



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA QUÍMICA



PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

Otimização da programação de petróleo em uma refinaria com regras
específicas

Bárbara Jahn

Uberlândia

2015



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA QUÍMICA



PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

Otimização da programação de petróleo em uma refinaria com regras
específicas

Bárbara Jahn

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Química da Universidade Federal de Uberlândia como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Química.

Uberlândia

2015

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

J25o
2015 Jahn, Bárbara,
 Otimização da programação de petróleo em uma refinaria com
 regras específicas [recurso eletrônico] / Bárbara Jahn. - 2015.

Orientadora: Valéria Viana Murata.
Coorientador: Sérgio Mauro da Silva Neiro.
Dissertação (mestrado) - Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Engenharia Química.
Modo de acesso: Internet.
Disponível em: <http://dx.doi.org/10.14393/ufu.di.2018.1195>
Inclui bibliografia.
Inclui ilustrações.

1. Engenharia química. 2. Petróleo - Refinaria - Controle de
produção. 3. Planejamento da produção. I. Murata, Valéria Viana
(Orient.). II. Neiro, Sérgio Mauro da Silva (Coorient.). III. Universidade
Federal de Uberlândia. Programa de Pós-Graduação em Engenharia
Química. IV. Título.

CDU: 66.0

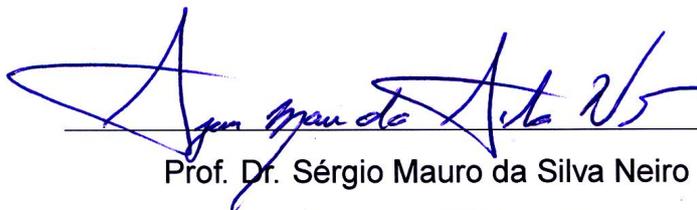
Maria Salete de Freitas Pinheiro - CRB6/1262

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA DA UNIVERSIDADE FEDERAL DE UBERLÂNDIA EM 31 DE JULHO DE 2015 COMO PARTE DOS REQUISITOS PARA OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA QUÍMICA.

BANCA EXAMINADORA:



Profa. Dra. Valéria Viana Murata
Orientadora (PPG-FEQUI/UFU)



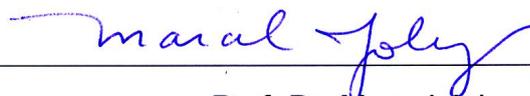
Prof. Dr. Sérgio Mauro da Silva Neiro
Co-orientador (FEQUI/UFU)



Prof. Dr. Kleber Carlos Ribeiro Pinto
Professor convidado (FAGEN/UFU)



Prof. Dr. Humberto Molinar Henrique
Professor convidado (FEQUI/UFU)



Prof. Dr. Marcel Joly
Professor convidado (Escola Politécnica/USP)

*Aos meus avós Célia de Andrade Melo
e Luiz Augusto de Melo.*

AGRADECIMENTOS

Este trabalho é o resultado de uma cooperação entre várias pessoas e instituições. Primeiramente quero agradecer a Deus pela oportunidade de realizá-lo. Agradeço ao Professor Sérgio Neiro por motivar e conduzir os esforços a um lugar seguro no mundo científico. Agradeço à Professora Valéria Murata pela contribuição nas conversas e representação institucional. Agradeço ao CENPES pelo fomento e oportunidade de investigar um problema real. Agradeço ao meu marido, Christoph Jahn, por incessante incentivo e assistência. Por fim, agradeço a todos do Programa de Pós-Graduação em Engenharia Química da Universidade Federal de Uberlândia pelo auxílio durante os dois anos de mestrado.

SUMÁRIO

Sumário	i
Lista de Figuras	v
Lista de Tabelas	xv
Lista de Siglas	xix
Nomenclatura	xxi
Resumo	xxv
Abstract	xxvii
1 Introdução	1
2 Conceitos Preliminares	5
2.1 Programação da produção na indústria de processos químicos	5
2.2 Conceitos de programação matemática	11
2.2.1 Programas inteiro-misto lineares	11
2.2.2 Programas inteiro-misto não-lineares	17
2.3 Formulações da literatura	19
2.4 Classificação dos modelos	26
3 Revisão bibliográfica	33
3.1 Programação de operações em refinarias	33
3.2 Programação do suprimento de petróleo	37
4 Problema real de uma refinaria brasileira	47
4.1 Regras de negócio	54

4.2	Cenários otimizados	55
4.2.1	Cenário um	57
4.2.2	Cenário dois	58
4.2.3	Cenário três	60
4.2.4	Cenário quatro	60
4.2.5	Cenário cinco	62
4.2.6	Cenário seis	63
5	Modelo matemático	71
5.1	Cliques	72
5.2	Modelagem base	81
5.2.1	Variáveis	81
5.2.2	Função objetivo	82
5.2.3	Restrições	82
5.3	Modelagem com descarregamentos não-sobrepostos	87
5.4	Modelagem com descarregamentos sobrepostos	87
5.5	Estratégia de solução	88
5.6	Considerações sobre a precisão do modelo	93
6	Resultados e discussão	95
6.1	Performance	97
6.1.1	Cenário dois	98
6.1.2	Cenário três	102
6.1.3	Cenário quatro	106
6.1.4	Cenário cinco	108
6.1.5	Cenário seis	112
6.2	Melhor programa	114
6.2.1	Cenário dois	117
6.2.2	Cenário três	124
6.2.3	Cenário quatro	130
6.2.4	Cenário cinco	136
6.2.5	Cenário seis	141

7 Conclusões e sugestões	147
Apêndices	151
A Dados de entrada dos cenários	153
A.1 Todos os cenários	153
A.1.1 Lastro e capacidade dos tanques	153
A.1.2 Valor marginal dos petróleos	154
A.1.3 Propriedades dos petróleos	155
A.1.4 Limites de vazão	156
A.1.5 Limites de volume transferido	157
A.2 Cenário um	158
A.2.1 Dados das parcelas	159
A.2.2 Inventário inicial nos tanques de carga	160
A.2.3 Volume útil inicial nos tanques da carga	161
A.3 Cenário dois	162
A.3.1 Dados das parcelas	163
A.3.2 Inventário inicial nos tanques de carga	164
A.3.3 Volume útil inicial nos tanques da carga	165
A.4 Cenário três	166
A.4.1 Dados das parcelas	167
A.4.2 Inventário inicial nos tanques de carga	168
A.4.3 Volume útil inicial nos tanques da carga	169
A.5 Cenário quatro	170
A.5.1 Dados das parcelas	171
A.5.2 Inventário inicial nos tanques de carga	172
A.5.3 Volume útil inicial nos tanques da carga	173
A.6 Cenário cinco	174
A.6.1 Dados das parcelas	175
A.6.2 Inventário inicial nos tanques de carga	176
A.6.3 Volume útil inicial dos tanques da carga	177
A.7 Cenário seis	178
A.7.1 Dados das parcelas	179

A.7.2	Inventário inicial nos tanques de carga	180
A.7.3	Volume útil inicial dos tanques da carga	181
B	Modelos	183
B.1	Modelagem base	183
B.2	Modelagem com descarregamentos não-sobrepostos	186
B.3	Modelagem com descarregamentos sobrepostos	186
C	Resultados da programação	187
C.1	Modelagem com descarregamentos não-sobrepostos	187
C.1.1	Cenário dois	187
C.1.2	Cenário três	192
C.1.3	Cenário quatro	197
C.1.4	Cenário cinco	202
C.1.5	Cenário seis	207
C.2	Modelagem com descarregamentos sobrepostos	212
C.2.1	Cenário dois	212
C.2.2	Cenário três	217
C.2.3	Cenário quatro	222
C.2.4	Cenário cinco	227
C.2.5	Cenário seis	232
D	Resultados computacionais	237
D.1	Modelagem com descarregamentos não-sobrepostos	239
D.2	Modelagem com descarregamentos sobrepostos	240
	Referências Bibliográficas	241

LISTA DE FIGURAS

1.1	Principal diferença entre (b) o modelo com descarregamentos não-sobrepostos (DNS) e (c) o modelo com descarregamentos sobrepostos (DS), supondo a chegada de duas parcelas no tanque AA durante um horizonte de programação de 50 h.	4
2.1	Diagrama com o fluxo da informação em um sistema de produção (adaptado de PINEDO, 2012b).	7
2.2	Representação de um modelo de evacuação através de grafo direcional, um problema de fluxo de rede (<i>network flow problem</i>) (HAMACHER; KLAMROTH, 2006).	10
2.3	Esquema ilustrativo da árvore de busca <i>Branch & Bound</i> (adaptado de POCHET; WOLSEY, 2006a).	15
2.4	Regiões (a) convexas e (b) não-convexas (FLOUDAS, 1995a).	18
2.5	Diferença entre (a) mistura de bateladas ou mistura de dois lotes do mesmo material produzidos pela mesma tarefa, (b) mistura de materiais ou quando uma tarefa requer dois tipos de material, (c) separação de batelada ou quando um lote é dividido e consumido em várias tarefas subsequentes e (d) tarefa que produz materiais diferentes.	21
2.6	Exemplo de processo sequencial multi-estágio (adaptado de PINTO; GROSSMANN, 1995).	22
2.7	Hierarquia da complexidade de problemas de programação da produção com relação ao ambiente máquina (adaptado de PINEDO, 2012a).	22
2.8	Exemplo de uma rede estado-tarefa (adaptado de KONDILI et al., 1993).	23
2.9	Exemplo de uma rede recurso-tarefa (adaptado de MÉNDEZ et al., 2006).	24
2.10	Processo hipotético de produção de levedura.	25

2.11 Critérios de classificação dos modelos de programação da produção segundo Méndez et al. (2006).	26
2.12 Diferença entre a representação de eventos por <i>slots</i> (a) síncronos e (b) assíncronos (adaptado de MOURET et al., 2011).	30
3.1 Divisão acadêmica da refinaria proposta por JIA et al., 2003 para abordar o problema de programação da produção em refinarias (adaptado de JIA et al., 2003).	34
3.2 Produto entre duas variáveis contínuas $f(x, y) = x \cdot y$: (a) gráfico e (b) curvas de nível.	35
3.3 Arranjo ilustrativo do sistema de suprimento de petróleo.	37
3.4 Linha do tempo com alguns trabalhos sobre programação do suprimento de petróleo.	40
4.1 Topologia da refinaria real modelada.	48
4.2 Operações de saída da parcela 3, O_{prc3}	51
4.3 Operações de saída do tanque GG, O_{GG}	52
4.4 Operação de saída do misturador 1, O_{mrx1}	52
4.5 Dados de entrada sobre a programação das parcelas no cenário um com instantes de chegada e final do descarregamento de cada parcela em h.	58
4.6 Valor marginal unitária e número de acidez total (TAN) das misturas em cada recurso para o cenário um.	58
4.7 Dados de entrada sobre a programação das parcelas no cenário dois com instantes de chegada e final do descarregamento de cada parcela em h.	59
4.8 Valor marginal unitário e número de acidez total (TAN) das misturas em cada recurso para o cenário dois.	59
4.9 Dados de entrada sobre a programação das parcelas no cenário três com instantes de chegada e final do descarregamento de cada parcela em h.	60
4.10 Valor marginal unitário e número de acidez total (TAN) das misturas em cada recurso para o cenário três.	61

4.11 Dados de entrada sobre a programação das parcelas no cenário quatro com instantes de chegada e final do descarregamento de cada parcela em h.	61
4.12 Valor marginal unitário e número de acidez total (TAN) das misturas em cada recurso para o cenário quatro.	62
4.13 Dados de entrada sobre a programação das parcelas no cenário cinco com instantes de chegada e final do descarregamento de cada parcela em h.	62
4.14 Valor marginal unitário e número de acidez total (TAN) das misturas em cada recurso para o cenário cinco.	63
4.15 Dados de entrada sobre a programação das parcelas no cenário seis com instantes de chegada e final do descarregamento de cada parcela em h.	64
4.16 Valor marginal unitário e número de acidez total (TAN) das misturas em cada recurso para o cenário seis.	64
4.17 Cenário 1: cinco parcelas de petróleo R_P programadas, oito tanques de carga R_C disponíveis e horizonte de tempo de 168 h.	65
4.18 Cenário 2: três parcelas de petróleo R_P programadas, oito tanques de carga R_C disponíveis e horizonte de tempo de 168 h.	66
4.19 Cenário 3: cinco parcelas de petróleo R_P programadas, nove tanques de carga R_C disponíveis e horizonte de tempo de 240 h.	67
4.20 Cenário 4: três parcelas de petróleo R_P programadas, nove tanques de carga R_C disponíveis e horizonte de tempo de 168 h.	68
4.21 Cenário 5: quatro parcelas de petróleo R_P programadas, nove tanques de carga R_C disponíveis e horizonte de tempo de 144 h.	69
4.22 Cenário 6: duas parcelas de petróleo R_P programadas, nove tanques de carga R_C disponíveis e horizonte de tempo de 240 h.	70
5.1 Clique com descarregamentos não-sobrepostos.	73
5.2 Cliques com descarregamentos sobrepostos.	74

5.3	Solução hipotética para as operações de descarregamento no modelo com descarregamentos não-sobrepostos (DNS) e no modelo com descarregamentos sobrepostos (DS) usando a programação de parcelas do cenário três.	75
5.4	Operações de entrada e saída do tanque de carga AA.	76
5.5	Cliques entre operações de entrada e saída do tanque de carga AA na modelagem com descarregamentos não-sobrepostos.	77
5.6	Cliques entre operações de entrada e saída do tanque de carga AA.	77
5.7	Estratégia de decomposição programação inteiro-misto linear (MILP)-programação não-linear (NLP).	89
5.8	Pool de soluções do <i>solver</i> MILP.	90
6.1	Visão geral do <i>pool</i> de soluções do cenário dois com $n = 6$ na modelagem com descarregamentos (a) não-sobrepostos e (b) sobrepostos.	99
6.2	Programação do descarregamento de parcelas do melhor programa para o cenário dois.	118
6.3	Programação nos misturadores e unidades de destilação atmosférica (UDAs) para o cenário dois com $n = 6$ e descarregamentos não-sobrepostos (DNS).	122
6.4	Programação nos misturadores e UDAs para o cenário dois com $n = 3$ e descarregamentos sobrepostos (DS).	123
6.5	Programação do descarregamento de parcelas do melhor programa para o cenário três.	124
6.6	Vazão de carga nas UDAs no melhor programa encontrado pelo modelo com (a) DNS e (b) DS para o cenário três.	125
6.7	Programação nos misturadores e UDAs para o cenário três com $n = 7$ e DNS.	128
6.8	Programação nos misturadores e UDAs para o cenário três com $n = 6$ e DS.	129
6.9	Programação do descarregamento de parcelas do melhor programa para o cenário quatro.	130

6.10	Programação nos misturadores e UDAs para o cenário quatro com $n = 7$ e DNS.	134
6.11	Programação nos misturadores e UDAs para o cenário quatro com $n = 6$ e DS.	135
6.12	Programação do descarregamento de parcelas do melhor programa para o cenário cinco.	136
6.13	Programação nos misturadores e UDAs para o cenário cinco com $n = 6$ e DNS.	139
6.14	Programação nos misturadores e UDAs para o cenário cinco com $n = 4$ e DS.	140
6.15	Programação do descarregamento de parcelas do melhor programa para o cenário seis.	141
6.16	Programação nos misturadores e UDAs para o cenário seis com $n = 5$ e DNS.	145
6.17	Programação nos misturadores e UDAs para o cenário seis com $n = 5$ e DS.	146
A.1	Cenário 1.	158
A.2	Cenário 2.	162
A.3	Cenário 3.	166
A.4	Cenário 4.	170
A.5	Cenário 5.	174
A.6	Cenário 6.	178
C.1	Perfil de acidez nas operações de destilação W_D no cenário dois com descarregamentos não-sobrepostos.	187
C.2	Gráfico de Gantt para o cenário dois com descarregamentos não-sobrepostos ($n = 6$).	188
C.3	Programação nos misturadores com informações de volume e vazão para o cenário dois com descarregamentos não-sobrepostos ($n = 6$).	189
C.4	Inventário nos tanques de carga (a) AA, (b) BB, (c) CC, (d) DD, (e) FF, (f) HH, (g) II, e (h) LL no cenário dois com descarregamentos não-sobrepostos.	190

C.5	Vazão de saída dos tanques de carga (a) AA, (b) BB, (c) CC, (d) DD, (e) FF, (f) HH, (g) II, e (h) LL no cenário dois com descarregamentos não-sobrepostos.	191
C.6	Perfil de acidez nas operações de destilação W_D no cenário três com descarregamentos não-sobrepostos.	192
C.7	Gráfico de Gantt para o cenário três com descarregamentos não-sobrepostos ($n = 8$).	193
C.8	Programação nos misturadores com informações de volume e vazão para o cenário três com descarregamentos não-sobrepostos ($n = 8$).	194
C.9	Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário três com descarregamentos não-sobrepostos.	195
C.10	Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário três com descarregamentos não-sobrepostos.	196
C.11	Perfil de acidez nas operações de destilação W_D no cenário quatro com descarregamentos não-sobrepostos.	197
C.12	Gráfico de Gantt para o cenário quatro com descarregamentos não-sobrepostos ($n = 7$).	198
C.13	Programação nos misturadores com informações de volume e vazão para o cenário quatro com descarregamentos não-sobrepostos ($n = 7$).	199
C.14	Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário quatro com descarregamentos não-sobrepostos.	200
C.15	Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário quatro com descarregamentos não-sobrepostos.	201
C.16	Perfil de acidez nas operações de destilação W_D no cenário cinco com descarregamentos não-sobrepostos.	202
C.17	Gráfico de Gantt para o cenário cinco com descarregamentos não-sobrepostos ($n = 6$).	203

C.18 Programação nos misturadores com informações de volume e vazão para o cenário cinco com descarregamentos não-sobrepostos ($n = 6$).	204
C.19 Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário cinco com descarregamentos não-sobrepostos.	205
C.20 Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário cinco com descarregamentos não-sobrepostos.	206
C.21 Perfil de acidez nas operações de destilação W_D no cenário seis com descarregamentos não-sobrepostos.	207
C.22 Gráfico de Gantt para o cenário seis com descarregamentos não-sobrepostos ($n = 5$).	208
C.23 Programação nos misturadores com informações de volume e vazão para o cenário seis com descarregamentos não-sobrepostos ($n = 5$).	209
C.24 Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário seis com descarregamentos não-sobrepostos.	210
C.25 Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário seis com descarregamentos não-sobrepostos.	211
C.26 Perfil de acidez nas operações de destilação W_D no cenário dois com descarregamentos sobrepostos.	212
C.27 Gráfico de Gantt para o cenário dois com descarregamentos sobrepostos ($n = 3$).	213
C.28 Programação nos misturadores com informações de volume e vazão para o cenário dois com descarregamentos sobrepostos ($n = 3$).	214
C.29 Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário dois com descarregamentos sobrepostos.	215
C.30 Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário dois com descarregamentos sobrepostos.	216

C.31 Perfil de acidez nas operações de destilação W_D no cenário três com descarregamentos sobrepostos.	217
C.32 Gráfico de Gantt para o cenário três com descarregamentos sobrepostos ($n = 6$).	218
C.33 Programação nos misturadores com informações de volume e vazão para o cenário três com descarregamentos sobrepostos ($n = 6$).	219
C.34 Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário três com descarregamentos sobrepostos.	220
C.35 Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário três com descarregamentos sobrepostos.	221
C.36 Perfil de acidez nas operações de destilação W_D no cenário quatro com descarregamentos sobrepostos.	222
C.37 Gráfico de Gantt para o cenário quatro com descarregamentos sobrepostos ($n = 6$).	223
C.38 Programação nos misturadores com informações de volume e vazão para o cenário quatro com descarregamentos sobrepostos ($n = 6$).	224
C.39 Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário quatro com descarregamentos sobrepostos.	225
C.40 Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário quatro com descarregamentos sobrepostos.	226
C.41 Perfil de acidez nas operações de destilação W_D no cenário cinco com descarregamentos sobrepostos.	227
C.42 Gráfico de Gantt para o cenário cinco com descarregamentos sobrepostos ($n = 4$).	228
C.43 Programação nos misturadores com informações de volume e vazão para o cenário cinco com descarregamentos sobrepostos ($n = 4$).	229

C.44 Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário cinco com descarregamentos sobrepostos.	230
C.45 Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário cinco com descarregamentos sobrepostos.	231
C.46 Perfil de acidez nas operações de destilação W_D no cenário seis com descarregamentos sobrepostos.	232
C.47 Gráfico de Gantt para o cenário seis com descarregamentos sobrepostos ($n = 5$).	233
C.48 Programação nos misturadores com informações de volume e vazão para o cenário seis com descarregamentos sobrepostos ($n = 5$).	234
C.49 Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário seis com descarregamentos sobrepostos.	235
C.50 Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário seis com descarregamentos sobrepostos.	236

LISTA DE TABELAS

3.1	Características dos problemas de programação do suprimento de petróleo estudados na literatura consultada.	41
4.1	Regras operacionais adotadas nos estudos de caso de Lee et al. (1996).	49
4.2	Regras operacionais adotadas na indústria petroquímica brasileira.	50
4.3	Restrições que devem ser respeitadas ao longo de todo o horizonte de programação na refinaria real modelada.	53
4.4	Visão geral dos cenários otimizados.	55
5.1	Número de vértices do grafo de operações não-sobrepostas em cada cenário.	73
5.2	Conjuntos clique $W' \in \text{clique}(G_{NO})$ da modelagem com descarregamentos não-sobrepostos.	78
5.3	Conjuntos clique com tempo de transição $W' \in \text{clique}(G_{NOTR})$ da modelagem com descarregamentos não-sobrepostos.	78
5.4	Conjuntos clique $W' \in \text{clique}(G_{NO})$ da modelagem com descarregamentos sobrepostos.	79
5.5	Conjuntos clique com tempo de transição $W' \in \text{clique}(G_{NOTR})$ da modelagem com descarregamentos sobrepostos.	80
5.6	Parâmetros do <i>pool</i>	90
5.7	Critérios de parada para determinação de $ T $	93
6.1	Resultados computacionais do cenário dois com DNS.	98
6.2	Resultados computacionais do cenário dois com DS.	98
6.3	Comparação entre os modelos em termos de função objetivo para o cenário dois.	100

6.4	Comparação entre os modelos em termos de tempo computacional para o cenário dois.	101
6.5	Resultados computacionais do cenário três com DNS.	102
6.6	Resultados computacionais do cenário três com DS.	102
6.7	Comparação entre os modelos em termos de função objetivo para o cenário três.	104
6.8	Comparação entre os modelos em termos de tempo computacional para o cenário três.	105
6.9	Resultados computacionais do cenário quatro com DNS.	106
6.10	Resultados computacionais do cenário quatro com DS.	107
6.11	Comparação entre os modelos em termos de função objetivo para o cenário quatro.	107
6.12	Comparação entre os modelos em termos de tempo computacional para o cenário quatro.	107
6.13	Resultados computacionais do cenário cinco com DNS.	109
6.14	Resultados computacionais do cenário cinco com DS.	109
6.15	Comparação entre os modelos em termos de função objetivo para o cenário cinco.	110
6.16	Comparação entre os modelos em termos de tempo computacional para o cenário cinco.	110
6.17	Resultados computacionais do cenário seis com DNS.	112
6.18	Resultados computacionais do cenário seis com DS.	112
6.19	Comparação entre os modelos em termos de função objetivo para o cenário seis.	113
6.20	Comparação entre os modelos em termos de tempo computacional para o cenário seis.	113
6.21	Melhor programa para todos os cenários com os modelos com DNS e DS.	114
6.22	Alinhamentos ativos na destilação do modelo com DNS ao longo do horizonte de tempo para o cenário dois.	120
6.23	Alinhamentos ativos na destilação do modelo com DS ao longo do horizonte de tempo para o cenário dois.	121

6.24	Maior salto de número de acidez total (TAN) na alimentação das UDAs para o cenário dois.	121
6.25	Alinhamentos ativos na destilação do modelo com DNS ao longo do horizonte de tempo para o cenário três.	126
6.26	Alinhamentos ativos na destilação do modelo com DS ao longo do horizonte de tempo para o cenário três.	127
6.27	Maior salto de TAN na alimentação das UDAs para o cenário três. . . .	127
6.28	Alinhamentos ativos na destilação do modelo com DNS ao longo do horizonte de tempo para o cenário quatro.	132
6.29	Alinhamentos ativos na destilação do modelo com DS ao longo do horizonte de tempo para o cenário quatro.	133
6.30	Maior salto de TAN na alimentação das UDAs para o cenário quatro. . .	133
6.31	Alinhamentos ativos na destilação do modelo com DNS ao longo do horizonte de tempo para o cenário cinco.	137
6.32	Alinhamentos ativos na destilação do modelo com DS ao longo do horizonte de tempo para o cenário cinco.	138
6.33	Maior salto de TAN na alimentação das UDAs para o cenário cinco. . .	138
6.34	Alinhamentos ativos na destilação do modelo com DNS ao longo do horizonte de tempo para o cenário seis.	143
6.35	Alinhamentos ativos na destilação do modelo com DS ao longo do horizonte de tempo para o cenário seis.	144
6.36	Maior salto de TAN na alimentação das UDAs para o cenário seis. . . .	144
A.1	Lastro e capacidade máxima nos tanques de carga R_C em m^3	153
A.2	Valor marginal dos petróleos C	154
A.3	Propriedades dos petróleos C	155
A.4	Limites de vazão de operações $O_{r \setminus R_{C2}}$ em m^3/h	156
A.5	Limites de vazão na saída dos tanques de carga R_{C2} em m^3/h	156
A.6	Dados das parcelas no cenário um.	159
A.7	Inventário inicial nos tanques de carga R_C em m^3 no cenário um. . . .	160
A.8	Volume útil inicial nos tanques de carga R_C em m^3 no cenário um. . . .	161
A.9	Dados das parcelas no cenário dois.	163

A.10 Inventário inicial nos tanques de carga R_C em m^3 no cenário dois. . . .	164
A.11 Volume útil inicial nos tanques de carga R_C em m^3 no cenário dois. . .	165
A.12 Dados das parcelas no cenário três.	167
A.13 Inventário inicial nos tanques de carga R_C em m^3 no cenário três. . . .	168
A.14 Volume útil inicial nos tanques de carga R_C em m^3 no cenário três. . . .	169
A.15 Dados das parcelas no cenário quatro.	171
A.16 Inventário inicial nos tanques de carga R_C em m^3 no cenário quatro. . .	172
A.17 Volume útil inicial nos tanques de carga R_C em m^3 no cenário quatro. .	173
A.18 Dados das parcelas no cenário cinco.	175
A.19 Inventário inicial nos tanques de carga R_C em m^3 no cenário cinco. . .	176
A.20 Volume útil inicial nos tanques de carga R_C em m^3 no cenário cinco. . .	177
A.21 Dados das parcelas no cenário seis.	179
A.22 Inventário inicial nos tanques de carga R_C em m^3 no cenário seis. . . .	180
A.23 Volume útil inicial nos tanques de carga R_C em m^3 no cenário seis. . . .	181
D.1 Resultados computacionais da modelagem com descarregamentos não-sobrepostos.	239
D.2 Resultados computacionais da modelagem com descarregamentos sobrepostos.	240

LISTA DE SIGLAS

APICS *American Production and Inventory Control Society.*

CPU unidade central de processamento.

DNS descarregamentos não-sobrepostos.

DS descarregamentos sobrepostos.

ERP Planejamento dos Recursos da Empresa.

FO-LS melhor limite superior da função objetivo no programa inteiro-misto linear em 1×10^{-3} \$.

FO-MIP função objetivo do programa inteiro-misto principal em 1×10^{-3} \$.

FO-MIPa função objetivo do programa inteiro-misto associado em 1×10^{-3} \$.

FO-NLP função objetivo do programa não-linear em 1×10^{-3} \$.

GAMS *General Algebraic Modelling System.*

GAP *gap* de otimalidade do programa inteiro-misto linear principal em %.

GAPa *gap* de otimalidade do programa inteiro-misto linear associado em %.

GAPd estimativa do *gap* de otimalidade com estratégia de decomposição MILP-NLP em %.

LR relaxação linear.

MILP programação inteiro-misto linear.

MINLP programação inteiro-misto não-linear.

MOS sequenciamento múltiplo de operações.

MRP I Planejamento das Provisões de Material.

MRP II Planejamento dos Recursos de Produção.

NE número de equações.

NEL número de equações lineares.

NENL número de equações não-lineares.

NLP programação não-linear.

NN-MIP número de nós do programa inteiro-misto linear principal.

NP-hard *non-deterministic polynomial-time hard*.

NSIP número de soluções inviáveis no *pool* de soluções.

NV número de variáveis.

NVB número de variáveis binárias.

NVC número de variáveis contínuas.

NZ-MIP número de elementos não-zero na matriz de coeficientes do modelo MILP.

NZ-NL número de elementos não-lineares não-zero na matriz de coeficientes do modelo NLP.

PSE Engenharia de Sistemas de Processo.

REVAP Refinaria Henrique Lage ou Refinaria do Vale do Paraíba.

RTN rede recurso-tarefa.

SM status do modelo MILP principal.

SOS sequenciamento único de operações.

STN rede estado-tarefa.

t-AC tempo acumulado necessário para determinar o número de *slots* de prioridade que gera o melhor programa em minutos.

TAN número de acidez total.

t-MIP tempo computacional do programa inteiro-misto linear em minutos.

t-NLP tempo computacional do programa não-linear linear em minutos.

t-TOTAL tempo computacional total em minutos.

UDA unidade de destilação atmosférica.

UIS armazenamento ilimitado de intermediários.

NOMENCLATURA

Conjuntos

C Conjunto de crus.

clique(G_{NO}) Conjunto dos conjuntos clique que não envolvem tempo de preparação $W' \in \text{clique}(G_{NO})$.

clique(G_{NOTR}) Conjunto dos conjuntos clique que envolvem tempo de preparação $W' \in \text{clique}(G_{NOTR})$.

E Conjunto de arestas ou arcos de um grafo.

I_r Conjunto de operações de entrada do recurso r ($W \times R \supset I_r$).

K_m Conjunto propriedades chave dos crus em base mássica.

K_v Conjunto propriedades chave dos crus em base volumétrica.

O_r Conjunto de operações de saída do recurso r ($W \times R \supset O_r$).

R Conjunto de recursos.

R_C Conjunto de tanques de carga ($R \supset R_C$).

R_{C2} Conjunto de tanques de carga com duas possíveis correntes de saída ($R \supset R_{C2}$).

R_D Conjunto de unidades de destilação ($R \supset R_D$).

R_M Conjunto de misturadores.

R_P Conjunto de parcelas que chegam por oleoduto ($R \supset R_P$).

R_S Conjunto de tanques de armazenamento ($R \supset R_S$).

R_V Conjunto de navios ($R \supset R_V$).

T Conjunto de *slots* de prioridade.

V Conjunto de vértices de um grafo.

W Conjunto de operações.

W_D Conjunto de operações de carga das unidades de destilação ($W \supset W_D$).

W_M Conjunto de operações de mistura ($W \supset W_M$).

W_U Conjunto de operações de descarga dos navios ou parcelas ($W \supset W_U$).

Parâmetros

G_c Valor marginal unitário do petróleo c .

\dot{G}_{iv} Valor marginal unitário relativo da mistura transferida na operação $v \in W$ no *slot* de prioridade $i \in T$.

\overline{G}_0 Valor marginal médio no começo do horizonte de tempo.

H Horizonte de tempo da programação.

L_{0r}^t Volume total no recurso r no começo do horizonte de programação.

$\overline{\text{TAN}}_0$ Número de acidez total médio no começo do horizonte de tempo.

$TR_{v_1v_2}$ Tempo de preparação entre as operações $v_1 \in W'$ e $v_2 \in W'$.

x_{ck} Valor da propriedade chave k no petróleo c .

Variáveis

D_{iv} Duração da operação $v \in W$ no *slot* de prioridade $i \in T$.

E_{iv} Tempo final da operação $v \in W$ no *slot* de prioridade $i \in T$.

λ_{ir} Duração do *slot* de tempo $i \in T$ no tanque de carga $r \in R_C$.

L_{irc} Volume de petróleo $c \in C$ acumulado no recurso $r \in R$ antes da operação atribuída ao *slot* de prioridade $i \in T$.

L_{ir}^t Volume total acumulado no recurso $r \in R$ antes da operação atribuída ao *slot* de prioridade $i \in T$.

S_{iv} Tempo inicial da operação $v \in W$ no *slot* de prioridade $i \in T$.

V_{ivc} Volume de cru $c \in C$ transferido na operação $v \in W$ no *slot* de prioridade $i \in T$.

V_{iv}^t Volume total transferido na operação $v \in W$ no *slot* de prioridade $i \in T$.

$wv(i, n)$ Variável binária de alocação da tarefa $i \in I$ ao ponto de evento $n \in N$.

$X_{ii'}$ Variável binária de sequenciamento da tarefa $i \in I$ e $i' \in I$.

$y(i, j, n)$ Variável binária de alocação da tarefa $i \in I$ à unidade $j \in J$ no ponto de evento $n \in N$.

$yv(j, n)$ Variável binária de alocação da unidade $j \in J$ ao ponto de evento $n \in N$.

Z_{iv} Variável binária de atribuição da operação $v \in W$ ao *slot* de prioridade $i \in T$.

Letras gregas

α Ambiente máquina.

β Características de processamento.

γ Objetivo da programação.

ρ Densidade em g cm^{-3} .

Glossário de termos técnicos

gap de otimalidade Desvio relativo máximo entre a solução ótima e a solução factível retornada pelo algoritmo *Branch & Bound* quando a árvore de busca do programa inteiro é truncada (POCHET; WOLSEY, [2006a](#)).

RESUMO

Empresas do ramo de processamento de petróleo, assim como a maioria dos demais empreendimentos, dependem cada vez mais de ferramentas de otimização. O objetivo principal desse trabalho é investigar a viabilidade do uso de técnicas de otimização aplicadas à modelos para programação do suprimento de petróleo em uma refinaria real com regras operacionais específicas da indústria brasileira. A regra operacional que caracteriza o sistema estudado é que uma **unidade de destilação atmosférica (UDA)** pode ser alimentada por até dois tanques de carga simultaneamente. Uma das formulações de tempo contínuo mais promissoras da literatura, a formulação **sequenciamento múltiplo de operações (MOS)** de Mouret et al. (2011), foi selecionada para modelar o sistema produtivo da refinaria: um sistema multi-estágio constituído por até cinco parcelas de petróleo, nove tanques de carga, três **unidades de destilação atmosférica (UDAs)** e trinta e seis tipos de petróleo. A função objetivo do problema de otimização foi maximizar a margem de contribuição das misturas de petróleo destiladas. Obteve-se então um problema de **programação inteiro-misto não-linear (MINLP)**, que foi solucionado com estratégia de decomposição **MILP-NLP** usando algoritmos implementados em *solvers* comerciais. Dois modelos foram desenvolvidos a partir da formulação de Mouret et al. (2011): um modelo com operações de descarregamento não-sobrepostas e outro com operações de descarregamento sobrepostas. Ambos modelos foram implementados em *General Algebraic Modelling System (GAMS)* e aplicados a seis cenários diferentes no sistema de interesse. O horizonte de tempo dos cenários otimizados variou entre 6 e 10 dias. Os modelos foram bem-sucedidos em representar o sistema produtivo de uma refinaria real e suas regras operacionais, e em encontrar soluções viáveis para o problema da programação do suprimento de petróleo em tempos computacionais praticáveis. A corrida correspondente ao melhor programa dos cenários otimizados foi solucionada pelo modelo com descarregamentos não-sobrepostos em até 28,32 min, e pelo modelo com descarregamentos sobrepostos em até 13,94 min.

Palavras-chave: *programação da produção, programação do abastecimento de petróleo, programação misto-inteiro linear*

ABSTRACT

Process-oriented oil and chemical companies, like most industries, are becoming increasingly dependent on optimization tools. The main objective of this work is to investigate the viability of using optimization techniques applied to crude supply scheduling models in a real oil refining plant with specific operating rules of the Brazilian industry. The system representative operating rule is that up to two charging tanks can feed one crude distillation unit simultaneously. One of the most promising continuous-time formulations in the literature, the multi-operation sequencing (MOS) formulation from Mouret et al. (2011), was selected to model the oil refining plant production system, a multi-stage system consisting in up to five crude parcels, nine charging tanks, three crude distillation units (CDU) and thirty six types of crude oil. The optimization problem objective function was to maximize the distilled crude mixtures gross profit margin. A mixed-integer non-linear programming (MINLP) problem was then obtained, which was solved with MILP-NLP decomposition strategy by algorithms implemented in commercial solvers. Two models were developed with the MOS formulation: one model with non-overlapping unloading operations and the other model with overlapping unloading operations. Both models were implemented in the *General Algebraic Modelling System* (GAMS) and applied to six different scenarios in the oil refining system. The scenarios time horizon varied from 6 to 10 days. The models represented the oil refining system and its operating rules successfully, as well as were able to find feasible solutions to the crude oil scheduling problem (COSP) in feasible computational times. The model with non-overlapping unloading operations was able to solve the optimized scenarios in up to 28,32 minutes, and the model with overlapping unloading operations in up to 13,94 minutes.

Keywords: *batch scheduling, crude oil scheduling, mixed-integer linear programming*

INTRODUÇÃO

Na indústria de processos químicos a programação da produção aparece em várias aplicações: da indústria petroquímica ao setor farmacêutico e de químicos especiais. A maioria dos trabalhos recentes na literatura de [Engenharia de Sistemas de Processo \(PSE\)](#) – que surgiram no fim da década de 1970 e começo de 1980 – consideravam processos em batelada de maneira similar aos problemas de manufatura discreta. Especificamente, os trabalhos consideravam problemas nos quais bateladas precisam ser processadas em estágios consecutivos (sem mistura e separação) da mesma maneira que tarefas discretas passando por estágios ou operações.

O escopo da programação de processos químicos foi estendido no começo da década de 1990 para incluir plantas de estrutura arbitrária, restrições de recursos e várias outras restrições de processo. Uma variedade de modelos e métodos de solução foram propostos desde então para lidar com uma gama cada vez maior de problemas e aplicações. Como resultado, a programação da produção na indústria química se tornou uma subárea de pesquisa importante de operações de processo (MARAVELIAS, [2012](#)).

No contexto da *Process Systems Engineering* ou [PSE](#) o problema da programação da produção é colocado como um problema de otimização. Primeiramente uma formulação adequada ao sistema é selecionada. Depois, elabora-se o modelo matemático de forma a representar esse sistema, o que tipicamente leva a um problema de [programação inteiro-misto linear \(MILP\)](#) ou [programação inteiro-misto não-linear \(MINLP\)](#). Estes, por sua vez, são solucionados com técnicas de pesquisa operacional, também chamadas por Méndez et al. ([2006](#)) de métodos exatos.

Empresas do ramo de processamento de petróleo, assim como a maioria dos demais empreendimentos, dependem cada vez mais de ferramentas de otimização. No começo da década de 1970 desenvolvimentos comerciais em tecnologia e pes-

quisa operacional incentivaram empresas pela primeira vez a monitorar de perto sua performance e terem maior visibilidade operacional através da coleta de dados. No começo da década de 1980 empresas do ramo químico e petroquímico começaram a focar em performance operacional. Na década de 1990 houve uma conscientização de que há um limite quando se foca apenas em melhorar a eficiência, e a meta passou a ser produtividade (LASSCHUIT; THIJSSSEN, 2004).

O progresso da capacidade de planejamento nos últimos 10 a 15 anos foi impulsionado fundamentalmente pela habilidade de traduzir objetivos estratégicos e operacionais em planos e programas com auxílio de computadores. Por esse motivo, ferramentas de apoio à decisão e de integração entre departamentos auxiliadas por computadores são uma funcionalidade crítica no mundo de hoje.

Nesse sentido a descontinuidade do fluxo de informação é o maior obstáculo. Atrasos nas decisões e comunicados através da cadeia de abastecimento por causa de planilhas desatualizadas, índices de performance tendenciosos, barreiras funcionais entre departamentos e falta de transparência através do empreendimento deixam decisões do dia-a-dia lentas e ineptas.

No atual cenário mundial, ferramentas computacionais de apoio à decisão projetadas para guiar o empreendimento nos níveis econômico e operacional devem levar em conta informações disponíveis que reflitam os estados reais do processo e do mercado: sem dados precisos não é possível progredir (LASSCHUIT; THIJSSSEN, 2004).

Artiba e Elmaghraby (1997) salientam que há uma lacuna entre teoria e prática na área de programação da produção. Barancsi et al. (1983) apud Artiba e Elmaghraby (1997) identificaram cinco causas para esse fenômeno:

1. a estrutura organizacional das empresas não autoriza a criação de modelos abrangentes;
2. os responsáveis pelas tomadas de decisão são céticos em relação à modelos teóricos e aversos à sua aplicação;
3. responsáveis por projetos de sistemas de processo não têm conhecimento sobre modelos disponíveis e sobre como eles são aplicados;
4. o suporte computacional existente é orientado à atividades em banco de dados;

5. vários modelos foram criados com uma perspectiva puramente matemática do problema, sem levar em consideração as possibilidades de aplicação.

Culturas estabelecidas e relações de poder também contribuem para essa lacuna entre teoria e prática, e estão relacionadas ao fator humano.

O objetivo principal desse trabalho é investigar a viabilidade do uso de técnicas de otimização aplicadas à modelos para programação do suprimento de petróleo em uma refinaria real com regras operacionais específicas da indústria brasileira. A regra operacional que caracteriza o sistema estudado é que uma **unidade de destilação atmosférica (UDA)** pode ser alimentada por até dois tanques de carga simultaneamente, e um tanque de carga pode alimentar até duas **UDAs** simultaneamente.

Uma das formulações mais promissoras da literatura, a formulação **sequenciamento múltiplo de operações (MOS)** de Mouret et al. (2011), foi selecionada para modelar o sistema produtivo da refinaria, um sistema multi-estágio constituído por parcelas de petróleo, tanques de carga e **UDAs**. Obteve-se então um problema de **programação inteiro-misto não-linear (MINLP)**, que foi resolvido com técnicas de programação matemática implementadas em *solvers* comerciais.

Dois modelos foram desenvolvidos a partir da formulação de Mouret et al. (2011): um modelo com operações de descarregamento não-sobrepostas (DNS) e outro com operações de descarregamento sobrepostas (DS). A principal diferença entre os dois modelos é a abordagem usada para construir os conjuntos clique da formulação de Mouret et al. (2011).

A Figura 1.1 ilustra a principal diferença entre os dois modelos. No modelo com descarregamentos não-sobrepostos (DNS) todas as operações de descarregamento de parcelas devem acontecer em *slots* diferentes, como mostra a Figura 1.1 (b). Já no modelo com descarregamentos sobrepostos (DS) as operações de descarregamento de parcelas podem acontecer no mesmo *slot* desde que sejam de parcelas diferentes, como mostra a Figura 1.1 (c). Restrições foram incluídas no modelo com descarregamentos sobrepostos para assegurar a consistência dos balanço de massa nos tanques.

Cada modelo foi aplicado a seis cenários diferentes no sistema de interesse. Os cenários testados diferem entre si quanto ao horizontes de tempo, quantidade de

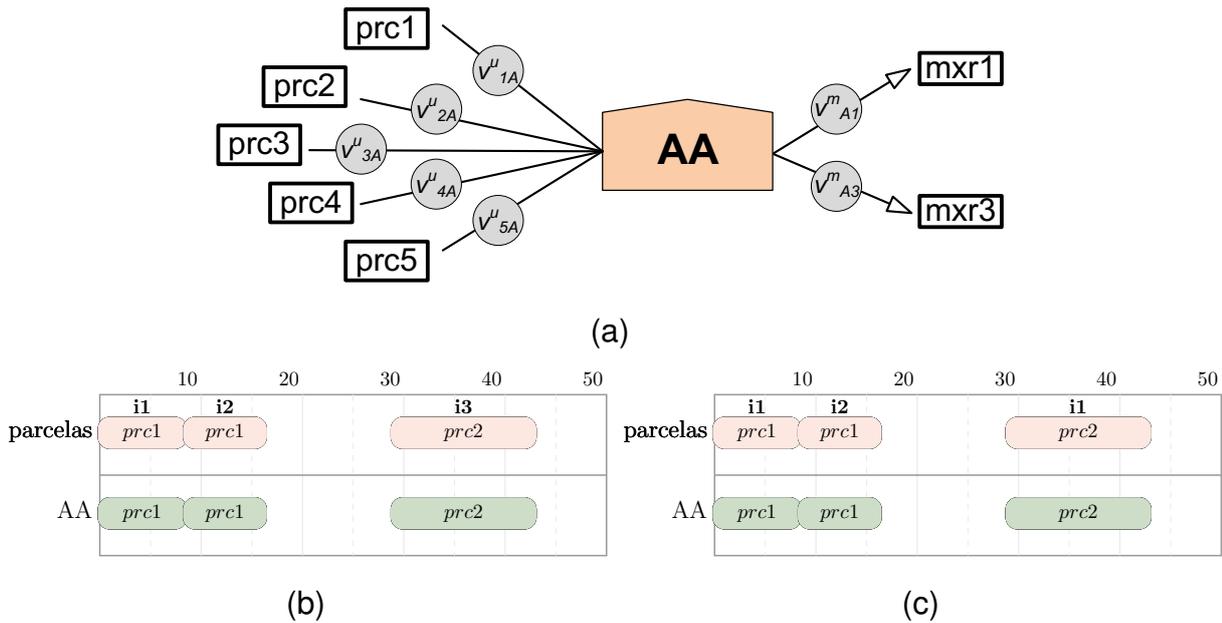


Figura 1.1: Principal diferença entre (b) o modelo com descarregamentos não-sobrepostos (DNS) e (c) o modelo com descarregamentos sobrepostos (DS), suponto a chegada de duas parcelas no tanque AA durante um horizonte de programação de 50 h.

petróleos no sistema, quantidade de parcelas de petróleo que chegam ao longo do horizonte de tempo, quantidade de tanques de carga disponíveis para descarregamento das parcelas e inventário inicial nos tanques de carga.

Os objetivos específicos desse trabalho são:

- solucionar os dois modelos com estratégia de decomposição MILP-NLP;
- comparar a qualidade da solução retornada pelos dois modelos, tanto no que diz respeito a função objetivo quanto aos resultados da programação;
- comparar o tempo computacional necessário para solucionar os dois modelos elaborados;
- investigar a adequabilidade da formulação MOS de Mouret et al. (2011) ao sistema de interesse.

Os modelos foram implementados e solucionados em *General Algebraic Modelling System (GAMS)* (BROOKE et al., 1998).

CONCEITOS PRELIMINARES

Nesse capítulo o problema do planejamento e programação da produção é apresentado e discutido. As diferenças e semelhanças entre indústria de processos químicos (refinarias de petróleo, por exemplo) e indústria discreta (indústria automobilística, por exemplo) são esclarecidas, na medida em que certos problemas de planejamento e programação na indústria de processos químicos podem ser resolvidos pelos mesmos métodos usados para resolver os da indústria discreta. Posteriormente a discussão é voltada para os métodos de solução usados nessa dissertação, em uma breve revisão sobre programação matemática e otimização. Neste ponto são apresentados conceitos de [programação inteiro-misto linear \(MILP\)](#) e [programação inteiro-misto não-linear \(MINLP\)](#), e enumerados alguns algoritmos de solução.

Programação da produção na indústria de processos químicos

Em consequência do aumento da competitividade global e valorização da informação, as organizações industriais passaram a administrar seus sistemas de produção com estratégias integradas às metas globais, visando reduzir custos, aumentar a flexibilidade e minimizar a variabilidade dos níveis de qualidade. Este conceito é conhecido como estratégia de produção (HEIDRICH, [2005](#)).

A programação como um processo de tomada de decisão é de intrínseca importância na maioria dos sistemas produtivos. De acordo com Harjunkoski et al. ([2014](#)) a programação da produção no contexto da [PSE](#) é apenas um componente do gerenciamento da cadeia de suprimentos, e uma entre as várias funções do planejamento.

Segundo Pochet e Wolsey ([2006b](#)) o planejamento da cadeia de suprimentos

(*supply chain planning*) é similar ao planejamento da produção (*production planning*), mas com o escopo estendido à decisões de fornecimento e distribuição.

Pinedo (2012a) define programação ou *scheduling* como um processo de tomada de decisão que lida com alocação de recursos à tarefas durante um período de tempo a fim de otimizar um ou mais objetivos. Recursos e tarefas em uma organização podem ser de diferentes naturezas. Os objetivos são também de vários tipos.

A função da programação deve interagir com várias outras funções (PINEDO, 2012b; HARJUNKOSKI et al., 2014; MARAVELIAS, 2012). Essas interações geralmente acontecem dentro de um sistema de informação que abrange toda a empresa, o **Planejamento dos Recursos da Empresa (ERP)**. Há empresas especializadas no desenvolvimento desse tipo de sistema. O **ERP** conecta os sistemas de apoio à decisão: planejamento, programação e controle.

A Figura 2.1 mostra o fluxo de informação em um sistema de produção. O planejamento fornece um plano mestre de produção com pedidos, datas de entrega, lista de materiais necessários e plano de distribuição. Os horizontes de tempo considerados no nível de planejamento são maiores que os da programação. Segundo Maravelias (2012), a programação fica encarregada das decisões do número de tarefas a ser executado e seus respectivos tamanhos (*batching e lot-sizing*), decisões de atribuição de tarefas à recursos (*assignment*) e decisões de sequenciamento e cronometragem das tarefas (*sequencing e timing*). Na base da cadeia de informação fica o gerenciamento e controle do chão de fábrica, onde todas as medidas do processo e sua performance são determinadas.

De acordo com Lasschuit e Thijssen (2004) gerenciar a cadeia de suprimentos proporciona uma integração horizontal, ao sistematizar o comportamento produtivo através dos departamentos em um empreendimento, e também integração vertical, pois conecta as camadas estratégicas de coordenação: planejamento, programação e operações.

Vários métodos, ferramentas e técnicas surgiram entre as décadas 1970 e 1990 para racionalizar e gerenciar a produção. Um sistema bastante difundido é o **Planejamento das Provisões de Material (MRP I)**, que se popularizou em meados de 1970. O **MRP I** é uma ferramenta de suporte à decisão para acompanhamento dos níveis de inventário de cada item da lista de materiais. Há vários pacotes comerciais

2.1. PROGRAMAÇÃO DA PRODUÇÃO NA INDÚSTRIA DE PROCESSOS QUÍMICOS⁷

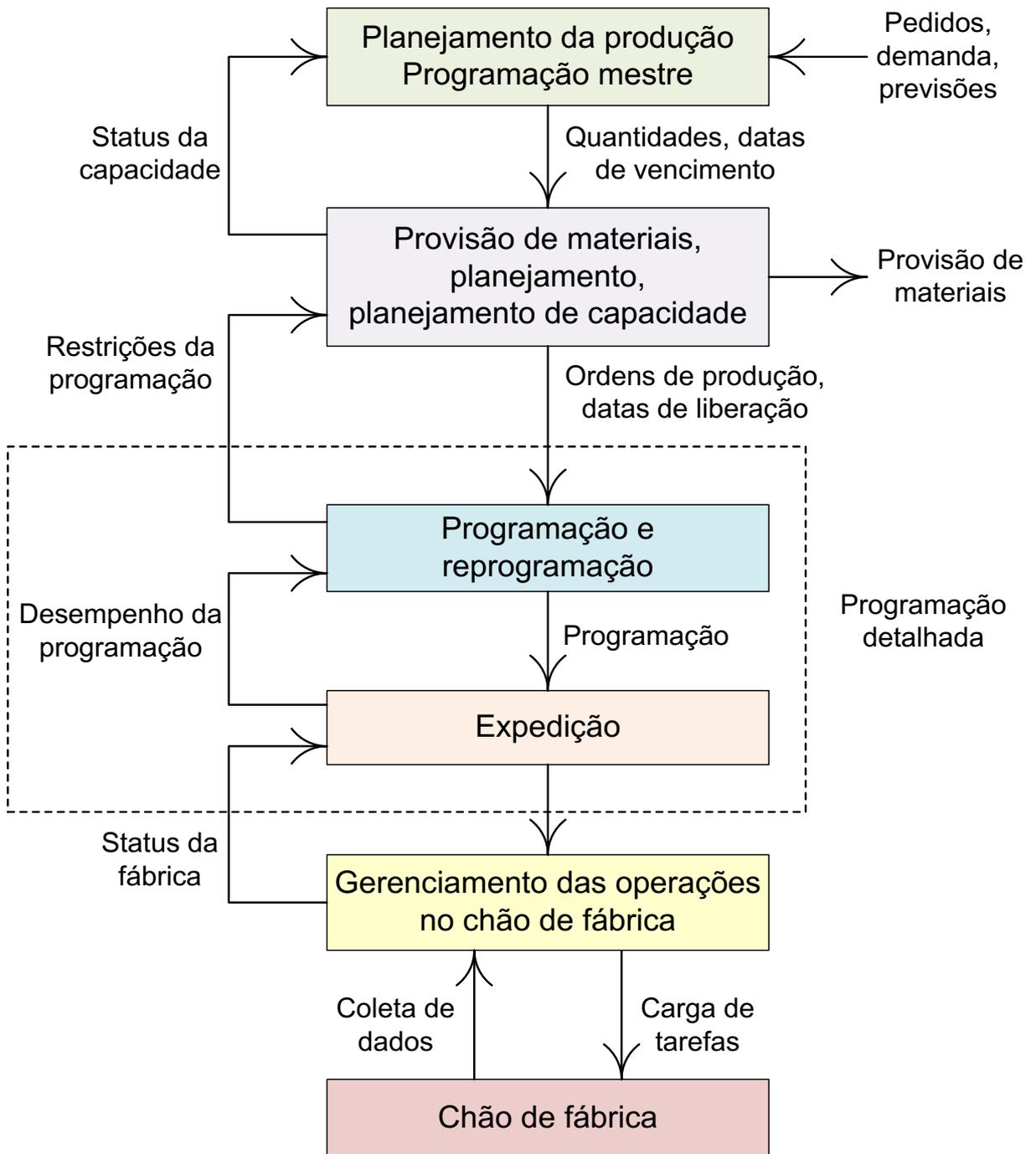


Figura 2.1: Diagrama com o fluxo da informação em um sistema de produção (adaptado de PINEDO, 2012b).

MRP I. As falhas do **MRP I** levaram ao desenvolvimento do **Planejamento dos Recursos de Produção (MRP II)**, um método que vai além do acompanhamento dos níveis de inventário e abrange o planejamento de todos os recursos de manufatura (ARTIBA; ELMAGHRABY, 1997).

A fim de alocar os recursos disponíveis e atingir o objetivo, pedidos devem ser transformados em tarefas com datas de entrega associadas. Essas tarefas devem ser processadas nos recursos disponíveis obedecendo uma determinada sequência ou receita. Eventos como manutenção, tempos de preparação de unidades, tempos de processamento, entre outros, devem ser levados em consideração para que um programa eficiente seja elaborado (PINEDO, 2012b).

O dicionário da *American Production and Inventory Control Society (APICS)* (BLACKSTONE, 2013) define indústria de processos químicos como aquela que emprega um processo de produção que agrega valor por mistura, separação ou formação de materiais. Alguns exemplos de indústrias de processos são indústrias química, petroquímica, de papel, alimentícia e bebida. Elas podem operar continuamente ou em batelada.

Existe uma distinção clara entre processos contínuos e em batelada, não apenas acerca do modo de operação, mas também do nível de especialização do sistema produtivo. Plantas contínuas produzem grandes volumes de poucas variedades de produtos com equipamentos especializados, que permitem fluxo contínuo de material. Elas geralmente operam em modo campanha: todos os recursos da planta são dedicados à produção de alguns produtos, principalmente quando a previsão de demanda a longo prazo é conhecida.

Em contraste com processos contínuos, nos processos em batelada pequenos volumes de vários tipos de produtos são produzidos com equipamentos multipropósito, que não são tão especializados quanto unidades contínuas. Nos processos em batelada uma quantidade fixa de material é processada por vez em um recurso, ou seja, não há fluxo contínuo nos equipamentos. Processos não-contínuos – semi-contínuos ou semi-batelada – combinam aspectos de ambos.

A principal vantagem de processos não-contínuos é flexibilidade. Quanto mais flexível o processo mais fácil é a prática de um controle de qualidade rígido e variações de um mesmo produto podem ser feitas em pouco tempo (BEBER; BECKER,

2.1. PROGRAMAÇÃO DA PRODUÇÃO NA INDÚSTRIA DE PROCESSOS QUÍMICOS⁹

2014). Produtos que precisam passar por transformações similares ou procedimentos complexos de síntese – por exemplo produtos químicos especiais como farmacêuticos, cosméticos, eletrônicos (*wafers*), produtos alimentícios, etc. – podem ser produzidos em janelas de tempo relativamente pequenas e de acordo com as flutuações de demanda, pois uma unidade é adequada para processar vários tipos de produto (GANESH, 2004). Essa característica reivindica que a produção seja planejada.

Em contrapartida à indústria de processos químicos, há também definida no dicionário da APICS a indústria discreta, do tipo linha de montagem. A principal diferença entre elas é que a indústria de processos químicos lida com produtos não-discretos (fluidos) e a indústria discreta com produtos discretos (peças). A indústria automobilística é um exemplo de indústria discreta. Grande parte das matérias-primas, intermediários e produtos na indústria de processos químicos são fluidos e podem se misturar, se separar ou se transformar via reação química (MARAVELIAS, 2012).

Isso agrega complexidade ao problema da programação na indústria de processos químicos e é precisamente a principal diferença entre o problema da programação na indústria de processos químicos e discreta. Fluxos podem se misturar formando correntes multi-componente e materiais podem se transformar em outros via reação química, por exemplo.

Um sistema de produção pode ser representado por um grafo direcional $G = (V, E)$, composto por um conjunto de nós ou vértices V e um conjunto de arestas ou arcos E , que conectam os vértices (BEBER; BECKER, 2014). Cada arco é um par ordenado de vértices (HAMACHER; KLAMROTH, 2006). A Figura 2.2 ilustra um exemplo de grafo direcional.

A Figura 2.2 exibe a planta de um prédio a ser evacuado e o grafo que representa o fluxo de pessoas nos cômodos do prédio. Os cômodos 1, 2, 3 e 4 são conectados por portas e assume-se que só é possível passar pelas portas em uma direção. Deseja-se saber qual é a forma mais rápida de evacuar o prédio para uma pessoa que se encontra no cômodo 1. Para responder essa pergunta, a planta do prédio é modelada como um grafo direcional. Os cômodos são vértices do grafo e os arcos representam a possibilidade de movimento entre os cômodos. Para se mover do cômodo i para o cômodo j , c_{ij} unidades de tempo são necessárias. Em cada cômodo i há b_i pessoas e no máximo u_{ij} pessoas podem se mover entre os cômodos i e j por

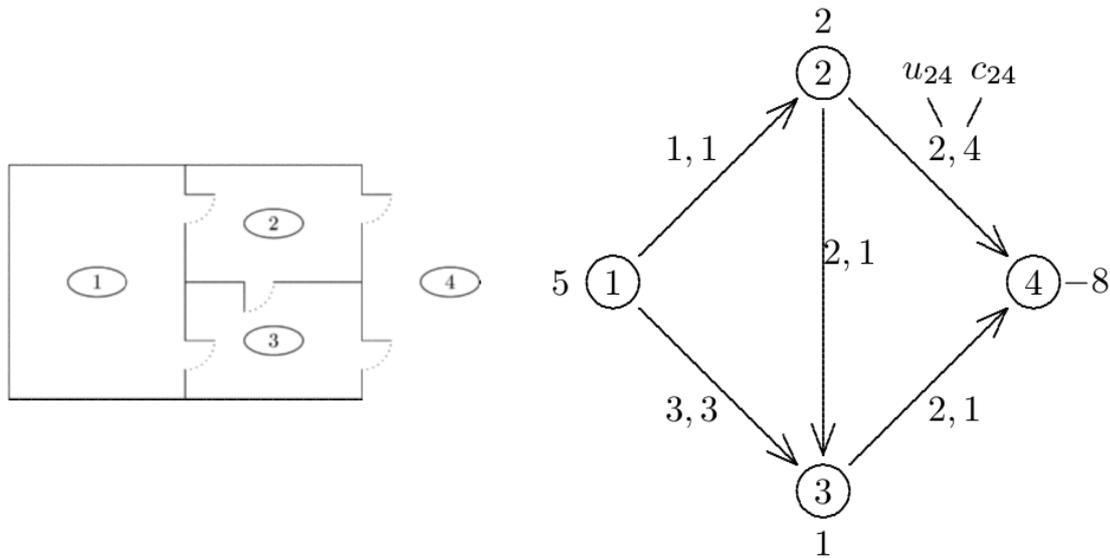


Figura 2.2: Representação de um modelo de evacuação através de grafo direcional, um problema de fluxo de rede (*network flow problem*) (HAMACHER; KLAMROTH, 2006).

unidade de tempo. Esse tipo de problema é chamado na literatura de problema do caminho mais curto (*shortest path*) ou problema de fluxo de rede (*network flow*).

