2664	1152
4	5	1	599	149	357	1105	3769	952
5	3	1	599	205	379	1183	4952	641
6	2	1	599	275	464	1338	6290	253

Se a nova sequência de entrega fosse também executada na produção, o *makespan* (data de coleta) alcançaria o valor de 701 o que representaria um aumento (piora) da solução de 17,03% sobre o valor anterior (599) e um aumento (piora) de 9,73% sobre solução final (de 6290 para 6902). Na Tabela 4.11, é apresentado o resumo dos resultados do exemplo.

Tabela 4.11 – Resumo dos tempos de fluxo totais (TFT)

	LPT	NEH	IG_{SA} antes	IG_{SA} após
Data Coleta	700	616	599	599
Tempo Fluxo	9061	6519	6496	6290

4.11. Verificação da hipótese

No exemplo original de Li, Wang e Wu (2008), o algoritmo proposto pelos autores para o critério C_{max} solucionou exatamente (ótimo global) o problema, obtendo como solução final a sequência {6,3,2,4,5,1} e valor $C_{max} = 565$ (Tabela 4.12), o que é menor (melhor) que aquele obtido pelo método IG_{SA} para o programa de produção ($C_{max} = 599$) na solução final do PIPD-NWFS-DVC, usando somente o critério TFT.

Se a sequência {6,3,2,4,5,1} for usada para programar também a distribuição, o resultado obtido (6607) será pior que o resultado obtido para o problema integrado, mesmo antes da fase de perturbação da rota. O resultado é apresentado na Tabela 4.13.

Tabela 4.12 – Programa de produção usando C_{max}

Seq	J_j	Tempo Proc	Distância	Data Conclusão	Espera	Data Coleta
1	6	151	151	151	414	565
2	3	283	153	304	261	565
3	2	353	120	424	141	565
4	4	221	31	455	110	565
5	5	182	50	505	60	565
6	1	198	60	<u>565</u>	0	565

Entretanto, se os algoritmos propostos resolverem o PPIPD-NWFS-DVC, buscando minimizar o critério C_{max} na geração do programa inicial completo (com as operações de produção e distribuição) e, em seguida, melhorá-lo por meio do algoritmo de perturbação de rota, utilizando o critério TFT, o resultado obtido (Tabela 4.14) é melhor do que aquele apresentado na Tabela 4.10 que foi obtido usando apenas o critério TFT. Na verdade, o resultado apresentado na Tabela 4.14 é a solução exata (ótimo global) para o problema. Este resultado evidencia, portanto, a hipótese e é confirmado pelos resultados do modelo matemático do Apêndice D.

Este exemplo demonstra o fato de que a programação integrada usando os critérios C_{max} e TFT pode levar (e na verdade leva na maior parte das vezes) a resultados melhores do que a programação integrada usando apenas o critério TFT em um SIPD onde uma frota de veículos com restrição de capacidade é gerida pelo embarcador e o desempenho do sistema é mensurado globalmente pelo critério TFT.

Tabela 4.13 – Programa de entrega usando C_{max}

Seq	J _j	Veículo	Data Coleta	Serviço Entrega	Tempo Viagem	Data Entrega	Tempo Fluxo	Cap Liq
1	6	1	565	30	151	746	746	1612
2	3	1	565	86	164	815	1561	1301
3	2	1	565	156	249	970	2531	913
4	4	1	565	200	470	1235	3766	670
5	5	1	565	236	560	1361	5127	470
6	1	1	565	275	640	1480	<u>6607</u>	253

Vale notar que o tempo total de fluxo (soma da coluna ‘Data Conclusão’) obtido a partir da Tabela 4.6 (2202) para o programa de produção (PP1) gerado usando o critério TFT é menor do que aquele obtido pelo programa de produção (PP2) da Tabela 4.12 (2404) que usou inicialmente o critério C_{max} para a programação integrada e, posteriormente, o TFT na perturbação da rota.

Apesar disso, devido ao critério C_{max} , a data de coleta calculada para PP2 (565) é menor do que a data de coleta de PP1 (599), com diferença de $599 - 565 = 34$ unidades de tempo (u.t.) por ordem entregue e $6 \times 34 = 204$ u.t. no total.

Tabela 4.14 – Programa de entrega após a perturbação de rota (IG_{SA})

Seq	J _j	Veículo	Data Coleta	Serviço Entrega	Tempo Viagem	Data Entrega	Tempo Fluxo	Cap Liq
1	6	1	565	30	151	746	746	1612
2	4	1	565	74	202	841	1587	1369
3	1	1	565	113	297	975	2562	1152
4	5	1	565	149	357	1071	3633	952
5	3	1	565	205	379	1149	4782	641
6	2	1	565	275	464	1304	<u>6086</u>	253

Como a data de coleta é igual para todas as ordens, a diferença de 34 u.t. é evidenciada em cada uma das datas de entrega. Com uma data de coleta menor, o programa de entrega (PE2) gerado a partir de PP2 (Tabela 4.14) apresenta uma solução (6086) que é 204 u.t. (3,24%) menor do que a solução (6290) apresentada pelo programa de entrega (PE1) gerado a partir de PP1 (Tabela 4.10).

No Apêndice C, são apresentados exemplos de resultados de execuções do algoritmo $IG_{SA}^2 - XCHINS$ para uma instância de Taillard (1983) de tamanho N20-M20 para ambos os modelos DFC e DVC. Os mesmos conceitos descritos nesta seção são usados para explicar os resultados dos exemplos do apêndice.

5. Delineamento dos Experimentos Computacionais

Neste capítulo, é descrito o procedimento experimental elaborado para comparar os desempenhos dos métodos heurísticos propostos para diferentes configurações do SIPD-NWFS.

Os principais objetivos do procedimento experimental foram:

- 1) Avaliar o quanto cada um dos três métodos propostos melhorou a solução inicial gerada por cada uma das heurísticas construtivas usadas como *benchmark* (ex.: IG_{SA}^1 versus NEH; IG_{SA}^2 versus CHINS; ILS_{SA-VNS} versus HC1).
- 2) Comparar a qualidade das soluções geradas pelos três métodos propostos entre si para a mesma heurística de iniciação (ex.: $IG_{SA}^1 - NEH$ versus $IG_{SA}^2 - NEH$; $IG_{SA}^2 - HC2$ versus $ILS_{SA-VNS} - HC2$).
- 3) Comparar a qualidade das soluções geradas pelos três métodos propostos entre si independente da heurística de iniciação adotada (ex.: ILS_{SA-VNS} versus IG_{SA}^2 ; IG_{SA}^1 versus IG_{SA}^2).
- 4) Comparar a qualidade da solução gerada pela combinação de um método e uma heurística com outras combinações de métodos e heurísticas (ex.: $IG_{SA}^1 - NEH$ versus $IG_{SA}^2 - CHINS$; $ILS_{SA-VNS} - HC2$ versus $IG_{SA}^2 - XCHINS$).
- 5) Verificar experimentalmente a hipótese proposta nesta pesquisa.

O procedimento experimental foi dividido em cinco fases:

Fase 1. Elaboração do plano experimental.

Fase 2. Escolha das instâncias do problema e parâmetros de execução.

Fase 3. Codificação e execução dos algoritmos computacionais.

Fase 4. Análise estatística dos resultados.

Fase 5. Conclusões e proposições.

As fases 1 a 3 são descritas neste capítulo. As fases 4 e 5 são descritas nos Capítulos 6 e 7 respectivamente.

5.1. Plano experimental

O plano experimental foi concebido para investigar o impacto de três fatores no desempenho do SIPD-NWFS para os modelos DFC e DVC. Ainda, com o objetivo de verificar a hipótese proposta no Capítulo 4, para ambos os modelos foram testadas duas ‘abordagens de otimização’:

- 1) A primeira usou o critério C_{max} para programar inicialmente o sistema integrado (fases 1 a 4). Em seguida, o critério TFT foi usado pelo subalgoritmo de perturbação de rota (fase 5) para melhorar a solução. Para esta abordagem, os modelos de distribuição receberam as denominações de DFC1 e DVC1.
- 2) A segunda usou apenas o critério TFT para programar o sistema integrado (fases 1 a 5). Para esta abordagem, os modelos de distribuição receberam as denominações de DFC2 e DVC2.

O plano experimental seguiu as diretrizes estabelecidas por Montgomery (2017), e foi composto (veja Tabela 5.3 para o resumo):

- 1) dos fatores e respectivos níveis experimentais, cujas interações supostamente afetavam (influenciavam) o resultado da variável de resposta,
- 2) dos tratamentos (combinações entre níveis experimentais) e
- 3) do número de replicações e execuções dos métodos.

A variável de resposta escolhida (critério de otimização) para medir o desempenho global do sistema foi o tempo total de fluxo (TFT).

Quatro conjuntos de experimentos foram delineados, dois deles tendo o objetivo de avaliar os desempenhos dos métodos na solução do PPIPD-NWFS para as abordagens DFC1 e DFC2, e os outros dois tendo o objetivo de avaliar os desempenhos dos métodos na solução do PPIPD-NWFS para as abordagens DVC1 e DVC2.

Os fatores experimentais selecionados foram: a) tamanho da instância (fator 1); b) metaheurística de programação (fator 2); c) heurística de iniciação (fator 3).

Para o fator 1 – tamanho da instância – foram escolhidos onze níveis experimentais. Cada nível foi definido pelo número de ordens (n) e máquinas (m) de cada instância. As instâncias foram selecionadas do conjunto de instâncias de *benchmark* de Taillard (1993a, 1993b) para o problema de programação *flow shop*. Na Tabela 5.1, são apresentados os níveis experimentais do fator 1 e o respectivo grupo ao qual cada instância foi classificada de acordo com o seu tamanho.

Tabela 5.1 – Níveis do fator 1

Nível	No. Ordens (n)	No. Máquinas (m)	Grupo da Instância
1	20	5	Pequena
2	20	10	Pequena
3	20	20	Pequena
4	50	5	Média
5	50	10	Média
6	50	20	Média
7	100	5	Grande
8	100	10	Grande
9	100	20	Grande
10	200	10	Grande
11	200	20	Grande

Para o fator 2 – metaheurística de programação – foram escolhidos três níveis experimentais: IG_{SA}^1 (nível 1), IG_{SA}^2 (nível 2), ILS_{SA-VNS} (nível 3).

Para o fator 3 – heurística de iniciação – foram escolhidos seis níveis experimentais: NEH (nível 1), XNEH (nível 2), CHINS (nível 3), XCHINS (nível 4), HC1 (nível 5) e HC2 (nível 6). Na Tabela 5.2, são apresentados os fatores experimentais e seus níveis para ambos os problemas.

Cada conjunto de experimentos foi constituído de onze experimentos determinados com base nos níveis do fator 1.

Cada tratamento foi definido como a combinação (multiplicação) dos níveis de cada fator para cada um dos dois problemas e abordagens de otimização. Com base nisto, o número total de tratamentos foi calculado como: Problema $\times$ Abordagem $\times$ Níveis Fator 1 $\times$ Níveis Fator 2 $\times$ Níveis Fator 3. Como todos os níveis deveriam combinar, o número total resultou em $2 \times 2 \times 11 \times 3 \times 6 = 792$ tratamentos.

Tabela 5.2 – Fatores e níveis experimentais

Fator 1 (Instância)	Fator 2 (Metaheurística)	Fator 3 (Heurística Iniciação)
20 $\times$ 5	IG_{SA}^1 IG_{SA}^2 ILS_{SA-VNS}	NEH XNEH CHINS XCHINS HC1 HC2
20 $\times$ 10		
20 $\times$ 20		
50 $\times$ 5		
50 $\times$ 10		
50 $\times$ 20		
100 $\times$ 5		
100 $\times$ 10		
100 $\times$ 20		
200 $\times$ 10		
200 $\times$ 20		

Para cada tratamento, cinco réplicas (instâncias) foram selecionadas, totalizando 3.960 replicações. Deste total, 1.980 replicações corresponderam aos experimentos do problema PIPD-NWFS-DFC (990 para cada abordagem) e as outras 1.980 corresponderam aos experimentos do problema PIPD-NWFS-DVC (990 para cada abordagem).

Pelo motivo dos métodos heurísticos serem algoritmos estocásticos, cada réplica (instância) foi executada 3 vezes e foi coletado o melhor resultado obtido dentre as três execuções. Na Tabela 5.3, é apresentado o resumo do plano experimental.

Tabela 5.3 – Plano experimental

Tabela 6.6 - Plano Experimental				
Problema e Abordagem	Níveis do Tamanho Instância	Níveis do Método Principal	Níveis do Método Iniciação	Número de Replicações por Tratamento
4	11	3	6	5
Número Total de Tratamentos				
792				
Número Total de Replicações Por Problema Estudado (ambas as abordagens)				
1.980		1.980		
Número Total de Replicações				
3.960				
Número Total de Execuções dos Métodos (3 por réplica)				
11.880				

A análise estatística dos resultados seguiu os procedimentos descritos em Ruiz e Allahverdi (2007, 2009), nas respectivas seções de avaliação de resultados computacionais, e em Montgomery (2017) e é detalhada no Capítulo 6.

5.2. Parâmetros de execução

As metaheurísticas estudadas nesta pesquisa possuem seus conjuntos de parâmetros de execução cujos valores podem ser a) definidos e fixados anteriormente à execução do algoritmo ou b) ajustados dinamicamente ao longo de cada execução.

Como os valores desses parâmetros influenciam a qualidade dos resultados e nos desempenhos dos algoritmos, se faz necessária a avaliação e discussão dos valores selecionados. Nesta pesquisa, tais valores foram a) determinados com base em

experimentações iniciais de ‘calibração’ dos métodos ou b) selecionados de estudos conduzidos por outros pesquisadores cujos trabalhos serviram de base para o procedimento experimental.

Alguns dos estudos que serviram como fontes de conhecimento para a elaboração dos experimentos preliminares (simulações) e para a obtenção das fórmulas e dos valores dos parâmetros de execução foram Fernandez-Viagas e Framinan (2014, 2019) e Dubois-Lacoste, Pagnozzi e Stützle (2017). A seguir é feita uma discussão sobre como os valores dos principais parâmetros de execução foram determinados.

5.2.1. Metaheurística IG_{SA}

A seguir são descritos os procedimentos usados para a escolha dos principais parâmetros da metaheurística IG_{SA} .

- 1) O tempo máximo t_{max} de execução do método IG_{SA} foi definido pela fórmula $t_{max} = n \times m \times f_n \times w$ (em milissegundos), onde n é o número de ordens, m é o número de máquinas e f_n e w são fatores de ajuste. Os valores de $f_n = \{15, 30, 60, 90, 120, 240\}$ foram avaliados em diferentes execuções do método e para diferentes tamanhos de instância. Os valores finais foram escolhidos de acordo com o modelo de distribuição, sendo: $f_{n<100} = 15$, $f_{n=100} = 30$ e $f_{n=200} = 90$ para o modelo DVC e $f_{n<100} = 60$, $f_{n=100} = 120$ e $f_{n=200} = 240$ para o modelo DFC. Os valores de w foram definidos de acordo com o subalgoritmo executado. Para os subalgoritmos de destruição-construção e SA, $w = 1$; para o subalgoritmo de perturbação de rota, $w = 1/10$.
- 2) O número de ordens extraídas da solução na fase de destruição (fase 2) foi definido como $d_n = n/f_n$, onde n é o número de ordens e f_n é um fator de ajuste. Os valores de $f_n = \{2, 4, 8\}$ foram avaliados em diferentes execuções da metaheurística e para os diferentes tamanhos de instância. Os valores finais escolhidos para a variante IG_{SA}^1 foram $f_{n<100} = 4$ e $f_{n \geq 100} = 8$. Para a variante IG_{SA}^2 , $f_n = f = 2$ para todas as instâncias.

- 3) Na fase de reconstrução da solução (fase 3), para a variante IG_{SA}^1 , foi utilizada a heurística NEH com a regra LPT. Para a variante IG_{SA}^2 , foi utilizada a heurística NEH com a regra SPT.
- 4) Na fase de reconstrução da solução (fase 3), somente para a variante IG_{SA}^2 , antes da execução da heurística NEH para a reinserção das ordens, ocorre a divisão de $L_D(d_n)$ em três subconjuntos $L_{D_i}(d_n)$, definidos pelas quantidades d_n^1 , d_n^2 e d_n^3 . Diferentes valores de d_n^i foram avaliados em várias execuções para diferentes tamanhos de instância. Os valores finais selecionados são apresentados na Tabela 5.4.

Tabela 5.4 – Tamanhos dos subconjuntos de inserção

n	d_n^1	d_n^2	d_n^3
20	2	3	5
50	4	8	13
100	10	15	25
200	20	30	50

- 5) Tanto para a variante IG_{SA}^1 quanto para IG_{SA}^2 , para as estratégias DFC1 e DCV1 e nas fases diferentes da fase de ‘perturbação de rota’ (fase 5), a regra de seleção e desempate do valor de $C(\pi')$ adotada para todas as iterações das metaheurísticas foi a mesma e definida como segue: a) primeiramente, avaliava-se o critério C_{max} , selecionando o menor valor; b) caso os valores fossem iguais, avaliava-se o critério TFT, selecionando o menor valor; c) caso os valores fossem ainda iguais, avaliava-se o critério MFT, selecionando o menor valor; d) caso ainda fossem iguais, usava-se a menor posição relativa das ordens na lista de avaliação.
- 6) Tanto para a variante IG_{SA}^1 quanto para IG_{SA}^2 , para as estratégias DFC1 e DVC1 e somente para a fase de ‘perturbação de rota’ (fase 5), a regra de seleção e desempate do valor de $C(\pi')$ adotada para todas as iterações das metaheurísticas foi a mesma e definida como segue: a) primeiramente, avaliava-se o critério de desempenho principal (TFT), selecionando o menor valor; b) caso os valores fossem iguais, avaliava-se o critério MFT,

- selecionando o menor valor; c) caso os valores fossem ainda iguais, avaliava-se o critério C_{max} , selecionando o menor valor; d) caso ainda fossem iguais, usava-se a menor posição relativa das ordens na lista de avaliação.
- 7) Tanto para a variante IG_{SA}^1 quanto para IG_{SA}^2 , para as estratégias DFC2 e DVC2, independente da fase do método de solução, a regra de seleção e desempate do valor de $C(\pi')$ adotada para todas as iterações das metaheurísticas foi a mesma e definida como segue: a) primeiramente, avaliava-se o critério de desempenho principal (TFT), selecionando o menor valor; b) caso os valores fossem iguais, avaliava-se o critério MFT, selecionando o menor valor; c) caso os valores fossem ainda iguais, avaliava-se o critério C_{max} , selecionando o menor valor; d) caso ainda fossem iguais, usava-se a menor posição relativa das ordens na lista de avaliação.
- 8) Para a variante IG_{SA}^1 , na fase de aceitação (fase 4), a perturbação inicial realizada pelo subalgoritmo SA selecionava o número de ordens para extração e posterior reinserção de acordo com a fórmula $d_n = f_n$, onde n é o número de ordens. Diferentes valores para f_n foram avaliados. Os valores de f_n finais escolhidos foram: $f_{20} = 1$, $f_{50} = 3$, $f_{100} = 5$ e $f_{200} = 7$.
- 9) O número de iterações até a mudança de temperatura (L_{SA}) do subalgoritmo SA (fase 4) foi avaliado para os valores $L_{SA} = \{1, 3, 5\}$. O valor $L_{SA} = 1$ foi selecionado e fixado para todas as execuções.
- 10) O fator de filtragem δ do subalgoritmo SA (fase 4) foi avaliado para os valores $\delta = \{0,01; 0,02; \dots; 0,05\}$ para diferentes instâncias. O valor $\delta = 0,05$ foi selecionado e fixado para todas as execuções.
- 11) A temperatura inicial T_0 do subalgoritmo SA (fase 4) foi calculada por meio da fórmula $T_0 = \frac{\sum_{i,j} p_{i,j}}{n}$ que define o tempo médio de processamento da ordem em cada instância.
- 12) A temperatura final $T(t_{SA})$ e a taxa de resfriamento α do subalgoritmo SA (fase 4) foram calculados pelas fórmulas $T(t_{SA}) = 1$ e $\alpha = -\ln\left(\frac{1}{T_0}\right) \cdot \frac{1}{t_{SA}}$ respectivamente.

5.2.2. Metaheurística ILS_{SA-VNS}

A seguir são descritos os procedimentos usados para a escolha dos principais parâmetros da metaheurística ILS_{SA-VNS} .

- 1) O tempo máximo t_{max} de execução do método ILS_{SA-VNS} foi definido pela fórmula $t_{max} = n \times m \times f_n \times w$ (em milissegundos), onde n é o número de ordens, m é o número de máquinas e f_n e w são fatores de ajuste. Os valores de $f_n = \{15, 30, 60, 90, 120, 240\}$ foram avaliados em diferentes execuções do método e para diferentes tamanhos de instância. Os valores finais foram escolhidos de acordo com o modelo de distribuição, sendo: $f_{n<100} = 15$, $f_{n=100} = 30$ e $f_{n=200} = 90$ para o modelo DVC e $f_{n<100} = 60$, $f_{n=100} = 120$ e $f_{n=200} = 240$ para o modelo DFC. Os valores de w foram definidos de acordo com o subalgoritmo executado. Para os subalgoritmos de busca local simples e VNS, $w = 1/2$; para o subalgoritmo SA, $w = 1/3$; para o subalgoritmo de perturbação de rota, $w = 1/10$.
- 2) Para o subalgoritmo de busca local simples (fase 2), usou-se apenas a inserção de ordens como procedimento de geração de estruturas de vizinhança.
- 3) Nas fases 2 e 4, o número de pares de ordens selecionados para as mudanças de posição (inserção e swap) foi avaliado entre os valores $\{1, 2, 3\}$ e fixado em 1 (um).
- 4) Na execução do subalgoritmo SA (fase 3), o número de ordens selecionadas para extração e posterior reinserção foi determinado de acordo com a fórmula $d_n = f_n$, onde n é o número de ordens. Diferentes valores foram avaliados para f_n . Os valores de f_n finais escolhidos foram: $f_{20} = 1$, $f_{50} = 3$, $f_{100} = 5$ e $f_{200} = 10$.
- 5) O número de iterações até a mudança de temperatura (L_{SA}) do subalgoritmo SA (fase 3) foi avaliado para os valores $L_{SA} = \{1, 3, 5\}$. O valor $L_{SA} = 1$ foi selecionado e fixado para todas as execuções.
- 6) O fator de filtragem δ do subalgoritmo SA (fase 3) foi avaliado para os valores $\delta = \{0,01; 0,02; \dots; 0,05\}$ para diferentes instâncias. O valor $\delta = 0,05$ foi selecionado e fixado para todas as execuções.

- 7) A temperatura inicial T_0 do subalgoritmo SA (fase 3) foi calculada por meio da fórmula $T_0 = \frac{\sum_{i,j} p_{i,j}}{n}$ que define o tempo médio de processamento da ordem em cada instância.
- 8) A temperatura final $T(t_{SA})$ e a taxa de resfriamento α do subalgoritmo SA (fase 4) foram calculados pelas fórmulas $T(t_{SA}) = 1$ e $\alpha = -\ln\left(1/T_0\right) \cdot \frac{1}{t_{SA}}$ respectivamente.
- 9) O subalgoritmo VNS (fase 4) usou, em alternância, os procedimentos de inserção (primeiramente) e de *swap* (em seguida) de ordens para a criação de vizinhança. A troca de pares adjacentes (*pairwise exchange*) mostrou-se inefetiva e, portanto, foi descartada. O subalgoritmo de busca local usou o mesmo procedimento de mudança de posição usado para a geração de vizinhança.

5.3. Dados complementares das instâncias

Todos os dados complementares necessários à execução de cada instância foram obtidos a partir dos dados providos pelas próprias instâncias, ou seja, pelo número de ordens (n), número de máquinas (m) e tempos de processamento ($p_{i,j}$).

As fórmulas para geração dos dados complementares são propostas nesta Tese. Especificamente, as fórmulas de cálculo dos parâmetros s_j , Q , I e q_j foram criadas de modo empírico após inúmeras execuções dos algoritmos, tendo dois objetivos principais:

- 1) Usar apenas os dados providos pelas instâncias;
- 2) Obter-se um número médio de ordens alocadas por veículo igual ou superior a cinco independente da instância.

É importante observar que os dados complementares foram gerados por instância e não variaram ao longo da execução da programação da instância. A seguir, é descrito como os dados complementares de entrada foram gerados:

- 1) Cada ordem j definiu um único cliente k ($j = l$). Portanto, o número de clientes variou de 20 a 200, dependendo do tamanho da instância.
- 2) Os tempos de transporte entre clientes $t_{i,j}$ foram determinados pela transposição dos elementos da matriz de 'distâncias' (atrasos) entre ordens. Portanto, a matriz de tempos de transporte entre clientes foi determinada pela fórmula (5.1):

$$D' = D^T = (D_{j-1,j})_{n \times n}^T = (t_{j,j-1})_{n \times n} \quad (5.1)$$

- 3) Os tempos de transporte $t_{0,j}$ da fábrica até os clientes imediatos da rota foram definidos como $t_{0,j} = D_{0,j}$.
- 4) O tempo de serviço s_j foi calculado pela fórmula (5.2) determinada por experimentação:

$$s_j = \frac{1}{m} \times \sum_{i=1}^m p_{i,j} \quad (5.2)$$

- 5) A capacidade do veículo Q foi determinada pelas fórmulas (5.3) e (5.4) determinadas por experimentação, onde f é um fator de ajuste:

$$Q = 7 \times \left(1 + \frac{1}{f \times m}\right) \times \sum_{j=1}^n \sum_{i=1}^m p_{i,j} \quad (5.3)$$

$$f = \begin{cases} 10 & \text{se } \frac{n}{10 \times m} < 1 \\ 100 & \text{se } \frac{n}{10 \times m} \geq 1 \end{cases} \quad (5.4)$$

- 6) O intervalo entre coletas I para o modelo DFC não variou ao longo do tempo (fixo) e foi calculado pelas fórmulas (5.5) e (5.6) determinadas por experimentação, onde f é um fator de ajuste:

$$I = 1,45 \times \left(1 + \frac{1}{f \times m}\right) \times \sum_{j=1}^n \sum_{i=1}^m p_{i,j} \quad (5.5)$$

$$f = \begin{cases} 10 & \text{se } \frac{n}{10 \times m} < 1 \\ 100 & \text{se } \frac{n}{10 \times m} \geq 1 \end{cases} \quad (5.6)$$

7) O tamanho da ordem q_j foi calculado pelas fórmulas (5.7) e (5.8) determinadas por experimentação, onde f é um fator de ajuste:

$$q_j = \left(1 + \frac{n}{f \times m}\right) \times \sum_{i=1}^m p_{i,j} \quad (5.7)$$

$$f = \begin{cases} 10 & \text{se } \frac{n}{10 \times m} < 1 \\ 100 & \text{se } \frac{n}{10 \times m} \geq 1 \end{cases} \quad (5.8)$$

5.4. Codificação e execução dos algoritmos computacionais

Os métodos heurísticos foram codificados em C++ usando o *software* de código aberto *Code::Blocks* (versão 20.03). *Code::Blocks* é um ambiente integrado de desenvolvimento multiplataforma (*Integrated Development Environment* – IDE) para as linguagens C, C++ e Fortran.

Os algoritmos foram executados em uma máquina *Lenovo ThinkPad* com processador *Intel CORE i5 8350U* (CPU 1.70GHz / 1.90 GHz) e memória *RAM* de 16 GB, rodando o sistema operacional *Windows 10 Pro* 64bit.

6. Análise Estatística dos Resultados

Nesta seção, os resultados obtidos com as execuções dos métodos heurísticos são analisados estatisticamente seguindo os procedimentos descritos em Ruiz e Allahverdi (2007, 2009) e Montgomery (2017). Os resultados das análises são apresentados em seções deste capítulo e nos Apêndices A e B.

6.1. Considerações iniciais

Quatro conjuntos de dados (experimentos) foram analisados: dois para o modelo DFC (abordagens 1 e 2) e outros dois para o modelo DVC (abordagens 1 e 2).

As análises objetivaram:

- 1) Comparar os resultados das execuções dos três métodos propostos entre si para cada uma das heurísticas de iniciação tomadas individualmente como *benchmark*.
- 2) Comparar os resultados das execuções dos três métodos propostos entre si independente da heurística de iniciação adotada.
- 3) Comparar os resultados das execuções entre os tratamentos (combinações entre as metaheurísticas e as heurísticas de iniciação) tomados conjuntamente, em um primeiro momento, e dois a dois (pareados) em um segundo momento.
- 4) Verificar experimentalmente a hipótese proposta na pesquisa.

As comparações dos itens 1 e 2 acima foram realizadas usando a medida DRP (desvio relativo percentual). As comparações do item 3 acima foram realizadas por meio da técnica de análise de variância (ANOVA) e dos testes de Tukey e de Fisher (LSD). Para o item 4, foi utilizado o percentual de variação entre os valores da função objetivo (FO).

Especificamente para o modelo DFC, os resultados demonstraram que a abordagem 1 (minimizar C_{max} e TFT) não mostrou ser superior à abordagem 2 (minimizar TFT). A abordagem 2 obteve resultados moderadamente melhores para a quase totalidade dos experimentos. Portanto, para o modelo DFC, apenas os resultados gerados para a abordagem 2 foram analisados e são apresentados neste capítulo.

6.1.1. Desvio relativo percentual

A principal medida numérica utilizada nas análises de desempenho dos métodos heurísticos propostos foi o desvio relativo percentual (DRP), com fórmula geral dada por:

$$DRP_i^A = \frac{FO_i^A - FO_i^*}{FO_i^*} \times 100$$

Onde FO_i^A é o melhor valor da função objetivo obtido por algum algoritmo A para a instância i e FO_i^* é o melhor valor da função objetivo obtido para a instância i dentre todos os algoritmos avaliados.

Com base na definição geral de DRP, foram propostas três fórmulas, DRP1, DRP2 e DRP3 que foram usadas para analisar os resultados. As respectivas formulações são apresentadas a seguir.

Para a medida DRP1 tem-se:

$$DRP1 = DRP_s^{A,h} = \frac{FO_s^{A,h} - FO_s^*}{FO_s^*} \times 100 \quad \forall A, h, s$$

$$FO_s^{A,h} = \frac{1}{I_s} \times \sum_{i=1}^{I_s} FO_i^{A,h} \quad \forall A, h, s$$

$$FO_s^* = \min_A \{FO_s^{A,h}\} \quad \forall h, s$$

Onde $FO_s^{A,h}$ é a média dos valores da função objetivo obtidos pela metaheurística A executando a heurística de iniciação h para todas as instâncias i de determinado tamanho s ($n \times m$), sendo I_s o número de instâncias de mesmo tamanho; FO_s^* é o menor valor de $FO_s^{A,h}$ dentre todas as metaheurísticas que usam a heurística de iniciação h .

DRP1 foi calculada para cada combinação de uma metaheurística (IG_{SA}^1 , IG_{SA}^2 e ILS_{SA-VNS}) e uma heurística de iniciação (NEH, XNEH, CHINS, XCHINS, HC1 e HC2). Quanto menor o valor de DRP1 melhor o desempenho da metaheurística e respectiva heurística de iniciação, sendo o menor valor igual a zero.

Para a medida DRP2 tem-se:

$$DRP2 = DRP_s^A = \frac{FO_s^A - FO_s^*}{FO_s^*} \times 100 \quad \forall A, s$$

$$FO_s^A = \min_A \left\{ \frac{1}{I_s} \times \sum_{i=1}^{I_s} FO_i^{A,h} \right\} \quad \forall A, s$$

$$FO_s^* = \min_A \{FO_s^A\} \quad \forall s$$

Onde FO_s^A é o valor mínimo dentre todas as médias de valores da função objetivo obtidos pela metaheurística A executando a heurística de iniciação h para todas as instâncias i de determinado tamanho s ($n \times m$), sendo I_s o número de instâncias de mesmo tamanho; FO_s^* é o menor valor de FO_s^A dentre todas as metaheurística independente da heurística de iniciação.

DRP2 foi calculada para cada metaheurística (IG_{SA}^1 , IG_{SA}^2 e ILS_{SA-VNS}) não importando qual heurística de iniciação foi usada. Quanto menor o valor de $DRP2$ melhor o desempenho da metaheurística, sendo o menor valor igual a zero.

Para a medida $DRP3$ tem-se:

$$DRP3 = DRP_s^A = \frac{FO_s^A - FO_s^*}{FO_s^*} \times 100 \quad \forall A, s$$

$$FO_s^A = \frac{1}{H + I_s} \times \sum_{h=1}^H \sum_{i=1}^{I_s} FO_i^{A,h} \quad \forall A, s$$

$$FO_s^* = \min_A \{FO_s^A\} \quad \forall s$$

Onde FO_s^A é a média de todos os valores da função objetivo obtidos pela metaheurística A para todas as heurísticas de iniciação h e para todas as instâncias i de determinado tamanho s ($n \times m$), sendo I_s o número de instâncias de mesmo tamanho e H o número de heurísticas de iniciação avaliadas; FO_s^* é o menor valor de FO_s^A dentre todas as metaheurística independente da heurística de iniciação.

$DRP3$ foi calculada para cada metaheurística (IG_{SA}^1 , IG_{SA}^2 e ILS_{SA-VNS}) não importando qual heurística de iniciação foi usada. Quanto menor o valor de $DRP3$ melhor o desempenho da metaheurística, sendo o menor valor igual a zero.

Além da medida DRP , outras medidas complementares foram calculadas: média ponderada (MP), calculada com base nos tamanhos de instância ($n \times m$); média simples (MS); desvio padrão (DV), coeficiente de variação (CV); o número (contagem) de resultados vencedores (NRV), calculado com base nos melhores valores de DRP , e o percentual de melhoria do critério de otimização. Este último mede em valores percentuais e absolutos o quanto a solução inicial gerada pela heurística construtiva foi melhorada pela metaheurística. Esta medida é apresentada nas tabelas conjuntamente com a medida $DRP1$.

6.1.2. ANOVA e testes de Tukey e Fisher

ANOVA e os testes de Tukey e Fisher (LSD) foram aplicados objetivando avaliar a existência de diferenças significativas entre os tratamentos (por meio de suas médias) para cada problema, abordagem de otimização e tamanho de instância ($n \times m$).

Os testes de Tukey e Fisher (LSD) foram realizados apenas para os experimentos onde a ANOVA concluiu haver diferenças significativas (testes protegidos).

Os valores de comparação entre pares de tratamentos foram calculados pelas fórmulas a seguir.