Vários métodos podem ser usados para resolver o problema de planejamento e programação da produção na indústria de processos químicos (MÉNDEZ et al., 2006). Para sistemas pequenos é possível encontrar boas soluções com simples heurística. Brucker e Knust (2012) apresentam algoritmos para solucionar problemas do caminho mais curto (*shortest path*), de fluxo máximo e custo mínimo. Méndez et al. (2006) relata que uma das formas usadas para resolver esses problemas são os métodos exatos, entre os quais se destacam aqueles baseados no princípio *Branch & Bound*.

Segundo Pinto (2003), as principais áreas que descrevem teoria e conceitos de otimização aplicada são a programação matemática e a pesquisa operacional. A programação matemática lida com a caracterização de propriedades teóricas de problemas e algoritmos de otimização, enquanto a pesquisa operacional foca na aplicação e implementação de métodos de otimização para uso prático. Em PSE foca-se na aplicação de métodos de otimização à problemas reais da indústria de processos químicos.

Conceitos de programação matemática

Programas inteiro-misto lineares

De acordo com Pochet e Wolsey (2006a) um programa inteiro-misto linear **MILP** é um programa de otimização que envolve variáveis contínuas, variáveis inteiras e funções lineares. Qualquer **MILP** pode ser escrito na forma das equações 2.1 e 2.2, onde o conjunto X é o conjunto de soluções factíveis ou a região factível, descrita por m restrições lineares, restrições de não-negatividade nas variáveis x, y e restrições de integralidade nas variáveis y .

$$Z(X) = \max_{(x,y)} \{cx + fy \mid (x, y) \in X\} \quad (2.1)$$

$$X = \{(x, y) \in \mathbb{R}_+^n \times \{0, 1\}^p \mid Ax + By \geq b\} \quad (2.2)$$

sendo

- $Z(X)$ é o valor ótimo da função objetivo quando a otimização é efetuada sobre o conjunto factível X ;
- x e y são respectivamente o vetor coluna de dimensão n das variáveis contínuas não-negativas e o vetor coluna de dimensão p de variáveis inteiras não negativas ou variáveis binárias;
- $c \in \mathbb{R}^n$ e $f \in \mathbb{R}^p$ são os vetores linha de coeficientes da função objetivo;
- $b \in \mathbb{R}^m$ é o vetor coluna dos coeficientes no lado direito das m restrições;
- A e B são as matrizes $(m \times n)$ e $(m \times p)$, respectivamente, de coeficientes reais das restrições.

Programas inteiro-misto lineares são resolvidos com uma combinação dos algoritmos *Branch & Bound*, *Cutting Planes*, heurística e técnicas de pré processamento (ACHTERBERG; WUNDERLING, 2013). De acordo com Hillier e Lieberman (2001) essa combinação é chamada de *Branch & Cut*. Para solucionar o **MILP** o algoritmo *Branch & Bound* resolve uma sequência de programas lineares. Programas lineares

podem ser resolvidos pelo algoritmo simplex. A Figura 2.3 mostra um esquema ilustrativo da árvore de busca construída pelo algoritmo *Branch & Bound*. Achterberg e Wunderling (2013) argumentam que *solvers MILP* de última geração são determinísticos: é garantido que ao rodar o mesmo *solver* com os mesmos dados de entrada no mesmo *hardware*, o algoritmo promoverá o mesmo caminho de solução e retornará o mesmo resultado.

Algoritmo *Branch & Bound*

Relaxação linear A *relaxação linear (LR)* de um *MILP* desempenha um papel crucial no algoritmo de solução do *MILP* (POCHET; WOLSEY, 2006a). O primeiro programa linear solucionado pelo algoritmo *Branch & Bound* é a *LR*. A *LR* é obtida quando se elimina as restrições de integralidade das variáveis y . Em outras palavras, y pode assumir qualquer valor real entre 0 e 1. A *LR* de um programa *MILP* pode ser escrita pelas equações 2.3 e 2.4, onde o conjunto P_X é o conjunto de soluções factíveis da *LR*, que é um programa linear.

$$Z(P_X) = \max_{(x,y)} \{cx + fy : (x,y) \in P_X\} \quad (2.3)$$

$$P_X = \{(x,y) \in \mathbb{R}_+^n \times [0,1]^p : Ax + By \geq b\} \quad (2.4)$$

Observa-se que a região factível X do programa *MILP* é o conjunto de pontos de P_X nos quais y é inteiro, $X = P_X \cap (\mathbb{R}^n \times \{0,1\}^p)$.

Limites inferior e superior Os limites inferior e superior no valor da função objetivo são parâmetros importantes para o algoritmo *Branch & Bound*. No caso de um problema de maximização, a função objetivo da *LR* define o limite superior no valor ótimo da função objetivo, $Z(X) \leq Z(P_X)$.

Ainda para problemas de maximização, o valor da função objetivo $Z = cx + fy$ alcançado por qualquer solução factível \bar{Z} do *MILP* $(x,y) \in X$ fornece um limite inferior para o valor ótimo, $Z(X) \geq \bar{Z}$. Portanto,

$$\bar{Z} \leq Z(X) \leq Z(P_X). \quad (2.5)$$

O limite superior (maximização) inicial do *MILP* em $Z(X)$ é fornecido pelo valor ótimo $Z(P_X)$ da relaxação linear *LR* (Figura 2.3), que é um programa fácil de resolver

tanto em termos de complexidade quanto na prática, por ser um programa linear. Seja (x^*, y^*) uma solução ótima da LR.

Solucionar o problema MILP significa encontrar (x, y) que forneça o melhor valor de função objetivo no conjunto P_X com $y \in \{0, 1\}$. Para alcançar esse objetivo uma sequencia de programas lineares é solucionada.

Se a solução da LR for factível também para o MILP, i.e. $y^* \in \{0, 1\}$, além de proporcionar um limite inferior de $Z(X)$, ela é também o limite superior de $Z(X)$. Como consequência, a solução da LR é a solução ótima do MILP porque os limites superior e inferior da função objetivo são iguais: $cx^* + fy^* = Z(P_X) \leq Z(X) \leq \bar{Z} = cx^* + fy^*$, o que implica $Z(X) = cx^* + fy^*$. Caso contrário, se $y^* \notin \{0, 1\}$ e a solução (x^*, y^*) não for factível para o MILP, adiciona-se novas restrições lineares para eliminar essa solução da LR.

Seja y_j com $j \in \{1 \dots p\}$ alguma variável de valor fracionário y_j^* na solução da LR (x^*, y^*) . Em qualquer solução inteira factível deve-se ter $y_j \leq \lfloor y_j^* \rfloor$ ou $y_j \geq \lceil y_j^* \rceil$, sendo $\lfloor y_j^* \rfloor$ e $\lceil y_j^* \rceil$ o valor de y_j^* arredondado para o inteiro mais próximo abaixo e acima, respectivamente.

Com a finalidade de eliminar a solução (x^*, y^*) e as soluções $\lfloor y_j^* \rfloor \leq y_j^* \leq \lceil y_j^* \rceil$ o conjunto P_X é substituído pela união de dois conjuntos disjuntos P_X^0 e P_X^1 (POCHET; WOLSEY, 2006a):

$$P_X^0 = P_X \cap \{(x, y) \in \mathbb{R}_+^n \times \mathbb{R}_+^p : y_j \leq \lfloor y_j^* \rfloor\} \quad \text{e} \quad (2.6)$$

$$P_X^1 = P_X \cap \{(x, y) \in \mathbb{R}_+^n \times \mathbb{R}_+^p : y_j \geq \lceil y_j^* \rceil\} \quad (2.7)$$

A variável y_j é chamada de variável de ramificação (*branching variable*) e as restrições $y_j \leq \lfloor y_j^* \rfloor$ e $y_j \geq \lceil y_j^* \rceil$ são chamadas de restrições de ramificação (*branching constraints*).

Dessa forma é possível substituir a busca da melhor solução inteira em P_X pela busca da melhor solução inteira em $P_X^0 \cup P_X^1$. O preço de preservar a linearidade dos programas é que um programa linear foi substituído por dois programas lineares definidos sobre dois conjuntos disjuntos.

Depois da ramificação (*branching*) a melhor solução inteira da lista $L = \{P_X^0, P_X^1\}$ é selecionada e prossegue-se com a abordagem de decomposição da mesma maneira. Isso requer que cada formulação da lista L seja analisada separadamente.

Iteração principal Neste ponto são conhecidos a lista de formulações L e o valor \bar{Z} da melhor solução inteira encontrada até o momento. Se nenhuma solução inteira for encontrada o limite inferior do MILP é definido para $\bar{Z} = +\infty$.

Seleção e solução Seja V o conjunto de pontos dentro da região viável do problema na Equação 2.3. Seleciona-se uma formulação V da lista L , e soluciona-se o programa linear correspondente para obter $Z(V)$ e uma solução ótima (x^V, y^V) . A solução $Z(V)$ encontrada é um limite inferior (maximização) da melhor solução inteira do conjunto V .

Eliminação O subconjunto da região factível V pode ser eliminado ou dividido em mais subconjuntos, a depender de algumas eventualidades:

- (a) Se $Z(V) \leq \bar{Z}$, então a melhor solução em V não pode ser estritamente melhor que \bar{Z} , porque $Z(V)$ é um limite superior (maximização) no valor da melhor solução em V . Portanto não é necessário considerar as soluções inteiras em V , que é simplesmente removido da lista L . Este procedimento é chamado de eliminação por limite ou *pruning by bound*;
- (b) Como um caso especial do item anterior, quando V é vazio tem-se $Z(V) = +\infty \geq \bar{Z}$ e pode-se remover V da lista L . Esse procedimento é chamado eliminação por inviabilidade ou *pruning by infeasibility*;
- (c) Se $Z(V) > \bar{Z}$ e $y^V \in \{0, 1\}$, então a melhor solução inteira (x^V, y^V) em V foi encontrada, e essa solução supera o valor da melhor solução conhecida até o momento. Como consequência não é necessário continuar decompondo o conjunto V . O melhor valor da solução é armazenado, e \bar{Z} é atualizado para $\bar{Z} = Z(V)$. O conjunto V é removido da lista L . Este procedimento é chamado eliminação por integralidade *pruning by integrality*;
- (d) Se $Z(V) \geq \bar{Z}$ e $y^V \notin \{0, 1\}$, então a solução ótima do programa linear V é fracionada e o valor da melhor solução inteira em V pode ou não superar a melhor solução conhecida \bar{Z} . Portanto é preciso decompor o conjunto V em V^0 e V^1 e adicioná-los à lista L e remover V da lista. Este procedimento é chamado de ramificação ou *branching*.

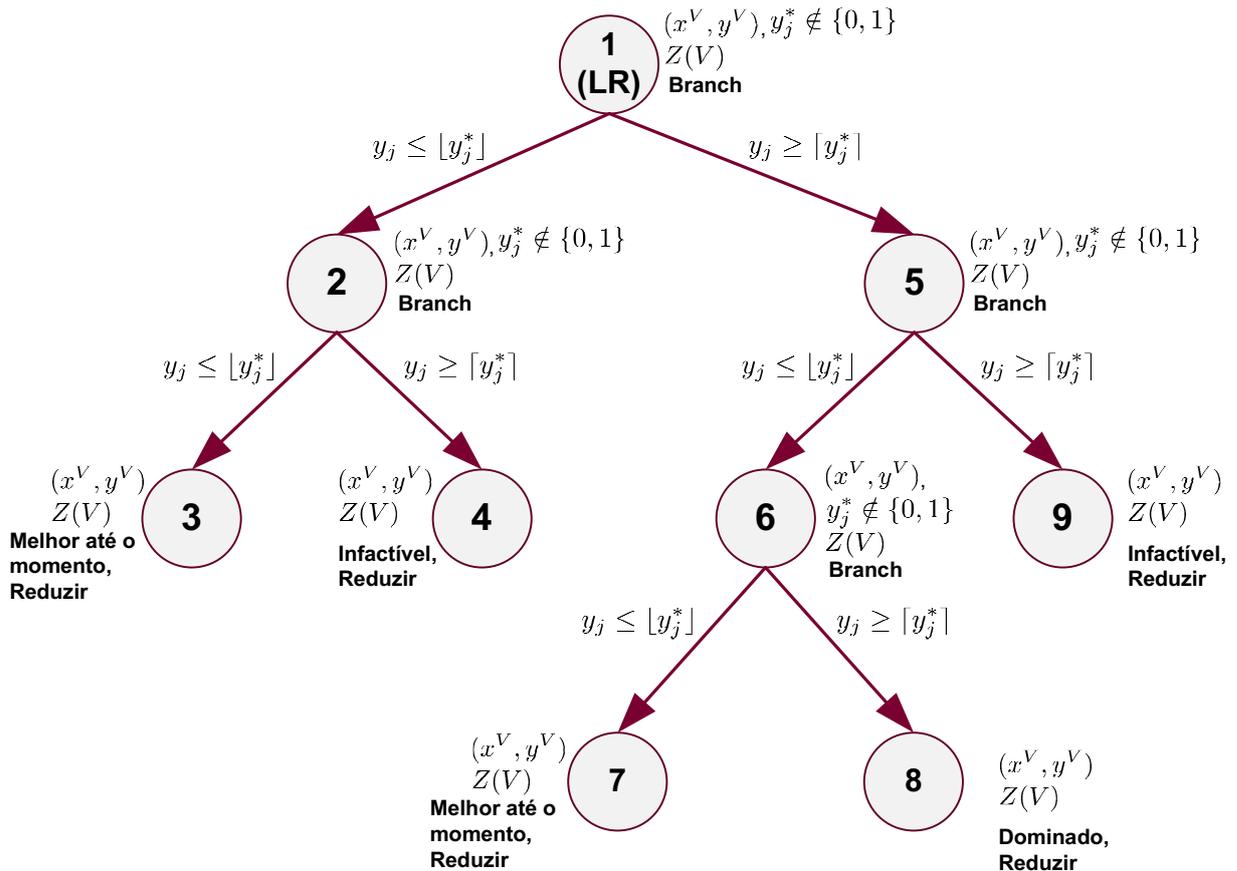


Figura 2.3: Esquema ilustrativo da árvore de busca *Branch & Bound* (adaptado de POCHET; WOLSEY, 2006a).

A Figura 2.3 mostra um esquema ilustrativo da árvore de busca construída pelo algoritmo *Branch & Bound*. No primeiro nó a *LR* é solucionada, fornecendo um limite superior $Z(P_X)$ (maximização) para a função objetivo. O primeiro nó é ramificado ao arredondarmos o valor fracionado da variável $y_j \notin \{0, 1\}$ para o menor inteiro mais próximo, gerando o segundo nó. No segundo nó também é feita uma ramificação, que dá origem aos nós três e quatro. Observa-se que a árvore de busca ilustrada na Figura 2.3 prioriza uma busca em profundidade. Essa é uma regra de ramificação ou regra de *branching*, que pode ser selecionada pelo usuário em algoritmos comerciais. No terceiro nó uma solução inteira é encontrada, e ela passa a ser o limite inferior \bar{Z} . O quarto nó é reduzido por inviabilidade, ou seja, não há solução inteira na região factível V do nó quatro. O algoritmo volta então à ramificação do primeiro nó ao arredondar $y_j \notin \{0, 1\}$ para cima, e começa a busca no nó cinco. O nó cinco é ramificado para o

seis e nove. O nó seis é ramificado para o sete e oito. No nó sete o algoritmo encontra outra solução viável melhor que a anterior $Z(V) > \bar{Z}$, e o limite inferior \bar{Z} é atualizado (eliminação por integralidade). No oitavo nó a solução $Z(V)$ também é viável, mas menor que o limite inferior \bar{Z} , e o nó pode ser descartado (eliminação por limite). Por último o algoritmo explora o nó número nove, que é reduzido por inviabilidade.

Qualidade da solução Teoricamente o número de iterações requeridas pelo algoritmo *Branch & Bound* é uma função exponencial do número de variáveis inteiras p do modelo. No entanto, aplicações industriais – em particular problemas de programação e planejamento da produção – geralmente dão origem à modelos muito grandes e é conveniente truncar a árvore de busca do *Branch & Bound*, pois resolver o MILP até a otimalidade pode levar muito tempo. Nessas ocasiões os procedimentos de seleção de nós e regras de *branching* são especialmente relevantes para a qualidade da solução, que pode ser medida pelo *gap* de otimalidade (POCHET; WOLSEY, 2006a).

Quando a enumeração do *Branch & Bound* não é finalizada, o *gap de otimalidade* mede o desvio relativo máximo entre a solução retornada e o melhor limite superior encontrado pelo algoritmo. Ele é definido por Pochet e Wolsey (2006a) como:

$$\text{Gap de otimalidade (\%)} = 100 \cdot \frac{\text{Melhor limite superior} - \text{Melhor limite inferior}}{\text{Melhor limite superior}} \quad (2.8)$$

O *gap de otimalidade* também é chamado na literatura de *gap de dualidade* (POCHET; WOLSEY, 2006a) e *gap de integralidade* (MÉNDEZ et al., 2006).

Complexidade Como saber se um algoritmo é adequado para resolver um dado problema? Os dois critérios naturais de desempenho usados para analisar ou comparar comportamento de algoritmos são a qualidade da solução e o tempo necessário para encontrá-la (POCHET; WOLSEY, 2006a). No caso de MILPs a qualidade de uma solução (x, y) é definida pelo valor do objetivo $cx + fy$ ou por uma função desse valor como, por exemplo, o *gap* de otimalidade.

Em teoria da complexidade a duração de um algoritmo é medida pelo número de operações algébricas elementares $(+, -, \times, /)$ realizadas, e para investigar esse parâmetro adota-se uma perspectiva assintótica, que é a ordem de duração (BRUCKER; KNUST, 2012). A grosso modo, a ordem de duração $f(n)$ de um algoritmo pode

ser enxergada como o limite superior da duração quando o tamanho do modelo n é suficientemente grande. A notação matemática para a ordem de duração é $O(f(n))$ e valores típicos de $f(n)$ são n , n^2 , $\log n$, 2^n .

Se um algoritmo polinomial – cuja ordem de duração é $O(n^p)$, p fixo – é capaz de resolver um dado problema até a otimalidade, esse problema é considerado "fácil". Programas lineares pertencem à classe de problemas polinomiais. Programas inteiro-misto requerem algoritmos exponenciais e pertencem à uma classe especial de problemas "difíceis", chamada *non-deterministic polynomial-time hard* (NP-hard) (BRUCKER; KNUST, 2012; PINEDO, 2012c). O problema da mochila (*knapsack problem*) é um exemplo de MILP NP-hard.

Programas inteiro-misto não-lineares

Um problema de *programação inteiro-misto não-linear* (MINLP) é um problema de otimização que envolve variáveis contínuas, variáveis inteiras e equações não-lineares, que podem estar ou nas restrições ou na função objetivo. Quando a função objetivo é linear e não-linearidades aparecem apenas nas restrições, esse problema de otimização pode ser representado pelas equações 2.9 e 2.10.

$$Z(X) = \max_{(x,y)} \{cx + fy \mid (x, y) \in X\} \quad (2.9)$$

$$X = \{(x, y) \in \mathbb{R}_+^n \times \{0, 1\}^p \mid f(x, y) \geq b\} \quad (2.10)$$

onde

- $f(x, y)$ é uma função não necessariamente linear das variáveis x contínua e y binária.

De acordo com Floudas (1995a), para uma função duas vezes diferenciável $g(x)$, o ponto x é um ótimo local x^* , se e somente se estiver de acordo com as condições necessárias de otimalidade:

- o valor da primeira derivada da função objetivo é zero $\nabla g(x^*) = 0$;
- a matriz hessiana $H(x^*)$ deve ser definida positiva.

O ponto $x \in \mathcal{R}^n$ é um máximo local x^* se houver ao redor dele uma esfera de raio ϵ onde todas as outras funções objetivo são menores que a de x^* : $g(x^*) \geq g(x), \forall x \in B_\epsilon(x^*)$. Já um máximo global x^* é o ponto em que o valor da função é máximo em comparação a todos os demais: $g(x^*) \geq g(x), \forall x \in \mathcal{R}^n$.

Uma característica importante da região factível X nas equações 2.2, 2.4 e 2.10, é se ela é convexa ou não-convexa, essa última também chamada de côncava. Regiões convexas e não-convexas possuem propriedades matemáticas diferentes (FLOUDAS, 1995b).

Numa região convexa, se ligarmos dois pontos interiores da região X por uma reta, todos os pontos da reta estarão dentro de X . Já numa região não-convexa, se ligarmos dois pontos interiores da região X por uma reta, alguns pontos da reta ficam fora de X . A Figura 2.4 ilustra ambas regiões em duas dimensões.

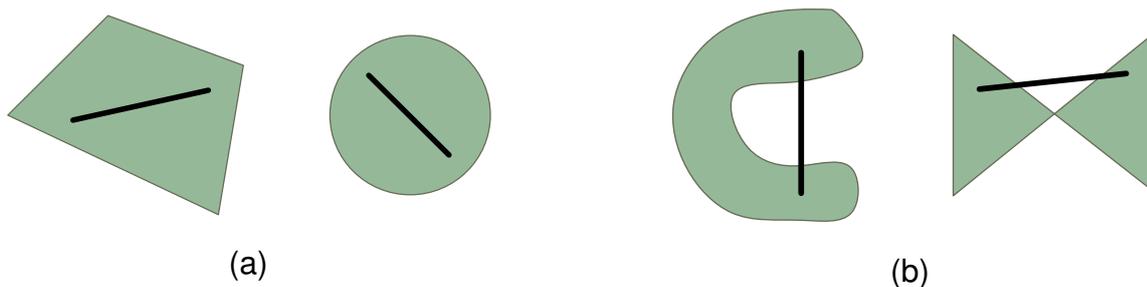


Figura 2.4: Regiões (a) convexas e (b) não-convexas (FLOUDAS, 1995a).

Quando a região X é convexa e também o é a função objetivo $Z(X)$, é possível provar matematicamente que um ótimo local é também o ótimo global. Quando X é não-convexa algoritmos de busca local nem sempre conseguem encontrar o ótimo global. Para solucionar problemas NLP ou MINLPs cuja região factível X é não-convexa até a otimalidade algoritmos específicos de otimização global são necessários. Problemas de programação matemática que combinam programas inteiros (NP-hard) e regiões não-convexas, caso dos MINLPs, são difíceis.

Segundo Liberti e Pantelides (2005) algoritmos do tipo *Branch & Bound* Espacial (*Spatial Branch & Bound*, SBB) estão entre os métodos mais eficientes para resolver NLPs ou MINLPs não-convexos. Nesses métodos, um limite superior (maximização) da função objetivo é obtido quando se substitui a restrição não-convexa por sua relaxação convexa. Um requerimento importante dos algoritmos SBB é que seja possível construir uma relaxação convexa justa do NLP em toda a região factível X .

Alguns algoritmos de busca local para solucionar problemas NLP são métodos de Gradiente Reduzido (*Generalized Reduced Gradient*, GRG) e métodos de Programação Quadrática Sequencial (*Sequential Quadratic Programming*, SQP). Exemplos de algoritmos de busca local para solução de problemas MINLP são métodos de Decomposição de *Benders* (*Generalized Benders Decomposition*, GBD) e Aproximação Exterior (*Outer Approximation*, OA).

A maioria dos algoritmos estado-da-arte para encontrar soluções do tipo ótimo global são baseados no algoritmo *Branch & Bound*¹. Alguns deles são Navegador de Otimização *Branch & Reduce* (*Branch & Reduce Optimization Navigator*, BARON) e Otimizador Global Inteiro-misto Quadrático (*Global Mixed-integer Quadratic Optimizer*, GloMIQO)

O *solver* usado para solucionar os problemas NLP nessa dissertação tem como base métodos de Gradiente Reduzido (GRG). Nos métodos de Gradiente Reduzido a ideia básica é resolver uma sequência de subproblemas com restrições generalizadas, onde os subproblemas são resolvidos por eliminação de variáveis.

Formulações da literatura

Para selecionar um modelo adequado ao sistema que se deseja estudar é preciso antes conhecer algumas características do sistema. Méndez et al. (2006) apresentam o estado da arte dos modelos de otimização para programação da produção de processos em batelada na literatura de PSE. Eles apresentam exemplos e discutem sobre quais modelos de otimização são mais adequados para quais tipos de problema de programação.

O principal critério usado por Méndez et al. (2006) na hora de selecionar um modelo de programação adequado ao sistema é sua topologia: sequencial ou em rede. Processos sequenciais podem ser modelados de maneira análoga à processos da indústria discreta (MARAVELIAS, 2012). Já processos em rede exigem balanços de massa de acordo com a receita dos produtos.

Segundo Pinedo (2012a) um problema genérico de programação é descrito

¹Em: Universidade Humboldt de Berlim – *Nonconvex Mixed-Integer Nonlinear Programming*, <<http://www.math.hu-berlin.de/~stefan/B19/>>. Acesso em 22 de Maio de 2015.

pela tríade $\alpha|\beta|\gamma$. O campo α descreve o ambiente máquina, o campo β pode conter várias entradas e fornece detalhes sobre o processamento e limitações de recursos, como por exemplo datas de entrega, restrições de precedência, tempos de preparação vinculados à sequencia das tarefas, famílias de tarefas, recirculação de material, condição zero espera (*no wait*), entre outros. O campo γ descreve o objetivo da programação. Os ambientes máquina dizem respeito ao tipo de configuração disponível para processar as tarefas.

Os ambientes máquina da indústria discreta são (MARAVELIAS, 2012; PINEDO, 2012a; BŁAŻEWICZ et al., 2007):

- Máquina única ;
- Máquinas em paralelo ;
- *Flow-shop* ;
- *Job-shop* ;
- *Open-shop* ;
- *Flow-shop* flexível ;
- *Job-shop* flexível ;
- *Open-shop* flexível .

Os ambientes máquina da indústria discreta possuem similaridades com os da indústria de processos químicos em batelada. Em ambos os casos um lote mantém sua identidade ao longo do processo, isto é, não se separa para formar vários menores, nem se junta com nenhum outro.

A topologia do processo está condicionada à existência ou não de mistura e separação de bateladas no processo. A Figura 2.5 ilustra os conceitos de mistura e separação de bateladas conforme Maravelias (2012). Mistura de bateladas é a mistura de dois lotes do mesmo material produzidos pela mesma tarefa. Se permitida, essa característica constitui processo em rede. Mistura de materiais acontece quando uma tarefa requer dois tipos de material que precisam, portanto, ser misturados. Separação de batelada acontece quando um lote é dividido e consumido em várias tarefas subsequentes, que é diferente de quando uma tarefa produz materiais diferentes. De acordo com a classificação de Maravelias (2012), separação de bateladas também constitui processo em rede.

Processos sequenciais, segundo Maravelias (2012), são aqueles em que não há mistura e separação de bateladas. Eles consomem e produzem um único material e bateladas não se misturam nem se separam. Instalações dessa natureza podem

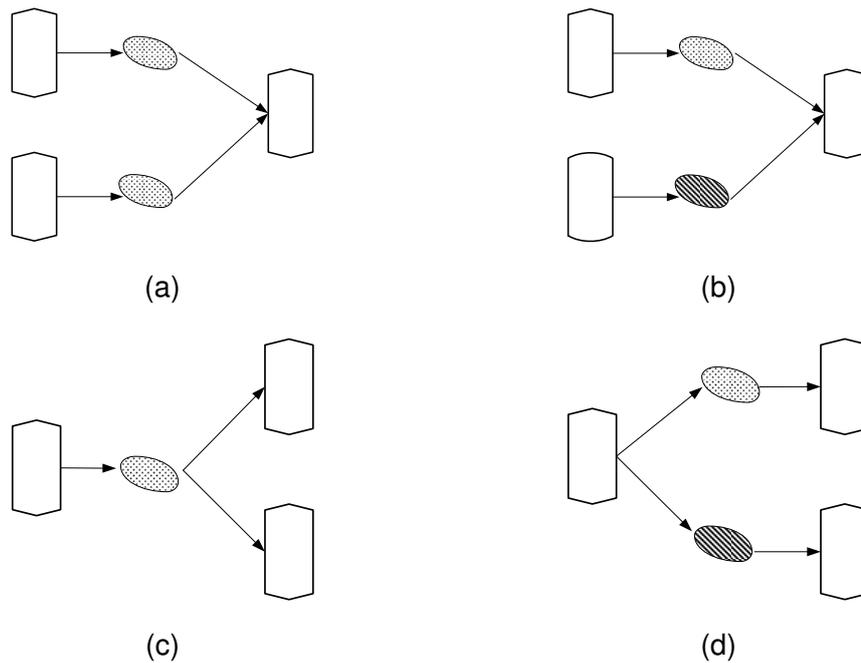


Figura 2.5: Diferença entre (a) mistura de bateladas ou mistura de dois lotes do mesmo material produzidos pela mesma tarefa, (b) mistura de materiais ou quando uma tarefa requer dois tipos de material, (c) separação de batelada ou quando um lote é dividido e consumido em várias tarefas subsequentes e (d) tarefa que produz materiais diferentes.

ser multi-estágio, análogas ao *flow-shop* flexível, ou multi-propósito, análogas ao *job-shop flexível*. A Figura 2.6 mostra um exemplo de processo sequencial multi-estágio estudado por Pinto e Grossmann (1995), composto por cinco estágios, vinte e cinco unidades ou máquinas e que precisa processar cinco pedidos – cinco produtos diferentes – num horizonte de tempo de 500 horas. Os pedidos $\{1, 2, 3, 5\}$ não podem ser processados nas unidades $\{18, 19, 22, 25\}$ e o pedido $\{2\}$ não pode ser processado nas unidades $\{20, 21, 23, 24\}$. Há tempos de transição vinculados à unidade e a capacidade de armazenamento entre estágios é ilimitada.

De acordo com Pinedo (2012a) frequentemente o algoritmo usado para resolver um problema de programação P_1 pode ser aplicado a outro problema P_2 também. Diz-se então que P_2 se reduz a P_1 ou simplesmente $P_2 \alpha P_1$. Naturalmente há também problemas que não são comparáveis com nenhum outro. A Figura 2.7 mostra a hierarquia da complexidade de alguns problemas determinísticos de programação com relação ao ambiente máquina.

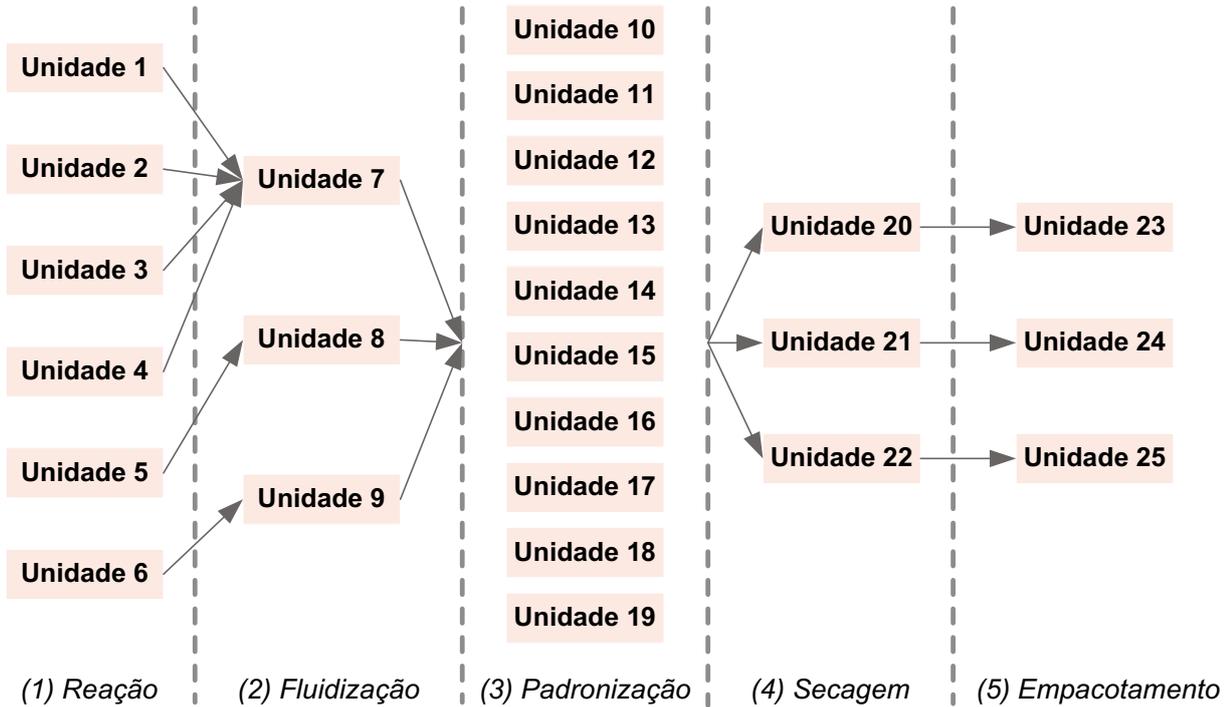


Figura 2.6: Exemplo de processo sequencial multi-estágio (adaptado de PINTO; GROSSMANN, 1995).

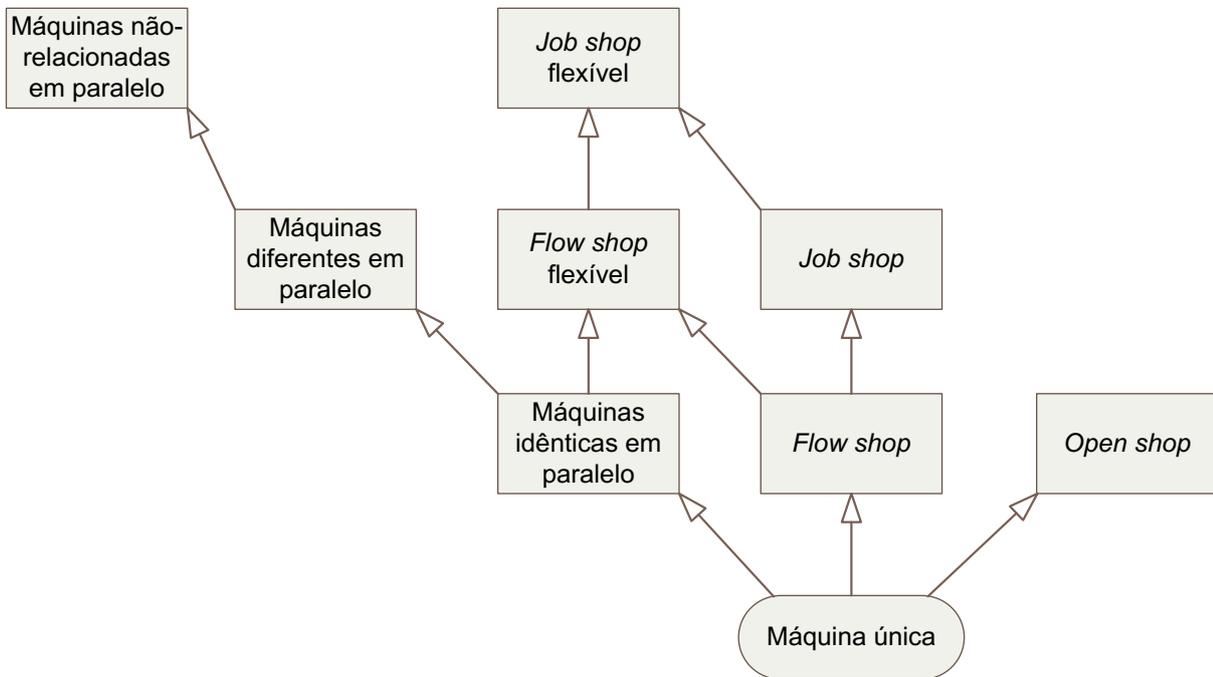


Figura 2.7: Hierarquia da complexidade de problemas de programação da produção com relação ao ambiente máquina (adaptado de PINEDO, 2012a).

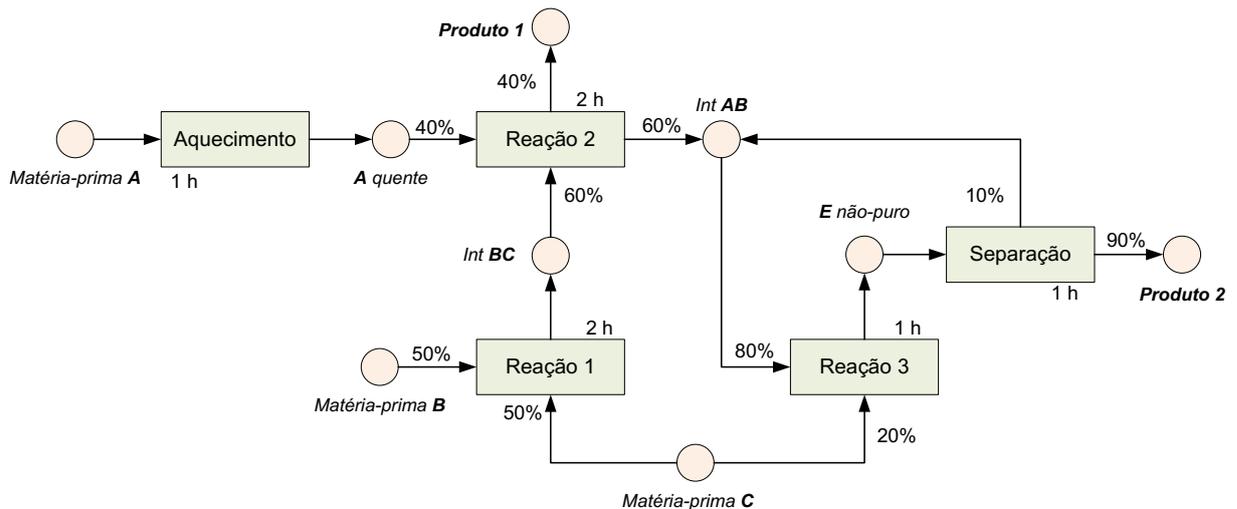


Figura 2.8: Exemplo de uma rede estado-tarefa (adaptado de KONDILI et al., 1993).

Em contraste com processos sequenciais há os processos em rede: aqueles em que não há restrições acerca das movimentações de materiais. Processos em rede são caracterizados pela existência de mistura e separação. Um aspecto notável dos processos em rede é que o material pode escoar livremente do estoque para unidades de processamento e vice-versa. Uma tarefa é identificada pelos materiais que ela consome e produz e pelos coeficientes de conversão correspondentes, e não por um material passando por um estágio como nos processos sequenciais. Além disso, o conceito de batelada é irrelevante porque as bateladas não mantêm sua identidade (MARAVELIAS, 2012).

Em modelos de programação da produção para processos em rede, os balanços de massa são equacionados de acordo com as representações *rede estado-tarefa* (STN) ou *rede recurso-tarefa* (RTN). As redes STN e RTN são grafos orientados bipartidos (CASTRO et al., 2001).

A representação STN, ilustrada na Figura 2.8, é um grafo orientado bipartido proposta por Kondili et al. (1993). No sistema representado pela Figura 2.8 observamos também um exemplo de mistura de materiais conforme a definição de Maravelias (2012), item (b) da Figura 2.5. As reações 1, 2 e 3 precisam de dois materiais que se misturam antes de serem processados.

Outra representação por meio de grafo bipartido usada para modelar proble-

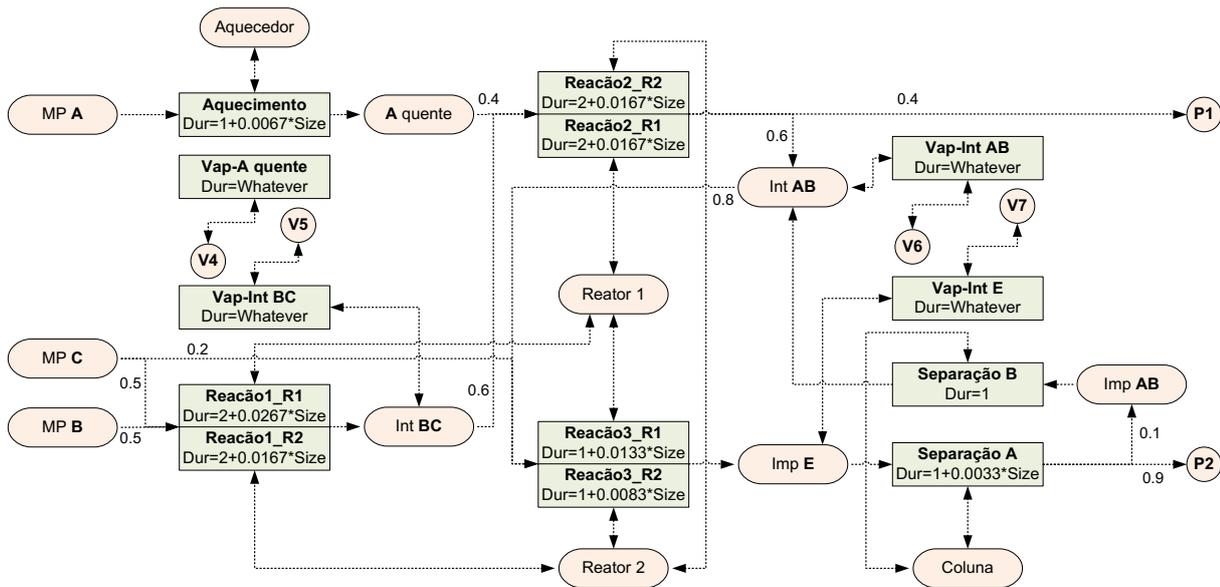


Figura 2.9: Exemplo de uma rede recurso-tarefa (adaptado de MÉNDEZ et al., 2006).

mas de programação da produção é a RTN proposta por Pantelides (1994). Há vértices para tarefas –os retângulos– e vértices para recursos –os círculos. Cada tarefa é uma operação que transforma um certo conjunto de recursos em outro conjunto. Castro et al. (2001) destacam que nesta representação o conceito de recurso é abrangente: são todas as entidades envolvidas no decorrer do processo, i.e. materiais ou estados tais como matérias-primas, intermediários e produtos; equipamentos ou unidades de processamento e armazenamento tais como tanques, reatores, etc; e utilidades tais como mão-de-obra, vapor, etc. A Figura 2.9 exibe a representação RTN do mesmo processo da Figura 2.8.

As Figuras 2.8 e 2.9 representam um processo em rede de fabricação de dois produtos a partir de três matérias-primas *A*, *B*, e *C*. O primeiro passo na receita é aquecer a matéria-prima *A* por 1 h. As matérias-primas *B* e *C* são misturadas numa proporção de 1:1 e reagem por 2 h na reação 1. Depois de aquecido, *A* reage com o intermediário *BC* na reação 2, na proporção de 2:3 por 2 h, formando intermediário *AB* e o produto 1 na proporção de 3:2. Na reação 3 a matéria-prima *C* reage com o intermediário *AB* por 1 h na proporção de 1:4 para formar o intermediário *E* mais impurezas. O intermediário *E* é separado da corrente de saída da reação por destilação para formar o produto 2 (90% em 1 h) e intermediário *AB* (10% em 2 h), que é reci-

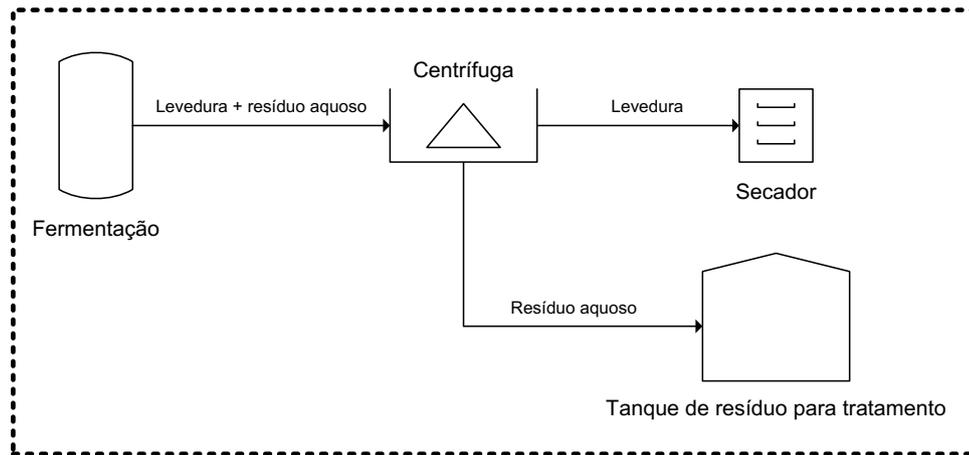


Figura 2.10: Processo hipotético de produção de levedura.

clado de volta para a reação 3. Os equipamentos disponíveis são um aquecedor, dois reatores e uma coluna de destilação. Para as matérias-primas A , B e C a capacidade de armazenamento é ilimitada – política de inventário ilimitado – e para A quente e os demais intermediários a capacidade de armazenamento é finita – política de inventário finito.

Maravelias (2012) descreve um terceiro tipo de ambiente de produção: os processos híbridos. Eles não são nem sequenciais nem em rede, mas exibem características de ambos. O principal atributo dos processos híbridos é que a identidade da batelada deve ser mantida e monitorada somente para alguns materiais, enquanto outros – e.g. aditivos e subprodutos – podem se movimentar livremente.

Por exemplo, no processo hipotético de fabricação de levedura da Figura 2.10 a centrifugação de uma batelada que acaba de ser fermentada resulta em um concentrado líquido e um subproduto aquoso. O concentrado é enviado para secagem sem mistura ou separação de bateladas (processo sequencial), e o subproduto aquoso precisa ser tratado. Se houver uma capacidade de armazenamento finita para a fase aquosa, ela pode ser misturada à subprodutos de outras fermentações (processo em rede) até que um volume suficientemente grande seja coletado para tratamento. Se neste exemplo a capacidade de armazenamento da fase aquosa for ilimitada, [armazenamento ilimitado de intermediários \(UIS\)](#), o processo pode ser tratado como sequencial.

Outros critérios apresentados por Méndez et al. (2006) para selecionar o mo-

delo de programação da produção mais adequado ao sistema de estudo são considerações acerca da política de inventário, conectividade entre equipamentos, tempos de transição, padrões de demanda, custos, grau de certeza, tamanho do lote ou batelada, restrições de recursos, entre outros.

Classificação dos modelos

De acordo com Méndez et al. (2006) a formulação matemática tem um impacto direto na performance computacional, limitações e resultados do modelo. Cada formulação possui suas peculiaridades. Pochet e Wolsey (2006b) relatam que modelar o planejamento da produção é quase uma arte, na medida que estratégias de reformulação podem ser usadas para melhorar a performance do modelo.

Méndez et al. (2006) classifica as formulações da literatura segundo (i) representação do tempo, (ii) balanços materiais, (iii) representação dos eventos e (iv) função objetivo, conforme ilustrado na Figura 2.11.

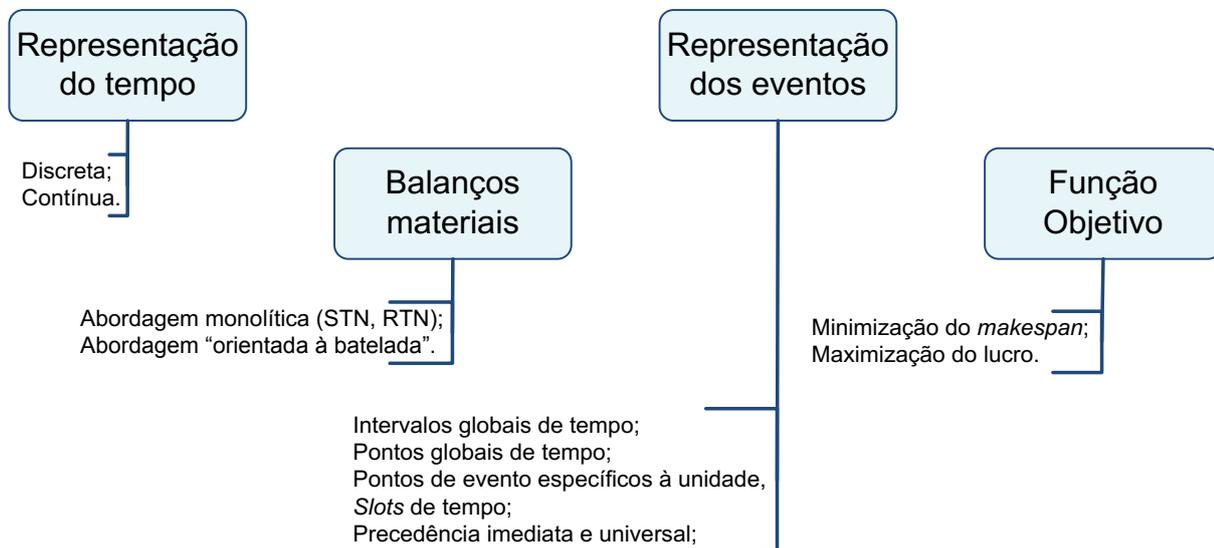


Figura 2.11: Critérios de classificação dos modelos de programação da produção segundo Méndez et al. (2006).

As alternativas de representação do tempo são modelos de tempo discreto ou de tempo contínuo. Em modelos de tempo discreto o horizonte de tempo pode ser

dividido em um número finito de intervalos iguais ou múltiplas grades não uniformes de tempo (HARJUNKOSKI et al., 2014). Em modelos de tempo contínuo as decisões de cronometragem são explicitamente representadas por conjuntos de variáveis contínuas, o que permite uma considerável redução no número de variáveis. Por outro lado, quando há no sistema recursos compartilhados ou inventário finito nos recursos a cronometragem é feita por restrições do tipo *Big-M*. Esse tipo de restrição compromete a performance computacional do modelo, pois leva à relaxações lineares P_X pouco justas.

Quanto aos balanços materiais Méndez et al. (2006) cita métodos monolíticos e métodos orientados à batelada, para processos em rede e para processos sequenciais, respectivamente. Método monolíticos resolvem simultaneamente os problemas do número e tamanho das bateladas, alocação dos recursos, sequenciamento e cronometragem das tarefas. Estes métodos são capazes de lidar com redes arbitrárias de processamento e receitas complexas. Eles fazem uso de representação *STN* ou *RTN* para modelar o fluxo de matéria no sistema produtivo. Métodos orientados à batelada ou ao pedido são realizados em duas etapas. Com base em um certo critério de otimalidade agrupa-se os pedidos em várias bateladas de tamanho definido, ou seja, a determinação do número e tamanho das bateladas. Numa segunda etapa faz-se a alocação dos recursos às tarefas ao longo do tempo e o sequenciamento das bateladas

Modelos de programação da produção se fundamentam em diferentes ideias para acomodar os eventos do programa ao longo do tempo e garantir que a capacidade máxima dos recursos compartilhados nunca seja excedida (MÉNDEZ et al., 2006). Cada tipo de representação de eventos é base para o desenvolvimento de várias formulações. Elas diferem entre si na adequabilidade aos dois tipos de topologia em sua abrangência, que depende da forma de representação do tempo.

Méndez et al. (2006) reporta cinco tipos de representação do tempo: intervalos globais de tempo, pontos globais de tempo, eventos de tempo específicos à unidade *slots* de tempo e precedência.

Para modelos de tempo discreto, uma opção são os intervalos globais de tempo, adequado para lidar tanto com processos sequenciais quanto em rede. Nestas formulações o problema original se transforma num problema de alocação, no qual a

principal decisão do modelo é a designação do intervalo de tempo em que cada tarefa acontece. A grande vantagem de usar grade de tempo fixa é que aspectos do problema que dependem do tempo – e.g. consumo de utilidades, demanda por vários produtos, suprimento de matéria-prima ao longo do tempo – podem ser modelados de maneira relativamente simples sem comprometer a linearidade do modelo. A formulação dos balanços materiais podem ser baseadas em grafos [STN](#) ou [RTN](#). Há formulações em que ambos início e fim da tarefa devem coincidir com os do intervalo e em que apenas o início da tarefa deve coincidir com o do intervalo.

Em modelos de tempo contínuo várias são as alternativas de representação de evento tanto para processos sequenciais quanto redes. Para processos em rede as abordagens compatíveis são pontos globais de tempo e eventos de tempo específicos à unidade e para processos sequenciais as alternativas são *slots* de tempo e representações baseadas na precedência de tarefas.

Na representação por pontos globais de tempo a posição do limite entre os intervalos, ou seja, os pontos globais, são decididos pelo modelo. A grade de tempo é variável – i.e. os intervalos não possuem necessariamente a mesma duração – e comum à todos os recursos.

A representação por eventos de tempo específicos à unidade tem um melhor desempenho computacional que as representações com pontos globais, pois requer menos pontos de evento (MOURET et al., [2009](#)). Ela é ainda mais geral: a grade de tempo também é variável mas cada recurso tem sua própria grade, isto é, as grades de tempo são específicas à unidade. Para equacionar o fluxo de matéria no processo usam-se os grafos [STN](#) ou [RTN](#).

O desempenho computacional de modelos com pontos globais de tempo e eventos de tempo específicos à unidade está fortemente condicionado à quantidade mínima de pontos de tempo ou de eventos necessários para gerar um programa factível (MÉNDEZ et al., [2006](#); MARAVELIAS; GROSSMANN, [2003](#); IERAPETRITOU; FLOUDAS, [1998a](#)). Uma vez que a quantidade de pontos de tempo ou eventos necessária para gerar um programa ótimo não é sabida *a priori*, um critério prático adotado na literatura para determiná-los é aumentar essa quantidade de um em um até que não haja melhora na função objetivo. Mouret et al. ([2011](#)) chama esse procedimento de abordagem aditiva.

Esse procedimento heurístico foi proposto por Ierapetritou e Floudas (1998a) quando a representação por pontos de evento específicos à unidade foi introduzida. A ideia inédita do modelo de Ierapetritou e Floudas (1998a) foi desacoplar os eventos-tarefa dos eventos-unidade, implementada ao se considerar um conjunto de variáveis binárias para cada uma dessas ocorrências: $wv(i, n)$ para ocorrência da tarefa i no ponto de evento n e $yv(j, n)$ para utilização da unidade j no ponto de evento n .

A mesma estratégia foi usada por Maravelias e Grossmann (2003) sob a premissa de que diminui a quantidade de variáveis binárias do modelo. No entanto, Sundaramoorthy e Karimi (2005) mostraram que a quantidade de variáveis binárias é exatamente a mesma com ou sem a estratégia de desacoplamento. Segundo Sundaramoorthy e Karimi (2005) quando se desacopla eventos-tarefa de eventos-unidade cada tarefa i deve ser executada por apenas uma unidade j . Caso uma tarefa i possa ser executada em qualquer das unidades j_1 ou j_2 , há de se criar duas tarefas no modelo para representá-la: a tarefa i em j_1 e a tarefa i' em j_2 . A única diferença entre as duas abordagens é que com $y(i, j, n)$ a informação acerca da unidade j em que a tarefa i acontece está explícita, enquanto com $wv(i, n)$ ela está oculta no índice i .

Pinto e Grossmann (1995) introduziram a representação de evento por *slots* de tempo. De acordo com Méndez et al. (2006) *slots* são intervalos de tempo de duração desconhecida. A ideia principal é estabelecer um número apropriado de *slots* de tempo para cada unidade de processamento com a finalidade de alocar os *slots* às tarefas. Pinto e Grossmann (1995) propõem uma equação para definir o melhor número de *slots* $k \in K$ para cada unidade $j \in J$ em um sistema sequencial com $l \in L$ estágios.

$$|K_j| = \left\lceil \frac{|I_j|}{J_l (l \in L_j)} \right\rceil \quad \forall j \in J \quad (2.11)$$

Na equação 2.11, K é o conjunto de *slots*, I o conjunto de tarefas, J as unidades e L os estágios do processo sequencial. De acordo com Pinto e Grossmann (1995) o tamanho do problema é drasticamente influenciado pelo número de *slots* em cada unidade $|K_j|$. Os autores reportam que $|K_j|$ deve ser o menor possível para manter mínimas exigências computacionais, mas ao mesmo tempo grande o suficiente para que nenhuma solução viável seja excluída. Portanto, na formulação de Pinto e Grossmann (1995) o número de *slots* necessários para um programa satisfatório em

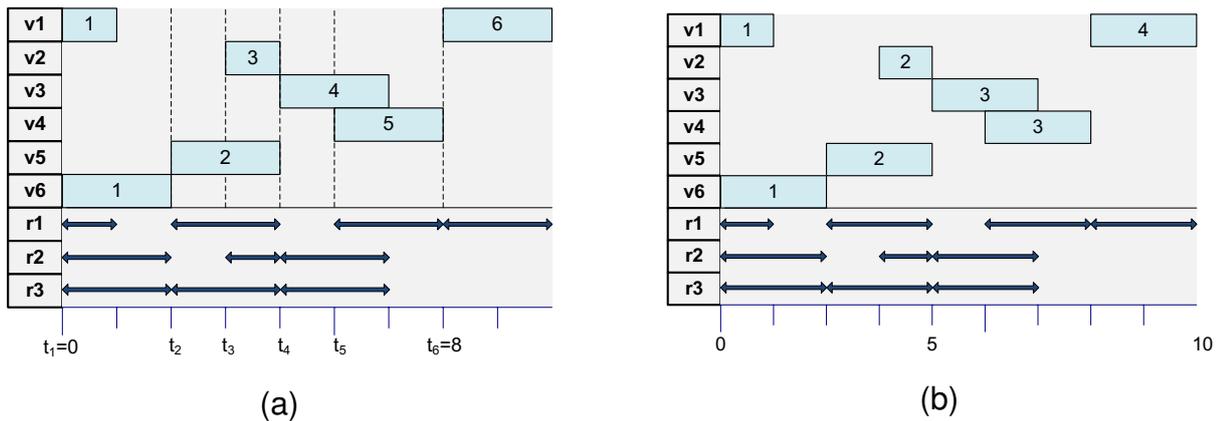


Figura 2.12: Diferença entre a representação de eventos por *slots* (a) síncronos e (b) assíncronos (adaptado de MOURET et al., 2011).

cada unidade $|K_j|$ é determinado *a priori*.

A formulação de Pinto e Grossmann (1995) foi proposta para processos sequenciais. No entanto há modelos na literatura que usam representação por *slots* para modelar processos em rede. Exemplos são os modelos [sequenciamento único de operações \(SOS\)](#) de Mouret et al. (2009) e [sequenciamento múltiplo de operações \(MOS\)](#) de Mouret et al. (2011). Para encontrar o número mínimo de *slots* para gerar o melhor programa os autores usam uma abordagem aditiva.

De acordo com Méndez et al. (2006) representações por *slots* de tempo podem ser síncronas ou assíncronas. A Figura 2.12 ilustra a diferença entre *slots* de tempo síncronos e assíncronos. Observa-se na Figura 2.12 seis operações (v_1 a v_6) sendo alocadas em três recursos (r_1, r_2, r_3). *Slots* de tempo síncronos se assemelham à ideia dos pontos globais de tempo no sentido de que *slots* comuns são definidos para todas as unidades. *Slots* de tempo assíncronos permitem que a posição de um *slot* seja diferente em unidades diferentes.

Outra abordagem compatível com processos sequenciais são os modelos constituídos por relações de precedência entre operações.

Nos modelos de precedência, variáveis e restrições controlam explicitamente a alocação e sequenciamento dos recursos compartilhados. Incluir tempos de transição ou *changeover times* vinculados à sequência nesse tipo de formulação é consideravelmente mais simples. Méndez et al. (2006) destaca que as aplicações se limitam à programação da produção de processos sem controle de inventário. Nos modelos

de precedência imediata a variável binária $X_{ii'}$ é não nula caso a tarefa $i \in I$ aconteça imediatamente antes da tarefa $i' \in I$, enquanto nos modelos de precedência universal o conceito de precedência é estendido para considerar todas as operações executadas anteriormente na mesma unidade de processamento $j \in J$.

Alguns objetivos da programação da produção na indústria de processos químicos são maximização das vendas, minimização dos custos, minimização do *makespan* – tempo total necessário para produzir uma certa quantidade de produto –, minimização da precocidade e minimização dos atrasos (IERAPETRITOU; FLOUDAS, 1998a; IERAPETRITOU; FLOUDAS, 1998b; KONDILI et al., 1993; JIA; IERAPETRITOU, 2004).

REVISÃO BIBLIOGRÁFICA

Neste capítulo os problemas de programação da produção em refinarias são apresentados, com atenção especial ao problema da programação do suprimento de petróleo, objeto de estudo deste trabalho. O problema da programação do suprimento de petróleo é definido, assim como sua configuração e regras operacionais considerados nas aplicações da literatura. Depois discute-se alguns trabalhos que modelam e solucionam o problema de interesse por métodos exatos.

Programação de operações em refinarias

De acordo com Mouret et al. (2009) desde o começo da década de 1980, o uso de procedimentos de análise baseados em programação linear sucessiva (*Branch & Bound*) têm trazido benefícios operacionais para a indústria petroquímica. Formulações que tratam do assunto geralmente levam à modelos inteiro-misto não lineares (MINLP) por causa das regras de mistura e dos modelos reacionais, que são não-lineares.

Jia et al. (2003) propõem uma divisão do sistema produtivo na refinaria em três partes, como mostra a Figura 3.1. É importante frisar que essa divisão é meramente acadêmica, pois em uma refinaria real não há critérios que dividam o processo produtivo dessa forma. O primeiro subproblema é o do suprimento de petróleo: lida com o descarregamento dos navios, mistura de petróleos e controle de inventário. No segundo subproblema é feita a programação das unidades de processo, que envolvem fracionamento e reação. Por último atende-se ao problema do *blending* de produtos finais e expedições.

Nos problemas de programação de operações em refinaria uma mistura acontece quando duas correntes de composição diferente entram em um recurso. Quando

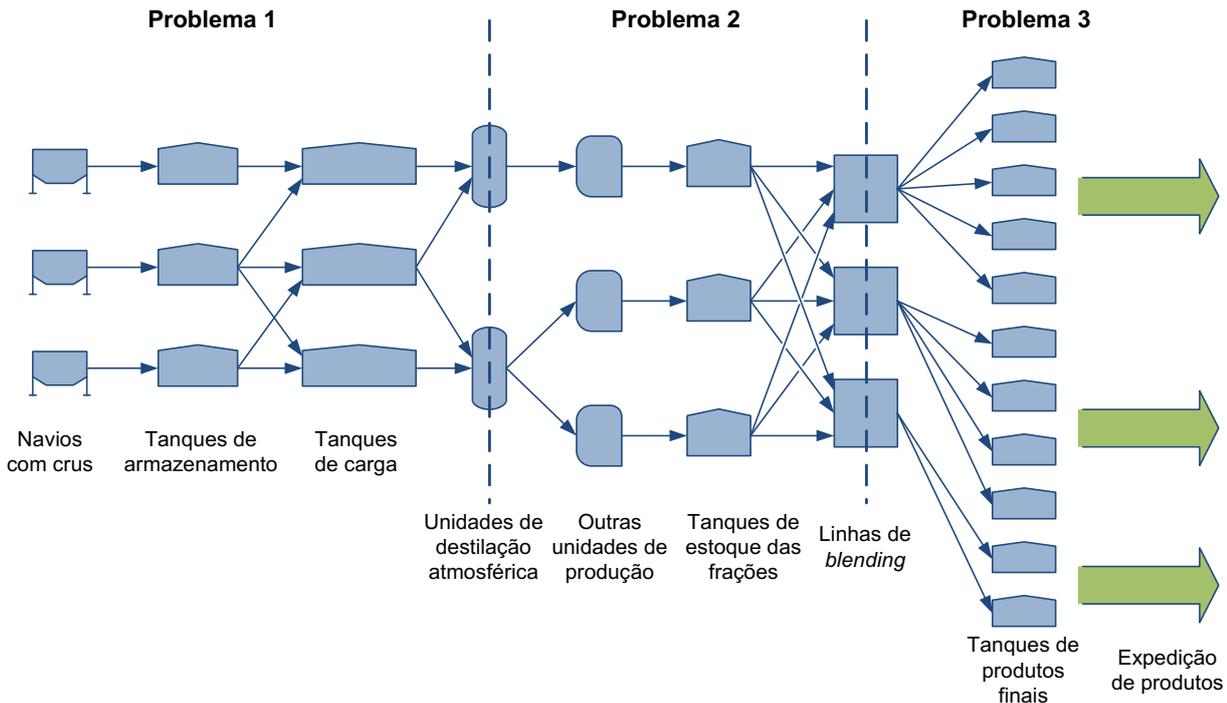


Figura 3.1: Divisão acadêmica da refinaria proposta por JIA et al., 2003 para abordar o problema de programação da produção em refinarias (adaptado de JIA et al., 2003).

um recurso é parcialmente descarregado há uma separação da mistura de petróleo, intermediário ou produto final dentro do recurso: parte permanece e outra parte, de mesma composição, é transferida (MOURET et al., 2009). Essas são as características de um problema de *pooling*.

Segundo Andrei (2013) um problema de *pooling* é um problema de mistura de fluxos multicomponentes de matérias primas em um ou mais reservatórios (*pool*), de modo a obter produtos de especificação desejada em um terceiro estágio. Termos bilineares – que são produtos entre duas variáveis – são característicos desses modelos e surgem por causa das regras de mistura (QUESADA; GROSSMANN, 1995). Por conseguinte, o modelo da refinaria real estudada neste trabalho dá origem a um programa **MINLP** de função objetivo linear e em que uma das restrições é não-linear.

No problema da programação do suprimento de petróleo os termos bilineares surgem na restrição que impõe igualdade de composição nos recursos e em suas respectivas operações de saída. Eles também podem surgir por causa das regras de mistura e de imposições sobre vazões. Nos modelos desenvolvidos nessa dissertação a restrição não linear envolve dois termos bilineares. Devido a esse fato, a região do

problema MINLP resultante do estudo de caso é não-convexa. A Figura 3.2 mostra o gráfico resultado do produto entre duas variáveis contínuas.

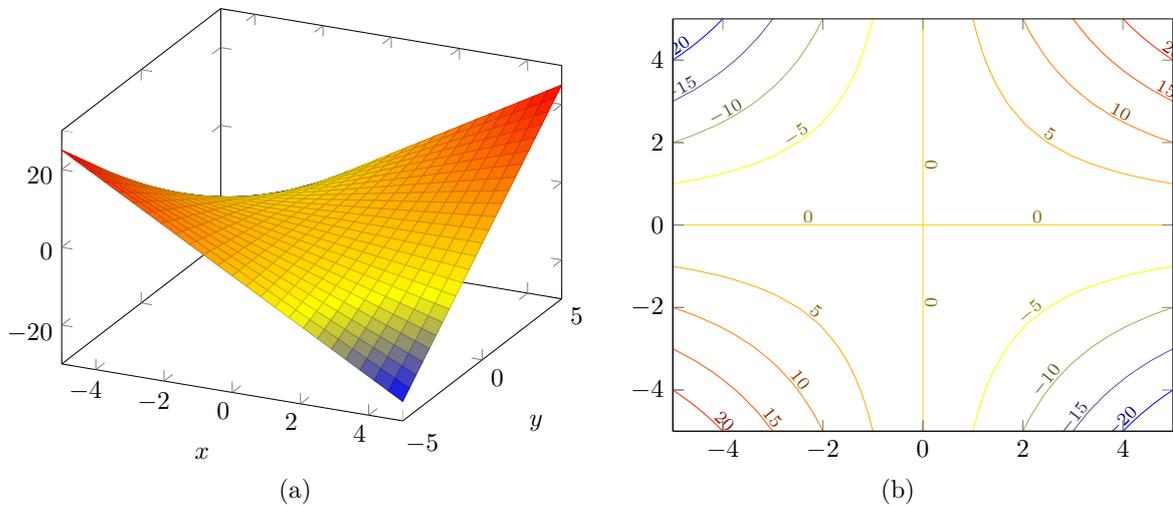


Figura 3.2: Produto entre duas variáveis contínuas $f(x, y) = x \cdot y$: (a) gráfico e (b) curvas de nível.

É possível usar técnicas de linearização para transformar uma região não-convexa em convexa. Para modelar não-linearidades envolvendo variáveis contínuas Floudas (1995c) enumera três abordagens dependendo do tipo de função não-linear: (i) funções não-convexas separáveis, (ii) procedimentos de convexificação e (iii) não-convexidades. Se a função não-convexa for separável é possível usar a primeira abordagem e substituí-la por funções lineares por partes. Procedimentos de convexificação são adequados para algumas classes de funções apenas. Uma das técnicas de convexificação mais conhecidas na literatura é substituir a restrição contendo termos bilineares por sua relaxação convexa de McCormick (LIBERTI; PANTELIDES, 2005). Se há no modelo uma função não-linear não-separável e que não pode ser convexificada, então algoritmos de otimização global precisam ser usados para encontrar o ótimo global com garantia teórica.

Geralmente encontrar o ótimo global de formulações não-lineares é um procedimento difícil e demorado. De acordo com (FLOUDAS, 1995c) linearizar termos bilineares de variáveis contínuas afeta a eficiência do algoritmo e também a qualidade da solução.

Geißler et al. (2012) destaca que manter a linearidade do problema viabiliza o uso de métodos de solução do mundo da programação linear, promissores por serem hoje em dia muito maduros. Tais algoritmos são rápidos, robustos e capazes de resolver problemas com até milhões de variáveis. Lundell (2009) apud Geißler et al. (2012) questiona se reformular um problema com o propósito de resolvê-lo por métodos mais conhecidos é de fato uma saída apropriada: "Se você tem um martelo, tudo é um prego".

Quesada e Grossmann (1995) estudam o uso de técnicas de linearização em problemas de programação matemática envolvendo mistura e separação de fluxos multicomponentes. Os autores mostram que tais processos não podem ser rigorosamente reformulados por programas lineares e que, conseqüentemente, reclamam técnicas de otimização global.

Não obstante, Kallrath (2002) aponta que o problema da programação é fruto de um problema de planejamento preparado para um horizonte de médio a longo prazo. Todos os dados de entrada e objetivos da programação têm origem no planejamento. O propósito da programação é transformar o plano em um programa útil e aplicável à todas as operações dentro de um horizonte de tempo de alguns dias. Neste sentido, programar a produção é mais uma questão de viabilidade do que de otimalidade.

No problema 1 da Figura 3.1 os *pools* são os tanques de carga. No problema 2 da Figura 3.1 os *pools* são unidades de processamento contínuo e tanques. No problema 3 da Figura 3.1 os *pools* são os tanques de *blending*. A restrição não-linear garante que sempre que uma operação de transferência aconteça, sua composição esteja de acordo com a composição do *pool*.

O problema do suprimento de petróleo é o mais estudado na literatura, seguido do problema do *blending*. Segundo Jia et al. (2003) o problema intermediário, otimização da programação nas unidades de processo, é o menos explorado por sua grande complexidade: lida com diferentes tipos processos contínuos envolvendo mistura de fluxos multicomponentes.

Essa dissertação se dedica ao problema da programação do suprimento de petróleo, um sistema multi-estágio constituído por navios, tanques de armazenamento, tanques de carga e UDAs, no qual há mistura e separação de bateladas. Portanto

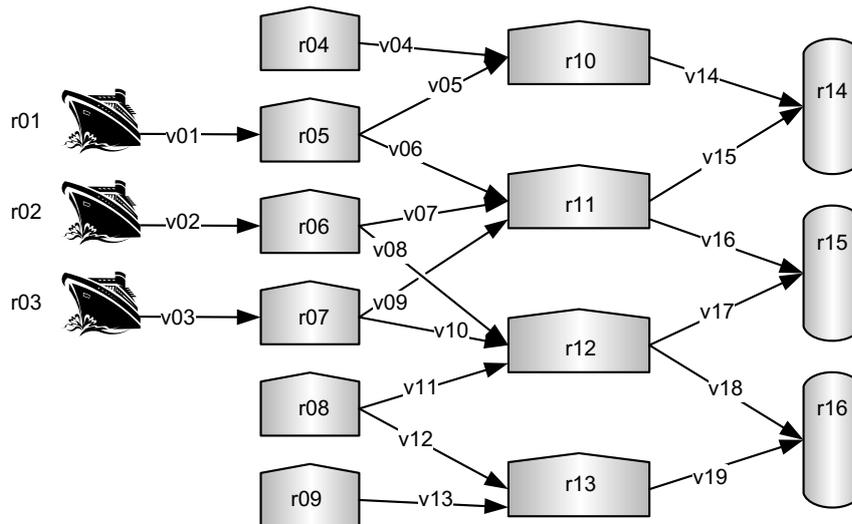


Figura 3.3: Arranjo ilustrativo do sistema de suprimento de petróleo.

segundo a classificação de Maravelias (2012), o sistema de suprimento de petróleo é um processo em rede. Embora cada sistema tenha suas particularidades, o problema genérico do suprimento de petróleo pode ser ilustrado pela Figura 3.3.

Programação do suprimento de petróleo

Durante um horizonte de tempo conhecido, navios carregados com petróleo chegam aos arredores da estação de ancoragem da refinaria. Caso ela esteja ocupada, o navio aguarda até que o precedente seja completamente descarregado. Lá o petróleo é descarregado para tanques de armazenamento.

Depois disso, eles são transferidos dos tanques de armazenamento para tanques de carga, que funcionam como *buffers* para produzir uma determinada mistura de composição conhecida na saída da UDA. Essa especificação é definida *a priori*, no nível de planejamento. As misturas de petróleo nos tanques de carga são, posteriormente, enviadas às UDAs.

Cada navio traz à refinaria um tipo de petróleo com propriedades chave conhecidas. Dessa forma, o primeiro estágio do processo sequencial não envolve fluxos multicomponentes. Uma simplificação da literatura é considerar os tanques de armazenamento dedicados, ou seja, também não há mistura de petróleo nos tanques de armazenamento. Então a mistura é feita nos tanques de carga, que armazenam mis-

turas com diferentes faixas de propriedade chave. Na prática, o planejamento usa uma representação agregada da tancagem.

A propriedade chave mais usada para caracterizar as misturas nos estudos de caso da literatura é a fração de enxofre, mas outras propriedades também podem ser usadas.

O problema da programação do suprimento de petróleo é definido abaixo.

Dados:

- a topologia da planta;
- a hora de chegada de cada navio;
- a capacidade dos tanques;
- o limite das vazões de transferência entre equipamentos;
- as propriedades dos petróleos;
- a quantidade de cada petróleo nos navios e tanques (inventário inicial);
- a faixa desejada para fração de componente chave em cada estágio ou recurso;
- o horizonte de tempo considerado;
- a meta de produção definida pelo planejamento.

Deseja-se determinar

- tempo de espera de cada navio no mar;
- tempo de descarregamento de cada navio;
- vazão de descarregamento de petróleo dos navios para os tanques de armazenamento;
- vazão de transferência de petróleo dos tanques de armazenamento para tanques de carga;
- vazão de transferência de petróleo dos tanques de carga para as [UDAs](#);

- sequência de alimentação de cada mistura de petróleo à cada UDA;

que satisfaçam as restrições de capacidade dos equipamentos, de composição e demanda, maximizando a valor marginal total das misturas de petróleo destiladas. O custo total do processo é calculado por Lee et al. (1996) como a soma dos custos de espera dos navios no mar, custo de descarregamento dos navios, custo de inventário nos tanques de armazenamento e carga e custo de transição das operações de alimentação nas UDAs.

Na prática as regras operacionais variam dependendo da refinaria. As regras operacionais adotadas nos consagrados estudos de caso de Lee et al. (1996) são que apenas um navio pode descarregar de cada vez, os tanques de carga não podem receber e enviar correntes simultaneamente, cada tanque de carga pode alimentar somente uma UDA por vez, cada UDA é alimentada por somente um tanque de carga por vez e as UDAs devem operar continuamente durante todo o horizonte de tempo.

Uma prática comum que Lee et al. (1996) não considera em seus estudos de caso é deixar petróleo decantar por algum tempo (*brine settling time*) nos tanques de armazenamento entre recebimento e envio, com o objetivo de evitar que água salgada seja injetada nas UDAs, período este chamado de tempo de preparação.

Nas refinarias brasileiras uma unidade de destilação atmosférica (UDA) pode ser alimentada por até dois tanques de carga simultaneamente. Conseqüentemente, um tanque de carga pode alimentar até duas UDAs simultaneamente. Essas regras operacionais geram várias mudanças no modelo e na forma de resolver o problema.

A Figura 3.4 mostra um linha do tempo com alguns dos trabalhos mais expressivos sobre programação do suprimento de petróleo realizados nos últimos trinta anos. A Tabela 3.1 resume algumas características dos problemas estudados nos trabalhos exibidos na Figura 3.4, discutidos na sequência.

Lee et al. (1996), um dos trabalhos pioneiros na área, desenvolveu um modelo MILP de tempo discreto para otimizar a programação do suprimento de petróleo. O autor aplicou sua formulação à quatro estudos de caso, entre os quais um problema de dimensões industriais, um estudo de caso conhecido como Lee4. O estudo de caso Lee4 possui três navios, seis tanques de armazenamento, quatro tanques de carga, três UDAs, um horizonte de tempo de 15 h e oito tipos de petróleo cru. Lee et

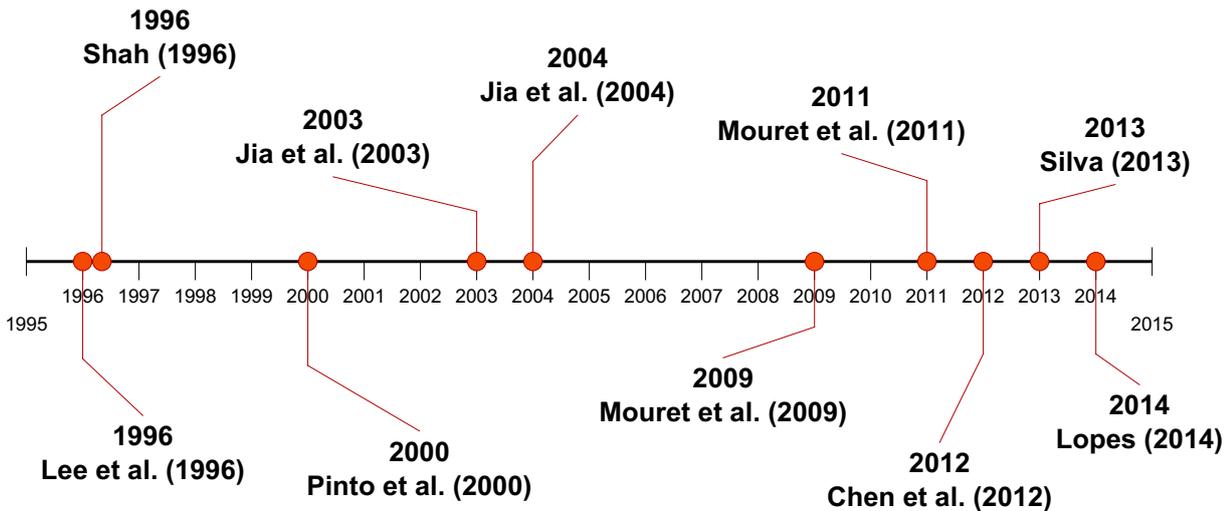


Figura 3.4: Linha do tempo com alguns trabalhos sobre programação do suprimento de petróleo.

al. (1996) obtiveram a solução ótima ao discretizar o horizonte de tempo de oito dias em quinze intervalos globais de tempo. O modelo foi implementado na plataforma **GAMS** e resolvido com o *solver* **OSL**. A principal limitação do modelo de Lee et al. (1996) é que, para evitar a geração de um problema **MINLP**, a restrição que garante consistência de composição dentro dos recursos e nas operações de transferência foi negligenciada. O modelo levou mais de 1 h de tempo de **CPU** para solucionar o problema.

Outro trabalho pioneiro é o de Shah (1996). Com a finalidade de gerar um programa para um mês o autor decompôs o problema em dois: a jusante, que lida com as operações envolvendo tanques de carga e **UDAs**, e a montante, que lida com operações envolvendo navios e tanques de armazenamento. Dois problemas **MILP** de tempo discreto são apresentadas, uma para cada problema. Os resultados de sequenciamento e cronometragem das operações de alimentação nos tanques de carga a jusante servem de dados de entrada para o problema a montante. O problema a jusante é resolvido com minimização de saltos de composição na alimentação das **UDAs**, e o problema a montante é resolvido apenas para garantir a viabilidade do programa. Caso este não seja factível, executa-se um processo iterativo entre os dois modelos até que um programa viável seja obtido. Shah (1996) aplicou sua formulação

Tabela 3.1: Características dos problemas de programação do suprimento de petróleo estudados na literatura consultada.

Referência	Tempo	Eventos	Estudo de caso	Solução	Tempo de CPU
Lee et al. (1996)	discreto	intervalos globais	Lee1 a Lee4	OSL	0,3 min; 5,0 min; 18,2 min; 72,9 min
Shah (1996)	discreto	intervalos globais	$H = 1$ mês, $ C = 4$, $ R_S = 7$, $ R_C = 8$, $ R_D = 2$	Xpress	minutos
Pinto et al. (2000)	contínuo	slots	$H = 112$ h, $ C = 3$, $ R_P = 4$, $ R_C = 6$, $ R_D = 1$	OSL	-
Jia et al. (2003)	contínuo	pontos de evento	Lee1 a Lee4	DICOPT	6 h (Lee4)
Jia e Ierapetritou (2004)	contínuo	pontos de evento	Lee1 a Lee4	CPLEX	17,1 s; 4,9 s; 1,5 s; 7,9 s
Mouret et al. (2009)	contínuo	slots	Lee1 a Lee4	Xpress, CONOPT	0 s; 1 s; 2 s; 3 s
Mouret et al. (2011)	contínuo	slots	Lee1 a Lee4	CPLEX	1,16 s; 3,64 s; 1,74 s; 1,58 s
Silva (2013)	discreto	intervalos globais	$H = 12$ dias, $ C = 3$, $ R_V = 3$, $ R_C = 3$, $ R_D = 2$	DICOPT	18,7 min
Lopes (2014)	contínuo	slots	$H = 12$ h, $ C = 7$, $ R_V = 3$, $ R_S = 3$, $ R_C = 2$, $ R_M = 2$, $ R_D = 2$	DICOPT	98,16 s

a um estudo de caso com quatro tipos de petróleo, sete tanques de armazenamento, oito tanques de carga e duas UDAs, com horizonte de tempo discretizado em intervalos de um dia e resolveu o modelo com C++ *XPRESS Modelling and Optimization System*. Shah (1996) reporta que ambos modelos foram solucionados em poucos minutos de tempo de CPU.

Pinto et al. (2000) propuseram uma formulação MILP de tempo contínuo com *slots* de tempo, e aplicaram o modelo à um caso real da Refinaria Henrique Lage ou Refinaria do Vale do Paraíba (REVAP) em São José dos Campos (SP), um sistema com quatro parcelas de petróleo contendo três tipos de petróleo no total, seis tanques de carga com *brine settling time* de 24 horas e uma UDA. O lucro máximo do processo para um horizonte de tempo de 112 horas foi obtido com 19 *slots* de tempo. O modelo foi implementado em GAMS e resolvido com o *solver* OSL. Os autores não reportam o tempo de CPU necessário para resolver o problema.

Jia et al. (2003) apresentaram uma formulação MILP de tempo contínuo com eventos específicos à unidade baseada na representação STN. Para fins de comparação, os autores usaram a formulação proposta para solucionar os estudos de caso de Lee et al. (1996). As restrições contendo termos bilineares foram negligenciadas para obter uma comparação legítima. O problema MILP foi implementado em GAMS e resolvido com CPLEX 7.5. A formulação de tempo contínuo proposta por Jia et al. (2003) deu origem a modelos sensivelmente menores que a formulação de tempo discreto de Lee et al. (1996). Posteriormente, restrições com termos bilineares foram incluídas na modelagem dos estudos de caso de Lee et al. (1996), dando origem a modelos MINLP que foram implementados em GAMS. Jia et al. (2003) reporta que para solucionar o estudo de caso de dimensão industrial Lee4 foram necessárias 6 h de tempo de CPU, tempo atribuído principalmente à soluções de sub-problemas MILP.

Seguindo a mesma diretriz, Jia e Ierapetritou (2004) propuseram uma formulação MILP de tempo contínuo com eventos específicos à unidade para cada um dos subproblemas da refinaria. No problema 2 da Figura 3.1 um processo representativo de fabricação de óleo lubrificante foi analisado. Não há nenhum tipo de integração entre os modelos e cada um deles é resolvido e comparado com estudos de casos específicos de cada subproblema. Para modelar o suprimento de petróleo Jia e Ierapetritou (2004) evita não-linearidades ao especificar faixas de propriedade-chave em

cada tanque de carga e operação de transferência. Os estudos de caso de Lee et al. (1996) foram resolvidos com GAMS/CPLEX 7.0. A formulação proposta por Jia e Ierapetritou (2004) solucionou o estudo de caso Lee4 em 7,87 s de tempo de CPU para uma corrida.

Mouret et al. (2009) apresentaram uma nova formulação MINLP de tempo contínuo com *slots* de prioridade, cuja quantidade precisa ser postulada *a priori*. Cada *slot* de prioridade é atribuído a exatamente uma operação de transferência específica dentre as da Figura 3.3. Essa nova formulação representa o programa do suprimento de petróleo por uma única sequência de operações de transferência – ou seja, no máximo uma operação é atribuída a cada *slot* – e foi batizada de modelo de *sequenciamento único de operações* (SOS). Para resolver o MINLP, os autores propuseram uma estratégia de decomposição em duas etapas. Primeiramente o problema MINLP é decomposto em dois problemas, um MILP e outro NLP. A única diferença entre eles é a existência da restrição não-linear com as regras de mistura. O modelo MILP é solucionado e depois, fixada a solução inteira do MILP, trata-se o problema de NLP associado. A formulação foi aplicada aos estudos de caso de Lee et al. (1996), implementados em GAMS e resolvidos com Xpress 17.1 e CONOPT 3. A formulação de Mouret et al. (2009) resolveu o estudo de caso Lee4 em 3 s de tempo de CPU para uma corrida.

Mouret et al. (2011) continuaram o trabalho de Mouret et al. (2009) e desenvolveram uma formulação MINLP que representa o programa do suprimento de petróleo por múltiplas sequências de operações de transferência, o modelo MOS. É admitido que várias operações sejam atribuídas ao mesmo *slot* de prioridade, desde que sejam operações sobrepostas ou *overlapping*. Os autores aplicaram o modelo aos estudos de caso de Lee et al. (1996) e usaram a mesma estratégia de solução de Mouret et al. (2009). O desempenho do modelo MOS na primeira etapa da decomposição, o MILP, foi investigado com o *solver* com GAMS/CPLEX 11. Segundo Chen et al. (2012), o modelo MOS pode ser visto como um extensão dos modelos de grades de tempo específicas à unidade: enquanto nos modelos com grades de tempo específicas à unidade há uma sequência de eventos para cada unidade, o modelo MOS combina essa sequência e a cronometragem de operações em diferentes unidades. A formulação de Mouret et al. (2009) resolveu o estudo de caso Lee4 em 4,17 s de tempo de CPU

para uma corrida.

Chen et al. (2012) apresentaram uma análise comparativa das representações de grade de tempo múltipla mais bem-sucedidas da literatura para programação do suprimento de petróleo: o modelo *event-based* de Jia et al. (2003), o modelo *unit-slot* de Hu e Zhu (2007) e o modelo MOS de Mouret et al. (2011). Os autores ressaltaram os pontos positivos e negativos de cada formulação e as aplicaram aos quatro estudos de caso de Lee et al. (1996), usando dois tipos de função objetivo: maximização das vendas e minimização dos custos operacionais. Os modelos foram implementados em GAMS e resolvidos por estratégia de decomposição MILP-NLP com os solvers CPLEX e CONOPT 3.14. Chen et al. (2012) concluíram que o modelo MOS é o mais promissor.

Silva (2013) estendeu a formulação de tempo discreto de Lee et al. (1996) para aferir movimentações do componente chave enxofre no sistema. O autor formulou dois modelos a fim de investigar o efeito da restrição não-linear: um modelo MINLP com restrição contendo termos bilineares para eliminação de discrepâncias de composição, e um modelo MILP sem essa restrição e sem técnicas de linearização, assim como no trabalho de Lee et al. (1996). Os modelos foram aplicados aos estudos de caso de Lee et al. (1996) e Jia et al. (2003) com o horizonte de tempo discretizado em intervalos de um dia. Os modelos MILP e MINLP foram implementados em GAMS e resolvidos com os solvers CPLEX e DICOPT, respectivamente. Silva (2013) concluiu que a restrição que elimina discrepância de composição é imprescindível para modelar corretamente problemas reais de programação do suprimento de petróleo. Silva (2013) solucionou um estudo de caso com três navios, três tanques de carga, duas UDAs e um horizonte de tempo de 12 dias em 1.124,77 s de tempo de CPU com o solver DICOPT para uma corrida.

Lopes (2014) agregou ao modelo MOS novos elementos matemáticos para incluir nas regras operacionais do problema a operação polegada. Operação polegada ou *temporary injection of multiple tanks* é o nome dado à operação de alimentação simultânea de uma UDA por dois tanques contendo diferentes misturas de petróleo. A finalidade da operação polegada é minimizar os custos de transição ou *changeover* correspondente ao período de tempo em que a coluna fica em regime transiente, até se adaptar às novas condições de composição. Lopes (2014) formulou dois modelos –

um com operação polegada e outro sem – para um sistema com três navios, três tanques de armazenamento dedicados, dois tanques de carga, dois misturadores, duas UDAs e um horizonte de tempo de 12 h e comparou os resultados. Os modelos foram implementados em GAMS e resolvidos com o *solver* local DICOPT. Lopes (2014) solucionou seu estudo de caso em até 98,16 s de tempo de CPU para uma corrida.

PROBLEMA REAL DE UMA REFINARIA BRASILEIRA

A Figura 4.1 mostra o sistema que se deseja modelar. Ele é composto por parcelas de petróleo R_P , tanques de carga R_C , misturadores R_M e UDAs R_D . Não há recursos do tipo navios R_V nem tanques de armazenamento R_S . O petróleo chega em parcelas R_P por oleoduto direto para os tanques de carga R_C . Há trinta e seis tipo de petróleo C no sistema. As propriedades e valor marginal de cada petróleo podem ser encontradas no Apêndice A.

Para modelar a programação do suprimento de petróleo em um sistema de dimensão industrial com regras operacionais específicas da indústria brasileira usou-se a formulação proposta por Mouret et al. (2011), chamada pelos autores de *sequenciamento múltiplo de operações* (MOS). Mouret et al. (2011) aplicam sua formulação aos estudos de caso de Lee et al. (1996), problemas de pequeno porte também chamados de problemas brinquedo (*toy problems*).

Conforme discutido anteriormente modelos de programação da produção de sistemas reais são complexos, possuem dezenas de milhares de variáveis e equações e raramente podem ser resolvidos até a otimalidade. Lee et al. (1996) faz várias simplificações em seus estudos de caso e, ainda assim, eles dão origem à programas difíceis de solucionar, que precisam de tempos computacionais razoáveis para serem resolvidos. Por exemplo, Lee et al. (1996) assumem em seus estudos de caso que apenas um tanque pode alimentar uma UDA de cada vez. Essa premissa simplifica consideravelmente o problema de programação, pois evita a possibilidade de acontecimento de operações sobrepostas na saída dos tanques de carga e, por conseguinte, que correntes se misturem antes de serem enviadas às UDAs.

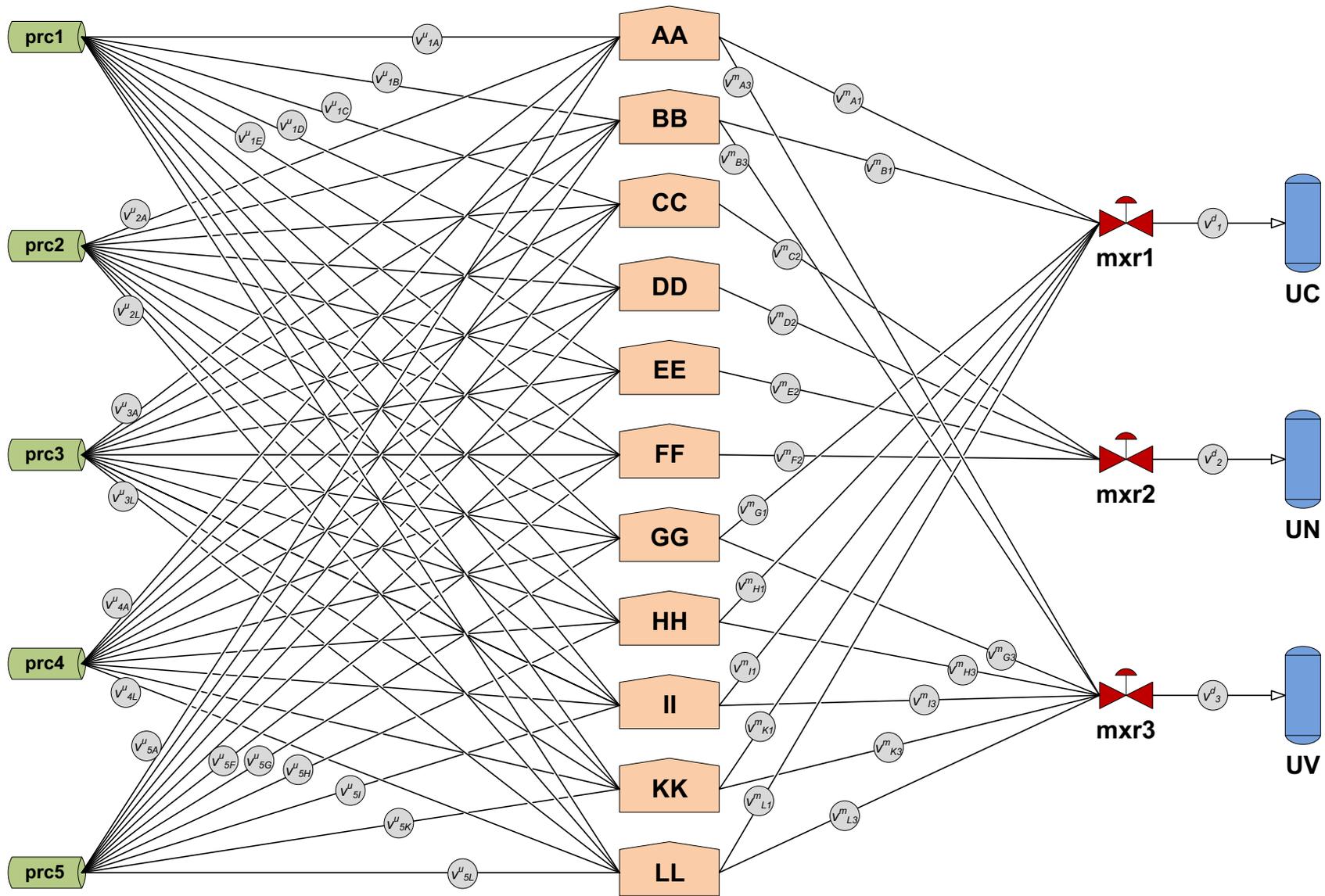


Figura 4.1: Topologia da refinaria real modelada.

As regras operacionais usadas por Lee et al. (1996) em seus estudos de caso são exibidas na Tabela 4.1.

Tabela 4.1: Regras operacionais adotadas nos estudos de caso de Lee et al. (1996).

-
- (i) apenas um navio pode descarregar de cada vez;
 - (ii) um navio pode descarregar para um único tanque de armazenamento;
 - (iii) não são permitidas operações simultâneas de entrada e saída nos tanques (operação pulmão);
 - (iv) cada tanque de carga pode alimentar somente uma UDA por vez;
 - (v) cada UDA é alimentada por somente um tanque de carga por vez;
 - (vi) as UDAs devem operar continuamente durante todo o horizonte de tempo;
 - (vii) não há tempo de preparação.
-

Sistemas produtivos reais, além de maiores, seguem regras operacionais mais elaboradas. A principal diferença entre as regras operacionais simplificadas da Tabela 4.1 e as praticadas em refinarias reais no Brasil é que uma UDA pode ser alimentada por até dois tanques de carga R_C simultaneamente. Conseqüentemente, os tanques de carga R_C podem alimentar até duas UDAs ao mesmo tempo.

É importante mencionar que essa não é a única diferença entre as regras operacionais simplificadas da literatura e as praticadas no Brasil. Outras regras operacionais adotadas na indústria brasileira são a operação polegada e o alinhamento concorrente, que não fazem parte do escopo desse trabalho.

A possibilidade de ocorrência de duas correntes de saída em um tanque de carga exige que declarações do tipo se-então-senão (*if-then-else*) sejam incluídas no modelo. Declarações do tipo se-então-senão são escritas em programação inteirista como restrições do tipo *Big-M*. Esse tipo de restrição deteriora a performance computacional do modelo, pois gera relaxações lineares menos justas. Além disso, quando um tanque de carga pode enviar mais de uma corrente para a destilação simultaneamente faz-se necessário modelar a mistura desses dois fluxos, que são fluxos multicomponentes. Adiciona-se portanto mais um nível de complexidade às regras operacionais simplificadas da Tabela 4.1. Para modelar mistura de fluxos multicomponentes é preciso incluir novos recursos na modelagem e isso agrega um novo

estágio ao sistema. A tabela 4.2 exibe as regras operacionais do sistema real modelado nessa dissertação. Entende-se por regras operacionais o modo de operação do sistema produtivo.

Tabela 4.2: Regras operacionais adotadas na indústria petroquímica brasileira.

-
- (i) parcelas devem ser descarregadas sem interrupções e de acordo com o programa;
 - (ii) uma parcela pode ser descarregada para vários tanques;
 - (iii) não são permitidas operações simultâneas de entrada e saída nos tanques de carga (operação pulmão);
 - (iv) cada tanque de carga pode alimentar até duas UDAs simultaneamente;
 - (v) cada UDA é alimentada por até dois tanques de carga simultaneamente;
 - (vi) as UDAs devem operar continuamente durante todo o horizonte de tempo.
 - (vii) há tempo de preparação de 24 h nos tanques de carga;
 - (viii) limite de duração mínima de operações.
-

Em todos os trabalhos da literatura elucidados no Capítulo 3.2 as misturas de petróleo cru com especificações desejadas são preparadas nos tanques de carga R_C , antes de alimentar a destilação. No sistema estudado neste trabalho as restrições de qualidade são impostas apenas nas operações de destilação W_D . A mistura é preparada em linha ao alimentar a destilação como resultado das movimentações de até dois tanques. Dessa forma o tanque não precisa se manter dentro das especificações ao longo do horizonte de tempo. Essa mudança que gera flexibilidade operacional e, além disso, tem de grande impacto na modelagem e forma de resolver o problema.

Na etapa MILP a restrição com termos bilineares é negligenciada e há, portanto, discrepância de composição entre as misturas nos tanques de carga e em suas respectivas correntes de saída. Essa discrepância de composição precisa estar dentro de uma amplitude limitada para que a etapa NLP retorne uma solução viável e, por isso, a etapa NLP retorna menos soluções viáveis do que quando as restrições de qualidade são impostas nos tanques de carga R_C .

Parcelas R_P com volume de petróleo $c \in C$ conhecido chegam em instantes conhecidos. A vazão de descarregamento das parcelas R_P é constante e, portanto,

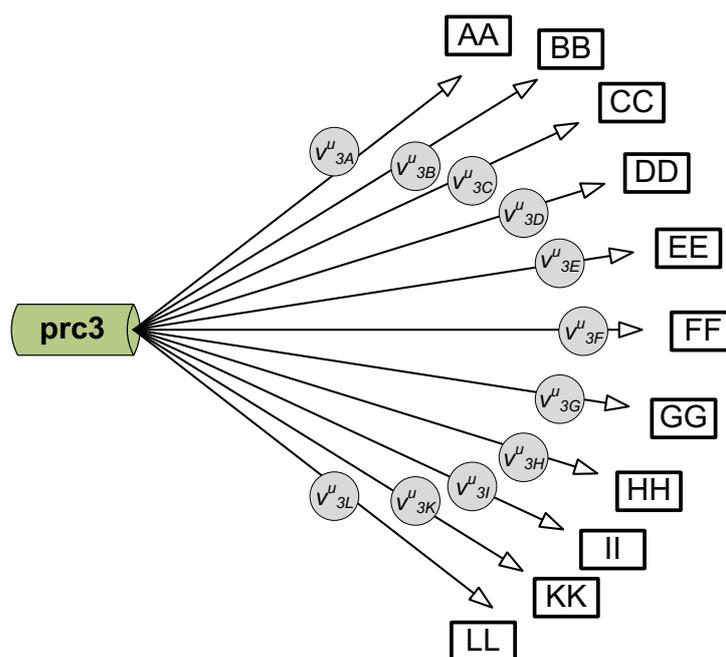


Figura 4.2: Operações de saída da parcela 3, O_{prc3} .

também é conhecido o instante em que elas são completamente descarregadas. As parcelas R_P podem descarregar para qualquer tanque e podem, inclusive ser enviadas à mais de um tanque durante o horizonte de tempo. Observa-se, portanto, conectividade total entre recursos no primeiro estágio W_U . A Figura 4.2 ilustra a nomenclatura das operações de descarregamento W_U .

Já no segundo estágio, as operações de mistura W_M , não há conectividade total entre equipamentos. Os tanques AA, BB, GG, HH, II, KK, LL podem enviar petróleo para as UDAs UC e UV. Os tanques CC, DD, EE e FF conectam-se apenas com a UDA UN. A Figura 4.3 ilustra a nomenclatura das operações de mistura W_M .

No terceiro estágio, as operações de destilação W_D , não há conectividade total entre equipamentos. Cada misturador $r \in R_M$ alimenta apenas uma UDA $r \in R_D$. A Figura 4.4 ilustra a nomenclatura das operações de destilação W_D .

As quatro restrições da Tabela 4.3 devem ser respeitadas durante todo o horizonte de tempo no sistema da Figura 4.1. Entende-se por restrições de processo as limitações relacionadas ao processo, que podem envolver recursos, operações e especificação de produtos finais.

As parcelas devem ser descarregadas sem interrupções, respeitando restri-

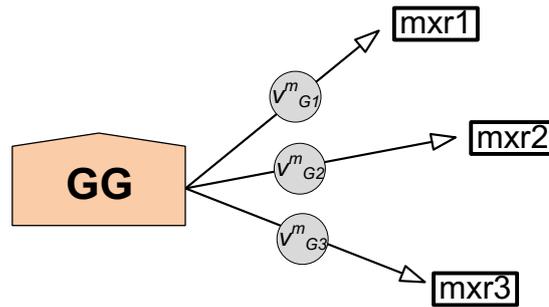


Figura 4.3: Operações de saída do tanque GG, O_{GG} .

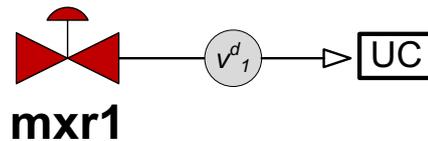


Figura 4.4: Operação de saída do misturador 1, O_{mxr1} .

ções de vazão e datas de chegada. As parcelas podem ser segmentadas e descarregadas para vários tanques de carga.

A capacidade de inventário é finita e o volume de petróleo nos tanques de carga R_C deve respeitar as restrições de lastro e capacidades operacionais máximas da refinaria, definidas no Apêndice A. Em nenhum instante ao longo do horizonte de tempo o volume armazenado no tanque de carga $r \in R_C$ pode exceder sua capacidade máxima, nem ficar abaixo do lastro, o volume mínimo permitido.

Não há restrições de demanda, apenas de vazão mínima e máxima para todas as operações de descarregamento e destilação $W_U \cup W_D$. A vazão de descarregamento de parcelas W_U varia de $1.200 \text{ m}^3 \text{ h}^{-1}$ a $2.000 \text{ m}^3 \text{ h}^{-1}$. A vazão máxima de operações de destilação W_D é $567 \text{ m}^3 \text{ h}^{-1}$, $400 \text{ m}^3 \text{ h}^{-1}$ e $220,8 \text{ m}^3 \text{ h}^{-1}$ para as UDAs UC, UN e UV, respectivamente. Adotou-se para a vazão mínima nas operações de destilação W_D 90% da vazão máxima.

Nas operações de mistura W_M os limites de vazão mínima e máxima se aplicam à todas as operações de saída de um mesmo tanque $r \in R_C$, pois elas são movimentadas pela mesma bomba. A capacidade de bombeamento dos tanques varia de $100 \text{ m}^3 \text{ h}^{-1}$ a $800 \text{ m}^3 \text{ h}^{-1}$. Para fins de modelagem, nos tanques de carga R_C com apenas uma corrente de saída – CC, DD e FF – tais limites são impostos por operação. Já nos tanques de carga com duas correntes de saída R_{C2} – AA, BB, HH, II, LL –

Tabela 4.3: Restrições que devem ser respeitadas ao longo de todo o horizonte de programação na refinaria real modelada.

-
- (a) parcelas R_P chegam pela mesma tubulação e devem ser descarregadas uma de cada vez;
 - (b) capacidade $\overline{L_{ir}^t}$ e lastro $\underline{L_{ir}^t}$ dos tanques de carga R_C ;
 - (c) limites de vazão em todas as operações W ;
 - (e) restrições de acidez nas operações de destilação W_D .
-

os limites são impostos por recurso.

Operações de descarregamento W_U não devem durar menos de 3 h, operações de mistura W_M não devem durar menos de 24 h e operações de destilação W_D não devem durar menos de 24 h. Para modelar essa característica do sistema uma restrição de duração mínima foi adicionada.

O limite inferior de volume total transferido em uma operação W , $\underline{V_v^t}$, foi calculado a partir dos valores de vazão mínima $\underline{FR_v}$ e duração mínima $\underline{Dur_v}$.

Usou-se como limite superior de volume total transferido nas operações de descarregamento W_U o próprio volume inicial da parcela. Dessa maneira uma parcela pode ser inteiramente descarregada em um *slot* de prioridade. Nas operações de mistura W_M o limite superior de volume total transferido foi a capacidade útil do tanque, ou seja, a diferença entre capacidade máxima e lastro. Sendo assim, um tanque pode enviar todo seu volume operacional para a destilação em apenas um *slot* de prioridade. Nas operações de destilação W_D usou-se como limite superior no volume total o dobro da máxima capacidade útil nos tanques de carga R_C .

Deseja-se monitorar a acidez nas operações de destilação. Nos estudos de caso da literatura uma UDA pode ser alimentada por apenas uma corrente e as restrições de qualidade são impostas nas misturas de petróleo em inventário nos tanques de carga R_C . Na refinaria estudada as misturas de petróleo em inventário não precisam estar especificadas: as restrições de qualidade são impostas nas operações de alimentação das UDAs. Essa prática é uma consequência da possibilidade de alimentar uma UDA com até duas correntes simultâneas. Nas operações de destilação W_D o TAN não deve ultrapassar 1,30 mg_{KOH}/g.

Regras de negócio

As contribuições deste trabalho são:

- o modelo **sequenciamento múltiplo de operações** (MOS) proposto por Mouret et al. (2011) foi modificado para representar uma refinaria brasileira real, suas regras operacionais e restrições de processo, originando um problema de grande porte, sem precedentes na literatura consultada;
- para incluir a chegada de parcelas de petróleo por oleoduto e não por navios, cada parcela foi incluída na modelagem como um novo recurso;
- o descarregamento de uma parcela pode ser fragmentado em várias operações e, assim, uma parcela pode alimentar vários tanques de carga R_C ou um mesmo tanque mais de uma vez;
- como um tanque pode enviar até duas correntes simultâneas para a destilação, restrições de vazão nos tanques de carga com duas possíveis correntes de saída foram estabelecidas por recurso, e não por operação;
- quando há duas correntes de saída ativas em um tanque de carga as operações devem ter a mesma duração, e uma grade comum à essas duas operações foi considerada na modelagem;
- a restrição de qualidade ($TAN \leq 1,30 \text{ mg}_{\text{KOH}}/\text{g}$) é imposta nas operações de destilação W_D , e não há restrições de qualidade nas misturas de petróleo em inventário nos tanques de carga R_C ;
- já que uma **UDA** pode ser alimentada por até dois tanques de carga R_C , misturadores foram adicionados ao sistema para calcular a vazão e a composição na entrada das colunas;
- quando há duas operações W_M ativas na entrada das **UDAs** elas devem ter a mesma duração, e uma grade comum à essas duas operações foi considerada na modelagem;
- imposição de duração mínima nas operações.

Cenários otimizados

O uso de um modelo matemático para representar o comportamento do sistema e também os resultados obtidos dependem principalmente da qualidade dessa representação¹. O entendimento do problema é a etapa mais importante desse processo. Para testar a performance do modelo seis cenários diferentes foram testados no sistema da Figura 4.1.

Esses seis cenários são dados reais de uma refinaria brasileira, e diferem entre si em quantidade de parcelas programadas, instante de chegada e vazão das parcelas, tanques de carga R_C disponíveis e inventário inicial nos tanques de carga R_C . Os dados de entrada de cada cenário podem ser encontrados no Apêndice A.

A Tabela 4.4 exibe uma visão geral dos cenários um a seis, com informações de inventário inicial nos tanques de carga R_C em m^3 , inventário inicial útil nos tanques de carga R_C (inventário inicial menos o lastro) em m^3 , volume total das parcelas programadas em m^3 , número de acidez total das misturas de petróleo em cada recurso \overline{TAN}_0 em mg_{KOH}/g , mvalor marginal unitário das misturas de petróleo em cada recurso \overline{G}_0 em \$ e horizonte de tempo em dias. Observa-se na Tabela 4.4 que os cenários possuem dados de entrada bem diversos.

Tabela 4.4: Visão geral dos cenários otimizados.

cenário	Inv. inicial R_C	Inv. útil inicial R_C	Vol. parcelas	\overline{TAN}_0	\overline{G}_0	H (dias)
um	176.396,05	124.226,05	136.000,00	0,99	233,16	7
dois	175.063,03	122.893,03	112.000,00	0,87	227,44	7
três	223.385,29	161.785,29	122.123,00	0,48	220,82	10
quatro	243.825,04	182.225,04	64.830,00	0,51	226,45	7
cinco	278.620,89	217.020,89	84.000,00	0,66	233,10	6
seis	294.902,05	233.302,05	93.250,00	0,81	234,71	10

¹Em: *Process Systems Engineering: The Way Ahead*, <http://www.researchgate.net/profile/Roger_Sargent/publication/272157379_Process_Systems_Engineering_-_The_Way_Ahead/links/54dbe9ca0cf2a7769d944d2f.pdf>. Acesso em 13 de Junho de 2015.

O parâmetro $\overline{\text{TAN}}_0$ foi calculado por uma média ponderada, usando o volume inicial do recurso L_{0rc} como peso, conforme Equação 4.1. Por ser uma propriedade em base mássica, a densidade também foi usada para o cálculo do parâmetro $\overline{\text{TAN}}_0$.

$$\overline{\text{TAN}}_0 = \frac{\sum_r \sum_c x_{ck_1} x_{ck_2} L_{0rc}}{\sum_r \sum_c x_{ck_2} L_{0rc}}, \quad r \in R, c \in C, k_1 = \{\text{TAN}\}, k_2 = \{\rho\} \quad (4.1)$$

O valor médio da valor marginal das misturas de petróleo no sistema \overline{G}_0 foi calculado por uma média ponderada, usando o volume inicial do recurso L_{0rc} como peso, conforme Equação 4.2. Uma vez que o TAN é uma propriedade em base mássica, é necessário usar a densidade para calcular o parâmetro \overline{G}_0 .

$$\overline{G}_0 = \frac{\sum_r \sum_c G_c L_{0rc}}{\sum_r \sum_c L_{0rc}}, \quad r \in R, c \in C \quad (4.2)$$

Seja o volume disponível para destilação igual à soma do inventário útil inicial nos tanques de carga ($\sum_r (L_{0r}^t - \underline{L}_r^t), r \in R_C$) e o volume que chega no sistema pelas parcelas ($\sum_r L_{0r}^t, r \in R_P$). Para comparar o volume disponível para destilação entre os cenários é preciso levar em conta o horizonte de tempo. Seja o volume relativo disponível para destilação igual ao volume disponível para destilação dividido pelo horizonte de tempo em dias ($\frac{1}{H} \sum_{r_1} L_{0r_1}^t - \underline{L}_{r_1}^t + \sum_{r_2} L_{0r_2}^t, r_1 \in R_C, r_2 \in R_P$).

O volume relativo disponível para destilação é $37.175,150 \text{ m}^3 \text{ d}^{-1}$, $33.556,147 \text{ m}^3 \text{ d}^{-1}$, $28.390,829 \text{ m}^3 \text{ d}^{-1}$, $35.293,577 \text{ m}^3 \text{ d}^{-1}$, $50.170,148 \text{ m}^3 \text{ d}^{-1}$ e $32.655,205 \text{ m}^3 \text{ d}^{-1}$ para os cenários um dois, três, quatro, cinco e seis, respectivamente.

Os cenários que mais se assemelham entre si são os cenários um e dois. Eles têm o mesmo horizonte de tempo (7 dias), volumes relativos disponíveis para destilação semelhantes e valores de $\overline{\text{TAN}}_0$ próximos. Nesses cenários há apenas vinte e seis tipos de petróleo, em contraste com trinta e seis nos demais cenários e, por isso, as possibilidades de mistura são mais limitadas. O número de acidez total médio $\overline{\text{TAN}}_0$ é maior nos cenários um e dois. Por isso pode-se esperar que eles sejam os mais difíceis de solucionar.

Comparando os cenários um e dois com o quatro, que também tem horizonte de tempo de 7 dias, nota-se que os três se aproximam em termos de volume relativo disponível para destilação. A principal diferença entre o cenário quatro e os cenários um e dois é seu valor de $\overline{\text{TAN}}_0$, que é menor ($0,51 \text{ mg}_{\text{KOH}}/\text{g}$).

Os cenários dois e seis possuem horizonte de tempo diferentes e diferentes quantidades de petróleo disponíveis no sistema. No entanto eles se assemelham quanto ao volume relativo disponível para destilação e \overline{TAN}_0 .

Comparando os cenários três e seis, que têm um horizonte de 10 dias, nota-se que o cenário três tem um menor volume relativo disponível para destilação que o seis e um \overline{TAN}_0 menor. O volume relativo disponível para destilação é $28.390,829 \text{ m}^3 \text{ d}^{-1}$ e $32.655,205 \text{ m}^3 \text{ d}^{-1}$ nos cenários três e seis, respectivamente. Esses dois cenários são, portanto, bem diferentes.

O cenário cinco é o único para o qual o horizonte de tempo é de 6 dias. Ele possui \overline{TAN}_0 baixo e um volume relativo disponível para destilação muito superior ao de todos os outros cenários ($50.170,148 \text{ m}^3 \text{ d}^{-1}$). Pode-se esperar, portanto, que ao final do horizonte de tempo haja no cenário cinco o maior inventário útil disponível nos tanques de carga R_C quando comparado aos demais cenários.

Observa-se que o menor volume relativo disponível para destilação é o do cenário três, $28.390,829 \text{ m}^3 \text{ d}^{-1}$, e o maior é o do cenário cinco, $50.170,148 \text{ m}^3 \text{ d}^{-1}$.

A capacidade máxima de destilação do sistema é de $28.507,2 \text{ m}^3 \text{ d}^{-1}$. O único cenário cujo volume relativo disponível para destilação é menor que a capacidade máxima do sistema é o cenário três. Isso significa que no cenário três pelo menos uma das operações de destilação W_D ativas ao longo do horizonte de tempo não terá vazão máxima.

Cenário um

No cenário um há vinte e seis tipos de petróleo no sistema. Cinco parcelas estão programadas ao longo do horizonte de tempo de 168 h (7 dias). As parcelas contém apenas um tipo de petróleo e podem ser descarregadas para um dos oito tanques: AA, BB, CC, DD, FF, HH, II e LL. Os tanques EE, GG e KK não estão disponíveis neste cenário.

O volume de petróleo nas parcelas é 22.000 m^3 , 32.000 m^3 , 24.000 m^3 , 34.000 m^3 e 24.000 m^3 para $prc1$, $prc2$, $prc3$, $prc4$ e $prc5$, respectivamente. A vazão de descarregamento é $1.700 \text{ m}^3 \text{ h}^{-1}$ para todas as parcelas. A Figura 4.5 mostra o instante de chegada e final do descarregamento de cada parcela.

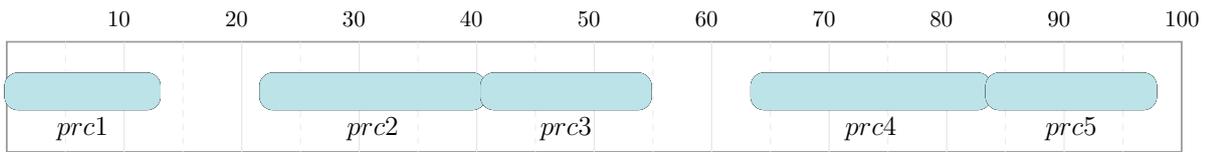


Figura 4.5: Dados de entrada sobre a programação das parcelas no cenário um com instantes de chegada e final do descarregamento de cada parcela em h.

O inventário útil inicial nos tanques de carga R_C (inventário inicial menos o lastro dos tanques) é de $124.226,05 \text{ m}^3$. A Figura 4.6 exibe o valor do TAN e da valor marginal unitário das misturas em cada recurso no cenário um. O valor médio da

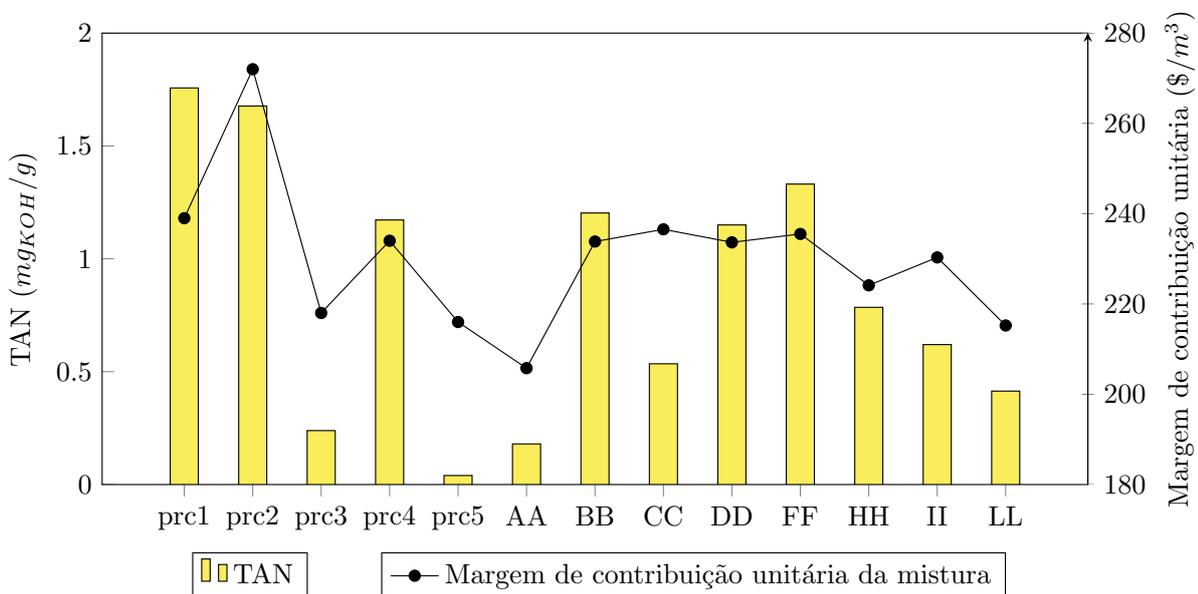


Figura 4.6: Valor marginal unitária e número de acidez total (TAN) das misturas em cada recurso para o cenário um.

propriedade-chave TAN em todos os recursos é $0,99 \text{ mg}_{\text{KOH}}/\text{g}$ e o valor médio da valor marginal das misturas de petróleo no sistema é $233,16 \$$.

Cenário dois

No cenário dois há também vinte e seis tipos de petróleo no sistema. Três parcelas estão programadas ao longo de um horizonte de tempo de 168 h (7 dias). As parcelas contém apenas um tipo de petróleo e podem ser descarregadas para os mesmos tanques do cenário um: AA, BB, CC, DD, FF, HH, II e LL. O inventário inicial

dos cenários um e dois são diferentes.

O volume de petróleo que chega nas parcelas é 24.000 m^3 , 58.000 m^3 e 30.000 m^3 para *prc1*, *prc2* e *prc3*, respectivamente. A vazão de descarregamento é $1.200 \text{ m}^3 \text{ h}^{-1}$ para a parcela *prc1* e $1.700 \text{ m}^3 \text{ h}^{-1}$ para as parcelas *prc2* e *prc3*. A Figura 4.7 mostra o instante de chegada e final do descarregamento de cada parcela.