Para o teste de Tukey tem-se:

$$V_{c,l} = \frac{(\overline{FO}_c - \overline{FO}_l)}{T_\alpha}$$

Para o teste de Fisher tem-se:

$$V_{c,l} = \frac{(\overline{FO}_c - \overline{FO}_l)}{LSD}$$

Onde $\overline{FO}_c$ e $\overline{FO}_l$ são as médias dos resultados de execução de todas as instâncias (replicações) dos tratamentos c e l comparados dois a dois; T_α e LSD são parâmetros de comparação de cada um dos testes estatísticos.

Para ambos os testes, os pares de tratamentos diferem se $|V_{c,l}| > 1$. Um valor significativamente negativo ($V_{c,l} < -1$) estabelece que o tratamento c é estatisticamente superior ao tratamento l . Um valor significativamente positivo ($V_{c,l} > 1$) estabelece que o tratamento c é estatisticamente inferior a l .

Os resultados dos testes de Tukey e Fisher para os três modelos (DFC2, DVC1 e DVC2) são apresentados no Apêndice B.

Nas próximas seções deste capítulo são apresentados e analisados os resultados das execuções para os modelos DFC2 e DVC1 por meio da medida DRP, estatísticas complementares e teste ANOVA.

As mesmas medidas usadas neste capítulo são empregadas para a análise do modelo DVC2 cujos resultados são apresentados no Apêndice A.

Uma comparação quanto ao número de resultados vencedores (NRV) entre as abordagens de otimização 1 e 2 para o modelo DVC é apresentada neste capítulo e no final do Apêndice A.

No final do capítulo são apresentadas as principais conclusões das análises estatísticas.

6.2. Análise dos resultados experimentais para o modelo DFC2

6.2.1. Análise DRP1 usando a heurística NEH na iniciação

Na Tabela 6.1, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando NEH para iniciação e b) o percentual de melhoria sobre o valor do NEH tomado como *benchmark*. Na Tabela 6.2, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Os dados da Tabela 6.1 demonstram que os melhores resultados foram obtidos pelo método IG_{SA}^2 com NRV igual a onze, além de obter melhorias superiores a 9% para todas as instâncias frente a heurística NEH.

Apesar dos outros métodos terem também conseguido melhorias frente a heurística NEH, nenhum deles apresentou resultado superior ao IG_{SA}^2 .

Tabela 6.1 – DRP1 e percentual de melhoria usando NEH (DFC2)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	16,87	<u>0,00</u>	12,82	5,14	<u>14,03</u>	5,90
20	10	12,87	<u>0,00</u>	13,12	2,61	<u>14,04</u>	2,87
20	20	8,48	<u>0,00</u>	8,50	4,44	<u>12,06</u>	7,23
50	5	13,65	<u>0,00</u>	14,52	5,18	<u>14,68</u>	9,87
50	10	8,47	<u>0,00</u>	9,00	4,14	<u>13,80</u>	4,12
50	20	16,87	<u>0,00</u>	12,82	4,27	<u>17,75</u>	4,07
100	5	12,87	<u>0,00</u>	13,12	3,85	<u>14,05</u>	7,20
100	10	8,48	<u>0,00</u>	8,50	2,98	<u>9,94</u>	2,74
100	20	13,65	<u>0,00</u>	14,52	2,30	<u>13,66</u>	2,29
200	10	8,47	<u>0,00</u>	9,00	1,89	<u>9,71</u>	1,13
200	20	16,87	<u>0,00</u>	12,82	2,07	<u>17,75</u>	1,58
NRV		0	11	0	0	11	0

Tabela 6.2 – Estatísticas complementares usando NEH (DFC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	10,18	<u>0,00</u>	10,14	2,68	<u>11,61</u>	2,69
MS	11,06	<u>0,00</u>	9,96	3,53	<u>13,08</u>	4,45
DP	2,97	<u>0,00</u>	3,18	<u>1,21</u>	2,42	2,76
CV	<u>0,27</u>	-	0,32	0,34	<u>0,19</u>	0,62

6.2.2. Análise DRP1 usando a heurística XNEH na iniciação

Na Tabela 6.3, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando XNEH para iniciação e b) o percentual de melhoria sobre o valor do XNEH tomado como *benchmark*. Na Tabela 6.4, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Os dados da Tabela 6.3 demonstram que os melhores resultados foram obtidos pelo método IG_{SA}^2 com NRV igual a onze, além de obter melhorias superiores a 8% para todas as instâncias frente à heurística XNEH.

Apesar dos outros métodos terem também conseguido melhorias em comparação à heurística XNEH, nenhum deles apresentou resultado superior ao IG_{SA}^2 .

Tabela 6.3 – DRP1 e percentual de melhoria usando XNEH (DFC2)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	9,11	<u>0,00</u>	6,84	3,21	<u>11,27</u>	5,25
20	10	13,54	<u>0,00</u>	8,27	1,03	<u>12,87</u>	5,66
20	20	6,73	<u>0,00</u>	5,59	3,93	<u>9,92</u>	4,95
50	5	10,76	<u>0,00</u>	9,11	3,00	<u>12,39</u>	4,43
50	10	10,66	<u>0,00</u>	10,50	3,34	<u>12,67</u>	3,49
50	20	6,02	<u>0,00</u>	6,74	2,82	<u>8,34</u>	2,16
100	5	15,11	<u>0,00</u>	13,23	1,82	<u>14,72</u>	3,41
100	10	14,23	<u>0,00</u>	14,19	1,32	<u>13,60</u>	1,34
100	20	8,25	<u>0,00</u>	8,09	1,25	<u>8,78</u>	1,40
200	10	12,94	<u>0,00</u>	14,06	1,84	<u>13,08</u>	0,87
200	20	8,43	<u>0,00</u>	9,14	1,67	<u>9,32</u>	1,03
NRV		0	11	0	0	11	0

Tabela 6.4 – Estatísticas complementares usando XNEH (DFC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	9,89	<u>0,00</u>	10,08	1,88	<u>10,66</u>	1,70
MS	10,53	<u>0,00</u>	9,61	2,29	<u>11,54</u>	3,09
DP	3,10	<u>0,00</u>	3,02	<u>0,99</u>	2,14	1,81
CV	<u>0,29</u>	-	0,31	0,43	<u>0,19</u>	0,59

6.2.3. Análise DRP1 usando a heurística CHINS na iniciação

Na Tabela 6.5, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando CHINS para iniciação e b) o percentual de melhoria sobre o valor do CHINS tomado como *benchmark*. Na Tabela 6.6, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Os dados da Tabela 6.5 demonstram que o método IG_{SA}^2 obteve desempenho superior na contagem geral dada pelo NRV. O método ILS_{SA-VNS} foi superior para as instâncias grandes: três contagens vencedoras para o ILS_{SA-VNS} frente a duas do método IG_{SA}^2 .

Tabela 6.5 – DRP1 e percentual de melhoria usando CHINS (DFC2)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	1,59	<u>0,00</u>	0,11	1,76	<u>3,31</u>	3,17
20	10	4,11	<u>0,00</u>	4,13	3,26	<u>7,05</u>	3,37
20	20	3,53	<u>0,00</u>	1,95	2,13	<u>5,49</u>	3,62
50	5	3,27	2,85	<u>0,00</u>	2,29	2,66	<u>5,37</u>
50	10	<u>0,00</u>	0,32	0,24	<u>2,36</u>	2,05	2,13
50	20	0,15	<u>0,00</u>	0,58	1,82	<u>1,97</u>	1,41
100	5	2,33	0,31	<u>0,00</u>	1,66	3,60	<u>3,90</u>
100	10	0,18	<u>0,00</u>	0,13	1,88	<u>2,06</u>	1,93
100	20	0,25	<u>0,00</u>	0,32	1,16	<u>1,40</u>	1,09
200	10	0,23	0,02	<u>0,00</u>	1,27	1,47	<u>1,49</u>
200	20	0,23	0,10	<u>0,00</u>	1,00	1,13	<u>1,23</u>
NRV		1	6	4	1	6	4

Para instâncias menores, o método IG_{SA}^2 obteve resultados melhores que o método ILS_{SA-VNS} na contagem do NRV. Usando as estatísticas MP e MS calculadas a partir do DRP1 (Tabela 6.6), o método IG_{SA}^2 foi superior com 0,12 e 0,33 respectivamente.

Tabela 6.6 – Estatísticas complementares usando CHINS (DFC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	0,55	<u>0,12</u>	0,26	1,41	<u>1,82</u>	1,69
MS	1,44	<u>0,33</u>	0,68	1,87	<u>2,93</u>	2,61
DP	1,59	<u>0,85</u>	1,28	<u>0,64</u>	1,85	1,37
CV	<u>1,1</u>	2,58	1,88	<u>0,34</u>	0,63	0,52

O método ILS_{SA-VNS} obteve os melhores resultados de percentual de melhoria para as instâncias maiores. Com base nesta medida, o método IG_{SA}^2 obteve os melhores resultados para as estatísticas MP e MS. Para as instâncias médias, os três métodos obtiveram resultados iguais ($NRV = 1$).

Todos os métodos conseguiram melhorar as soluções iniciais geradas pela heurística CHINS. Em geral, os melhores resultados para as instâncias grandes foram obtidos pelos métodos ILS_{SA-VNS} . Para instâncias pequenas o método IG_{SA}^2 foi melhor. Para instâncias médias, houve empate entre os métodos.

6.2.4. Análise DRP1 usando a heurística XCHINS na iniciação

Na Tabela 6.7, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando XCHINS para iniciação e b) o percentual de melhoria sobre o valor do XCHINS tomado como *benchmark*. Na Tabela 6.8, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Tabela 6.7 – DRP1 e percentual de melhoria usando XCHINS (DFC2)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	1,75	<u>0,00</u>	0,79	1,26	<u>3,00</u>	2,20
20	10	4,35	<u>0,00</u>	4,41	1,03	<u>5,12</u>	0,97
20	20	2,13	<u>0,00</u>	1,35	1,23	<u>3,30</u>	1,98
50	5	1,11	2,23	<u>0,00</u>	2,05	0,97	<u>3,15</u>
50	10	<u>0,00</u>	0,61	0,02	<u>1,04</u>	0,43	1,06
50	20	0,33	<u>0,00</u>	0,59	0,85	<u>1,18</u>	0,60
100	5	0,77	<u>0,00</u>	0,20	1,58	<u>2,34</u>	2,12
100	10	0,28	0,37	<u>0,00</u>	0,61	0,52	<u>0,89</u>
100	20	<u>0,00</u>	0,10	0,21	<u>0,91</u>	0,82	0,71
200	10	0,05	0,36	<u>0,00</u>	0,52	0,21	<u>0,57</u>
200	20	<u>0,00</u>	0,37	0,17	<u>0,82</u>	0,45	0,64
NRV		3	5	3	3	5	3

Os dados da Tabela 6.7 demonstram que os três métodos obtiveram desempenhos semelhantes quanto ao NRV, sendo o método IG_{SA}^2 superior na contagem geral. Para as instâncias de maior porte, os métodos ILS_{SA-VNS} e IG_{SA}^1 empataram com duas contagens frente a apenas uma do método IG_{SA}^2 .

Para instâncias menores, o método IG_{SA}^2 obteve resultados melhores que os outros dois métodos na contagem do NRV. Usando a estatística MP calculada a partir do DRP1 (Tabela 6.8), o método IG_{SA}^1 obteve resultado superior (0,27). Para a estatística MS, o método IG_{SA}^2 foi superior (0,37).

Tabela 6.8 – Estatísticas complementares usando XCHINS (DFC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	<u>0,27</u>	0,30	0,28	<u>0,86</u>	0,82	0,85
MS	0,98	<u>0,37</u>	0,70	1,08	<u>1,67</u>	1,35
DP	1,34	<u>0,65</u>	1,30	<u>0,44</u>	1,57	0,86
CV	<u>1,37</u>	1,76	1,86	<u>0,41</u>	0,94	0,64

O mesmo resultado foi obtido para o percentual de melhoria, com empate dos métodos ILS_{SA-VNS} e IG_{SA}^1 . Ainda, o método IG_{SA}^1 obteve valor superior para a estatística MP (0,86) e o método IG_{SA}^2 obteve e resultado superior para MS (1,67).

Todos os métodos conseguiram melhorar as soluções iniciais geradas pela heurística XCHINS. Em geral, para as instâncias grandes, os melhores resultados foram obtidos pelos métodos ILS_{SA-VNS} e IG_{SA}^1 . Para instâncias pequenas, o método IG_{SA}^2 foi melhor. Houve empate entre os métodos para instâncias de médio porte.

6.2.5. Análise DRP1 usando a heurística HC1 na iniciação

Na Tabela 6.9, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando HC1 para iniciação e b) o percentual de melhoria sobre o

valor do HC1 tomado como *benchmark*. Na Tabela 6.10, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Os dados da Tabela 6.9 demonstram que os três métodos obtiveram desempenhos semelhantes quanto ao NRV, com o método ILS_{SA-VNS} empatando com o método IG_{SA}^2 .

Tabela 6.9 – DRP1 e percentual de melhoria usando HC1 (DFC2)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	3,36	<u>0,00</u>	1,86	1,52	<u>4,73</u>	2,94
20	10	5,75	<u>0,00</u>	2,18	1,39	<u>6,68</u>	4,66
20	20	3,49	<u>0,00</u>	3,30	0,81	<u>4,19</u>	1,00
50	5	1,92	2,57	<u>0,00</u>	2,63	1,97	<u>4,44</u>
50	10	<u>0,00</u>	0,93	0,12	<u>1,50</u>	0,58	1,38
50	20	<u>0,00</u>	0,62	0,03	<u>1,30</u>	0,70	1,26
100	5	1,49	0,28	<u>0,00</u>	1,23	2,40	<u>2,66</u>
100	10	0,31	0,73	<u>0,00</u>	1,03	0,62	<u>1,34</u>
100	20	0,07	<u>0,00</u>	0,14	1,01	<u>1,08</u>	0,94
200	10	<u>0,00</u>	0,16	0,03	<u>0,46</u>	0,31	0,43
200	20	0,35	0,60	<u>0,00</u>	0,55	0,30	<u>0,90</u>
NRV		3	4	4	3	4	4

O método ILS_{SA-VNS} obteve os melhores resultados de DRP1 e percentual de melhoria para as instâncias grandes e para a estatística MP calculada a partir desta última medida (1,15). Para MS, o método IG_{SA}^2 foi superior (2,14). Usando as estatísticas MP e MS calculadas a partir do DRP1 (Tabela 6.10), o método ILS_{SA-VNS} foi superior aos demais.

Todos os métodos conseguiram melhorar as soluções iniciais geradas pela heurística HC1. Em geral, para as instâncias grandes, os melhores resultados foram obtidos pelo método ILS_{SA-VNS} . Para instâncias médias, o método IG_{SA}^1 foi melhor e para instâncias pequenas o método IG_{SA}^2 foi melhor.

Tabela 6.10 – Estatísticas complementares usando HC1 (DFC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	0,50	0,44	<u>0,20</u>	0,86	0,90	<u>1,15</u>
MS	1,52	0,54	<u>0,70</u>	1,22	<u>2,14</u>	2,00
DP	1,93	<u>0,75</u>	1,17	<u>0,59</u>	2,15	1,47
CV	<u>1,27</u>	1,39	1,67	<u>0,48</u>	1,01	0,74

6.2.6. Análise DRP1 usando a heurística HC2 na iniciação

Na Tabela 6.11, são apresentados: a) os valores de DRP1 para as três metaheurísticas, usando HC2 para iniciação e b) o percentual de melhoria sobre o valor do HC2 tomado como *benchmark*. Na Tabela 6.12, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Tabela 6.11 – DRP1 e percentual de melhoria usando HC2 (DFC2)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	2,54	<u>0,00</u>	1,22	1,50	<u>3,94</u>	2,78
20	10	3,75	<u>0,00</u>	3,98	1,80	<u>5,30</u>	1,54
20	20	2,01	<u>0,00</u>	1,30	1,79	<u>3,72</u>	2,46
50	5	3,48	3,92	<u>0,00</u>	1,95	1,51	<u>5,20</u>
50	10	<u>0,00</u>	0,38	0,08	<u>1,06</u>	0,68	0,98
50	20	<u>0,00</u>	1,03	0,80	<u>1,29</u>	0,29	0,51
100	5	1,15	0,16	<u>0,00</u>	1,14	2,10	<u>2,27</u>
100	10	0,14	0,09	<u>0,00</u>	0,79	0,84	<u>0,92</u>
100	20	0,08	0,34	<u>0,00</u>	1,05	0,80	<u>1,13</u>
200	10	0,36	0,37	<u>0,00</u>	0,25	0,24	<u>0,60</u>
200	20	0,25	0,48	<u>0,00</u>	0,64	0,40	<u>0,88</u>
NRV		2	3	6	2	3	6

Os dados da Tabela 6.11 demonstram que o método ILS_{SA-VNS} obteve resultados melhores que os outros métodos na contagem do NRV, especificamente para instâncias grandes.

Para instâncias pequenas, o método IG_{SA}^2 obteve resultados melhores que os outros dois métodos na contagem do NRV. Usando a estatística MP calculada a partir do DRP1 (Tabela 6.12), o método ILS_{SA-VNS} foi superior aos demais (0,19). Usando MS, o método IG_{SA}^2 foi superior (0,62).

Tabela 6.12 – Estatísticas complementares usando HC2 (DFC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	0,44	0,48	<u>0,19</u>	0,84	0,80	<u>1,08</u>
MS	1,25	<u>0,62</u>	0,67	1,21	1,80	<u>1,75</u>
DP	1,45	<u>1,14</u>	1,21	<u>0,53</u>	1,75	1,38
CV	<u>1,16</u>	1,84	1,81	<u>0,44</u>	0,97	0,79

Todos os métodos conseguiram melhorar as soluções iniciais geradas pela heurística HC2. Em geral, os melhores resultados para as instâncias grandes foram obtidos pelo método ILS_{SA-VNS} . Para instâncias médias, o método IG_{SA}^1 foi melhor. Para instâncias pequenas o método IG_{SA}^2 foi melhor.

6.2.7. Resumo da análise DRP1

Na Tabela 6.13, é apresentado o resumo do número de resultados vencedores (NRV) usando a medida DRP1 para as três metaheurística e respectivas heurísticas de iniciação.

Os resultados demonstram que, no cômputo geral da contagem NRV, a metaheurística IG_{SA}^2 foi a vencedora em 40 resultados vencedores. A metaheurística ILS_{SA-VNS} ficou em segundo lugar com 17 resultados vencedores.

Tabela 6.13 – Resumo NRV usando DRP1 (DFC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
NEH	0	11	0	0	11	0
XNEH	0	11	0	0	11	0
CHINS	1	6	4	1	6	4
XCHINS	3	5	3	3	5	3
HC1	3	4	4	3	4	4
HC2	2	3	6	2	3	6
Total	9	40	17	9	40	17

Na Tabela 6.13B, é apresentado um resumo do NRV por tamanho de instância. O método IG_{SA}^2 foi o vencedor para todos os tamanhos de instâncias, mas seu desempenho ficou muito próximo do método ILS_{SA-VNS} para instâncias grandes. Neste caso, o método ILS_{SA-VNS} superou os demais para a metade das heurísticas de iniciação.

Tabela 6.13B – Resumo NRV por tamanho de instância (DFC2)

	Pequena ($I_s = 20$)			Média ($I_s = 50$)			Grande ($I_s \geq 100$)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
NEH	0	3	0	0	3	0	0	5	0
XNEH	0	3	0	0	3	0	0	5	0
CHINS	0	3	0	1	1	1	0	2	3
XCHINS	0	3	0	1	1	1	2	1	2
HC1	0	3	0	2	0	1	1	1	3
HC2	0	3	0	2	0	1	0	0	5
Total	0	18	0	6	8	4	3	14	13

6.2.8. Análise DRP independente da heurística de iniciação

As avaliações dos resultados das execuções das três metaheurísticas independentes das heurísticas de iniciação são feitas por meio das medidas DRP2 e DRP3. Na Tabela 6.14, são apresentados os valores destas medidas para as três metaheurísticas. Os melhores valores estão grifados.

Tabela 6.14 – DRP2 e DRP3 para as três metaheurísticas (DFC2)

n	m	DRP2 (%)			DRP3 (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	1,75	<u>0,00</u>	0,37	4,79	<u>0,00</u>	3,38
20	10	3,75	<u>0,00</u>	2,45	7,47	<u>0,00</u>	5,99
20	20	2,25	<u>0,00</u>	1,47	4,42	<u>0,00</u>	3,17
50	5	2,57	3,28	<u>0,00</u>	3,35	<u>0,00</u>	0,58
50	10	<u>0,00</u>	0,62	0,01	3,28	<u>0,00</u>	3,32
50	20	<u>0,00</u>	0,13	0,18	1,91	<u>0,00</u>	2,32
100	5	1,15	0,16	<u>0,00</u>	6,24	<u>0,00</u>	4,32
100	10	0,28	0,37	<u>0,00</u>	4,47	<u>0,00</u>	4,38
100	20	0,08	0,18	<u>0,00</u>	2,79	<u>0,00</u>	2,81
200	10	0,05	0,21	<u>0,00</u>	4,41	<u>0,00</u>	4,65
200	20	0,17	0,54	<u>0,00</u>	2,70	<u>0,00</u>	2,80
NRV		2	3	6	0	11	0

Usando DRP2 para medir as melhores execuções, a metaheurística ILS_{SA-VNS} foi a vencedora para seis tamanhos de instância, notadamente para as instâncias grandes. O método IG_{SA}^2 ficou em segundo lugar, obtendo os melhores resultados para as instâncias pequenas. O número de resultados vencedores (NRV) é apresentado no final da tabela.

Usando DRP3, a metaheurística IG_{SA}^2 é a vencedora para os onze tamanhos de instância. O número de resultados vencedores (NRV) é apresentado no final da tabela.

6.2.9. Análise de variância (ANOVA)

Na Tabela 6.15, são apresentados os resultados da análise de variância (ANOVA) para os onze experimentos do conjunto de experimentos do PPIPD-NWFS-DFC2.

Na coluna ‘Difere?’, são identificados os experimentos cujos tratamentos (combinação de uma metaheurística e uma heurística de iniciação) diferem significativamente entre si. Este resultado sugere que a combinação de uma

metaheurística e uma heurística de iniciação afeta significativamente os resultados em quase todos os tamanhos de instância avaliados.

Tabela 6.15 – Resultados ANOVA (DFC2)

n	m	F0	Fcrit	F0/Fcrit	α	Valor-p	Difere?
20	5	1,223	1,767	0,69	0,05	0,270	Não
20	10	1,761	1,767	1,00	0,05	0,051	Não
20	20	1,714	1,767	0,97	0,05	0,060	Não
50	5	7,844	1,767	4,44	0,05	1,39E-10	Sim
50	10	11,749	1,767	6,65	0,05	1,41E-14	Sim
50	20	4,499	1,767	2,55	0,05	3,13E-06	Sim
100	5	21,791	1,767	12,33	0,05	8,63E-22	Sim
100	10	15,463	1,767	8,75	0,05	1,00E-17	Sim
100	20	33,536	1,767	18,98	0,05	1,99E-27	Sim
200	10	219,210	1,767	124,06	0,05	6,62E-55	Sim
200	20	42,251	1,767	23,91	0,05	1,34E-30	Sim

Para os experimentos que diferem, os métodos *post-hoc* Tukey e Fisher (LSD) são empregados para avaliar as diferenças entre os tratamentos. Os resultados das análises realizadas por ambos os métodos são apresentados no Apêndice B.

6.3. Análise dos resultados experimentais para o modelo DVC1

6.3.1. Análise DRP1 usando a heurística NEH na iniciação

Na Tabela 6.16, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando NEH para iniciação e b) o percentual de melhoria sobre o valor do NEH tomado como *benchmark*. Na Tabela 6.17, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Os dados da Tabela 6.16 demonstram que os melhores resultados foram majoritariamente obtidos pelo método IG_{SA}^2 com NRV igual a dez, além de obter melhorias superiores a 3% para todas as instâncias frente a heurística NEH. O método ILS_{SA-VNS} foi superior em apenas um tamanho de instância na contagem NRV, mas obteve resultado inferior para o percentual de melhoria.

Apesar dos outros métodos terem também conseguido melhorias frente a heurística NEH, nenhum deles apresentou resultado majoritariamente superior ao IG_{SA}^2 .

Tabela 6.16 – DRP1 e percentual de melhoria usando NEH (DVC1)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	2,44	<u>0,00</u>	3,22	8,59	<u>10,83</u>	7,98
20	10	3,81	<u>0,00</u>	3,49	7,56	<u>10,88</u>	7,82
20	20	3,86	<u>0,00</u>	3,46	5,76	<u>9,20</u>	6,11
50	5	2,69	<u>0,00</u>	2,03	5,70	<u>8,15</u>	6,31
50	10	2,76	<u>0,00</u>	2,33	4,39	<u>6,94</u>	4,80
50	20	2,46	<u>0,00</u>	2,39	4,27	<u>6,56</u>	4,32
100	5	2,48	<u>0,00</u>	2,38	3,20	<u>5,47</u>	3,31
100	10	3,41	<u>0,00</u>	3,39	2,98	<u>6,18</u>	3,01
100	20	0,39	0,01	<u>0,00</u>	2,87	3,25	<u>3,24</u>
200	10	1,79	<u>0,00</u>	1,71	1,54	<u>3,26</u>	1,61
200	20	1,90	<u>0,00</u>	1,89	1,51	<u>3,35</u>	1,53
NRV		0	10	1	0	10	1

Tabela 6.17 – Estatísticas complementares usando NEH (DVC1)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	1,98	<u>0,00</u>	1,84	2,68	<u>4,55</u>	2,81
MS	2,54	<u>0,00</u>	2,39	4,40	<u>6,73</u>	4,55
DP	0,99	<u>0,00</u>	1,03	2,31	2,82	<u>2,27</u>
CV	0,39	-	0,43	0,53	<u>0,42</u>	0,5

6.3.2. Análise DRP1 usando a heurística XNEH na iniciação

Na Tabela 6.18, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando XNEH para iniciação e b) o percentual de melhoria sobre o valor do XNEH tomado como *benchmark*. Na Tabela 6.19, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Os dados da Tabela 6.18 demonstram que os melhores resultados foram obtidos pelo método IG_{SA}^2 com NRV igual a onze, além de obter melhorias superiores a 2,5% para todas as instâncias frente à heurística XNEH.

Tabela 6.18 – DRP1 e percentual de melhoria usando XNEH (DVC1)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	1,92	<u>0,00</u>	1,70	6,61	<u>8,39</u>	6,81
20	10	1,68	<u>0,00</u>	1,33	6,53	<u>8,04</u>	6,82
20	20	3,45	<u>0,00</u>	3,10	5,50	<u>8,59</u>	5,81
50	5	2,09	<u>0,00</u>	2,23	6,58	<u>8,49</u>	6,45
50	10	2,99	<u>0,00</u>	2,63	3,70	<u>6,49</u>	4,05
50	20	2,29	<u>0,00</u>	2,14	3,79	<u>5,93</u>	3,92
100	5	2,86	<u>0,00</u>	2,87	3,51	<u>6,15</u>	3,48
100	10	2,77	<u>0,00</u>	2,77	2,75	<u>5,35</u>	2,74
100	20	0,14	<u>0,00</u>	0,13	2,84	<u>2,98</u>	2,86
200	10	1,04	<u>0,00</u>	1,07	1,48	<u>2,48</u>	1,45
200	20	1,52	<u>0,00</u>	1,47	1,42	<u>2,90</u>	1,48
NRV		0	11	0	0	11	0

Tabela 6.19 – Estatísticas complementares usando XNEH (DVC1)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	1,58	<u>0,00</u>	1,52	2,53	<u>4,03</u>	2,59
MS	2,07	<u>0,00</u>	1,95	4,06	<u>5,98</u>	4,17
DP	0,96	<u>0,00</u>	0,90	<u>1,96</u>	2,34	2,02
CV	0,46	-	0,46	0,48	<u>0,39</u>	0,48

Apesar dos outros métodos terem também conseguido melhorias em comparação à heurística XNEH, nenhum deles apresentou resultado superior ao IG_{SA}^2 .

6.3.3. Análise DRP1 usando a heurística CHINS na iniciação

Na Tabela 6.20, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando CHINS para iniciação e b) o percentual de melhoria sobre o valor do CHINS tomado como *benchmark*. Na Tabela 6.21, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Os dados da Tabela 6.20 demonstram que os melhores resultados foram obtidos pelo método IG_{SA}^2 com NRV igual a onze, além de obter melhorias superiores a 1,6% para todas as instâncias frente a heurística CHINS.

Tabela 6.20 – DRP1 e percentual de melhoria usando CHINS (DVC1)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	0,84	<u>0,00</u>	2,11	11,38	<u>12,08</u>	10,25
20	10	1,74	<u>0,00</u>	3,84	8,42	<u>10,00</u>	6,54
20	20	4,37	<u>0,00</u>	4,70	6,18	<u>10,11</u>	5,88
50	5	2,12	<u>0,00</u>	1,88	5,85	<u>7,79</u>	6,07
50	10	2,04	<u>0,00</u>	2,10	4,16	<u>6,08</u>	4,10
50	20	1,19	<u>0,00</u>	0,74	3,49	<u>4,61</u>	3,92
100	5	3,00	<u>0,00</u>	2,70	2,89	<u>5,59</u>	3,16
100	10	1,13	<u>0,00</u>	1,04	2,50	<u>3,61</u>	2,59
100	20	1,34	<u>0,00</u>	1,12	2,19	<u>3,48</u>	2,40
200	10	0,40	<u>0,00</u>	0,33	1,31	<u>1,70</u>	1,38
200	20	0,77	<u>0,00</u>	0,74	1,26	<u>2,02</u>	1,30
NRV		0	11	0	0	11	0

Apesar dos outros métodos terem também conseguido melhorias frente a heurística CHINS, nenhum deles apresentou resultado superior ao IG_{SA}^2 .

Tabela 6.21 – Estatísticas complementares usando CHINS (DVC1)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	1,18	<u>0,00</u>	1,12	2,37	<u>3,49</u>	2,44
MS	1,72	<u>0,00</u>	1,94	4,51	<u>6,10</u>	4,33
DP	1,15	<u>0,00</u>	1,37	3,18	3,49	<u>2,68</u>
CV	0,67	-	0,71	0,71	<u>0,57</u>	0,62

6.3.4. Análise DRP1 usando a heurística XCHINS na iniciação

Na Tabela 6.22, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando XCHINS para iniciação e b) o percentual de melhoria sobre o valor do XCHINS tomado como *benchmark*. Na Tabela 6.23, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Os dados da Tabela 6.22 demonstram que os melhores resultados foram majoritariamente obtidos pelo método IG_{SA}^2 com NRV igual a dez, além de obter melhorias superiores a 1,2% para todas as instâncias frente a heurística XCHINS.

Tabela 6.22 – DRP1 e percentual de melhoria usando XCHINS (DVC1)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	0,40	<u>0,00</u>	1,65	10,44	<u>10,73</u>	9,29
20	10	2,53	<u>0,00</u>	3,75	8,04	<u>10,36</u>	6,93
20	20	3,37	<u>0,00</u>	4,15	5,93	<u>8,99</u>	5,22
50	5	3,07	<u>0,00</u>	3,62	6,02	<u>8,82</u>	5,52
50	10	2,06	<u>0,00</u>	1,99	3,77	<u>5,70</u>	3,83
50	20	1,19	<u>0,00</u>	1,49	3,49	<u>4,62</u>	3,21
100	5	0,78	<u>0,00</u>	0,79	2,61	<u>3,33</u>	2,59
100	10	0,37	<u>0,00</u>	0,39	2,56	<u>2,92</u>	2,54
100	20	1,18	<u>0,00</u>	1,17	2,30	<u>3,43</u>	2,31
200	10	1,09	<u>0,00</u>	1,10	1,43	<u>2,48</u>	1,41
200	20	0,08	0,09	<u>0,00</u>	1,26	1,25	<u>1,34</u>
NRV		0	10	1	0	10	1

O método ILS_{SA-VNS} foi superior em apenas um tamanho de instância na contagem NRV, obtendo resultado superior também para o percentual de melhoria.

Apesar dos outros métodos terem também conseguido melhorias frente a heurística XCHINS, nenhum deles apresentou resultado majoritariamente superior ao IG_{SA}^2 .

Tabela 6.23 – Estatísticas complementares usando XCHINS (DVC1)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	0,88	<u>0,03</u>	0,94	2,37	<u>3,16</u>	2,30
MS	1,47	<u>0,01</u>	1,83	4,35	<u>5,69</u>	4,02
DP	1,13	<u>0,03</u>	1,41	2,93	3,43	<u>2,49</u>
CV	<u>0,77</u>	3	<u>0,77</u>	0,67	<u>0,6</u>	0,62

6.3.5. Análise DRP1 usando a heurística HC1 na iniciação

Na Tabela 6.24, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando HC1 para iniciação e b) o percentual de melhoria sobre o valor do HC1 tomado como *benchmark*. Na Tabela 6.25, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Os dados da Tabela 6.24 demonstram que os melhores resultados foram majoritariamente obtidos pelo método IG_{SA}^2 com NRV igual a dez, além de obter melhorias superiores a 2% para todas as instâncias frente a heurística HC1.

O método ILS_{SA-VNS} foi superior em apenas um tamanho de instância na contagem NRV, obtendo resultado superior também para o percentual de melhoria.

Apesar dos outros métodos terem também conseguido melhorias frente a heurística HC1, nenhum deles apresentou resultado majoritariamente superior ao IG_{SA}^2 .

Tabela 6.24 – DRP1 e percentual de melhoria usando HC1 (DVC1)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	2,66	<u>0,00</u>	1,51	5,26	<u>7,68</u>	6,30
20	10	4,67	<u>0,00</u>	2,86	4,64	<u>8,97</u>	6,33
20	20	6,19	<u>0,00</u>	3,40	4,39	<u>9,94</u>	6,90
50	5	1,28	<u>0,00</u>	0,89	4,35	<u>5,57</u>	4,75
50	10	3,05	<u>0,00</u>	2,06	4,61	<u>7,37</u>	5,52
50	20	4,81	<u>0,00</u>	3,52	3,16	<u>7,59</u>	4,35
100	5	0,00	0,81	<u>0,00</u>	2,85	2,08	<u>2,87</u>
100	10	3,02	<u>0,00</u>	2,40	2,62	<u>5,48</u>	3,20
100	20	2,59	<u>0,00</u>	2,03	2,62	<u>5,07</u>	3,14
200	10	2,07	<u>0,00</u>	1,85	1,52	<u>3,50</u>	1,72
200	20	2,81	<u>0,00</u>	3,00	1,59	<u>4,27</u>	1,40
NRV		0	10	1	0	10	1

Tabela 6.25 – Estatísticas complementares usando HC1 (DVC1)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	2,84	<u>0,03</u>	2,43	2,38	<u>5,02</u>	2,75
MS	3,01	<u>0,07</u>	2,14	3,42	<u>6,14</u>	4,23
DP	1,71	<u>0,24</u>	1,07	<u>1,29</u>	2,39	1,90
CV	0,57	3,43	<u>0,5</u>	<u>0,38</u>	0,39	0,45

6.3.6. Análise DRP1 usando a heurística HC2 na iniciação

Na Tabela 6.26, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando HC1 para iniciação e b) o percentual de melhoria sobre o valor do HC2 tomado como *benchmark*. Na Tabela 6.27, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Os dados da Tabela 6.26 demonstram que os melhores resultados foram obtidos pelo método IG_{SA}^2 com NRV igual a onze, além de obter melhorias superiores a 1,5% para todas as instâncias frente a heurística HC2.

Apesar dos outros métodos terem também conseguido melhorias frente a heurística HC2, nenhum deles apresentou resultado superior ao IG_{SA}^2 .

Tabela 6.26 – DRP1 e percentual de melhoria usando HC2 (DVC1)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	2,51	<u>0,00</u>	1,67	7,35	<u>9,64</u>	8,13
20	10	2,79	<u>0,00</u>	1,92	6,82	<u>9,39</u>	7,60
20	20	2,46	<u>0,00</u>	2,13	6,39	<u>8,61</u>	6,69
50	5	1,64	<u>0,00</u>	2,05	5,19	<u>6,67</u>	4,83
50	10	2,05	<u>0,00</u>	1,52	4,00	<u>5,93</u>	4,50
50	20	1,21	<u>0,00</u>	1,04	3,30	<u>4,44</u>	3,45
100	5	1,58	<u>0,00</u>	1,33	2,68	<u>4,16</u>	2,92
100	10	0,37	<u>0,00</u>	0,31	2,65	<u>3,00</u>	2,72
100	20	0,54	<u>0,00</u>	0,33	2,76	<u>3,28</u>	2,97
200	10	0,51	<u>0,00</u>	0,38	1,37	<u>1,86</u>	1,50
200	20	0,47	<u>0,00</u>	0,41	1,31	<u>1,78</u>	1,38
NRV		0	11	0	0	11	0

Tabela 6.27 – Estatísticas complementares usando HC2 (DVC1)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	0,80	<u>0,00</u>	0,65	2,41	<u>3,17</u>	2,56
MS	1,47	<u>0,00</u>	1,19	3,98	<u>5,34</u>	4,24
DP	0,91	<u>0,00</u>	0,73	<u>2,15</u>	2,91	2,34
CV	0,62	-	0,61	<u>0,54</u>	<u>0,54</u>	0,55

6.3.7. Resumo da análise DRP1

Na Tabela 6.28, é apresentado o resumo das contagens de resultados vencedores (NRV) usando a medida DRP1 para as três metaheurística e respectivas heurísticas de iniciação.

Os resultados demonstram que o método IG_{SA}^2 foi o vencedor com 64 resultados melhores. O método ILS_{SA-VNS} ficou em segundo lugar com 2 apenas resultados melhores.

Diferente do modelo DFC onde o método ILS_{SA-VNS} obteve os melhores resultados para instâncias grandes, para o modelo DVC o método IG_{SA}^2 foi o dominante.

Tabela 6.28 – Resumo NRV usando DRP1 (DVC1)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
NEH	0	10	1	0	10	1
XNEH	0	11	0	0	11	0
CHINS	0	11	0	0	11	0
XCHINS	0	10	1	0	10	1
HC1	0	10	1	0	10	1
HC2	0	11	0	0	11	0
Total	0	63	3	0	63	3

O desempenho do método IG_{SA}^2 foi superior aos demais métodos para todos os tamanhos de instância. Na Tabela 6.28B, é apresentado o resumo dos resultados por tamanho de instância.

Tabela 6.28B – Resumo NRV por tamanho de instância (DVC1)

	Pequena ($I_s = 20$)			Média ($I_s = 50$)			Grande ($I_s \geq 100$)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
NEH	0	3	0	0	3	0	0	4	1
XNEH	0	3	0	0	3	0	0	5	0
CHINS	0	3	0	0	3	0	0	5	0
XCHINS	0	3	0	0	3	0	0	4	1
HC1	0	3	0	0	3	0	0	4	1
HC2	0	3	0	0	3	0	0	5	0
Total	0	18	0	0	18	0	0	27	3

6.3.8. Análise DRP independente da heurística de iniciação

As avaliações dos resultados das execuções das três metaheurísticas independente da heurística de iniciação são feitas por meio das medidas DRP2 e DRP3.

Na Tabela 6.29, são apresentados os valores de DRP2 e DRP3 para as três metaheurística. Os melhores valores estão grifados.

Tanto para a medida DRP2 quanto para DRP3, o método IG_{SA}^2 foi o vencedor para todos os tamanhos de instância.

Tabela 6.29 – DRP2 e DRP3 para as três metaheurísticas (DVC1)

n	m	DRP2 (%)			DRP3 (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	0,82	<u>0,00</u>	2,08	1,80	<u>0,00</u>	1,97
20	10	3,08	<u>0,00</u>	2,21	2,86	<u>0,00</u>	2,86
20	20	3,10	<u>0,00</u>	2,76	3,95	<u>0,00</u>	3,49
50	5	2,24	<u>0,00</u>	1,97	2,15	<u>0,00</u>	2,12
50	10	2,72	<u>0,00</u>	2,19	2,49	<u>0,00</u>	2,11
50	20	1,33	<u>0,00</u>	1,50	2,19	<u>0,00</u>	1,88
100	5	0,53	<u>0,00</u>	0,53	1,65	<u>0,00</u>	1,55
100	10	1,08	<u>0,00</u>	1,01	1,84	<u>0,00</u>	1,71
100	20	0,22	<u>0,00</u>	0,21	1,03	<u>0,00</u>	0,80
200	10	0,51	<u>0,00</u>	0,38	1,15	<u>0,00</u>	1,08
200	20	0,46	<u>0,00</u>	0,38	1,24	<u>0,00</u>	1,23
NRV		0	11	0	0	11	0

6.3.9. Análise de variância (ANOVA)

Na Tabela 6.30, são apresentados os resultados da análise de variância (ANOVA) para os onze experimentos do conjunto de experimentos do PPIPD-NWFS-DVC1.

Na coluna 'Difere?', são identificados os experimentos cujos tratamentos (combinação de uma metaheurística e uma heurística de iniciação) diferem significativamente entre si. Este resultado sugere que a combinação de uma metaheurística e uma heurística de iniciação afeta significativamente os resultados para as instâncias maiores.

Para os experimentos que diferem, os métodos *post-hoc* Tukey e Fisher (LSD) são empregados para avaliar as diferenças entre os tratamentos. Os resultados das análises realizadas por ambos os métodos são apresentados no apêndice B.

Tabela 6.30 – Resultados ANOVA (DVC1)

n	m	F0	Fcrit	F0/Fcrit	α	Valor-p	Difere?
20	5	0,200	1,767	0,11	0,05	1,000	Não
20	10	0,261	1,767	0,15	0,05	0,998	Não
20	20	1,719	1,767	0,97	0,05	0,059	Não
50	5	0,611	1,767	0,35	0,05	0,873	Não
50	10	1,101	1,767	0,62	0,05	0,370	Não
50	20	0,969	1,767	0,55	0,05	0,501	Não
100	5	0,700	1,767	0,40	0,05	0,793	Não
100	10	1,467	1,767	0,83	0,05	0,132	Não
100	20	2,307	1,767	1,31	0,05	0,007	Sim
200	10	3,221	1,767	1,82	0,05	2,716E-04	Sim
200	20	3,074	1,767	1,74	0,05	4,620E-04	Sim

6.3.10. Verificação da hipótese – DVC1 versus DVC2

Um dos objetivos dos experimentos computacionais foi a verificação da hipótese matemática proposta no Capítulo 4. Para isso, os métodos de solução foram alterados para incluírem em seus algoritmos as expressões matemáticas da hipótese como critérios de otimização.

Na Tabela 6.31, é apresentado um resumo da contagem de resultados onde o modelo DVC1 (minimizar C_{max} e TFT) mostrou ser majoritariamente melhor (valor menor da FO) que o modelo DVC2 (minimizar TFT) para cada tamanho de instância e metaheurística. Estes resultados corroboram a hipótese.

No Apêndice A, são apresentadas as variações percentuais entre os valores das FO dos modelos DVC1 e DVC2 para cada tamanho de instância, metaheurística e heurística de iniciação.

Tabela 6.31 – Resultados onde DVC1 foi melhor que DVC2

n	m	Totais			Soma	%
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}		
20	5	2	1	2	5	27,8
20	10	6	3	4	13	72,2
20	20	2	6	2	10	55,6
50	5	4	6	3	13	72,2
50	10	6	6	6	18	100
50	20	5	6	5	16	88,9
100	5	3	1	3	7	38,9
100	10	6	6	6	18	100
100	20	6	6	6	18	100
200	10	6	6	6	18	100
200	20	6	6	6	18	100

6.4. Conclusões da análise estatística

Nesta seção são apresentadas as conclusões de todas as análises estatísticas (DRP, ANOVA e testes de Tukey e Fisher) realizadas nos resultados dos experimentos.

As principais conclusões obtidas da análise estatística dos resultados experimentais foram:

- 1) Na contagem final de resultados vencedores (NRV) obtidos das medidas DRP1 e DRP3, o método IG_{SA}^2 obteve desempenho superior aos demais métodos tanto para o modelo DFC2 quanto para os modelos DVC1 e DVC2.
- 2) Na contagem final de resultados vencedores (NRV) obtidos da medida DRP2, o método IG_{SA}^2 obteve desempenho superior aos demais métodos para o modelo DVC1.
- 3) Na contagem final dos resultados vencedores (NRV) obtidos da medida DRP2, o método ILS_{SA-VNS} obteve desempenho superior aos demais métodos tanto para o modelo DFC2 quanto para o modelo DVC2.
- 4) Na contagem final de resultados vencedores (NRV) obtidos da medida DRP2, O método IG_{SA}^2 ficou em segundo lugar para os modelos DFC2 e DVC2.
- 5) Na comparação dos resultados obtidos da medida DRP1, O método ILS_{SA-VNS} ficou em segundo lugar para os modelos DFC2 e DVC2.
- 6) Na avaliação geral baseada em NRV obtidos das três formulações da medida DRP, o método IG_{SA}^2 foi superior aos demais métodos.
- 7) Na contagem final de resultados vencedores (NRV) obtidos da medida DRP1 para o modelo DVC1, o método IG_{SA}^2 foi superior aos demais para todos os tamanhos de instância.
- 8) Na contagem final de resultados vencedores (NRV) obtidos da medida DRP1 para o modelo DVC2, o método IG_{SA}^2 foi superior aos demais para instâncias pequenas e grandes. Para instâncias médias, o método ILS_{SA-VNS} foi superior.
- 9) Na contagem final de resultados vencedores (NRV) obtidos da medida DRP1 para o modelo DFC2, o método IG_{SA}^2 foi notadamente superior aos demais para instâncias pequenas. Para instâncias médias, o método IG_{SA}^2 obteve resultados superiores aos demais, mas próximo do método IG_{SA}^1 . Para instâncias grandes,

os métodos IG_{SA}^2 e ILS_{SA-VNS} obtiveram resultados próximos, sendo superiores ao método IG_{SA}^1 .

- 10) Para o modelo DVC, a abordagem de otimização 1 obteve resultados melhores que a abordagem 2 para a quase totalidade dos tamanhos de instância.
- 11) A abordagem DFC1 não mostrou ser superior à abordagem DFC2. Esta última obteve resultados moderadamente melhores para a grande parte dos experimentos.
- 12) Para os três modelos DFC2, DVC1 e DVC2, a ANOVA demonstrou existir diferenças significativas entre os resultados dos métodos heurísticos para três ou mais tamanhos de instância. Para os modelos DFC2 e DVC2, as diferenças significativas ocorreram em quase todos os tamanhos de instância. Para o modelo DVC1, apenas as instâncias maiores apresentaram diferenças significativas. Este resultado sugere que a combinação de uma metaheurística e uma heurística de iniciação afeta significativamente os resultados para os diferentes tamanhos de instância avaliados.
- 13) Os testes de Tukey e Fisher demonstraram que:
 - a) Para o modelo DFC2, em geral, o método $IG_{SA}^1 - NEH$ teve o pior desempenho para quase todos os tamanhos de instância avaliados, seguido pelos métodos $IG_{SA}^1 - XNEH$, $ILS_{SA-VNS} - NEH$ e $ILS_{SA-VNS} - XNEH$.
 - b) Para o modelo DVC1, em geral, o método $IG_{SA}^1 - HC1$ teve o pior desempenho para todos os tamanhos de instância avaliados, seguido pelos métodos $IG_{SA}^1 - NEH$, $ILS_{SA-VNS} - HC1$, $ILS_{SA-VNS} - NEH$ e $ILS_{SA-VNS} - XNEH$.
 - c) Para o modelo DVC2, em geral, os métodos $IG_{SA}^1 - NEH$ e $IG_{SA}^1 - XNEH$ tiveram os piores desempenhos para quase todos os tamanhos de instância avaliados, seguidos pelos métodos $ILS_{SA-VNS} - NEH$ e $ILS_{SA-VNS} - XNEH$.
 - d) Para os três modelos DFC2, DVC1 e DVC2, nenhum método mostrou ser significativamente superior aos demais.
 - e) Para as execuções do modelo DVC1, Apesar de não ter obtido resultados significativamente superiores às outras combinações, o

método $IG_{SA}^2 - HC2$ foi relativamente melhor que os demais métodos para instâncias grandes.

- f) Para as execuções do modelo DVC2, apesar de não terem obtido resultados significativamente superiores às outras combinações, os métodos $IG_{SA}^2 - HC1$ e $IG_{SA}^2 - HC2$ foram relativamente melhores que os demais métodos para instâncias pequenas.
- g) Para as execuções do modelo DVC2, Apesar de não ter obtido resultados significativamente superiores às outras combinações, o método $IG_{SA}^1 - XCHINS$ foi relativamente melhor que os demais métodos para instâncias médias.
- h) Para as execuções do modelo DVC2, Apesar de não terem obtido resultados significativamente superiores às outras combinações, os métodos $IG_{SA}^2 - CHINS$ e $IG_{SA}^2 - XCHINS$ foram relativamente melhores que os demais métodos para instâncias grandes.

7. Conclusões e Proposições

As operações de produção e logística de distribuição são dois processos de negócio fundamentais da cadeia de suprimentos. A gestão eficiente destes processos requer a programação e controle detalhados das atividades inerentes. A Integração destas atividades tem sido apontada por muitos autores como o esforço que muitas empresas necessitam realizar para melhorar seus desempenhos operacionais, reduzir custos e simultaneamente atender aos níveis de serviço contratados.

Com o objetivo de contribuir quantitativamente para esta abordagem da gestão de operações, a Tese aqui desenvolvida estudou o problema de otimização da programação detalhada e integrada de operações de produção e distribuição em sistemas *no-wait flow shop* (PPIPD-NWFS) onde duas estratégias distintas de transporte foram consideradas.

A primeira estratégia, denominada 'data fixa de coleta' (DFC), é adotada quando a fábrica não tem controle sobre as datas ou intervalos de chegada dos veículos para a coleta dos produtos. Nesta situação, a fábrica deve se adequar à política de transporte definida pelo cliente ou transportador a qual impõe a periodicidade (datas fixas) de visita do veículo para realizar o transporte. A segunda estratégia, denominada 'data variável de coleta' (DVC), é adotada quando a fábrica tem controle sobre as datas de embarque e define qual o melhor momento para realizar a coleta dos produtos, permitindo assim total gestão sobre os despachos dos produtos. Em geral, as duas estratégias devem ser adotadas pela fábrica a depender do cliente ou mercado que atende.

O problema aqui estudado foi inspirado em um caso real de uma empresa manufatureira de componentes eletrônicos que busca integrar as atividades de planejamento e programação das áreas funcionais de produção e logística.

Para solucionar o problema de otimização (PPIP-D-NWFS), três métodos heurísticos foram desenvolvidos e analisados por técnicas estatísticas.

Fundamentado na pressuposição de que a melhoria na coordenação e sincronização das operações de produção e distribuição nos SIPD-NWFS levam à minimização dos tempos de entrega e atrasos, e conseqüentemente ao aumento do nível de serviço, os métodos computacionais propostos na pesquisa implementam algoritmos baseados nas metaheurísticas híbridas ILS_{SA-VNS} e IG_{SA} que programam de forma conjunta operações de produção e distribuição, respeitando as restrições relacionadas à fábrica, à ordem e à estratégia de transporte.

Os desempenhos dos métodos propostos foram analisados por meio de um conjunto de experimentos computacionais que buscaram investigar o impacto de fatores de influência e abordagens de otimização na performance do sistema de produção NWFS integrado, adotando as duas estratégias de transporte.

Uma das principais conclusões obtidas da análise estatística dos resultados foi que o método IG_{SA}^2 se mostrou superior aos demais para a principal medida usada no estudo. Em segundo lugar, ficou o método ILS_{SA-VNS} e em terceiro o método IG_{SA}^1 . Outra conclusão alcançada com o estudo foi a confirmação da hipótese proposta na pesquisa que mostrou ser verdadeira para a maior parte das instâncias simuladas.

Mais importante do que os próprios resultados experimentais é a conclusão de que os métodos propostos podem ser aplicados na programação integrada da produção e distribuição em sistemas reais, apoiando os programadores e gestores na avaliação da eficiência e eficácia das decisões de alocação de recursos de produção e transporte sob diferentes condições e configurações do sistema integrado.

Entre as várias aplicações dos métodos desenvolvidos nesta Tese podem ser citadas: a) no nível mais amplo de planejamento de operações, simular a execução de ordens no sistema de produção para determinar a melhor sequência de produção e entrega; b) diariamente, determinar a rotina de liberação de ordens e contratação de serviços de transporte; c) apoiar na decisão de atribuição de datas estimadas de

entrega (*promising*) e de aceitação de pedidos de clientes; d) ajudar na resolução de problemas conjuntos, tais como a reprogramação de produção e realocação de carga.

Como proposições relevantes para o aprofundamento da pesquisa atual e para futuros trabalhos, têm-se:

- 1) Avaliar os métodos atuais usando diferentes valores dos parâmetros de execução.
- 2) Realizar experimentos complementares utilizando outras instâncias (maiores) de Taillard e instâncias de *benchmark* propostas por outros autores (VALLADA; RUIZ; FRAMINAN, 2017).
- 3) Avaliar outros critérios de otimização, tais como o tempo médio de fluxo, a data máxima de entrega e o atraso de atendimento.
- 4) Estudar ‘variantes’ do sistema *flow shop*, tais como os sistemas *flow shop* flexíveis, com ou sem restrições de espera (*no-wait*), sistemas *flow shop* com operações de montagem e com tempos de *setup* dependentes da sequência.
- 5) Analisar o efeito do uso de outras regras de prioridade especializadas (desenvolvidas inicialmente com o propósito de minimizar o tempo de atendimento em sistemas de produção não integrados) adaptadas ao contexto dos SIPD, tais como o ATC (*Apparent Tardness Cost*) BD (*Bottleneck Dynamics*) e COVERT (*Cost over Time*). Estas regras seriam usadas em substituição as regras LPT e SPT na execução das heurísticas NEH e XNEH e para *benchmark*.
- 6) Desenvolver e testar novos métodos computacionais de otimização para o PPIPD-NWFS, tais como algoritmos genéticos (GA) e otimização por exames de partículas (PSO).
- 7) Aprofundar o estudo da hipótese proposta na pesquisa, buscando pela fundamentação matemática necessária à sua prova (se ela existir) e assim propor melhorias nos métodos heurísticos.
- 8) Realizar experimentos comparando os resultados da resolução do modelo em PLIM com aqueles obtidos pelos métodos heurísticos propostos.
- 9) Desenvolver possíveis valores de limites inferiores (*lower bound*) e superiores (*upper bound*) para o modelo matemático, avaliar suas qualidades e seus possíveis usos pelos métodos heurísticos.

Referências Bibliográficas

ADRESSI, A.; HASSANPOUR, S. T.; AZIZI, V. Solving group scheduling problem in no-wait flexible flowshop with random machine breakdown. **Decision Science Letters**, v. 5, n. 1, p. 157-168, 2016.

ADULYASAK, Y.; CORDEAU, J. F.; JANS, R. The production routing problem: a review of formulations and solutions algorithms. **Computers and Operations Research**, v. 55, p. 141-152, 2015.

AGNETIS, A.; ALOULOU, M. A.; KOVALYOV, M. Y. Integrated production scheduling and batch delivery with fixed departure times and inventory holding costs. **International Journal of Production Research**, v. 55, n. 20, p. 6193-6206, 2017.

ALDOWAISAN, T.; ALLAHVERDI, A. Minimizing total tardiness in no-wait flowshops. **Foundations of Computing and Decision Sciences**, v. 37, n. 3, p. 149-162, 2012.

ALLAHVERDI, A. A survey of scheduling problems with no-wait in process. **European Journal of Operational Research**, v. 255, n. 3, p. 665-686, 2016.

ALLAHVERDI, A. *et al.* A survey of scheduling problems with setup times or costs. **European Journal of Operational Research**, v. 187, n. 3, p. 985-1032, 2008.

ALLAHVERDI, A; AYDILEK, H. Total completion time with makespan constraint in no-wait flowshops with setup times. **European Journal of Operational Research**, v. 238, p. 724-734, 2014.

AYDILEK, H.; ALLAHVERDI, A. New heuristics for no-wait flowshops with performance measure of makespan and mean completion time. **International Journal of Operations Research**, v.10, n.1, p. 29-37, 2013.

AYDILEK, H.; ALLAHVERDI, A. Heuristics for no-wait flowshops with makespan subject to mean completion time. **Applied Mathematics and Computation**, v. 219, p. 351-359, 2012.

BAKER, K. R.; TRIETSCH, D. **Principles of Sequencing and Scheduling**, Nova York: John Wiley & Sons, 2009.

BASIR, S. A.; MAZDEH, M. M.; NAMAKSHENAS, M. Bi-level genetic algorithms for two-stage assembly flow-shop scheduling problem with batch delivery system. **Computers and Industrial Engineering**, v. 126, p. 217-231, 2018.

BIANCO, L.; GIORDANI, D. P. Flow shop no-wait scheduling with sequence dependent setup times and release dates. **INFOR**, v. 37, n. 1, p. 2-19, 1999.

BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. **Information Sciences**, v. 237, p. 82-117, 2013.

BRAEKERS, K.; RAMAEKERS, K.; NIEUWENHUYSE, I. The vehicle routing problem: state of art classification and review. **Computers and Industrial Engineering**, v. 99, p. 300-313, 2016.

BRÄYSY, O.; GENDREAU, M. Vehicle routing problem with time windows, part I: route construction and local search algorithms. **Transportation Science**, v. 39, n. 1, p. 104-118, 2005a.

BRÄYSY, O.; GENDREAU, M. vehicle routing problem with time windows, part II: metaheuristics. **Transportation Science**, v. 39, n. 1, p. 119-139, 2005b.

CAMPBELL, A. M.; WILSON, J. H. Forty years of periodic vehicle routing. **Networks**, v. 63, n. 1, p. 2-15, 2014.

CARLIER, J. Ordonnancements a contraintes disjonctives. **RAIRO – Recherche Operationnelle**, v. 12, p. 333-352, 1978.

CERNY, V. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. **Journal of Optimization Theory and Applications**, v. 45, p. 45-51, 1985.

CHANG, Y. C.; LEE, C. Y. Machine scheduling with job delivery coordination. **European Journal of Operational Research**, v. 158, n. 2, p. 470-487, 2004.

CHEN, H. K.; HSUEH, C. F.; CHANG, M. S. Production scheduling and vehicle routing with time windows for perishable food products. **Computers and Operations Research**, v. 36, p. 2311-2319, 2009.

CHEN, Z.-L. Integrated Production and Outbound Distribution Scheduling: Review and Extensions. **Operations Research**, v. 58, n. 1, p. 130–148, 2010.

CHEN, Z. L.; PUNDOOR, G. Order assignment and scheduling in a supply chain. **Operations Research**, v. 54, n. 3, p. 555-572, 2006.

CHEN, Z. L.; VAIRAKTARAKIS, L. Integrated scheduling of production and distribution operations. **Management Science**, v. 51, n. 4, p. 519-678, 2005.

CHIANG *et al.* A simulation/metaheuristic approach to newspaper production and distribution supply chain problems. **International Journal of Production Economics**, v. 121, p. 752-767, 2009.

CLARKE, G.; WRIGHT, J. Scheduling of vehicles from a central depot to a number of delivery points. **Operations Research**, v. 12, n. 4, p. 558-581, 1964.

DANTZIG, G. B.; RAMSER, J. H.; The truck dispatching problem. **Management Science**, v. 6, n. 1, p. 80-91, 1959.

DEVAPRIYA, P.; FERRELL, W.; GEISMAR, N. Integrated production and distribution scheduling with a perishable product. **European Journal of Operational Research**, v. 259, p. 906-916, 2017.

DIAS-MADROÑERO, M.; PEIDRO, D.; MULA, J. A review of tactical optimization models for integrated production and transport routing planning decisions. **Computers and Industrial Engineering**, v. 88, p. 518-535, 2015.

DING, J-Y. *et al.* An improved iterated greedy algorithm with a taboo-based reconstruction strategy for the no-wait flowshop scheduling problem. **Applied Soft Computing Journal**, v. 30, p. 604-613, 2015a.

DING, J-Y. *et al.* Accelerated methods for total tardiness minimization in no-wait flowshops. **International Journal of Production Research**, v. 53, n.4, p. 1002-1018, 2015b.

DONG, J. *et al.* No-wait two-stage flowshop problem with multi-task flexibility of the first machine. **Information Science**, v. 544, p. 25-38, 2021.

DUBOIS-LACOSTE, J.; PAGNOZZI, F.; STÜTZLE, T. An iterated greedy algorithm with optimization of partial solutions for the makespan permutation flowshop problem. **Computers and Operations Research**, v.81, p. 160-166, 2017.

EGLESE, R. W. Simulated annealing: a tool for operations research. **European Journal of Operational Research**, v. 46, p. 271-281, 1990.

EHM, J.; FREITAG, M. The benefit of integrating production and transport scheduling. **Procedia CIRP**, v. 41, p. 585-590, 2016.

FAHIMNIA, B. *et al.* A review and critique on integrated production-distribution planning models and techniques. **Journal of Manufacturing Systems**, v. 32. n. 1, p. 1-19, 2013.

FERNANDEZ-VIAGAS, V.; FRAMINAN, J. M. A best-of-breed iterated greedy for the permutation flowshop scheduling problem with makespan objective. **Computers and Operations Research**, v. 112, n. 104767, 2019.

FERNANDEZ-VIAGAS, V.; FRAMINAN, J. M. A beam-search-based constructive heuristic for the PFSP to minimise total flowtime. **Computers and Operations Research**, v. 81, p. 167-177, 2017.

FERNANDEZ-VIAGAS, V.; FRAMINAN, J. M. On insertion tie-breaking rules in heuristics for the permutation flowshop scheduling problem. **Computers and Operations Research**, v. 45, p. 60-67, 2014.

FERNANDEZ-VIAGAS, V.; RUIZ, R.; FRAMINAN, J. M. A new vision of approximate methods for the permutation flowshop to minimize makespan: State-of-the-art and computational evaluation. **European Journal of Operational Research**, v. 257, n. 3, p. 707-721, 2017.

FINK, A.; VOß, S. Solving the continuous flow-shop scheduling problem by metaheuristics. **European Journal of Operational Research**, v. 151, n. 2, p. 400-414, 2003.

FRAMINAN, J. M.; GUPTA, J. N. D.; LEISTEN, R. A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. **Journal of the Operational Research Society**, v. 55, p. 1243-1255, 2004.

FRAMINAN, J. M.; LEISTEN, R.; RAJENDRAN, C. Different initial sequences for the heuristic of Nawaz, Ensore and Ham to minimize makespan, idle time e flowtime in the static permutation flowshop sequencing problem. **International Journal of Production Research**, v. 41, n. 1, p. 121-148, 2003.

FU, B.; HUO, Y.; ZHAO, H. Coordinated scheduling of production and delivery with production window and delivery capacity constraints. **Theoretical Computer Science**, v.422, p. 39-51, 2012.

GAMS. **GAMS documentation center**, 2021. Disponível em <https://www.gams.com/latest/docs/>. Acesso em: 02 mar. 2021.

GARCIA, J. M.; LOZANO, S. Production and vehicle scheduling for ready-mix operations. **Computer and Industrial Engineering**, v. 46, n. 4, p. 803-816, 2005.

GAREY, M. R.; JOHNSON, D. S.; SETHI, R. The complexity of flowshop and jobshop scheduling. **Mathematics of Operations Research**, v. 1, n. 2, p. 117-129, 1976.

GEISMAR, H. N *et al.* The integrated production and transportation scheduling problem for a product with a short lifespan. **INFORMS Journal on Computing**, v. 20, n. 1, p. 21-33, 2008.

GRABOWSKI, J.; PEMPERA, J. Some local search algorithms for no-wait flow-shop problem with makespan criterion. **Computer and Operations Research**, v. 32, n. 8, p.2197-2212, 2005.

GU, J.; GU, M.; GU, X. A mutualism quantum genetic algorithm to optimize the flow shop scheduling with pick-up and delivery considerations. **Mathematical Problems in Engineering**, v. 2015, n. 387082, 2015.

GUPTA, J. N. D; STAFFORD JR., E. F. Flowshop scheduling after five decades. **European Journal of Operational Research**, v. 169, n. 3, p. 699-711, 2006.

GUROBI. **Gurobi Documentation**, 2020. Disponível em: <https://www.gurobi.com/documentation/>. Acesso em: 21 abr. 2021.

HALL, N. G.; LESAOANA, M.; POTTS, C. N. Scheduling with fixed delivery dates. **Operations Research**, v. 49, n. 1, p. 134-144, 2001.

HALL, N. G.; SRISKANDARAJAH, C. A Survey of machine scheduling problems with blocking and no-wait in process. **Operations Research**, v. 44, n. 3, p. 423-528, 1996.

HAN, D. *et al.* Integrated production, inventory, and outbound distribution operations with fixed departure times in a three-stage supply chain. **Transportation Research Part E**, v. 125, p. 334-347, 2019.

HANSEN, P.; MLADENOVIC, N. Variable neighborhood search: principles and applications. **European Journal of Operational Research**, v. 130, n. 1, p. 449-467, 2001.

HAUPT, R. A survey on priority rule-based scheduling. **OR Spektrum**, v. 11, n. 1, p. 3-16, 1989.

HELLER, J. Some numerical experiments for an *MxJ* flow shop and its decision theoretical aspects. **Operations Research**, v.8, n. 2, p.178-184, 1960.

HUO, Y.; LEUNG, J. Y. T.; WANG, X. Integrated production and delivery scheduling with disjoint windows. **Discrete Applied Mathematics**, v. 158, p. 921-931, 2010.

IBM. **IBM Ilog Cplex Optimization Studio V12.10.0 documentation**, 2020. Disponível em: <https://www.ibm.com/docs/en/icos/12.10.0>. Acesso em: 20 abr. 2021.

JACOBS, L. W.; BRUSCO, M. J. A local search heuristic for large set-covering problems. **Naval Research Logistics**, v. 42, p. 1129-1140, 1995.

JARBOUI B.; EDDALY M.; SIARRY, P. An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems. **Computers and Operations Research**, v. 36, p. 2638-2646, 2009.

KIRKPATRICK, C. G.; VECCHI, M. Optimization by simulated annealing. **Science**, v. 220, p. 671-680, 1983.

KOMAKI, M.; MALAKOOTI, B. General variable neighborhood search algorithm to minimize makespan of the distributed *no-wait flow shop* scheduling problem. **Production Management**, v. 11, p. 315-329, 2017.

KUMAR, R. *et al.* Quantitative approaches for the integration of production and distribution planning in the supply chain: a systematic literature review. **International Journal of Production Research**, v. 58, n. 11, p. 3527-3553, 2020.

KUMAR, S. N.; PANNEERSELVAM, R. A Survey on the vehicle routing problem and its variants. **Intelligent Information Management**, v.4, p. 66-74, 2012.

LAPORTE, G. Fifty years of vehicle routing. **Transportation Science**, v. 43, n. 4, p. 408-416, 2009.

LEE, J. *et al.* The nuclear medicine production and delivery problem. **European Journal of Operational Research**, v. 236, n. 2, p. 461-472, 2014.

LENSTRA, J. K.; RINNOOY-KHAN, A. H. G. Complexity of vehicle routing and scheduling problems. **Networks**, v. 11, p. 221-227, 1981.

LEUNG, J. Y. T.; CHEN, Z. L. Integrated production and distribution with fixed delivery departure dates. **Operations Research Letters**, v. 41, p. 290-293, 2013.

LI, F.; CHEN, Z.; TANG, L. Integrated production, inventory and delivery problems: complexity and algorithms. **INFORMS Journal on Computing**, v. 29, n. 2, p. 232-250, 2017.

LI, K. *et al.* Integrated production and delivery with single machine and multiple vehicles. **Expert Systems with Applications**, v. 57, p. 12-20, 2016.

LI, X. *et al.* An iterated greedy heuristic for no-wait flow shops with sequence dependent setup times, learning and forgetting effects. **Information Sciences**, v. 453, p. 408-425, 2018.

LI, K. P.; GANESAN, V. K.; SIVAKUMAR, A. I. Synchronized scheduling of assembly and multi-destination air-transportation in a consumer electronics supply chain. **International Journal of Production Research**, v. 43, p. 2671-2685, 2005.

LI, K.; JIA, Z. H.; LEUNG, J. Y. T. Integrated production and delivery on parallel batching machines. **European Journal of Operational Research**, v. 247, p. 755- 763, 2015.

LI, S.; LI, M. Integrated production and distribution scheduling problems related to fixed delivery departure dates and weights of late orders. **Scientific World Journal**, v. 2015, n. 804968, 2015.

LI, S.; LI, M. Integrated production and distribution scheduling problems related with fixed delivery departure dates and number of late orders. **Journal of Inequalities and Applications**, v. 2014, n. 409, 2014.