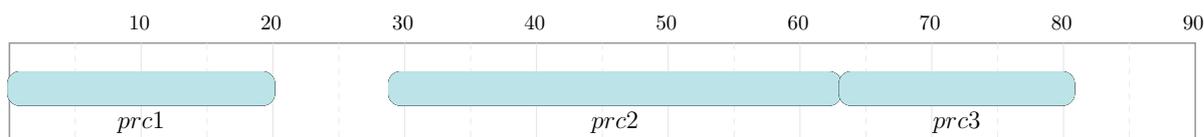


Figura 4.7: Dados de entrada sobre a programação das parcelas no cenário dois com instantes de chegada e final do descarregamento de cada parcela em h.

O inventário útil inicial nos tanques de carga R_C (inventário inicial menos o lastro dos tanques) é de $122.893,03 \text{ m}^3$. A Figura 4.8 exibe o valor do TAN e da valor marginal unitário das misturas em cada recurso no cenário dois. O valor médio da

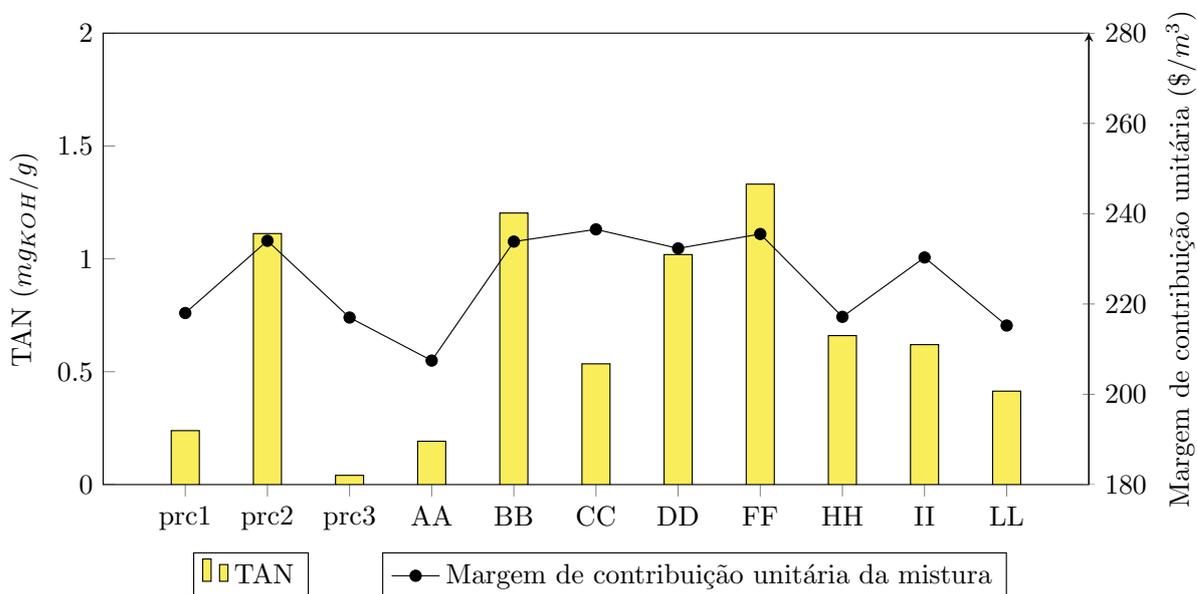


Figura 4.8: Valor marginal unitário e número de acidez total (TAN) das misturas em cada recurso para o cenário dois.

propriedade-chave TAN em todos os recursos é $0,87 \text{ mgKOH/g}$, e o valor médio da valor marginal das misturas de petróleo no sistema é $227,44 \text{ \$}$.

Cenário três

Os cenários três, quatro, cinco e seis são mais complexos, pois possuem trinta e seis tipos de petróleo no sistema. Além disso algumas parcelas possuem mais de um tipo de petróleo, e suas operações de descarregamento são fluxos multicomponentes. Conseqüentemente, os cenários três, quatro, cinco e seis dão origem a modelos com mais restrições não-lineares.

Os tanques disponíveis para a programação no cenário três são AA, DD, EE, FF, GG, HH, II, KK e LL . Há cinco parcelas programadas ao longo de um horizonte de tempo de 240 h (10 dias). Algumas parcelas contém mais de um tipo de petróleo.

O volume de petróleo que chega nas parcelas é 7.123 m^3 , 30.000 m^3 , 5.000 m^3 , 20.000 m^3 e 60.000 m^3 para *prc1*, *prc2*, *prc3*, *prc4* e *prc5*, respectivamente. A vazão de descarregamento é $1.600 \text{ m}^3 \text{ h}^{-1}$ para as parcelas *prc1*, *prc3*, *prc4* e *prc5* e $1.450 \text{ m}^3 \text{ h}^{-1}$ para a parcela *prc2*. A parcela *prc1* contém dois tipos de petróleo. A Figura 4.9 mostra o instante de chegada e final do descarregamento de cada parcela.

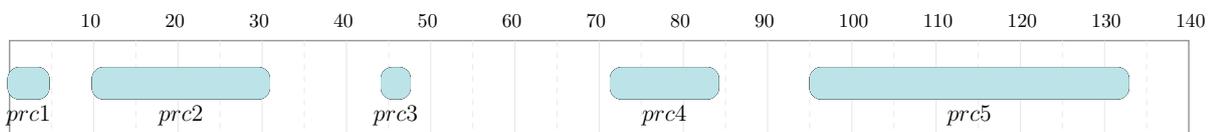


Figura 4.9: Dados de entrada sobre a programação das parcelas no cenário três com instantes de chegada e final do descarregamento de cada parcela em h.

O inventário útil inicial nos tanques de carga R_C (inventário inicial menos o lastro dos tanques) é de $161.785,29 \text{ m}^3$. A Figura 4.10 exibe o valor do TAN e da valor marginal unitário das misturas em cada recurso no cenário três. O valor médio da propriedade-chave TAN em todos os recursos é $0,48 \text{ mg}_{\text{KOH}}/\text{g}$, e o valor médio da valor marginal das misturas de petróleo no sistema é $220,82 \$$.

Cenário quatro

No cenário quatro há trinta e seis tipos de petróleo no sistema. Os tanques disponíveis para a programação no cenário quatro são AA, DD, EE, FF, GG, HH, II, KK e LL. Três parcelas estão programadas ao longo de um horizonte de tempo de 168 h (7 dias). Algumas parcelas contém mais de um tipo de petróleo.

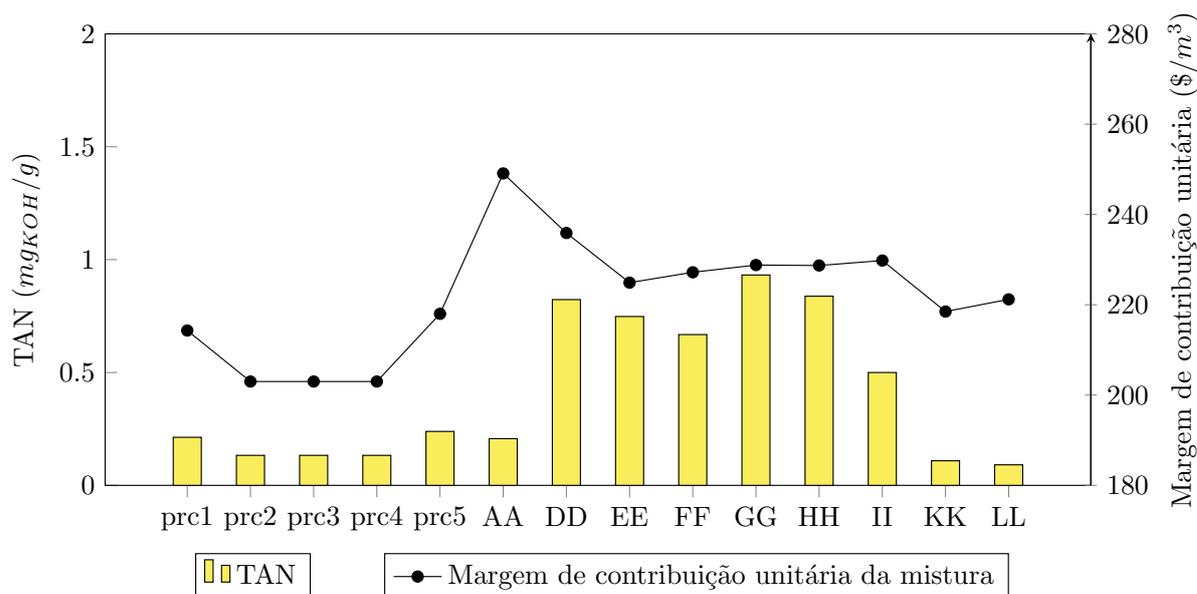


Figura 4.10: Valor marginal unitário e número de acidez total (TAN) das misturas em cada recurso para o cenário três.

O volume de petróleo que chega nas parcelas é 2.830 m^3 , 35.000 m^3 e 27.000 m^3 para *prc1*, *prc2* e *prc3*, respectivamente. A vazão de descarregamento é $1.600 \text{ m}^3 \text{ h}^{-1}$ para a parcela *prc1*, $1.750 \text{ m}^3 \text{ h}^{-1}$ para a parcela *prc2*, e $2.000 \text{ m}^3 \text{ h}^{-1}$ para a parcela *prc3*. A parcela *prc2* contém três tipos de petróleo. A Figura 4.11 mostra o instante de chegada e final do descarregamento de cada parcela.

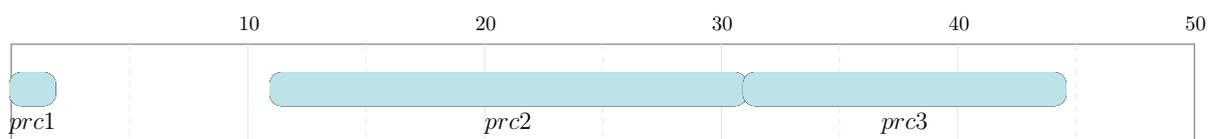


Figura 4.11: Dados de entrada sobre a programação das parcelas no cenário quatro com instantes de chegada e final do descarregamento de cada parcela em h.

O inventário útil inicial nos tanques de carga R_C (inventário inicial menos o lastro dos tanques) é de $182.225,04 \text{ m}^3$. A Figura 4.12 exhibe o valor do TAN e da valor marginal unitário das misturas em cada recurso no cenário quatro. O valor médio da propriedade-chave TAN em todos os recursos é $0,51 \text{ mg}_{\text{KOH}}/\text{g}$, e o valor médio da valor marginal das misturas de petróleo no sistema é $226,44 \$$.

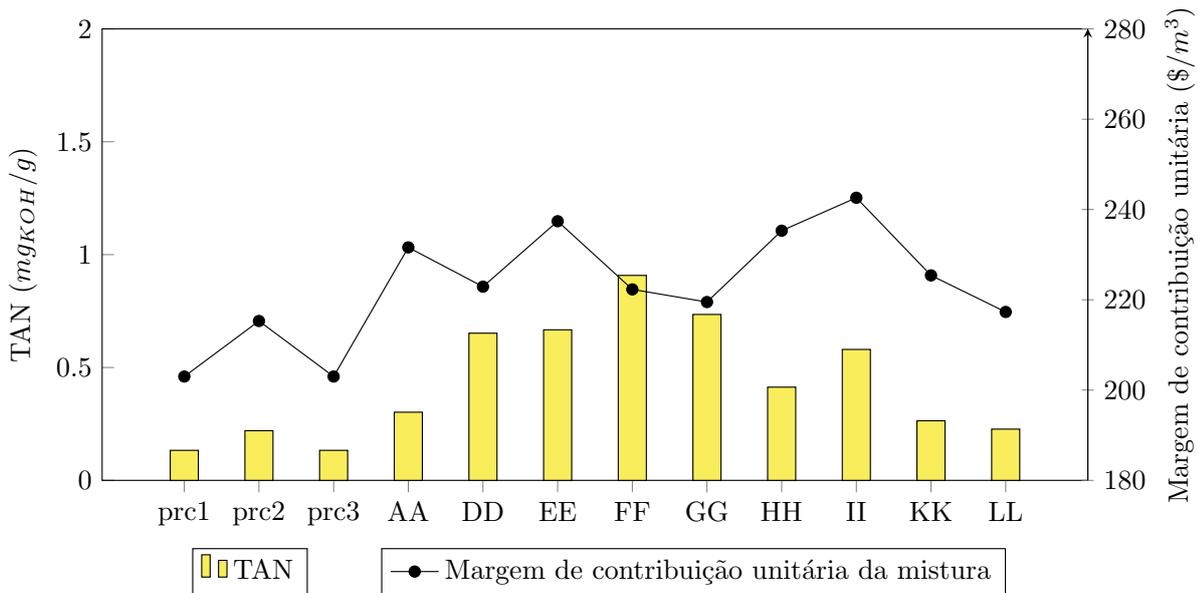


Figura 4.12: Valor marginal unitário e número de acidez total (TAN) das misturas em cada recurso para o cenário quatro.

Cenário cinco

No cenário cinco há trinta e seis tipos de petróleo no sistema. Os tanques disponíveis para a programação são AA, DD, EE, FF, GG, HH, II, KK e LL. Há quatro parcelas programadas num horizonte de tempo de 144 h (6 dias). Algumas parcelas contêm mais de um tipo de petróleo.

O volume de petróleo que chega nas parcelas é 34.000 m^3 , 10.000 m^3 , 15.000 m^3 e 25.000 m^3 para *prc1*, *prc2*, *prc3* e *prc4*, respectivamente. A vazão de descarregamento é $1.800 \text{ m}^3 \text{ h}^{-1}$ para as parcelas *prc1*, *prc3* e *prc4* e $1.300 \text{ m}^3 \text{ h}^{-1}$ para a parcela *prc2*. As parcelas *prc1* e *prc2* contêm dois tipos de petróleo. A Figura 4.13 mostra o instante de chegada e final do descarregamento de cada parcela.

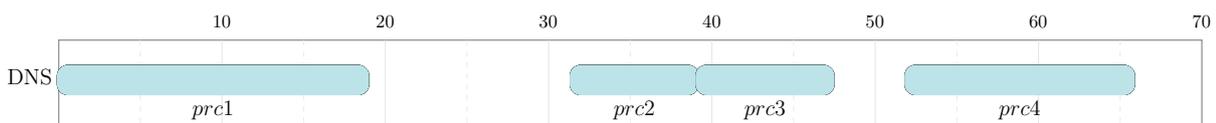


Figura 4.13: Dados de entrada sobre a programação das parcelas no cenário cinco com instantes de chegada e final do descarregamento de cada parcela em h.

O inventário útil inicial nos tanques de carga R_C (inventário inicial menos o

lastro dos tanques) é de 217.020,90 m³. A Figura 4.14 exibe o valor do TAN e da valor marginal unitário das misturas em cada recurso no cenário quatro.

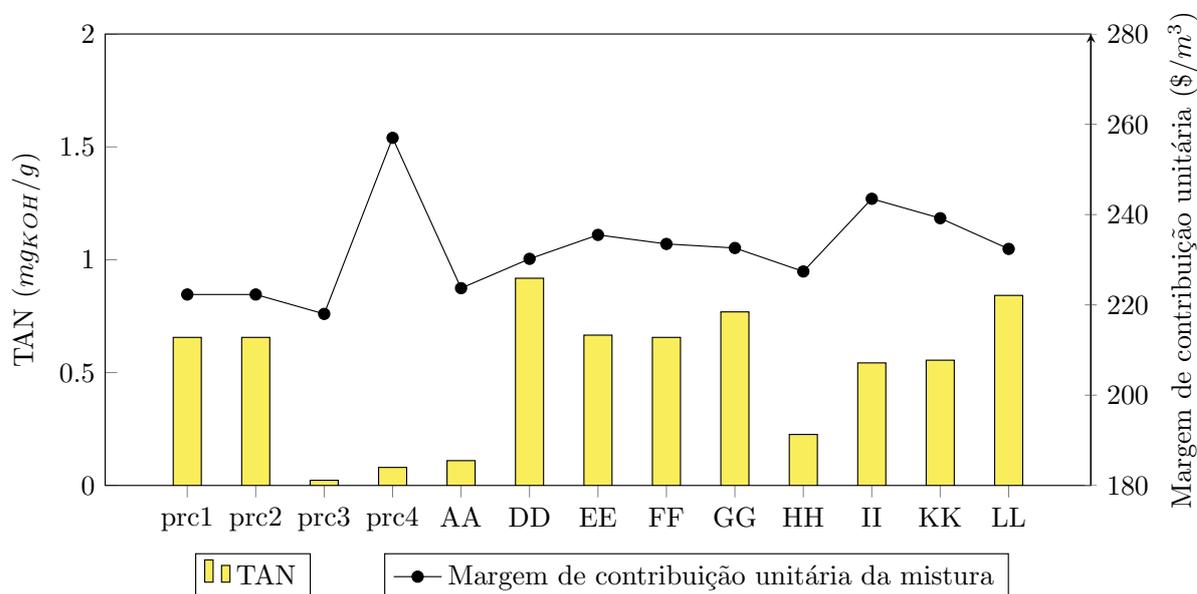


Figura 4.14: Valor marginal unitário e número de acidez total (TAN) das misturas em cada recurso para o cenário cinco.

O valor médio da propriedade-chave TAN em todos os recursos é 0,66 mg_{KOH}/g, e o valor médio da valor marginal das misturas de petróleo no sistema é 233,10 \$.

Cenário seis

No cenário seis há trinta e seis tipos de petróleo no sistema. Duas parcelas estão programadas ao longo de um horizonte de tempo de 240 h (10 dias). Os tanques disponíveis para a programação são AA, DD, EE, FF, GG, HH, II, KK e LL. Algumas parcelas contém mais de um tipo de petróleo.

O volume de petróleo que chega nas parcelas é 66.550 m³ e 26.700 m³ para *prc1* e *prc2*, respectivamente. A vazão de descarregamento é 1.550 m³ h⁻¹ para a parcela *prc1* e 1.700 m³ h⁻¹ para a parcela *prc2*. A parcela *prc1* contém dois tipos de petróleo. O inventário útil inicial nos tanques de carga R_C (inventário inicial menos o lastro dos tanques) é de 233.302,05 m³. A Figura 4.15 mostra o instante de chegada e final do descarregamento de cada parcela.

A Figura 4.16 exibe o valor do TAN e do valor marginal unitário das misturas

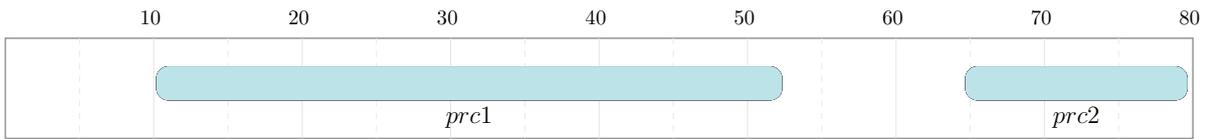


Figura 4.15: Dados de entrada sobre a programação das parcelas no cenário seis com instantes de chegada e final do descarregamento de cada parcela em h.

em cada recurso no cenário quatro. O valor médio da propriedade-chave TAN em

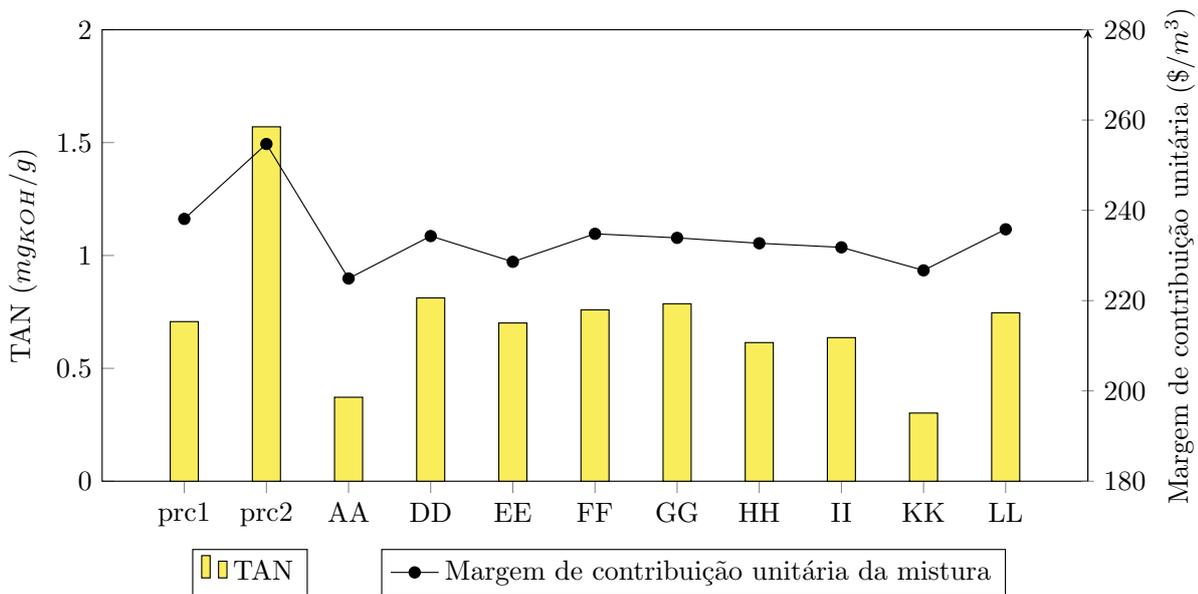


Figura 4.16: Valor marginal unitário e número de acidez total (TAN) das misturas em cada recurso para o cenário seis.

todos os recursos é $0,81 \text{ mg}_{\text{KOH}}/\text{g}$, e o valor médio do valor marginal das misturas de petróleo no sistema é $234,71 \text{ \$}$.

As Figuras 4.17 a 4.22 exibem a configuração dos seis cenários otimizados.

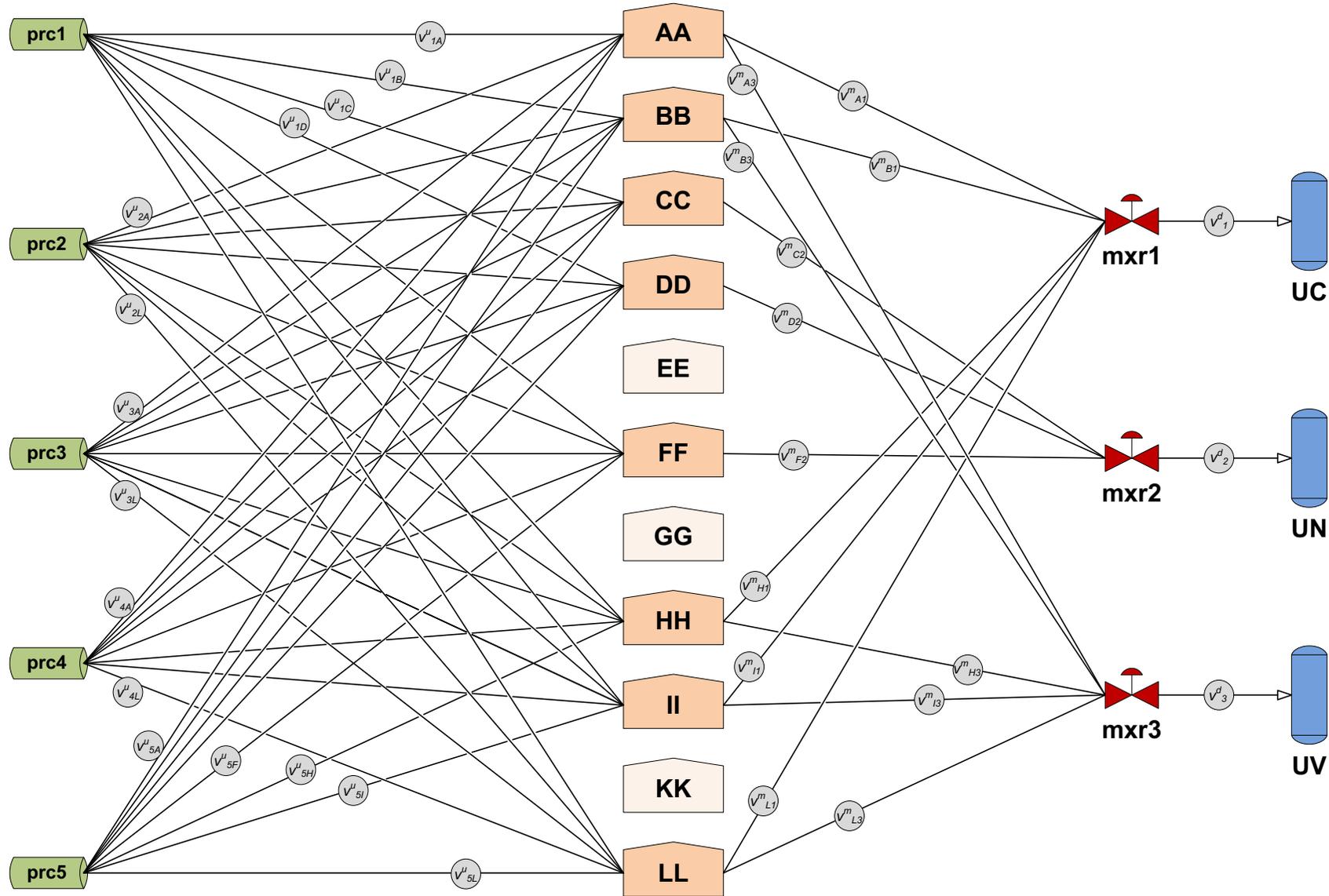


Figura 4.17: Cenário 1: cinco parcelas de petróleo R_P programadas, oito tanques de carga R_C disponíveis e horizonte de tempo de 168 h.

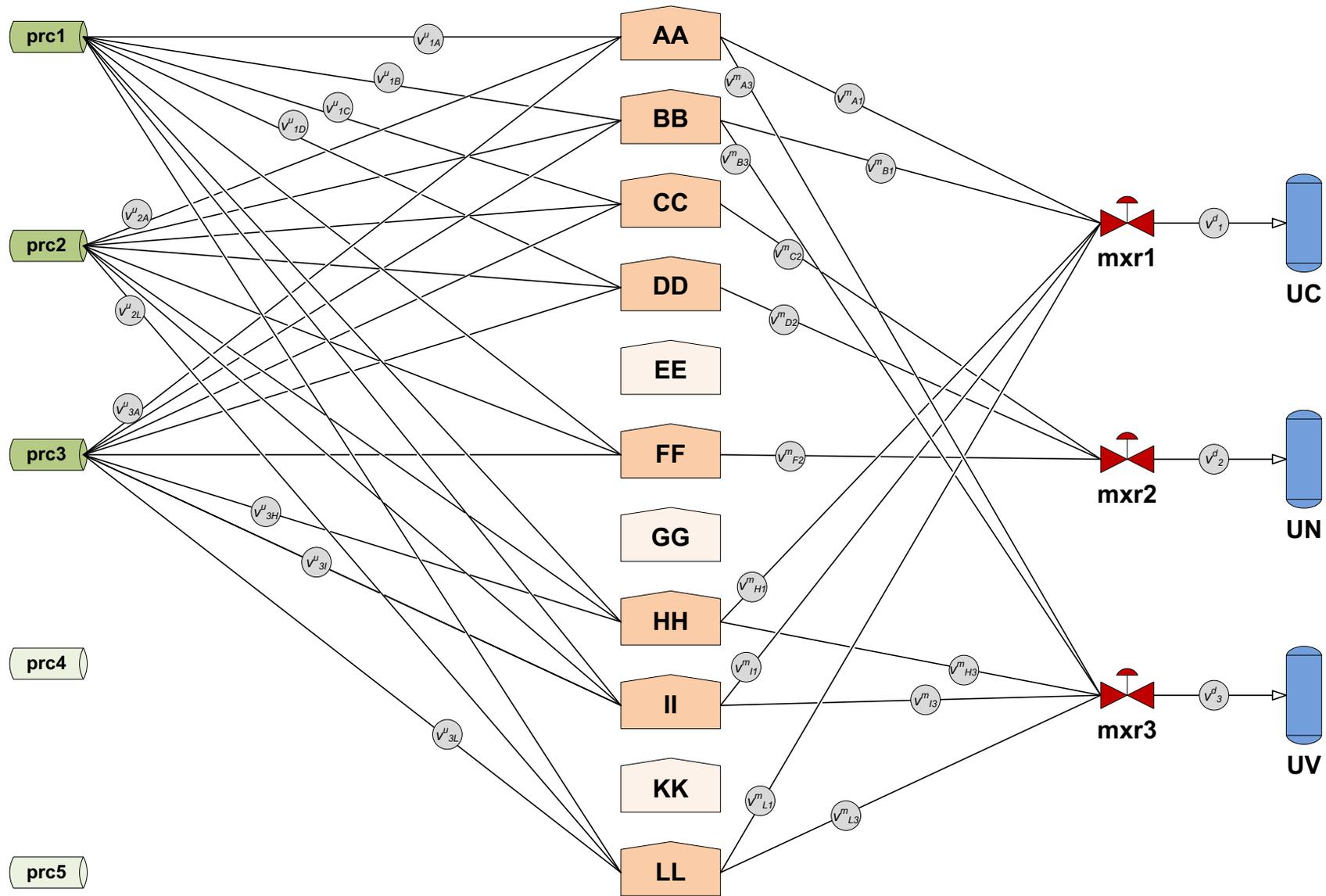


Figura 4.18: Cenário 2: três parcelas de petróleo R_P programadas, oito tanques de carga R_C disponíveis e horizonte de tempo de 168 h.

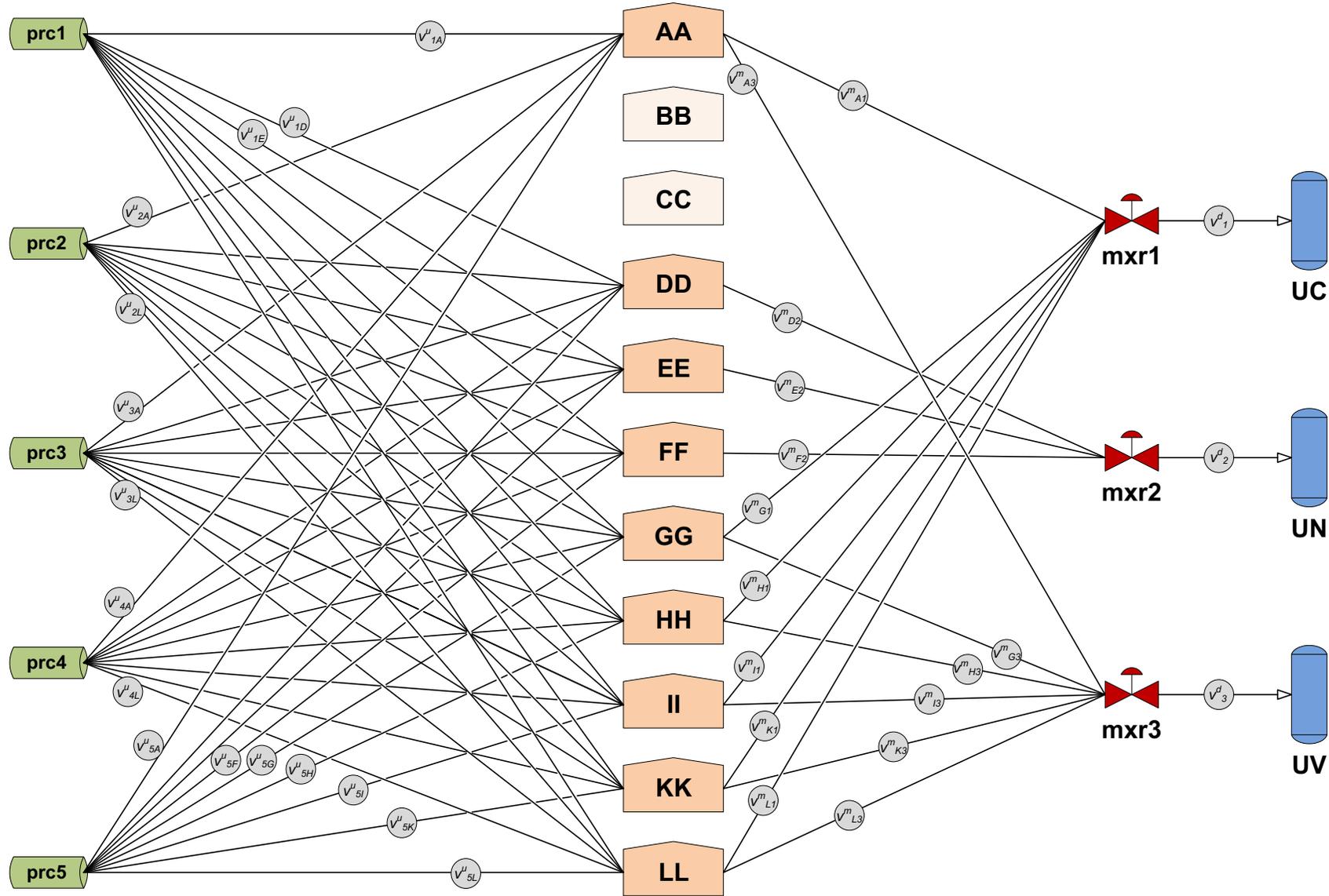


Figura 4.19: Cenário 3: cinco parcelas de petróleo R_P programadas, nove tanques de carga R_C disponíveis e horizonte de tempo de 240 h.

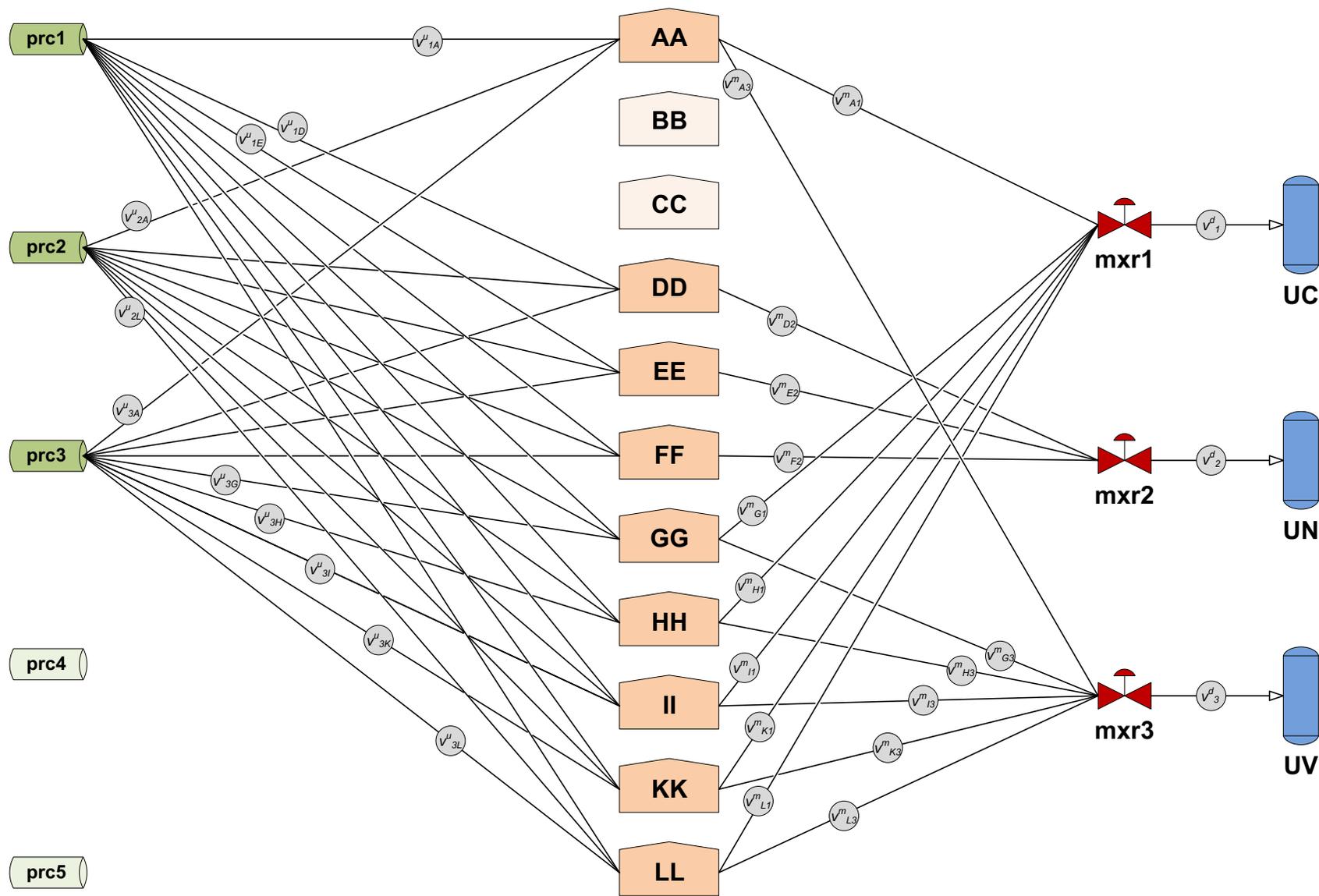


Figura 4.20: Cenário 4: três parcelas de petróleo R_P programadas, nove tanques de carga R_C disponíveis e horizonte de tempo de 168 h.

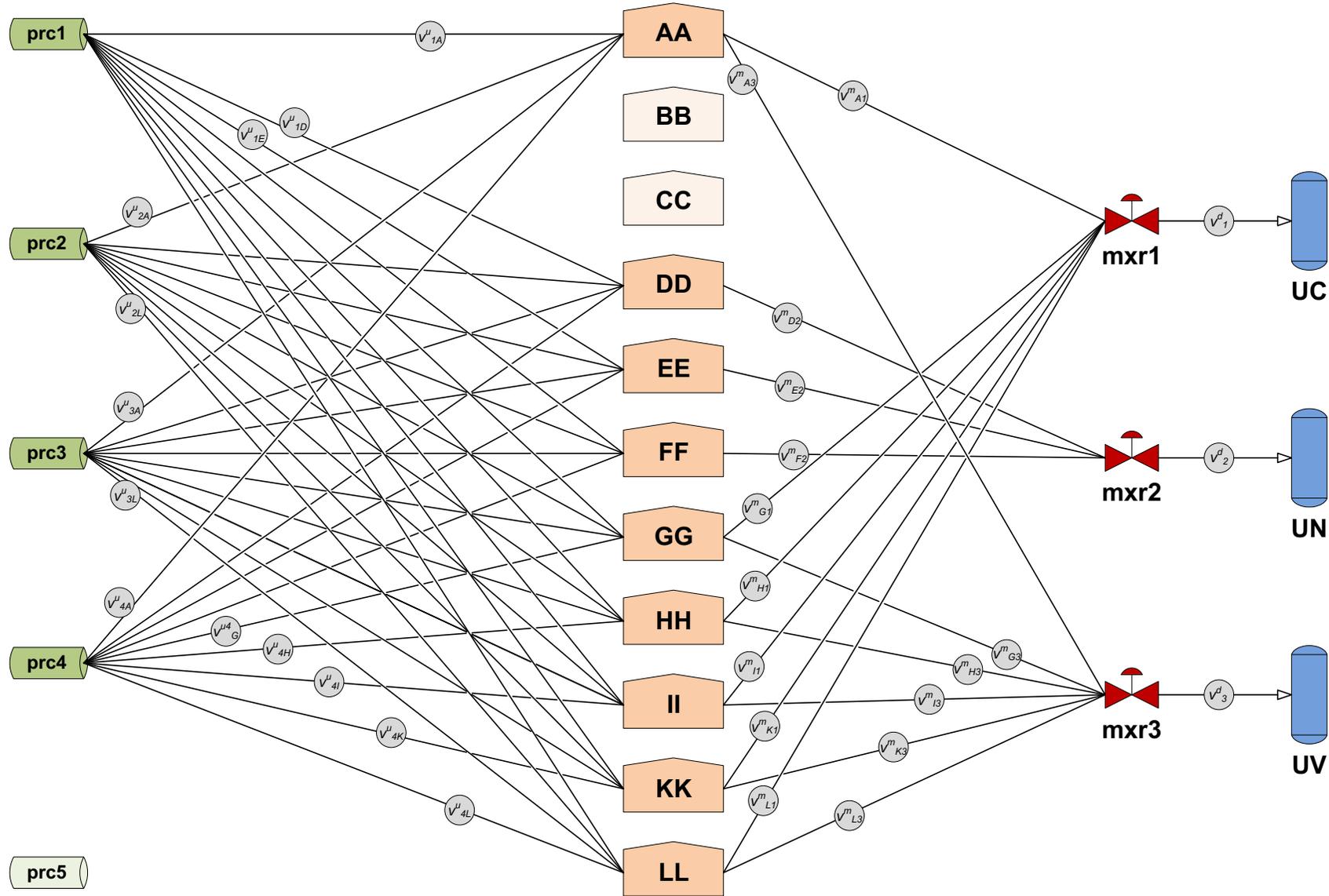


Figura 4.21: Cenário 5: quatro parcelas de petróleo R_P programadas, nove tanques de carga R_C disponíveis e horizonte de tempo de 144 h.

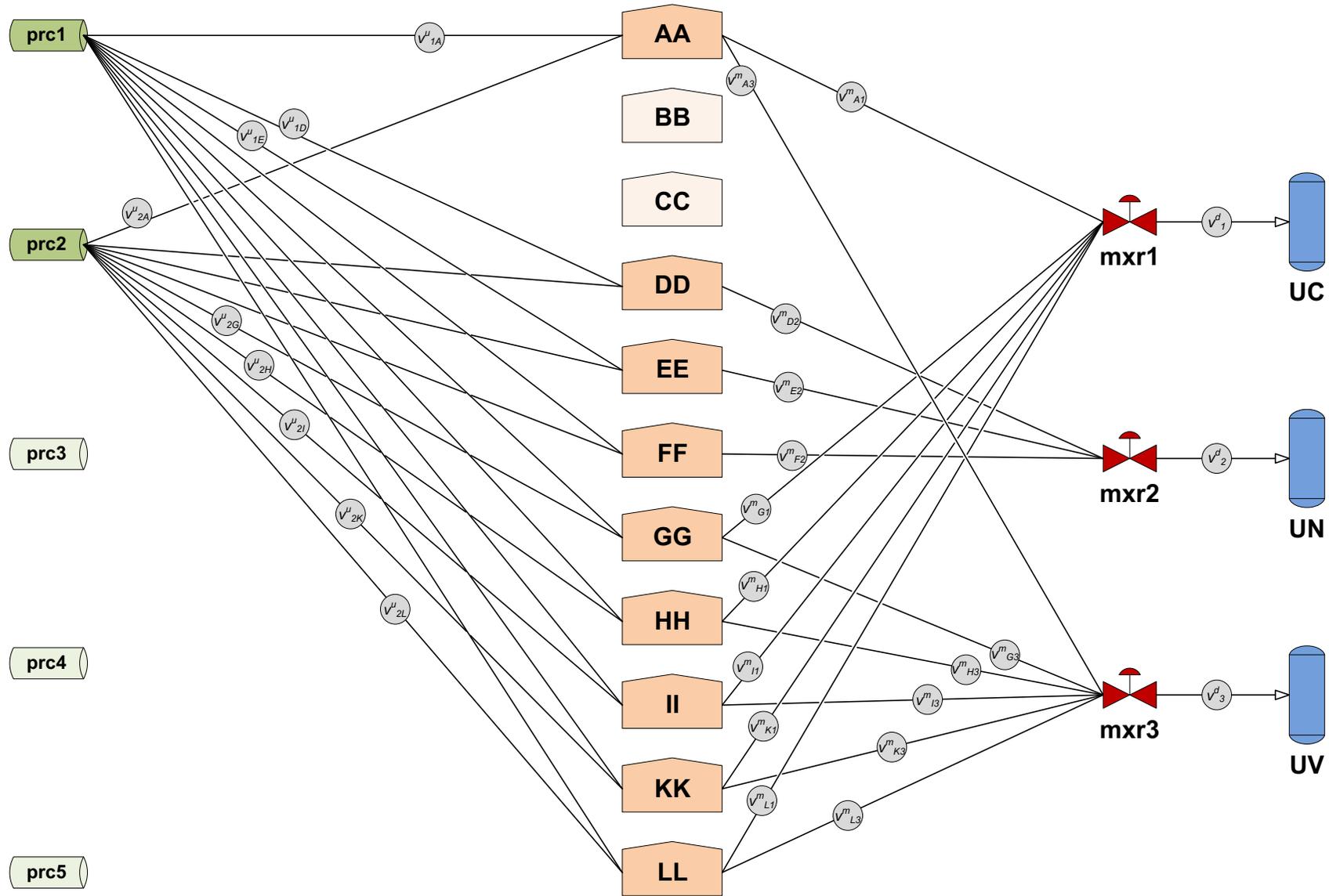


Figura 4.22: Cenário 6: duas parcelas de petróleo R_P programadas, nove tanques de carga R_C disponíveis e horizonte de tempo de 240 h.

MODELO MATEMÁTICO

A formulação **MOS** foi selecionada para modelar o caso real da indústria brasileira por ser uma das formulações mais eficientes de acordo com o estudo comparativo feito por Chen et al. (2012). Ela usa *slots* de prioridade para representar eventos. *Slots* de prioridade são definidos por Mouret et al. (2009) como uma posição i na sequência de operações que possui prioridade maior no programa do que outro *slot* de prioridade j , sendo $j > i$.

A eficiência da formulação **MOS** pode ser atribuída às três características marcantes que a diferem das demais formulações para programação do suprimento de petróleo encontradas na literatura. A primeira é que a formulação é baseada em movimentações entre os recursos enquanto em outras formulações a modelagem está baseada na definição dos recursos. Esta mudança de paradigma se reflete nas variáveis de decisão do modelo e nas restrições.

A segunda característica é a restrição de quebra de simetria, que reduz o número de soluções viáveis de mesma função objetivo. A restrição de quebra de simetria foi originalmente proposta por Mouret et al. (2009) no modelo **SOS**. Soluções simétricas geram o mesmo programa, mas com sequências de *slots* diferentes. Segundo Kallrath (2002) problemas de programação da produção são marcados por simetria, o que os torna particularmente difíceis de resolver. A restrição de quebra de simetria reduz drasticamente a árvore de busca do algoritmo *Branch & Bound* – que deixa de explorar nós com soluções inteiras redundantes – e, conseqüentemente, o tempo de CPU necessário para resolver o modelo.

A terceira característica é a representação de operações sobrepostas e não-sobrepostas através de conjuntos clique. Cliques são conjuntos usados para modelar operações ou tarefas que podem acontecer em uma mesma unidade ou equipamento. Sequenciamento e cronometragem de operações em recursos unários – aqueles que

só podem realizar uma tarefa por vez – são feitos por restrições de não-sobreposição (*non-overlapping*) fundamentadas nos cliques. A formulação MOS usa uma representação unificada para todas as restrições de não-sobreposição, embasada no grafo de operações não-sobrepostas e no conceito de clique. Mouret et al. (2011) explicam detalhadamente o uso de cliques para coordenar operações não-sobrepostas. Os autores apontam que usar cliques máximos na formulação conduz a relaxações lineares mais justas e, conseqüentemente, melhores performances computacionais.

Cliques

Srinivasan (2000) define um clique como um subconjunto dos vértices de um grafo em que cada elemento é adjacente à todos os outros. No que diz respeito ao modelo MOS, essa definição pode ser fraseada como um subconjunto do conjunto de operações $W' \in W$ tal que quaisquer duas operações de W' não devam se sobrepor. É importante notar que todas as operações são não-sobrepostas consigo mesmas.

De acordo com Mouret et al. (2011) o principal objetivo da representação por grafos no modelo MOS é criar restrições reforçadas para operações não-sobrepostas. Os autores relatam que usar cliques máximos ao invés de não-máximos geralmente leva à LR mais justas e a um modelo mais compacto. Um clique máximo é um clique que não está contido em nenhum outro. Encontrar cliques máximos (*maximal cliques*) em um grafo não é um procedimento trivial. Mouret et al. (2011) usa um algoritmo para enumerar os cliques máximos ao resolver os estudos de caso de Lee et al. (1996).

O problema da enumeração de cliques máximos (*maximal clique enumeration*, MCE) consiste em enumerar todos os cliques máximos de um grafo. Uma vez enumerados, cliques máximos são amplamente utilizados para resolver problemas nas áreas de alinhamento estrutural 3-D de proteínas, mapeamento de genoma, análise de expressão de genes e identificação de hierarquias sociais (SCHMIDT et al., 2009). Até os mais eficientes algoritmos para enumeração de cliques máximos demandam muito tempo para enumerar todos os cliques em redes com centenas, milhares ou até mais vértices. Schmidt et al. (2009) e Cheng et al. (2012) propõem algoritmos para enumeração de cliques máximos.

No problema da programação do suprimento de petróleo estudado nesse tra-

Tabela 5.1: Número de vértices do grafo de operações não-sobrepostas em cada cenário.

cenário	vértices
um	56
dois	40
três	63
quatro	45
cinco	54
seis	36

balho, os vértices do grafo de operações não-sobrepostas são as operações W . A Tabela 5.1 exibe a quantidade de vértices do grafo de operações não-sobrepostas em cada cenário. Pela Tabela 5.1 verifica-se que o cenário três dá origem ao problema de maior dimensão, seguido dos cenários um, cinco, quatro, dois e seis.

As operações de descarregamento W_U são sempre não-sobrepostas, pois as parcelas chegam em instantes diferentes ao longo do horizonte de tempo. Se todas as operações de descarregamento W_U forem modeladas como um único clique, conforme ilustra a Figura 5.1, é obrigatório que cada uma delas aconteça em *slots* de prioridade distintos. A Equação 5.34 é responsável por alocar operações de um conjunto clique W' a *slots* diferentes. Se houver, por exemplo, cinco parcelas chegando ao longo do horizonte de tempo, são necessários no mínimo cinco *slots* de prioridade para resolver o modelo. Na modelagem com DNS foi construído um conjunto clique contendo todas as operações de descarregamento de parcelas.

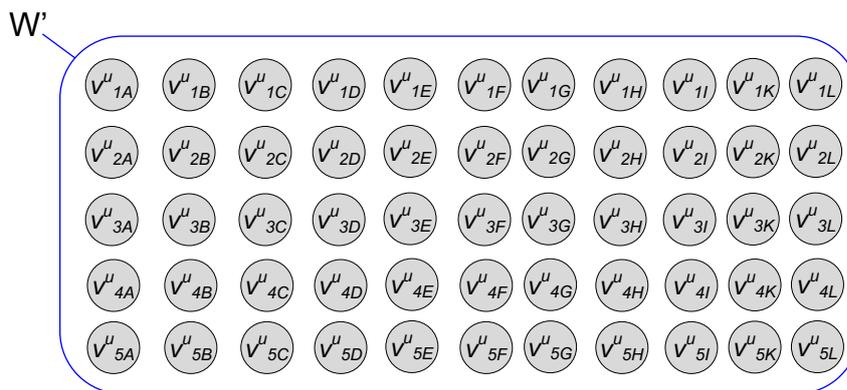


Figura 5.1: Clique com descarregamentos não-sobrepostos.

Outra abordagem é modelar as operações de descarregamento W_U como cliques disjuntos, conforme ilustra a Figura 5.2. É permitido então que descarregamen-

tos de parcelas distintas aconteçam em um mesmo *slot* de prioridade. Por exemplo, é possível resolver um cenário com cinco parcelas programadas com apenas três *slots* de prioridade. Nesse caso os descarregamentos de cada parcela $O_{r,r \in R_P}$ seriam distribuídos pelo modelo em três *slots*. Na modelagem com **DS** foram construídos conjuntos clique disjuntos, contendo operações de descarregamento de cada parcela separadamente.



Figura 5.2: Cliques com descarregamentos sobrepostos.

Embora a modelagem dos cliques com descarregamentos W_U não-sobrepostos (Figura 5.1) gere um modelo mais compacto, ela exige maior quantidade de *slots* de prioridade do a modelagem dos cliques com descarregamentos W_U sobrepostos (Figura 5.2). Isso acontece porque as operações de um conjunto clique são sempre alocadas à *slots* de prioridade distintos, segundo a Equação 5.34. A performance computacional do modelo depende fortemente do número de *slots* de prioridade da formulação (MARAVELIAS; GROSSMANN, 2003, IERAPETRITOU; FLOUDAS, 1998a, MÉNDEZ et al., 2008). Por conseguinte, usar cliques máximos na modelagem de um problema real pode levar a tempos computacionais impraticáveis. Por esta razão as duas modelagens foram testadas, uma com descarregamentos não-sobrepostos e outra com descarregamentos sobrepostos. Como não foram usados algoritmos de enumeração de cliques máximos, não há evidência científica de que nenhum dos cliques nas Figuras 5.1 e 5.2 seja máximo.

A Figura 5.3 ilustra uma solução hipotética para as operações de descarregamento das duas modelagens.

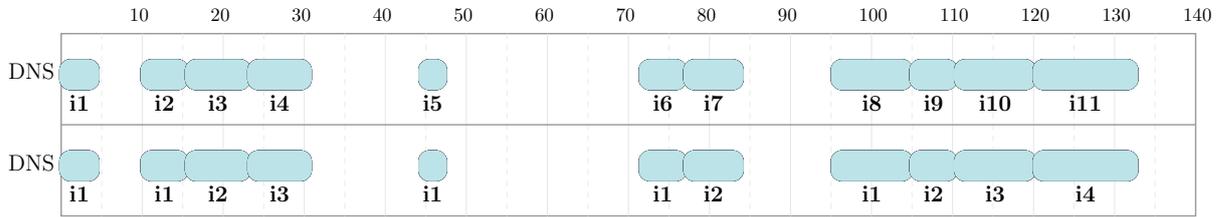


Figura 5.3: Solução hipotética para as operações de descarregamento no modelo com descarregamentos não-sobrepostos (DNS) e no modelo com descarregamentos sobrepostos (DS) usando a programação de parcelas do cenário três.

A Figura 5.3 ilustra o fato de que para fragmentar o descarregamento de parcelas em muitas operações o modelo com descarregamentos sobrepostos requer bem menos *slots* de prioridade. Isso é uma consequência dos conjuntos cliques mostrados nas Figuras 5.1 e 5.2. Na formulação MOS de Mouret et al. (2011), operações que pertencem a um mesmo conjunto clique precisam ser alocadas à *slots* de prioridade diferentes. Isso é conseguido pela restrição de alocação na Equação 5.1.

$$\sum_{v \in W'} Z_{iv} \leq 1 \quad i \in T, W' \in \text{clique}(G_{NO}) \cup \text{clique}(G_{NOTR}) \quad (5.1)$$

Na equação 5.1, $\text{clique}(G_{NO})$ é o conjunto dos conjuntos clique, e $\text{clique}(G_{NOTR})$ é o conjunto dos conjuntos clique com tempo de transição. De acordo com a Equação 5.1, a soma de todas as variáveis de alocação Z_{iv} para o *slot* de prioridade i das operações contidas em um mesmo conjunto clique devem ser menor ou igual a um. Ou seja, apenas uma das operações contidas em um mesmo clique pode ser alocada ao *slot* i .

Portanto, a forma de construir os conjuntos clique das operações de descarregamento no modelo com descarregamentos sobrepostos permite que o modelo encontre soluções viáveis com menos *slots* de prioridade. Na solução hipotética da Figura 5.3, por exemplo, o modelo com descarregamentos não-sobrepostos precisaria de pelo menos 11 *slots* para encontrar essa solução, e o modelo com descarregamentos sobrepostos precisaria de pelo menos 4 *slots*. Essa é a principal vantagem do

modelo com descarregamentos sobrepostos, pois o tempo de CPU necessário para solucionar o modelo é também menor.

Por outro lado, usar descarregamentos sobrepostos pode provocar duas situações indesejáveis em relação ao modelo nos tanques de carga:

1. um tanque de carga R_C pode receber descarregamentos provindos de parcelas diferentes, em instantes diferentes, mas em um mesmo *slot* de prioridade;
2. um tanque de carga R_C pode receber descarregamentos provindos de parcelas diferentes, em instantes diferentes, em *slots* de prioridade distintos, mas que não formam uma sequência crescente (i_4 antes de i_2 , por exemplo).

Essas duas situações invalidam os balanços de massa nos tanques, e restrições foram adicionadas à modelagem com descarregamentos sobrepostos para evitá-las.

De acordo com a regra operacional (ii) da tabela 4.2 não são permitidas operações simultâneas de entrada e saída nos tanques de carga (operação pulmão). É possível identificar cliques entre operações de entrada e saída de um mesmo tanque de carga $r \in R_C$. A Figura 5.4 ilustra as operações de entrada e saída do tanque AA.

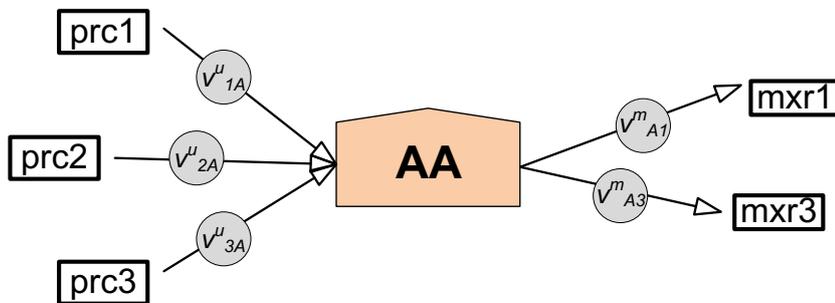


Figura 5.4: Operações de entrada e saída do tanque de carga AA.

Cada operação de entrada forma um clique com cada operação de saída. De acordo com a regra (iii) da Tabela 4.2 as operações de saída do tanque podem acontecer simultaneamente e são, portanto, operação sobrepostas, ou seja, não formam clique entre si.

Na modelagem com descarregamentos não-sobrepostos procurou-se construir conjuntos clique de maior cardinalidade possível, com a intenção de aproximá-los dos cliques máximos. Os cliques com tempo de transição nos tanques de carga R_C

foram construídos da forma ilustrada na Figura 5.5 para o tanque AA. Todas as operações de entrada com uma operação de saída do tanque. Na modelagem com descarregamentos não-sobrepostos há portanto em cada tanque de carga $r \in R_C$ tantos cliques quanto operações de saída.

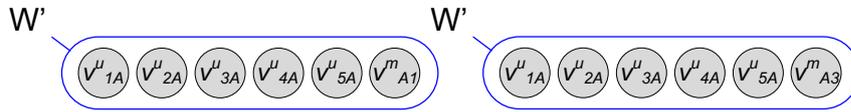


Figura 5.5: Cliques entre operações de entrada e saída do tanque de carga AA na modelagem com descarregamentos não-sobrepostos.

Na modelagem com descarregamentos sobrepostos procurou-se fazer construir conjuntos clique de menor cardinalidade. Os cliques com tempo de transição nos tanques de carga R_C foram construídos da forma ilustrada na Figura 5.6 para o tanque AA. Cada operação de saída forma um clique com cada operação de entrada. Na modelagem com descarregamentos sobrepostos há portanto $|I_r| \times |O_r|$, $r \in R_C$ cliques associados à cada tanque de carga R_C .

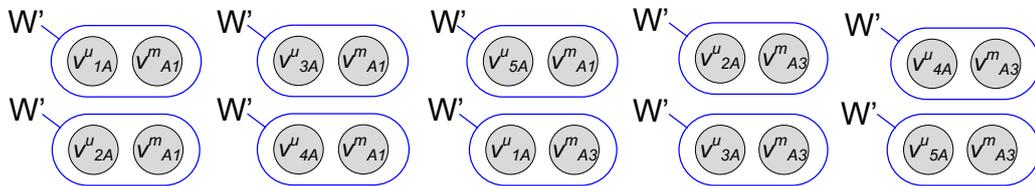


Figura 5.6: Cliques entre operações de entrada e saída do tanque de carga AA.

As operações de destilação W_D são não-sobrepostas apenas consigo mesmas em ambas modelagens. Além disso, elas formam cliques isolados, que são sempre máximos (MOURET et al., 2011).

As principais diferenças entre as modelagens com descarregamentos não-sobrepostos e descarregamentos sobrepostos são:

- os conjuntos clique(G_{NO}) e clique com tempo de transição clique(G_{NOTR});
- as restrições que fazem a cronometragem de operações não-sobrepostas.

As Tabelas 5.2 e 5.3 exibem os conjuntos clique clique(G_{NO}) e clique(G_{NOTR}) usados na modelagem com descarregamentos não-sobrepostos. As Tabelas 5.4 e 5.5

Tabela 5.2: Conjuntos clique $W' \in \text{clique}(G_{NO})$ da modelagem com descarregamentos não-sobrepostos.

W_1	$\{v_{1A}^u, v_{2A}^u, v_{3A}^u, v_{4A}^u, v_{5A}^u, v_{1B}^u, v_{2B}^u, v_{3B}^u, v_{4B}^u, v_{5B}^u,$ $v_{1C}^u, v_{2C}^u, v_{3C}^u, v_{4C}^u, v_{5C}^u, v_{1D}^u, v_{2D}^u, v_{3D}^u, v_{4D}^u, v_{5D}^u,$ $v_{1E}^u, v_{2E}^u, v_{3E}^u, v_{4E}^u, v_{5E}^u, v_{1F}^u, v_{2F}^u, v_{3F}^u, v_{4F}^u, v_{5F}^u,$ $v_{1G}^u, v_{2G}^u, v_{3G}^u, v_{4G}^u, v_{5G}^u, v_{1H}^u, v_{2H}^u, v_{3H}^u, v_{4H}^u, v_{5H}^u,$ $v_{1I}^u, v_{2I}^u, v_{3I}^u, v_{4I}^u, v_{5I}^u, v_{1K}^u, v_{2K}^u, v_{3K}^u, v_{4K}^u, v_{5K}^u,$ $v_{1L}^u, v_{2L}^u, v_{3L}^u, v_{4L}^u, v_{5L}^u\}$	W_{20}	$\{v_1^d\}$
		W_{21}	$\{v_2^d\}$
		W_{22}	$\{v_3^d\}$

Tabela 5.3: Conjuntos clique com tempo de transição $W' \in \text{clique}(G_{NOTR})$ da modelagem com descarregamentos não-sobrepostos.

W_2	$\{v_{1A}^u, v_{2A}^u, v_{3A}^u, v_{4A}^u, v_{5A}^u, v_{A1}^m\}$	W_{11}	$\{v_{1G}^u, v_{2G}^u, v_{3G}^u, v_{4G}^u, v_{5G}^u, v_{G3}^m\}$
W_3	$\{v_{1A}^u, v_{2A}^u, v_{3A}^u, v_{4A}^u, v_{5A}^u, v_{A3}^m\}$	W_{12}	$\{v_{1H}^u, v_{2H}^u, v_{3H}^u, v_{4H}^u, v_{5H}^u, v_{H1}^m\}$
W_4	$\{v_{1B}^u, v_{2B}^u, v_{3B}^u, v_{4B}^u, v_{5B}^u, v_{B1}^m\}$	W_{13}	$\{v_{1H}^u, v_{2H}^u, v_{3H}^u, v_{4H}^u, v_{5H}^u, v_{H3}^m\}$
W_5	$\{v_{1B}^u, v_{2B}^u, v_{3B}^u, v_{4B}^u, v_{5B}^u, v_{B3}^m\}$	W_{14}	$\{v_{1I}^u, v_{2I}^u, v_{3I}^u, v_{4I}^u, v_{5I}^u, v_{I1}^m\}$
W_6	$\{v_{1C}^u, v_{2C}^u, v_{3C}^u, v_{4C}^u, v_{5C}^u, v_{C2}^m\}$	W_{15}	$\{v_{1I}^u, v_{2I}^u, v_{3I}^u, v_{4I}^u, v_{5I}^u, v_{I3}^m\}$
W_7	$\{v_{1D}^u, v_{2D}^u, v_{3D}^u, v_{4D}^u, v_{5D}^u, v_{D2}^m\}$	W_{16}	$\{v_{1K}^u, v_{2K}^u, v_{3K}^u, v_{4K}^u, v_{5K}^u, v_{K1}^m\}$
W_8	$\{v_{1E}^u, v_{2E}^u, v_{3E}^u, v_{4E}^u, v_{5E}^u, v_{E2}^m\}$	W_{17}	$\{v_{1K}^u, v_{2K}^u, v_{3K}^u, v_{4K}^u, v_{5K}^u, v_{K3}^m\}$
W_9	$\{v_{1F}^u, v_{2F}^u, v_{3F}^u, v_{4F}^u, v_{5F}^u, v_{F2}^m\}$	W_{18}	$\{v_{1L}^u, v_{2L}^u, v_{3L}^u, v_{4L}^u, v_{5L}^u, v_{L1}^m\}$
W_{10}	$\{v_{1G}^u, v_{2G}^u, v_{3G}^u, v_{4G}^u, v_{5G}^u, v_{G1}^m\}$	W_{19}	$\{v_{1L}^u, v_{2L}^u, v_{3L}^u, v_{4L}^u, v_{5L}^u, v_{L3}^m\}$

exibem os conjuntos $\text{clique}(G_{NO})$ e $\text{clique}(G_{NOTR})$ usados na modelagem com descarregamentos sobrepostos. Na modelagem com descarregamentos não-sobrepostos há no total $|\text{clique}(G_{NO}) \cup \text{clique}(G_{NOTR})|$ vinte e dois conjuntos clique e na modelagem com descarregamentos sobrepostos há no total cento e catorze conjuntos clique.

Tabela 5.4: Conjuntos clique $W' \in \text{clique}(G_{NO})$ da modelagem com descarregamentos sobrepostos.

W_1	$\{v_{1A}^u, v_{2A}^u, v_{3A}^u, v_{4A}^u, v_{5A}^u\}$	W_8	$\{v_{1H}^u, v_{2H}^u, v_{3H}^u, v_{4H}^u, v_{5H}^u\}$
W_2	$\{v_{1B}^u, v_{2B}^u, v_{3B}^u, v_{4B}^u, v_{5B}^u\}$	W_9	$\{v_{1I}^u, v_{2I}^u, v_{3I}^u, v_{4I}^u, v_{5I}^u\}$
W_3	$\{v_{1C}^u, v_{2C}^u, v_{3C}^u, v_{4C}^u, v_{5C}^u\}$	W_{10}	$\{v_{1K}^u, v_{2K}^u, v_{3K}^u, v_{4K}^u, v_{5K}^u\}$
W_4	$\{v_{1D}^u, v_{2D}^u, v_{3D}^u, v_{4D}^u, v_{5D}^u\}$	W_{11}	$\{v_{1L}^u, v_{2L}^u, v_{3L}^u, v_{4L}^u, v_{5L}^u\}$
W_5	$\{v_{1E}^u, v_{2E}^u, v_{3E}^u, v_{4E}^u, v_{5E}^u\}$	W_{112}	$\{v_1^d\}$
W_6	$\{v_{1F}^u, v_{2F}^u, v_{3F}^u, v_{4F}^u, v_{5F}^u\}$	W_{113}	$\{v_2^d\}$
W_7	$\{v_{1G}^u, v_{2G}^u, v_{3G}^u, v_{4G}^u, v_{5G}^u\}$	W_{114}	$\{v_3^d\}$

Tabela 5.5: Conjuntos clique com tempo de transição $W' \in \text{clique}(G_{NOTR})$ da modelagem com descarregamentos sobrepostos.

W_{12}	$\{v_{1A}^u, v_{A1}^m\}$	W_{37}	$\{v_{1D}^u, v_{D2}^m\}$	W_{62}	$\{v_{1G}^u, v_{G3}^m\}$	W_{87}	$\{v_{1K}^u, v_{K1}^m\}$
W_{13}	$\{v_{2A}^u, v_{A1}^m\}$	W_{38}	$\{v_{2D}^u, v_{D2}^m\}$	W_{63}	$\{v_{2G}^u, v_{G3}^m\}$	W_{88}	$\{v_{2K}^u, v_{K1}^m\}$
W_{14}	$\{v_{3A}^u, v_{A1}^m\}$	W_{39}	$\{v_{3D}^u, v_{D2}^m\}$	W_{64}	$\{v_{3G}^u, v_{G3}^m\}$	W_{89}	$\{v_{3K}^u, v_{K1}^m\}$
W_{15}	$\{v_{4A}^u, v_{A1}^m\}$	W_{40}	$\{v_{4D}^u, v_{D2}^m\}$	W_{65}	$\{v_{4G}^u, v_{G3}^m\}$	W_{90}	$\{v_{4K}^u, v_{K1}^m\}$
W_{16}	$\{v_{5A}^u, v_{A1}^m\}$	W_{41}	$\{v_{5D}^u, v_{D2}^m\}$	W_{66}	$\{v_{5G}^u, v_{G3}^m\}$	W_{91}	$\{v_{5K}^u, v_{K1}^m\}$
W_{17}	$\{v_{1A}^u, v_{A3}^m\}$	W_{42}	$\{v_{1E}^u, v_{E2}^m\}$	W_{67}	$\{v_{1H}^u, v_{H1}^m\}$	W_{92}	$\{v_{1K}^u, v_{K3}^m\}$
W_{18}	$\{v_{2A}^u, v_{A3}^m\}$	W_{43}	$\{v_{2E}^u, v_{E2}^m\}$	W_{68}	$\{v_{2H}^u, v_{H1}^m\}$	W_{93}	$\{v_{2K}^u, v_{K3}^m\}$
W_{19}	$\{v_{3A}^u, v_{A3}^m\}$	W_{44}	$\{v_{3E}^u, v_{E2}^m\}$	W_{69}	$\{v_{3H}^u, v_{H1}^m\}$	W_{94}	$\{v_{3K}^u, v_{K3}^m\}$
W_{20}	$\{v_{4A}^u, v_{A3}^m\}$	W_{45}	$\{v_{4E}^u, v_{E2}^m\}$	W_{70}	$\{v_{4H}^u, v_{H1}^m\}$	W_{95}	$\{v_{4K}^u, v_{K3}^m\}$
W_{21}	$\{v_{5A}^u, v_{A3}^m\}$	W_{46}	$\{v_{5E}^u, v_{E2}^m\}$	W_{71}	$\{v_{5H}^u, v_{H1}^m\}$	W_{96}	$\{v_{5K}^u, v_{K3}^m\}$
W_{22}	$\{v_{1B}^u, v_{B1}^m\}$	W_{47}	$\{v_{1F}^u, v_{F2}^m\}$	W_{72}	$\{v_{1H}^u, v_{H3}^m\}$	W_{97}	$\{v_{1L}^u, v_{L1}^m\}$
W_{23}	$\{v_{2B}^u, v_{B1}^m\}$	W_{48}	$\{v_{2F}^u, v_{F2}^m\}$	W_{73}	$\{v_{2H}^u, v_{H3}^m\}$	W_{98}	$\{v_{2L}^u, v_{L1}^m\}$
W_{24}	$\{v_{3B}^u, v_{B1}^m\}$	W_{49}	$\{v_{3F}^u, v_{F2}^m\}$	W_{74}	$\{v_{3H}^u, v_{H3}^m\}$	W_{99}	$\{v_{3L}^u, v_{L1}^m\}$
W_{25}	$\{v_{4B}^u, v_{B1}^m\}$	W_{50}	$\{v_{4F}^u, v_{F2}^m\}$	W_{75}	$\{v_{4H}^u, v_{H3}^m\}$	W_{100}	$\{v_{4L}^u, v_{L1}^m\}$
W_{26}	$\{v_{5B}^u, v_{B1}^m\}$	W_{51}	$\{v_{5F}^u, v_{F2}^m\}$	W_{76}	$\{v_{5H}^u, v_{H3}^m\}$	W_{101}	$\{v_{5L}^u, v_{L1}^m\}$
W_{27}	$\{v_{1B}^u, v_{B3}^m\}$	W_{52}	$\{v_{1G}^u, v_{G1}^m\}$	W_{77}	$\{v_{1I}^u, v_{I1}^m\}$	W_{102}	$\{v_{1L}^u, v_{L3}^m\}$
W_{28}	$\{v_{2B}^u, v_{B3}^m\}$	W_{53}	$\{v_{2G}^u, v_{G1}^m\}$	W_{78}	$\{v_{2I}^u, v_{I1}^m\}$	W_{103}	$\{v_{2L}^u, v_{L3}^m\}$
W_{29}	$\{v_{3B}^u, v_{B3}^m\}$	W_{54}	$\{v_{3G}^u, v_{G1}^m\}$	W_{79}	$\{v_{3I}^u, v_{I1}^m\}$	W_{104}	$\{v_{3L}^u, v_{L3}^m\}$
W_{30}	$\{v_{4B}^u, v_{B3}^m\}$	W_{55}	$\{v_{4G}^u, v_{G1}^m\}$	W_{80}	$\{v_{4I}^u, v_{I1}^m\}$	W_{105}	$\{v_{4L}^u, v_{L3}^m\}$
W_{31}	$\{v_{5B}^u, v_{B3}^m\}$	W_{56}	$\{v_{5G}^u, v_{G1}^m\}$	W_{81}	$\{v_{5I}^u, v_{I1}^m\}$	W_{106}	$\{v_{5L}^u, v_{L3}^m\}$
W_{32}	$\{v_{1C}^u, v_{C2}^m\}$	W_{57}	$\{v_{1G}^u, v_{G1}^m\}$	W_{82}	$\{v_{1I}^u, v_{I3}^m\}$	W_{107}	$\{v_{1L}^u, v_{L3}^m\}$
W_{33}	$\{v_{2C}^u, v_{C2}^m\}$	W_{58}	$\{v_{2G}^u, v_{G1}^m\}$	W_{83}	$\{v_{2I}^u, v_{I3}^m\}$	W_{108}	$\{v_{2L}^u, v_{L3}^m\}$
W_{34}	$\{v_{3C}^u, v_{C2}^m\}$	W_{59}	$\{v_{3G}^u, v_{G1}^m\}$	W_{84}	$\{v_{3I}^u, v_{I3}^m\}$	W_{109}	$\{v_{3L}^u, v_{L3}^m\}$
W_{35}	$\{v_{4C}^u, v_{C2}^m\}$	W_{60}	$\{v_{4G}^u, v_{G1}^m\}$	W_{85}	$\{v_{4I}^u, v_{I3}^m\}$	W_{110}	$\{v_{4L}^u, v_{L3}^m\}$
W_{36}	$\{v_{5C}^u, v_{C2}^m\}$	W_{61}	$\{v_{5G}^u, v_{G1}^m\}$	W_{86}	$\{v_{5I}^u, v_{I3}^m\}$	W_{111}	$\{v_{5L}^u, v_{L3}^m\}$

Modelagem base

Nesta seção será apresentada a modelagem que é comum aos dois modelos testados neste trabalho.

Variáveis

Dados os conjuntos de *slots* de prioridade T , o conjunto de operações W , composto pelas operações de descarregamento de parcelas W_U , operações de mistura W_M e operações de destilação W_D ; o conjunto de recursos R , composto de parcelas R_P , tanques de carga R_C e UDAs R_D ; o conjunto de tanques de carga com duas correntes de saída $R_{C2} \subset R_C$; o conjunto de petróleos C ; o conjunto das propriedades-chave de interesse em base mássica K_m e em base volumétrica K_v ; o conjunto de operações de entrada em um recurso I_r e o conjunto de operações de saída de um recurso O_r ; as variáveis do modelo são:

- **Variáveis de atribuição:** $Z_{iv} \in \{0, 1\}$, $i \in T$, $v \in W$. $Z_{iv} = 1$ caso a operação $v \in W$ seja atribuída ao *slot* de prioridade $i \in T$, caso contrário $Z_{iv} = 0$.
- **Variáveis de tempo:** $S_{iv} \geq 0$, $D_{iv} \geq 0$, $E_{iv} \geq 0$, $i \in T$, $v \in W$. S_{iv} é o instante de começo da operação $v \in W$, D_{iv} é a duração da operação $v \in W$ e E_{iv} é o instante de finalização da operação $v \in W$. Caso a operação $v \in W$ não seja atribuída ao *slot* de prioridade $i \in T$ as variáveis de tempo são zero, $S_{iv} = 0$, $D_{iv} = 0$, $E_{iv} = 0$.
- **Variável de sincronização de operações de mistura:** $\lambda_{ir} \geq 0$, $i \in T$, $r \in R_{C2}$. λ_{ir} é a duração da operações $v \in W$ de saída de um tanque com duas possíveis correntes de saída $r \in R_{C2}$. Caso a operação $v \in O_{R_{C2}}$ não seja atribuída ao *slot* de prioridade $i \in T$ a variável λ_{ir} é zero, $\lambda_{ir} = 0$.
- **Variáveis de operação:** $V_{iv}^t \geq 0$, $V_{ivc} \geq 0$, $i \in T$, $v \in W$, $c \in C$. V_{iv}^t é o volume total de petróleo transferido na operação $v \in W$ e V_{ivc} é o volume de petróleo $c \in C$ transferido na operação $v \in W$. Caso a operação v não seja atribuída ao *slot* de prioridade i as variáveis de operação são zero, $V_{iv}^t = 0$, $V_{ivc} = 0$.

- **Variáveis de recursos:** $L_{ir}^t \geq 0$, $L_{irc} \geq 0$ $i \in T$, $r \in R_S \cup R_C$, $c \in C$. L_{ir}^t é volume total acumulado nos tanques de carga antes da operação atribuída ao *slot* de prioridade $i \in T$ e L_{irc} é volume de petróleo $c \in C$ acumulado nos tanques de carga antes da operação atribuída ao *slot* de prioridade i .

Função objetivo

O objetivo é maximizar o valor marginal das misturas de petróleo destiladas, de acordo com a Equação 5.2, onde G_c é o valor marginal unitário do petróleo $c \in C$.

$$\max \sum_{i \in T} \sum_{r \in R_D} \sum_{v \in I_r} \sum_{c \in C} G_c V_{ivc} \quad (5.2)$$

Restrições

Limites nas variáveis de tempo são expressos com as seguintes restrições:

$$S_{iv} \geq \underline{S}_v Z_{iv} \quad i \in T, v \in W_U \quad (5.3)$$

$$E_{iv} \leq H Z_{iv} \quad i \in T, v \in W \quad (5.4)$$

$$E_{iv} = S_{iv} + D_{iv} \quad i \in T, v \in W \quad (5.5)$$

A restrição 5.3 assegura um limite inferior para o instante de começo de cada operação de descarregamento de parcelas com o parâmetro \underline{S}_v . As restrições 5.4 e 5.5 fazem $S_{iv} = 0$ se a operação de descarregamento $v \in W_U$ não estiver atribuída ao *slot* de prioridade $i \in T$. Analogamente, a restrição 5.4 assegura um limite superior para o instante de finalização de todas as operações, que é o próprio horizonte de tempo H . A restrição 5.5 afirma que o tempo de finalização de uma operação é igual à seu tempo de começo mais a duração.

A restrição 5.6 assegura que os limites de duração mínima de operações são respeitados.

$$D_{iv} \geq \underline{Dur}_v \cdot Z_{iv} \quad i \in T, v \in W \quad (5.6)$$

A restrição de cardinalidade 5.7 controla o número de operações de destilação que podem acontecer ao longo de todo o horizonte de tempo e, por conseguinte, o número de transições nas UDAs. A restrição de cardinalidade para o primeiro *slot*

5.8 força que haja uma alocação das operações de destilação no primeiro *slot* de prioridade.

controla o número de operações de destilação que podem acontecer no primeiro *slot* de prioridade.

$$\underline{N}_D \leq \sum_{i \in T} \sum_{v \in W_D} Z_{iv} \leq \overline{N}_D \quad (5.7)$$

$$\underline{N}_D \leq \sum_{v \in W_D} Z_{iv} \leq \overline{N}_D \quad v \in W_D, i = i1 \quad (5.8)$$

A restrição 5.9 garante operação contínua das UDAs durante todo o horizonte de tempo.

$$\sum_{i \in T} \sum_{v \in I_r} D_{iv} = H, \quad r \in R_D \quad (5.9)$$

Limites de volume total transferido (V_{iv}^t) são expressos pelas restrições 5.10. A restrição 5.11 garante que o volume total da operação $v \in W$ seja a soma dos volumes de cada tipo de petróleo $c \in C$.

$$\underline{V}_v^t Z_{iv} \leq V_{iv}^t \leq \overline{V}_v^t Z_{iv} \quad i \in T, v \in W \quad (5.10)$$

$$V_{iv}^t = \sum_{c \in C} V_{ivc} \quad i \in T, v \in W \quad (5.11)$$

O controle de inventário é feito pelas variáveis de volume nos recursos. A restrição 5.12 define o volume em cada recurso no começo do *slot* de prioridade $i \in T$ como sendo o volume inicial total L_{0r}^t mais soma de todas as movimentações que chegaram menos a soma de todas as movimentações que saíram do recurso nos *slots* de prioridade anteriores. Analogamente, a restrição 5.13 define o volume de cada tipo de petróleo $c \in C$ em cada recurso no começo do *slot* de prioridade $i \in T$. Note que as equações 5.12 e 5.13 calculam o volume de inventário no começo de cada *slot* de prioridade, sem contabilizar as movimentações daquele *slot*. A restrição 5.14 garante que o volume total seja igual à soma dos volumes de cada petróleo $c \in C$ em todos os recursos e *slots*.

$$L_{ir}^t = L_{0r}^t + \sum_{j \in T, j < i} \sum_{v \in I_r} V_{jv}^t - \sum_{j \in T, j < i} \sum_{v \in O_r} V_{jv}^t \quad i \in T, r \in R - R_M \quad (5.12)$$

$$L_{irc} = L_{0rc} + \sum_{j \in T, j < i} \sum_{v \in I_r} V_{jvc} - \sum_{j \in T, j < i} \sum_{v \in O_r} V_{jvc}, \quad i \in T, r \in R - R_M, c \in C \quad (5.13)$$

$$L_{ir}^t = \sum_{c \in C} L_{irc} \quad i \in T, r \in R - R_M \quad (5.14)$$

Os misturadores são recursos sem acúmulo de material e, portanto, sem inventário. As equações 5.15 e 5.16 fazem os balanços de massa global e por tipo de petróleo, respectivamente, nesses elementos.

$$\sum_{v \in I_r} V_{iv}^t = \sum_{v \in O_r} V_{iv}^t \quad i \in T, r \in R_M \quad (5.15)$$

$$\sum_{v \in I_r} V_{ivc} = \sum_{v \in O_r} V_{ivc} \quad i \in T, c \in C, r \in R_M \quad (5.16)$$

De acordo com a regra operacional (iv) da Tabela 4.2, cada UDA pode ser alimentada por no máximo dois tanques de carga simultaneamente. A restrição 5.17 assegura que o no máximo duas operações estejam ativas simultaneamente na entrada dos misturadores.

$$\sum_{v \in I_r} Z_{iv} \leq 2 \quad i \in T, r \in R_M \quad (5.17)$$

Para sincronizar as operações de entrada e saída dos misturadores R_M use-se as equações 5.18 e 5.19 para igualar as variáveis S_{iv} no caso de operações sobrepostas, e as equações 5.20 e 5.21 para igualar as variáveis E_{iv} .

$$S_{iv} \leq S_{iv_1} + H \cdot (1 - Z_{iv}) \quad i \in T, r \in R_M, v \in I_r, v_1 \in O_r \quad (5.18)$$

$$S_{iv} \geq S_{iv_1} - H \cdot (1 - Z_{iv}) \quad i \in T, r \in R_M, v \in I_r, v_1 \in O_r \quad (5.19)$$

$$E_{iv} \leq E_{iv_1} + H \cdot (1 - Z_{iv}) \quad i \in T, r \in R_M, v \in I_r, v_1 \in O_r \quad (5.20)$$

$$E_{iv} \geq E_{iv_1} - H \cdot (1 - Z_{iv}) \quad i \in T, r \in R_M, v \in I_r, v_1 \in O_r \quad (5.21)$$

Nas operações de saída dos tanques de carga com duas possíveis correntes de saída R_{C2} os limites de vazão são assegurados pela restrição 5.22. Para demais operações, a Equação 5.23 controla os limites de vazão.

$$\underline{FR}_r \lambda_{iv} \leq \sum_{v \in O_r} V_{iv}^t \leq \overline{FR}_r \lambda_{iv} \quad i \in T, r \in R_{C2} \quad (5.22)$$

$$\underline{FR}_v D_{iv} \leq V_{iv}^t \leq \overline{FR}_v D_{iv} \quad i \in T, r \in R - R_{C2}, v \in O_r \quad (5.23)$$

A λ_{ir} cria uma grade de tempo comum à todas as operações de saída de um tanque de carga W_M . Para calcular o valor de λ_{ir} foram usadas restrições do tipo *Big-M*, que igualam S_{iv} e λ_{ir} caso duas operações simultâneas aconteçam na saída

do tanque, as equações 5.24, 5.25, 5.26, 5.27.

$$S_{iv} \leq S_{iv_1} + H \cdot (2 - Z_{iv} - Z_{iv_1}) \quad i \in T, r \in R_C, v \in O_r, v_1 \in O_r \quad (5.24)$$

$$S_{iv} \geq S_{iv_1} - H \cdot (2 - Z_{iv} - Z_{iv_1}) \quad i \in T, r \in R_C, v \in O_r, v_1 \in O_r \quad (5.25)$$

$$D_{iv} \leq \lambda_{ir} + H \cdot (2 - Z_{iv} - Z_{iv_1}) \quad i \in T, r \in R_C, v \in O_r, v_1 \in O_r \quad (5.26)$$

$$D_{iv} \geq \lambda_{ir} - H \cdot (2 - Z_{iv} - Z_{iv_1}) \quad i \in T, r \in R_C, v \in O_r, v_1 \in O_r \quad (5.27)$$

Caso não aconteçam operações de saída nos tanques de carga com duas possíveis correntes de saída R_{C2} , a duração do *slot* de prioridade no tanque λ_{ir} deve ser zero. A Equação 5.28 assegura essa condição.

$$\lambda_{ir} \leq H \cdot \sum_{v \in O_r} Z_{iv} \quad i \in T, r \in R_{C2} \quad (5.28)$$

As restrições de qualidade em base volumétrica e mássica são expressas pelas restrições 5.29 e 5.30, respectivamente. Na alimentação das UDAs deseja-se que a propriedade acidez não ultrapasse 1,31,3 mg_{KOH}/g.

$$\frac{x_{vk} V_{iv}^t}{\sum_{c \in C} x_{ck} V_{ivc}} \leq \overline{x_{vk} V_{iv}^t} \quad i \in T, v \in W_D, k \in K_v \quad (5.29)$$

$$\frac{x_{vk} \sum_{c \in C} \rho_c V_{ivc}}{\sum_{c \in C} x_{ck} \rho_c V_{ivc}} \leq \overline{x_{vk} \sum_{c \in C} \rho_c V_{ivc}} \quad i \in T, v \in W_D, k \in K_m \quad (5.30)$$

A igualdade de composição da mistura dentro dos recursos e em suas respectivas operações de saída é assegurada pela restrição não-linear 5.31. Ambos lados dessa restrição são compostos por termos bilineares, i.e. um produto entre duas variáveis.

$$V_{ivc} L_{ir}^t = L_{irc} V_{iv}^t \quad i \in T, r \in R - R_M, v \in O_r, c \in C \quad (5.31)$$

Limites de inventário total e por petróleo ao fim da programação são estabelecidos pelas restrições 5.32 e 5.33, respectivamente.

$$\underline{L}_r^t \leq L_{0r}^t + \sum_{i \in T} \sum_{v \in I_r} V_{iv}^t - \sum_{i \in T} \sum_{v \in O_r} V_{iv}^t \leq \overline{L}_r^t, \quad r \in R_P \cup R_C \quad (5.32)$$

$$0 \leq L_{0rc} + \sum_{i \in T} \sum_{v \in I_r} V_{ivc} - \sum_{i \in T} \sum_{v \in O_r} V_{ivc} \leq \overline{L}_{rc}^t, \quad i \in T, r \in R_P \cup R_C, c \in C \quad (5.33)$$

Para gerenciar as multi-sequências de operação, o modelo MOS permite que apenas operações sobrepostas sejam atribuídas a um mesmo *slot* de prioridade. A restrição 5.34 assegura essa prerrogativa.

$$\sum_{v \in W'} Z_{iv} \leq 1 \quad i \in T, W' \in \text{clique}(G_{NO}) \cup \text{clique}(G_{NOTR}) \quad (5.34)$$

A restrição de quebra de simetria 5.35 alega que uma operação v não pode ser atribuída ao *slot* de prioridade i se nenhuma outra operação não-sobreposta a ela for atribuída ao slot de prioridade precedente $i - 1$.

$$Z_{iv} \leq \sum_{v' \in W} Z_{(i-1)v'} \quad i \in T, i \neq 1, v \in W \quad NO_{vv'} = 1 \quad (5.35)$$

A cronometragem de operações não-sobrepostas sem tempo de preparação foi realizada pela equação 5.36.

$$\sum_{v \in W'} E_{i_1 v} + \sum_{i \in T, i_1 < i < i_2} \sum_{v \in W'} D_{iv} \leq \sum_{v \in W'} S_{i_2 v} + H \cdot \left(1 - \sum_{v \in W'} Z_{i_2 v}\right) \\ i_1, i_2 \in T, i_1 < i_2, W' \in \text{clique}(G_{NO}) \quad (5.36)$$

Com o propósito de evitar soluções redundantes quando o modelo é resolvido com quantidades diferentes de *slots* de prioridade, qualquer solução que não utilize todos os *slots* de prioridade é rejeitada através da restrição 5.37

$$\sum_{v \in W} Z_{iv} \geq 1 \quad i \in T \quad (5.37)$$

Retomando as contribuições desse trabalho em termos de modelagem, destacam-se a modelagem dos misturadores (Equações 5.17, 5.15, 5.16, 5.18, 5.19, 5.20, 5.21), a modelagem de tanques de carga R_C com até duas correntes de saída (Equações 5.28, 5.24, 5.25, 5.26, 5.27, 5.22), as restrições de qualidade impostas nas operações de destilação (Equações 5.29, 5.30) e a restrição de duração mínima de operações (Equação 5.6).

Na modelagem base apresentada nesta seção foram adicionadas quinze novas equações ao modelo de Mouret et al. (2011). Doze equações foram mantidas iguais à da formulação original (Equações 5.2, 5.37, 5.34, 5.7, 5.8, 5.5, 5.3, 5.4, 5.10, 5.11, 5.9, 5.35). As demais foram modificadas para condizer com o sistema estudado. As Equações apresentadas nas próximas seções também são contribuições desse trabalho.

Modelagem com descarregamentos não-sobrepostos

Para cronometrar operações não sobrepostas que envolvem tempo de preparação usou-se a Equação 5.38.

$$\begin{aligned} \sum_{v \in W'} E_{i_1 v} + \sum_{i \in T, i_1 < i < i_2} \sum_{v \in W'} D_{iv} + \sum_{v \in W'} \sum_{w \in W'} TR_{v,w} \cdot Z_{i_1 v} \leq \sum_{v \in W'} S_{i_2 v} + \\ (H + \frac{1}{|R_P|} \sum_{v \in W'} \sum_{w \in W'} TR_{v,w}) \cdot (1 - \sum_{v \in W'} Z_{i_2 v}) \\ i_1, i_2 \in T, i_1 < i_2, (W_1; W_2) \in \text{clique}(G_{NOTR}) \end{aligned} \quad (5.38)$$

O parâmetro $TR_{v_1 v_2}$ representa o tempo de preparação entre as operações $v_1 \in W_U$ e $v_2 \in W_M$, e é igual a 24 h.

Modelagem com descarregamentos sobrepostos

Na modelagem com com descarregamentos sobrepostos a equação 5.39 foi usada para cronometrar operações não-sobrepostas com tempo de preparação.

$$\begin{aligned} E_{i_1 v_1} + E_{i_1 v_2} + \sum_{i \in T, i_1 < i < i_2} (D_{iv_1} + D_{iv_2}) + TR_{v_1 v_2} \cdot Z_{i_1 v_1} + TR_{v_2 v_1} \cdot Z_{i_1 v_2} \leq \\ S_{i_2 v_1} + S_{i_2 v_2} + (H + TR_{v_1 v_2} + TR_{v_2 v_1}) \cdot (1 - Z_{i_2 v_1} - Z_{i_2 v_2}) \\ i_1, i_2 \in T, i_1 < i_2, v_1 \in W', v_2 \in W'', W', W'' \in \text{clique}(G_{NOTR}) \end{aligned} \quad (5.39)$$

O parâmetro $TR_{v_1 v_2}$ representa o tempo de preparação entre as operações $v_1 \in W_U$ e $v_2 \in W_M$, e é igual a 24 h.

Uma vez que as operações de descarregamento W_U são sobrepostas nesse modelo, é necessário adicionar a restrição 5.40 para evitar que descarregamentos oriundos de parcelas distintas $v_1 \in O_{r_1}, v_2 \in O_{r_2}, r_1, r_2 \in R_P, r_1 \neq r_2$ para um determinado tanque de carga $r \in R_C$ sejam atribuídos ao mesmo *slot* de prioridade $i \in T$.

$$\sum_{v \in W_U} Z_{iv} \leq 1 \quad i \in T, r \in R_C, v \in I_r \quad (5.40)$$

Outra consequência do uso de descarregamentos sobrepostos na modelagem é que a sequência dos *slots* de prioridade nos tanques de carga R_C pode não ser

crescente. Se isso acontece os balanços de massa nos tanques – Equações 5.12 e 5.13 – não fazem sentido. As restrições 5.41 e 5.42 asseguram ordem crescente dos *slots* de prioridade nos tanques de carga e garantem, portanto, consistência dos balanços de massa nesses recursos.

$$\sum_{v \in I_r} E_{i_1 v} + \sum_{i \in T, i_1 < i < i_2} \sum_{v \in I_r} D_{iv} + \sum_{i \in T, i_1 < i < i_2} \lambda_{ir} \leq \sum_{v \in I_r} S_{i_2 v} + H \cdot \left(1 - \sum_{v \in I_r} Z_{i_2 v}\right) \\ i_1, i_2 \in T, i_1 < i_2, r \in R_{C2} \quad (5.41)$$

$$\sum_{v \in I_r} E_{i_1 v} + \sum_{i \in T, i_1 < i < i_2} \sum_{v \in I_r} D_{iv} + \sum_{i \in T, i_1 < i < i_2} \sum_{v \in O_r} D_{iv} \leq \sum_{v \in I_r} S_{i_2 v} + H \cdot \left(1 - \sum_{v \in I_r} Z_{i_2 v}\right) \\ i_1, i_2 \in T, i_1 < i_2, r \in R_C - R_{C2} \quad (5.42)$$

Estratégia de solução

De um ponto de vista matemático, a maioria dos problemas de programação da produção encontrados em ambientes industriais são problemas de otimização complexos e de grande escala que raramente podem ser resolvidos até a otimalidade dentro de um tempo computacional razoável (MÉNDEZ et al., 2006). Geralmente há um compromisso entre qualidade da solução e tempo computacional despendido em encontrá-la. Um requisito desejável em modelos de planejamento e programação da produção é que o programador seja capaz de rodar vários casos antes de escolher um programa definitivo. Portanto, para servir aos propósitos do empreendimento esses problemas precisam ser solucionados em tempos computacionais razoáveis e, ainda assim, retornar boas soluções.

Ambos modelos foram implementados e resolvido em GAMS (BROOKE et al., 1998). Neste trabalho usou-se a decomposição MILP-NLP proposta por Mouret et al. (2009) para resolver os modelos MINLP. A Figura 5.7 ilustra a estratégia de solução.

Primeiro resolve-se o problema MILP, composto de todas as restrições exceto a restrição não-linear de composição, Equação 5.31. Posteriormente fixa-se as variáveis binárias de atribuição Z_{iv} nos níveis da solução encontrada e resolve-se o problema NLP associado. Em outras palavras, o NLP é resolvido usando a solução inteira do MILP. A solução obtida na segunda etapa pode não ser o ótimo global do

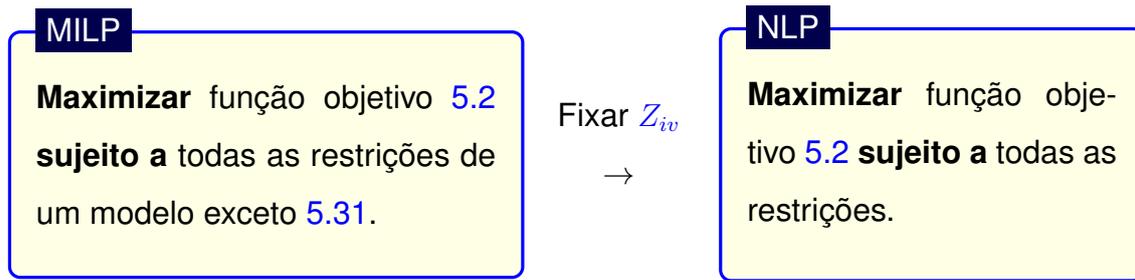


Figura 5.7: Estratégia de decomposição MILP-NLP.

problema completo, e uma *estimativa do gap de otimalidade com estratégia de decomposição MILP-NLP em % (GAPd)* pode ser calculada com as soluções do problema MILP e NLP, de acordo com a Equação 5.43 (MOURET et al., 2009).

$$\text{GAPd} = 100 \cdot \frac{\text{Função objetivo MILP} - \text{Função objetivo NLP}}{\text{Função objetivo MILP}} \quad (5.43)$$

Mouret et al. (2009) argumentam que o modelo NLP pode ser solucionado ou por um *solver* global, como o BARON, ou por *solvers* locais, como o CONOPT. No primeiro caso o *solver* retornará a solução global do NLP, se factível. No segundo caso, diferentes ótimos locais podem ser obtidos, a depender da solução inteira fixada. Inclusive, com a mesma solução inteira é possível obter soluções diferentes com CONOPT e BARON, pois o problema é não convexo e o CONOPT utiliza um algoritmo de busca local. Com algoritmos de busca local não há garantia de que o procedimento iterativo da Figura 5.7 retorne o ótimo global ou até mesmo uma solução viável para o problema original. Segundo Mouret et al. (2009) se o NLP retornar uma solução idêntica à do MILP, esta é o ótimo global.

A solução MILP pode não levar à uma solução NLP factível. Afim de contornar tais situações usou-se o *pool* de soluções MILP. O *pool* de soluções é uma ferramenta do *solver* MILP, o CPLEX, que armazena um determinado número de soluções MILP a ser escolhido pelo usuário (EXTENSIONS, 2001). Depois de armazenadas, todas as soluções MILP do *pool* são usadas como solução inteira do problema NLP, gerando tantas soluções NLP quanto soluções MILP no *pool*. Dessa forma é menos provável que uma solução NLP inviável seja obtida. A Figura 5.8 ilustra a estratégia de solução MILP-NLP com o *pool* de soluções MILP.

Primeiramente um programa MILP principal é solucionado. Se o programa MILP principal encontra soluções viáveis, várias soluções são armazenadas para se-

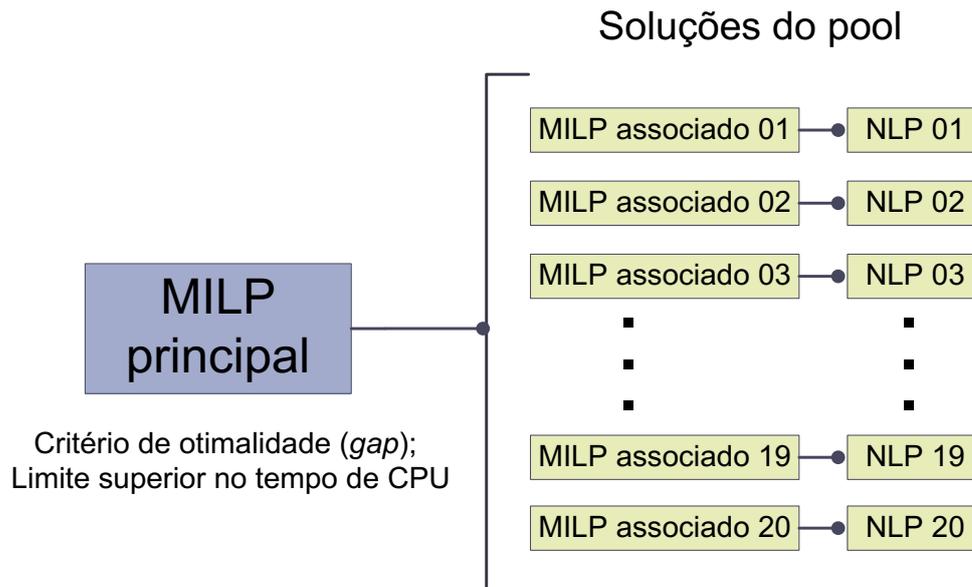


Figura 5.8: Pool de soluções do *solver* MILP.

Tabela 5.6: Parâmetros do *pool*.

<i>solnpoolintensity</i>	2
<i>solnpoolpop</i>	2
<i>solnpoolcapacity</i>	20
<i>solnpoolgap</i>	0,4
<i>populatelim</i>	100
<i>solnpoolreplace</i>	2

rem, posteriormente, usadas como solução inteira para vários programas NLP. Ao final da estratégia de decomposição MILP-NLP adota-se a solução NLP de maior função objetivo. É importante destacar que apenas a solução NLP de maior função objetivo é reportada e discutida.

As várias soluções MILP são selecionadas de acordo com os parâmetros do *pool*. A Tabela 5.6 exibe os parâmetros do *pool* utilizados na estratégia de solução.

O significado de cada parâmetro da Tabela 5.6 pode ser encontrado em Brooke et al. (1998). O parâmetro *solnpoolintensity* controla o *trade-off* entre número de soluções geradas para o *pool* e a quantidade necessária de memória e tempo. O parâmetro *solnpoolpop* ativa um procedimento de triagem para popular o *pool*. O parâmetro *solnpoolcapacity* determina quantas soluções serão armazenadas no *pool*.

O parâmetro *solnpoolgap* fixa uma tolerância relativa no valor da função objetivo em relação ao MILP principal para admitir soluções no *pool*. O parâmetro *populatelim* determina um número máximo de soluções MILP que serão analisadas pelo *solver* no procedimento de triagem. O parâmetro *solnpoolreplace* escolhe um critério de substituição segundo o qual as soluções armazenadas no *pool* são substituídas conforme o *solver* encontra mais soluções factíveis. Um valor de *solnpoolreplace* igual a 2 substitui as soluções de forma a gerar um conjunto o mais diverso possível.

Portanto, segundo as configurações do *pool* exibidas na Tabela 5.6, 100 soluções são analisadas pelo *solver* no processo de triagem, das quais apenas as 20 mais diferentes entre si são escolhidas. Todas as soluções armazenadas no *pool* são até 40 % piores que a do MILP principal em termos de função objetivo. Essa tolerância é controlada pelo parâmetro *solnpoolgap*.

Nessa dissertação apenas 20 soluções foram selecionadas para o *pool* com a finalidade de se obter bons tempos computacionais na etapa NLP. Outras abordagens podem ser usadas para resolver o problema. Uma abordagem mais pragmática é armazenar muitas soluções no *pool*, e solucionar o problema com faixas mais justas de *slots* de prioridade n . Armazenar mais soluções no *pool* aumenta a possibilidade de encontrar o ótimo global com estratégia de decomposição MILP-NLP.

A quantidade de *slots* de prioridade necessária para resolver o modelo com descarregamentos sobrepostos é menor que a quantidade de *slots* de prioridade necessária para resolver o modelo com descarregamentos não-sobrepostos. Após testes preliminares, faixas diferentes de $|T|$ foram selecionadas para os testes. Na modelagem com operações de descarregamento W_U não-sobrepostas cada cenário foi testado com $n = \{5, 6, 7, 8, 9\}$ *slots de prioridade*. Na modelagem com operações de descarregamento W_U sobrepostas cada cenário foi testado com $n = \{3, 4, 5, 6, 7\}$ *slots de prioridade*.

Para determinar a quantidade de *slots* de prioridade necessária para gerar o melhor programa, Mouret et al. (2011) adotam uma abordagem aditiva ilustrada no Algoritmo 1. O modelo é resolvido várias vezes, sempre com um incremento unitário no valor de $|T|$, a cardinalidade do conjunto de *slots*. Como critério de parada Mouret et al. (2011) sugere três abordagens.

A Tabela 5.7 exhibe os três critérios de parada sugeridos por Mouret et al.

Algoritmo 1 Abordagem aditiva para determinação do número de *slots* necessários para achar o melhor programa.

```

1: procedimento ABORDAGEM ADITIVA( $z^*$ , cutoff)
2:    $z^* \leftarrow \emptyset$ ;
3:   cutoff  $\leftarrow -\infty$ ;
4:    $n \leftarrow n_0$ ;
5:   enquanto [critério de parada] faça;
6:      $z \leftarrow \text{Maximizar}(\text{MODELO MOS}(n), \text{cutoff})$ ;
7:      $\delta \leftarrow z.\text{objval}() - z^*.\text{objval}()$ ;
8:     se  $\delta > 0$  então
9:        $z^* \leftarrow z$ 
10:      cutoff  $\leftarrow z^*.\text{objval}()$ 
11:    fim se
12:     $n \leftarrow n + 1$ 
13:  fim enquanto
14: fim procedimento

```

(2011) para determinação do melhor número de *slots* de prioridade. Nenhum deles garante otimalidade global da solução. No primeiro critério de parada uma tolerância ϵ no valor da função objetivo alcançada com números adjacentes de *slots* é usada para abortar ou não o procedimento de busca. A principal vantagem desse critério é que poucas iterações são necessárias para encontrar o melhor programa pelo Algoritmo 1. O segundo critério de parada é usar um limite superior no número de *slots* ($n < \bar{n}$). O último critério de parada sugerido por Mouret et al. (2011) é um limite superior no tempo computacional.

Nessa dissertação, os três critérios de parada da Tabela 5.7 foram usados para o Algoritmo 1. Para determinar o número de *slots* que gera o melhor programa usou-se o 1º critério de parada da Tabela 5.7. Ele foi escolhido por ser o mais utilizado na literatura (IERAPETRITOU; FLOUDAS, 1998a, IERAPETRITOU; FLOUDAS, 1998b, MARAVELIAS; GROSSMANN, 2003, SUNDARAMOORTHY; KARIMI, 2005) e também por demandar menos tempo computacional que o 2º critério, um requisito primordial para viabilizar a utilização de técnicas de otimização como ferramentas de

Tabela 5.7: Critérios de parada para determinação de $|T|$.

1º critério	incremento na função objetivo	$\epsilon > 0$
2º critério	número máximo de <i>slots</i> (descarregamentos não-sobrepostos)	$\bar{n} = 9$
	número máximo de <i>slots</i> (descarregamentos sobrepostos)	$\bar{n} = 7$
3º critério	tempo computacional máximo	2 h*

* na etapa MILP para cada corrida.

suporte às decisões de planejamento e programação.

Considerações sobre a precisão do modelo

A precisão de operações algébricas usada no modelo foi fixada em 1×10^{-5} através do parâmetro *zeros* do GAMS. Isso significa que em algumas operações, se o valor absoluto de um número for menor ou igual a 1×10^{-5} ele é considerado zero.

Para atenuar o efeito da grande diferença na ordem de grandeza das variáveis e evitar que um programa mal-condicionado fosse gerado, usou-se um fator de escala nas variáveis de recurso L_{ir}^t e L_{irc} e operação V_{iv}^t e V_{ivc} de 1×10^3 , isto é, as variáveis L_{ir}^t , L_{irc} , V_{iv}^t e V_{ivc} foram divididas por 1×10^3 . Portanto a precisão de variáveis de recurso e operação é de, $1 \times 10^{-5} \times 1 \times 10^3 = 1 \times 10^{-2}$, duas casas decimais. Para as variáveis de tempo S_{iv} , D_{iv} e E_{iv} , assim como para a variável de sincronização de operações de mistura λ_{ir} , a precisão é 1×10^{-5} .

Vazão de operações de mistura e destilação $W_M \cup W_D$ e vazão total na saída dos tanque de carga com duas possíveis correntes de saída R_{C2} foram calculadas após a corrida a partir da resposta do modelo para as variáveis V_{iv}^t e D_{iv} , pelas equações 5.44 e 5.45, respectivamente.

$$FR_{iv} = \frac{V_{iv}^t}{D_{iv}} \quad i \in T, v \in W_M \cup W_D \quad (5.44)$$

$$FR_{ir}^{C2} = \sum_{v \in O_r} \frac{V_{iv}^t}{D_{iv}} \quad i \in T, r \in R_{C2} \quad (5.45)$$

A precisão das respostas de vazão é 1×10^{-2} , igual à do fator menos preciso nas Equações 5.44 e 5.45.

As propriedades densidade e acidez x_{vk} também não são calculadas explicitamente pelo modelo. A propriedade TAN em $\text{mg}_{\text{KOH}}/\text{g}$ nas operações de destilação foi calculada pela equação 5.46 e a propriedade densidade ρ em g cm^{-3} foi calculada pela equação 5.47.

$$x_{vk} = \frac{\sum_{c \in C} x_{ck} \times x_{ck'} \times V_{ivc}}{\sum_{c \in C} x_{ck'} \times V_{ivc}} \quad k = \{\text{TAN}\}, k' = \{\rho\}, i \in T, v \in W_D \quad (5.46)$$

$$x_{vk} = \frac{\sum_{c \in C} x_{ck} \times V_{ivc}}{V_{iv}^t} \quad k = \{\rho\}, i \in T, v \in W_D \quad (5.47)$$

As propriedades de cada petróleo x_{ck} são valores conhecidos e assumidos constantes ao longo do horizonte de tempo. A precisão das respostas de propriedade chave x_{vk} é 1×10^{-2} , conforme equações 5.46 e 5.47.

Nota-se que em virtude do fator de escala de 1×10^3 usado para as variáveis de recurso L_{ir}^t e L_{irc} e operação V_{iv}^t e V_{ivc} , o valor da função objetivo apresentado em todas as tabelas nessa dissertação também está sob efeito desse fator, conforme a Equação 5.2. Ou seja, os valores de função objetivo estão multiplicados por 1×10^{-3} .

RESULTADOS E DISCUSSÃO

Ambos modelos foram implementados e resolvidos em **GAMS** com os *solvers* CPLEX 12.6.0.1 e CONOPT 3 nas etapas **MILP** e **NLP**, respectivamente. Os testes foram realizados em um *notebook* HP Pavilion dv7 com processador **Intel i7-3610QM** e 8 GB de RAM.

O critério de otimalidade do **MILP** principal¹ (**GAP**) foi definido para zero na bateria de testes realizada. Uma bateria de testes é composta por sessenta corridas: dez corridas para cada cenário, sendo seis com a modelagem de descarregamentos não-sobrepostos e seis com a modelagem de descarregamentos sobrepostos. Em cada corrida 20 soluções **NLP** foram obtidas no *pool*, e a melhor delas (melhor função objetivo) é reportada e discutida. O tempo computacional da corrida é igual à soma do tempo de solução do problema **MILP**, tempo do procedimento de triagem de soluções **MILP** para preencher o *pool*, e tempo de solução dos 20 problemas **NLP**. Parâmetros pré-definidos (*default*) dos *solvers*² foram usados em todas as corridas.

A notação usada para apresentar os resultados computacionais é:

- **status do modelo MILP principal (SM)** de acordo com Rosenthal (2014):
 - 8: solução factível encontrada;
 - 10: não existe solução factível;
 - 13: busca interrompida por falta de memória;
 - 14: solução não encontrada;

¹Em: *The GAMS Support Wiki – What is optca/optcr?*, <http://support.gams.com/doku.php?id=solver:what_is_optca_optcr>. Acesso em 03 de Julho de 2015.

²No cenário três com $n = 8$ usou-se o parâmetro do *solver* CPLEX *repeatpresolve* igual a 0 devido à um *bug* do CPLEX. Em: <<http://dw.developer-works.com/article/10050996/Negative+Gap+by+CPLEX!>>. Acesso em 15 de Julho de 2015.

- número de variáveis (NV);
- número de variáveis contínuas (NVC);
- número de variáveis binárias (NVB);
- número de equações (NE);
- número de equações lineares (NEL);
- número de equações não-lineares (NENL);
- função objetivo do programa inteiro-misto principal em 1×10^{-3} \$ (FO-MIP);
- número de nós do programa inteiro-misto linear principal (NN-MIP);
- função objetivo do programa não-linear em 1×10^{-3} \$ (FO-NLP);
- estimativa do *gap* de otimalidade com estratégia de decomposição MILP-NLP em % (GAPd) de acordo com a Equação 5.43 ³;
- tempo computacional do programa inteiro-misto linear em minutos (t-MIP);
- tempo computacional do programa não-linear linear em minutos (t-NLP);
- tempo computacional total em minutos (t-TOTAL);
- tempo acumulado necessário para determinar o número de *slots* de prioridade que gera o melhor programa em minutos (t-AC).

Nenhum dos modelos testados nessa dissertação conseguiu solucionar o cenário um. O modelo com **descarregamentos não-sobrepostos** (DNS) procurou a solução na etapa MILP por 2 h, até atingir o 3º critério de parada da Tabela 5.7, e não achou solução para nenhum dos *slots* testados. No modelo com **descarregamentos sobrepostos** (DS) o programa MILP foi inviável para todos os *slots* testados. Por isso a discussão sobre o cenário um será suprimida desse capítulo.

O modelo com DNS foi capaz de encontrar soluções ótimas para todos os cenários e *slots*, exceto para o cenário um, para o cenário três com $n = 5$ e para o cenário quatro com $n = 9$. O cenário três com $n = 5$ foi inviável. No cenário quatro com $n = 9$ o algoritmo atingiu o 3º critério de parada da Tabela 5.7 – um limite superior de 2 h no tempo computacional – e não encontrou solução na etapa MILP.

³A função objetivo MILP usada para estimar o GAPd foi a do MILP principal.

O modelo com DS foi capaz de encontrar soluções ótimas para todos os cenários e *slots*, exceto para o cenário um, para o cenário cinco com $n = 7$ e para o cenário seis com $n = 3$. No cenário cinco com $n = 7$ o algoritmo atingiu o 3º critério de parada da Tabela 5.7 e não encontrou solução na etapa MILP. O cenário seis com $n = 3$ foi inviável.

Os resultados da programação de ambos modelos respeitaram as regras operacionais da Tabela 4.2 e as restrições de processo da Tabela 4.3. Os gráficos de Gantt, inventário nos tanques de carga R_C , vazão na saída dos tanques de carga R_C e perfil de acidez nas operações de destilação W_D do melhor programa para os seis cenários com as duas modelagens testadas estão dispostos no Apêndice C. Eles comprovam que as regras operacionais e restrições de processo foram respeitadas.

Performance

Na hora de escolher um modelo matemático para programação da produção que bem represente o sistema e retorne soluções viáveis que se encaixam no perfil de operação da planta, três critérios são de extrema importância: função objetivo, tempo de CPU e resultados da programação, que determinam o modo de operação da refinaria. Os dois primeiros resultados – função objetivo e tempo de CPU – são discutidos nessa seção, e os resultados da programação são discutidos na próxima seção.

É importante frisar que por se tratarem de modelos diferentes, com restrições diferentes, comparar a qualidade dos resultados obtidos é válido apenas para fins de projeto. Como o objetivo desse trabalho é avaliar a viabilidade do uso de técnicas de otimização aplicadas à programação do suprimento de petróleo, os resultados dos dois modelos são comparados a nível de projeto, isto é, para investigar a potencialidade de cada modelo como núcleo resolvidor para aplicativos industriais.

A comparação é feita por corrida, ou seja, a corrida de $n = 5$ do modelo com DNS foi comparada com a corrida de $n = 5$ do modelo com DS. Uma vez que a faixa de *slots* de tempo testada é diferente para as duas modelagens, essa comparação foi possível para $n = 5$, $n = 6$ e $n = 7$.

Cenário dois

No cenário dois há no sistema $33.556,147 \text{ m}^3 \text{ d}^{-1}$ disponíveis para destilação. A quantidade relativa de petróleo destilado no cenário dois foi igual à capacidade máxima de destilação do sistema, $28.507,2 \text{ m}^3 \text{ d}^{-1}$. O volume total destilado em sete dias foi $199.550,40 \times 10^3 \text{ m}^3$. Esse resultado foi obtido pelas duas modelagens testadas.

As Tabelas 6.1 e 6.2 mostram os resultados computacionais do cenário dois modelado com DNS e DS, respectivamente.

Tabela 6.1: Resultados computacionais do cenário dois com DNS.

n	SM	NVC	NVB	NEL	NENL	FO-MIP	NN-MIP	FO-NLP	GAPd	t-MIP	t-NLP	t-TOTAL
i5	8	7.711	200	5.515	1.690	46.500,09	15.470	45.597,09	1,94	2,30	1,31	3,61
i6	8	9.253	240	6.555	2.028	46.662,99	61.156	45.680,23	2,11	26,75	1,56	28,32
i7	8	10.795	280	7.612	2.366	46.705,07	49.053	45.605,85	2,35	24,14	3,00	27,14
i8	8	12.337	320	8.686	2.704	46.705,07	528.328	45.624,43	2,31	89,10	2,79	91,89
i9	8	13.879	360	9.777	3.042	46.609,96	73.766	44.810,03	3,86	69,15	3,04	72,18

* SM para *status* do modelo, NVC para número de variáveis contínuas, NVB para número de variáveis binárias, NEL para número de equações lineares, NENL para número de equações não-lineares, FO-MIP para função objetivo do programa MILP, NN-MIP para número de nós do programa MILP, FO-NLP para função objetivo do programa NLP, GAPd para estimativa do *gap* de otimalidade, t-MIP para tempo de CPU da etapa MILP, t-NLP para tempo de CPU da etapa NLP e t-TOTAL para tempo de CPU total.

Tabela 6.2: Resultados computacionais do cenário dois com DS.

n	SM	NVC	NVB	NEL	NENL	FO-MIP	NN-MIP	FO-NLP	GAPd	t-MIP	t-NLP	t-TOTAL
i3	8	4.627	120	3.819	1.014	46.488,32	24.928	45.610,22	1,89	1,49	0,20	1,69
i4	8	6.169	160	5.086	1.352	46.727,98	26.291	45.421,01	2,80	2,61	0,48	3,10
i5	8	7.711	200	6.445	1.690	46.750,23	57.092	45.555,02	2,54	10,82	1,48	12,30
i6	8	9.253	240	7.896	2.028	46.758,41	65.611	45.486,89	2,72	23,45	1,49	24,93
i7	8	10.795	280	9.439	2.366	46.756,37	116.045	45.591,83	2,49	103,21	2,58	105,79

* SM para *status* do modelo, NVC para número de variáveis contínuas, NVB para número de variáveis binárias, NEL para número de equações lineares, NENL para número de equações não-lineares, FO-MIP para função objetivo do programa MILP, NN-MIP para número de nós do programa MILP, FO-NLP para função objetivo do programa NLP, GAPd para estimativa do *gap* de otimalidade, t-MIP para tempo de CPU da etapa MILP, t-NLP para tempo de CPU da etapa NLP e t-TOTAL para tempo de CPU total.

O modelo com DNS não foi resolvido até a otimalidade (0% de *gap de otimalidade do programa inteiro-misto linear principal em % (GAP)*) para cenário dois com $n = 8$ e $n = 9$. O MILP principal foi interrompido por falta de memória, e a árvore de busca foi truncada em 0,09% e 0,30% de GAP, respectivamente. Alguns artifícios podem ser usados para contornar esse problema de memória no GAMS⁴.

⁴Em: *The GAMS Support Wiki – Error 1001: Out of Memory*, <<http://support.gams.com/doku>>.

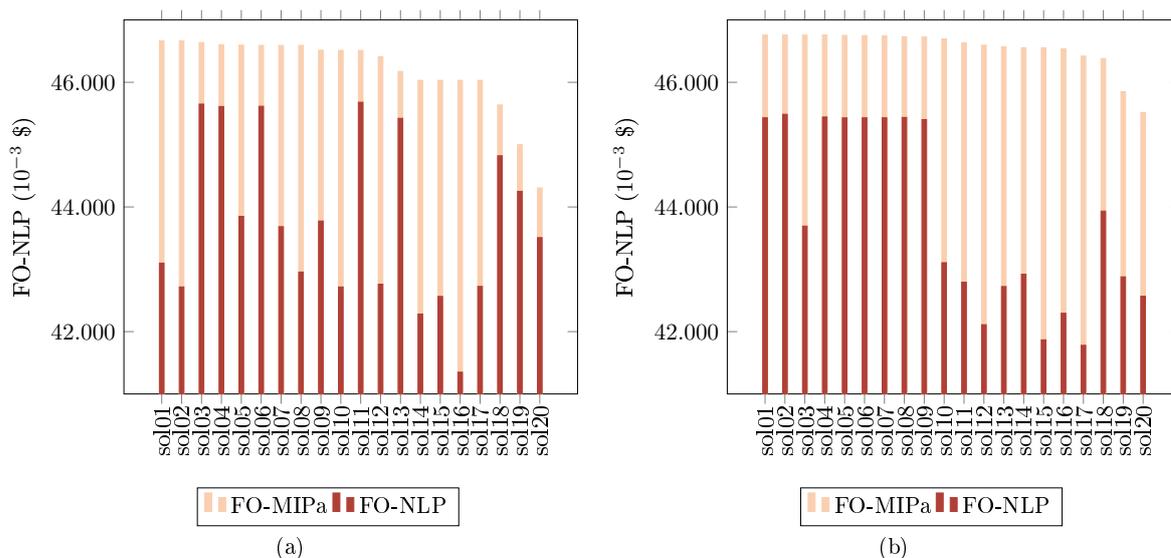


Figura 6.1: Visão geral do *pool* de soluções do cenário dois com $n = 6$ na modelagem com descarregamentos (a) não-sobrepostos e (b) sobrepostos.