LI, K; SIVAKUMAR, A. I.; GANESAN, V. K. Complexities and algorithms for synchronized scheduling of parallel machine assembly and air transportation in consumer electronics supply chain. **European Journal of Production Research**, v. 187, p. 442-455, 2008.

LI, X.; WANG, Q.; WU, C. Heuristic for no-wait flow shops with makespan minimization. **International Journal of Production Research**, v. 46, n. 9, p. 2519-2530, 2008.

LIN, C. *et al.* Survey of green vehicle routing problem: past and future trends. **Expert Systems with Application**, v. 41, n. 4, p. 1118-1138, 2014.

LIN, S. W.; YING, K. C. Optimization of makespan for no-wait flowshop scheduling problems using efficient metaheuristics. **Omega**, v. 64, p. 115-125, 2016.

LIU, C. J.; HSU, C. Dynamic job shop scheduling with fixed interval deliveries. **Production Engineering**, v. 9, n. 3, p. 377-391, 2015.

LIU, J.; REEVES, C. R. Constructive and composite heuristic solutions to the $P//\sum C_j$ scheduling problem. **European Journal of Operational Research**, v. 132, n. 1, p. 439-452, 2001.

LIU, Z.; ZHANG, Y.; LI, M. Integrated scheduling of ready-mix concrete production and delivery. **Automation in Construction**, v. 48, p. 31-43, 2014.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: GENDREAU, M.; POTVIN, J.-Y. **Handbook of Metaheuristics**, ed. 2, Nova York: Springer Science, 2016, p. 363–397.

LOW, C. *et al.* Coordination of production scheduling and delivery problems with heterogeneous fleet. **International Journal of Production Economics**, v. 153, p. 139-148, 2014.

LU, L.; YUAN, J.; ZHANG, L. Single machine scheduling with release dates and job delivery to minimize the makespan. **Theoretical Computer Science**, v. 393, n. 1-3, p. 102-108, 2008.

MA, X.; WANG, S.; BAI, Q. Coordination of production scheduling and vehicle routing problems for perishable food products. **International Journal of Internet Manufacturing and Services**, v. 6, n. 1, p. 79-96, 2019.

MACCHIAROLI, S.; MOLÈ, S.; RIEMMA, S. Modelling and optimization of industrial manufacturing processes subject to no-wait constraints. **International Journal of Production Research**, v. 37, n. 11, p. 2585-2607, 1999.

MARANDI, F.; GHOMI, S. M. T. F. Integrated multi-factory production and distribution scheduling applying vehicle routing approach. **International Journal of Production Research**, v. 57, n. 3, p. 722-748, 2019.

MARANDI, F.; ZEGORDI, S. H. Integrated production and distribution scheduling for perishable products. **Scientia Iranica**, v. 24, n. 4, p. 2105-2118, 2017.

MARCHIAROLI, E.; STEENBEEK, A. An evolutionary algorithm for large scale set covering problems with application to airline crew scheduling. In: CAGNONI *et al.* **Real-world applications of evolutionary computing**, Berlin: Springer-Verlag, v. 1803, p. 367-381, 2000.

MENSENDIEK, A.; GUPTA, J. N. D; HERRMANN, J. Scheduling identical parallel machines with fixed delivery dates to minimize total tardiness. **European Journal of Operational Research**, v. 243, n. 2, p. 514-522, 2015.

MARTINS, L. C. *et al.* Combining production and distribution in supply chains: the hybrid flow shop vehicle routing problem. **Computers and Industrial Engineering**, v. 159, n. 107486, 2021.

METROPOLIS, N. *et al.* Equation for state calculations by fast computing machines. **Journal of Chemical Physics**, v. 21, p. 1087-1090, 1953.

MLADENOVIC, N.; HANSEN, P. Variable neighborhood search. **Computers and Operations Research**, v. 24, n. 11, p. 1097-1100, 1997.

MOHAMMADI, S; CHERAGHALIKHANI, A.; RAMEZANIAN, R. A joint scheduling of production and distribution operations in a flow shop manufacturing system, **Scientia Iranica**, v. 25, n. 2, p. 911-930, 2018.

MONTGOMERY, D. C. **Design and Analysis of Experiments**, ed. 9, Nova York: John Wiley & Sons, 2017.

MONTOYA-TORRES, J. R. *et al.* A literature review on the vehicle routing problem with multiple depots. **Computers and Industrial Engineering**, v. 79, p. 115-129, 2015.

MOONS, S. *et al.* Integrating production scheduling and vehicle routing decisions at the operational decision level: A review and discussion. **Computers and Industrial Engineering**, v. 104, p. 224-245, 2017.

NADERI, B. *et al.* Multi-objective no-wait flowshop scheduling problems: models and algorithms. **International Journal of Production Research**, v. 50, n. 10, p. 2592-2608, 2012.

NADERI, B.; RUIZ, R.; ZANDIEH, M. Algorithms for a realistic variant of flowshop scheduling. **Computers and Operations Research**, v. 37, n. 2, p. 236-246, 2010.

NAGANO, M. S.; MIYATA, H. H. A high-quality solution constructive heuristic for no-wait flow shop scheduling Problem. **Industrial Engineering & Management Systems**, v. 15, n. 3, p. 206–214, 2016a.

NAGANO, M. S.; MIYATA, H. H. Review and classification of constructive heuristics mechanisms for no-wait flow shop problem. **International Journal of Advanced Manufacturing Technology**, v. 86, p. 2161–2174, 2016b.

NAGANO, M. S.; MIYATA, H. H.; ARAÚJO, D. C. A constructive heuristic for total flowtime minimization in a no-wait flowshop with sequence-dependent setup times. **Journal of Manufacturing Systems**, v. 36, p. 224-230, 2015.

NAWAZ, M.; ENSCORE, E. E.; HAM, I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. **Omega**, v. 11, n. 1, p. 91-95, 1983.

NEUFELD, J. S.; GUPTA, N. D.; BUSCHER, U. A comprehensive review of flowshop group scheduling literature. **Computers and Operations Research**, v. 70, p. 56–74, 2016.

NIKOLAEV, A. G.; JACOBSON, S. H. Simulated annealing. In: GENDREAU, M.; POTVIN, J.-Y. **Handbook of Metaheuristics**, 2. ed., Nova York: Springer Science & Business, cap. 1, p. 1-40, 2010.

NOROOZI, A. *et al.* Coordinating order acceptance and integrated lot streaming-batch delivery scheduling considering third party logistics. **Uncertain Supply Chain Management**, v.7, p. 73-96, 2019.

PAN, J. C. H. *et al.* Coordinating scheduling with batch deliveries in a two-machine flow shop. **International Journal of Advanced Manufacturing Technology**, v. 40, p. 607-616, 2009.

PAN, Q. K.; RUIZ, R. A comprehensive review and evaluation of permutation flowshop heuristics to minimize flowtime. **Computers and Operations Research**, v. 40, n. 1, p. 117-128, 2013.

PAN, Q.; WANG, W.; ZHU, J. Effective heuristics based on differential evolution and variable neighborhood search for no-wait flow shop. **China Mechanical Engineering**, v. 17. n. 2, p. 157-160, 2006.

PARRAGH, S. N.; DOERNER, K. F.; HARTL, R. F. A survey on pickup and delivery problems - Part 1: Transportation between customers and depot. **Journal für Betriebswirtschaft**, v. 58, n. 1, p. 21-51, 2008a.

PARRAGH, S. N.; DOERNER, K. F.; HARTL, R. F. A survey on pickup and delivery problems - Part 2: Transportation between pickup and delivery locations. **Journal für Betriebswirtschaft**, v. 58, n. 2, p. 81-117, 2008b.

PILLAC, V. *et al.* A review of dynamic vehicle routing problems. **European Journal of Operational Research**, v. 225, n. 1, p. 1-11, 2013.

PIRLOT, M. General local search methods. **European Journal of Operational Research**, v.92, p. 493-511, 1996.

PORSELVI, S.; BALAJI, A. N.; JAWAHAR, N. Artificial immune system and particle swarm optimization algorithms for an integrated production and distribution scheduling problem. **International Journal of Logistics Systems and Management**, v. 30, n. 1, p. 31-68, 2018.

PUNDOOR, G.; CHEN, Z. L. Scheduling a production-distribution system to optimize the tradeoff between delivery tardiness and distribution cost. **Naval Research Logistics**, v. 52, n. 6, p. 571–589, 2005.

RAMEZANIAN R.; MOHAMMADI, S.; CHERAGHALIKHANI, A. Toward an integrated modeling approach for production and delivery operations in flow shop system: trade-off between direct and routing delivery methods, **Journal of Manufacturing Systems**, v. 44, n. 1, p. 79-92, 2017.

REDDI, S. S.; RAMAMOORTHY, C. V. On the flow-shop sequencing problem with no wait in process. **Journal of the Operational Research Society**, v. 23, n. 2, p. 323-331, 1972.

REEVES, C. R. A genetic algorithm for flowshop sequencing. **Computers and Operations Research**, v. 22, n. 1, p. 5-13, 1995.

REIMANN, M.; NETO, R. T.; BOGENDORFER, E. Joint optimization of production planning and vehicle routing problems: a review of existing strategies. **Pesquisa Operacional**, v. 34, p. 189-214, 2014.

RIBAS, I.; LEISTEN, R.; FRAMIÑAN, J. M. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. **Computers and Operations Research**, v. 37, n. 8, p. 1439-1454, 2010.

RÖCK, H. The three-machine no-wait flowshop problem is NP-complete. **Journal of the Association of Computer Machinery**, v.31, pp. 336-345, 1984.

RUIZ, R.; ALLAHVERDI, A. New heuristics for no-wait flow shops with a linear combination of makespan and maximum lateness. **International Journal of Production Research**, v. 47, n. 20, p. 5717-5738, 2009.

RUIZ, R.; ALLAHVERDI, A. Some effective heuristics for no-wait flowshops with setup times to minimize total completion time. **Annals of Operations Research**, v. 156, n. 1, p. 143–171, 2007.

RUIZ, R.; STÜTZLE, T. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. **European Journal of Operational Research**, v. 177, n. 3, p. 2033–2049, 2007.

SAMARGHANDI, H.; ELMEKKAWY, T. Y. An efficient hybrid algorithm for the two-machine no-wait flow shop problem with separable setup times and single server. **European Journal of Industrial Engineering**, v. 5, n. 2, 2011.

SAPKAL, S. U.; LAHA, D. A heuristic for no-wait flow shop scheduling. **International Journal Advanced Manufacturing Technology**, v. 68, p.1327-1338, 2013.

SCHOLZ-REITER, B. S. *et al.* An approach for the sustainable integration of production and transportation scheduling. **International Journal of Logistics Systems and Management**, v. 10, n. 2, p. 158-179, 2011.

SCHOLZ-REITER, B.; FRAZZON, E. M.; MAKUSCHEWITZ, T. Integrating manufacturing and logistics systems along global supply chains. **CIRP Journal of Manufacturing Science and Technology**, v. 2, n. 3, p. 216-223, 2010.

SEDDIK, Y.; GONZALES, C.; KEDAD-SIDHOUM, S. Single machine scheduling with delivery dates and cumulative payoffs. **Journal on Scheduling**, v. 16, p. 313-329, 2013.

SHAO, W.; PI, D.; SHAO, Z. A hybrid iterated greedy algorithm for the distributed no-wait flow shop scheduling problem. **Proceedings on IEEE Congress on Evolutionary Computation**, n. 7969289, p. 9-16, 2017.

SOUKHAL, A.; OULAMARA, A.; MARTINEAU, P. Complexity of flow shop scheduling problems with transportation constraints. **European Journal of Operational Research**, v. 161, n. 1, p. 32-41, 2005.

SRISKANDARAJAH, C.; LADET, P. Some no-wait shops scheduling problems: complexity aspect. **European Journal of Operational Research**, v. 24, n. 3, p. 424-438, 1986.

STECKE, K. E.; ZHAO, X. Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode. **Manufacturing and Service Operations Management**, v. 9, n.2, p. 206-224, 2007.

STÜTZLE, T. Local search algorithms for combinatorial problems: analysis, improvements, and new applications. In: **Dissertations in Artificial Intelligence**, Amsterdam: IOS Press, cap. 5, p. 119-158, 1999.

TAILLARD, E. Benchmarks for basic scheduling problems. **European Journal of Operational Research**, v. 64, n. 2, p. 278-285, 1993a.

TAILLARD, E. Scheduling instances, 1993b. Disponível em: <http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>. Acesso em: 10 maio 2021.

TAVARES-NETO, R. F.; NAGANO, M. S. An iterated greedy approach to integrate production by multiple parallel machines and distribution by a single capacitated vehicle. **Swarm and Evolutionary Computation**, v. 44, p. 612-621, 2019.

ULLRICH, C. A. Integrated machine scheduling and vehicle routing with time windows. **European Journal of Operational Research**, v. 227, n. 1, p. 152-165, 2013.

VALLADA, E.; RUIZ, R.; FRAMINAN, J. M. New hard benchmark for flowshop scheduling problems minimising makespan. **European Journal of Operational Research**, v. 240, n. 3, p. 666-677, 2017.

WANG, K. *et al.* Permutation flow shop with batch delivery to multiple customers in supply chains. **IEEE Transactions on Systems, Man and Cybernetics: Systems**, v. 48, n. 10, p. 1826-1837, 2018.

WANG, K. *et al.* Coordinated scheduling of production and transportation in a two-stage assembly flowshop. **International Journal of Production Research**, v. 54, n. 22, p. 6891–6911, 2016.

WANG, T. Y.; YANG, Y. H.; LIN, H. J. Comparison of scheduling efficiency in two/three-machine no-wait flow shop problem using simulated annealing and genetic algorithm. **Asia Pacific Journal of Operational Research**, v. 23, n. 1, p. 41-59, 2006.

WISMER, D. A. Solution of the flowshop-scheduling problem with no intermediate queues. **Operations Research**, v. 20, n.3, p. 689-697, 1972.

YAGMUR, E.; KESEN, S. E. Memetic algorithm for joint production and distribution scheduling with due dates. **Computer and Industrial Engineering**, v. 142, n. 106342, 2020.

YE, H.; LI, W.; MIAO, E. An effective heuristic for no-wait flow shop production to minimize makespan. **Journal of Manufacturing Systems**, v. 40, p. 2-7, 2016.

YENISEY, M. M.; YAGMAHAN, B. Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. **Omega**, v. 45, p. 119-135, 2014.

ZHONG, W.; CHEN, Z. L.; CHEN, M. Integrated production and distribution scheduling with committed delivery dates. **Operations Research Letters**, v. 38, p. 133-138, 2010.

ZOU, X. *et al.* A coordinated algorithm for integrated production scheduling and vehicle routing problem. **International Journal of Production Research**, v. 56, n. 15, p. 5005-5024, 2018.

Apêndice A – Análise dos Resultados Experimentais (DVC2)

A.1 Análise DRP1 usando a heurística NEH na iniciação

Na Tabela A.1, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando NEH para iniciação e b) o percentual de melhoria sobre o valor do NEH tomado como *benchmark*. Na Tabela A.2, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Tabela A.1 – DRP1 e percentual de melhoria usando NEH (DVC2)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	12,26	<u>0,00</u>	6,49	5,76	<u>16,01</u>	10,52
20	10	11,86	<u>0,00</u>	7,96	5,35	<u>15,40</u>	8,66
20	20	8,37	<u>0,00</u>	5,82	4,71	<u>12,04</u>	6,97
50	5	9,99	<u>0,00</u>	7,07	5,98	<u>14,46</u>	8,49
50	10	7,63	<u>0,00</u>	5,04	4,62	<u>11,37</u>	6,89
50	20	5,29	<u>0,00</u>	1,23	4,00	<u>8,70</u>	7,62
100	5	8,72	<u>0,00</u>	7,11	3,67	<u>11,33</u>	5,13
100	10	6,47	<u>0,00</u>	3,96	3,10	<u>8,97</u>	5,38
100	20	3,32	<u>0,00</u>	1,18	2,25	<u>5,38</u>	4,26
200	10	5,43	<u>0,00</u>	4,54	1,64	<u>6,66</u>	2,47
200	20	2,78	<u>0,00</u>	2,15	1,61	<u>4,27</u>	2,22
NRV		0	11	0	0	11	0

Tabela A.2 – Estatísticas complementares usando NEH (DVC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	4,85	<u>0,00</u>	3,15	2,55	<u>6,97</u>	4,10
MS	7,47	<u>0,00</u>	4,78	3,88	<u>10,42</u>	6,24
DP	3,16	<u>0,00</u>	2,41	<u>1,58</u>	3,99	2,61
CV	0,42	-	0,50	0,41	<u>0,38</u>	0,42

A.2 Análise DRP1 usando a heurística XNEH na iniciação

Na Tabela A.3, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando XNEH para iniciação e b) o percentual de melhoria sobre o valor do XNEH tomado como *benchmark*. Na Tabela A.4, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Tabela A.3 – DRP1 e percentual de melhoria usando XNEH (DVC2)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	9,26	<u>0,00</u>	5,65	3,79	<u>11,87</u>	6,93
20	10	6,28	<u>0,00</u>	4,95	5,20	<u>10,88</u>	6,44
20	20	5,17	<u>0,00</u>	5,67	5,21	<u>9,80</u>	4,82
50	5	7,27	<u>0,00</u>	6,29	4,42	<u>10,87</u>	5,28
50	10	3,99	<u>0,00</u>	2,84	4,06	<u>7,72</u>	5,11
50	20	2,40	<u>0,00</u>	1,22	3,00	<u>5,17</u>	4,09
100	5	7,37	<u>0,00</u>	7,21	3,08	<u>9,71</u>	3,21
100	10	1,59	<u>0,00</u>	0,71	1,68	<u>3,24</u>	2,54
100	20	0,65	0,48	<u>0,00</u>	2,08	2,23	<u>2,71</u>
200	10	2,96	<u>0,00</u>	2,73	1,11	<u>3,92</u>	1,33
200	20	1,07	<u>0,00</u>	0,48	1,04	<u>2,09</u>	1,62
NRV		0	10	1	0	10	1

Tabela A.4 – Estatísticas complementares usando XNEH (DVC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	2,28	<u>0,08</u>	1,65	1,96	<u>4,00</u>	2,56
MS	4,36	<u>0,04</u>	3,43	3,15	<u>7,05</u>	4,01
DP	2,90	<u>0,14</u>	2,61	<u>1,52</u>	3,78	1,88
CV	<u>0,67</u>	3,50	0,76	0,48	0,54	<u>0,47</u>

A.3 Análise DRP1 usando a heurística CHINS na iniciação

Na Tabela A.5, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando CHINS para iniciação e b) o percentual de melhoria sobre o valor do CHINS tomado como *benchmark*. Na Tabela A.6, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Tabela A.5 – DRP1 e percentual de melhoria usando CHINS (DVC2)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	1,91	1,33	<u>0,00</u>	4,37	4,87	<u>6,15</u>
20	10	2,78	<u>0,00</u>	1,00	2,24	<u>4,90</u>	3,96
20	20	2,35	<u>0,00</u>	1,18	1,67	<u>3,92</u>	2,79
50	5	0,73	0,47	<u>0,00</u>	2,60	2,86	<u>3,30</u>
50	10	1,52	0,64	<u>0,00</u>	1,83	2,66	<u>3,31</u>
50	20	0,01	<u>0,00</u>	0,19	1,41	<u>1,41</u>	1,23
100	5	0,98	<u>0,00</u>	0,38	1,28	<u>2,24</u>	1,87
100	10	1,29	<u>0,00</u>	0,56	1,08	<u>2,33</u>	1,79
100	20	0,85	<u>0,00</u>	0,22	0,99	<u>1,82</u>	1,61
200	10	1,84	<u>0,00</u>	1,13	0,49	<u>2,28</u>	1,17
200	20	1,79	<u>0,00</u>	1,18	0,23	<u>1,97</u>	0,83
NRV		0	8	3	0	8	3

Tabela A.6 – Estatísticas complementares usando CHINS (DVC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	1,42	<u>0,05</u>	0,76	0,85	<u>2,18</u>	1,49
MS	1,46	<u>0,22</u>	0,53	1,65	<u>2,84</u>	2,55
DP	0,79	<u>0,43</u>	0,50	<u>1,14</u>	1,20	1,57
CV	<u>0,54</u>	1,95	0,94	0,69	<u>0,42</u>	0,62

A.4 Análise DRP1 usando a heurística XCHINS na iniciação

Na Tabela A.7, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando XCHINS para iniciação e b) o percentual de melhoria sobre o valor do XCHINS tomado como *benchmark*. Na Tabela A.8, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Tabela A.7 – DRP1 e percentual de melhoria usando XCHINS (DVC2)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	1,93	<u>0,00</u>	1,35	3,63	<u>5,44</u>	4,16
20	10	2,28	<u>0,00</u>	1,46	2,20	<u>4,37</u>	3,00
20	20	1,07	<u>0,00</u>	0,33	1,48	<u>2,51</u>	2,20
50	5	1,18	1,01	<u>0,00</u>	2,15	2,32	<u>3,30</u>
50	10	0,15	1,08	<u>0,00</u>	2,14	1,23	<u>2,28</u>
50	20	0,53	0,85	<u>0,00</u>	1,09	0,78	<u>1,61</u>
100	5	0,47	0,57	<u>0,00</u>	1,44	1,34	<u>1,90</u>
100	10	0,35	0,12	<u>0,00</u>	1,22	1,45	<u>1,57</u>
100	20	0,07	0,13	<u>0,00</u>	1,11	1,05	<u>1,17</u>
200	10	0,44	<u>0,00</u>	0,38	0,81	<u>1,25</u>	0,87
200	20	0,40	0,07	<u>0,00</u>	0,62	0,94	<u>1,02</u>
NRV		0	4	7	0	4	7

Tabela A.8 – Estatísticas complementares usando XCHINS (DVC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	0,43	0,22	<u>0,11</u>	1,03	1,24	<u>1,35</u>
MS	0,81	0,35	<u>0,32</u>	1,63	2,06	<u>2,10</u>
DP	0,73	<u>0,44</u>	0,56	<u>0,86</u>	1,52	1,03
CV	<u>0,90</u>	1,26	1,75	0,53	0,74	<u>0,49</u>

A.5 Análise DRP1 usando a heurística HC1 na iniciação

Na Tabela A.9, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando HC1 para iniciação e b) o percentual de melhoria sobre o valor do HC1 tomado como *benchmark*. Na Tabela A.10, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Tabela A.9 – DRP1 e percentual de melhoria usando HC1 (DVC2)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	4,09	<u>0,00</u>	3,26	2,56	<u>6,45</u>	3,32
20	10	4,93	<u>0,00</u>	4,78	1,99	<u>6,60</u>	2,15
20	20	3,07	<u>0,00</u>	1,84	2,25	<u>5,14</u>	3,40
50	5	2,74	<u>0,00</u>	1,66	3,71	<u>6,17</u>	4,71
50	10	0,94	1,11	<u>0,00</u>	3,35	3,18	<u>4,24</u>
50	20	0,37	0,45	<u>0,00</u>	1,93	1,80	<u>2,28</u>
100	5	0,57	<u>0,00</u>	0,05	2,56	<u>3,09</u>	3,06
100	10	1,19	<u>0,00</u>	0,22	1,54	<u>2,67</u>	2,49
100	20	0,46	<u>0,00</u>	0,05	1,37	<u>1,82</u>	1,77
200	10	0,80	<u>0,00</u>	0,70	1,26	<u>2,05</u>	1,35
200	20	0,65	<u>0,00</u>	0,29	0,84	<u>1,48</u>	1,19
NRV		0	9	2	0	9	2

Tabela A.10 – Estatísticas complementares usando HC1 (DVC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	0,90	<u>0,08</u>	0,45	1,47	<u>2,25</u>	1,90
MS	1,80	<u>0,14</u>	1,17	2,12	<u>3,68</u>	2,72
DP	1,62	<u>0,35</u>	1,59	0,88	2,02	1,14
CV	<u>0,90</u>	2,50	1,36	<u>0,42</u>	0,55	<u>0,42</u>

A.6 Análise DRP1 usando a heurística HC2 na iniciação

Na Tabela A.11, são apresentados: a) os valores de DRP1 para as três metaheurísticas usando HC2 para iniciação e b) o percentual de melhoria sobre o valor do HC2 tomado como *benchmark*. Na Tabela A.12, são apresentadas as estatísticas complementares. Os melhores valores estão grifados.

Tabela A.11 – DRP1 e percentual de melhoria usando HC2 (DVC2)

n	m	DRP1 (%)			Melhoria (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	2,27	<u>0,00</u>	0,37	2,43	<u>4,53</u>	4,17
20	10	1,95	<u>0,00</u>	1,50	1,85	<u>3,65</u>	2,31
20	20	1,51	<u>0,00</u>	1,22	1,57	<u>3,04</u>	1,84
50	5	1,62	0,86	<u>0,00</u>	2,37	3,11	<u>3,92</u>
50	10	0,30	0,50	<u>0,00</u>	2,67	2,47	<u>2,95</u>
50	20	0,93	1,09	<u>0,00</u>	1,66	1,51	<u>2,57</u>
100	5	1,42	<u>0,00</u>	0,42	1,49	<u>2,89</u>	2,45
100	10	0,58	0,47	<u>0,00</u>	1,36	1,47	<u>1,93</u>
100	20	0,54	0,52	<u>0,00</u>	1,59	1,60	<u>2,10</u>
200	10	0,78	<u>0,00</u>	0,41	1,54	<u>2,30</u>	1,90
200	20	0,63	<u>0,00</u>	0,08	0,86	<u>1,47</u>	1,39
NRV		0	6	5	0	6	5

Tabela A.12 – Estatísticas complementares usando HC2 (DVC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
MP	0,77	0,26	<u>0,18</u>	1,39	1,88	<u>1,95</u>
MS	1,14	<u>0,31</u>	0,36	1,76	<u>2,55</u>	2,50
DP	0,65	<u>0,40</u>	0,53	<u>0,53</u>	1,01	0,87
CV	<u>0,57</u>	1,29	1,47	<u>0,30</u>	0,40	0,35

A.7 Resumo da análise DRP1

Na Tabela A.13, é apresentado o resumo das contagens de resultados vencedores (NRV) usando a medida DRP1 para as três metaheurística e respectivas heurísticas de iniciação.

Os resultados demonstram que a metaheurística IG_{SA}^2 é a vencedora com 48 tamanhos de instância. A metaheurística ILS_{SA-VNS} está em segundo lugar com dezoito tamanhos de instância.

Tabela A.13 – Resumo NRV usando DRP1 (DVC2)

	DRP1 (%)			Melhoria (%)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
NEH	0	11	0	0	11	0
XNEH	0	10	1	0	10	1
CHINS	0	8	3	0	8	3
XCHINS	0	4	7	0	4	7
HC1	0	9	2	0	9	2
HC2	0	6	5	0	6	5
Total	0	48	18	0	48	18

Tabela A.13B – Resumo NRV por tamanho de instância (DVC2)

	Pequena ($I_s = 20$)			Média ($I_s = 50$)			Grande ($I_s \geq 100$)		
	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
NEH	0	3	0	0	3	0	0	5	0
XNEH	0	3	0	0	3	0	0	4	1
CHINS	0	2	1	0	1	2	0	5	0
XCHINS	0	3	0	0	0	3	0	1	4
HC1	0	3	0	0	1	2	0	5	0
HC2	0	3	0	0	0	3	0	3	2
Total	0	17	1	0	8	10	0	23	7

A.8 Análise DRP independente da heurística de iniciação

As avaliações dos resultados das execuções das três metaheurísticas independentes da heurística de iniciação são feitas por meio das medidas DRP2 e DRP3.

Na Tabela A.14, são apresentados os valores de DRP2 e DRP3 para as três metaheurísticas. Os melhores valores estão grifados.

Usando DRP2 para medir as melhores execuções, a metaheurística ILS_{SA-VNS} é a vencedora com sete tamanhos de instância. Usando DRP3, a metaheurística IG_{SA}^2 é a vencedora para todos os onze tamanhos de instância. O número de resultados vencedores (NRV) é apresentado no final da tabela.

Tabela A.14 – DRP2 e DRP3 para as três metaheurísticas (DVC2)

n	m	DRP2 (%)			DRP3 (%)		
		IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}	IG_{SA}^1	IG_{SA}^2	ILS_{SA-VNS}
20	5	1,93	<u>0,00</u>	0,54	5,05	<u>0,00</u>	2,63
20	10	3,42	<u>0,00</u>	2,50	5,00	<u>0,00</u>	3,60
20	20	1,40	<u>0,00</u>	0,66	3,59	<u>0,00</u>	2,68
50	5	1,18	0,97	<u>0,00</u>	3,53	<u>0,00</u>	2,11
50	10	0,20	0,64	<u>0,00</u>	1,87	<u>0,00</u>	0,77
50	20	0,53	0,54	<u>0,00</u>	1,20	<u>0,00</u>	0,05
100	5	0,47	0,13	<u>0,00</u>	3,18	<u>0,00</u>	2,46
100	10	0,35	0,12	<u>0,00</u>	1,83	<u>0,00</u>	0,82
100	20	0,07	0,13	<u>0,00</u>	0,80	<u>0,00</u>	0,06
200	10	0,44	<u>0,00</u>	0,38	2,06	<u>0,00</u>	1,67
200	20	0,40	0,07	<u>0,00</u>	1,22	<u>0,00</u>	0,69
NRV		0	4	7	0	11	0

A.9 Análise de variância (ANOVA)

Na Tabela A.15, são apresentados os resultados da análise de variância (ANOVA) para os onze experimentos do conjunto de experimentos do PPIPD-NWFS-DVC2.

Na coluna 'Difere?', são identificados os experimentos cujos tratamentos (combinação de uma metaheurística e uma heurística de iniciação) diferem significativamente entre si. Este resultado sugere que a combinação de uma metaheurística e uma heurística de iniciação afeta significativamente os resultados em quase todos os tamanhos de instância avaliados.

Para os experimentos que diferem, os métodos *post-hoc* Tukey e Fisher (LSD) são empregados para avaliar as diferenças entre os tratamentos. Os resultados das análises realizadas por ambos os métodos são apresentados no Apêndice B.

Tabela A.15 – Resultados ANOVA (DVC2)

n	m	F0	Fcrit	F0/Fcrit	α	Valor-p	Difere?
20	5	1,516	1,767	0,86	0,05	0,113	Não
20	10	0,950	1,767	0,54	0,05	0,521	Não
20	20	4,374	1,767	2,48	0,05	4,77E-06	Sim
50	5	4,372	1,767	2,47	0,05	4,81E-06	Sim
50	10	7,030	1,767	3,98	0,05	1,28E-09	Sim
50	20	2,618	1,767	1,48	0,05	2,42E-03	Sim
100	5	5,101	1,767	2,89	0,05	4,31E-07	Sim
100	10	6,157	1,767	3,49	0,05	1,62E-08	Sim
100	20	12,631	1,767	7,15	0,05	2,33E-15	Sim
200	10	11,558	1,767	6,54	0,05	2,10E-14	Sim
200	20	7,157	1,767	4,05	0,05	9,00E-10	Sim

A.10 Variações percentuais entre as FO de DVC1 e DVC2

Na Tabela A.16, são apresentadas as variações percentuais entre os valores de função objetivo (FO) dos modelos DVC1 e DVC2 para os diferentes tamanhos de instância.

A variação percentual é calculada como $\Delta FO = 100 \times (FO_1 - FO_2) / FO_2$, onde FO_1 é o valor da FO do modelo DVC1 e FO_2 é o valor da FO do modelo DVC2.

Valores negativos de ΔFO indicam as execuções onde a abordagem 1 (DVC1) gerou melhores resultados que a abordagem 2 (DVC2).

Tabela A.16 – Variações percentuais entre as FO de DVC1 e DVC2

n	m	NEH			XNEH			CHINS			XCHINS			HC1			HC2		
		IG1	IG2	ILS	IG1	IG2	ILS	IG1	IG2	ILS	IG1	IG2	ILS	IG1	IG2	ILS	IG1	IG2	ILS
20	5	-7,0	1,9	-1,2	-5,0	1,8	-2,0	0,2	-0,1	3,4	0,7	2,2	2,5	1,0	2,4	0,7	1,3	1,1	2,4
20	10	-7,1	0,1	-4,1	-3,8	0,5	-3,0	-1,0	0,0	2,8	-0,3	-0,6	1,7	-0,5	-0,3	-2,1	-0,3	-1,1	-0,7
20	20	-5,4	-1,3	-3,5	-2,6	-1,0	-3,4	0,8	-1,1	2,3	1,5	-0,7	3,1	2,1	-0,9	0,6	0,1	-0,8	0,1
50	5	-7,3	-0,7	-5,4	-6,3	-1,5	-5,3	1,0	-0,9	1,5	1,0	-1,9	2,7	-2,5	-1,1	-1,8	-1,3	-2,2	0,7
50	10	-8,3	-4,0	-6,4	-6,0	-5,1	-5,3	-1,7	-2,8	-0,1	-0,1	-3,1	-0,1	-1,3	-4,3	-1,3	-1,0	-3,2	-1,3
50	20	-7,1	-4,6	-3,5	-4,3	-4,2	-3,3	0,1	-1,1	-0,5	-0,5	-2,0	0,3	-0,5	-5,1	-1,3	-2,2	-3,5	-1,4
100	5	-5,1	0,7	-3,8	-3,6	0,7	-3,4	1,8	-0,2	2,1	2,8	1,9	3,3	-0,7	0,6	-0,2	0,9	0,7	1,6
100	10	-7,8	-5,1	-5,6	-6,4	-7,4	-5,5	-2,7	-2,6	-2,1	-1,9	-2,0	-1,5	-1,6	-3,4	-1,3	-3,1	-3,4	-2,6
100	20	-8,2	-5,5	-6,6	-6,3	-6,3	-5,7	-1,2	-1,6	-0,7	-0,3	-1,5	-0,3	-1,7	-3,7	-1,8	-2,8	-3,3	-2,5
200	10	-7,3	-4,0	-6,6	-5,6	-3,8	-5,4	-2,4	-1,0	-1,8	-0,9	-1,6	-0,9	-2,5	-3,7	-2,6	-4,5	-4,2	-4,2
200	20	-7,8	-7,0	-7,3	-6,0	-6,4	-5,5	-3,8	-2,9	-3,3	-2,4	-2,1	-2,1	-2,0	-4,1	-1,5	-4,1	-3,9	-3,6

Legenda: IG1 = IG_{SA}^1 ; IG2 = IG_{SA}^2 ; ILS = ILS_{SA-VNS} .

Apêndice B – Resultados das Análises de Tukey e Fisher

Esta seção apresenta os resultados das análises *post-hoc* de Tukey e Fisher (LSD) para os resultados das execuções dos três métodos heurísticos (IG_{SA}^1 , IG_{SA}^2 e ILS_{SA-VNS}) combinados com as seis heurísticas de iniciação (NEH, XNEH, CHINS, XCHINS, HC1 e HC2) as quais totalizaram 153 comparações pareadas para cada tamanho de instância ($n \times m$), para cada modelo de transporte e abordagem de otimização. Estes resultados são comentados no Capítulo 6 nas conclusões das análises estatísticas.

As análises de Tukey e Fisher foram aplicadas apenas para os experimentos nos quais os tratamentos diferiram significativamente na análise de variância (ANOVA). Os resultados da ANOVA são apresentados no Capítulo 6.

O embasamento teórico de ambos os métodos é descrito em (MONTGOMERY, 2017).

Nas Tabelas de B.1 a B.16, são apresentados os resultados para o modelo DFC2. Nas Tabelas B.17 a B.40, são apresentados os resultados para o modelo DVC1 e DVC2.

Os tamanhos de instância são apresentados nos títulos das tabelas com a notação $N_j - M_i$, onde j e i representam, respectivamente, o número de ordens e de máquinas.

Os valores que demonstram diferenças significativas entre pares de tratamentos são apresentados em negrito.

Os valores de comparação entre pares de tratamentos foram calculados pelas fórmulas seguintes.

Para o teste de Tukey tem-se:

$$V_{c,l} = \frac{(\overline{FO_c} - \overline{FO_l})}{T_\alpha}$$

Para o teste de Fisher tem-se:

$$V_{c,l} = \frac{(\overline{FO_c} - \overline{FO_l})}{LSD}$$

Onde $\overline{FO_c}$ e $\overline{FO_l}$ são as médias dos valores (resultados de execução) de todas as instâncias dos tratamentos c e l comparados dois a dois; T_α e LSD são parâmetros de comparação de cada um dos testes estatísticos.

Para ambos os testes, os pares de tratamentos diferem significativamente se $|V_{c,l}| > 1$.

Um valor significativamente negativo ($V_{c,l} < -1$) estabelece que o tratamento identificado na coluna c é estatisticamente superior ao tratamento identificado na linha l . Por exemplo, na Tabela B.1, na coluna 2 e linha 7, o tratamento $IG_{SA}^2 - XNEH$ (coluna) é estatisticamente superior ao tratamento $ILS_{SA-VNS} - XNEH$ (linha), com valor $V_{2,7} = -1,43$.

Já um valor significativamente positivo ($V_{c,l} > 1$) estabelece que o tratamento identificado na coluna c é estatisticamente inferior ao tratamento identificado na linha l . Por exemplo, na Tabela B.1, na coluna 1 e linha 1, o tratamento $IG_{SA}^1 - NEH$ (coluna) é estatisticamente inferior ao tratamento $IG_{SA}^2 - XNEH$ (linha) com valor $V_{1,1} = 1,81$.

Tabela B.1 – Resultado teste Tukey instâncias N50-M5 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	IG1- XNEH	IG1- HC1	ILS- NEH	ILS- CHINS	ILS- XCHINS
IG2-XNEH	1,81																
IG1-CHINS	1,67	-0,14															
IG1-XCHINS	1,92	0,11	0,25														
IG2-HC1	1,77	-0,04	0,10	-0,15													
IG1-HC2	1,78	-0,03	0,11	-0,14	0,01												
IG2-NEH	1,75	-0,05	0,09	-0,16	-0,01	-0,02											
ILS-XNEH	0,38	-1,43	-1,29	-1,54	-1,39	-1,40	-1,38										
IG2-CHINS	1,73	-0,08	0,06	-0,19	-0,04	-0,05	-0,02	1,35									
IG2-XCHINS	1,75	-0,06	0,08	-0,17	-0,02	-0,03	-0,01	1,37	0,02								
ILS-HC1	2,16	0,35	0,50	0,25	0,40	0,38	0,41	1,79	0,43	0,42							
IG2-HC2	1,71	-0,10	0,05	-0,21	-0,06	-0,07	-0,04	1,33	-0,02	-0,03	-0,45						
IG1-XNEH	0,12	-1,69	-1,55	-1,80	-1,65	-1,66	-1,64	-0,26	-1,61	-1,63	-2,04	-1,59					
IG1-HC1	1,87	0,06	0,20	-0,05	0,10	0,09	0,11	1,49	0,14	0,12	-0,29	0,16	1,75				
ILS-NEH	0,87	-0,94	-0,80	-1,05	-0,90	-0,91	-0,89	0,49	-0,86	-0,88	-1,30	-0,84	0,75	-1,00			
ILS-CHINS	2,17	0,36	0,50	0,25	0,40	0,39	0,41	1,79	0,44	0,42	0,00	0,46	2,05	0,30	1,30		
ILS-XCHINS	2,09	0,28	0,42	0,17	0,32	0,31	0,33	1,71	0,36	0,34	-0,07	0,38	1,97	0,22	1,22	-0,08	
ILS-HC2	2,31	0,50	0,64	0,39	0,54	0,53	0,55	1,93	0,58	0,56	0,14	0,60	2,19	0,44	1,44	0,14	0,22

Tabela B.2 – Resultado teste Fisher instâncias N50-M5 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	IG1- XNEH	IG1- HC1	ILS- NEH	ILS- CHINS	ILS- XCHINS
IG2-XNEH	2,54																
IG1-CHINS	2,34	-0,20															
IG1-XCHINS	2,70	0,15	0,35														
IG2-HC1	2,48	-0,06	0,14	-0,21													
IG1-HC2	2,50	-0,04	0,16	-0,19	0,02												
IG2-NEH	2,47	-0,08	0,13	-0,23	-0,02	-0,03											
ILS-XNEH	0,53	-2,01	-1,81	-2,16	-1,95	-1,97	-1,94										
IG2-CHINS	2,43	-0,11	0,09	-0,26	-0,05	-0,07	-0,03	1,90									
IG2-XCHINS	2,45	-0,09	0,11	-0,24	-0,03	-0,05	-0,01	1,92	0,02								
ILS-HC1	3,04	0,50	0,70	0,35	0,56	0,54	0,57	2,51	0,61	0,59							
IG2-HC2	2,41	-0,14	0,06	-0,29	-0,08	-0,09	-0,06	1,88	-0,03	-0,05	-0,63						
IG1-XNEH	0,17	-2,38	-2,17	-2,53	-2,32	-2,33	-2,30	-0,36	-2,26	-2,29	-2,87	-2,24					
IG1-HC1	2,63	0,08	0,29	-0,07	0,14	0,13	0,16	2,10	0,20	0,17	-0,41	0,22	2,46				
ILS-NEH	1,22	-1,32	-1,12	-1,48	-1,27	-1,28	-1,25	0,69	-1,21	-1,24	-1,82	-1,19	1,05	-1,41			
ILS-CHINS	3,05	0,51	0,71	0,35	0,56	0,55	0,58	2,52	0,62	0,59	0,01	0,64	2,88	0,42	1,83		
ILS-XCHINS	2,94	0,39	0,60	0,24	0,45	0,44	0,47	2,41	0,51	0,48	-0,10	0,53	2,77	0,31	1,72	-0,11	
ILS-HC2	3,24	0,70	0,90	0,55	0,76	0,74	0,78	2,71	0,81	0,79	0,20	0,84	3,08	0,62	2,03	0,20	0,31

Tabela B.3 – Resultado teste Tukey instâncias N50-M10 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	IG1- XNEH	IG1- HC1	ILS- NEH	ILS- CHINS	ILS- XCHINS
IG2-XNEH	2,03																
IG1-CHINS	2,09	0,06															
IG1-XCHINS	2,16	0,13	0,07														
IG2-HC1	1,99	-0,04	-0,10	-0,17													
IG1-HC2	2,07	0,04	-0,02	-0,08	0,08												
IG2-NEH	2,04	0,01	-0,05	-0,12	0,05	-0,03											
ILS-XNEH	0,13	-1,90	-1,96	-2,03	-1,86	-1,94	-1,91										
IG2-CHINS	2,03	0,00	-0,06	-0,12	0,04	-0,04	-0,01	1,90									
IG2-XCHINS	2,05	0,02	-0,04	-0,11	0,06	-0,03	0,01	1,92	0,01								
ILS-HC1	2,14	0,11	0,05	-0,02	0,15	0,06	0,10	2,01	0,10	0,09							
IG2-HC2	2,00	-0,03	-0,09	-0,15	0,01	-0,07	-0,03	1,88	-0,03	-0,04	-0,13						
IG1-XNEH	0,10	-1,93	-1,99	-2,06	-1,89	-1,97	-1,94	-0,03	-1,93	-1,95	-2,04	-1,90					
IG1-HC1	2,16	0,13	0,07	0,00	0,17	0,08	0,12	2,03	0,13	0,11	0,02	0,15	2,06				
ILS-NEH	0,00	-2,03	-2,09	-2,15	-1,99	-2,07	-2,04	-0,13	-2,03	-2,04	-2,13	-2,00	-0,10	-2,15			
ILS-CHINS	2,05	0,02	-0,04	-0,11	0,06	-0,03	0,01	1,92	0,02	0,00	-0,09	0,04	1,95	-0,11	2,05		
ILS-XCHINS	2,16	0,13	0,06	0,00	0,17	0,08	0,12	2,03	0,12	0,11	0,02	0,15	2,06	0,00	2,15	0,11	
ILS-HC2	2,06	0,03	-0,03	-0,10	0,07	-0,02	0,02	1,93	0,03	0,01	-0,08	0,05	1,96	-0,10	2,06	0,01	-0,10

Tabela B.4 – Resultado teste Fisher instâncias N50-M10 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	IG1- XNEH	IG1- HC1	ILS- NEH	ILS- CHINS	ILS- XCHINS
IG2-XNEH	2,85																
IG1-CHINS	2,94	0,09															
IG1-XCHINS	3,03	0,18	0,09														
IG2-HC1	2,80	-0,06	-0,14	-0,23													
IG1-HC2	2,91	0,06	-0,02	-0,12	0,12												
IG2-NEH	2,87	0,01	-0,07	-0,16	0,07	-0,05											
ILS-XNEH	0,18	-2,67	-2,76	-2,85	-2,62	-2,73	-2,69										
IG2-CHINS	2,86	0,00	-0,08	-0,17	0,06	-0,06	-0,01	2,68									
IG2-XCHINS	2,88	0,02	-0,06	-0,15	0,08	-0,04	0,01	2,70	0,02								
ILS-HC1	3,00	0,15	0,06	-0,03	0,20	0,09	0,14	2,82	0,15	0,13							
IG2-HC2	2,82	-0,04	-0,12	-0,21	0,02	-0,10	-0,05	2,64	-0,04	-0,06	-0,18						
IG1-XNEH	0,14	-2,71	-2,80	-2,89	-2,66	-2,77	-2,72	-0,04	-2,72	-2,74	-2,86	-2,68					
IG1-HC1	3,03	0,18	0,09	0,00	0,24	0,12	0,17	2,85	0,18	0,16	0,03	0,22	2,89				
ILS-NEH	0,00	-2,85	-2,93	-3,03	-2,79	-2,91	-2,86	-0,18	-2,85	-2,87	-3,00	-2,81	-0,14	-3,03			
ILS-CHINS	2,88	0,03	-0,06	-0,15	0,08	-0,04	0,01	2,70	0,02	0,00	-0,12	0,06	2,74	-0,15	2,87		
ILS-XCHINS	3,03	0,18	0,09	0,00	0,23	0,12	0,16	2,85	0,17	0,15	0,03	0,21	2,89	0,00	3,03	0,15	
ILS-HC2	2,89	0,04	-0,05	-0,14	0,10	-0,02	0,03	2,71	0,04	0,02	-0,11	0,08	2,75	-0,14	2,89	0,01	-0,14

Tabela B.5 – Resultado teste Tukey instâncias N50-M20 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	IG1- XNEH	IG1- HC1	ILS- NEH	ILS- CHINS	ILS- XCHINS
IG2-XNEH	1,25																
IG1-CHINS	1,24	-0,01															
IG1-XCHINS	1,26	0,01	0,02														
IG2-HC1	1,20	-0,05	-0,04	-0,06													
IG1-HC2	1,34	0,09	0,11	0,09	0,15												
IG2-NEH	1,26	0,01	0,02	0,00	0,06	-0,08											
ILS-XNEH	-0,04	-1,29	-1,28	-1,29	-1,24	-1,38	-1,30										
IG2-CHINS	1,27	0,02	0,03	0,01	0,07	-0,08	0,01	1,31									
IG2-XCHINS	1,32	0,07	0,08	0,06	0,12	-0,03	0,06	1,36	0,05								
ILS-HC1	1,31	0,06	0,07	0,05	0,11	-0,03	0,05	1,35	0,04	-0,01							
IG2-HC2	1,15	-0,10	-0,09	-0,11	-0,05	-0,20	-0,11	1,19	-0,12	-0,17	-0,16						
IG1-XNEH	0,10	-1,15	-1,14	-1,16	-1,10	-1,24	-1,16	0,14	-1,17	-1,22	-1,21	-1,05					
IG1-HC1	1,31	0,07	0,08	0,06	0,12	-0,03	0,06	1,35	0,05	0,00	0,01	0,17	1,22				
ILS-NEH	-0,04	-1,29	-1,28	-1,30	-1,24	-1,38	-1,30	0,00	-1,31	-1,36	-1,35	-1,19	-0,14	-1,36			
ILS-CHINS	1,16	-0,09	-0,08	-0,10	-0,04	-0,19	-0,10	1,19	-0,11	-0,16	-0,15	0,01	1,06	-0,16	1,20		
ILS-XCHINS	1,21	-0,04	-0,03	-0,05	0,01	-0,14	-0,05	1,24	-0,06	-0,11	-0,10	0,06	1,11	-0,11	1,25	0,05	
ILS-HC2	1,19	-0,06	-0,05	-0,06	-0,01	-0,15	-0,07	1,23	-0,08	-0,13	-0,12	0,04	1,09	-0,12	1,23	0,04	-0,01

Tabela B.6 – Resultado teste Fisher instâncias N50-M20 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	IG1- XNEH	IG1- HC1	ILS- NEH	ILS- CHINS	ILS- XCHINS
IG2-XNEH	1,76																
IG1-CHINS	1,74	-0,02															
IG1-XCHINS	1,76	0,01	0,02														
IG2-HC1	1,68	-0,07	-0,06	-0,08													
IG1-HC2	1,89	0,13	0,15	0,12	0,21												
IG2-NEH	1,77	0,01	0,03	0,00	0,09	-0,12											
ILS-XNEH	-0,05	-1,81	-1,80	-1,82	-1,74	-1,94	-1,82										
IG2-CHINS	1,78	0,02	0,04	0,02	0,10	-0,11	0,01	1,84									
IG2-XCHINS	1,85	0,10	0,11	0,09	0,17	-0,04	0,08	1,91	0,07								
ILS-HC1	1,84	0,08	0,10	0,08	0,16	-0,05	0,07	1,89	0,06	-0,01							
IG2-HC2	1,61	-0,14	-0,13	-0,15	-0,07	-0,28	-0,16	1,67	-0,17	-0,24	-0,23						
IG1-XNEH	0,14	-1,62	-1,60	-1,63	-1,54	-1,75	-1,63	0,19	-1,64	-1,71	-1,70	-1,47					
IG1-HC1	1,85	0,09	0,11	0,08	0,17	-0,04	0,08	1,90	0,07	-0,01	0,01	0,24	1,71				
ILS-NEH	-0,06	-1,81	-1,80	-1,82	-1,74	-1,95	-1,83	0,00	-1,84	-1,91	-1,90	-1,67	-0,20	-1,91			
ILS-CHINS	1,62	-0,13	-0,12	-0,14	-0,06	-0,26	-0,15	1,68	-0,16	-0,23	-0,22	0,01	1,48	-0,22	1,68		
ILS-XCHINS	1,69	-0,06	-0,05	-0,07	0,01	-0,19	-0,07	1,75	-0,09	-0,16	-0,15	0,08	1,56	-0,15	1,75	0,07	
ILS-HC2	1,67	-0,08	-0,07	-0,09	-0,01	-0,21	-0,10	1,73	-0,11	-0,18	-0,17	0,06	1,53	-0,17	1,73	0,05	-0,02

Tabela B.7 – Resultado teste Tukey instâncias N100-M5 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	ILS- NEH	ILS- CHINS	ILS- XCHINS	ILS- HC2	IG1- XNEH
IG2-XNEH	2,94																
IG1-CHINS	2,80	-0,14															
IG1-XCHINS	3,11	0,17	0,31														
IG2-HC1	3,28	0,34	0,48	0,17													
IG1-HC2	3,25	0,32	0,46	0,15	-0,03												
IG2-NEH	2,83	-0,10	0,04	-0,27	-0,44	-0,42											
ILS-XNEH	0,73	-2,21	-2,07	-2,38	-2,55	-2,53	-2,11										
IG2-CHINS	3,13	0,19	0,33	0,02	-0,15	-0,12	0,29	2,40									
IG2-XCHINS	3,23	0,30	0,44	0,13	-0,04	-0,02	0,40	2,51	0,10								
ILS-HC1	3,32	0,39	0,53	0,22	0,05	0,07	0,49	2,60	0,19	0,09							
IG2-HC2	3,41	0,48	0,62	0,31	0,13	0,16	0,58	2,69	0,28	0,18	0,09						
ILS-NEH	0,68	-2,25	-2,11	-2,42	-2,59	-2,57	-2,15	-0,04	-2,45	-2,55	-2,64	-2,73					
ILS-CHINS	3,18	0,24	0,38	0,07	-0,10	-0,07	0,34	2,45	0,05	-0,05	-0,14	-0,23	2,50				
ILS-XCHINS	3,20	0,26	0,40	0,09	-0,08	-0,05	0,36	2,47	0,07	-0,03	-0,12	-0,21	2,52	0,02			
ILS-HC2	3,44	0,50	0,64	0,33	0,16	0,19	0,60	2,71	0,31	0,21	0,12	0,03	2,76	0,26	0,24		
IG1-XNEH	0,41	-2,52	-2,38	-2,69	-2,86	-2,84	-2,42	-0,31	-2,71	-2,82	-2,91	-3,00	-0,27	-2,77	-2,79	-3,02	
IG1-HC1	3,08	0,14	0,28	-0,03	-0,20	-0,17	0,24	2,35	-0,05	-0,15	-0,24	-0,33	2,40	-0,10	-0,12	-0,36	2,67

Tabela B.8 – Resultado teste Fisher instâncias N100-M5 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	IG2- NEH	IG2- CHINS	IG2- XCHINS	IG2- HC2	IG2- XNEH
IG2-XNEH	4,13																
IG1-CHINS	3,93	-0,20															
IG1-XCHINS	4,37	0,24	0,44														
IG2-HC1	4,61	0,48	0,68	0,24													
IG1-HC2	4,57	0,44	0,64	0,21	-0,04												
IG2-NEH	3,99	-0,14	0,06	-0,38	-0,62	-0,59											
IG2-XNEH	1,02	-3,11	-2,91	-3,35	-3,59	-3,55	-2,97										
IG2-CHINS	4,40	0,27	0,47	0,03	-0,21	-0,17	0,41	3,38									
IG2-XCHINS	4,54	0,42	0,61	0,18	-0,06	-0,03	0,56	3,52	0,15								
IG2-HC1	4,67	0,54	0,74	0,30	0,06	0,10	0,68	3,65	0,27	0,13							
IG2-HC2	4,80	0,67	0,87	0,43	0,19	0,22	0,81	3,78	0,40	0,25	0,13						
IG2-NEH	0,96	-3,17	-2,97	-3,41	-3,65	-3,61	-3,03	-0,06	-3,44	-3,58	-3,71	-3,84					
IG2-CHINS	4,47	0,34	0,54	0,10	-0,14	-0,10	0,48	3,45	0,07	-0,08	-0,20	-0,33	3,51				
IG2-XCHINS	4,50	0,37	0,57	0,13	-0,11	-0,07	0,51	3,48	0,10	-0,05	-0,17	-0,30	3,54	0,03			
IG2-HC2	4,83	0,71	0,90	0,47	0,23	0,26	0,85	3,81	0,44	0,29	0,16	0,04	3,87	0,36	0,33		
IG1-XNEH	0,58	-3,55	-3,35	-3,79	-4,03	-3,99	-3,41	-0,44	-3,82	-3,96	-4,09	-4,22	-0,38	-3,89	-3,92	-4,25	
IG1-HC1	4,33	0,20	0,40	-0,04	-0,28	-0,24	0,34	3,31	-0,07	-0,21	-0,34	-0,47	3,37	-0,14	-0,17	-0,50	3,75

Tabela B.9 – Resultado teste Tukey instâncias N100-M10 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	ILS- NEH	ILS- CHINS	ILS- XCHINS	ILS- HC2	IG1- XNEH
IG2-XNEH	2,23																
IG1-CHINS	2,23	0,00															
IG1-XCHINS	2,29	0,06	0,06														
IG2-HC1	2,13	-0,10	-0,10	-0,16													
IG1-HC2	2,24	0,00	0,00	-0,06	0,10												
IG2-NEH	2,15	-0,09	-0,09	-0,14	0,01	-0,09											
ILS-XNEH	-0,12	-2,36	-2,36	-2,41	-2,26	-2,36	-2,27										
IG2-CHINS	2,26	0,03	0,03	-0,03	0,13	0,03	0,12	2,39									
IG2-XCHINS	2,28	0,04	0,04	-0,01	0,14	0,04	0,13	2,40	0,01								
ILS-HC1	2,26	0,02	0,02	-0,04	0,12	0,02	0,11	2,38	-0,01	-0,02							
IG2-HC2	2,24	0,01	0,01	-0,05	0,11	0,01	0,10	2,36	-0,02	-0,03	-0,01						
ILS-NEH	-0,04	-2,28	-2,28	-2,34	-2,18	-2,28	-2,19	0,08	-2,31	-2,32	-2,30	-2,29					
ILS-CHINS	2,24	0,01	0,01	-0,05	0,11	0,01	0,10	2,36	-0,02	-0,03	-0,01	0,00	2,29				
ILS-XCHINS	2,34	0,10	0,10	0,05	0,20	0,10	0,19	2,46	0,08	0,06	0,08	0,10	2,38	0,10			
ILS-HC2	2,26	0,02	0,02	-0,03	0,12	0,02	0,11	2,38	0,00	-0,02	0,00	0,02	2,30	0,02	-0,08		
IG1-XNEH	-0,13	-2,36	-2,36	-2,42	-2,26	-2,36	-2,28	-0,01	-2,39	-2,41	-2,38	-2,37	-0,08	-2,37	-2,47	-2,39	
IG1-HC1	2,20	-0,03	-0,03	-0,09	0,07	-0,03	0,06	2,33	-0,06	-0,07	-0,05	-0,04	2,25	-0,04	-0,13	-0,05	2,33

Tabela B.10 – Resultado teste Fisher instâncias N100-M10 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	ILS- NEH	ILS- CHINS	ILS- XCHINS	ILS- HC2	IG1- XNEH
IG2-XNEH	3,14																
IG1-CHINS	3,14	0,00															
IG1-XCHINS	3,22	0,08	0,08														
IG2-HC1	3,00	-0,14	-0,14	-0,22													
IG1-HC2	3,14	0,00	0,00	-0,08	0,14												
IG2-NEH	3,02	-0,12	-0,12	-0,20	0,02	-0,12											
ILS-XNEH	-0,17	-3,31	-3,31	-3,39	-3,17	-3,31	-3,19										
IG2-CHINS	3,18	0,04	0,04	-0,04	0,18	0,04	0,16	3,35									
IG2-XCHINS	3,20	0,06	0,06	-0,02	0,20	0,06	0,18	3,37	0,02								
ILS-HC1	3,17	0,03	0,03	-0,05	0,17	0,03	0,15	3,34	-0,01	-0,03							
IG2-HC2	3,15	0,01	0,01	-0,07	0,15	0,01	0,13	3,32	-0,03	-0,05	-0,02						
ILS-NEH	-0,06	-3,20	-3,20	-3,28	-3,06	-3,20	-3,08	0,11	-3,24	-3,26	-3,23	-3,21					
ILS-CHINS	3,15	0,01	0,01	-0,07	0,15	0,01	0,13	3,32	-0,03	-0,05	-0,02	0,00	3,21				
ILS-XCHINS	3,29	0,15	0,15	0,07	0,29	0,14	0,27	3,46	0,11	0,09	0,12	0,13	3,35	0,14			
ILS-HC2	3,18	0,03	0,04	-0,05	0,17	0,03	0,16	3,35	-0,01	-0,03	0,00	0,02	3,24	0,02	-0,11		
IG1-XNEH	-0,18	-3,32	-3,32	-3,40	-3,18	-3,32	-3,20	-0,01	-3,36	-3,38	-3,35	-3,33	-0,12	-3,33	-3,47	-3,36	
IG1-HC1	3,10	-0,04	-0,04	-0,12	0,10	-0,04	0,08	3,27	-0,08	-0,10	-0,07	-0,06	3,16	-0,05	-0,19	-0,08	3,28

Tabela B.11 – Resultado teste Tukey instâncias N100-M20 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	ILS- NEH	ILS- CHINS	ILS- XCHINS	ILS- HC2	IG1- XNEH
IG2-XNEH	3,30																
IG1-CHINS	3,38	0,09															
IG1-XCHINS	3,47	0,17	0,08														
IG2-HC1	3,47	0,17	0,08	0,00													
IG1-HC2	3,52	0,23	0,14	0,06	0,06												
IG2-NEH	3,04	-0,26	-0,35	-0,43	-0,43	-0,49											
ILS-XNEH	0,04	-3,25	-3,34	-3,43	-3,42	-3,48	-2,99										
IG2-CHINS	3,48	0,19	0,10	0,02	0,02	-0,04	0,45	3,44									
IG2-XCHINS	3,43	0,13	0,05	-0,04	-0,04	-0,09	0,39	3,39	-0,05								
ILS-HC1	3,41	0,11	0,03	-0,06	-0,06	-0,11	0,37	3,37	-0,07	-0,02							
IG2-HC2	3,42	0,12	0,04	-0,05	-0,05	-0,10	0,38	3,38	-0,06	-0,01	0,01						
ILS-NEH	-0,01	-3,30	-3,39	-3,48	-3,48	-3,53	-3,04	-0,05	-3,49	-3,44	-3,42	-3,43					
ILS-CHINS	3,36	0,06	-0,03	-0,11	-0,11	-0,17	0,32	3,32	-0,13	-0,07	-0,05	-0,06	3,37				
ILS-XCHINS	3,38	0,09	0,00	-0,08	-0,08	-0,14	0,35	3,34	-0,10	-0,05	-0,03	-0,04	3,39	0,03			
ILS-HC2	3,56	0,26	0,17	0,09	0,09	0,03	0,52	3,51	0,07	0,13	0,15	0,13	3,56	0,20	0,17		
IG1-XNEH	-0,02	-3,32	-3,41	-3,49	-3,49	-3,55	-3,06	-0,07	-3,51	-3,45	-3,44	-3,45	-0,02	-3,38	-3,41	-3,58	
IG1-HC1	3,44	0,14	0,06	-0,03	-0,03	-0,09	0,40	3,40	-0,05	0,01	0,03	0,02	3,45	0,08	0,06	-0,12	3,46

Tabela B.12 – Resultado teste Fisher instâncias N100-M20 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	ILS- NEH	ILS- CHINS	ILS- XCHINS	ILS- HC2	IG1- XNEH
IG2-XNEH	4,63																
IG1-CHINS	4,76	0,12															
IG1-XCHINS	4,88	0,24	0,12														
IG2-HC1	4,87	0,24	0,12	0,00													
IG1-HC2	4,95	0,32	0,20	0,08	0,08												
IG2-NEH	4,27	-0,37	-0,49	-0,61	-0,61	-0,69											
ILS-XNEH	0,06	-4,57	-4,70	-4,81	-4,81	-4,89	-4,21										
IG2-CHINS	4,90	0,26	0,14	0,02	0,02	-0,06	0,63	4,84									
IG2-XCHINS	4,82	0,19	0,06	-0,05	-0,05	-0,13	0,55	4,76	-0,08								
ILS-HC1	4,79	0,16	0,04	-0,08	-0,08	-0,16	0,52	4,73	-0,10	-0,03							
IG2-HC2	4,81	0,17	0,05	-0,07	-0,06	-0,15	0,54	4,75	-0,09	-0,01	0,02						
ILS-NEH	-0,01	-4,65	-4,77	-4,89	-4,89	-4,97	-4,28	-0,07	-4,91	-4,83	-4,80	-4,82					
ILS-CHINS	4,72	0,09	-0,04	-0,15	-0,15	-0,23	0,45	4,66	-0,18	-0,10	-0,07	-0,09	4,73				
ILS-XCHINS	4,76	0,12	0,00	-0,12	-0,12	-0,20	0,49	4,70	-0,14	-0,06	-0,04	-0,05	4,77	0,04			
ILS-HC2	5,00	0,36	0,24	0,12	0,12	0,04	0,73	4,94	0,10	0,18	0,20	0,19	5,01	0,28	0,24		
IG1-XNEH	-0,03	-4,67	-4,79	-4,91	-4,91	-4,99	-4,30	-0,10	-4,93	-4,86	-4,83	-4,84	-0,02	-4,76	-4,79	-5,03	
IG1-HC1	4,83	0,20	0,08	-0,04	-0,04	-0,12	0,57	4,77	-0,06	0,01	0,04	0,03	4,85	0,11	0,08	-0,16	4,87

Tabela B.13 – Resultado teste Tukey instâncias N200-M10 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	ILS- NEH	ILS- CHINS	ILS- XCHINS	ILS- HC2	IG1- XNEH
IG2-XNEH	7,93																
IG1-CHINS	8,60	0,67															
IG1-XCHINS	8,81	0,89	0,22														
IG2-HC1	8,30	0,37	-0,30	-0,51													
IG1-HC2	8,58	0,65	-0,02	-0,24	0,28												
IG2-NEH	8,08	0,15	-0,52	-0,73	-0,22	-0,50											
ILS-XNEH	-0,42	-8,35	-9,02	-9,23	-8,72	-9,00	-8,50										
IG2-CHINS	8,72	0,79	0,12	-0,09	0,42	0,14	0,64	9,14									
IG2-XCHINS	8,63	0,70	0,03	-0,18	0,33	0,05	0,55	9,05	-0,09								
ILS-HC1	8,37	0,44	-0,22	-0,44	0,07	-0,20	0,29	8,79	-0,35	-0,26							
IG2-HC2	8,57	0,64	-0,03	-0,24	0,27	-0,01	0,49	8,99	-0,15	-0,06	0,20						
ILS-NEH	-0,52	-8,45	-9,12	-9,33	-8,82	-9,10	-8,60	-0,10	-9,24	-9,15	-8,89	-9,09					
ILS-CHINS	8,73	0,80	0,13	-0,08	0,43	0,15	0,65	9,15	0,01	0,10	0,36	0,16	9,25				
ILS-XCHINS	8,84	0,91	0,24	0,03	0,54	0,27	0,76	9,26	0,12	0,21	0,47	0,27	9,36	0,11			
ILS-HC2	8,78	0,86	0,19	-0,03	0,49	0,21	0,70	9,20	0,07	0,16	0,41	0,22	9,30	0,05	-0,06		
IG1-XNEH	0,25	-7,68	-8,35	-8,57	-8,05	-8,33	-7,83	0,67	-8,47	-8,38	-8,13	-8,32	0,77	-8,48	-8,60	-8,54	
IG1-HC1	8,39	0,46	-0,21	-0,42	0,09	-0,19	0,31	8,81	-0,33	-0,24	0,02	-0,18	8,91	-0,34	-0,45	-0,39	8,14

Tabela B.14 – Resultado teste Fisher instâncias N200-M10 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	ILS- NEH	ILS- CHINS	ILS- XCHINS	ILS- HC2	IG1- XNEH
IG2-XNEH	11,14																
IG1-CHINS	12,08	0,94															
IG1-XCHINS	12,39	1,24	0,30														
IG2-HC1	11,67	0,52	-0,42	-0,72													
IG1-HC2	12,06	0,91	-0,03	-0,33	0,39												
IG2-NEH	11,36	0,22	-0,73	-1,03	-0,31	-0,70											
ILS-XNEH	-0,59	-11,73	-12,68	-12,98	-12,26	-12,65	-11,95										
IG2-CHINS	12,26	1,11	0,17	-0,13	0,59	0,20	0,90	12,85									
IG2-XCHINS	12,13	0,99	0,05	-0,26	0,47	0,07	0,77	12,72	-0,12								
ILS-HC1	11,77	0,63	-0,32	-0,62	0,10	-0,29	0,41	12,36	-0,49	-0,36							
IG2-HC2	12,04	0,90	-0,04	-0,34	0,38	-0,01	0,69	12,63	-0,21	-0,09	0,28						
ILS-NEH	-0,73	-11,87	-12,81	-13,12	-12,40	-12,79	-12,09	-0,14	-12,99	-12,86	-12,50	-12,77					
ILS-CHINS	12,27	1,13	0,19	-0,12	0,61	0,22	0,91	12,86	0,02	0,14	0,50	0,23	13,00				
ILS-XCHINS	12,43	1,29	0,34	0,04	0,76	0,37	1,07	13,02	0,17	0,30	0,66	0,38	13,16	0,16			
ILS-HC2	12,35	1,21	0,26	-0,04	0,68	0,29	0,99	12,94	0,09	0,22	0,58	0,30	13,08	0,08	-0,08		
IG1-XNEH	0,35	-10,80	-11,74	-12,04	-11,32	-11,71	-11,01	0,94	-11,91	-11,78	-11,42	-11,70	1,08	-11,93	-12,08	-12,00	
IG1-HC1	11,80	0,65	-0,29	-0,59	0,13	-0,26	0,44	12,39	-0,46	-0,34	0,03	-0,25	12,53	-0,48	-0,63	-0,55	11,45