De acordo com Mouret et al. (2011) a etapa MILP é a mais difícil e crucial do procedimento de decomposição. O cenário dois foi o único no qual a fase MILP da decomposição MILP-NLP foi a etapa limitante em todas as corridas, isto é, $t_{\text{MIP}} > t_{\text{NLP}}$. Isso pode ser explicado por três características do cenário dois: (i) enquanto nos outros cenários há nove tanques de carga R_C disponíveis para descarregamento de parcelas W_U , no cenário dois há apenas oito; (ii) enquanto nos outros cenários há trinta e seis tipos de petróleo C no sistema, no cenário dois há apenas vinte e seis tipos de petróleo C e, conseqüentemente, reduzidas possibilidades de mistura; (iii) a acidez média do sistema $\overline{\text{TAN}}_0$ é a segunda maior (0,87 $\text{mg}_{\text{KOH}}/\text{g}$), atrás apenas do cenário um.

Essas três características explicam o fato de que no cenário dois as decisões inteiras foram as mais onerosas e mais nós foram explorados pelo algoritmo *Branch & Cut* quando comparado com os demais cenários.

Cada linha das Tabelas 6.1 e 6.2 é uma corrida. Em cada corrida 20 soluções foram armazenadas no *pool* e somente a melhor solução NLP foi selecionada. A Figura 6.1 mostra uma visão geral das soluções no *pool* para as corridas de $n = 6$ nos dois modelos.

Observa-se na Figura 6.1 que a função objetivo do MILP associado é sempre

maior que a função objetivo **NLP**, pois a região factível do programa **NLP** é formada pelas mesmas restrições que formam a região factível do programa **MILP**, mais a restrição não-linear, Equação 5.31. Na Figura 6.1 (a) somente as soluções 03, 04, 06, 11, 13, 18, 19 e 20 foram viáveis, e a melhor solução viável foi a 11. Na Figura 6.1 (b) somente as soluções 01, 02, 04, 05, 06, 07, 08 e 09 foram viáveis, e a melhor solução viável foi a 02. Somente a melhor solução viável é reportada nas Tabelas 6.1 e 6.2 e discutida.

Na Tabela 6.3 pode-se visualizar o valor da função objetivo **NLP** encontrado pelos dois modelos quando solucionados com a mesma quantidade de *slots*. A Tabela 6.4 exibe o tempo computacional total **t-TOTAL** requerido pelos dois modelos quando solucionados com a mesma quantidade de *slots*.

Tabela 6.3: Comparação entre os modelos em termos de função objetivo para o cenário dois.

n	DNS	DS	diferença
i3		45.610,22	
i4		45.421,01	
i5	45.597,09	45.555,02	42,07
i6	45.680,23	45.486,89	193,35
i7	45.605,85	45.591,83	14,03
i8	45.624,43		
i9	44.810,03		

Observa-se na Tabela 6.3 que, para o cenário dois, o modelo com **DNS** retornou funções objetivo maiores que as do modelo com **DS** quando solucionado com o mesmo número de *slots*. Essa diferença foi de até $193,35 \times 10^3$ \$. Na Tabela 6.4 observa-se que o tempo computacional total **t-TOTAL** do modelo com **DNS** também foi melhor, exceto para $n = 6$. Nas corridas de $n = 6$ o cenário dois foi solucionado em tempos computacionais semelhantes pelas duas modelagens.

O **GAPd** é uma estimativa de quão distante a solução **NLP** encontrada está do ótimo global. Segundo Mouret et al. (2009) se a solução da etapa **NLP** for igual à da etapa **MILP**, ela é o ótimo global. Ou seja, se **GAPd** = 0, então um ótimo global foi encontrado. Nenhum ótimo global foi encontrado nessa dissertação.

Tabela 6.4: Comparação entre os modelos em termos de tempo computacional para o cenário dois.

n	DNS	DS	diferença
i3		1,69	
i4		3,10	
i5	3,61	12,30	-8,69
i6	28,32	24,93	3,38
i7	27,14	105,79	-78,65
i8	91,89		
i9	72,18		

Nas corridas de $n = 5$, $n = 6$ e $n = 7$ o **GAPd** do modelo com **DNS** foi 1,94 %, 2,11 % e 2,35 %, respectivamente. Para o modelo com **DS**, o **GAPd** das corridas de $n = 5$, $n = 6$ e $n = 7$ foi 2,54 %, 2,72 % e 2,49 %, respectivamente. Em todas as três corridas o modelo com **DNS** obteve **GAPd** menores. Dois fatores contribuíram para que o *gap* do modelo com **DNS** fosse menor: (i) a solução **NLP** do modelo com **DNS** foi maior que a do modelo com **DS** e (ii) a solução **MILP**⁵ do modelo com **DNS** foi menor que a do modelo com **DS** (exceto para $n = 5$). Esse resultado sugere que usar **DNS** gera **LR** mais justas.

O melhor programa para o cenário dois com o modelo **DNS** foi o de $n = 6$ e, com o modelo **DS** foi o de $n = 3$. Entende-se por melhor programa aquele que retornou a melhor função objetivo de acordo com o Algoritmo 1 e o 1º critério de parada da Tabela 5.7. O tempo computacional da corrida correspondente ao melhor programa foi 28,32 min para o modelo com **DNS** e 1,69 min para o modelo com **DS**.

Ao usar o Algoritmo 1 para determinar o tempo computacional acumulado necessário para encontrar o melhor programa **t-AC**, o modelo com **DNS** levou 59,07 min. Nesse caso **t-AC** é a soma de **t-TOTAL** das corridas de $n = 5$, $n = 6$ e $n = 7$. O tempo acumulado **t-AC** do modelo com **DS** foi melhor, 4,79 min, que é a soma das corridas de $n = 3$ e $n = 4$.

⁵O **GAPd** é calculado com o valor da função objetivo do **MILP** principal como limite superior.

Cenário três

No cenário três há no sistema $28.390,829 \text{ m}^3 \text{ d}^{-1}$ disponíveis para destilação. O cenário três é o único entre os cenários otimizados no qual o volume relativo disponível para destilação é menor que a capacidade máxima de destilação do sistema ($28.507,2 \text{ m}^3 \text{ d}^{-1}$). Todo o volume relativo disponível para destilação foi processado no cenário três. O volume total destilado em dez dias foi $283.908,3 \times 10^3 \text{ m}^3$. Esse resultado foi obtido pelas duas modelagens testadas.

As Tabelas 6.5 e 6.6 mostram os resultados computacionais do cenário três modelado com DNS e DS, respectivamente.

Tabela 6.5: Resultados computacionais do cenário três com DNS.

n	SM	NVC	NVB	NEL	NENL	FO-MIP	NN-MIP	FO-NLP	GAPd	t-MIP	t-NLP	t-TOTAL
i5	10	15.536	315	8.499			84			0,11		0,11
i6	8	18.643	378	10.058	3.240	60.873,47	17.380	59.823,50	1,72	6,13	10,22	16,35
i7	8	21.750	441	11.636	3.780	66.920,14	27.500	65.891,75	1,54	31,26	20,52	51,77
i8	8	24.857	504	13.233	4.320	66.921,98	4.270	65.899,90	1,53	3,49	24,36	27,85
i9	8	27.964	567	14.849	4.860	66.921,98	3.015	65.850,89	1,60	1,37	45,39	46,76

* SM para status do modelo, NVC para número de variáveis contínuas, NVB para número de variáveis binárias, NEL para número de equações lineares, NENL para número de equações não-lineares, FO-MIP para função objetivo do programa MILP, NN-MIP para número de nós do programa MILP, FO-NLP para função objetivo do programa NLP, GAPd para estimativa do gap de otimalidade, t-MIP para tempo de CPU da etapa MILP, t-NLP para tempo de CPU da etapa NLP e t-TOTAL para tempo de CPU total.

Tabela 6.6: Resultados computacionais do cenário três com DS.

n	SM	NVC	NVB	NEL	NENL	FO-MIP	NN-MIP	FO-NLP	GAPd	t-MIP	t-NLP	t-TOTAL
i3	8	9.322	189	6.101	1.620	66.640,70	52.485	65.480,87	1,74	3,33	1,06	4,40
i4	8	12.429	252	8.139	2.160	66.921,98	124.423	65.815,85	1,65	11,50	2,67	14,18
i5	8	15.536	315	10.344	2.700	66.921,98	8.979	65.875,13	1,56	1,56	6,31	7,86
i6	8	18.643	378	12.716	3.240	66.921,98	15.807	65.876,19	1,56	3,74	10,20	13,94
i7	8	21.750	441	15.255	3.780	66.921,98	12.532	65.732,48	1,78	6,96	19,97	26,92

* SM para status do modelo, NVC para número de variáveis contínuas, NVB para número de variáveis binárias, NEL para número de equações lineares, NENL para número de equações não-lineares, FO-MIP para função objetivo do programa MILP, NN-MIP para número de nós do programa MILP, FO-NLP para função objetivo do programa NLP, GAPd para estimativa do gap de otimalidade, t-MIP para tempo de CPU da etapa MILP, t-NLP para tempo de CPU da etapa NLP e t-TOTAL para tempo de CPU total.

Conforme discutido no Capítulo 5.1 o cenário três gera o problema de maior dimensão porque possui mais operações W em comparação aos outros cenários. Por isso ele gerou modelos com mais variáveis. Por exemplo, no modelo do cenário três com DNS e $n = 5$ há 15.851 variáveis. O número de variáveis para os cenários dois,

quatro, cinco e seis com $n = 5$ e **DNS** são 7.911, 11.701, 13.776 e 9.626, respectivamente. No modelo do cenário três com **DS** e $n = 3$ há 9.511 variáveis. O número variáveis para os cenários dois, quatro, cinco e seis com $n = 3$ e **DS** são 4.747, 7.021, 8.266 e 5.776, respectivamente.

Observa-se nas Tabelas 6.5 e 6.6 que o tempo de solução da etapa **MILP** superou o da etapa **NLP** somente para a corrida de $n = 7$ no modelo com **DNS** e nas corridas de $n = 3$ e $n = 4$ no modelo com **DS**. Nas demais corridas a etapa **NLP** foi a mais demorada. Armazenar muitas soluções no *pool* aumenta a possibilidade de encontrar boas soluções **NLP**. Por outro lado, pode aumentar significativamente o tempo computacional desta etapa da decomposição.

Nos modelos investigados nessa dissertação o número de transições permitidas nas **UDAs** é limitado pela quantidade de *slots* de prioridade usada para resolver o modelo. Conforme discutido anteriormente, no modelo com **DS** são permitidas menos transições, porque são necessários menos *slots* para encontrar uma solução viável. Por outro lado, no modelo com **DS** as parcelas podem ser fragmentadas em mais operações de descarregamento, pois operações de descarregamento oriundas de parcelas diferentes podem se sobrepor, isto é, acontecer no mesmo *slot* de prioridade. Consequentemente, mais tanques podem receber descarregamento de parcelas. Sendo assim, o sistema como um todo precisará de mais tempo de preparação, já que um tanque não pode operar por pelo menos 24 h depois que é carregado.

Certamente a fragmentação ou não de parcelas é decisão da otimização. A restrição de atribuição mínima, Equação 5.37, é usada nos dois modelos e exige que pelo menos uma operação $v \in W$ seja alocada em cada *slot* de prioridade. Como o número de operações de destilação não pode ser maior que n por causa da restrição de cardinalidade máxima, Equação 5.7, a única opção do modelo é alocar *slots* extras – *slots* nos quais não acontecem operações de mistura W_M nem de destilação W_D – para operações de descarregamento, provocando maior fragmentação de parcelas no modelo com **DS**. O número de operações de descarregamento realizadas pelo modelo com **DS** foi sempre maior que o número de operações de descarregamento realizadas pelo modelo com **DNS** nos seis cenários otimizados nessa dissertação.

Essa característica do modelo explica porque o número de nós explorados pelo algoritmo *Branch & Cut* foi sensivelmente maior no modelo com **DS** para $n = 3$

e $n = 4$ slots de prioridade. O programa MILP é mais difícil quando se pode realizar poucas operações de destilação em um sistema com tempo de preparação e pouco volume relativo disponível para destilação. Para $n = 4$, por exemplo, o modelo pode fazer no máximo 24 operações de descarregamento neste cenário (já contabilizando a restrição de duração mínima), e pode fazer no máximo 4 operações de destilação em cada UDA. Essa característica do modelo com DS pode gerar programas inviáveis para combinações de sistemas pequenos, horizontes de tempo curtos, pouco volume relativo disponível para destilação e poucos slots de prioridade. Outra explicação para a maior quantidade de nós explorados pode ser a relaxação mais pobre do modelo com DS, como mostra a Tabela 6.6.

Na Tabela 6.7 pode-se visualizar o valor da função objetivo NLP encontrado pelos dois modelos quando solucionados com a mesma quantidade de slots. A Tabela 6.8 exibe o tempo computacional total t-TOTAL requerido pelos dois modelos quando solucionados com a mesma quantidade de slots.

Tabela 6.7: Comparação entre os modelos em termos de função objetivo para o cenário três.

n	DNS	DS	diferença
i3		65.480,87	
i4		65.815,85	
i5		65.875,13	
i6	59.823,50	65.876,19	-6.052,69
i7	65.891,75	65.732,48	159,26
i8	65.899,90		
i9	65.850,89		

Observa-se na Tabela 6.7 que, quanto à função objetivo, os dois modelos proporcionam benefícios semelhantes para o negócio. Para $n = 6$ o modelo com DS encontrou uma função objetivo $6.052,69 \times 10^3$ \$ maior que a do modelo com DNS. Para $n = 7$ o modelo com DNS encontrou uma função objetivo $159,26 \times 10^3$ \$ maior que a do modelo com DS.

Por outro lado, observa-se na Tabela 6.8 que o modelo com DNS levou 24,85 min a mais que o modelo com DS para solucionar o cenário três com $n = 7$. Para $n = 6$ o

Tabela 6.8: Comparação entre os modelos em termos de tempo computacional para o cenário três.

n	DNS	DS	diferença
i3		4,40	
i4		14,18	
i5	0,11	7,86	
i6	16,35	13,94	2,41
i7	51,77	26,92	24,85
i8	27,85		
i9	46,76		

tempo computacional dos dois modelos foi semelhante.

O incremento da função objetivo do modelo DNS com $n = 6$ em comparação com $n = 5$ foi muito grande, $6.068,25 \times 10^3$ \$. Isso sugere que no cenário três é necessário segmentar o descarregamento de parcelas em várias operações para obter um bom programa. Segmentar as parcelas é um ponto forte do modelo com DS. Neste cenário o modelo com DNS realizou sete operações de descarregamento, e o modelo com DS realizou onze operações de descarregamento.

Não há comparação entre as corridas de $n = 5$ porque o modelo com DNS foi inviável. Ou seja, não existe solução inteira quando se limita em $n = 5$ o número de operações de descarregamento W_U que podem acontecer durante o horizonte de tempo. Esse fato confirma a necessidade de segmentar as parcelas no cenário três.

Então, é razoável presumir que o modelo com DS seja mais adequado para solucionar o cenário três, já que segmenta o descarregamento de parcelas em mais operações. No entanto, isso não foi observado na prática: o melhor programa para o cenário três foi encontrado pelo modelo com DNS na corrida de $n = 8$.

O GAPd dos dois modelos foram semelhantes. Nas corridas de $n = 6$ e $n = 7$ o GAPd do modelo com DNS foi 1,72% e 1,54%, respectivamente. Para o modelo com DS, o GAPd das corridas de $n = 6$ e $n = 7$ foi 1,56% e 1,78%, respectivamente.

O melhor programa para o cenário três com o modelo DNS foi o de $n = 8$ e, com o modelo DS foi o de $n = 6$. Entende-se por melhor programa aquele que retornou a melhor função objetivo de acordo com o Algoritmo 1 e o 1º critério de

parada da Tabela 5.7. O tempo computacional da corrida correspondente ao melhor programa foi 27,85 min para o modelo com DNS e 13,94 min para o modelo com DS.

O tempo computacional acumulado necessário para encontrar o melhor programa t-AC do modelo com DNS foi 142,83 min, a soma dos tempos das corridas de $n = 5$, $n = 6$, $n = 7$, $n = 8$ e $n = 9$. O tempo acumulado do modelo com DS foi 67,30 min, que é a soma dos tempos das corridas de $n = 3$, $n = 4$, $n = 5$, $n = 6$ e $n = 7$.

Cenário quatro

No cenário quatro há no sistema $35.293,577 \text{ m}^3 \text{ d}^{-1}$ disponíveis para destilação. A quantidade relativa de petróleo destilado no cenário dois foi igual à capacidade máxima de destilação do sistema, $28.507,2 \text{ m}^3 \text{ d}^{-1}$. O volume total destilado em sete dias foi $199.550,4 \times 10^3 \text{ m}^3$. Esse resultado foi obtido pelas duas modelagens testadas.

As Tabelas 6.9 e 6.10 mostram os resultados computacionais do cenário quatro modelado com DNS e DS, respectivamente.

Tabela 6.9: Resultados computacionais do cenário quatro com DNS.

n	SM	NVC	NVB	NEL	NENL	FO-MIP	NN-MIP	FO-NLP	GAPd	t-MIP	t-NLP	t-TOTAL
i5	8	11.476	225	6.999	2.700	48.514,79	5.576	46.590,43	3,49	0,99	2,98	3,97
i6	8	13.771	270	8.284	3.240	48.559,41	7.841	46.732,09	3,72	4,41	6,40	10,81
i7	8	16.066	315	9.588	3.780	48.569,13	9.583	47.031,54	3,17	6,86	8,47	15,33
i8	8	18.361	360	10.911	4.320	48.275,22	62.539	46.722,62	3,22	120,00	8,42	128,43
i9	14	20.656	405	12.253			86.061			120,00		120,00

* SM para status do modelo, NVC para número de variáveis contínuas, NVB para número de variáveis binárias, NEL para número de equações lineares, NENL para número de equações não-lineares, FO-MIP para função objetivo do programa MILP, NN-MIP para número de nós do programa MILP, FO-NLP para função objetivo do programa NLP, GAPd para estimativa do gap de otimalidade, t-MIP para tempo de CPU da etapa MILP, t-NLP para tempo de CPU da etapa NLP e t-TOTAL para tempo de CPU total.

Na modelagem com DNS o cenário quatro não foi resolvido até a otimalidade para $n = 8$. O terceiro critério de parada da Tabela 5.7 foi atingido, e a árvore de busca foi truncada em 0,15% de GAP. Esse acontecimento não influenciou no tempo acumulado necessário para encontrar o melhor programa t-AC.

Na Tabela 6.11 pode-se visualizar o valor da função objetivo NLP encontrado pelos dois modelos quando solucionados com a mesma quantidade de slots. A Tabela 6.12 exibe o tempo computacional total t-TOTAL requerido pelos dois modelos quando solucionados com a mesma quantidade de slots.

Tabela 6.10: Resultados computacionais do cenário quatro com DS.

n	SM	NVC	NVB	NEL	NENL	FO-MIP	NN-MIP	FO-NLP	GAPd	t-MIP	t-NLP	t-TOTAL
i3	8	6.886	135	4.867	1.620	48.555,34	4.716	46.835,03	3,54	0,27	0,75	1,01
i4	8	9.181	180	6.413	2.160	48.569,13	3.333	46.872,49	3,49	0,38	1,89	2,27
i5	8	11.476	225	8.064	2.700	48.569,13	4.633	46.878,36	3,48	1,19	4,24	5,43
i6	8	13.771	270	9.820	3.240	48.569,13	2.942	46.945,64	3,34	1,02	7,15	8,17
i7	8	16.066	315	11.681	3.780	48.569,13	23.614	46.907,02	3,42	15,53	13,02	28,55

* SM para *status* do modelo, NVC para número de variáveis contínuas, NVB para número de variáveis binárias, NEL para número de equações lineares, NENL para número de equações não-lineares, FO-MIP para função objetivo do programa MILP, NN-MIP para número de nós do programa MILP, FO-NLP para função objetivo do programa NLP, GAPd para estimativa do *gap* de otimalidade, t-MIP para tempo de CPU da etapa MILP, t-NLP para tempo de CPU da etapa NLP e t-TOTAL para tempo de CPU total.

Tabela 6.11: Comparação entre os modelos em termos de função objetivo para o cenário quatro.

n	DNS	DS	diferença
i3		46.835,03	
i4		46.872,49	
i5	46.590,43	46.878,36	-287,92
i6	46.732,09	46.945,64	-213,55
i7	47.031,54	46.907,02	124,53
i8	46.722,62		
i9			

Tabela 6.12: Comparação entre os modelos em termos de tempo computacional para o cenário quatro.

n	DNS	DS	diferença
i3		1,01	
i4		2,27	
i5	3,97	5,43	-1,46
i6	10,81	8,17	2,65
i7	15,33	28,55	-13,22
i8	128,43		
i9	120,00		

Observa-se na Tabela 6.11 que, em termos de função objetivo, o modelo com DS teve melhor desempenho que o modelo com DNS. Na corrida de $n = 5$ sua função

objetivo foi $287,92 \times 10^3$ maior que a do modelo com **DNS**. Em termos de tempo computacional, Tabela 6.12, o modelo com **DNS** teve o melhor desempenho. Na corrida de $n = 7$ o modelo com **DNS** foi solucionado em 13,22 min a menos que o modelo com **DS**. Esse resultado está condizente com a argumentação de Mouret et al. (2011), de que cliques com maior cardinalidade levam à modelos mais compactos. Modelos compactos (com menos equações) podem ser solucionados mais rapidamente.

Observa-se na Tabela 6.11 uma tendência que também pode ser observada no cenário três (Tabela 6.7): conforme se aumenta o número de *slots* de prioridade n a diferença entre a função objetivo dos modelos com **DNS** e **DS** é reduzida. Nos cenários três e quatro a função objetivo do modelo com **DNS** chega a superar a do modelo com **DS** em $n = 7$.

O **GAPd** dos dois modelos foram semelhantes. Nas corridas de $n = 5$, $n = 6$ e $n = 7$ o **GAPd** do modelo com **DNS** foi 3,49 %, 3,72 % e 3,17 %, respectivamente. Para o modelo com **DS**, o **GAPd** das corridas de $n = 5$, $n = 6$ e $n = 7$ foi 3,48 %, 3,34 % e 3,42 %, respectivamente.

O melhor programa para o cenário quatro encontrado pelo modelo com **DNS** foi o de $n = 7$ e pelo modelo com **DS** foi o de $n = 6$. Entende-se por melhor programa aquele que retornou a melhor função objetivo de acordo com o Algoritmo 1 e o 1º critério de parada da Tabela 5.7. O tempo computacional da corrida correspondente ao melhor programa foi 15,33 min para o modelo com **DNS** e 8,17 min para o modelo com **DS**.

O tempo computacional acumulado necessário para encontrar o melhor programa **t-AC** do modelo com **DNS** foi 158,54 min, a soma dos tempos das corridas de $n = 5$, $n = 6$, $n = 7$ e $n = 8$. O tempo acumulado do modelo com **DS** foi 45,3 min, que é a soma dos tempos das corridas de $n = 3$, $n = 4$, $n = 5$, $n = 6$ e $n = 7$.

Cenário cinco

No cenário cinco há no sistema $50.170,148 \text{ m}^3 \text{ d}^{-1}$ disponíveis para destilação. A quantidade relativa de petróleo destilado no cenário dois foi igual à capacidade máxima de destilação do sistema, $28.507,2 \text{ m}^3 \text{ d}^{-1}$. O volume total destilado em seis dias foi $171.043,2 \times 10^3 \text{ m}^3$. Esse resultado foi obtido pelas duas modelagens testadas.

As Tabelas 6.13 e 6.14 mostram os resultados computacionais do cenário cinco modelado com DNS e DS, respectivamente.

Tabela 6.13: Resultados computacionais do cenário cinco com DNS.

n	SM	NVC	NVB	NEL	NENL	FO-MIP	NN-MIP	FO-NLP	GAPd	t-MIP	t-NLP	t-TOTAL
i5	8	13.506	270	7.794	2.700	41.646,25	2.946	39.942,48	4,09	0,80	5,19	5,99
i6	8	16.207	324	9.225	3.240	42.156,75	7.665	40.207,97	4,62	2,43	8,12	10,55
i7	8	18.908	378	10.675	3.780	42.036,04	11.037	40.053,95	4,72	3,60	9,69	13,29
i8	8	21.609	432	12.144	4.320	41.873,11	102.109	39.974,59	4,53	50,14	14,63	64,77
i9	8	24.310	486	13.632	4.860	41.454,70	58.877	39.823,70	3,93	47,84	12,79	60,63

* SM para *status* do modelo, NVC para número de variáveis contínuas, NVB para número de variáveis binárias, NEL para número de equações lineares, NENL para número de equações não-lineares, FO-MIP para função objetivo do programa MILP, NN-MIP para número de nós do programa MILP, FO-NLP para função objetivo do programa NLP, GAPd para estimativa do *gap* de otimalidade, t-MIP para tempo de CPU da etapa MILP, t-NLP para tempo de CPU da etapa NLP e t-TOTAL para tempo de CPU total.

Tabela 6.14: Resultados computacionais do cenário cinco com DS.

n	SM	NVC	NVB	NEL	NENL	FO-MIP	NN-MIP	FO-NLP	GAPd	t-MIP	t-NLP	t-TOTAL
i3	8	8.104	162	5.511	1.620	42.325,87	5.923	40.492,95	4,33	0,25	1,29	1,54
i4	8	10.805	216	7.312	2.160	42.327,64	6.205	40.514,40	4,28	0,63	3,11	3,73
i5	8	13.506	270	9.249	2.700	42.327,64	4.633	40.234,79	4,94	1,08	6,59	7,67
i6	8	16.207	324	11.322	3.240	42.327,64	11.747	40.254,88	4,90	3,55	11,44	14,99
i7	14	18.908	378	13.531			309.339			120,00		120,00

* SM para *status* do modelo, NVC para número de variáveis contínuas, NVB para número de variáveis binárias, NEL para número de equações lineares, NENL para número de equações não-lineares, FO-MIP para função objetivo do programa MILP, NN-MIP para número de nós do programa MILP, FO-NLP para função objetivo do programa NLP, GAPd para estimativa do *gap* de otimalidade, t-MIP para tempo de CPU da etapa MILP, t-NLP para tempo de CPU da etapa NLP e t-TOTAL para tempo de CPU total.

O modelo com DNS não foi resolvido até a otimalidade para o cenário cinco com $n = 8$ e $n = 9$. O MILP principal foi interrompido por falta de memória, e a árvore de busca foi truncada em 0,02% e 1,37% de GAP, respectivamente. Esse acontecimento não influenciou no tempo acumulado necessário para encontrar o melhor programa.

Na Tabela 6.11 pode-se visualizar o valor da função objetivo NLP encontrado pelos dois modelos quando solucionados com a mesma quantidade de *slots*. A Tabela 6.12 exibe o tempo computacional total t-TOTAL requerido pelos dois modelos quando solucionados com a mesma quantidade de *slots*.

Não há comparação entre as corridas de $n = 7$ porque o modelo com DS não encontrou solução viável na etapa MILP até atingir o limite de 2 h.

Observa-se na Tabela 6.15 que, em termos de função objetivo, o modelo com

Tabela 6.15: Comparação entre os modelos em termos de função objetivo para o cenário cinco.

n	DNS	DS	diferença
i3		40.492,95	
i4		40.514,40	
i5	39.942,48	40.234,79	-292,32
i6	40.207,97	40.254,88	-46,91
i7	40.053,95		
i8	39.974,59		
i9	39.823,70		

Tabela 6.16: Comparação entre os modelos em termos de tempo computacional para o cenário cinco.

n	DNS	DS	diferença
i3		1,54	
i4		3,73	
i5	5,99	7,67	-1,68
i6	10,55	14,99	-4,44
i7	13,29	120,00	
i8	64,77		
i9	60,63		

DS teve melhor desempenho que o modelo com DNS. Na corrida de $n = 5$ ele retornou uma função objetivo $292,32 \times 10^3$ maior que a do modelo com DNS. Em termos de tempo computacional, Tabela 6.16, o modelo com DNS obteve melhores desempenhos: ele foi capaz de solucionar o cenário cinco com $n = 6$ slots de prioridade em 4,44 min a menos que o modelo DS.

No cenário cinco é possível identificar a mesma tendência dos cenários três e quatro: conforme se aumenta o número de slots de prioridade n a diferença entre a função objetivo dos modelos com DNS e DS é reduzida. Esse resultado indica que o modelo com DNS precisa de mais slots de prioridade que o modelo com DS para encontrar o melhor programa.

O GAPd do modelo com DNS foi menor que o do modelo com DS. Nas corri-

das de $n = 5$, $n = 6$ e $n = 7$ o **GAPd** do modelo com **DNS** foi 4,09 %, 4,62 % e 4,72 %, respectivamente. Para o modelo com **DS**, o **GAPd** das corridas de $n = 5$ e $n = 6$ foi 4,94 % e 4,90 %, respectivamente. Uma explicação para esse resultado é que a função objetivo **MILP (FO-MIP)** – o limite superior para cálculo de **GAPd** – do modelo com **DS** foi maior que a do modelo com **DNS** em todas as corridas.

O melhor programa para o cenário cinco encontrado pelo modelo com **DNS** foi o de $n = 6$ e pelo modelo com **DS** foi o de $n = 4$. Entende-se por melhor programa aquele que retornou a melhor função objetivo de acordo com o Algoritmo 1 e o 1º critério de parada da Tabela 5.7. O tempo computacional da corrida correspondente ao melhor programa foi 10,55 min para o modelo com **DNS** e 3,73 min para o modelo com **DS**.

O tempo computacional acumulado necessário para encontrar o melhor programa **t-AC** do modelo com **DNS** foi 29,84 min, a soma dos tempos das corridas de $n = 5$, $n = 6$ e $n = 7$. O tempo acumulado do modelo com **DS** foi 12,95 min, que é a soma dos tempos das corridas de $n = 3$, $n = 4$ e $n = 5$.

Cenário seis

No cenário seis há no sistema $32.655,205 \text{ m}^3 \text{ d}^{-1}$ disponíveis para destilação. A quantidade relativa de petróleo destilado no cenário dois foi igual à capacidade máxima de destilação do sistema, $28.507,2 \text{ m}^3 \text{ d}^{-1}$. O volume total destilado em dez dias foi $285.072,0 \times 10^3 \text{ m}^3$. Esse resultado foi obtido pelas duas modelagens testadas.

As Tabelas 6.17 e 6.18 mostram os resultados computacionais do cenário seis modelado com DNS e DS, respectivamente.

Tabela 6.17: Resultados computacionais do cenário seis com DNS.

<i>n</i>	SM	NVC	NVB	NEL	NENL	FO-MIP	NN-MIP	FO-NLP	GAPd	t-MIP	t-NLP	t-TOTAL
i5	8	9.446	180	6.384	2.700	69.267,31	17.780	67.830,30	2,06	6,14	1,33	7,47
i6	8	11.335	216	7.559	3.240	69.400,94	13.666	67.827,06	2,27	5,55	2,61	8,16
i7	8	13.224	252	8.753	3.780	69.400,94	2.897	67.768,63	2,35	1,89	3,86	5,75
i8	8	15.113	288	9.966	4.320	69.400,94	5.948	67.727,13	2,41	5,46	5,49	10,96
i9	8	17.002	324	11.198	4.860	69.400,94	4.844	67.596,66	2,60	6,78	9,82	16,60

* SM para *status* do modelo, NVC para número de variáveis contínuas, NVB para número de variáveis binárias, NEL para número de equações lineares, NENL para número de equações não-lineares, FO-MIP para função objetivo do programa MILP, NN-MIP para número de nós do programa MILP, FO-NLP para função objetivo do programa NLP, GAPd para estimativa do *gap* de otimalidade, t-MIP para tempo de CPU da etapa MILP, t-NLP para tempo de CPU da etapa NLP e t-TOTAL para tempo de CPU total.

Tabela 6.18: Resultados computacionais do cenário seis com DS.

<i>n</i>	SM	NVC	NVB	NEL	NENL	FO-MIP	NN-MIP	FO-NLP	GAPd	t-MIP	t-NLP	t-TOTAL
i3	10	5.668	108	4.331			9.067			1,41		1,41
i4	8	7.557	144	5.658	2.160	69.400,94	82.219	67.753,87	2,37	10,37	0,82	11,19
i5	8	9.446	180	7.059	2.700	69.400,94	7.624	67.806,80	2,30	2,38	1,44	3,82
i6	8	11.335	216	8.534	3.240	69.400,94	15.452	67.803,43	2,30	5,97	2,85	8,82
i7	8	13.224	252	10.083	3.780	69.400,94	6.266	67.674,54	2,49	5,73	5,10	10,82

* SM para *status* do modelo, NVC para número de variáveis contínuas, NVB para número de variáveis binárias, NEL para número de equações lineares, NENL para número de equações não-lineares, FO-MIP para função objetivo do programa MILP, NN-MIP para número de nós do programa MILP, FO-NLP para função objetivo do programa NLP, GAPd para estimativa do *gap* de otimalidade, t-MIP para tempo de CPU da etapa MILP, t-NLP para tempo de CPU da etapa NLP e t-TOTAL para tempo de CPU total.

Na Tabela 6.19 pode-se visualizar o valor da função objetivo NLP encontrado pelos dois modelos quando solucionados com a mesma quantidade de *slots*. A Tabela 6.20 exibe o tempo computacional total t-TOTAL requerido pelos dois modelos quando solucionados com a mesma quantidade de *slots*.

Observa-se na Tabela 6.19 que, em termos de função objetivo, o modelo com DNS teve melhor desempenho que o modelo com DS. A função objetivo do modelo

Tabela 6.19: Comparação entre os modelos em termos de função objetivo para o cenário seis.

n	DNS	DS	diferença
i3			
i4		67.753,87	
i5	67.830,30	67.806,80	23,50
i6	67.827,06	67.803,43	23,63
i7	67.768,63	67.674,54	94,09
i8	67.727,13		
i9	67.596,66		

Tabela 6.20: Comparação entre os modelos em termos de tempo computacional para o cenário seis.

n	DNS	DS	diferença
i3		1,41	
i4		11,19	
i5	7,47	3,82	3,65
i6	8,16	8,82	-0,67
i7	5,75	10,82	-5,08
i8	10,96		
i9	16,60		

com DNS e $n = 7$ slots de prioridade foi $94,09 \times 10^3$ \$ maior que a do modelo com DS. Em termos de tempo computacional, Tabela 6.20, o modelo com DNS obteve melhores tempos: 5,08 min a menos que o modelo com DS em $n = 7$.

O GAPd do modelo com DNS foi menor que o do modelo com DS. Nas corridas de $n = 5$, $n = 6$ e $n = 7$ o GAPd do modelo com DNS foi 2,06 %, 2,27 % e 2,35 %, respectivamente. Para o modelo com DS, o GAPd das corridas de $n = 5$, $n = 6$ e $n = 7$ foi 2,30 %, 2,30 % e 2,49 %, respectivamente. Uma explicação para esse resultado é que o modelo com DNS encontrou melhores funções objetivos na etapa NLP (FO-NLP) que o modelo com DS.

O melhor programa para o cenário seis encontrado pelo modelo com DNS foi o de $n = 5$ e pelo modelo DS foi também o de $n = 5$. Entende-se por melhor programa

aquele que retornou a melhor função objetivo de acordo com o Algoritmo 1 e o 1º critério de parada da Tabela 5.7. O tempo computacional da corrida correspondente ao melhor programa foi 7,47 min para o modelo com DNS e 3,82 min para o modelo com DS.

O tempo computacional acumulado necessário para encontrar o melhor programa t-AC do modelo com DNS foi 15,63 min, a soma dos tempos das corridas de $n = 5$ e $n = 6$. O tempo acumulado do modelo com DS foi 25,24 min, que é a soma dos tempos das corridas de $n = 3$, $n = 4$, $n = 5$ e $n = 6$. O cenário seis foi o único entre os cenários otimizados em que o modelo com DNS encontrou o melhor programa em um tempo acumulado t-AC menor que o do modelo com DS.

Melhor programa

A Tabela 6.21 exibe um resumo do melhor programa em cada cenário encontrado pelos modelos com DNS e DS. Entende-se por melhor programa aquele que retornou a melhor função objetivo de acordo com o Algoritmo 1 e o 1º critério de parada da Tabela 5.7 (MOURET et al., 2011).

Tabela 6.21: Melhor programa para todos os cenários com os modelos com DNS e DS.

modelo	cenário	n	NV	NE	FO-MIP	NN-MIP	FO-NLP	GAPd	t-MIP	t-NLP	t-TOTAL	t-AC
DNS	dois	i6	9.493	8.583	46.662,99	61.156	45.680,23	2,11	26,75	1,56	28,32	59,07
	três	i8	25.361	17.553	66.921,98	4.270	65.899,90	1,53	3,49	24,36	27,85	142,83
	quatro	i7	16.381	13.368	48.569,13	9.583	47.031,54	3,17	6,86	8,47	15,33	158,54
	cinco	i6	16.531	12.465	42.156,75	7.665	40.207,97	4,62	2,43	8,12	10,55	29,84
	seis	i5	9.626	9.084	69.267,31	17.780	67.830,30	2,06	6,14	1,33	7,47	15,63
DS	dois	i3	4.747	4.833	46.488,32	24.928	45.610,22	1,89	1,49	0,20	1,69	4,79
	três	i6	19.021	15.956	66.921,98	15.807	65.876,19	1,56	3,74	10,20	13,94	67,30
	quatro	i6	14.041	13.060	48.569,13	2.942	46.945,64	3,34	1,02	7,15	8,17	45,43
	cinco	i4	11.021	9.472	42.327,64	6.205	40.514,40	4,28	0,63	3,11	3,73	12,95
	seis	i5	9.626	9.759	69.400,94	7.624	67.806,80	2,30	2,38	1,44	3,82	25,24

Observa-se na Tabela 6.21 que o modelo com DNS encontrou a melhor função objetivo para todos os cenários exceto o cinco. Para o cenário cinco o modelo com DS encontrou a melhor função objetivo. A diferença absoluta entre as funções objetivos FO-NLP encontradas pelos modelos com DNS e DS foi $70,017 \times 10^3$ \$, $23,710 \times 10^3$ \$,

$85,901 \times 10^3$ \$, $306,42 \times 10^3$ \$ e $23,499 \times 10^3$ \$ para os cenários dois, três, quatro, cinco e seis, respectivamente.

Pode acontecer de o resultado da programação encontrado pelo modelo não ser praticável, ou apresentar características que não são desejáveis no nível operacional. É desejável, por exemplo, evitar manobras que exijam a intervenção do operador.

Um aspecto muito discutido na literatura é o número de transições ou *changeovers* nas UDAs. É desejável que elas sejam apenas tantas quanto necessárias, pois há custos relacionados a essas transições. Na função objetivo formulada por Lee et al. (1996), por exemplo, os autores contabilizam o custo de transições nas UDAs. Portanto, quando um determinado tanque é alinhado à uma UDA, é vantajoso que todo o inventário no tanque seja consumido para evitar transições desnecessárias.

Os custos de transição estão relacionados ao tempo em que a coluna de destilação fica em regime transiente por causa da variação de composição na entrada. Variações bruscas ou saltos de composição ou de vazão prejudicam o desempenho da coluna, pois grandes perturbações na entrada produzem grandes perturbações na saída, e as propriedades dos produtos da destilação tendem a ficar fora das especificações. É desejável também que esses saltos de composição sejam o menor possível para uma operação mais estável.

Em ambos modelos investigados a restrição de cardinalidade, Equação 5.7, impõe que no máximo n operações de destilação W_D podem acontecer em uma UDA durante o horizonte de tempo. No modelo com DS menos operações de destilação W_D acontecem durante o horizonte de tempo, pois são necessários menos *slots* de prioridade para encontrar uma solução viável. Por exemplo, quando se resolve o modelo com DS com $n = 3$ *slots*, no máximo 3 operações de destilação são permitidas em cada UDA. Analogamente, quando se resolve o modelo com DNS com $n = 5$ *slots*, no máximo 5 operações de destilação são permitidas em cada UDA.

Essa é outra vantagem do modelo com DS: como são necessários menos *slots* de prioridade para resolver o modelo, a probabilidade é grande de que haverá menos transições nas UDAs que no modelo com DNS. Isso depende inclusive da estratégia empregada para encontrar a quantidade de *slots* que gera o melhor programa. Essa vantagem tem relação direta com a primeira vantagem: menos *slots* são requeridos para encontrar soluções viáveis. É possível implementar no modelo com DNS um

limite superior no número de operações de destilação W_D igual ao do modelo com DS, ou seja, um limite superior de $n - 2$ operações de destilação W_D . Analogamente, é possível implementar no modelo com DS um limite superior de $n + 2$ operações de destilação W_D .

Ao permitir que mais operações de destilação aconteçam ao longo do horizonte de tempo, o modelo com DNS permite mais transições nas UDAs. Isso implica que uma variedade maior de misturas é destilada, pois trocas entre tanques podem ser realizadas mais frequentemente caso necessário. Esse fato pode contribuir para que melhores funções objetivos sejam encontradas pelo modelo com DNS.

Na refinaria real modelada as restrições de qualidade são impostas apenas nas operações de destilação W_D , como uma consequência da possibilidade de alimentar uma UDA com duas correntes simultâneas procedentes de dois tanques de carga diferentes, contendo misturas diferentes. Essa característica do sistema e suas implicações no modelo e solução do problema são discutidas no Capítulo 4.

Se mais transições são permitidas nas UDAs, o modelo é mais flexível quanto ao gerenciamento das misturas destiladas. Pode acontecer de os mesmos tanques estarem alinhados com a mesma UDA em *slots* de prioridade adjacentes, mas em proporções diferentes. Nota-se que o modelo com DS também possibilita essa operação. A diferença é que menos transições são permitidas.

A função objetivo, Equação 5.2, é maximizar o valor marginal das misturas de petróleo destiladas. Portanto, os modelos tendem a manter os petróleos de baixo valor marginal em inventário e destilar apenas misturas contendo petróleos mais nobres.

No contexto do planejamento estratégico a função objetivo não representa os interesses do negócio. Os petróleos que chegam à refinaria e em inventário serão eventualmente processados. No chão de fábrica, o objetivo da programação é manter a operação nas UDAs o mais estável possível.

O modelo com DS cria mais possibilidades de mistura nos tanques de carga R_C , pois as parcelas podem ser fragmentadas em mais operações de descarregamento. Por outro lado, ao limitar o número de destilações W_D , não permite que essas misturas sejam enviadas às UDAs até o final do horizonte de programação. O modelo com DNS faz o contrário, pois permite mais transições nas UDAs e limita as possibilidades de mistura nos tanques de carga, já que as operações de descarregamento W_U

estão contidas em um único conjunto clique.

Deste modo, os dois modelos tendem a destilar apenas as misturas mais nobres de petróleo, mas o modelo com **DNS** cumpre essa tarefa mais eficientemente e retorna melhores funções objetivos.

É interessante notar que os dois modelos destilam exatamente o mesmo volume de misturas de petróleo em todos os cenários. As três **UDAs** operam na vazão máxima durante todo o horizonte de programação nos cenários dois, quatro, cinco e seis. No cenário três uma **UDA** não opera em capacidade máxima porque o volume relativo disponível para destilação – inventário útil inicial nos tanques de carga R_C mais volume que chega nas parcelas – é menor que a capacidade máxima de destilação da refinaria.

A função objetivo, Equação 5.2, não considera custos relacionados ao processo como, por exemplo, custos de inventário, de movimentações, custos de transições nas **UDAs** e custos ocasionados por operação instável das **UDAs** decorrentes dos saltos de composição. Sugere-se para investigações futuras que se compare o desempenho dos modelos com **DNS** e **DS** condicionados a uma função objetivo que represente mais satisfatoriamente a realidade da refinaria.

Os resultados da Tabela 6.21 são fortemente dependente da estratégia que define a quantidade de *slots* de prioridade que gera o melhor programa. Para determinar essa quantidade usou-se a abordagem aditiva considerando a função objetivo da etapa **NLP**. Se a função objetivo da etapa **MILP** fosse considerada, os resultados da Tabela 6.21 não seriam os mesmos.

Cenário dois

O melhor programa para o cenário dois na modelagem com **DNS** foi o de $n = 6$, e na modelagem com **DS** foi o de $n = 3$.

A Figura 6.2 mostra a programação das operações de descarregamento W_U obtida pelos dois modelos no cenário dois. O modelo com **DNS** realizou em seu melhor programa ($n = 6$) cinco operações de descarregamento de parcelas. Ele dividiu a parcela *prec1* nos tanques II e FF, dividiu a parcela *prec2* nos tanques AA e BB e descarregou toda a parcela *prec3* para o tanque LL. O modelo com **DS** realizou em seu

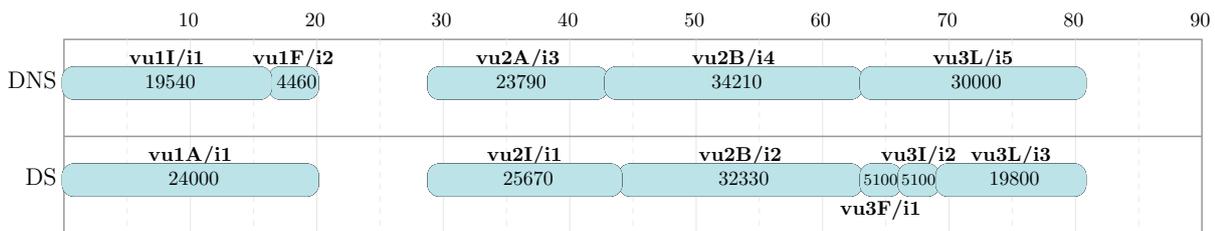


Figura 6.2: Programação do descarregamento de parcelas do melhor programa para o cenário dois.

melhor programa ($n = 3$) seis operações de descarregamento de parcelas. A parcela $prc1$ foi totalmente descarregada para o tanque AA, a parcela $prc2$ foi dividida entre os tanques II e BB e a parcela $prc3$ foi dividida entre os tanques FF, II e LL.

Uma das consequências do uso de **DS** na modelagem é que a cardinalidade de operações ativas de descarregamento não é limitada pelo número de *slots*. Com **DNS** apenas n descarregamentos W_U podem acontecer ao longo do horizonte de tempo. Com **DS** no máximo $|R_P| \times n$ operações de descarregamento podem acontecer⁶.

As Figuras 6.3 e 6.4 exibem o resultado do melhor programa nos misturadores e **UDAs**. Acima da barra no gráfico de Gantt é possível visualizar a qual operação ela se refere e em que *slot* de prioridade a operação acontece. Dentro da barra, pode-se visualizar da esquerda para a direita o valor da propriedade **TAN** da operação em $\text{mg}_{\text{KOH}}/\text{g}$, o valor marginal unitário relativa da mistura \dot{G}_{iv} em $\$/\text{h}$ e a duração da operação D_{iv} em h.

No cenário dois todas as três **UDAs** operam em vazão máxima em todo o horizonte de tempo nas duas modelagens. Sempre que um tanque foi alinhado à uma **UDA** o modelo optou por mantê-lo na carga até consumir todo seu inventário. Houve casos em que esse arranjo não foi possível porque o inventário no tanque era insuficiente para suprir a **UDA** até o final do horizonte de tempo. Esse resultado é desejável no nível operacional, pois evita transições desnecessárias nas **UDAs**.

Observa-se nas Figuras 6.3 e 6.4 que o modelo com **DS** retornou funções objetivos menores que as do modelo com **DNS** porque destila misturas de menor valor marginal relativo \dot{G}_{iv} . O valor marginal relativo da mistura sendo transferida na

⁶Na modelagem com **DS** o limite superior para cardinalidade de operações de descarregamento é a priori $|R_P| \times n$. No entanto, esse limite é menor quando se considera a restrição de duração mínima.

operação $v \in W$ no *slot* de prioridade $i \in T$ é calculada pela Equação 6.1.

$$\dot{G}_{iv} = \sum_{c \in C} \frac{G_c V_{ivc}}{D_{iv}}, v \in W, i \in T \quad (6.1)$$

O valor marginal relativo das misturas destiladas pelo modelo com DS é menor porque a duração da operação, termo no denominador, é maior.

O valor marginal das operações de alimentação de uma UDA pode ser calculada pela Equação 6.2.

$$\sum_{v \in I_r} \sum_{i \in T} \dot{G}_{iv} D_{iv}, r \in \{UC, UN, UV\} \quad (6.2)$$

As operações de destilação W_D podem ser observadas nas Figuras 6.3 e 6.4. A operação de saída do misturador *mxr1* alimenta a UDA UC, a do misturador *mxr2* alimenta a UDA UN e a do misturador *mxr3* alimenta a UDA UV.

No modelo com DNS o valor marginal das operações de destilação em UC, UN e UV é $21.691,75 \times 10^3$ \$, $15.706,28 \times 10^3$ \$ e $8.282,20 \times 10^3$ \$, respectivamente. Somando essas três valores marginais tem-se a função objetivo $45.680,23 \times 10^3$ \$. No modelo com DS o valor marginal das operações de destilação em UC, UN e UV é $21.590,09 \times 10^3$ \$, $15.695,18 \times 10^3$ \$ e $8.324,94 \times 10^3$ \$, respectivamente. Somando esses três valores marginais tem-se a função objetivo $45.610,22 \times 10^3$ \$. Apenas o valor marginal das misturas destiladas na UDA UV no modelo com DS supera a do modelo com DNS. Nas UDAs UC e UN o valor marginal das misturas destiladas é maior no modelo com DNS.

As Tabelas 6.22 e 6.23 exibem os alinhamentos entre tanques de carga e UDAs ativos ao longo do horizonte de tempo para os modelos com DNS e DS, respectivamente. Esses alinhamentos também podem ser observados nas Figuras 6.3 e 6.4. Pelas Tabelas 6.22 e 6.23 é possível constatar que o modelo com DNS consegue enviar para a destilação uma variedade maior de misturas de petróleo.

É interessante notar que no modelo com DNS, quando os mesmos alinhamentos são mantidos por vários *slots* de prioridade nem sempre as operações têm a mesma vazão. Por exemplo, na Tabela 6.22, a UDA UN mantém os mesmos alinhamentos em $i1$ e $i2$ (tanques CC e DD). No entanto a proporção entre os tanques é diferente em cada *slot*. Em $i1$ a UDA UN recebe $123 \text{ m}^3 \text{ h}^{-1}$ do tanque CC e $277 \text{ m}^3 \text{ h}^{-1}$ do tanque DD. Em $i2$ ela recebe $300 \text{ m}^3 \text{ h}^{-1}$ do tanque CC e $100 \text{ m}^3 \text{ h}^{-1}$ do tanque DD.

Tabela 6.22: Alinhamentos ativos na destilação do modelo com DNS ao longo do horizonte de tempo para o cenário dois.

	i1	i2	i3	i4	i5	i6
UC	BB	II	-	-	BB	BB
	HH	HH	HH	AA	AA	AA
	i1	i2	i3	i4		
UN	CC	CC	FF	FF		
	DD	DD	DD	DD		
	i1	i2	i3	i4	i5	i6
UV	HH	HH	HH	AA	-	LL
	LL	II	II	II	BB	BB

A vazão máxima da UDA UN é $400,0 \text{ m}^3 \text{ h}^{-1}$. A vazão das operações de mistura W_M podem ser encontradas no Apêndice C. A proporção de cada tanque nas correntes de alimentação das UDAs também pode ser observada nas Figuras 6.3 e 6.4, a partir da proporção entre o parâmetro 6.1 na entrada e saída dos misturadores.

Observa-se nas Figuras 6.3 e 6.4 que o modelo com DS resulta em saltos maiores de propriedade TAN na alimentação das UDAs. Essa tendência foi identificada em todos os cenários, exceto o três. A Tabela 6.24 exhibe o maior salto da propriedade TAN na alimentação das UDAs.

No modelo com DNS, Figura 6.3, o maior salto da propriedade TAN foi $0,73 \text{ mg}_{\text{KOH}}/\text{g}$ ($i1$ para $i2$), $0,51 \text{ mg}_{\text{KOH}}/\text{g}$ ($i2$ para $i3$) e $0,46 \text{ mg}_{\text{KOH}}/\text{g}$ ($i4$ para $i5$) nas UDAs UC, UN e UV, respectivamente. No modelo com DS, Figura 6.4, o maior salto da propriedade TAN foi $0,79 \text{ mg}_{\text{KOH}}/\text{g}$ ($i1$ para $i2$), $0,33 \text{ mg}_{\text{KOH}}/\text{g}$ ($i1$ para $i2$) e $0,47 \text{ mg}_{\text{KOH}}/\text{g}$ ($i2$ para $i3$) nas UDAs UC, UN e UV, respectivamente.

Uma possível explicação para os menores saltos na propriedade TAN obtidos pelo modelo com DNS é a maior diversidade de misturas de petróleo que são enviadas para destilação.

Tabela 6.23: Alinhamentos ativos na destilação do modelo com DS ao longo do horizonte de tempo para o cenário dois.

	i1	i2	i3
UC	BB	AA	BB
	HH	HH	II
	i1	i2	i3
UN	CC	DD	-
	DD	FF	FF
	i1	i2	i3
UV	HH	HH	-
	LL	-	BB

Tabela 6.24: Maior salto de TAN na alimentação das UDAs para o cenário dois.

UDA	DNS	DS
UC	0,73	0,79
UN	0,51	0,33
UV	0,46	0,48

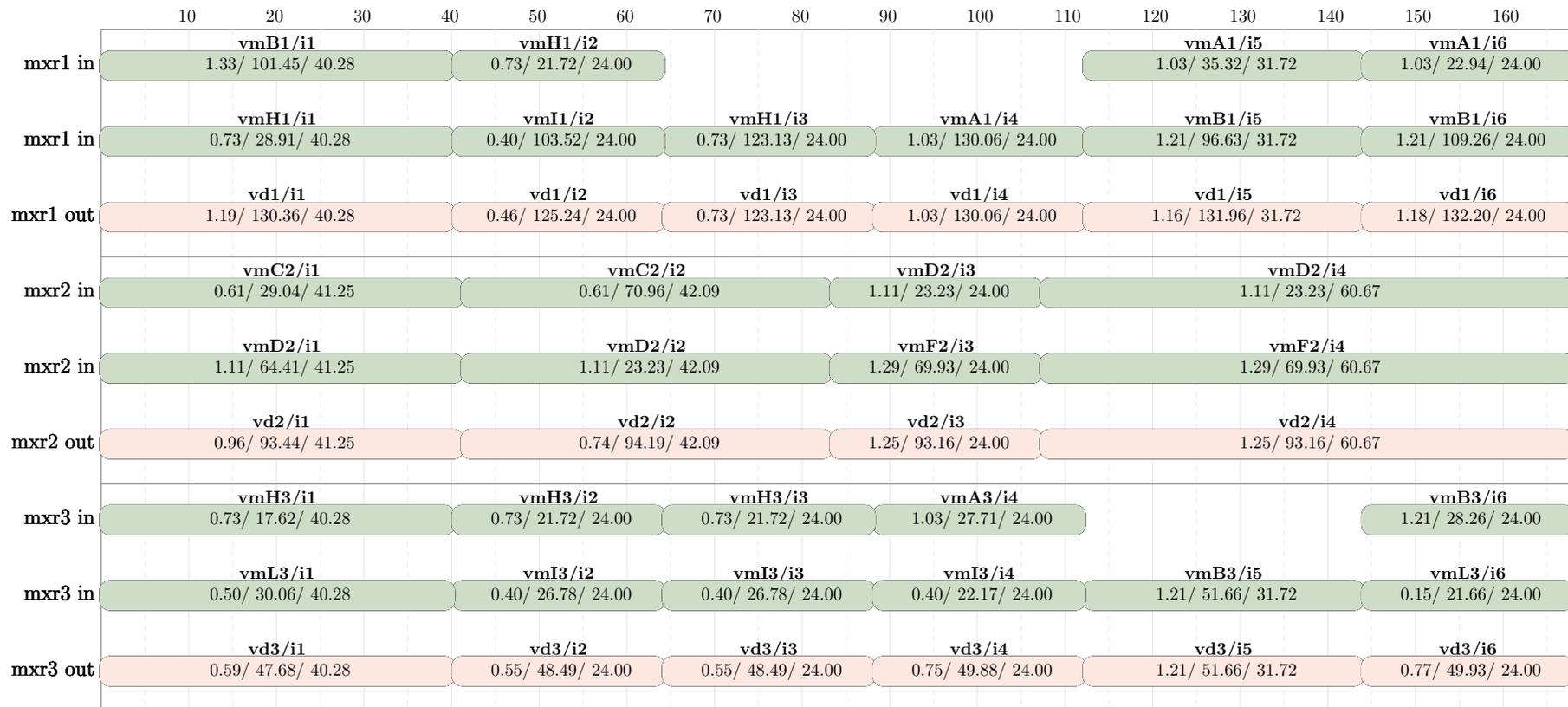


Figura 6.3: Programação nos misturadores e UDAs para o cenário dois com $n = 6$ e DNS.

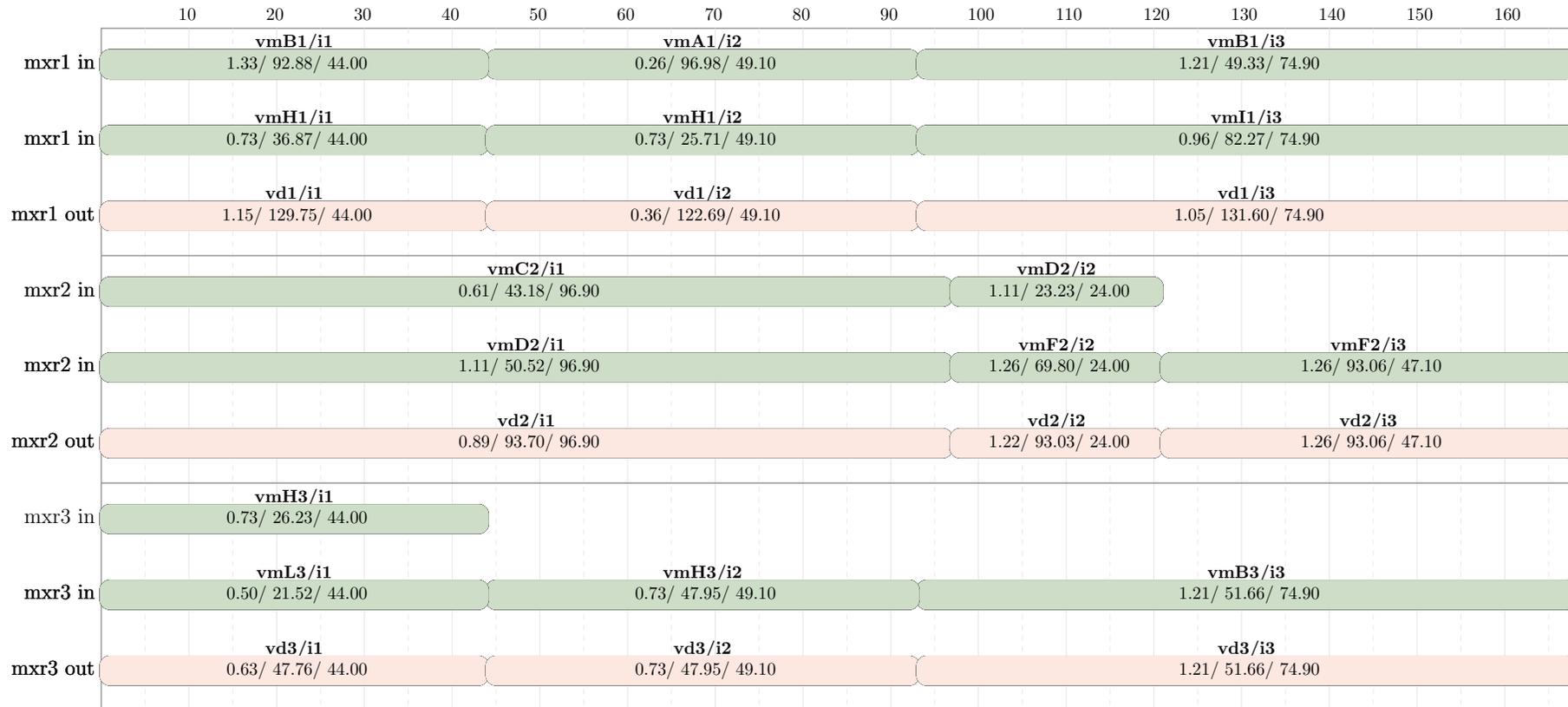


Figura 6.4: Programação nos misturadores e UDAs para o cenário dois com $n = 3$ e DS.

Cenário três

O melhor programa para o cenário três na modelagem com **DNS** foi o de $n = 8$, e na modelagem com **DS** foi o de $n = 6$.