Tabela B.15 – Resultado teste Tukey instâncias N200-M20 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	ILS- NEH	ILS- CHINS	ILS- XCHINS	ILS- HC2	IG1- XNEH
IG2-XNEH	3,60																
IG1-CHINS	3,63	0,03															
IG1-XCHINS	3,84	0,24	0,21														
IG2-HC1	3,66	0,06	0,03	-0,18													
IG1-HC2	3,78	0,18	0,15	-0,06	0,12												
IG2-NEH	3,55	-0,05	-0,08	-0,29	-0,11	-0,23											
ILS-XNEH	-0,23	-3,83	-3,86	-4,06	-3,88	-4,01	-3,78										
IG2-CHINS	3,68	0,08	0,06	-0,15	0,03	-0,10	0,13	3,91									
IG2-XCHINS	3,68	0,08	0,06	-0,15	0,03	-0,10	0,13	3,91	0,00								
ILS-HC1	3,91	0,31	0,28	0,07	0,25	0,13	0,36	4,13	0,22	0,22							
IG2-HC2	3,68	0,08	0,05	-0,15	0,02	-0,10	0,13	3,91	0,00	0,00	-0,23						
ILS-NEH	-0,22	-3,82	-3,85	-4,06	-3,88	-4,00	-3,77	0,00	-3,91	-3,91	-4,13	-3,90					
ILS-CHINS	3,72	0,12	0,10	-0,11	0,07	-0,06	0,17	3,95	0,04	0,04	-0,18	0,04	3,95				
ILS-XCHINS	3,76	0,16	0,14	-0,07	0,11	-0,02	0,21	3,99	0,08	0,08	-0,14	0,08	3,99	0,04			
ILS-HC2	3,88	0,28	0,26	0,05	0,23	0,10	0,33	4,11	0,20	0,20	-0,02	0,20	4,11	0,16	0,12		
IG1-XNEH	0,07	-3,53	-3,56	-3,77	-3,59	-3,71	-3,48	0,29	-3,62	-3,62	-3,84	-3,61	0,29	-3,66	-3,70	-3,82	
IG1-HC1	3,76	0,16	0,13	-0,08	0,10	-0,02	0,21	3,99	0,08	0,08	-0,15	0,08	3,98	0,04	0,00	-0,12	3,69

Tabela B.16 – Resultado teste Fisher instâncias N200-M20 (DFC2)

	IG1- NEH	IG2- XNEH	IG1- CHINS	IG1- XCHINS	IG2- HC1	IG1- HC2	IG2- NEH	ILS- XNEH	IG2- CHINS	IG2- XCHINS	ILS- HC1	IG2- HC2	ILS- NEH	ILS- CHINS	ILS- XCHINS	ILS- HC2	IG1- XNEH
IG2-XNEH	5,06																
IG1-CHINS	5,10	0,04															
IG1-XCHINS	5,39	0,33	0,29														
IG2-HC1	5,14	0,08	0,04	-0,25													
IG1-HC2	5,31	0,25	0,21	-0,08	0,17												
IG2-NEH	4,99	-0,07	-0,11	-0,40	-0,15	-0,32											
ILS-XNEH	-0,32	-5,38	-5,42	-5,71	-5,46	-5,63	-5,31										
IG2-CHINS	5,18	0,12	0,08	-0,21	0,04	-0,14	0,19	5,50									
IG2-XCHINS	5,18	0,12	0,08	-0,21	0,04	-0,14	0,19	5,50	0,00								
ILS-HC1	5,49	0,43	0,39	0,10	0,35	0,18	0,50	5,81	0,31	0,31							
IG2-HC2	5,18	0,11	0,08	-0,22	0,03	-0,14	0,18	5,50	0,00	0,00	-0,32						
ILS-NEH	-0,31	-5,37	-5,41	-5,71	-5,45	-5,63	-5,30	0,01	-5,49	-5,49	-5,81	-5,49					
ILS-CHINS	5,24	0,17	0,14	-0,16	0,10	-0,08	0,24	5,56	0,06	0,06	-0,26	0,06	5,55				
ILS-XCHINS	5,29	0,23	0,19	-0,10	0,15	-0,02	0,30	5,61	0,11	0,11	-0,20	0,12	5,60	0,06			
ILS-HC2	5,46	0,40	0,36	0,07	0,32	0,14	0,47	5,78	0,28	0,28	-0,03	0,28	5,77	0,22	0,17		
IG1-XNEH	0,09	-4,97	-5,01	-5,30	-5,05	-5,22	-4,90	0,41	-5,08	-5,08	-5,40	-5,08	0,41	-5,14	-5,20	-5,36	
IG1-HC1	5,29	0,23	0,19	-0,11	0,15	-0,03	0,30	5,61	0,11	0,11	-0,20	0,11	5,60	0,05	0,00	-0,17	5,19

Tabela B.17 – Resultado teste Tukey instâncias N100-M20 (DVC1)

	IG1- CHINS	IG1- HC1	IG1- HC2	IG1- NEH	IG1- XCHINS	IG1- XNEH	IG2- CHINS	IG2- HC1	IG2- HC2	IG2- NEH	IG2- XCHINS	IG2- XNEH	ILS- CHINS	ILS- HC1	ILS- HC2	ILS- NEH	ILS- XCHINS
IG1-HC1	-0,48																
IG1-HC2	0,56	1,04															
IG1-NEH	0,45	0,94	-0,11														
IG1-XCHINS	0,10	0,58	-0,46	-0,35													
IG1-XNEH	0,71	1,19	0,14	0,25	0,61												
IG2-CHINS	0,62	1,10	0,06	0,16	0,52	-0,09											
IG2-HC1	0,70	1,19	0,14	0,25	0,60	0,00	0,09										
IG2-HC2	0,81	1,29	0,25	0,35	0,71	0,10	0,19	0,10									
IG2-NEH	0,63	1,11	0,07	0,18	0,53	-0,08	0,01	-0,07	-0,18								
IG2-XCHINS	0,64	1,13	0,08	0,19	0,54	-0,06	0,03	-0,06	-0,17	0,01							
IG2-XNEH	0,77	1,25	0,21	0,32	0,67	0,07	0,15	0,07	-0,04	0,14	0,13						
ILS-CHINS	0,10	0,58	-0,46	-0,35	0,00	-0,61	-0,52	-0,60	-0,71	-0,53	-0,54	-0,67					
ILS-HC1	-0,23	0,25	-0,79	-0,68	-0,33	-0,93	-0,84	-0,93	-1,04	-0,86	-0,87	-1,00	-0,33				
ILS-HC2	0,66	1,14	0,10	0,20	0,56	-0,05	0,04	-0,05	-0,15	0,03	0,02	-0,11	0,56	0,89			
ILS-NEH	0,63	1,12	0,07	0,18	0,53	-0,07	0,02	-0,07	-0,18	0,00	-0,01	-0,14	0,53	0,86	-0,03		
ILS-XCHINS	0,11	0,59	-0,45	-0,35	0,01	-0,60	-0,51	-0,60	-0,70	-0,52	-0,54	-0,66	0,01	0,33	-0,55	-0,53	
ILS-XNEH	0,71	1,20	0,15	0,26	0,61	0,01	0,10	0,01	-0,09	0,08	0,07	-0,06	0,61	0,94	0,06	0,08	0,61

Tabela B.18 – Resultado teste Fisher instâncias N100-M20 (DVC1)

	IG1- CHINS	IG1- HC1	IG1- HC2	IG1- NEH	IG1- XCHINS	IG1- XNEH	IG2- CHINS	IG2- HC1	IG2- HC2	IG2- NEH	IG2- XCHINS	IG2- XNEH	ILS- CHINS	ILS- HC1	ILS- HC2	ILS- NEH	ILS- XCHINS
IG1-HC1	-0,68																
IG1-HC2	0,79	1,47															
IG1-NEH	0,64	1,32	-0,15														
IG1-XCHINS	0,14	0,82	-0,65	-0,50													
IG1-XNEH	0,99	1,67	0,20	0,35	0,85												
IG2-CHINS	0,87	1,54	0,08	0,23	0,73	-0,13											
IG2-HC1	0,99	1,67	0,20	0,35	0,85	0,00	0,12										
IG2-HC2	1,14	1,81	0,35	0,50	1,00	0,14	0,27	0,15									
IG2-NEH	0,89	1,56	0,10	0,25	0,75	-0,11	0,02	-0,10	-0,25								
IG2-XCHINS	0,90	1,58	0,12	0,27	0,76	-0,09	0,04	-0,09	-0,23	0,02							
IG2-XNEH	1,08	1,76	0,30	0,45	0,94	0,09	0,22	0,09	-0,05	0,20	0,18						
ILS-CHINS	0,14	0,82	-0,65	-0,50	0,00	-0,85	-0,73	-0,85	-1,00	-0,75	-0,76	-0,94					
ILS-HC1	-0,32	0,36	-1,11	-0,96	-0,46	-1,31	-1,19	-1,31	-1,46	-1,21	-1,22	-1,40	-0,46				
ILS-HC2	0,93	1,60	0,14	0,29	0,79	-0,07	0,06	-0,06	-0,21	0,04	0,02	-0,16	0,78	1,25			
ILS-NEH	0,89	1,57	0,10	0,25	0,75	-0,10	0,02	-0,10	-0,25	0,00	-0,01	-0,19	0,75	1,21	-0,04		
ILS-XCHINS	0,15	0,83	-0,64	-0,49	0,01	-0,84	-0,72	-0,84	-0,99	-0,74	-0,75	-0,93	0,01	0,47	-0,78	-0,74	
ILS-XNEH	1,00	1,68	0,21	0,36	0,86	0,01	0,14	0,01	-0,13	0,12	0,10	-0,08	0,86	1,32	0,08	0,11	0,85

Tabela B.19 – Resultado teste Tukey instâncias N200-M10 (DVC1)

	IG1- CHINS	IG1- HC1	IG1- HC2	IG1- NEH	IG1- XCHINS	IG1- XNEH	IG2- CHINS	IG2- HC1	IG2- HC2	IG2- NEH	IG2- XCHINS	IG2- XNEH	ILS- CHINS	ILS- HC1	ILS- HC2	ILS- NEH	ILS- XCHINS
IG1-HC1	0,01																
IG1-HC2	0,83	0,82															
IG1-NEH	-0,52	-0,53	-1,35														
IG1-XCHINS	0,14	0,13	-0,69	0,66													
IG1-XNEH	-0,20	-0,21	-1,03	0,32	-0,34												
IG2-CHINS	0,15	0,14	-0,67	0,68	0,02	0,36											
IG2-HC1	0,80	0,79	-0,03	1,32	0,66	1,00	0,64										
IG2-HC2	1,02	1,01	0,19	1,54	0,89	1,23	0,87	0,23									
IG2-NEH	0,17	0,16	-0,65	0,69	0,04	0,38	0,02	-0,62	-0,85								
IG2-XCHINS	0,55	0,54	-0,27	1,08	0,42	0,76	0,40	-0,24	-0,47	0,38							
IG2-XNEH	0,20	0,19	-0,63	0,72	0,07	0,40	0,05	-0,60	-0,82	0,03	-0,35						
ILS-CHINS	0,03	0,02	-0,80	0,55	-0,11	0,23	-0,13	-0,77	-1,00	-0,15	-0,53	-0,17					
ILS-HC1	0,09	0,08	-0,74	0,61	-0,04	0,29	-0,06	-0,71	-0,93	-0,08	-0,46	-0,11	0,06				
ILS-HC2	0,88	0,87	0,05	1,40	0,74	1,08	0,73	0,08	-0,14	0,71	0,33	0,68	0,85	0,79			
ILS-NEH	-0,49	-0,50	-1,32	0,03	-0,63	-0,29	-0,65	-1,29	-1,51	-0,66	-1,05	-0,69	-0,52	-0,58	-1,37		
ILS-XCHINS	0,13	0,12	-0,70	0,65	-0,01	0,33	-0,02	-0,67	-0,89	-0,04	-0,42	-0,07	0,10	0,04	-0,75	0,62	
ILS-XNEH	-0,22	-0,22	-1,04	0,31	-0,35	-0,01	-0,37	-1,01	-1,24	-0,39	-0,77	-0,42	-0,24	-0,31	-1,09	0,28	-0,35

Tabela B.20 – Resultado teste Fisher instâncias N200-M10 (DVC1)

	IG1- CHINS	IG1- HC1	IG1- HC2	IG1- NEH	IG1- XCHINS	IG1- XNEH	IG2- CHINS	IG2- HC1	IG2- HC2	IG2- NEH	IG2- XCHINS	IG2- XNEH	ILS- CHINS	ILS- HC1	ILS- HC2	ILS- NEH	ILS- XCHINS
IG1-HC1	0,01																
IG1-HC2	1,16	1,15															
IG1-NEH	-0,73	-0,75	-1,90														
IG1-XCHINS	0,19	0,18	-0,97	0,92													
IG1-XNEH	-0,29	-0,30	-1,45	0,45	-0,48												
IG2-CHINS	0,22	0,20	-0,95	0,95	0,03	0,50											
IG2-HC1	1,12	1,11	-0,04	1,85	0,93	1,41	0,90										
IG2-HC2	1,44	1,42	0,27	2,17	1,25	1,72	1,22	0,32									
IG2-NEH	0,24	0,23	-0,92	0,98	0,05	0,53	0,03	-0,88	-1,19								
IG2-XCHINS	0,78	0,76	-0,39	1,51	0,59	1,06	0,56	-0,34	-0,66	0,54							
IG2-XNEH	0,28	0,27	-0,88	1,02	0,09	0,57	0,07	-0,84	-1,15	0,04	-0,50						
ILS-CHINS	0,04	0,02	-1,13	0,77	-0,15	0,32	-0,18	-1,08	-1,40	-0,21	-0,74	-0,24					
ILS-HC1	0,13	0,11	-1,04	0,86	-0,06	0,41	-0,09	-0,99	-1,31	-0,12	-0,65	-0,15	0,09				
ILS-HC2	1,24	1,22	0,07	1,97	1,05	1,52	1,02	0,12	-0,20	0,99	0,46	0,95	1,20	1,11			
ILS-NEH	-0,69	-0,71	-1,85	0,04	-0,88	-0,41	-0,91	-1,81	-2,13	-0,93	-1,47	-0,97	-0,73	-0,82	-1,93		
ILS-XCHINS	0,18	0,17	-0,98	0,92	-0,01	0,47	-0,03	-0,94	-1,25	-0,06	-0,59	-0,10	0,15	0,06	-1,05	0,87	
ILS-XNEH	-0,30	-0,32	-1,47	0,43	-0,49	-0,02	-0,52	-1,42	-1,74	-0,54	-1,08	-0,58	-0,34	-0,43	-1,54	0,39	-0,49

Tabela B.21 – Resultado teste Tukey instâncias N200-M20 (DVC1)

	IG1- CHINS	IG1- HC1	IG1- HC2	IG1- NEH	IG1- XCHINS	IG1- XNEH	IG2- CHINS	IG2- HC1	IG2- HC2	IG2- NEH	IG2- XCHINS	IG2- XNEH	ILS- CHINS	ILS- HC1	ILS- HC2	ILS- NEH	ILS- XCHINS
IG1-HC1	-1,09																
IG1-HC2	-0,21	0,87															
IG1-NEH	-0,49	0,60	-0,28														
IG1-XCHINS	0,14	1,23	0,35	0,63													
IG1-XNEH	-0,38	0,71	-0,16	0,11	-0,51												
IG2-CHINS	0,34	1,42	0,55	0,82	0,20	0,71											
IG2-HC1	0,14	1,22	0,35	0,62	0,00	0,51	-0,20										
IG2-HC2	-0,01	1,08	0,21	0,48	-0,14	0,37	-0,34	-0,14									
IG2-NEH	0,34	1,42	0,55	0,82	0,20	0,71	0,00	0,20	0,34								
IG2-XCHINS	0,13	1,22	0,34	0,62	-0,01	0,51	-0,20	0,00	0,14	-0,21							
IG2-XNEH	0,28	1,37	0,50	0,77	0,15	0,66	-0,05	0,15	0,29	-0,05	0,15						
ILS-CHINS	0,01	1,10	0,23	0,50	-0,12	0,39	-0,32	-0,12	0,02	-0,32	-0,12	-0,27					
ILS-HC1	-1,17	-0,08	-0,96	-0,68	-1,31	-0,80	-1,51	-1,31	-1,16	-1,51	-1,30	-1,46	-1,19				
ILS-HC2	-0,18	0,90	0,03	0,30	-0,32	0,19	-0,52	-0,32	-0,18	-0,52	-0,32	-0,47	-0,20	0,99			
ILS-NEH	-0,48	0,61	-0,27	0,01	-0,62	-0,11	-0,82	-0,62	-0,48	-0,82	-0,61	-0,77	-0,50	0,69	-0,30		
ILS-XCHINS	0,17	1,26	0,38	0,66	0,03	0,55	-0,16	0,04	0,18	-0,16	0,04	-0,11	0,16	1,34	0,36	0,65	
ILS-XNEH	-0,35	0,74	-0,14	0,14	-0,49	0,02	-0,69	-0,49	-0,34	-0,69	-0,48	-0,64	-0,37	0,82	-0,17	0,13	-0,52

Tabela B.22 – Resultado teste Fisher instâncias N200-M20 (DVC1)

	IG1- CHINS	IG1- HC1	IG1- HC2	IG1- NEH	IG1- XCHINS	IG1- XNEH	IG2- CHINS	IG2- HC1	IG2- HC2	IG2- NEH	IG2- XCHINS	IG2- XNEH	ILS- CHINS	ILS- HC1	ILS- HC2	ILS- NEH	ILS- XCHINS
IG1-HC1	-1,53																
IG1-HC2	-0,30	1,23															
IG1-NEH	-0,69	0,84	-0,39														
IG1-XCHINS	0,20	1,72	0,49	0,88													
IG1-XNEH	-0,53	1,00	-0,23	0,16	-0,72												
IG2-CHINS	0,47	2,00	0,77	1,16	0,28	1,00											
IG2-HC1	0,19	1,72	0,49	0,88	-0,01	0,72	-0,28										
IG2-HC2	-0,01	1,52	0,29	0,68	-0,20	0,52	-0,48	-0,20									
IG2-NEH	0,47	2,00	0,77	1,16	0,28	1,00	0,00	0,28	0,48								
IG2-XCHINS	0,18	1,71	0,48	0,87	-0,01	0,71	-0,29	-0,01	0,19	-0,29							
IG2-XNEH	0,40	1,93	0,70	1,09	0,20	0,93	-0,07	0,21	0,41	-0,07	0,22						
ILS-CHINS	0,02	1,55	0,32	0,71	-0,17	0,55	-0,45	-0,17	0,03	-0,45	-0,16	-0,38					
ILS-HC1	-1,65	-0,12	-1,35	-0,96	-1,84	-1,12	-2,12	-1,84	-1,64	-2,12	-1,83	-2,05	-1,67				
ILS-HC2	-0,26	1,27	0,04	0,43	-0,45	0,27	-0,73	-0,45	-0,25	-0,73	-0,44	-0,66	-0,28	1,39			
ILS-NEH	-0,68	0,85	-0,38	0,01	-0,87	-0,15	-1,15	-0,87	-0,67	-1,15	-0,86	-1,08	-0,70	0,97	-0,42		
ILS-XCHINS	0,24	1,77	0,54	0,93	0,05	0,77	-0,23	0,05	0,25	-0,23	0,06	-0,16	0,22	1,89	0,50	0,92	
ILS-XNEH	-0,49	1,04	-0,19	0,19	-0,69	0,03	-0,96	-0,68	-0,48	-0,97	-0,68	-0,89	-0,51	1,15	-0,23	0,18	-0,73

Tabela B.23 – Resultado teste Tukey instâncias N20-M20 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,64																
IG1-CHINS	1,30	0,66															
IG1-XCHINS	1,50	0,85	0,19														
IG1-HC1	1,14	0,49	-0,16	-0,36													
IG1-HC2	1,41	0,77	0,11	-0,08	0,27												
IG2-NEH	1,71	1,07	0,41	0,22	0,58	0,30											
IG2-XNEH	1,70	1,06	0,40	0,21	0,56	0,29	-0,01										
IG2-CHINS	1,78	1,14	0,48	0,28	0,64	0,37	0,07	0,08									
IG2-XCHINS	1,71	1,07	0,41	0,22	0,58	0,30	0,00	0,01	-0,07								
IG2-HC1	1,76	1,12	0,46	0,27	0,63	0,35	0,05	0,06	-0,02	0,05							
IG2-HC2	1,72	1,08	0,42	0,22	0,58	0,31	0,01	0,02	-0,06	0,01	-0,04						
ILS-NEH	0,52	-0,12	-0,78	-0,97	-0,62	-0,89	-1,19	-1,18	-1,26	-1,19	-1,24	-1,20					
ILS-XNEH	0,54	-0,10	-0,76	-0,96	-0,60	-0,87	-1,17	-1,16	-1,24	-1,17	-1,22	-1,18	0,02				
ILS-CHINS	1,54	0,90	0,24	0,04	0,40	0,13	-0,17	-0,16	-0,24	-0,17	-0,22	-0,18	1,02	1,00			
ILS-XCHINS	1,65	1,00	0,35	0,15	0,51	0,23	-0,07	-0,06	-0,13	-0,07	-0,12	-0,07	1,13	1,11	0,11		
ILS-HC1	1,39	0,74	0,09	-0,11	0,25	-0,02	-0,33	-0,31	-0,39	-0,33	-0,38	-0,33	0,87	0,85	-0,15	-0,26	
ILS-HC2	1,47	0,83	0,17	-0,03	0,33	0,06	-0,24	-0,23	-0,31	-0,24	-0,29	-0,25	0,95	0,93	-0,07	-0,18	0,08

Tabela B.24 – Resultado teste Fisher instâncias N20-M20 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,90																
IG1-CHINS	1,83	0,92															
IG1-XCHINS	2,10	1,20	0,27														
IG1-HC1	1,60	0,70	-0,23	-0,50													
IG1-HC2	1,98	1,08	0,15	-0,12	0,38												
IG2-NEH	2,41	1,50	0,58	0,31	0,81	0,42											
IG2-XNEH	2,39	1,49	0,56	0,29	0,79	0,41	-0,02										
IG2-CHINS	2,50	1,60	0,67	0,40	0,90	0,52	0,09	0,11									
IG2-XCHINS	2,41	1,50	0,58	0,31	0,81	0,42	0,00	0,02	-0,09								
IG2-HC1	2,48	1,58	0,65	0,38	0,88	0,50	0,07	0,09	-0,02	0,07							
IG2-HC2	2,42	1,51	0,59	0,32	0,82	0,43	0,01	0,03	-0,08	0,01	-0,06						
ILS-NEH	0,73	-0,17	-1,10	-1,37	-0,87	-1,25	-1,68	-1,66	-1,77	-1,68	-1,75	-1,69					
ILS-XNEH	0,76	-0,15	-1,07	-1,34	-0,84	-1,23	-1,65	-1,63	-1,74	-1,65	-1,72	-1,66	0,03				
ILS-CHINS	2,16	1,26	0,34	0,06	0,57	0,18	-0,24	-0,23	-0,34	-0,24	-0,32	-0,25	1,43	1,41			
ILS-XCHINS	2,31	1,41	0,48	0,21	0,71	0,33	-0,09	-0,08	-0,19	-0,09	-0,17	-0,10	1,58	1,56	0,15		
ILS-HC1	1,95	1,05	0,12	-0,15	0,35	-0,03	-0,46	-0,44	-0,55	-0,46	-0,53	-0,47	1,22	1,19	-0,21	-0,36	
ILS-HC2	2,07	1,16	0,24	-0,04	0,47	0,08	-0,34	-0,33	-0,44	-0,34	-0,41	-0,35	1,33	1,31	-0,10	-0,25	0,12

Tabela B.25 – Resultado teste Tukey instâncias N50-M5 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,39																
IG1-CHINS	1,50	1,11															
IG1-XCHINS	1,58	1,19	0,08														
IG1-HC1	1,12	0,73	-0,38	-0,46													
IG1-HC2	1,34	0,95	-0,16	-0,24	0,22												
IG2-NEH	1,61	1,22	0,11	0,03	0,49	0,27											
IG2-XNEH	1,57	1,18	0,07	-0,01	0,45	0,23	-0,04										
IG2-CHINS	1,54	1,15	0,04	-0,04	0,42	0,20	-0,07	-0,03									
IG2-XCHINS	1,61	1,22	0,11	0,03	0,48	0,27	-0,01	0,04	0,06								
IG2-HC1	1,57	1,18	0,07	-0,01	0,44	0,23	-0,05	0,00	0,02	-0,04							
IG2-HC2	1,46	1,07	-0,04	-0,12	0,34	0,12	-0,15	-0,11	-0,08	-0,15	-0,11						
ILS-NEH	0,47	0,08	-1,03	-1,11	-0,65	-0,87	-1,14	-1,10	-1,07	-1,14	-1,10	-0,99					
ILS-XNEH	0,55	0,16	-0,95	-1,03	-0,57	-0,79	-1,06	-1,02	-0,99	-1,06	-1,02	-0,91	0,08				
ILS-CHINS	1,62	1,23	0,12	0,04	0,50	0,28	0,01	0,05	0,08	0,01	0,05	0,16	1,15	1,07			
ILS-XCHINS	1,77	1,38	0,27	0,19	0,65	0,43	0,15	0,20	0,23	0,16	0,20	0,31	1,30	1,22	0,15		
ILS-HC1	1,30	0,91	-0,20	-0,28	0,18	-0,04	-0,32	-0,27	-0,25	-0,31	-0,27	-0,16	0,83	0,75	-0,32	-0,47	
ILS-HC2	1,60	1,21	0,10	0,02	0,48	0,26	-0,01	0,03	0,06	-0,01	0,03	0,14	1,13	1,05	-0,02	-0,17	0,30

Tabela B.26 – Resultado teste Fisher instâncias N50-M5 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,55																
IG1-CHINS	2,11	1,56															
IG1-XCHINS	2,22	1,67	0,11														
IG1-HC1	1,58	1,03	-0,53	-0,64													
IG1-HC2	1,88	1,33	-0,23	-0,34	0,30												
IG2-NEH	2,27	1,72	0,16	0,05	0,69	0,39											
IG2-XNEH	2,21	1,66	0,09	-0,02	0,63	0,32	-0,06										
IG2-CHINS	2,17	1,62	0,06	-0,05	0,59	0,29	-0,10	-0,04									
IG2-XCHINS	2,26	1,71	0,15	0,04	0,68	0,38	-0,01	0,05	0,09								
IG2-HC1	2,20	1,65	0,09	-0,02	0,62	0,32	-0,07	0,00	0,03	-0,06							
IG2-HC2	2,06	1,51	-0,06	-0,17	0,48	0,17	-0,21	-0,15	-0,11	-0,20	-0,15						
ILS-NEH	0,66	0,11	-1,45	-1,56	-0,92	-1,22	-1,61	-1,54	-1,51	-1,60	-1,54	-1,39					
ILS-XNEH	0,77	0,22	-1,34	-1,45	-0,81	-1,11	-1,50	-1,43	-1,40	-1,49	-1,43	-1,28	0,11				
ILS-CHINS	2,28	1,73	0,17	0,06	0,70	0,39	0,01	0,07	0,11	0,02	0,07	0,22	1,61	1,50			
ILS-XCHINS	2,49	1,94	0,38	0,27	0,91	0,60	0,22	0,28	0,32	0,23	0,28	0,43	1,82	1,71	0,21		
ILS-HC1	1,83	1,28	-0,29	-0,40	0,25	-0,06	-0,44	-0,38	-0,34	-0,43	-0,38	-0,23	1,16	1,05	-0,45	-0,66	
ILS-HC2	2,25	1,70	0,14	0,03	0,67	0,37	-0,02	0,04	0,08	-0,01	0,05	0,20	1,59	1,48	-0,03	-0,24	0,43

Tabela B.27 – Resultado teste Tukey instâncias N50-M10 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,69																
IG1-CHINS	1,93	1,24															
IG1-XCHINS	2,26	1,57	0,33														
IG1-HC1	1,80	1,11	-0,13	-0,46													
IG1-HC2	2,09	1,40	0,16	-0,17	0,29												
IG2-NEH	1,94	1,25	0,01	-0,32	0,13	-0,15											
IG2-XNEH	1,71	1,02	-0,22	-0,55	-0,09	-0,38	-0,22										
IG2-CHINS	2,15	1,46	0,22	-0,11	0,35	0,06	0,21	0,44									
IG2-XCHINS	2,03	1,34	0,10	-0,23	0,22	-0,07	0,09	0,31	-0,12								
IG2-HC1	1,76	1,07	-0,17	-0,50	-0,04	-0,33	-0,18	0,05	-0,39	-0,27							
IG2-HC2	2,04	1,35	0,11	-0,22	0,24	-0,05	0,11	0,33	-0,11	0,02	0,28						
ILS-NEH	0,66	-0,03	-1,27	-1,60	-1,15	-1,43	-1,28	-1,06	-1,49	-1,37	-1,10	-1,39					
ILS-XNEH	0,99	0,29	-0,95	-1,28	-0,82	-1,11	-0,95	-0,73	-1,17	-1,04	-0,78	-1,06	0,33				
ILS-CHINS	2,31	1,62	0,38	0,05	0,51	0,22	0,37	0,60	0,16	0,28	0,55	0,27	1,65	1,33			
ILS-XCHINS	2,30	1,61	0,37	0,04	0,49	0,21	0,36	0,58	0,15	0,27	0,54	0,26	1,64	1,31	-0,01		
ILS-HC1	2,04	1,35	0,11	-0,22	0,24	-0,05	0,10	0,33	-0,11	0,02	0,28	0,00	1,38	1,06	-0,27	-0,26	
ILS-HC2	2,17	1,48	0,24	-0,09	0,36	0,08	0,23	0,45	0,02	0,14	0,41	0,12	1,51	1,18	-0,14	-0,13	0,13

Tabela B.28 – Resultado teste Fisher instâncias N50-M10 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,97																
IG1-CHINS	2,71	1,74															
IG1-XCHINS	3,18	2,21	0,47														
IG1-HC1	2,54	1,56	-0,18	-0,64													
IG1-HC2	2,94	1,97	0,23	-0,24	0,41												
IG2-NEH	2,72	1,75	0,01	-0,45	0,19	-0,22											
IG2-XNEH	2,41	1,44	-0,30	-0,77	-0,13	-0,53	-0,31										
IG2-CHINS	3,02	2,05	0,31	-0,16	0,49	0,08	0,30	0,61									
IG2-XCHINS	2,85	1,88	0,14	-0,33	0,31	-0,09	0,12	0,44	-0,17								
IG2-HC1	2,48	1,50	-0,24	-0,70	-0,06	-0,47	-0,25	0,07	-0,55	-0,37							
IG2-HC2	2,87	1,90	0,16	-0,31	0,34	-0,07	0,15	0,46	-0,15	0,02	0,40						
ILS-NEH	0,92	-0,05	-1,79	-2,25	-1,61	-2,02	-1,80	-1,49	-2,10	-1,92	-1,55	-1,95					
ILS-XNEH	1,39	0,41	-1,33	-1,79	-1,15	-1,56	-1,34	-1,02	-1,64	-1,46	-1,09	-1,49	0,46				
ILS-CHINS	3,25	2,28	0,54	0,07	0,71	0,31	0,52	0,84	0,23	0,40	0,77	0,38	2,32	1,86			
ILS-XCHINS	3,23	2,26	0,52	0,05	0,69	0,29	0,51	0,82	0,21	0,38	0,76	0,36	2,31	1,85	-0,02		
ILS-HC1	2,87	1,90	0,16	-0,31	0,33	-0,07	0,15	0,46	-0,15	0,02	0,40	0,00	1,95	1,49	-0,38	-0,36	
ILS-HC2	3,05	2,08	0,33	-0,13	0,51	0,11	0,32	0,64	0,02	0,20	0,57	0,18	2,12	1,66	-0,20	-0,18	0,18

Tabela B.29 – Resultado teste Tukey instâncias N50-M20 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,54																
IG1-CHINS	1,48	0,94															
IG1-XCHINS	1,48	0,94	0,00														
IG1-HC1	0,89	0,35	-0,59	-0,60													
IG1-HC2	1,11	0,58	-0,37	-0,37	0,23												
IG2-NEH	0,96	0,42	-0,52	-0,53	0,07	-0,16											
IG2-XNEH	0,97	0,43	-0,51	-0,51	0,09	-0,14	0,01										
IG2-CHINS	1,48	0,94	0,00	0,00	0,60	0,37	0,52	0,51									
IG2-XCHINS	1,43	0,89	-0,05	-0,06	0,54	0,31	0,47	0,45	-0,05								
IG2-HC1	0,87	0,33	-0,61	-0,61	-0,02	-0,24	-0,09	-0,10	-0,61	-0,56							
IG2-HC2	1,09	0,55	-0,39	-0,40	0,20	-0,03	0,13	0,11	-0,39	-0,34	0,22						
ILS-NEH	0,74	0,20	-0,74	-0,75	-0,15	-0,38	-0,22	-0,24	-0,75	-0,69	-0,13	-0,35					
ILS-XNEH	0,75	0,21	-0,73	-0,73	-0,13	-0,36	-0,21	-0,22	-0,73	-0,67	-0,12	-0,33	0,02				
ILS-CHINS	1,45	0,91	-0,03	-0,04	0,56	0,33	0,49	0,47	-0,03	0,02	0,58	0,36	0,71	0,70			
ILS-XCHINS	1,58	1,04	0,09	0,09	0,69	0,46	0,62	0,60	0,09	0,15	0,70	0,49	0,84	0,82	0,13		
ILS-HC1	0,95	0,41	-0,53	-0,53	0,07	-0,16	-0,01	-0,02	-0,53	-0,47	0,08	-0,13	0,22	0,20	-0,50	-0,62	
ILS-HC2	1,28	0,74	-0,20	-0,20	0,39	0,17	0,32	0,31	-0,20	-0,15	0,41	0,19	0,54	0,53	-0,17	-0,29	0,33