A Figura 6.5 mostra a programação das operações de descarregamento W_U obtida pelos dois modelos no cenário três. O modelo com **DNS** realizou em seu me-

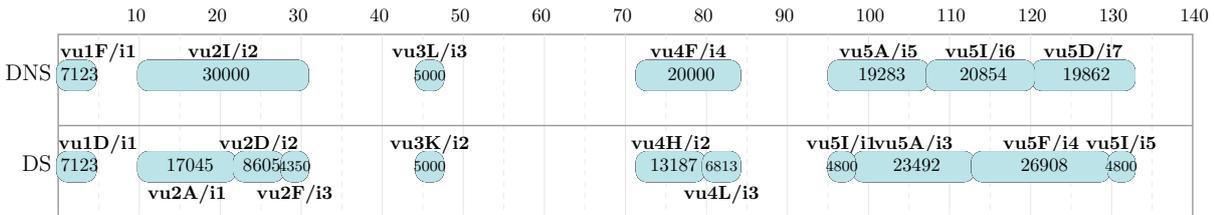


Figura 6.5: Programação do descarregamento de parcelas do melhor programa para o cenário três.

lhor programa ($n = 8$) sete operações de descarregamento de parcelas. As parcelas $prc1$, $prc2$, $prc3$ e $prc4$ foram completamente descarregadas para os tanques FF, II, LL e FF, respectivamente. A parcela $prc5$ foi dividida entre os tanques AA, II e DD. O modelo com **DS** realizou em seu melhor programa ($n = 6$) onze operações de descarregamento de parcelas. A parcela $prc1$ foi totalmente descarregada para o tanque DD. A parcela $prc2$ foi dividida entre os tanques AA, DD e FF. A parcela $prc3$ foi totalmente descarregada para o tanque KK. A parcela $prc4$ foi descarregada para os tanques HH e LL, e a parcela $prc5$ foi descarregada para os tanques II, AA, FF e II.

As Figuras 6.7 e 6.8 exibem o resultado do melhor programa nos misturadores e **UDAs**. Acima da barra no gráfico de Gantt é possível visualizar a qual operação ela se refere e em que *slot* de prioridade a operação acontece. Dentro da barra, pode-se visualizar da esquerda para a direita o valor da propriedade **TAN** da operação em $\text{mg}_{\text{KOH}}/\text{g}$, o valor marginal unitário relativo da mistura \dot{G}_{iv} em $\$/\text{h}$ e a duração da operação D_{iv} em h.

A Figura 6.6 exhibe a vazão de carga das **UDAs** ao longo do horizonte de programação. O cenário três foi o único no qual as **UDAs** não operam em vazão máxima durante todo o horizonte de tempo, porque o volume de petróleo disponível para destilação é menor que a capacidade máxima da refinaria. Na modelagem com **DS** as operações de destilação v_2^d em $i2$ e v_2^d em $i4$ não operam na vazão máxima, mas com

$360 \text{ m}^3 \text{ h}^{-1}$ e $396,2 \text{ m}^3 \text{ h}^{-1}$, respectivamente. Na modelagem com DNS a única operação de destilação que não opera em vazão máxima é v_2^d em $i8$ com $377,9 \text{ m}^3 \text{ h}^{-1}$.

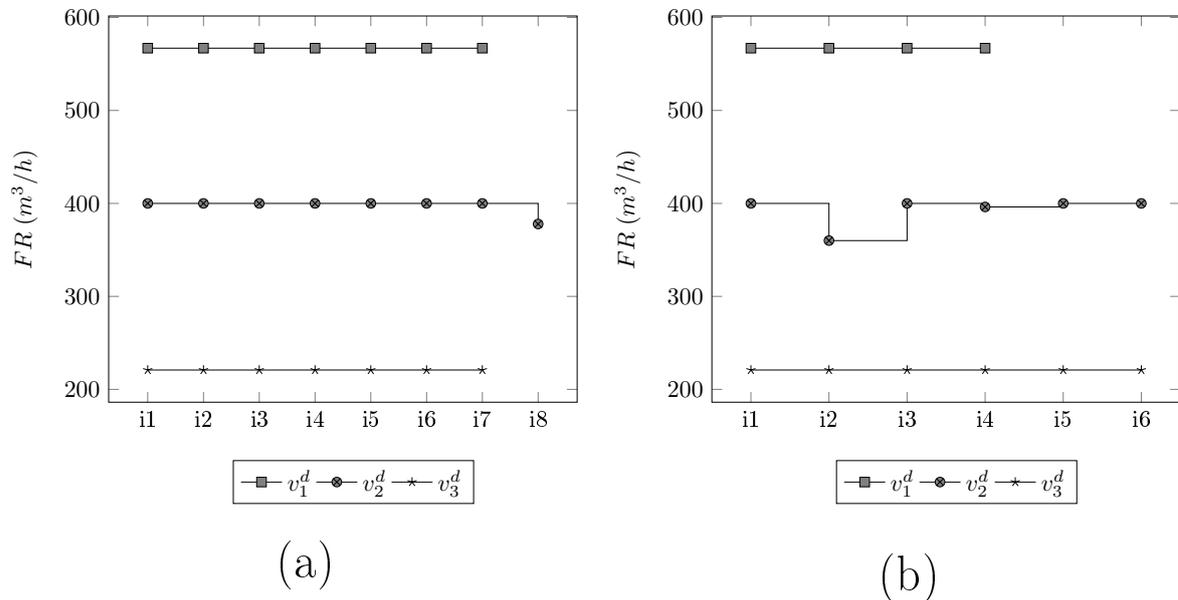


Figura 6.6: Vazão de carga nas UDAs no melhor programa encontrado pelo modelo com (a) DNS e (b) DS para o cenário três.

Observa-se nas Figuras 6.3 e 6.4 que, assim como no cenário dois, o modelo com DS retornou funções objetivos menores que as do modelo com DNS porque destila misturas de menor valor marginal relativo \hat{G}_{iv} . Esse resultado é consequência da maior duração de operações de destilação no modelo com DS.

No modelo com DNS o valor marginal das operações de destilação em UC, UN e UV é $31.477,39 \times 10^3 \$$, $22.268,36 \times 10^3 \$$ e $12.154,14 \times 10^3 \$$, respectivamente. Somando essas três valores marginais tem-se a função objetivo $65.899,90 \times 10^3 \$$. No modelo com DS o valor marginal das operações de destilação em UC, UN e UV é $31.416,15 \times 10^3 \$$, $22.065,15 \times 10^3 \$$ e $12.394,89 \times 10^3 \$$, respectivamente. Somando essas três valores marginais tem-se a função objetivo $65.876,19 \times 10^3 \$$. Apenas o valor marginal das misturas destiladas na UDA UV no modelo com DS supera a do modelo com DNS. Nas UDAs UC e UN o valor marginal das misturas destiladas é maior no modelo com DNS.

As Tabelas 6.25 e 6.26 exibem os alinhamentos entre tanques de carga e UDAs ativos ao longo do horizonte de tempo para os modelos com DNS e DS, respectivamente. Esses alinhamentos também podem ser observados nas Figuras 6.7 e

6.8. Pelas Tabelas 6.25 e 6.26 é possível constatar que o modelo com DNS consegue enviar para a destilação uma variedade maior de misturas de petróleo.

Tabela 6.25: Alinhamentos ativos na destilação do modelo com DNS ao longo do horizonte de tempo para o cenário três.

	i1	i2	i3	i4	i5	i6	i7	
UC	HH	HH	II	LL	LL	LL	LL	
	KK	KK	KK	KK	KK	AA	II	
	i1	i2	i3	i4	i5	i6	i7	i8
UN	DD	DD	DD	DD	FF	FF	FF	DD
	EE	EE	EE	EE	EE	-	-	-
	i1	i2	i3	i4	i5	i6	i7	
UV	GG	GG	GG	GG	GG	AA	II	
	KK	KK	KK	LL	-	-	-	

Também no cenário três a diversidade das misturas destiladas é maior no modelo com DNS, pois ele pode realizar mais transições nas UDAs. No entanto, o cenário três foi o único entre os cenários otimizados em que o modelo com DS resultou em menores saltos de TAN nas operações de destilação. A Tabela 6.27 exhibe o maior salto da propriedade TAN na alimentação das UDAs.

No modelo com DNS, Figura 6.7, o maior salto da propriedade TAN foi 0,56 mg_{KOH}/g (*i5* para *i6*), 0,71 mg_{KOH}/g (*i7* para *i8*) e 0,47 mg_{KOH}/g (*i4* para *i5*) nas UDAs UC, UN e UV, respectivamente. No modelo com DS, Figura 6.8, o maior salto da propriedade TAN foi 0,46 mg_{KOH}/g (*i2* para *i3*), 0,15 mg_{KOH}/g (*i4* para *i5*) e 0,52 mg_{KOH}/g (*i2* para *i3*) nas UDAs UC, UN e UV, respectivamente.

É pertinente lembrar que o cenário três é o único entre os cenários otimizados em que o volume relativo disponível para destilação é menor que a capacidade máxima de destilação da refinaria. Ele é também o cenário com a menor acidez média inicial do sistema \overline{TAN}_0 , 0,48 mg_{KOH}/g, considerando todos os recursos, inclusive parcelas.

Tabela 6.26: Alinhamentos ativos na destilação do modelo com DS ao longo do horizonte de tempo para o cenário três.

	i1	i2	i3	i4
UC	HH	AA	KK	AA
	-	-	-	LL
UN	EE	EE	EE	-
	-	-	DD	DD
UV	GG	GG	-	-
	KK	-	HH	HH

Tabela 6.27: Maior salto de TAN na alimentação das UDAs para o cenário três.

UDA	DNS	DS
UC	0,57	0,47
UN	0,72	0,15
UV	0,47	0,52



Figura 6.7: Programação nos misturadores e UDAs para o cenário três com $n = 7$ e DNS.

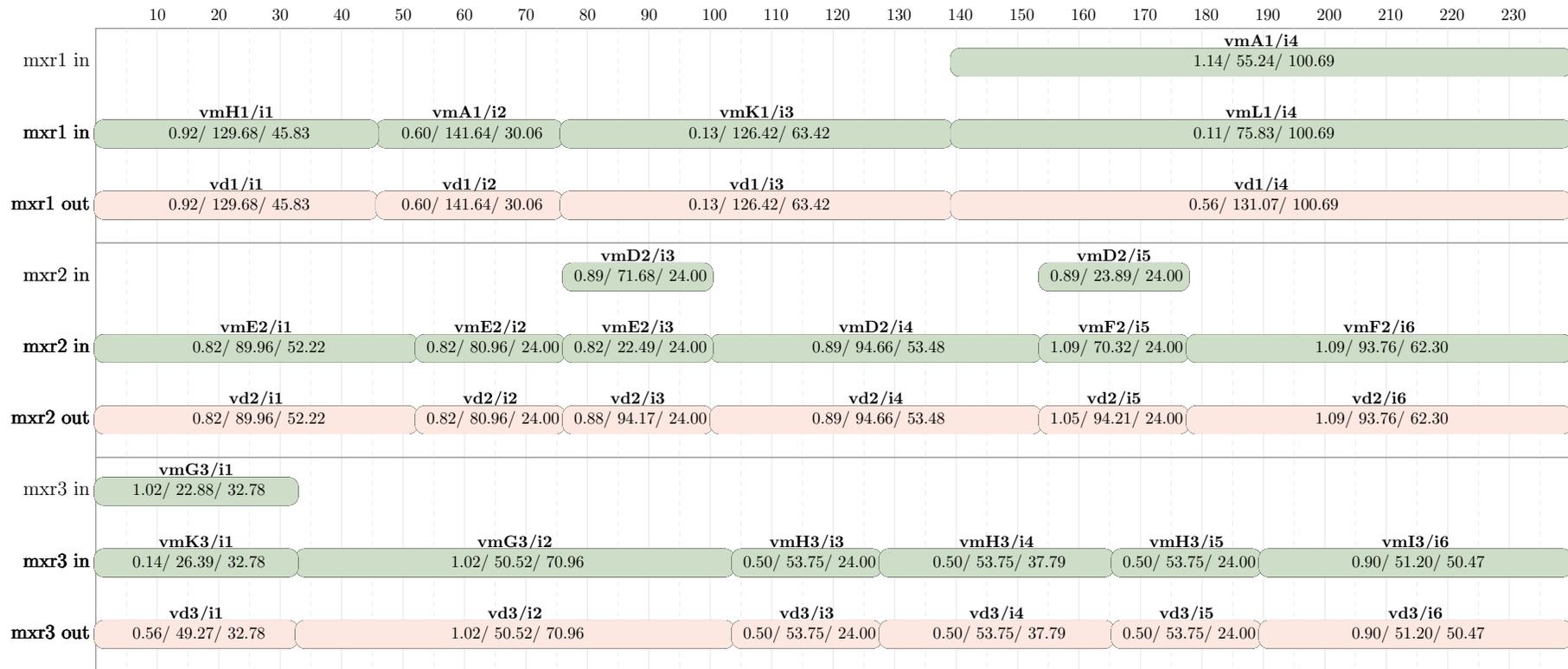


Figura 6.8: Programação nos misturadores e UDAs para o cenário três com $n = 6$ e DS.

Cenário quatro

O melhor programa para o cenário quatro na modelagem com **DNS** foi o de $n = 7$, e na modelagem com **DS** foi o de $n = 6$.

A Figura 6.9 mostra a programação das operações de descarregamento W_U obtida pelos dois modelos no cenário quatro. O modelo com **DNS** realizou em seu

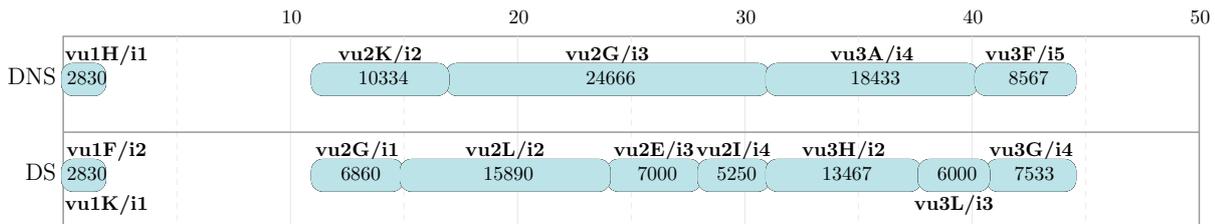


Figura 6.9: Programação do descarregamento de parcelas do melhor programa para o cenário quatro.

melhor programa ($n = 7$) cinco operações de descarregamento de parcelas. A parcela $prc1$ foi completamente descarregada para o tanque HH. A parcela $prc2$ foi dividida entre os tanques KK e GG. A parcela $prc3$ foi dividida entre os tanques AA e FF. O modelo com **DS** realizou em seu melhor programa ($n = 6$) nove operações de descarregamento de parcelas. A parcela $prc1$ foi totalmente descarregada para o tanque FF. A parcela $prc2$ foi dividida entre os tanques GG, LL, EE e II. A parcela $prc3$ foi dividida entre os tanques HH, LL e GG.

É interessante notar que o modelo com **DS** alocou uma operação de descarregamento da parcela $prc1$ para o tanque KK em $i1$, mas com duração D_{iv} nula. Esse resultado pode ser explicado pelo efeito da restrição de quebra de simetria. O modelo optou por alocar a operação v_{1K}^u em $i1$ para ativar o tanque KK.

A restrição de quebra de simetria, Equação 5.35, gera uma sequência de operações para cada conjunto clique. Segundo a restrição de quebra de simetria uma operação só pode acontecer no $slot$ de prioridade i se uma operação não-sobreposta a ela tenha acontecido no $slot$ de prioridade anterior $i - 1$. Essa condição vale para todos os $slots$ do programa, exceto para o primeiro. O primeiro $slot$ de prioridade tem autonomia para iniciar uma sequência a qualquer instante.

Nos tanques de carga R_C uma operação de saída só pode acontecer no $slot$ i se alguma outra operação não-sobreposta a ela estiver ativa em $i - 1$. As operações

não-sobrepostas às saídas dos tanques de carga R_C são as operações de entrada nesse mesmo tanque ou a própria operação de saída, uma vez que toda operação é não-sobreposta consigo mesma.

Nas duas modelagens testadas um tanque que não começa na carga em $t = 0$ e não recebe descarregamento de parcela W_U em $t = 0$ precisa receber pelo menos uma operação de descarregamento W_U para ser ativado mais adiante no horizonte de programação.

O modelo com **DS** optou por ativar o tanque **KK** com o descarregamento v_{1k}^u em $i1$ porque alinha este tanque em $i2$ com a **UDA UC**. O descarregamento v_{1k}^u em $i1$ tem duração nula, isto é $D_{iv} = 0$. Nota-se que o tempo de preparação foi respeitado entre as operações v_{1K}^u em $i1$ e v_{K1}^m em $i2$.

Ao analisar o melhor programa encontrado pelas duas modelagens para o cenário quatro conclui-se que o desempenho do modelo está vinculado a outros fatores além da segmentação das parcelas em várias operações de descarregamento W_U . O modelo com **DS** realizou nove operações de descarregamento em seu melhor programa ($n = 6$), mas mesmo assim sua função objetivo não foi tão boa quanto a do modelo **DNS**, que fez apenas cinco operações de descarregamento em seu melhor programa ($n = 7$).

As Figuras 6.10 e 6.11 exibem o resultado do melhor programa nos misturadores e **UDAs**. Acima da barra no gráfico de Gantt é possível visualizar a qual operação ela se refere e em que *slot* de prioridade a operação acontece. Dentro da barra, pode-se visualizar da esquerda para a direita o valor da propriedade **TAN** da operação em $\text{mg}_{\text{KOH}}/\text{g}$, o valor marginal unitário relativo da mistura \dot{G}_{iv} em $\$/\text{h}$ e a duração da operação D_{iv} em h.

No cenário quatro todas as três **UDAs** operam em vazão máxima em todo o horizonte de tempo nas duas modelagens.

No modelo com **DNS** o valor marginal das operações de destilação em UC, UN e UV é $22.616,33 \times 10^3 \$$, $15.495,49 \times 10^3 \$$ e $8.919,72 \times 10^3 \$$, respectivamente. Somando esses três valores marginais tem-se a função objetivo $47.031,54 \times 10^3 \$$. No modelo com **DS** o valor marginal das operações de destilação em UC, UN e UV é $22.491,54 \times 10^3 \$$, $15.685,00 \times 10^3 \$$ e $8.769,10 \times 10^3 \$$, respectivamente. Somando esses três valores marginais tem-se a função objetivo $65.876,19 \times 10^3 \$$. Apenas o valor

marginal das misturas destiladas na UDA UN no modelo com DS supera a do modelo com DNS. Nas UDAs UC e UV o valor marginal das misturas destiladas é maior no modelo com DNS.

As Tabelas 6.28 e 6.29 exibem os alinhamentos entre tanques de carga e UDAs ativos ao longo do horizonte de tempo para os modelos com DNS e DS, respectivamente. Esses alinhamentos também podem ser observados nas Figuras 6.10 e 6.11. Pelas Tabelas 6.28 e 6.29 é possível constatar que o modelo com DNS consegue enviar para a destilação uma variedade maior de misturas de petróleo.

Tabela 6.28: Alinhamentos ativos na destilação do modelo com DNS ao longo do horizonte de tempo para o cenário quatro.

	i1	i2	i3	i4	i5		
UC	AA	HH	HH	GG	AA		
	-	-	-	-	-		
	i1	i2	i3	i4	i5	i6	i7
UN	DD	DD	DD	DD	-	FF	FF
	EE	EE	EE	EE	EE	EE	-
	i1	i2	i3	i4	i5	i6	
UV	ll	ll	ll	ll	ll	ll	
	-	-	KK	-	-	-	

Observa-se nas Figuras 6.10 e 6.11 que o modelo com DS resulta em saltos maiores de propriedade TAN na alimentação das UDAs. A Tabela 6.30 exhibe o maior salto da propriedade TAN na alimentação das UDAs.

No modelo com DNS, Figura 6.10, o maior salto da propriedade TAN foi 0,09 mg_{KOH}/g (*i4* para *i5*), 0,05 mg_{KOH}/g (*i5* para *i6*) e 0,33 mg_{KOH}/g (*i3* para *i4*) nas UDAs UC, UN e UV, respectivamente. No modelo com DS, Figura 6.11, o maior salto da propriedade TAN foi 0,27 mg_{KOH}/g (*i1* para *i2*), 0,20 mg_{KOH}/g (*i3* para *i4*) e 0,11 mg_{KOH}/g (*i2* para *i3*) nas UDAs UC, UN e UV, respectivamente.

Tabela 6.29: Alinhamentos ativos na destilação do modelo com DS ao longo do horizonte de tempo para o cenário quatro.

	i1	i2	i3	i4	i5	i6
UC	AA	AA	AA	HH	-	-
	-	-	-	-	GG	GG
	i1	i2	i3	i4	i5	i6
UN	EE	-	FF	EE	EE	EE
	DD	DD	DD	-	-	-
	i1	i2	i3	i4	i5	i6
UV	AA	AA	-	-	II	II
	HH	-	HH	HH	HH	HH

Tabela 6.30: Maior salto de TAN na alimentação das UDAs para o cenário quatro.

UDA	DNS	DS
UC	0,09	0,27
UN	0,05	0,20
UV	0,33	0,11

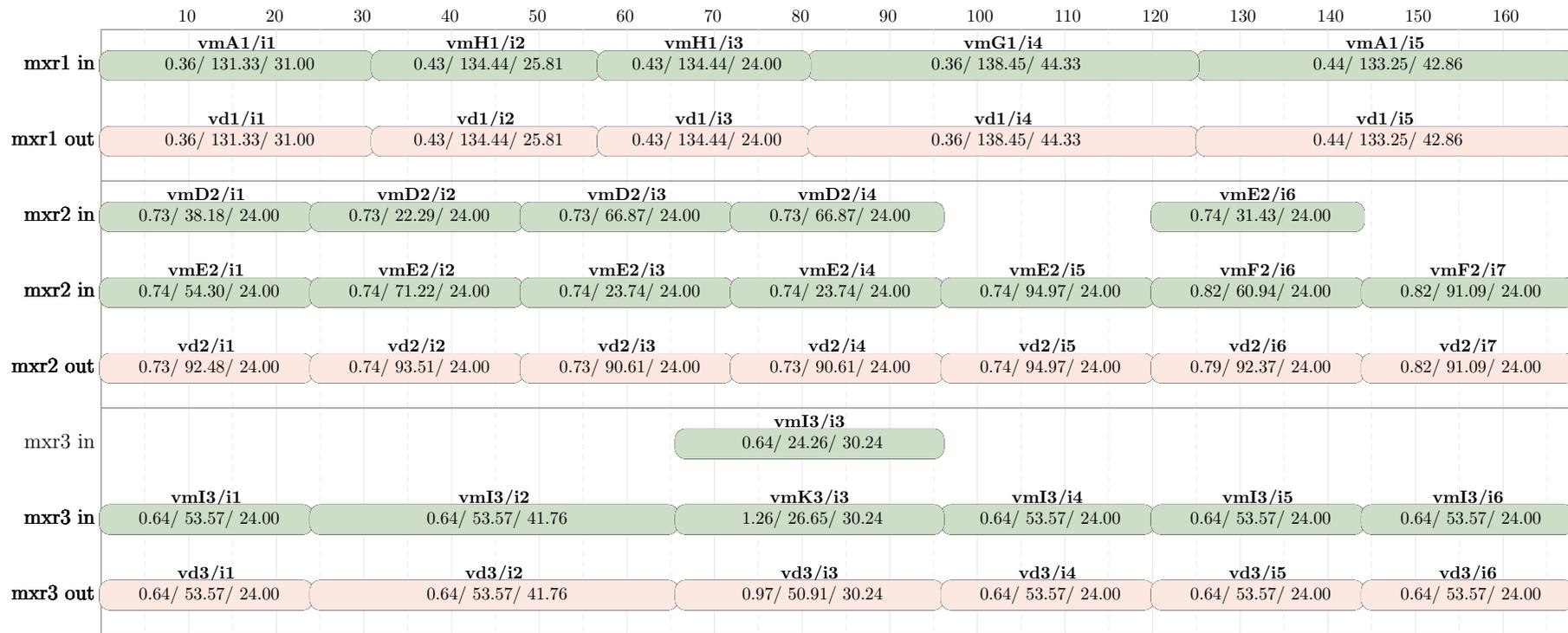


Figura 6.10: Programação nos misturadores e UDAs para o cenário quatro com $n = 7$ e DNS.



Figura 6.11: Programação nos misturadores e UDAs para o cenário quatro com $n = 6$ e DS.

Cenário cinco

O melhor programa para o cenário cinco na modelagem com **DNS** foi o de $n = 6$, e na modelagem com **DS** foi o de $n = 4$.

A Figura 6.12 mostra a programação das operações de descarregamento W_U obtida pelos dois modelos no cenário cinco. O modelo com **DNS** realizou em seu

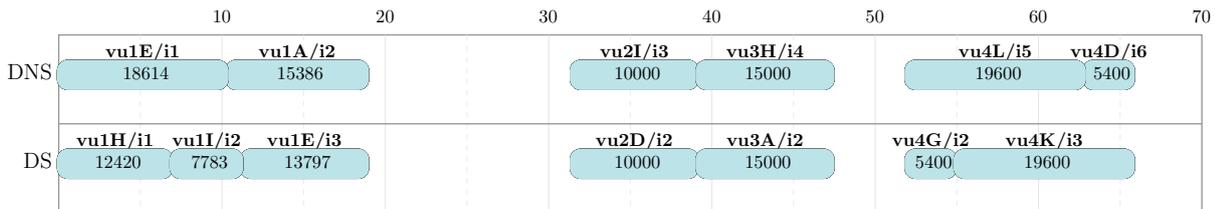


Figura 6.12: Programação do descarregamento de parcelas do melhor programa para o cenário cinco.

melhor programa ($n = 6$) seis operações de descarregamento de parcelas. A parcela *prc1* foi dividida entre os tanques EE e AA, a parcela *prc2* entre os tanques II e HH e a parcela *prc3* entre os tanques LL e DD. O modelo com **DS** realizou em seu melhor programa ($n = 4$) sete operações de descarregamento de parcelas. A parcela *prc1* foi dividida entre os tanques HH, II e EE, a parcela *prc2* entre os tanques DD e AA e a parcela *prc3* entre os tanques GG e KK.

As Figuras 6.13 e 6.14 exibem o resultado do melhor programa nos misturadores e **UDAs**. Acima da barra no gráfico de Gantt é possível visualizar a qual operação ela se refere e em que *slot* de prioridade a operação acontece. Dentro da barra, pode-se visualizar da esquerda para a direita o valor da propriedade **TAN** da operação em $\text{mg}_{\text{KOH}}/\text{g}$, o valor marginal unitário relativo da mistura \dot{G}_{iv} em $\$/\text{h}$ e a duração da operação D_{iv} em h.

No cenário cinco todas as três **UDAs** operam em vazão máxima em todo o horizonte de tempo nas duas modelagens.

O cenário cinco foi o único para o qual o modelo com **DS** encontrou o melhor programa. Sua função objetivo foi $306,42 \times 10^3$ \$ maior do que aquela encontrada pelo modelo com **DNS**. Entre os cenários otimizados, o cenário cinco possui o maior volume relativo disponível para destilação.

No modelo com **DNS** o valor marginal das operações de destilação em UC,

UN e UV é $19.410,27 \times 10^3$ \$, $13.361,71 \times 10^3$ \$ e $7.435,99 \times 10^3$ \$, respectivamente. Somando essas três valores marginais tem-se a função objetivo $40.207,97 \times 10^3$ \$. No modelo com DS o valor marginal das operações de destilação em UC, UN e UV é $19.710,35 \times 10^3$ \$, $13.358,74 \times 10^3$ \$ e $7.445,31 \times 10^3$ \$, respectivamente. Somando essas três valores marginais tem-se a função objetivo $40.514,40 \times 10^3$ \$. Apenas o valor marginal das misturas destiladas na UDA UN no modelo com DNS supera a do modelo com DS. Nas UDAs UC e UV o valor marginal das misturas destiladas é maior no modelo com DS.

As Tabelas 6.31 e 6.32 exibem os alinhamentos entre tanques de carga e UDAs ativos ao longo do horizonte de tempo para os modelos com DNS e DS, respectivamente. Esses alinhamentos também podem ser observados nas Figuras 6.13 e 6.14. Pelas Tabelas 6.31 e 6.32 é possível constatar que o modelo com DNS consegue enviar para a destilação uma variedade maior de misturas de petróleo.

Tabela 6.31: Alinhamentos ativos na destilação do modelo com DNS ao longo do horizonte de tempo para o cenário cinco.

	i1	i2	i3	i4	i5	i6
UC	KK	KK	KK	KK	-	-
	LL	LL	AA	II	II	LL
UN	DD	EE	EE	EE		
	FF	FF	FF	FF		
UV	LL	-	-	-	HH	LL
	GG	GG	GG	GG	GG	-

Observa-se nas Figuras 6.13 e 6.14 que o modelo com DS resultou em saltos maiores de propriedade TAN na alimentação das UDAs. A Tabela 6.33 exhibe o maior salto da propriedade TAN na alimentação das UDAs.

No modelo com DNS, Figura 6.13, o maior salto da propriedade TAN foi $0,16 \text{ mg}_{\text{KOH}}/\text{g}$ ($i1$ para $i2$), $0,19 \text{ mg}_{\text{KOH}}/\text{g}$ ($i1$ para $i2$) e $0,30 \text{ mg}_{\text{KOH}}/\text{g}$ ($i4$ para $i5$) nas UDAs UC, UN e

Tabela 6.32: Alinhamentos ativos na destilação do modelo com DS ao longo do horizonte de tempo para o cenário cinco.

	i1	i2	i3	i4
UC	AA	HH	II	-
	KK	KK	-	KK
	i1	i2	i3	i4
UN	FF	FF	FF	EE
	DD	-	DD	DD
	i1	i2	i3	
UV	GG	-	GG	
	LL	LL	-	

Tabela 6.33: Maior salto de TAN na alimentação das UDAs para o cenário cinco.

UDA	DNS	DS
UC	0,16	0,35
UN	0,19	0,21
UV	0,30	0,19

UV, respectivamente. No modelo com DS, Figura 6.14, o maior salto da propriedade TAN foi 0,35 mg_{KOH}/g (*i3* para *i4*), 0,21 mg_{KOH}/g (*i1* para *i2*) e 0,19 mg_{KOH}/g (*i2* para *i3*) nas UDAs UC, UN e UV, respectivamente.

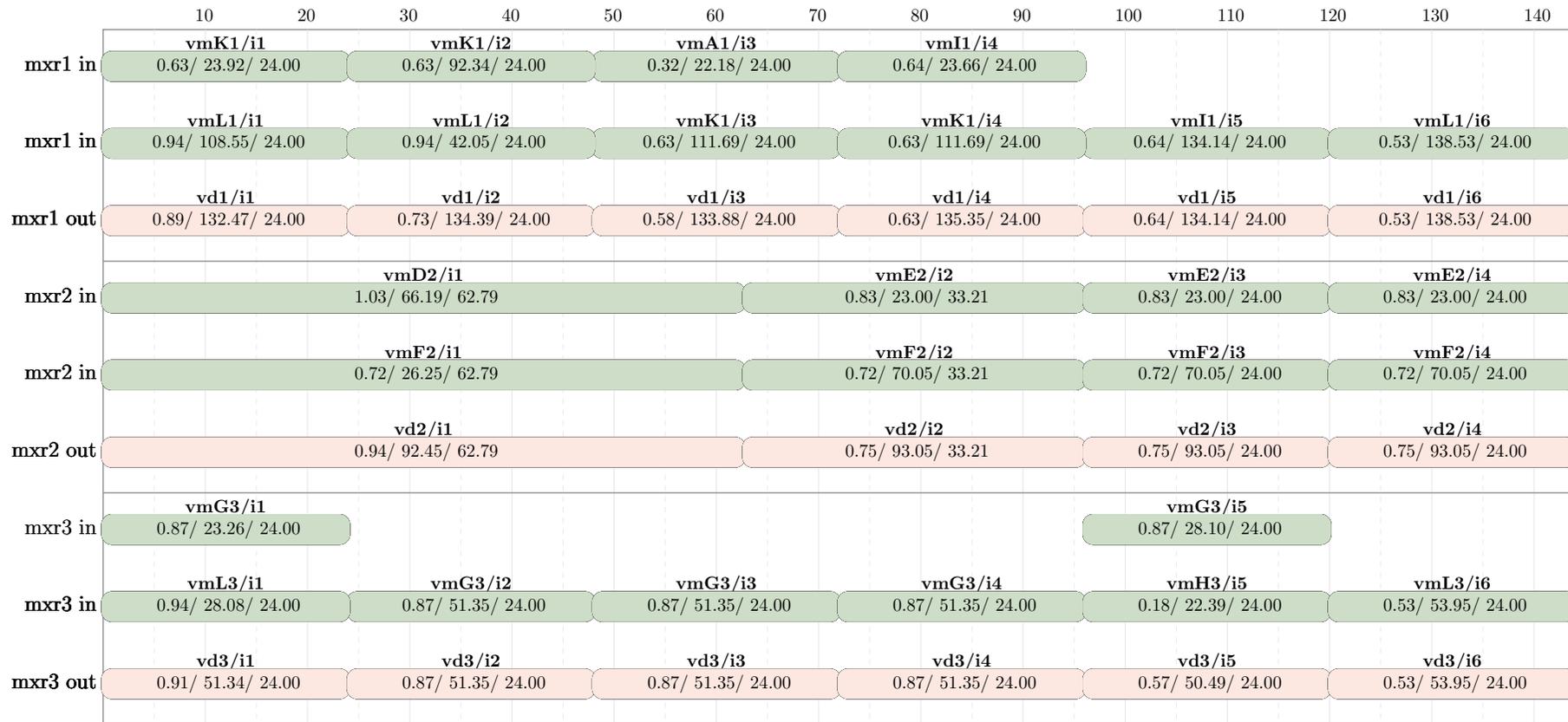


Figura 6.13: Programação nos misturadores e UDAs para o cenário cinco com $n = 6$ e DNS.

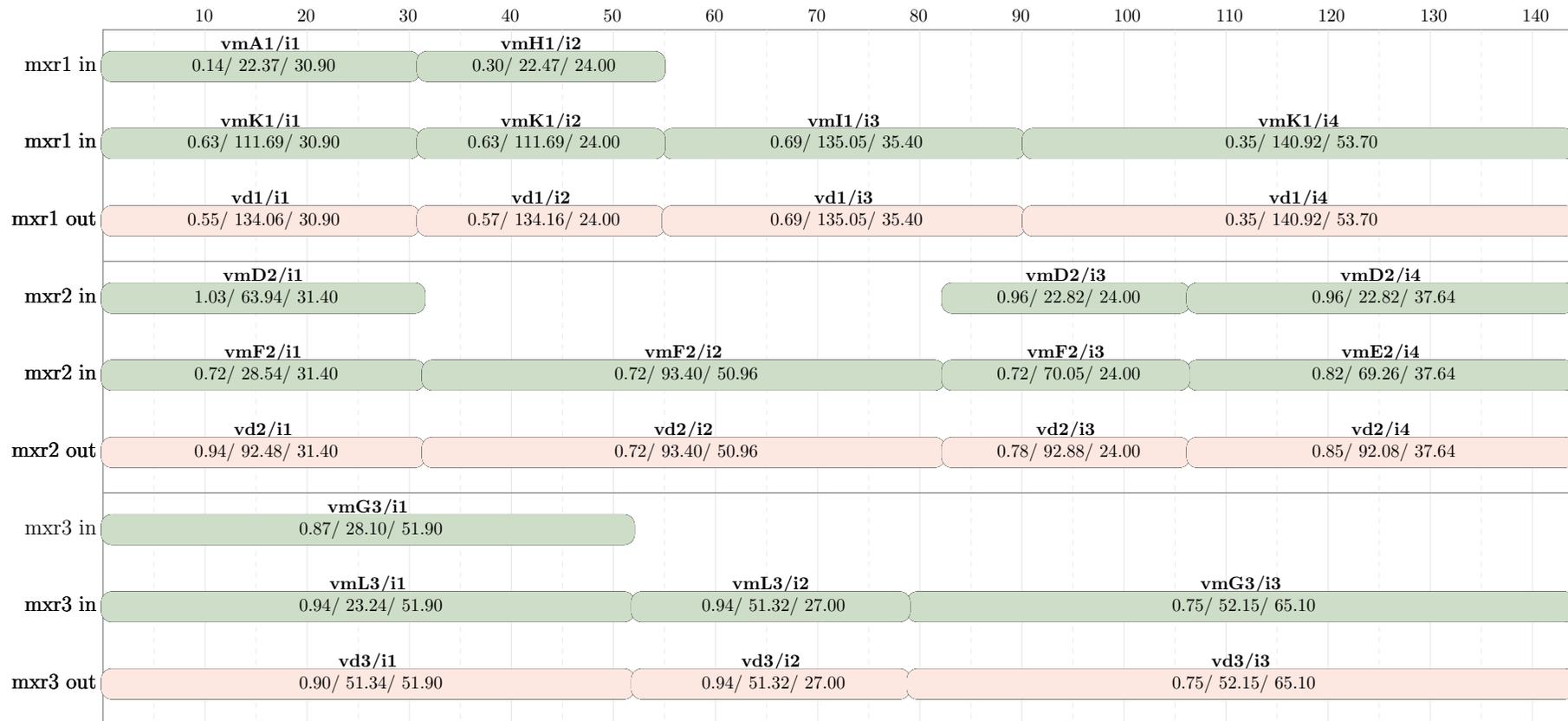


Figura 6.14: Programação nos misturadores e UDAs para o cenário cinco com $n = 4$ e DS.

Cenário seis

O melhor programa para o cenário seis foi o de $n = 5$, tanto na modelagem com DNS quanto na modelagem com DS.

O cenário seis foi o único para o qual se obteve o melhor programa com a mesma quantidade de *slots* de prioridade nas duas modelagens testadas. Como mostram as Tabelas 6.17 e 6.18 o melhor programa tem o mesmo número de variáveis nas duas modelagens. O modelo com DS tem mais equações porque usa cliques de menor cardinalidade em sua formulação.

As equações indexadas por conjunto clique na modelagem com DNS são 5.34, 5.36 e 5.38. Somadas elas são 285 equações – 95 Equações 5.34, 40 Equações 5.36 e 150 Equações 5.38 – e representam 4,46% de todas as equações lineares do problema.

As equações indexadas por conjunto clique na modelagem com DS são 5.34, 5.36, 5.39. Somadas elas são 825 equações – 175 Equações 5.34, 50 Equações 5.36 e 600 Equações 5.38 – e representam 11,69% de todas as equações lineares do problema.

A Equação 5.38 é responsável por sequenciar operações que envolvem tempo de preparação na modelagem com DNS. Na modelagem com DS esse papel é feito pela Equação 5.39. Tanto a Equação 5.38 quanto a Equação 5.39 são restrições do tipo *Big-M*.

A Figura 6.15 mostra a programação das operações de descarregamento W_U obtida pelos dois modelos no cenário seis. O modelo com DNS realizou em seu me-

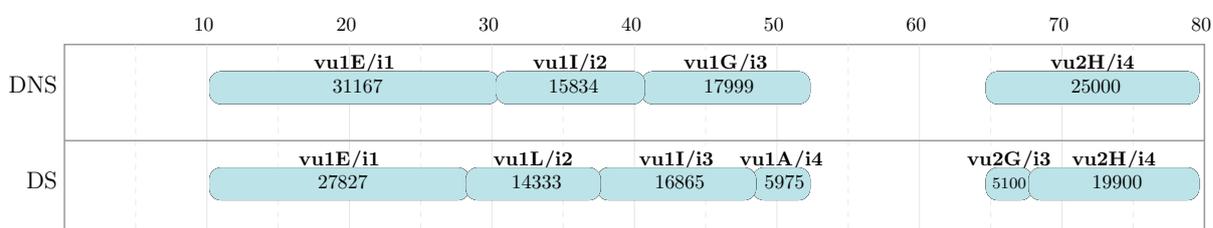


Figura 6.15: Programação do descarregamento de parcelas do melhor programa para o cenário seis.

lhor programa ($n = 5$) quatro operações de descarregamento de parcelas. A parcela *prc1* foi dividida entre os tanques EE, II e GG. A parcela *prc2* foi completamente des-

carregada para o tanque HH. O modelo com **DS** realizou em seu melhor programa ($n = 5$) seis operações de descarregamento de parcelas. A parcela *prc1* foi dividida entre os tanques EE, LL II e AA, e a parcela *prc2* entre os tanques GG e HH.

As Figuras 6.16 e 6.17 exibem o resultado do melhor programa nos misturadores e **UDAs**. Acima da barra no gráfico de Gantt é possível visualizar a qual operação ela se refere e em que *slot* de prioridade a operação acontece. Dentro da barra, pode-se visualizar da esquerda para a direita o valor da propriedade **TAN** da operação em $\text{mg}_{\text{KOH}}/\text{g}$, o valor marginal unitário relativo da mistura \dot{G}_{iv} em $\$/\text{h}$ e a duração da operação D_{iv} em h.

No cenário seis todas as três **UDAs** operam em vazão máxima em todo o horizonte de tempo nas duas modelagens.

No modelo com **DNS** o valor marginal das operações de destilação em UC, UN e UV é $33.040,98 \times 10^3 \$$, $22.481,03 \times 10^3 \$$ e $12.308,28 \times 10^3 \$$, respectivamente. Somando esses três valores marginais tem-se a função objetivo $67.830,30 \times 10^3 \$$. No modelo com **DS** o valor marginal das operações de destilação em UC, UN e UV é $32.869,58 \times 10^3 \$$, $22.476,91 \times 10^3 \$$ e $12.460,30 \times 10^3 \$$, respectivamente. Somando esses três valores marginais tem-se a função objetivo $67.806,80 \times 10^3 \$$. Apenas o valor marginal das misturas destiladas na **UDA** UV no modelo com **DS** supera a do modelo com **DNS**. Nas **UDAs** UC e UN o valor marginal das misturas destiladas é maior no modelo com **DNS**.

As Tabelas 6.34 e 6.35 exibem os alinhamentos entre tanques de carga e **UDAs** ativos ao longo do horizonte de tempo para os modelos com **DNS** e **DS**, respectivamente. Esses alinhamentos também podem ser observados nas Figuras 6.16 e 6.17. Pelas Tabelas 6.34 e 6.35 é possível constatar que o modelo com **DNS** consegue enviar para a destilação uma variedade maior de misturas de petróleo.

Observa-se nas Figuras 6.16 e 6.17 que o modelo com **DS** resulta em saltos maiores de propriedade **TAN** na alimentação das **UDAs**. A Tabela 6.36 exhibe o maior salto da propriedade **TAN** na alimentação das **UDAs**.

No modelo com **DNS**, Figura 6.16, o maior salto da propriedade **TAN** foi $0,51 \text{ mg}_{\text{KOH}}/\text{g}$ (i_4 para i_5), $0,14 \text{ mg}_{\text{KOH}}/\text{g}$ (i_2 para i_3) e $0,20 \text{ mg}_{\text{KOH}}/\text{g}$ (i_4 para i_5) nas **UDAs** UC, UN e UV, respectivamente. No modelo com **DS**, Figura 6.17, o maior salto da propriedade **TAN** foi $0,48 \text{ mg}_{\text{KOH}}/\text{g}$ (i_4 para i_5), $0,20 \text{ mg}_{\text{KOH}}/\text{g}$ (i_1 para i_2) e $0,21 \text{ mg}_{\text{KOH}}/\text{g}$ (i_4 para i_5)

Tabela 6.34: Alinhamentos ativos na destilação do modelo com [DNS](#) ao longo do horizonte de tempo para o cenário seis.

	i1	i2	i3	i4	i5
UC	GG	-	-	GG	GG
	HH	HH	II	II	HH
	i1	i2	i3		
UN	DD	EE	EE		
	FF	FF	FF		
	i1	i2	i3	i4	i5
UV	KK	KK	KK	KK	-
	LL	LL	LL	LL	LL

nas [UDAs](#) UC, UN e UV, respectivamente.

Tabela 6.35: Alinhamentos ativos na destilação do modelo com DS ao longo do horizonte de tempo para o cenário seis.

	i1	i2	i3	i4	i5
UC	HH	HH	-	GG	HH
	LL	-	LL	II	-
	i1	i2			
UN	DD	DD			
	FF	EE			
	i1	i2	i3	i4	i5
UV	KK	KK	KK	II	AA
	GG	GG	LL	LL	HH

Tabela 6.36: Maior salto de TAN na alimentação das UDAs para o cenário seis.

UDA	DNS	DS
UC	0,51	0,48
UN	0,14	0,20
UV	0,20	0,21

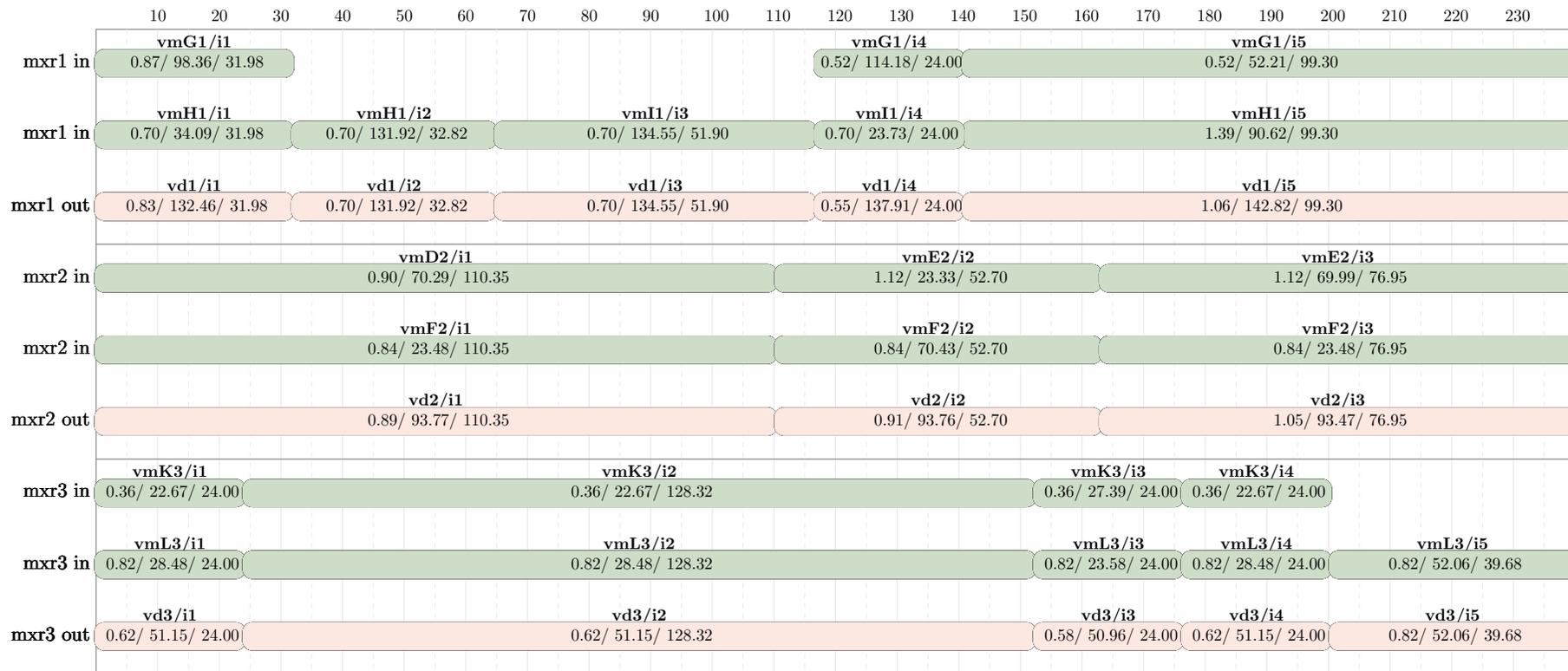


Figura 6.16: Programação nos misturadores e UDAs para o cenário seis com $n = 5$ e DNS.

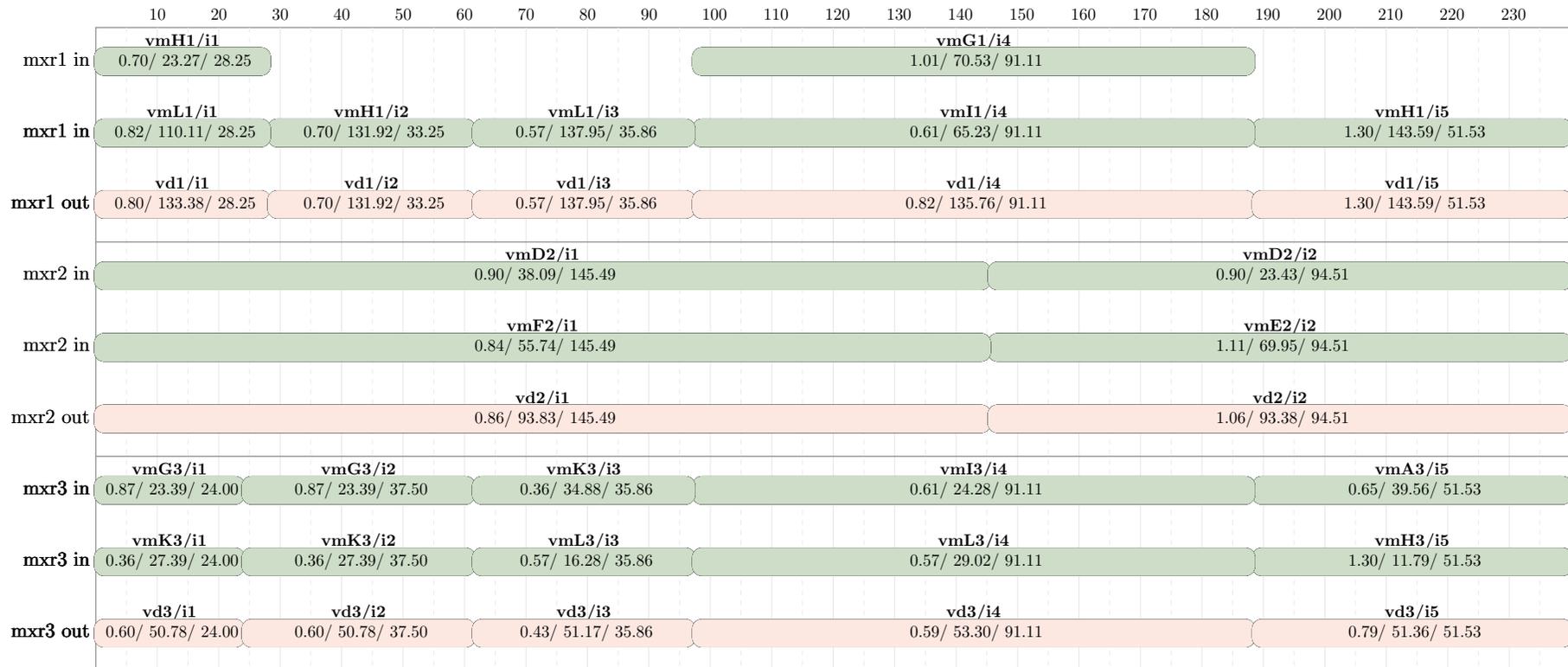


Figura 6.17: Programação nos misturadores e UDAs para o cenário seis com $n = 5$ e DS.

CONCLUSÕES E SUGESTÕES

Essa dissertação teve como objetivo principal investigar a viabilidade do uso de técnicas de otimização aplicadas à modelos para programação do suprimento de petróleo em uma refinaria real com regras operacionais específicas da indústria brasileira. Dois modelos foram desenvolvidos a partir da formulação de Mouret et al. (2011): um modelo com operações de descarregamento não-sobrepostas (DNS) e outro com operações de descarregamento sobrepostas (DS). Cada modelo foi aplicado a seis cenários diferentes no sistema de interesse e solucionado com estratégia de decomposição MILP-NLP.

Ambos modelos foram bem-sucedidos em representar o sistema produtivo de uma refinaria real e suas regras operacionais, e em encontrar soluções viáveis para o problema da programação do suprimento de petróleo. Soluções viáveis foram encontradas para os cenários dois, três, quatro, cinco e seis. Nenhum dos modelos foi capaz de encontrar soluções viáveis para o cenário um. A corrida correspondente ao melhor programa dos cenários otimizados foi solucionada pelo modelo com descarregamentos não-sobrepostos em até 28,32 min, e pelo modelo com descarregamentos sobrepostos em até 13,94 min. Os testes foram realizados em um *notebook* HP Pavilion dv7 com processador Intel i7-3610QM e 8 GB de RAM.

Problemas MILP são NP-hard. Além disso, os termos bilineares com as regras de mistura dão origem à regiões factíveis não-convexas para o programa NLP. Tendo em vista essas complicações, o equipamento usado para solucionar os modelos e a dimensão do problema – um sistema industrial com até cinco parcelas de petróleo, nove tanques de carga, três misturadores, três UDAs e trinta e seis tipos de petróleo – os tempos de solução são razoáveis e praticáveis em um empreendimento.

A formulação sequenciamento múltiplo de operações (MOS) de Mouret et al. (2011) mostrou-se eficiente para representar problemas de programação do supri-

mento de petróleo de grande dimensão. Os cenários otimizados deram origem à programas com até 27.964 variáveis contínuas, 567 variáveis binárias e 19.709 restrições. Mesmo depois de introduzidas várias restrições do tipo *Big-M* os modelos foram solucionados em tempos computacionais praticáveis por estratégia de decomposição MILP-NLP. Todas as soluções encontradas nessa dissertação são potenciais ótimos locais.

Melhores funções objetivo foram obtidas na modelagem com DNS e melhores tempos computacionais foram obtidos na modelagem com DS. Quanto aos resultados da programação, as duas modelagens se mostraram promissoras: o modelo com DS designa menos transições nas UDAs e o modelo com DNS realiza essas transições de forma mais suave, com menores saltos na propriedade TAN.

A estrutura do problema explorado neste trabalho é fortemente dependente da estratégia que define a quantidade de *slots* de prioridade que gera o melhor programa. Para determinar essa quantidade usou-se a abordagem aditiva considerando a função objetivo da etapa NLP. Se a função objetivo da etapa MILP fosse considerada, os resultados do melhor programa não seriam os mesmos.

A função objetivo do programa MINLP é maximizar a valor marginal das misturas de petróleo destiladas. Por conseguinte, os dois modelos testados nessa dissertação procuram enviar para as UDAs somente as misturas mais nobres de petróleo, e deixar as misturas menos nobres em inventário.

A modelagem com DNS rendeu melhores funções objetivo porque é mais eficiente em processar as misturas de petróleo mais nobres desde o começo do horizonte de programação. Ela consegue enviar para a destilação uma variedade maior de misturas de petróleo. Esse resultado é consequência de um número maior de transições permitidas nas UDAs no modelo com DNS.

O número de operações de destilação que podem acontecer em cada UDA é igual à quantidade de *slots* de prioridade usada para resolver o modelo. Na modelagem com DNS são necessários mais *slots* de prioridade para encontrar um programa factível e, conseqüentemente, são permitidas mais transições nas UDAs.

A modelagem com DS é capaz de fragmentar o descarregamento das parcelas em mais operações ao longo do horizonte de tempo, pois utiliza cliques disjuntos de menor cardinalidade em sua formulação. Ela retornou funções objetivo menores

porque o uso de cliques disjuntos para operações de descarregamento faz com que o modelo precise de menos *slots* de prioridade para encontrar um programa factível e, como consequência, limita o número de transições nas UDAs. Com menos transições nas UDAs o modelo é menos eficiente em destilar misturas de alto valor marginal.

A função objetivo usada nessa dissertação não considera custos relacionados ao processo como, por exemplo, custos de inventário, de movimentações, custos de transições nas UDAs e custos ocasionados por operação instável das UDAs decorrentes dos saltos de composição. Sugere-se para investigações futuras que se compare o desempenho dos modelos com DNS e DS condicionados a uma função objetivo que represente mais satisfatoriamente a realidade da refinaria. Outra possível extensão deste estudo é usar outros algoritmos de solução como DICOPT ou estratégia de *branching* local.

De acordo com a restrição de quebra de simetria, Equação 5.35, um operação só pode ser alocada ao *slot* de prioridade i se uma operação não-sobreposta a ela for alocada ao *slot* de prioridade $i - 1$. Essa premissa estabelece que um tanque que não começa na carga em $t = 0$ e não recebe descarregamento de parcela em $t = 0$ precisa receber pelo menos uma operação de descarregamento para ser ativado mais adiante no horizonte de programação.

Uma vez que a formulação MOS aproveita a estrutura do grafo de operações não-sobrepostas para criar modelos mais compactos, um sugestão para melhorar a performance global de modelos para programação do suprimento de petróleo concebidos a partir da formulação MOS é usar algoritmos para enumeração de cliques máximos. O grafo de operações não-sobrepostas possui 56, 40, 63, 45, 54, e 36 vértices nos cenários um, dois, três, quatro, cinco e seis, respectivamente.

Quando a janela de tempo disponível para programar a produção é curta ou é preciso reprogramá-la por causa de imprevistos, é indispensável que os modelos usados como ferramenta de suporte à decisão possam ser resolvidos rapidamente. Alguns artifícios podem ser empregados para melhorar o tempo computacional dos modelos. Para reduzir o tempo computacional da etapa MILP é possível usar procedimentos de seleção de nós e regras de *branching* apropriadas. Outra sugestão para reduzir o tempo computacional da etapa MILP é truncar a árvore de busca do algoritmo *Branch & Cut*. Uma sugestão para reduzir o tempo computacional da etapa

NLP é diminuir a quantidade de soluções armazenadas no *pool*. Tais medidas têm um impacto direto na qualidade da solução final.

Nessa dissertação apenas 20 soluções foram selecionadas para o *pool* com a finalidade de se obter bons tempos computacionais na etapa **NLP**. Outras abordagens podem ser usadas para resolver o problema. Uma abordagem mais pragmática é armazenar muitas soluções no *pool*, e solucionar o problema com faixas mais justas de *slots* de prioridade n . Armazenar mais soluções no *pool* aumenta a possibilidade de encontrar o ótimo global com estratégia de decomposição **MILP-NLP**.

Outra sugestão para trabalhos futuros é usar otimização robusta para conseguir soluções de cenários que geram modelos inviáveis, como foi o caso do cenário um.

Posto que programar a produção é mais uma questão de viabilidade do que de otimalidade, a escolha de métodos de solução para modelos de programação, assim como seus resultados devem ser pensados de forma crítica e aliados à experiência, habilidade e criatividade dos programadores.

Apêndices

APÊNDICE A

DADOS DE ENTRADA DOS CENÁRIOS

Todos os cenários

Lastro e capacidade dos tanques

Tabela A.1: Lastro e capacidade máxima nos tanques de carga R_C em m^3 .

tanques R_C	lastro \underline{L}_r^t	capacidade máxima \overline{L}_r^t
AA	5.000	29.300
BB	6.550	40.760
CC	4.920	29.220
DD	7.000	40.900
FF	7.100	42.200
HH	6.400	41.500
II	8.300	42.400
LL	6.900	40.500

Valor marginal dos petróleosTabela A.2: Valor marginal dos petróleos C .

$c \in C$	G_c (\$/m ³)	$c \in C$	G_c (\$/m ³)
P01	203,00	P19	272,00
P02	239,00	P20	220,00
P03	218,00	P21	218,00
P04	210,00	P22	266,00
P05	234,00	P23	216,00
P06	239,00	P24	228,00
P07	235,00	P25	214,00
P08	217,00	P26	201,00
P09	234,00	P27	234,00
P10	230,00	P28	221,00
P11	216,00	P29	224,00
P12	235,00	P30	208,00
P13	257,00	P31	234,00
P14	231,00	P32	265,00
P15	237,00	P33	257,00
P16	250,00	P34	228,00
P17	190,00	P35	252,00
P18	229,00	P36	257,00

Propriedades dos petróleos

Tabela A.3: Propriedades dos petróleos C .