Tabela B.30 – Resultado teste Fisher instâncias N50-M20 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,76																
IG1-CHINS	2,08	1,32															
IG1-XCHINS	2,09	1,33	0,00														
IG1-HC1	1,25	0,49	-0,84	-0,84													
IG1-HC2	1,57	0,81	-0,51	-0,52	0,32												
IG2-NEH	1,35	0,59	-0,73	-0,74	0,10	-0,22											
IG2-XNEH	1,37	0,61	-0,71	-0,72	0,12	-0,20	0,02										
IG2-CHINS	2,08	1,33	0,00	0,00	0,84	0,52	0,74	0,71									
IG2-XCHINS	2,01	1,25	-0,08	-0,08	0,76	0,44	0,66	0,64	-0,08								
IG2-HC1	1,22	0,47	-0,86	-0,86	-0,02	-0,34	-0,12	-0,14	-0,86	-0,78							
IG2-HC2	1,53	0,77	-0,55	-0,56	0,28	-0,04	0,18	0,16	-0,55	-0,48	0,30						
ILS-NEH	1,03	0,28	-1,05	-1,05	-0,21	-0,53	-0,31	-0,33	-1,05	-0,97	-0,19	-0,49					
ILS-XNEH	1,06	0,30	-1,02	-1,03	-0,19	-0,51	-0,29	-0,31	-1,02	-0,95	-0,17	-0,47	0,02				
ILS-CHINS	2,04	1,28	-0,05	-0,05	0,79	0,47	0,69	0,67	-0,05	0,03	0,81	0,51	1,00	0,98			
ILS-XCHINS	2,21	1,46	0,13	0,13	0,97	0,65	0,87	0,85	0,13	0,21	0,99	0,69	1,18	1,16	0,18		
ILS-HC1	1,34	0,58	-0,74	-0,75	0,09	-0,23	-0,01	-0,03	-0,74	-0,67	0,12	-0,19	0,30	0,28	-0,70	-0,88	
ILS-HC2	1,80	1,04	-0,28	-0,29	0,55	0,23	0,45	0,43	-0,28	-0,21	0,58	0,27	0,76	0,74	-0,24	-0,41	0,46

Tabela B.31 – Resultado teste Tukey instâncias N100-M5 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,43																
IG1-CHINS	1,49	1,07															
IG1-XCHINS	1,66	1,24	0,17														
IG1-HC1	1,47	1,05	-0,02	-0,19													
IG1-HC2	1,50	1,08	0,01	-0,16	0,03												
IG2-NEH	1,33	0,90	-0,16	-0,33	-0,14	-0,17											
IG2-XNEH	1,53	1,11	0,04	-0,13	0,06	0,03	0,20										
IG2-CHINS	1,64	1,22	0,15	-0,02	0,17	0,14	0,31	0,11									
IG2-XCHINS	1,65	1,22	0,15	-0,01	0,17	0,15	0,32	0,11	0,01								
IG2-HC1	1,56	1,13	0,07	-0,10	0,08	0,06	0,23	0,03	-0,08	-0,09							
IG2-HC2	1,71	1,29	0,22	0,05	0,24	0,21	0,38	0,18	0,07	0,07	0,15						
ILS-NEH	0,25	-0,18	-1,25	-1,42	-1,23	-1,26	-1,08	-1,29	-1,40	-1,40	-1,31	-1,47					
ILS-XNEH	0,45	0,02	-1,05	-1,21	-1,03	-1,05	-0,88	-1,09	-1,19	-1,20	-1,11	-1,26	0,20				
ILS-CHINS	1,58	1,16	0,09	-0,08	0,11	0,08	0,25	0,05	-0,06	-0,06	0,02	-0,13	1,34	1,14			
ILS-XCHINS	1,73	1,31	0,24	0,07	0,26	0,23	0,40	0,20	0,09	0,08	0,17	0,02	1,49	1,28	0,15		
ILS-HC1	1,55	1,13	0,06	-0,11	0,08	0,05	0,22	0,02	-0,09	-0,09	-0,01	-0,16	1,31	1,10	-0,03	-0,18	
ILS-HC2	1,65	1,22	0,16	-0,01	0,17	0,15	0,32	0,12	0,01	0,00	0,09	-0,06	1,40	1,20	0,06	-0,08	0,10

Tabela B.32 – Resultado teste Fisher instâncias N100-M5 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,60																
IG1-CHINS	2,10	1,50															
IG1-XCHINS	2,33	1,74	0,23														
IG1-HC1	2,07	1,47	-0,03	-0,26													
IG1-HC2	2,11	1,51	0,01	-0,23	0,04												
IG2-NEH	1,87	1,27	-0,23	-0,47	-0,20	-0,24											
IG2-XNEH	2,16	1,56	0,06	-0,18	0,08	0,05	0,29										
IG2-CHINS	2,31	1,71	0,21	-0,03	0,23	0,20	0,44	0,15									
IG2-XCHINS	2,31	1,72	0,21	-0,02	0,24	0,20	0,44	0,16	0,01								
IG2-HC1	2,19	1,59	0,09	-0,14	0,12	0,08	0,32	0,04	-0,11	-0,12							
IG2-HC2	2,41	1,81	0,31	0,07	0,33	0,30	0,54	0,25	0,10	0,09	0,21						
ILS-NEH	0,34	-0,25	-1,75	-1,99	-1,73	-1,76	-1,52	-1,81	-1,96	-1,97	-1,85	-2,06					
ILS-XNEH	0,63	0,03	-1,47	-1,70	-1,44	-1,48	-1,24	-1,53	-1,68	-1,68	-1,56	-1,78	0,29				
ILS-CHINS	2,23	1,63	0,13	-0,11	0,15	0,12	0,36	0,07	-0,08	-0,09	0,03	-0,18	1,88	1,60			
ILS-XCHINS	2,43	1,83	0,33	0,10	0,36	0,32	0,56	0,28	0,13	0,12	0,24	0,03	2,09	1,80	0,21		
ILS-HC1	2,18	1,58	0,08	-0,15	0,11	0,07	0,31	0,02	-0,13	-0,13	-0,01	-0,23	1,84	1,55	-0,05	-0,25	
ILS-HC2	2,32	1,72	0,22	-0,02	0,25	0,21	0,45	0,16	0,01	0,00	0,13	-0,09	1,97	1,69	0,09	-0,11	0,14

Tabela B.33 – Resultado teste Tukey instâncias N100-M10 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,63																
IG1-CHINS	1,70	1,07															
IG1-XCHINS	2,04	1,41	0,34														
IG1-HC1	1,58	0,95	-0,12	-0,46													
IG1-HC2	1,86	1,22	0,16	-0,18	0,28												
IG2-NEH	1,55	0,92	-0,15	-0,49	-0,03	-0,31											
IG2-XNEH	1,02	0,39	-0,68	-1,02	-0,56	-0,83	-0,53										
IG2-CHINS	2,00	1,37	0,30	-0,03	0,43	0,15	0,45	0,98									
IG2-XCHINS	2,09	1,46	0,39	0,05	0,51	0,23	0,54	1,07	0,09								
IG2-HC1	1,86	1,23	0,16	-0,18	0,28	0,00	0,31	0,84	-0,14	-0,23							
IG2-HC2	1,88	1,25	0,18	-0,16	0,30	0,03	0,33	0,86	-0,12	-0,21	0,02						
ILS-NEH	0,60	-0,03	-1,10	-1,43	-0,98	-1,25	-0,95	-0,42	-1,40	-1,49	-1,26	-1,28					
ILS-XNEH	0,85	0,22	-0,85	-1,19	-0,73	-1,01	-0,70	-0,17	-1,16	-1,24	-1,01	-1,03	0,25				
ILS-CHINS	1,87	1,24	0,17	-0,17	0,29	0,01	0,32	0,85	-0,13	-0,22	0,01	-0,01	1,27	1,02			
ILS-XCHINS	2,12	1,49	0,42	0,08	0,54	0,26	0,57	1,10	0,11	0,03	0,26	0,24	1,52	1,27	0,25		
ILS-HC1	1,81	1,18	0,11	-0,23	0,23	-0,05	0,26	0,79	-0,19	-0,28	-0,05	-0,07	1,21	0,96	-0,06	-0,31	
ILS-HC2	1,99	1,36	0,29	-0,04	0,42	0,14	0,44	0,97	-0,01	-0,10	0,13	0,11	1,39	1,14	0,12	-0,12	0,18

Tabela B.34 – Resultado teste Fisher instâncias N100-M10 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,89																
IG1-CHINS	2,39	1,50															
IG1-XCHINS	2,86	1,98	0,47														
IG1-HC1	2,22	1,33	-0,17	-0,65													
IG1-HC2	2,61	1,72	0,22	-0,25	0,39												
IG2-NEH	2,18	1,29	-0,21	-0,68	-0,04	-0,43											
IG2-XNEH	1,43	0,55	-0,95	-1,43	-0,78	-1,17	-0,74										
IG2-CHINS	2,81	1,93	0,43	-0,05	0,60	0,21	0,64	1,38									
IG2-XCHINS	2,94	2,05	0,55	0,08	0,72	0,33	0,76	1,50	0,12								
IG2-HC1	2,61	1,73	0,22	-0,25	0,40	0,00	0,43	1,18	-0,20	-0,33							
IG2-HC2	2,64	1,76	0,26	-0,22	0,43	0,04	0,46	1,21	-0,17	-0,29	0,03						
ILS-NEH	0,85	-0,04	-1,54	-2,02	-1,37	-1,76	-1,33	-0,59	-1,97	-2,09	-1,77	-1,80					
ILS-XNEH	1,19	0,30	-1,20	-1,67	-1,03	-1,42	-0,99	-0,24	-1,62	-1,75	-1,42	-1,45	0,35				
ILS-CHINS	2,63	1,74	0,24	-0,23	0,41	0,02	0,45	1,19	-0,19	-0,31	0,02	-0,02	1,78	1,44			
ILS-XCHINS	2,98	2,09	0,59	0,11	0,76	0,37	0,80	1,54	0,16	0,04	0,36	0,33	2,13	1,78	0,35		
ILS-HC1	2,54	1,65	0,15	-0,32	0,32	-0,07	0,36	1,11	-0,27	-0,40	-0,07	-0,10	1,70	1,35	-0,09	-0,43	
ILS-HC2	2,80	1,91	0,41	-0,06	0,58	0,19	0,62	1,37	-0,01	-0,14	0,19	0,16	1,95	1,61	0,17	-0,18	0,26

Tabela B.35 – Resultado teste Tukey instâncias N100-M20 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	1,09																
IG1-CHINS	2,62	1,52															
IG1-XCHINS	3,04	1,94	0,42														
IG1-HC1	2,00	0,91	-0,61	-1,03													
IG1-HC2	2,42	1,33	-0,19	-0,61	0,42												
IG2-NEH	1,35	0,26	-1,26	-1,68	-0,65	-1,07											
IG2-XNEH	1,16	0,07	-1,46	-1,87	-0,84	-1,26	-0,19										
IG2-CHINS	2,95	1,86	0,33	-0,08	0,95	0,53	1,60	1,79									
IG2-XCHINS	3,01	1,92	0,39	-0,02	1,01	0,59	1,66	1,85	0,06								
IG2-HC1	2,19	1,09	-0,43	-0,85	0,18	-0,24	0,83	1,03	-0,76	-0,82							
IG2-HC2	2,43	1,34	-0,19	-0,60	0,43	0,01	1,08	1,27	-0,52	-0,58	0,24						
ILS-NEH	0,87	-0,22	-1,75	-2,16	-1,13	-1,55	-0,48	-0,29	-2,08	-2,14	-1,32	-1,56					
ILS-XNEH	1,36	0,26	-1,26	-1,68	-0,65	-1,07	0,00	0,20	-1,59	-1,65	-0,83	-1,07	0,49				
ILS-CHINS	2,87	1,77	0,25	-0,17	0,86	0,44	1,51	1,71	-0,08	-0,14	0,68	0,44	2,00	1,51			
ILS-XCHINS	3,06	1,97	0,44	0,03	1,06	0,64	1,71	1,90	0,11	0,05	0,87	0,63	2,19	1,70	0,19		
ILS-HC1	2,17	1,07	-0,45	-0,87	0,16	-0,26	0,81	1,01	-0,79	-0,85	-0,02	-0,27	1,29	0,81	-0,70	-0,89	
ILS-HC2	2,64	1,54	0,02	-0,40	0,63	0,21	1,28	1,48	-0,32	-0,38	0,45	0,20	1,76	1,28	-0,23	-0,43	0,47

Tabela B.36 – Resultado teste Fisher instâncias N100-M20 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	1,54																
IG1-CHINS	3,68	2,14															
IG1-XCHINS	4,27	2,73	0,59														
IG1-HC1	2,82	1,28	-0,86	-1,45													
IG1-HC2	3,41	1,87	-0,27	-0,86	0,59												
IG2-NEH	1,90	0,36	-1,78	-2,36	-0,91	-1,51											
IG2-XNEH	1,63	0,09	-2,05	-2,64	-1,19	-1,78	-0,27										
IG2-CHINS	4,15	2,61	0,47	-0,12	1,33	0,74	2,25	2,52									
IG2-XCHINS	4,23	2,70	0,55	-0,03	1,42	0,83	2,33	2,60	0,08								
IG2-HC1	3,08	1,54	-0,60	-1,19	0,26	-0,33	1,17	1,44	-1,07	-1,16							
IG2-HC2	3,42	1,88	-0,26	-0,85	0,60	0,01	1,52	1,79	-0,73	-0,82	0,34						
ILS-NEH	1,23	-0,31	-2,45	-3,04	-1,59	-2,18	-0,68	-0,40	-2,92	-3,01	-1,85	-2,19					
ILS-XNEH	1,91	0,37	-1,77	-2,36	-0,91	-1,50	0,01	0,28	-2,24	-2,32	-1,17	-1,51	0,68				
ILS-CHINS	4,03	2,49	0,35	-0,23	1,22	0,62	2,13	2,40	-0,12	-0,20	0,96	0,61	2,80	2,12			
ILS-XCHINS	4,30	2,76	0,62	0,04	1,49	0,89	2,40	2,67	0,15	0,07	1,23	0,88	3,08	2,39	0,27		
ILS-HC1	3,05	1,51	-0,63	-1,22	0,23	-0,36	1,14	1,41	-1,11	-1,19	-0,03	-0,37	1,82	1,14	-0,99	-1,26	
ILS-HC2	3,71	2,17	0,03	-0,56	0,89	0,30	1,80	2,07	-0,45	-0,53	0,63	0,29	2,48	1,80	-0,33	-0,60	0,66

Tabela B.37 – Resultado teste Tukey instâncias N200-M10 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,90																
IG1-CHINS	2,17	1,27															
IG1-XCHINS	2,76	1,86	0,59														
IG1-HC1	2,15	1,26	-0,01	-0,60													
IG1-HC2	2,18	1,28	0,01	-0,58	0,02												
IG2-NEH	1,78	0,88	-0,39	-0,98	-0,37	-0,40											
IG2-XNEH	1,86	0,97	-0,30	-0,89	-0,29	-0,31	0,08										
IG2-CHINS	2,75	1,86	0,58	-0,01	0,60	0,58	0,97	0,89									
IG2-XCHINS	2,90	2,00	0,73	0,14	0,74	0,72	1,12	1,03	0,14								
IG2-HC1	2,41	1,51	0,24	-0,35	0,26	0,23	0,63	0,55	-0,34	-0,49							
IG2-HC2	2,43	1,53	0,26	-0,33	0,27	0,25	0,65	0,56	-0,33	-0,47	0,02						
ILS-NEH	0,29	-0,60	-1,87	-2,46	-1,86	-1,88	-1,49	-1,57	-2,46	-2,60	-2,12	-2,13					
ILS-XNEH	0,97	0,07	-1,20	-1,79	-1,18	-1,21	-0,81	-0,89	-1,78	-1,93	-1,44	-1,46	0,68				
ILS-CHINS	2,39	1,50	0,23	-0,36	0,24	0,22	0,61	0,53	-0,36	-0,50	-0,02	-0,03	2,10	1,42			
ILS-XCHINS	2,78	1,88	0,61	0,02	0,62	0,60	1,00	0,91	0,02	-0,12	0,37	0,35	2,48	1,81	0,38		
ILS-HC1	2,18	1,29	0,02	-0,57	0,03	0,01	0,40	0,32	-0,57	-0,71	-0,23	-0,24	1,89	1,21	-0,21	-0,59	
ILS-HC2	2,30	1,40	0,13	-0,46	0,14	0,12	0,52	0,43	-0,46	-0,60	-0,11	-0,13	2,00	1,32	-0,10	-0,48	0,11

Tabela B.38 – Resultado teste Fisher instâncias N200-M10 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	1,26																
IG1-CHINS	3,05	1,79															
IG1-XCHINS	3,88	2,62	0,83														
IG1-HC1	3,03	1,77	-0,02	-0,85													
IG1-HC2	3,06	1,80	0,01	-0,82	0,03												
IG2-NEH	2,50	1,24	-0,54	-1,37	-0,52	-0,56											
IG2-XNEH	2,62	1,36	-0,43	-1,26	-0,41	-0,44	0,12										
IG2-CHINS	3,87	2,61	0,82	-0,01	0,84	0,81	1,37	1,25									
IG2-XCHINS	4,07	2,81	1,03	0,20	1,05	1,01	1,57	1,45	0,20								
IG2-HC1	3,39	2,13	0,34	-0,49	0,36	0,33	0,89	0,77	-0,48	-0,68							
IG2-HC2	3,41	2,15	0,36	-0,46	0,39	0,35	0,91	0,79	-0,46	-0,66	0,02						
ILS-NEH	0,41	-0,85	-2,64	-3,46	-2,62	-2,65	-2,09	-2,21	-3,46	-3,66	-2,98	-3,00					
ILS-XNEH	1,36	0,11	-1,68	-2,51	-1,66	-1,70	-1,14	-1,25	-2,50	-2,71	-2,02	-2,05	0,95				
ILS-CHINS	3,36	2,10	0,32	-0,51	0,34	0,30	0,86	0,74	-0,51	-0,71	-0,02	-0,05	2,95	2,00			
ILS-XCHINS	3,90	2,64	0,86	0,03	0,88	0,84	1,40	1,28	0,03	-0,17	0,52	0,49	3,49	2,54	0,54		
ILS-HC1	3,07	1,81	0,02	-0,81	0,04	0,01	0,57	0,45	-0,80	-1,00	-0,32	-0,34	2,66	1,71	-0,29	-0,83	
ILS-HC2	3,23	1,97	0,18	-0,65	0,20	0,17	0,73	0,61	-0,64	-0,85	-0,16	-0,18	2,82	1,86	-0,14	-0,68	0,16

Tabela B.39 – Resultado teste Tukey instâncias N200-M20 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,69																
IG1-CHINS	1,64	0,94															
IG1-XCHINS	2,17	1,47	0,53														
IG1-HC1	1,46	0,77	-0,18	-0,71													
IG1-HC2	1,41	0,72	-0,23	-0,76	-0,05												
IG2-NEH	0,85	0,15	-0,79	-1,32	-0,61	-0,56											
IG2-XNEH	1,02	0,32	-0,62	-1,15	-0,44	-0,39	0,17										
IG2-CHINS	2,16	1,47	0,52	-0,01	0,70	0,75	1,31	1,14									
IG2-XCHINS	2,26	1,57	0,62	0,09	0,80	0,85	1,41	1,24	0,10								
IG2-HC1	1,66	0,96	0,02	-0,51	0,19	0,25	0,81	0,64	-0,51	-0,61							
IG2-HC2	1,60	0,90	-0,04	-0,57	0,14	0,19	0,75	0,58	-0,56	-0,66	-0,06						
ILS-NEH	0,19	-0,50	-1,45	-1,97	-1,27	-1,22	-0,66	-0,83	-1,97	-2,07	-1,46	-1,40					
ILS-XNEH	0,87	0,18	-0,76	-1,29	-0,59	-0,54	0,03	-0,14	-1,29	-1,39	-0,78	-0,72	0,68				
ILS-CHINS	1,82	1,12	0,18	-0,35	0,36	0,41	0,97	0,80	-0,34	-0,44	0,16	0,22	1,62	0,94			
ILS-XCHINS	2,28	1,59	0,64	0,11	0,82	0,87	1,43	1,26	0,12	0,02	0,63	0,69	2,09	1,41	0,46		
ILS-HC1	1,57	0,87	-0,07	-0,60	0,11	0,16	0,72	0,55	-0,59	-0,69	-0,09	-0,03	1,37	0,69	-0,25	-0,71	
ILS-HC2	1,57	0,88	-0,07	-0,60	0,11	0,16	0,72	0,55	-0,59	-0,69	-0,08	-0,02	1,38	0,70	-0,25	-0,71	0,00

Tabela B.40 – Resultado teste Fisher instâncias N200-M20 (DVC2)

	IG1- NEH	IG1- XNEH	IG1- CHINS	IG1- XCHINS	IG1- HC1	IG1- HC2	IG2- NEH	IG2- XNEH	IG2- CHINS	IG2- XCHINS	IG2- HC1	IG2- HC2	ILS- NEH	ILS- XNEH	ILS- CHINS	ILS- XCHINS	ILS- HC1
IG1-XNEH	0,98																
IG1-CHINS	2,30	1,33															
IG1-XCHINS	3,05	2,07	0,74														
IG1-HC1	2,05	1,08	-0,25	-0,99													
IG1-HC2	1,98	1,01	-0,32	-1,06	-0,07												
IG2-NEH	1,19	0,22	-1,11	-1,85	-0,86	-0,79											
IG2-XNEH	1,43	0,46	-0,87	-1,61	-0,62	-0,55	0,24										
IG2-CHINS	3,04	2,06	0,73	-0,01	0,98	1,05	1,84	1,61									
IG2-XCHINS	3,18	2,20	0,88	0,13	1,13	1,20	1,99	1,75	0,14								
IG2-HC1	2,33	1,35	0,02	-0,72	0,27	0,34	1,13	0,90	-0,71	-0,85							
IG2-HC2	2,24	1,27	-0,06	-0,80	0,19	0,26	1,05	0,81	-0,79	-0,93	-0,08						
ILS-NEH	0,27	-0,70	-2,03	-2,78	-1,78	-1,71	-0,92	-1,16	-2,77	-2,91	-2,06	-1,97					
ILS-XNEH	1,23	0,25	-1,07	-1,82	-0,83	-0,75	0,04	-0,20	-1,81	-1,95	-1,10	-1,02	0,96				
ILS-CHINS	2,56	1,58	0,25	-0,49	0,50	0,57	1,36	1,12	-0,48	-0,62	0,23	0,31	2,28	1,33			
ILS-XCHINS	3,21	2,23	0,90	0,16	1,15	1,23	2,02	1,78	0,17	0,03	0,88	0,96	2,94	1,98	0,65		
ILS-HC1	2,20	1,23	-0,10	-0,84	0,15	0,22	1,01	0,77	-0,83	-0,97	-0,12	-0,04	1,93	0,98	-0,35	-1,00	
ILS-HC2	2,21	1,23	-0,09	-0,84	0,16	0,23	1,02	0,78	-0,83	-0,97	-0,12	-0,03	1,94	0,98	-0,35	-1,00	0,01

Apêndice C – Exemplos de Resultados de Execução

Os exemplos a seguir correspondem aos resultados das execuções do algoritmo $IG_{SA}^2 - XCHINS$ para uma instância de Taillard (TAILLARD, 1993a, 1993b) de tamanho N20-M20.

São apresentados os programas de produção e de entrega para os modelos DFC2, DVC1 e DVC2.

Para os programas de entrega são apresentados os resultados das execuções anterior e posterior à perturbação de rota.

Nas Tabelas C.1 e C.2, são apresentados os tempos de processamento e os atributos de tamanho e tempo de serviço da ordem necessários ao cálculo do tempo de produção, à verificação de capacidade e ao cálculo do tempo de atendimento (tempo de serviço mais tempo de transporte) usados por todos os modelos.

Nas Tabelas C.3, C.7 e C.11, são apresentados os tempos de transporte para os dois programas de entrega (antes e depois da perturbação de rota) e para os três modelos DFC2, DVC1 e DVC2.

Nas Tabelas C.4 a C.6, C.8 a C.10 e C.12 a C.14, são apresentados, respectivamente, os programas de produção e entrega (antes e depois da perturbação de rota) para os modelos DFC2, DVC1 e DVC2.

A explicação de cada coluna em cada tabela é apresentada no Capítulo 4 na Seção 4.10.

Tabela C.1 – Tempos de processamento

	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20
M1	94	43	6	47	45	51	73	49	31	58	19	36	54	75	7	5	82	20	31	32
M2	3	18	43	41	83	62	27	75	52	58	18	1	61	67	55	72	16	12	87	21
M3	39	23	28	31	86	19	85	90	77	4	85	81	9	52	67	77	45	7	21	26
M4	1	92	83	43	3	3	51	5	38	37	44	66	31	8	79	42	85	92	89	29
M5	63	96	50	12	15	11	33	7	4	58	27	7	7	54	31	94	25	8	61	51
M6	86	36	19	71	8	77	8	6	40	39	24	82	69	82	39	52	85	48	22	57
M7	44	25	85	86	73	58	95	13	50	39	24	55	58	79	42	98	44	16	13	74
M8	19	82	12	70	6	64	3	4	29	43	40	77	88	74	13	13	17	45	2	22
M9	55	46	68	15	55	74	42	40	88	68	67	67	8	13	74	47	3	95	36	46
M10	67	66	66	32	8	30	92	40	13	79	19	29	27	18	42	86	30	41	27	50
M11	98	79	44	90	22	72	84	42	46	72	98	12	14	6	70	1	41	60	51	82
M12	30	63	88	79	44	54	12	15	3	89	84	23	4	32	60	70	20	43	2	52
M13	21	18	98	59	17	29	42	19	17	49	71	25	22	84	50	46	60	31	64	53
M14	63	20	70	12	1	75	72	79	48	77	16	60	47	67	45	67	35	55	26	32
M15	75	56	87	52	77	78	93	65	21	35	1	15	52	94	10	61	8	79	40	29
M16	20	50	37	68	23	64	15	10	20	15	19	92	63	69	15	94	94	54	45	51
M17	57	28	27	51	42	95	94	35	26	42	31	26	14	75	6	86	32	92	71	63
M18	73	11	38	53	79	40	56	60	25	54	78	78	53	39	50	64	67	15	24	74
M19	71	18	93	74	30	86	50	12	6	80	96	10	95	17	98	29	13	64	56	81
M20	38	4	40	98	22	8	31	41	25	91	95	83	38	93	62	87	27	33	58	26
Total	1017	874	1082	1084	739	1050	1058	707	659	1087	956	925	814	1098	915	1191	829	910	826	951

Tabela C.2 – Atributos da ordem

J _j	Tamanho	Tempo Serviço
1	1118	50
2	961	43
3	1190	54
4	1192	54
5	812	36
6	1155	52
7	1163	52
8	777	35
9	724	32
10	1195	54
11	1051	47
12	1017	46
13	895	40
14	1207	54
15	1006	45
16	1310	59
17	911	41
18	1001	45
19	908	41
20	1046	47

Tabela C.3 – Tempos de transporte (DFC2)

J _i De	J _j Para	Tempo Transporte (Antes)	J _i De	J _j Para	Tempo Transporte (Depois)
0	9	659	0	9	659
9	15	25	9	2	49
15	11	65	2	15	83
11	20	165	15	11	65
20	1	95	11	1	160
1	18	127	1	18	127
18	2	261	18	20	71
2	5	739	0	5	739
5	13	22	5	19	35
13	19	137	19	17	115
19	14	60	17	13	69
14	6	102	13	14	76
6	16	76	14	6	102
16	17	455	6	16	76
17	8	707	0	8	707
8	12	41	8	12	41
12	7	90	12	7	90
7	4	45	7	4	45
4	10	114	4	10	114
10	3	131	10	3	131

Tabela C.4 – Programa de produção final (DFC2)

Seq	J _j	Tempo Proc	Distância	Data Conclusão	Espera	Data Coleta
1	9	659	659	659	837	1496
2	15	915	354	1013	483	1496
3	11	956	193	1206	290	1496
4	20	951	82	1288	208	1496
5	1	1017	98	1386	110	1496
6	18	910	78	1464	32	1496
7	2	874	11	1475	21	1496
8	5	739	162	1637	1355	2992
9	13	814	174	1811	1181	2992
10	19	826	96	1907	1085	2992
11	14	1098	359	2266	726	2992
12	6	1050	162	2428	564	2992
13	16	1191	249	2677	315	2992
14	17	829	27	2704	288	2992
15	8	707	73	2777	1711	4488
16	12	925	395	3172	1316	4488
17	7	1058	230	3402	1086	4488
18	4	1084	171	3573	915	4488
19	10	1087	111	3684	804	4488
20	3	1082	135	3819	669	4488

Tabela C.5 – Programa de entrega antes da perturbação de rota (DFC2)

Seq	Jj	Veículo	Data Coleta	Serviço Entrega	Tempo Viagem	Data Entrega	Tempo Fluxo	Cap Liq
1	9	1	1496	32	659	2187	2187	6498
2	15	1	1496	77	684	2257	4444	5492
3	11	1	1496	124	749	2369	6813	4441
4	20	1	1496	171	914	2581	9394	3395
5	1	1	1496	221	1009	2726	12120	2277
6	18	1	1496	266	1136	2898	15018	1276
7	2	1	1496	309	1397	3202	18220	315
8	5	2	2992	36	739	3767	21987	6410
9	13	2	2992	76	761	3829	25816	5515
10	19	2	2992	117	898	4007	29823	4607
11	14	2	2992	171	958	4121	33944	3400
12	6	2	2992	223	1060	4275	38219	2245
13	16	2	2992	282	1136	4410	42629	935
14	17	2	2992	323	1591	4906	47535	24
15	8	3	4488	35	707	5230	52765	6445
16	12	3	4488	81	748	5317	58082	5428
17	7	3	4488	133	838	5459	63541	4265
18	4	3	4488	187	883	5558	69099	3073
19	10	3	4488	241	997	5726	74825	1878
20	3	3	4488	295	1128	5911	80736	688

Tabela C.6 – Programa de entrega depois da perturbação de rota (DFC2)

Seq	Jj	Veículo	Data Coleta	Serviço Entrega	Tempo Viagem	Data Entrega	Tempo Fluxo	Cap Liq
1	9	1	1496	32	659	2187	2187	6498
2	2	1	1496	75	708	2279	4466	5537
3	15	1	1496	120	791	2407	6873	4531
4	11	1	1496	167	856	2519	9392	3480
5	1	1	1496	217	1016	2729	12121	2362
6	18	1	1496	262	1143	2901	15022	1361
7	20	1	1496	309	1214	3019	18041	315
8	5	2	2992	36	739	3767	21808	6410
9	19	2	2992	77	774	3843	25651	5502
10	17	2	2992	118	889	3999	29650	4591
11	13	2	2992	158	958	4108	33758	3696
12	14	2	2992	212	1034	4238	37996	2489
13	6	2	2992	264	1136	4392	42388	1334
14	16	2	2992	323	1212	4527	46915	24
15	8	3	4488	35	707	5230	52145	6445
16	12	3	4488	81	748	5317	57462	5428
17	7	3	4488	133	838	5459	62921	4265
18	4	3	4488	187	883	5558	68479	3073
19	10	3	4488	241	997	5726	74205	1878
20	3	3	4488	295	1128	5911	80116	688

Tabela C.7 – Tempos de transporte (DVC1)

J _j De	J _j Para	Tempo Transporte (Antes)	J _j De	J _j Para	Tempo Transporte (Depois)
0	8	707	0	8	707
8	5	93	8	5	93
5	13	22	5	13	22
13	17	161	13	19	137
17	19	88	19	17	115
19	11	58	17	15	50
11	15	193	15	11	65
0	12	925	0	12	925
12	16	83	12	20	83
16	3	186	20	4	26
3	10	135	4	10	114
10	4	111	10	3	131
4	20	178	3	16	102
0	18	910	0	2	874
18	6	89	2	18	11
6	14	162	18	1	78
14	7	141	1	7	102
7	1	135	7	14	100
1	2	278	14	6	102
0	9	659	0	9	659

Tabela C.8 – Programa de produção final (DVC1)

Seq	J _j	Tempo Proc	Distância	Data Conclusão	Espera	Data Coleta
1	8	707	707	707	794	1501
2	5	739	118	825	676	1501
3	13	814	174	999	502	1501
4	17	829	69	1068	433	1501
5	19	826	115	1183	318	1501
6	11	956	253	1436	65	1501
7	15	915	65	1501	0	1501
8	12	925	123	1624	680	2304
9	16	1191	307	1931	373	2304
10	3	1082	102	2033	271	2304
11	10	1087	131	2164	140	2304
12	4	1084	114	2278	26	2304
13	20	951	26	2304	0	2304
14	18	910	71	2375	489	2864
15	6	1050	181	2556	308	2864
16	14	1098	102	2658	206	2864
17	7	1058	100	2758	106	2864
18	1	1017	102	2860	4	2864
19	2	874	4	2864	0	2864
20	9	659	49	2913	0	2913

Tabela C.9 – Programa de entrega antes da perturbação de rota (DVC1)

Seq	Jj	Veículo	Data Coleta	Serviço Entrega	Tempo Viagem	Data Entrega	Tempo Fluxo	Cap Liq
1	8	1	1501	35	707	2243	2243	6445
2	5	1	1501	71	800	2372	4615	5633
3	13	1	1501	111	822	2434	7049	4738
4	17	1	1501	152	983	2636	9685	3827
5	19	1	1501	193	1071	2765	12450	2919
6	11	1	1501	240	1129	2870	15320	1868
7	15	1	1501	285	1322	3108	18428	862
8	12	2	2304	46	925	3275	21703	6205
9	16	2	2304	105	1008	3417	25120	4895
10	3	2	2304	159	1194	3657	28777	3705
11	10	2	2304	213	1329	3846	32623	2510
12	4	2	2304	267	1440	4011	36634	1318
13	20	2	2304	314	1618	4236	40870	272
14	18	3	2864	45	910	3819	44689	6221
15	6	3	2864	97	999	3960	48649	5066
16	14	3	2864	151	1161	4176	52825	3859
17	7	3	2864	203	1302	4369	57194	2696
18	1	3	2864	253	1437	4554	61748	1578
19	2	3	2864	296	1715	4875	66623	617
20	9	4	2913	32	659	3604	70227	6498

Tabela C.10 – Programa de entrega depois da perturbação de rota (DVC1)

Seq	Jj	Veículo	Data Coleta	Serviço Entrega	Tempo Viagem	Data Entrega	Tempo Fluxo	Cap Liq
1	8	1	1501	35	707	2243	2243	6445
2	5	1	1501	71	800	2372	4615	5633
3	13	1	1501	111	822	2434	7049	4738
4	19	1	1501	152	959	2612	9661	3830
5	17	1	1501	193	1074	2768	12429	2919
6	15	1	1501	238	1124	2863	15292	1913
7	11	1	1501	285	1189	2975	18267	862
8	12	2	2304	46	925	3275	21542	6205
9	20	2	2304	93	1008	3405	24947	5159
10	4	2	2304	147	1034	3485	28432	3967
11	10	2	2304	201	1148	3653	32085	2772
12	3	2	2304	255	1279	3838	35923	1582
13	16	2	2304	314	1381	3999	39922	272
14	2	3	2864	43	874	3781	43703	6261
15	18	3	2864	88	885	3837	47540	5260
16	1	3	2864	138	963	3965	51505	4142
17	7	3	2864	190	1065	4119	55624	2979
18	14	3	2864	244	1165	4273	59897	1772
19	6	3	2864	296	1267	4427	64324	617
20	9	4	2913	32	659	3604	67928	6498

Tabela C.11 – Tempos de transporte (DVC2)

J _j De	J _j Para	Tempo Transporte (Antes)	J _j De	J _j Para	Tempo Transporte (Depois)
0	8	707	0	8	707
8	5	93	8	5	93
5	13	22	5	13	22
13	17	161	13	19	137
17	19	88	19	17	115
19	11	58	17	20	27
11	20	165	20	11	82
0	18	910	0	18	910
18	10	75	18	7	91
10	4	111	7	4	45
4	7	171	4	10	114
7	6	97	10	14	187
6	14	162	14	6	102
0	2	874	0	2	874
2	1	4	2	12	49
1	3	99	12	15	123
3	16	102	15	1	122
16	12	307	1	3	99
12	15	123	3	16	102
0	9	659	0	9	659

Tabela C.12 – Programa de produção final (DVC2)

Seq	J _j	Tempo Proc	Distância	Data Conclusão	Espera	Data Coleta
1	8	707	707	707	811	1518
2	5	739	118	825	693	1518
3	13	814	174	999	519	1518
4	17	829	69	1068	450	1518
5	19	826	115	1183	335	1518
6	11	956	253	1436	82	1518
7	20	951	82	1518	0	1518
8	18	910	71	1589	599	2188
9	10	1087	197	1786	402	2188
10	4	1084	114	1900	288	2188
11	7	1058	45	1945	243	2188
12	6	1050	141	2086	102	2188
13	14	1098	102	2188	0	2188
14	2	874	4	2192	850	3042
15	1	1017	278	2470	572	3042
16	3	1082	174	2644	398	3042
17	16	1191	186	2830	212	3042
18	12	925	83	2913	129	3042
19	15	915	129	3042	0	3042
20	9	659	25	3067	0	3067

Tabela C.13 – Programa de entrega antes da perturbação de rota (DVC2)

Seq	Jj	Veículo	Data Coleta	Serviço Entrega	Tempo Viagem	Data Entrega	Tempo Fluxo	Cap Liq
1	8	1	1518	35	707	2260	2260	6445
2	5	1	1518	71	800	2389	4649	5633
3	13	1	1518	111	822	2451	7100	4738
4	17	1	1518	152	983	2653	9753	3827
5	19	1	1518	193	1071	2782	12535	2919
6	11	1	1518	240	1129	2887	15422	1868
7	20	1	1518	287	1294	3099	18521	822
8	18	2	2188	45	910	3143	21664	6221
9	10	2	2188	99	985	3272	24936	5026
10	4	2	2188	153	1096	3437	28373	3834
11	7	2	2188	205	1267	3660	32033	2671
12	6	2	2188	257	1364	3809	35842	1516
13	14	2	2188	311	1526	4025	39867	309
14	2	3	3042	43	874	3959	43826	6261
15	1	3	3042	93	878	4013	47839	5143
16	3	3	3042	147	977	4166	52005	3953
17	16	3	3042	206	1079	4327	56332	2643
18	12	3	3042	252	1386	4680	61012	1626
19	15	3	3042	297	1509	4848	65860	620
20	9	4	3067	32	659	3758	69618	6498

Tabela C.14 – Programa de entrega depois da perturbação de rota (DVC2)

Seq	Jj	Veículo	Data Coleta	Serviço Entrega	Tempo Viagem	Data Entrega	Tempo Fluxo	Cap Liq
1	8	1	1518	35	707	2260	2260	6445
2	5	1	1518	71	800	2389	4649	5633
3	13	1	1518	111	822	2451	7100	4738
4	19	1	1518	152	959	2629	9729	3830
5	17	1	1518	193	1074	2785	12514	2919
6	20	1	1518	240	1101	2859	15373	1873
7	11	1	1518	287	1183	2988	18361	822
8	18	2	2188	45	910	3143	21504	6221
9	7	2	2188	97	1001	3286	24790	5058
10	4	2	2188	151	1046	3385	28175	3866
11	10	2	2188	205	1160	3553	31728	2671
12	14	2	2188	259	1347	3794	35522	1464
13	6	2	2188	311	1449	3948	39470	309
14	2	3	3042	43	874	3959	43429	6261
15	12	3	3042	89	923	4054	47483	5244
16	15	3	3042	134	1046	4222	51705	4238
17	1	3	3042	184	1168	4394	56099	3120
18	3	3	3042	238	1267	4547	60646	1930
19	16	3	3042	297	1369	4708	65354	620
20	9	4	3067	32	659	3758	69112	6498

Apêndice D – Modelo Matemático em GAMS e *Gurobipy*

Esta seção apresenta:

- 1) o código em *Python* usando a API *Gurobipy* (*Python* para Gurobi) para o modelo matemático proposto no Capítulo 4, com os dados do exemplo de aplicação (DVC) daquele capítulo, conjuntamente com o resumo do resultado de sua resolução, usando o *solver* GUROBI v.9.1.2 e *Python* 3.8.8.
- 2) o resumo do resultado da resolução da instância N20-M20 do Apêndice C (DVC) usando a API *Gurobipy*, GUROBI v.9.1.2 e *Python* 3.8.8.
- 3) O código na linguagem GAMS v.36.1 com os dados do exemplo de aplicação (DVC) do Capítulo 4, conjuntamente com o resumo do resultado de sua resolução usando o *solver* GUROBI v.9.1.2.

Para rodar o modelo usando GAMS, basta copiar o código (tal como está) e ‘colar’ no IDE do GAMS. Para rodar o modelo em *Python*, basta copiar o código (tal como está) e ‘colar’ em algum IDE para *Python*, tal qual o *Spyder* ou o *JupyterLab*, com a API *Gurobipy* e o *solver* GUROBI devidamente instalados.

D.1 Código em *Gurobipy* para o PIPD-NWFS-DVC

D.1.1 Código para o exemplo de aplicação

```
#Modelo PLIM de programacao integrada da producao e distribuicao em sistema no-wait flow shop
usando estrategia de transporte DVC

import gurobipy as gp
from gurobipy import GRB

#Indices e conjuntos
ordens = {'0','1','2','3','4','5','6'}
veiculos = {'1'}
```

Distancias

D = {

('0','1'): 198,

('0','2'): 353,

('0','3'): 283,

('0','4'): 221,

('0','5'): 182,

('0','6'): 151,

('1','1'): 0,

('1','2'): 227,

('1','3'): 214,

('1','4'): 95,

('1','5'): 80,

('1','6'): 88,

('2','1'): 48,

('2','2'): 0,

('2','3'): 85,

('2','4'): 31,

('2','5'): 22,

('2','6'): 13,

('3','1'): 48,

('3','2'): 120,

('3','3'): 0,

('3','4'): 32,

('3','5'): 22,

('3','6'): 13,

('4','1'): 66,

('4','2'): 221,

('4','3'): 151,

('4','4'): 0,

('4','5'): 50,

('4','6'): 51,

('5','1'): 60,

('5','2'): 215,

('5','3'): 196,

('5','4'): 90,

('5','5'): 0,

('5','6'): 84,

('6','1'): 52,

```

('6','2'): 207,
('6','3'): 153,
('6','4'): 88,
('6','5'): 39,
('6','6'): 0}

```

Tempo de transporte

```

TT = {
('0','1'): 198,
('0','2'): 353,
('0','3'): 283,
('0','4'): 221,
('0','5'): 182,
('0','6'): 151,
('1','1'): 0,
('1','2'): 48,
('1','3'): 48,
('1','4'): 66,
('1','5'): 60,
('1','6'): 52,
('2','1'): 227,
('2','2'): 0,
('2','3'): 120,
('2','4'): 221,
('2','5'): 215,
('2','6'): 207,
('3','1'): 214,
('3','2'): 85,
('3','3'): 0,
('3','4'): 151,
('3','5'): 196,
('3','6'): 153,
('4','1'): 95,
('4','2'): 31,
('4','3'): 32,
('4','4'): 0,
('4','5'): 90,
('4','6'): 88,
('5','1'): 80,
('5','2'): 22,

```



```
('5','3'): 22,
```

```
('5','4'): 50,
```

```
('5','5'): 0,
```

```
('5','6'): 39,
```

```
('6','1'): 88,
```

```
('6','2'): 13,
```

```
('6','3'): 13,
```

```
('6','4'): 51,
```

```
('6','5'): 84,
```

```
('6','6'): 0}
```

```
# Tamanho da ordem
```

```
Q = {
```

```
'1': 217,
```

```
'2': 388,
```

```
'3': 311,
```

```
'4': 243,
```

```
'5': 200,
```

```
'6': 166}
```

```
# Tempo de servico
```

```
TS = {
```

```
'1': 39,
```

```
'2': 70,
```

```
'3': 56,
```

```
'4': 44,
```

```
'5': 36,
```

```
'6': 30}
```

```
# Capacidade do veiculo
```

```
CV = {
```

```
'1': 1778}
```

```
# Numero suficientemente grande
```

```
BigM = 0
```

```
values1 = D.values()
```

```
values2 = TT.values()
```

```
BigM = sum(values1) + sum(values2)
```

```
# Modelo
```

```

m = gp.Model("ppipdnwfs")

# Variaveis de decisao
y = m.addVars(ordens, ordens, vtype=GRB.BINARY, name="y")
z = m.addVars(ordens, veiculos, vtype=GRB.BINARY, name="z")
x = m.addVars(ordens, ordens, veiculos, vtype=GRB.BINARY, name="x")
C = m.addVars(ordens, name="C")
F = m.addVars(ordens, veiculos, name="F")
S = m.addVars(veiculos, name="S")

# Funcao objetivo (r1)
TFT = gp.quicksum(F[j, k] for j in ordens for k in veiculos)
m.setObjective(TFT, GRB.MINIMIZE)

#Restricoes do sistema de producao
r2 = m.addConstrs((y[i,j] == 0 for i in ordens for j in ordens if i==j), "r2")
r3 = m.addConstrs((y.sum(i,'*') <= 1 for i in ordens if i=='0'), "r3")
r4 = m.addConstrs((y.sum('*',j) <= 1 for j in ordens if j=='0'), "r4")
r5 = m.addConstrs((y.sum('*',j) == 1 for j in ordens if j!='0'), "r5")
r6 = m.addConstrs((y.sum('*',j) == y.sum(j,'*') for j in ordens if j!='0'), "r6")
r7 = m.addConstrs((C[j] == 0 for j in ordens if j=='0'), "r7")
r8 = m.addConstrs((C[i] + D[i,j] - BigM*(1-y[i,j]) <= C[j] for i in ordens for j in ordens if j!='0'), "r8")

#Restricoes do sistema de transporte
r9 = m.addConstrs((z.sum(j,'*') <= 1 for j in ordens if j!='0'), "r9")
r10 = m.addConstrs((gp.quicksum(z[j,k]*Q[j] for j in ordens if j!='0') <= CV[k] for k in veiculos), "r10")
r11 = m.addConstrs((x[i,j,k] == 0 for i in ordens for j in ordens for k in veiculos if i==j), "r11")
r12 = m.addConstrs((x.sum(i,'*',k) <= 1 for i in ordens for k in veiculos if i=='0'), "r12")
r13 = m.addConstrs((x.sum('*',j,k) <= 1 for j in ordens for k in veiculos if j=='0'), "r13")
r14 = m.addConstrs((x.sum('*',j,'*') == 1 for j in ordens if j!='0'), "r14")
r15 = m.addConstrs((x.sum('*',j,k) == z[j,k] for j in ordens for k in veiculos if j!='0'), "r15")
r16 = m.addConstrs((x.sum('*',j,k) == x.sum(j,'*',k) for j in ordens for k in veiculos if j!='0'), "r16")
r17 = m.addConstrs((S[k] >= C[j] - BigM*(1-z[j,k]) for j in ordens for k in veiculos), "r17")
r18 = m.addConstrs((F[j,k] == 0 for j in ordens for k in veiculos if j=='0'), "r18")
r19 = m.addConstrs((F[i,k] + S[k] + TT[i,j] + TS[j] - BigM*(1-x[i,j,k]) <= F[j,k] for i in ordens for j in ordens for k in veiculos if i=='0' if j!='0'), "r19")
r20 = m.addConstrs((F[i,k] + TT[i,j] + TS[j] - BigM*(1-x[i,j,k]) <= F[j,k] for i in ordens for j in ordens for k in veiculos if j!='0'), "r20")

#Salvar modelo matematico

```

```

m.write('PPIPD_DVC_PY.lp')

#Parametros de execucao
m.Params.timeLimit = 1

#Resolver
m.optimize()

#Salvar solucao
m.write("PPIPD_DVC_PY.sol")
print (" ")

#Resultados
#Sequencia producao
print ("Valores selecionados de y(ordem,ordem):")
for i, j in y.keys():
    if (abs(y[i,j].x) > 0):
        print("y(" ,i, " , " ,j, ")=",round(y[i,j].x))
print (" ")

#Data de conclusao da ordem
print ("Valores calculados de C(ordem):")
for i in C.keys():
    if (abs(C[i].x) >= 0):
        print("C(" ,i, ")=",round(C[i].x))
print (" ")

#Data de coleta do veiculo
print ("Valores calculados de S(veiculo):")
for i in S.keys():
    if (abs(S[i].x) >= 0):
        print("S(" ,i, ")=",round(S[i].x))
print (" ")

#Ordem alocada ao veiculo
print ("Valores selecionados de z(ordem,veiculo):")
for i,j in z.keys():
    if (abs(z[i,j].x) > 0):
        print("z(" ,i, " , " ,j, ")=",round(z[i,j].x))
print (" ")

```

```

#Roteiro dos veiculos
print ("Valores selecionados de x(ordem,ordem,veiculo):")
for i,j,k in x.keys():
    if (abs(x[i,j,k].x) > 0):
        print("x(",i,"",j,"",k,"")=",round(x[i,j,k].x))
print (" ")

#Data de entrega
print ("Valores calculados de F(ordem,veiculo):")
for i,j in F.keys():
    if (abs(F[i,j].x) > 0):
        print("F(",i,"",j,"")=",round(F[i,j].x))
print (" ")

#Tempo total de entrega
print ("Valor calculado de TFT =",round(m.objVal))
print (" ")

```

D.1.2 Resultado da execução do exemplo de aplicação

```

Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 7.26.0 -- An enhanced Interactive Python.

Academic license - for non-commercial use only - expires 2022-08-06
Using license file C:\gurobi912\gurobi.lic
Changed value of parameter timeLimit to 1.0
Prev: inf  Min: 0.0  Max: inf  Default: inf
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 154 rows, 120 columns and 605 nonzeros
Model fingerprint: 0x3b7cb8dd
Variable types: 15 continuous, 105 integer (105 binary)
Coefficient statistics:
  Matrix range    [1e+00, 8e+03]
  Objective range [1e+00, 1e+00]
  Bounds range    [1e+00, 1e+00]
  RHS range       [1e+00, 8e+03]

```

Presolve removed 42 rows and 23 columns

Presolve time: 0.00s

Presolved: 112 rows, 97 columns, 447 nonzeros

Variable types: 13 continuous, 84 integer (84 binary)

Root relaxation: objective 7.260538e+02, 68 iterations, 0.00 seconds

Nodes		Current Node		Objective Bounds		Work	
Expl	Unexpl	Obj	Depth	IntInf	Incumbent	BestBd	Gap It/Node Time
0	0	726.05378	0	16	- 726.05378	- -	0s
0	0	779.68128	0	19	- 779.68128	- -	0s
H	0	0		7245.0000000	779.68128	89.2%	- 0s
H	0	0		7076.0000000	779.68128	89.0%	- 0s
0	0	783.31687	0	20	7076.00000	783.31687	88.9% - 0s
0	0	783.31999	0	21	7076.00000	783.31999	88.9% - 0s
0	0	803.13665	0	27	7076.00000	803.13665	88.6% - 0s

H	0	0		6903.0000000	803.13665	88.4%	- 0s
*	200	180	24	6374.0000000	1148.78509	82.0%	8.5 0s
H	231	206		6356.0000000	1148.78509	81.9%	8.5 0s
H	335	239		6200.0000000	1148.78509	81.5%	8.5 0s
H	5224	1065		6086.0000000	2744.09829	54.9%	9.3 0s

Cutting planes:

Learned: 12

Gomory: 6

Implied bound: 5

Projected implied bound: 1

MIR: 17

Flow cover: 62

Inf proof: 7

RLT: 6

Relax-and-lift: 7

Explored 6878 nodes (68695 simplex iterations) in 1.00 seconds

Thread count was 8 (of 8 available processors)

Solution count 10: 6086 6200 6356 ... 7245

Time limit reached

Best objective 6.086000000000e+03, best bound 2.812806230915e+03, gap 53.7823%

Valores seleccionados de $y(\text{ordem}, \text{ordem})$:

$$y(2, 4) = 1$$

$$y(1, 0) = 1$$

$$y(0, 6) = 1$$

$$y(5, 1) = 1$$

$$y(6, 3) = 1$$

$$y(3, 2) = 1$$

$$y(4, 5) = 1$$

Valores calculados de $C(\text{ordem})$:

$$C(2) = 424$$

$$C(1) = 565$$

$$C(0) = 0$$

$$C(5) = 505$$

$$C(6) = 151$$

$$C(3) = 304$$

$$C(4) = 455$$

Valores calculados de $S(\text{veiculo})$:

$$S(1) = 565$$

Valores seleccionados de $z(\text{ordem}, \text{veiculo})$:

$$z(2, 1) = 1$$

$$z(1, 1) = 1$$

$$z(5, 1) = 1$$

$$z(6, 1) = 1$$

$$z(3, 1) = 1$$

$$z(4, 1) = 1$$

Valores seleccionados de $x(\text{ordem}, \text{ordem}, \text{veiculo})$:

$$x(2, 0, 1) = 1$$

$$x(1, 5, 1) = 1$$

$$x(0, 6, 1) = 1$$

$$x(5, 3, 1) = 1$$

$$x(6, 4, 1) = 1$$

$$x(3, 2, 1) = 1$$

$$x(4, 1, 1) = 1$$

Valores calculados de $F(\text{ordem}, \text{veiculo})$:

$F(2, 1) = 1304$

$F(1, 1) = 975$

$F(5, 1) = 1071$

$F(6, 1) = 746$

$F(3, 1) = 1149$

$F(4, 1) = 841$

Valor calculado de TFT = 6086

D.1.3 Resultado da execução da instância N20-M20

Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]

Type "copyright", "credits" or "license" for more information.

IPython 7.26.0 -- An enhanced Interactive Python.

Academic license - for non-commercial use only - expires 2022-08-06

Using license file C:\gurobi912\gurobi.lic

Changed value of parameter timeLimit to 600.0

Prev: inf Min: 0.0 Max: inf Default: inf

Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)

Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 2629 rows, 2398 columns and 15052 nonzeros

Model fingerprint: 0x064dce84

Variable types: 109 continuous, 2289 integer (2289 binary)

Coefficient statistics:

Matrix range [1e+00, 2e+05]

Objective range [1e+00, 1e+00]

Bounds range [1e+00, 1e+00]

RHS range [1e+00, 2e+05]

Presolve removed 234 rows and 114 columns

Presolve time: 0.02s

Presolved: 2395 rows, 2284 columns, 10940 nonzeros

Variable types: 104 continuous, 2180 integer (2180 binary)

Root relaxation: objective 0.000000e+00, 681 iterations, 0.02 seconds

Nodes			Current Node			Objective Bounds			Work		
Expl	Unexpl	Obj	Depth	IntInf	Incumbent	BestBd	Gap	It/Node	Time		
H	0	0			105243.00000	0.00000	100%	-	0s		
H	0	0			101898.00000	0.00000	100%	-	0s		
	0	0	0.00000	0	63	101898.000	0.00000	100%	-	0s	
	0	0	0.00000	0	63	101898.000	0.00000	100%	-	0s	
	0	2	203.00000	0	63	101898.000	203.00000	100%	-	0s	
H	64	79			97162.000000	203.00000	100%	62.7	1s		
H	69	79			94132.000000	203.00000	100%	59.6	1s		
H	91	97			91300.000000	203.00000	100%	51.8	1s		
H	93	97			90637.000000	203.00000	100%	51.1	1s		
H	119	125			85376.000000	203.00000	100%	43.7	1s		
H	1693	1504			83006.000000	290.01702	100%	17.3	3s		
H	7861	5411			75368.000000	451.00000	99.4%	20.2	10s		

H16146	12582				75311.000000	451.00000	99.4%	16.3	14s		
H16147	12579				74352.000000	451.00000	99.4%	16.3	14s		
16864	14203	20883.7178	103	40	74352.0000	451.00000	99.4%	16.2	15s		

30281	19835	3504.68230	91	157	72731.0000	3312.07579	95.4%	15.4	40s		
31719	20732	4485.66752	171	101	72731.0000	3312.07579	95.4%	16.6	45s		
H31749	19761				71452.000000	3312.07579	95.4%	16.6	45s		
H32299	19268				70582.000000	3312.07579	95.3%	17.2	46s		
H33112	18901				70383.000000	3312.07579	95.3%	18.0	48s		
34220	19582	3481.77797	61	119	70383.0000	3321.11035	95.3%	18.4	50s		
39941	22887	11748.2005	212	72	70383.0000	3322.34079	95.3%	18.4	55s		
46813	25884	4369.61249	159	75	70383.0000	3322.34079	95.3%	17.9	60s		
H46934	25021				70321.000000	3322.34079	95.3%	17.9	60s		

H601601	458998				64999.000000	3327.56716	94.9%	16.1	553s		
602159	461691	3523.99567	121	85	64999.0000	3327.56716	94.9%	16.1	555s		
608072	464318	3525.50648	108	97	64999.0000	3327.56716	94.9%	16.1	560s		
614181	469284	3354.61782	112	97	64999.0000	3327.56716	94.9%	16.1	565s		
619105	473645	6307.79344	132	57	64999.0000	3327.56716	94.9%	16.1	570s		
624521	477127	28749.3893	201	35	64999.0000	3327.56716	94.9%	16.1	575s		
628139	480133	3572.14256	126	82	64999.0000	3327.56716	94.9%	16.1	580s		
634733	485629	3369.68644	101	94	64999.0000	3327.56716	94.9%	16.1	585s		
639502	487829	3402.90214	122	91	64999.0000	3327.56716	94.9%	16.1	590s		
H642966	491294				64993.000000	3327.56716	94.9%	16.1	593s		


```
645285 493132 4015.19573 164 77 64993.0000 3327.56716 94.9% 16.1 595s
649495 496063 4151.70855 159 65 64993.0000 3327.56716 94.9% 16.1 600s
```

Cutting planes:

Learned: 17

Gomory: 19

Implied bound: 126

Projected implied bound: 9

MIR: 5

Flow cover: 150

Zero half: 5

Relax-and-lift: 246

Explored 649980 nodes (10465818 simplex iterations) in 600.05 seconds

Thread count was 8 (of 8 available processors)

Solution count 10: 64993 64999 65124 ... 66064

Time limit reached

Best objective 6.499299999999e+04, best bound 3.327567164027e+03, gap 94.8801%

Valores seleccionados de y(ordem,ordem):

y(0 , 9)= 1

y(1 , 2)= 1

y(2 , 15)= 1

y(3 , 16)= 1

y(4 , 1)= 1

y(5 , 13)= 1

y(6 , 14)= 1

y(7 , 6)= 1

y(8 , 0)= 1

y(9 , 5)= 1

y(10 , 11)= 1

y(11 , 20)= 1

y(12 , 8)= 1

y(13 , 17)= 1

y(14 , 4)= 1

y(15 , 3)= 1

y(16 , 18)= 1

y(17 , 19)= 1

$$y(18, 10) = 1$$

$$y(19, 7) = 1$$

$$y(20, 12) = 1$$

Valores calculados de C(ordem):

$$C(0) = 0$$

$$C(1) = 1912$$

$$C(2) = 1916$$

$$C(3) = 2412$$

$$C(4) = 1874$$

$$C(5) = 830$$

$$C(6) = 1614$$

$$C(7) = 1473$$

$$C(8) = 3134$$

$$C(9) = 659$$

$$C(10) = 2828$$

$$C(11) = 2928$$

$$C(12) = 3093$$

$$C(13) = 1004$$

$$C(14) = 1716$$

$$C(15) = 2114$$

$$C(16) = 2598$$

$$C(17) = 1073$$

$$C(18) = 2631$$

$$C(19) = 1188$$

$$C(20) = 3010$$

Valores calculados de S(veiculo):

$$S(1) = 1916$$

$$S(2) = 1188$$

$$S(3) = 2631$$

$$S(4) = 3134$$

Valores selecionados de z(ordem,veiculo):

$$z(1, 1) = 1$$

$$z(2, 1) = 1$$

$$z(3, 3) = 1$$

$$z(4, 1) = 1$$

$$z(5, 2) = 1$$

$$z(6, 1) = 1$$

$z(7, 1) = 1$
 $z(8, 4) = 1$
 $z(9, 2) = 1$
 $z(10, 4) = 1$
 $z(11, 4) = 1$
 $z(12, 4) = 1$
 $z(13, 2) = 1$
 $z(14, 1) = 1$
 $z(15, 3) = 1$
 $z(16, 3) = 1$
 $z(17, 2) = 1$
 $z(18, 3) = 1$
 $z(19, 2) = 1$
 $z(20, 4) = 1$

Valores seleccionados de $x(\text{orden}, \text{orden}, \text{vehiculo})$:

$x(0, 2, 1) = 1$
 $x(0, 8, 4) = 1$
 $x(0, 9, 2) = 1$
 $x(0, 15, 3) = 1$
 $x(1, 4, 1) = 1$
 $x(2, 1, 1) = 1$
 $x(3, 0, 3) = 1$
 $x(4, 7, 1) = 1$
 $x(5, 19, 2) = 1$
 $x(6, 0, 1) = 1$
 $x(7, 14, 1) = 1$
 $x(8, 12, 4) = 1$
 $x(9, 5, 2) = 1$
 $x(10, 0, 4) = 1$
 $x(11, 10, 4) = 1$
 $x(12, 20, 4) = 1$
 $x(13, 0, 2) = 1$
 $x(14, 6, 1) = 1$
 $x(15, 18, 3) = 1$
 $x(16, 3, 3) = 1$
 $x(17, 13, 2) = 1$
 $x(18, 16, 3) = 1$
 $x(19, 17, 2) = 1$
 $x(20, 11, 4) = 1$

Valores calculados de $F(\text{ordem}, \text{veiculo})$:

$$F(1, 1) = 2887$$

$$F(2, 1) = 2833$$

$$F(3, 3) = 4095$$

$$F(4, 1) = 2979$$

$$F(5, 2) = 1966$$

$$F(6, 1) = 3510$$

$$F(7, 1) = 3202$$

$$F(8, 4) = 3876$$

$$F(9, 2) = 1879$$

$$F(10, 4) = 4376$$

$$F(11, 4) = 4222$$

$$F(12, 4) = 3963$$

$$F(13, 2) = 2307$$

$$F(14, 1) = 3356$$

$$F(15, 3) = 3591$$

$$F(16, 3) = 3855$$

$$F(17, 2) = 2198$$

$$F(18, 3) = 3763$$

$$F(19, 2) = 2042$$

$$F(20, 4) = 4093$$

Valor calculado de TFT = 64993

D.2 Código em GAMS para o PIPD-NWFS-DVC

D.2.1 Código para o exemplo de aplicação

*Modelo de programacao integrada da producao e distribuicao em sistema no-wait flow shop usando estrategia de transporte DVC

*Indices e conjuntos

set

j indice da ordem ou local (fabrica e cliente) /0*6/,

k indice do veiculo /1/;

alias(j,h,i,l,m);

* Dados de entrada

table D(i,j) distancia temporal entre ordens

	1	2	3	4	5	6
0	198	353	283	221	182	151
1	0	227	214	95	80	88
2	48	0	85	31	22	13
3	48	120	0	32	22	13
4	66	221	151	0	50	51
5	60	215	196	90	0	84
6	52	207	153	88	39	0;

table TT(l,j) tempo de transporte entre os locais l e j

	1	2	3	4	5	6
0	198	353	283	221	182	151
1	0	48	48	66	60	52
2	227	0	120	221	215	207
3	214	85	0	151	196	153
4	95	31	32	0	90	88
5	80	22	22	50	0	39
6	88	13	13	51	84	0;

parameter Q(j) tamanho (quantidade) da ordem

/

1 217,

2 388,

3 311,

4 243,

5 200,

6 166

/

parameter TS(j) tempo de servico da ordem

/

1 39,

2 70,

3 56,

4 44,

5 36,

6 30

/

parameter CV(k) capacidade do veiculo

/

1 1778

/

parameter BigM grande numero /8498/;

*Variaveis de decisao

variable

TFT tempo total (acumulado) de fluxo;

binary variable

y(i,j) ordem i precede ordem j no programa de producao,

z(j,k) apenas um veiculo k atende a um cliente j,

x(l,j,k) ordem j entregue apos ordem i pelo veiculo k;

positive variable

C(j) tempo de conclusao da ordem j,

F(j,k) tempo (de fluxo) ate a entrega da ordem j carregada pelo veiculo k,

VP(k) instante de saida do veiculo;

*FO e Restricoes

equations r1, r2, r3, r4, r5, r6, r7,
r8, r9, r10, r11, r12, r13, r14, r15, r16, r17, r18, r19, r20;

*Funcao-objetivo

r1.. TFT =e= sum((j,k), F(j,k));

*Restricoes do sistema de producao NWFS

r2(i,j)\$ (ord(i)=ord(j)).. y(i,j) =e= 0;
r3(i)\$ (ord(i)=1).. sum(j,y(i,j)) =l= 1;
r4(j)\$ (ord(j)=1).. sum(i,y(i,j)) =l= 1;
r5(j)\$ (ord(j)<>1).. sum(i,y(i,j)) =e= 1;
r6(j)\$ (ord(j)<>1).. sum(i,y(i,j)) =e= sum(h,y(j,h));
r7(j)\$ (ord(j)=1).. C(j) =e= 0;
r8(i,j)\$ (ord(j)<>1).. C(i) + D(i,j) - BigM*(1-y(i,j)) =l= C(j);

*Restricoes do sistema de transporte

r9(j)\$ (ord(j)<>1).. sum(k,z(j,k)) =l= 1;
r10(k).. sum(j, Q(j)*z(j,k)) =l= CV(k);
r11(l,j,k)\$ (ord(l)=ord(j)).. x(l,j,k) =e= 0;
r12(l,k)\$ (ord(l)=1).. sum(j,x(l,j,k)) =l= 1;
r13(j,k)\$ (ord(j)=1).. sum(l,x(l,j,k)) =l= 1;
r14(j)\$ (ord(j)<>1).. sum((l,k), x(l,j,k)) =e= 1;
r15(j,k)\$ (ord(j)<>1).. sum(l, x(l,j,k)) =l= z(j,k);
r16(j,k)\$ (ord(j)<>1).. sum(l,x(l,j,k)) =e= sum(m,x(j,m,k));
r17(j,k).. C(j) - BigM*(1-z(j,k)) =l= VP(k);
r18(j,k)\$ (ord(j)=1).. F(j,k) =e= 0;
r19(i,j,k)\$ (ord(i)=1 and ord(j)<>1).. F(i,k) + VP(k) + TT(i,j) + TS(j) - BigM*(1-x(i,j,k)) =l= F(j,k);
r20(i,j,k)\$ (ord(j)<>1).. F(i,k) + TT(i,j) + TS(j) - BigM*(1-x(i,j,k)) =l= F(j,k);

*Parametros de execucao

option mip = gurobi;
option sysout = on;

```

option limrow = 10000000;
option limcol = 10000000;
model ppidnwfs /all/;
solve ppidnwfs using mip minimizing TFT;
display TFT.I,y.I,x.I,C.I,VP.I,F.I;

```

D.2.2 Resultado da execução do exemplo de aplicação

```

GAMS 36.1.0
MODEL STATISTICS

BLOCKS OF EQUATIONS      20  SINGLE EQUATIONS      155
BLOCKS OF VARIABLES      7   SINGLE VARIABLES      121
NON ZERO ELEMENTS      613  DISCRETE VARIABLES      105

RESOURCE USAGE, LIMIT    134.157 100000000000.000
ITERATION COUNT, LIMIT  2580033 2147483647
*** This solver runs with a demo license. No commercial use.
GAMS/Gurobi demo/community requires a Gurobi license from Gurobi Optimization.
Gurobi library version 9.1.2
MIP status(2): Model was solved to optimality (subject to tolerances).
Fixed MIP status(2): Model was solved to optimality (subject to tolerances).

S O L V E   S U M M A R Y

MODEL ppidnwfs      OBJECTIVE TFT
TYPE  MIP           DIRECTION MINIMIZE
SOLVER GUROBI      FROM LINE 123

**** SOLVER STATUS   1 Normal Completion
**** MODEL STATUS    8 Integer Solution
**** OBJECTIVE VALUE      6086.0000

RESOURCE USAGE, LIMIT    98.297 100000000000.000
ITERATION COUNT, LIMIT  2580033 2147483647
*** This solver runs with a demo license. No commercial use.
GAMS/Gurobi demo/community requires a Gurobi license from Gurobi Optimization.

```


Gurobi library version 9.1.2

MIP status(2): Model was solved to optimality (subject to tolerances).

Fixed MIP status(2): Model was solved to optimality (subject to tolerances).

Final Solve: 6086.000000 (11 iterations)

Best possible: 6085.438144

Absolute gap: 0.561856

Relative gap: 0.000092

---- VARIABLE TFT.L = 6086.000 "tempo total (acumulado) de fluxo"

---- VARIABLE y.L "ordem i precede ordem j no programa de produção"

	0	1	2	3	4	5	6
0							1.000
1	1.000						
2					1.000		
3			1.000				
4						1.000	
5		1.000					
6				1.000			

---- VARIABLE x.L "ordem j entregue apos ordem i pelo veiculo k"

	1
0.6	1.000
1.5	1.000
2.0	1.000
3.2	1.000
4.1	1.000
5.3	1.000
6.4	1.000

---- VARIABLE C.L "tempo de conclusao da ordem j"

1 565.000, 2 424.000, 3 304.000, 4 455.000, 5 505.000, 6 151.000

---- VARIABLE S.L "instante de saída do veículo"

1 565.000

---- VARIABLE F.L "tempo (de fluxo) ate a entrega da ordem j carregada pelo veículo k"

1

1 975.000

2 1304.000

3 1149.000

4 841.000

5 1071.000

6 746.000