$c \in C$	densidade (g cm^{-3})	acidez ($\text{mg}_{\text{KOH}}/\text{g}$)	enxofre ($\% \text{m/m}$)
P01	0,88550	0,15000	0,44000
P02	0,92480	1,90000	0,60000
P03	0,88550	0,27000	0,35200
P04	0,84450	1,02000	0,14700
P05	0,93460	1,19000	0,74000
P06	0,93030	1,24000	0,64000
P07	0,93030	1,24000	0,64000
P08	0,79010	0,05200	0,05000
P09	0,93400	1,22000	0,68000
P10	0,82890	0,12500	0,22100
P11	0,80080	0,05000	0,06300
P12	0,91530	0,59000	0,67000
P13	0,88550	0,09000	0,58000
P14	0,93400	1,92000	0,58500
P15	0,88940	0,49000	0,13660
P16	0,91590	0,70000	0,63100
P17	0,84070	0,30000	0,11000
P18	0,90420	0,62000	0,50000
P19	0,92120	1,82000	0,42000
P20	0,82710	0,34000	0,12000
P21	0,78130	0,03000	0,01100
P22	0,99300	1,12500	0,77240
P23	0,81780	0,06000	0,12100
P24	0,87890	0,29000	0,37600
P25	0,85040	0,33000	0,08000
P26	0,87830	0,10000	0,27000
P27	0,93770	1,25000	0,76700
P28	0,75710	0,02290	0,00720
P29	0,92790	0,92790	0,49600
P30	0,95410	3,35000	0,56000
P31	0,93770	1,25000	0,76700
P32	0,88660	0,17000	0,43000
P33	0,94650	2,69000	0,68800
P34	0,82240	0,13000	0,12500
P35	0,91700	0,74000	0,72000
P36	0,88550	0,09000	0,58000

Limites de vazãoTabela A.4: Limites de vazão de operações $O_{r \setminus RC2}$ em m^3/h .

operação $v \in W$	vazão mínima \underline{FR}_v	vazão máxima \overline{FR}_v
$v \in O_{CC} \cup O_{DD} \cup O_{EE}$	100	800
$v \in O_{FF}$	100	500
$v \in I_{UC}$	510,3	567
$v \in I_{UN}$	360	400
$v \in I_{UV}$	198,7	220,8

Tabela A.5: Limites de vazão na saída dos tanques de carga $RC2$ em m^3/h .

tanque $r \in RC2$	vazão mínima \underline{FRC}_r	vazão máxima \overline{FRC}_r
AA, BB, GG, HH, II	100	800
KK, LL	100	1000

Limites de volume transferido

O volume mínimo de transferência \underline{V}_{iv}^t foi calculado pela Equação A.1.

$$\underline{V}_{iv}^t = \underline{Dur}_v \cdot \underline{FR}_v \quad v \in W \quad (\text{A.1})$$

O volume máximo de transferência \overline{V}_{iv}^t para operações de descarregamento W_U , mistura W_M e destilação W_D foram calculados pelas Equações A.2, A.3 e A.4, respectivamente.

$$\overline{V}_{iv}^t = L_{0r}^t \quad v \in O_r, r \in R_P, \quad (\text{A.2})$$

$$\overline{V}_{iv}^t = \overline{L}_r^t - \underline{L}_r^t \quad v \in O_r, r \in R_C, \quad (\text{A.3})$$

$$\overline{V}_{iv}^t = 2 \cdot \max_{r'} (\overline{L}_{r'}^t - \underline{L}_{r'}^t) \quad v \in O_r, r \in R_M, r' \in R_C \quad (\text{A.4})$$

Cenário um

$H = 168$ h

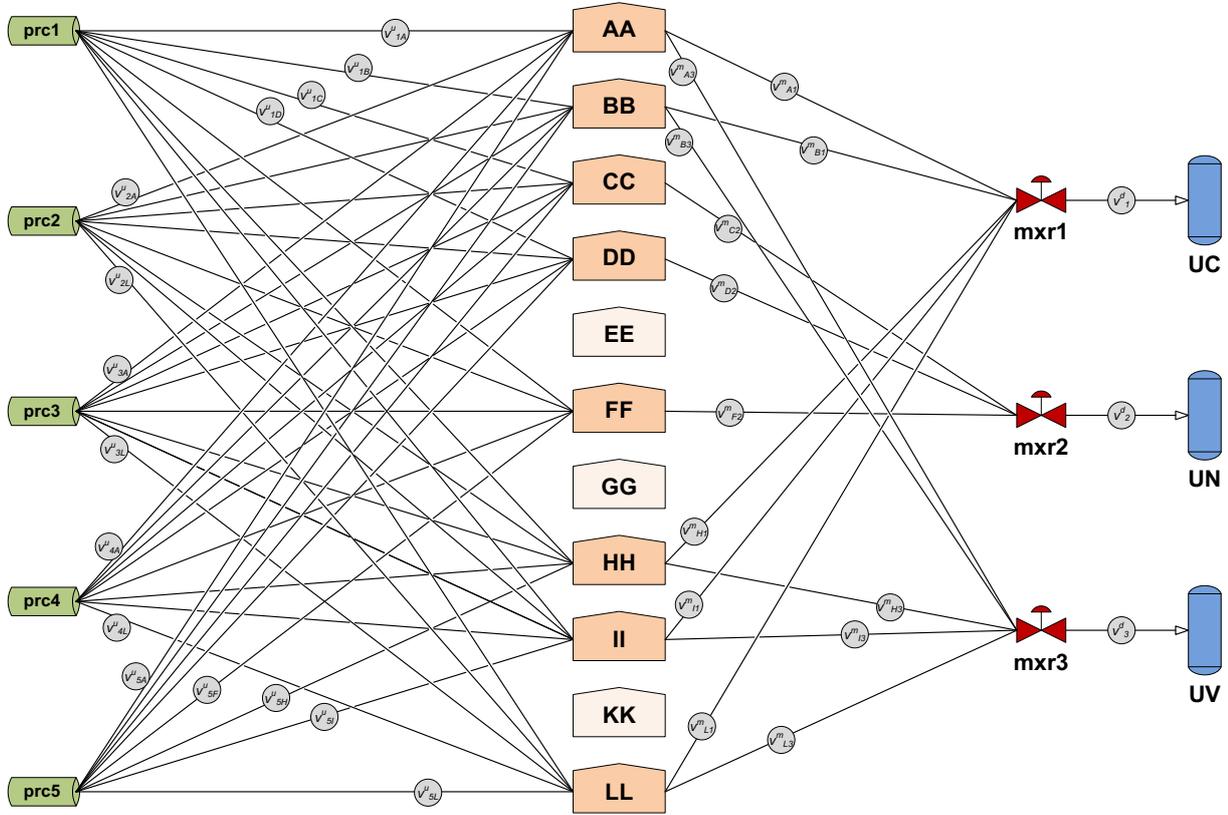


Figura A.1: Cenário 1.

Dados das parcelas

Tabela A.6: Dados das parcelas no cenário um.

Dado	Unidade	prc1	prc2	prc3	prc4	prc5
COMEÇO	h	0,00	21,80	40,60	63,60	83,60
VAZÃO	m ³ h ⁻¹	1.700,00	1.700,00	1.700,00	1.700,00	1.700,00
P02	m ³	22.000,00	0,00	0,00	0,00	0,00
P11	m ³	0,00	0,00	0,00	0,00	24.000,00
P19	m ³	0,00	32.000,00	0,00	0,00	0,00
P03	m ³	0,00	0,00	24.000,00	0,00	0,00
P27	m ³	0,00	0,00	0,00	34.000,00	0,00
FIM	h	12,94	40,62	54,72	83,60	97,72

Inventário inicial nos tanques de cargaTabela A.7: Inventário inicial nos tanques de carga R_C em m^3 no cenário um.

$c \in C$	AA	BB	CC	DD	FF	HH	II	LL
P01	9.995,45	1.681,50	0,00	556,20	2.585,77	4.560,09	3.300,88	0,00
P02	237,14	13.651,50	633,05	7.258,41	21.799,99	1.258,92	780,21	0,00
P03	1.624,41	0,00	316,53	0,00	676,28	223,81	3.075,82	2.283,46
P04	0,00	0,00	0,00	152,96	0,00	0,00	0,00	6.362,19
P05	0,00	370,50	0,00	180,77	0,00	0,00	0,00	0,00
P06	0,00	4.816,50	836,53	389,34	7.995,98	895,23	1.830,49	0,00
P07	0,00	484,50	1.266,10	236,39	2.784,67	195,83	6.451,72	0,00
P08	0,00	0,00	5.923,56	625,73	0,00	307,74	510,14	4.157,47
P09	0,00	2.878,50	0,00	1.557,36	0,00	1.468,74	0,00	0,00
P10	0,00	1.083,00	0,00	2.947,86	716,06	1.510,70	0,00	377,95
P11	0,00	2.992,50	497,40	0,00	795,62	0,00	4.096,09	1.401,57
P12	0,00	0,00	0,00	0,00	1.233,21	97,92	2.760,74	0,00
P13	0,00	541,50	1.786,11	0,00	716,06	265,77	4.711,26	330,71
P14	0,00	0,00	0,00	0,00	477,37	2.531,83	2.490,66	0,00
P15	0,00	0,00	0,00	0,00	0,00	167,86	0,00	220,47
P16	0,00	0,00	248,70	0,00	0,00	503,57	0,00	0,00
P17	0,00	0,00	452,18	0,00	0,00	0,00	0,00	0,00
P18	0,00	0,00	474,79	0,00	0,00	0,00	0,00	0,00
P19	0,00	0,00	565,23	0,00	0,00	0,00	0,00	0,00
P20	0,00	0,00	610,44	0,00	0,00	0,00	0,00	0,00
P21	0,00	0,00	2.735,69	0,00	0,00	0,00	0,00	0,00
P22	0,00	0,00	5.607,03	0,00	0,00	0,00	0,00	0,00
P23	0,00	0,00	339,14	0,00	0,00	0,00	0,00	0,00
P24	0,00	0,00	316,53	0,00	0,00	0,00	0,00	0,00
P25	0,00	0,00	0,00	0,00	0,00	0,00	0,00	236,22
P26	0,00	0,00	0,00	0,00	0,00	0,00	0,00	377,95
TOTAL	11.857,00	28.500,00	22.609,01	13.905,02	39.781,01	13.988,01	30.008,01	15.747,99

Volume útil inicial nos tanques da cargaTabela A.8: Volume útil inicial nos tanques de carga R_C em m^3 no cenário um.

tanques R_C	volume inicial L_{0r}^t	lastro L_r^t	volume útil inicial
AA	11.857,00	5.000,00	6.857,00
BB	28.500,00	6.550,00	21.950,00
CC	22.609,00	4.920,00	17.689,01
DD	13.905,00	7.000,00	6.905,02
FF	39.781,00	7.100,00	32.681,01
HH	13.988,00	6.400,00	7.588,01
II	30.008,00	8.300,00	21.708,01
LL	15.748,00	6.900,00	8.847,99

Cenário dois

$H = 168$ h

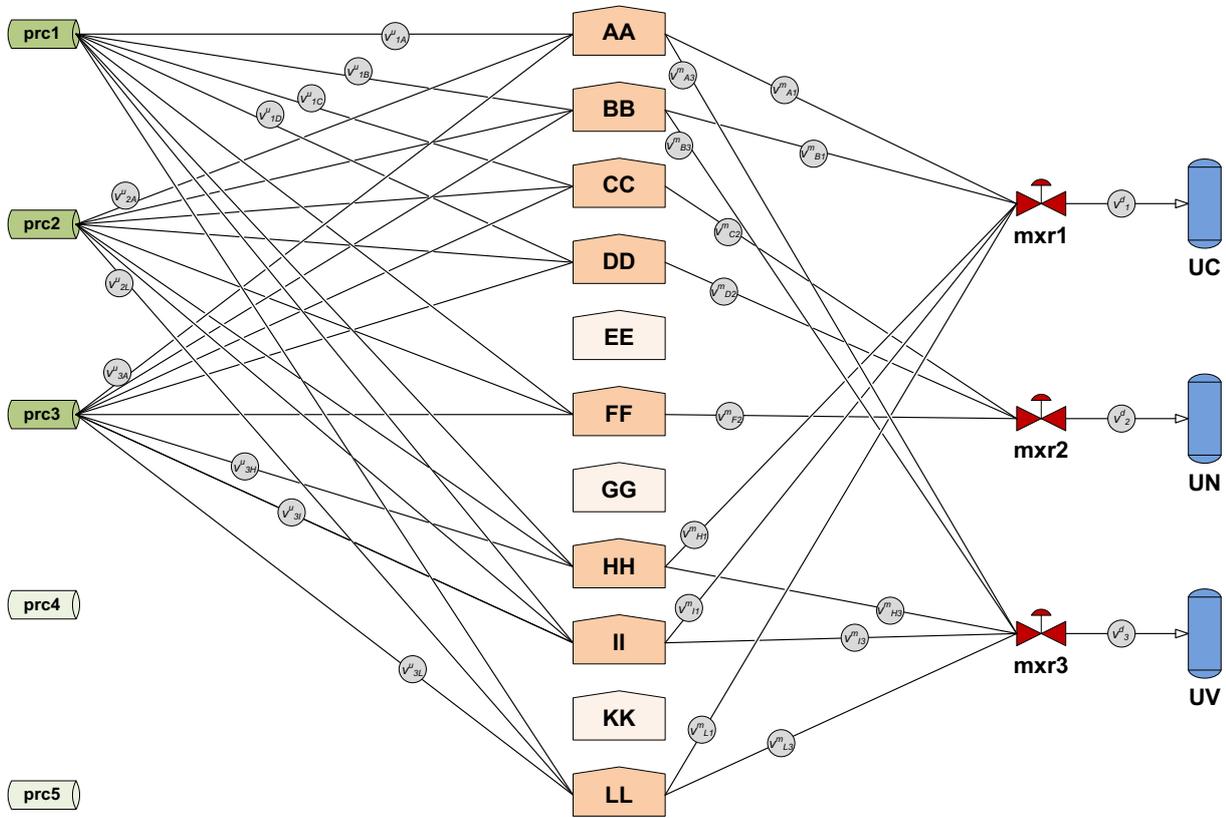


Figura A.2: Cenário 2.

Dados das parcelas

Tabela A.9: Dados das parcelas no cenário dois.

Dado	Unidade	prc1	prc2	prc3
COMEÇO	h	0,00	29,00	63,10
VAZÃO	m ³ h ⁻¹	1.200,00	1.700,00	1.700,00
P03	m ³	24.000,00	0,00	0,00
P05	m ³	0,00	58.000,00	0,00
P08	m ³	0,00	0,00	30.000,00
FIM	h	20,00	63,12	80,75

Inventário inicial nos tanques de carga

Tabela A.10: Inventário inicial nos tanques de carga R_C em m^3 no cenário dois.

$c \in C$	AA	BB	CC	DD	FF	HH	II	LL
P01	3.935,79	1.417,65	0,00	4.405,40	1.822,60	20.714,94	913,11	0,00
P02	90,02	11.509,41	633,05	7.191,72	15.365,92	6.128,47	215,83	0,00
P03	740,15	0,00	316,53	0,00	476,68	0,00	850,85	1.970,84
P04	0,00	0,00	0,00	0,00	0,00	0,00	0,00	5.491,17
P05	0,00	312,36	0,00	0,00	0,00	0,00	0,00	0,00
P06	0,00	4.060,73	836,53	4.706,63	5.636,04	2.723,76	506,36	0,00
P07	0,00	408,48	1.266,10	13.517,43	1.962,80	0,00	1.784,72	0,00
P08	0,00	0,00	5.923,56	0,00	0,00	0,00	141,12	3.588,29
P09	0,00	2.426,83	0,00	790,71	0,00	1.720,27	0,00	0,00
P10	0,00	913,06	0,00	1.355,51	504,72	1.505,24	0,00	326,21
P11	0,00	2.522,94	497,40	0,00	560,80	0,00	1.133,09	1.209,69
P12	0,00	0,00	0,00	5.685,60	869,24	0,00	763,69	0,00
P13	0,00	456,53	1.786,11	0,00	504,72	0,00	1.303,26	285,43
P14	0,00	0,00	0,00	0,00	336,48	2.544,57	688,98	0,00
P15	235,05	0,00	0,00	0,00	0,00	0,00	0,00	190,29
P16	0,00	0,00	248,70	0,00	0,00	501,75	0,00	0,00
P17	0,00	0,00	452,18	0,00	0,00	0,00	0,00	0,00
P18	0,00	0,00	474,79	0,00	0,00	0,00	0,00	0,00
P19	0,00	0,00	565,23	0,00	0,00	0,00	0,00	0,00
P20	0,00	0,00	610,44	0,00	0,00	0,00	0,00	0,00
P21	0,00	0,00	2.735,69	0,00	0,00	0,00	0,00	0,00
P22	0,00	0,00	5.607,03	0,00	0,00	0,00	0,00	0,00
P23	0,00	0,00	339,14	0,00	0,00	0,00	0,00	0,00
P24	0,00	0,00	316,53	0,00	0,00	0,00	0,00	0,00
P25	0,00	0,00	0,00	0,00	0,00	0,00	0,00	203,88
P26	0,00	0,00	0,00	0,00	0,00	0,00	0,00	326,21
Total	5.001,01	24.027,99	22.609,01	37.653,00	28.040,00	35.839,00	8.301,01	13.592,01

Volume útil inicial nos tanques da carga

Tabela A.11: Volume útil inicial nos tanques de carga R_C em m^3 no cenário dois.

tanques R_C	volume inicial L_{0r}^t	lastro L_r^t	volume útil inicial
AA	5.001,00	5.000,00	1,01
BB	24.028,00	6.550,00	17.477,99
CC	22.609,00	4.920,00	17.689,01
DD	37.653,00	7.000,00	30.653,00
FF	28.040,00	7.100,00	20.940,00
HH	35.839,00	6.400,00	29.439,00
II	8.301,00	8.300,00	1,01
LL	13.592,00	6.900,00	6.692,01

Cenário três

$H = 240$ h

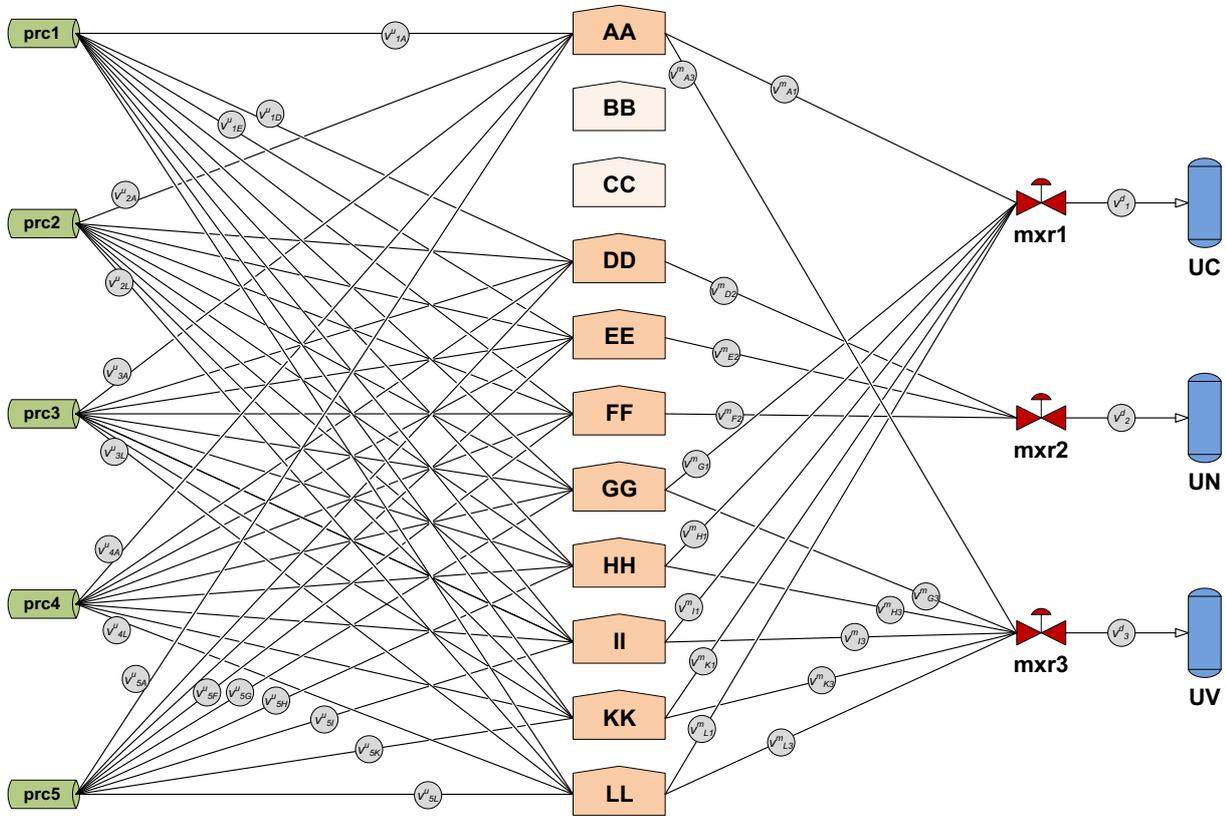


Figura A.3: Cenário 3.

Dados das parcelas

Tabela A.12: Dados das parcelas no cenário três.

Dado	Unidade	prc1	prc2	prc3	prc4	prc5
COMEÇO	h	0,00	10,00	44,30	71,50	95,20
VAZÃO	m ³ h ⁻¹	1.600,00	1.450,00	1.600,00	1.600,00	1.600,00
P15	m ³	1.780,75	0,00	0,00	0,00	0,00
P31	m ³	5.342,25	0,00	0,00	0,00	60.000,00
P16	m ³	0,00	30.000,00	0,00	0,00	0,00
P13	m ³	0,00	0,00	5.000,00	20.000,00	0,00
FIM	h	4,45	30,69	47,43	84,00	132,70

Inventário inicial nos tanques de carga

Tabela A.13: Inventário inicial nos tanques de carga R_C em m^3 no cenário três.

$c \in C$	AA	DD	EE	FF	GG	HH	II	KK	LL
P01	315,00	2.470,72	13.972,27	907,67	5.453,51	11.365,39	3.189,39	0,00	0,00
P02	0,00	5.735,60	3.473,55	851,93	2.998,14	10.068,67	0,00	0,00	0,00
P03	0,00	0,00	780,57	0,00	1.033,84	953,47	0,00	1.212,73	0,00
P04	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P05	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P06	90,00	3.750,20	1.639,21	525,49	1.783,37	5.682,70	167,34	543,64	0,00
P07	280,00	1.478,02	0,00	214,97	0,00	839,06	689,07	0,00	909,54
P08	180,00	0,00	1.678,23	214,97	0,00	0,00	206,72	23.878,25	4.783,50
P09	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P10	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P11	0,00	2.272,18	0,00	2.420,45	413,54	724,64	0,00	13.340,04	23.445,87
P12	120,00	661,80	0,00	95,54	0,00	419,53	315,00	0,00	437,93
P13	3.751,00	4.389,94	2.888,12	135,35	852,92	4.309,70	2.460,95	0,00	0,00
P14	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1.212,73	0,00
P16	155,00	882,40	0,00	191,09	0,00	0,00	246,10	585,46	0,00
P17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P18	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P19	0,00	0,00	0,00	0,00	516,92	0,00	0,00	0,00	0,00
P20	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P21	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P22	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P23	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P24	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P25	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	404,24
P26	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P27	0,00	0,00	0,00	238,86	0,00	0,00	118,13	585,46	0,00
P28	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P29	110,00	419,14	468,34	0,00	491,07	0,00	118,13	0,00	0,00
P30	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P31	0,00	0,00	12.762,38	1.910,88	12.302,70	3.775,75	2.332,98	460,00	0,00
P32	0,00	0,00	0,00	254,78	0,00	0,00	0,00	0,00	2.762,30
P33	0,00	0,00	1.366,00	0,00	0,00	0,00	0,00	0,00	0,00
P34	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	943,22
P35	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P36	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
TOTAL	5.001,00	22.060,00	39.028,67	7.961,98	25.846,01	38.138,91	9.843,81	41.818,31	33.686,60

Volume útil inicial nos tanques da cargaTabela A.14: Volume útil inicial nos tanques de carga R_C em m^3 no cenário três.

tanques R_C	volume inicial L_{0r}^t	lastro $\underline{L_r^t}$	volume útil inicial
AA	5.001,00	5.000,00	1,00
DD	22.060,00	7.000,00	15.060,00
EE	39.028,70	7.100,00	31.928,67
FF	7.962,00	7.100,00	861,98
GG	25.846,00	6.900,00	18.946,01
HH	38.138,90	6.400,00	31.738,91
II	9.843,80	8.300,00	1.543,81
KK	41.818,30	6.900,00	34.918,31
LL	33.686,60	6.900,00	26.786,60

Cenário quatro

$H = 168$ h

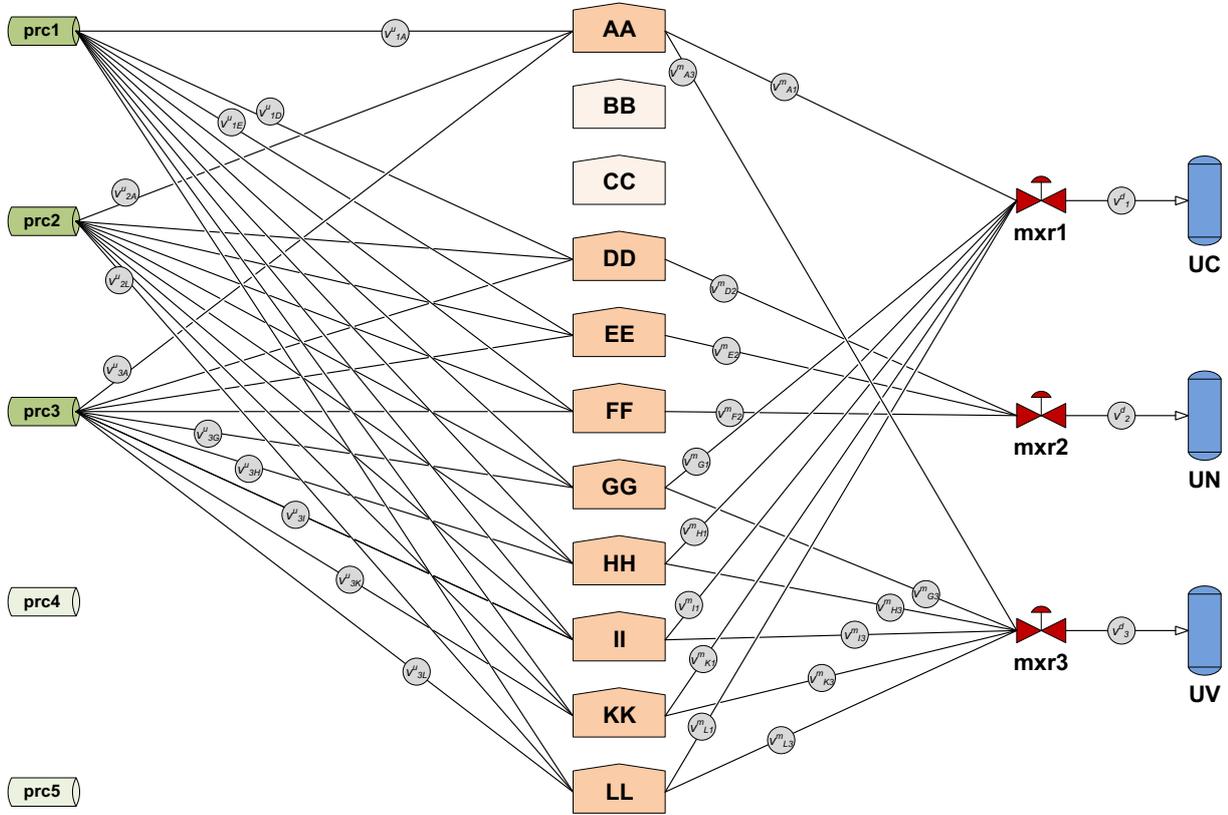


Figura A.4: Cenário 4.

Dados das parcelas

Tabela A.15: Dados das parcelas no cenário quatro.

Dado	Unidade	prc1	prc2	prc3
COMEÇO	h	0,0	11,0	31,0
VAZÃO	$\text{m}^3 \text{h}^{-1}$	1.600,0	1.750,0	2.000,0
P15	m^3	0,0	6.300,0	27.000,0
P30	m^3	0,0	10.500,0	0,0
P36	m^3	0,0	18.200,0	0,0
P13	m^3	2.830,0	0,0	0,0
FIM	h	1,8	31,0	44,5

Inventário inicial nos tanques de carga

Tabela A.16: Inventário inicial nos tanques de carga R_C em m^3 no cenário quatro.

$c \in C$	AA	DD	EE	FF	GG	HH	II	KK	LL
P01	739,54	11.148,14	2.092,50	3.682,67	2.121,98	6.681,15	1.961,83	0,00	0,00
P02	796,43	0,00	896,78	1.644,83	117,89	1.208,97	918,30	0,00	0,00
P03	910,21	1.099,11	1.606,74	1.673,94	1.193,62	0,00	542,63	0,00	0,00
P04	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P05	2.559,96	0,00	10.649,31	0,00	0,00	0,00	11.478,78	0,00	0,00
P06	767,99	471,05	186,83	858,80	110,52	731,75	751,34	0,00	0,00
P07	1.137,76	13.032,33	4.932,31	0,00	368,40	4.931,33	4.508,03	0,00	0,00
P08	0,00	4.788,99	0,00	1.761,28	839,95	2.863,35	459,15	20.662,26	2.119,66
P09	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P10	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	2.150,62	3.163,43
P12	540,44	5.691,83	2.428,79	0,00	176,83	2.417,94	2.128,79	0,00	0,00
P13	5.518,14	0,00	10.462,48	0,00	0,00	12.312,41	18.366,04	435,57	0,00
P14	0,00	2.590,76	1.233,08	4.163,02	1.945,15	668,12	0,00	0,00	0,00
P15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P16	312,88	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P18	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P19	0,00	0,00	0,00	0,00	0,00	0,00	0,00	3.185,09	0,00
P20	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P21	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P22	568,88	0,00	0,00	0,00	0,00	0,00	0,00	789,47	353,28
P23	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P24	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P25	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	9.763,26
P26	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P27	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P28	14.591,77	0,00	2.428,79	0,00	0,00	0,00	0,00	0,00	0,00
P29	0,00	431,79	0,00	0,00	353,66	0,00	0,00	0,00	289,04
P30	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P31	0,00	0,00	448,39	771,47	139,99	0,00	626,12	0,00	0,00
P32	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	369,33
P33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P34	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P35	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P36	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
TOTAL	28.444,00	39.254,00	37.366,00	14.556,01	7.367,99	31.815,02	41.741,01	27.223,01	16.058,00

Volume útil inicial nos tanques da cargaTabela A.17: Volume útil inicial nos tanques de carga R_C em m^3 no cenário quatro.

tanques R_C	volume inicial L_{0r}^t	lastro \underline{L}_r^t	volume útil inicial
AA	28.444,00	5.000,00	23.444,00
DD	39.254,00	7.000,00	32.254,00
EE	37.366,00	7.100,00	30.266,00
FF	14.556,00	7.100,00	7.456,01
GG	7.368,00	6.900,00	467,99
HH	31.815,00	6.400,00	25.415,02
II	41.741,00	8.300,00	33.441,01
KK	27.223,00	6.900,00	20.323,01
LL	16.058,00	6.900,00	9.158,00

Cenário cinco

$H = 144$ h

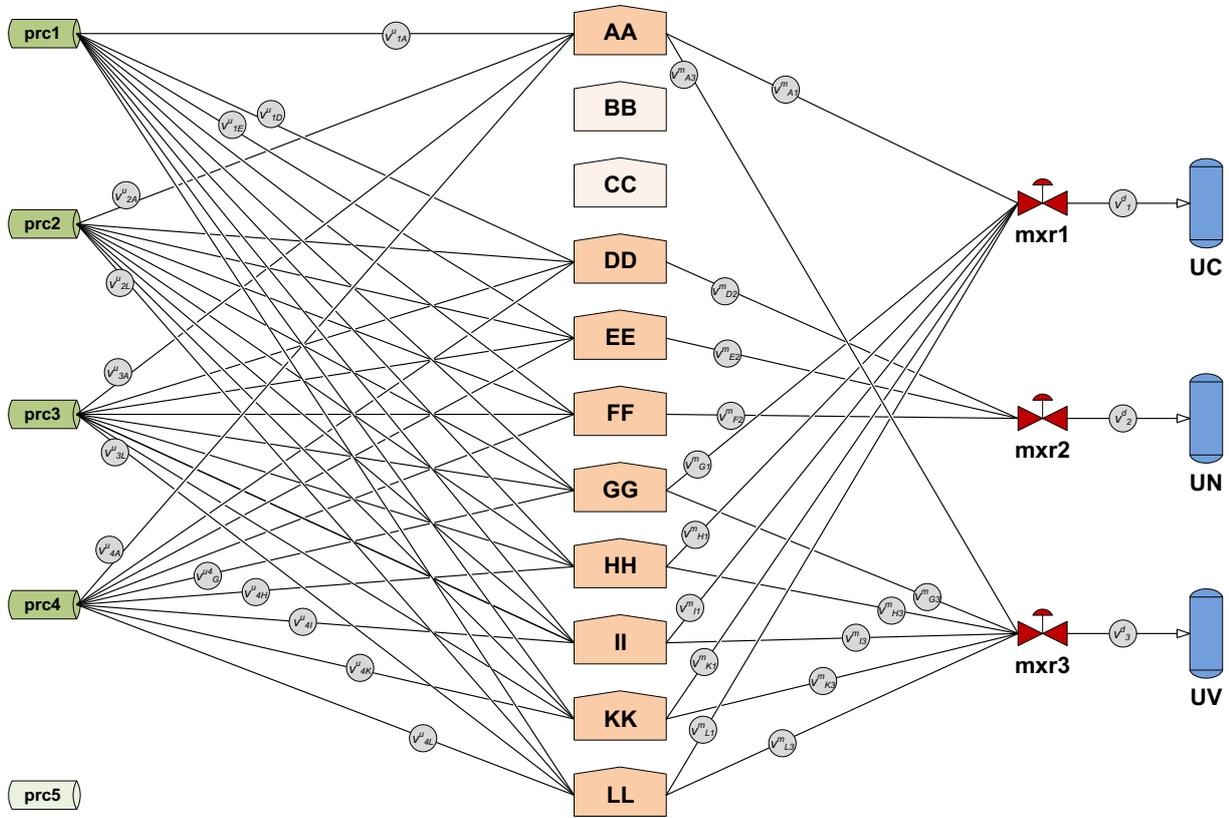


Figura A.5: Cenário 5.

Dados das parcelas

Tabela A.18: Dados das parcelas no cenário cinco.

Dado	Unidade	prc1	prc2	prc3	prc4
COMEÇO	h	0,00	31,40	39,10	51,90
VAZÃO	$\text{m}^3 \text{h}^{-1}$	1.800,00	1.300,00	1.800,00	1.800,00
P08	m^3	8.500,00	2.500,00	0,00	0,00
P29	m^3	25.500,00	7.500,00	0,00	0,00
P21	m^3	0,00	0,00	15.000,00	0,00
P13	m^3	0,00	0,00	0,00	25.000,00
FIM	h	18,89	39,09	47,43	65,79

Inventário inicial nos tanques de carga

Tabela A.19: Inventário inicial nos tanques de carga R_C em m^3 no cenário cinco.

$c \in C$	AA	DD	EE	FF	GG	HH	II	KK	LL
P01	0,00	911,30	811,13	5.046,50	0,00	997,62	1.502,92	520,64	0,00
P02	0,00	1.981,10	1.581,70	0,00	8.971,20	0,00	0,00	1.301,61	10.034,00
P03	1.530,53	4.477,27	1.439,75	3.852,29	8.672,16	6.190,36	988,22	6.160,95	4.013,60
P04	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P05	208,71	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P06	0,00	1.030,17	1.419,47	0,00	5.121,06	0,00	0,00	694,19	5.351,46
P07	222,62	2.099,96	2.210,32	13.521,54	1.719,48	1.330,16	5.394,06	1.561,93	2.321,59
P08	0,00	673,57	1.297,80	0,00	1.420,44	0,00	0,00	2.516,45	7.082,82
P09	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P10	0,00	10.935,64	304,17	1.964,67	5.831,28	14.861,98	0,00	6.681,60	393,49
P11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P12	0,00	990,55	973,35	6.394,80	822,36	639,50	2.573,50	780,97	1.141,12
P13	500,90	8.479,09	5.373,72	1.810,58	1.943,76	486,02	8.441,08	8.026,60	3.344,67
P14	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P15	0,00	0,00	0,00	385,23	0,00	0,00	0,00	0,00	0,00
P16	0,00	0,00	608,35	0,00	0,00	0,00	0,00	0,00	0,00
P17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P18	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P19	0,00	0,00	0,00	1.695,01	0,00	409,28	1.317,63	0,00	0,00
P20	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P21	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P22	403,50	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P23	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P24	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P25	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P26	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P27	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P28	11.047,64	0,00	0,00	0,00	1.046,64	665,08	0,00	0,00	0,00
P29	0,00	0,00	3.832,58	963,07	0,00	0,00	0,00	0,00	5.076,02
P30	0,00	8.043,25	425,84	0,00	485,94	0,00	0,00	4.165,15	0,00
P31	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P32	0,00	0,00	0,00	2.889,22	1.345,68	0,00	0,00	9.414,98	590,24
P33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P34	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P35	0,00	0,00	0,00	0,00	0,00	0,00	370,58	1.561,93	0,00
P36	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
TOTAL	13.913,90	39.621,90	20.278,18	38.522,91	37.380,00	25.580,00	20.587,99	43.387,00	39.349,01

Volume útil inicial dos tanques da cargaTabela A.20: Volume útil inicial nos tanques de carga R_C em m^3 no cenário cinco.

tanques R_C	volume inicial L_{0r}^t	lastro \underline{L}_r^t	volume útil inicial
AA	13.913,90	5.000,00	8.913,90
DD	39.621,90	7.000,00	32.621,90
EE	20.278,20	7.100,00	13.178,18
FF	38.522,90	7.100,00	31.422,91
GG	37.380,00	6.900,00	30.480,00
HH	25.580,00	6.400,00	19.180,00
II	20.588,00	8.300,00	12.287,99
KK	43.387,00	6.900,00	36.487,00
LL	39.349,00	6.900,00	32.449,01

Cenário seis

$H = 240$ h

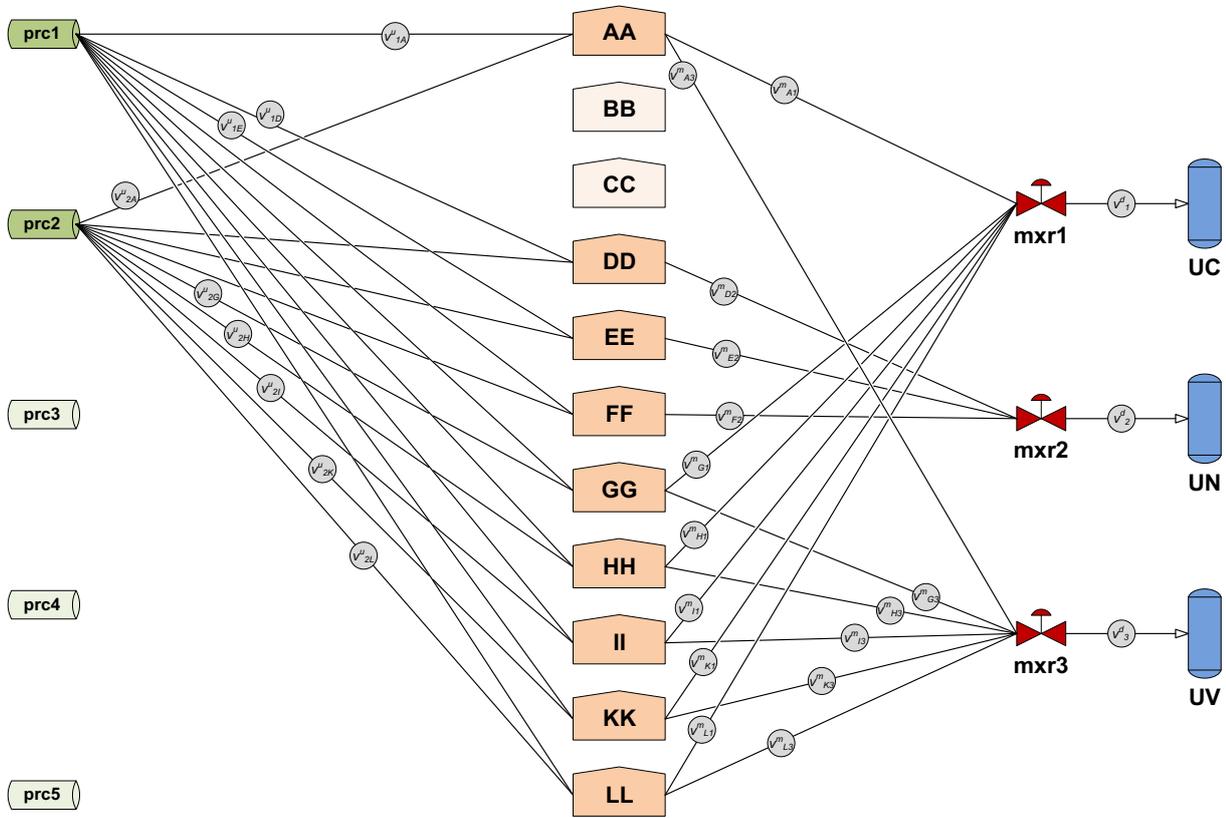


Figura A.6: Cenário 6.

Dados das parcelas

Tabela A.21: Dados das parcelas no cenário seis.

Dado	Unidade	prc1	prc2
COMEÇO	h	10,30	64,80
VAZÃO	$\text{m}^3 \text{h}^{-1}$	1.550,00	1.700,00
P07	m^3	39.000,00	0,00
P13	m^3	26.000,00	0,00
P19	m^3	0,00	25.000,00
FIM	h	52,24	79,51

Inventário inicial nos tanques de carga

Tabela A.22: Inventário inicial nos tanques de carga R_C em m^3 no cenário seis.

$c \in C$	AA	DD	EE	FF	GG	HH	II	KK	LL
P01	0,00	441,17	167,00	3.456,12	0,00	477,54	587,31	0,00	0,00
P02	613,14	1.002,65	423,05	582,96	1.876,88	915,29	280,89	1.952,50	2.637,19
P03	1.029,21	2.526,68	590,05	2.998,08	2.050,66	4.218,27	893,73	1.464,37	1.495,72
P04	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P05	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P06	372,27	882,33	378,52	416,40	1.807,36	636,72	0,00	1.179,63	1.849,97
P07	284,67	6.256,54	590,05	11.451,00	13.868,04	3.422,37	2.910,99	1.016,93	7.793,48
P08	0,00	0,00	1.603,15	1.790,52	3.197,64	2.706,06	3.268,48	25.585,83	2.086,13
P09	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P10	0,00	5.735,16	378,52	1.540,68	834,17	11.341,58	740,52	1.505,05	393,61
P11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P12	0,00	3.007,95	267,19	5.121,72	6.256,26	1.631,60	1.378,89	488,12	3.778,66
P13	591,25	10.427,56	1.425,02	5.163,36	2.989,10	5.969,25	5.158,07	3.660,93	9.131,75
P14	0,00	0,00	0,00	5.288,28	0,00	8.038,59	0,00	0,00	0,00
P15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P16	0,00	0,00	167,00	0,00	0,00	0,00	0,00	0,00	0,00
P17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P18	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P19	0,00	0,00	0,00	1.165,92	0,00	0,00	510,70	0,00	0,00
P20	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P21	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P22	262,78	0,00	0,00	0,00	0,00	0,00	0,00	813,54	0,00
P23	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P24	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P25	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P26	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P27	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P28	15.022,03	0,00	0,00	0,00	0,00	437,75	0,00	0,00	0,00
P29	0,00	5.374,20	4.764,92	666,24	1.876,88	0,00	9.192,60	0,00	9.525,36
P30	547,45	4.451,77	378,52	0,00	0,00	0,00	612,84	935,57	669,14
P31	3.175,21	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P32	0,00	0,00	0,00	1.998,72	0,00	0,00	0,00	2.074,53	0,00
P33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P34	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P35	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
P36	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
TOTAL	21.898,01	40.106,01	11.132,99	41.640,00	34.756,99	39.795,02	25.535,02	40.677,00	39.361,01

Volume útil inicial dos tanques da carga

Tabela A.23: Volume útil inicial nos tanques de carga R_C em m^3 no cenário seis.

tanques R_C	volume inicial L_{0r}^t	lastro $\underline{L_r^t}$	volume útil inicial
AA	21.898,00	5.000,00	16.898,01
DD	40.106,00	7.000,00	33.106,01
EE	11.133,00	7.100,00	4.032,99
FF	41.640,00	7.100,00	34.540,00
GG	34.757,00	6.900,00	27.856,99
HH	39.795,00	6.400,00	33.395,02
II	25.535,00	8.300,00	17.235,02
KK	40.677,00	6.900,00	33.777,00
LL	39.361,00	6.900,00	32.461,01

MODELOS

Modelagem base

Em todos os modelos as seguintes equações são usadas:

$$\max \sum_{i \in T} \sum_{r \in R_D} \sum_{v \in I_r} \sum_{c \in C} G_C V_{ivc} \quad (\text{B.1})$$

$$S_{iv} \geq \underline{S}_v Z_{iv} \quad i \in T, v \in W_U \quad (\text{B.2})$$

$$E_{iv} \leq H Z_{iv} \quad i \in T, v \in W \quad (\text{B.3})$$

$$E_{iv} = S_{iv} + D_{iv} \quad i \in T, v \in W \quad (\text{B.4})$$

$$D_{iv} \geq \underline{D_{ur}_v} \cdot Z_{iv} \quad i \in T, v \in W \quad (\text{B.5})$$

$$\underline{N_D} \leq \sum_{i \in T} \sum_{v \in W_D} Z_{iv} \leq \overline{N_D} \quad (\text{B.6})$$

$$\underline{N_D} \leq \sum_{v \in W_D} Z_{iv} \leq \overline{N_D} \quad i = i1 \quad (\text{B.7})$$

$$\sum_{i \in T} \sum_{v \in I_r} D_{iv} = H, \quad r \in R_D \quad (\text{B.8})$$

$$\underline{V}_v^t Z_{iv} \leq V_{iv}^t \leq \overline{V}_v^t Z_{iv} \quad i \in T, v \in W \quad (\text{B.9})$$

$$V_{iv}^t = \sum_{c \in C} V_{ivc} \quad i \in T, v \in W \quad (\text{B.10})$$

$$L_{ir}^t = L_{0r}^t + \sum_{j \in T, j < i} \sum_{v \in I_r} V_{jv}^t - \sum_{j \in T, j < i} \sum_{v \in O_r} V_{jv}^t \quad i \in T, r \in R - R_M \quad (\text{B.11})$$

$$L_{irc} = L_{0rc} + \sum_{j \in T, j < i} \sum_{v \in I_r} V_{jvc} - \sum_{j \in T, j < i} \sum_{v \in O_r} V_{jvc}, \quad i \in T, r \in R - R_M, c \in C \quad (\text{B.12})$$

$$L_{ir}^t = \sum_{c \in C} L_{irc} \quad i \in T, r \in R - R_M \quad (\text{B.13})$$

$$\sum_{v \in I_r} V_{iv}^t = \sum_{v \in O_r} V_{iv}^t \quad i \in T, r \in R_M \quad (\text{B.14})$$

$$\sum_{v \in I_r} V_{ivc} = \sum_{v \in O_r} V_{ivc} \quad i \in T, c \in C, r \in R_M \quad (\text{B.15})$$

$$\sum_{v \in I_r} Z_{iv} \leq 2 \quad i \in T, r \in R_M \quad (\text{B.16})$$

$$S_{iv} \leq S_{iv_1} + H \cdot (1 - Z_{iv}) \quad i \in T, r \in R_M, v \in I_r, v_1 \in O_r \quad (\text{B.17})$$

$$S_{iv} \geq S_{iv_1} - H \cdot (1 - Z_{iv}) \quad i \in T, r \in R_M, v \in I_r, v_1 \in O_r \quad (\text{B.18})$$

$$E_{iv} \leq E_{iv_1} + H \cdot (1 - Z_{iv}) \quad i \in T, r \in R_M, v \in I_r, v_1 \in O_r \quad (\text{B.19})$$

$$E_{iv} \geq E_{iv_1} - H \cdot (1 - Z_{iv}) \quad i \in T, r \in R_M, v \in I_r, v_1 \in O_r \quad (\text{B.20})$$

$$\underline{FR}_r \lambda_{iv} \leq \sum_{v \in O_r} V_{iv}^t \leq \overline{FR}_r \lambda_{iv} \quad i \in T, r \in R_{C2} \quad (\text{B.21})$$

$$\underline{FR}_v D_{iv} \leq V_{iv}^t \leq \overline{FR}_v D_{iv} \quad i \in T, r \in R_C - R_{C2}, v \in O_r \quad (\text{B.22})$$

$$S_{iv} \leq S_{iv_1} + H \cdot (2 - Z_{iv} - Z_{iv_1}) \quad i \in T, r \in R_C, v \in O_r, v_1 \in O_r \quad (\text{B.23})$$

$$S_{iv} \geq S_{iv_1} - H \cdot (2 - Z_{iv} - Z_{iv_1}) \quad i \in T, r \in R_C, v \in O_r, v_1 \in O_r \quad (\text{B.24})$$

$$D_{iv} \leq \lambda_{ir} + H \cdot (2 - Z_{iv} - Z_{iv_1}) \quad i \in T, r \in R_C, v \in O_r, v_1 \in O_r \quad (\text{B.25})$$

$$D_{iv} \geq \lambda_{ir} - H \cdot (2 - Z_{iv} - Z_{iv_1}) \quad i \in T, r \in R_C, v \in O_r, v_1 \in O_r \quad (\text{B.26})$$

$$\lambda_{ir} \leq H \cdot \sum_{v \in O_r} Z_{iv} \quad i \in T, r \in R_{C2} \quad (\text{B.27})$$

$$\underline{x}_{vk} V_{iv}^t \leq \sum_{c \in C} x_{ck} V_{ivc} \leq \overline{x}_{vk} V_{iv}^t \quad i \in T, v \in W, k \in K_v \quad (\text{B.28})$$

$$\underline{x}_{vk} \sum_{c \in C} \rho_c V_{ivc} \leq \sum_{c \in C} x_{ck} \rho_c V_{ivc} \leq \overline{x}_{vk} \sum_{c \in C} \rho_c V_{ivc} \quad i \in T, v \in W, k \in K_m \quad (\text{B.29})$$

$$V_{ivc} L_{ir}^t = L_{irc} V_{iv}^t \quad i \in T, r \in R - R_M, v \in O_r, c \in C \quad (\text{B.30})$$

$$\underline{L}_r^t \leq L_{0r}^t + \sum_{i \in T} \sum_{v \in I_r} V_{iv}^t - \sum_{i \in T} \sum_{v \in O_r} V_{iv}^t \leq \overline{L}_r^t, \quad r \in R_P \cup R_C \quad (\text{B.31})$$

$$0 \leq L_{0rc} + \sum_{i \in T} \sum_{v \in I_r} V_{ivc} - \sum_{i \in T} \sum_{v \in O_r} V_{ivc} \leq \overline{L}_r^t, \quad i \in T, r \in R_P \cup R_C, c \in C \quad (\text{B.32})$$

$$\sum_{v \in W'} Z_{iv} \leq 1 \quad i \in T, W' \in \text{clique}(G_{NO}) \cup \text{clique}(G_{NOTR}) \quad (\text{B.33})$$

$$Z_{iv} \leq \sum_{v' \in W} Z_{(i-1)v'} \quad i \in T, i \neq 1, v \in W \quad NO_{vv'} = 1 \quad (\text{B.34})$$

$$\sum_{v \in W'} E_{i_1 v} + \sum_{i \in T, i_1 < i_2} \sum_{v \in W'} D_{iv} \leq \sum_{v \in W'} S_{i_2 v} + H \cdot \left(1 - \sum_{v \in W'} Z_{i_2 v}\right) \quad (\text{B.35})$$

$i_1, i_2 \in T, i_1 < i_2, W' \in \text{clique}(G_{NO})$

$$\sum_{v \in W} Z_{iv} \geq 1 \quad i \in T \quad (\text{B.36})$$

Modelagem com descarregamentos não-sobrepostos

O modelo com descarregamentos não-sobrepostos é composto pelas equações apresentadas no Apêndice B.1 mais a seguinte equação:

$$\begin{aligned} \sum_{v \in W'} E_{i_1 v} + \sum_{i \in T, i_1 < i < i_2} \sum_{v \in W'} D_{iv} + \sum_{v \in W'} \sum_{w \in W'} TR_{v,w} \cdot Z_{i_1 v} \leq \sum_{v \in W'} S_{i_2 v} + \\ (H + \frac{1}{|R_P|} \sum_{v \in W'} \sum_{w \in W'} TR_{v,w}) \cdot (1 - \sum_{v \in W'} Z_{i_2 v}) \\ i_1, i_2 \in T, i_1 < i_2, (W_1; W_2) \in \text{clique}(G_{NOTR}) \end{aligned} \quad (\text{B.37})$$

Modelagem com descarregamentos sobrepostos

O modelo com descarregamentos sobrepostos é composto pelas equações apresentadas no Apêndice B.1 mais as seguintes equações:

$$\begin{aligned} E_{i_1 v_1} + E_{i_1 v_2} + \sum_{i \in T, i_1 < i < i_2} (D_{i v_1} + D_{i v_2}) + TR_{v_1 v_2} \cdot Z_{i_1 v_1} + TR_{v_2 v_1} \cdot Z_{i_1 v_2} \leq \\ S_{i_2 v_1} + S_{i_2 v_2} + (H + TR_{v_1 v_2} + TR_{v_2 v_1}) \cdot (1 - Z_{i_2 v_1} - Z_{i_2 v_2}) \\ i_1, i_2 \in T, i_1 < i_2, v_1 \in W', v_2 \in W'', W', W'' \in \text{clique}(G_{NOTR}) \end{aligned} \quad (\text{B.38})$$

$$\sum_{v \in W_U} Z_{iv} \leq 1 \quad i \in T, r \in R_C, v \in I_r \quad (\text{B.39})$$

$$\begin{aligned} \sum_{v \in I_r} E_{i_1 v} + \sum_{i \in T, i_1 < i < i_2} \sum_{v \in I_r} D_{iv} + \sum_{i \in T, i_1 < i < i_2} \lambda_{ir} \leq \sum_{v \in I_r} S_{i_2 v} + H \cdot (1 - \sum_{v \in I_r} Z_{i_2 v}) \\ i_1, i_2 \in T, i_1 < i_2, r \in R_{C2} \end{aligned} \quad (\text{B.40})$$

$$\begin{aligned} \sum_{v \in I_r} E_{i_1 v} + \sum_{i \in T, i_1 < i < i_2} \sum_{v \in I_r} D_{iv} + \sum_{i \in T, i_1 < i < i_2} \sum_{v \in O_r} D_{iv} \leq \sum_{v \in I_r} S_{i_2 v} + H \cdot (1 - \sum_{v \in I_r} Z_{i_2 v}) \\ i_1, i_2 \in T, i_1 < i_2, r \in R_C - R_{C2} \end{aligned} \quad (\text{B.41})$$

RESULTADOS DA PROGRAMAÇÃO

A seguir os resultados do melhor programa para a bateria de testes Bat-1. As especificações das bateria podem ser encontradas na Tabela ??.

Modelagem com descarregamentos não-sobrepostos

Cenário dois

O melhor programa para o cenário dois com descarregamentos não-sobrepostos na bateria de testes Bat-1 foi obtido com $n = 6$.

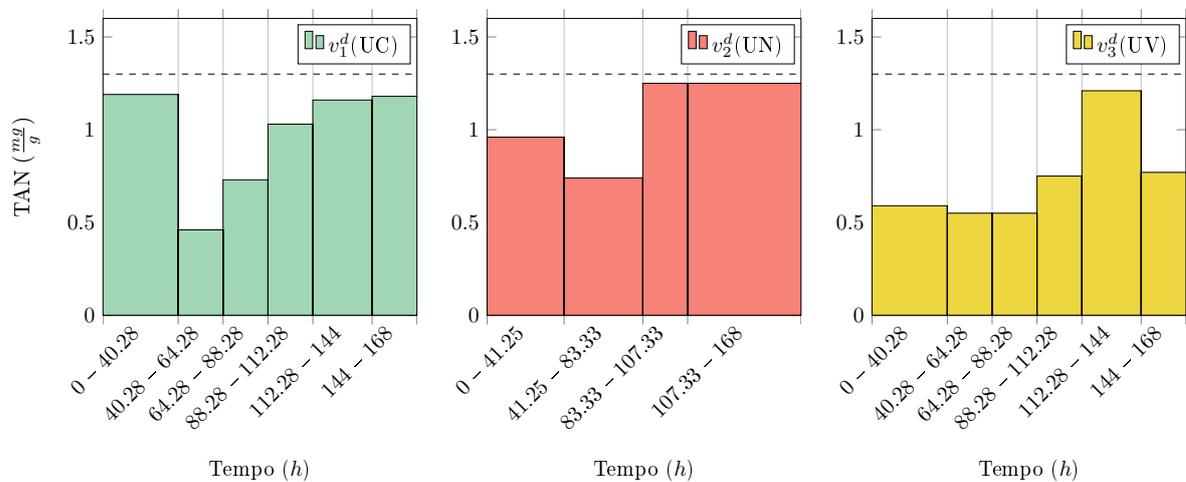


Figura C.1: Perfil de acidez nas operações de destilação W_D no cenário dois com descarregamentos não-sobrepostos.

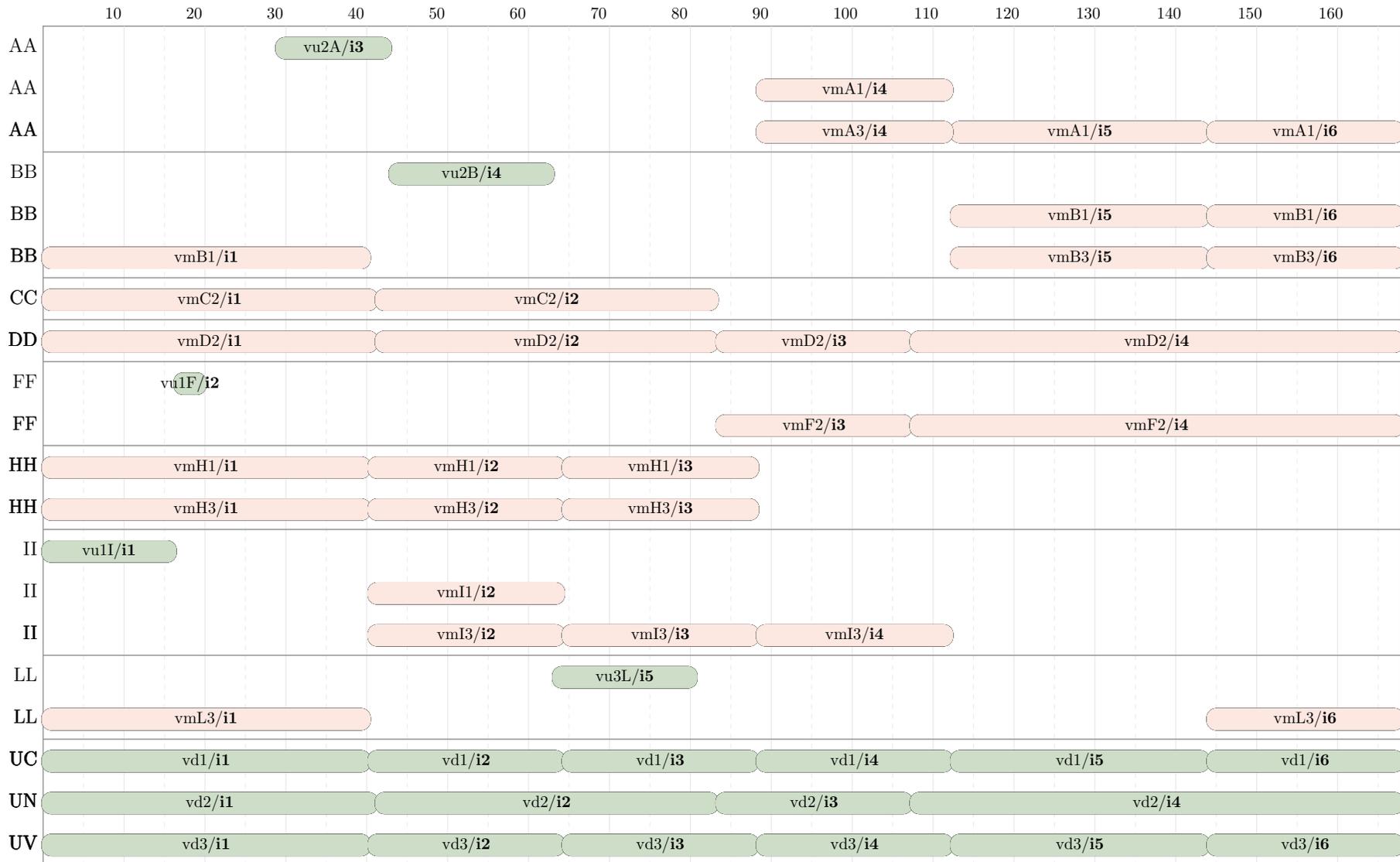


Figura C.2: Gráfico de Gantt para o cenário dois com descarregamentos não-sobrepostos ($n = 6$).

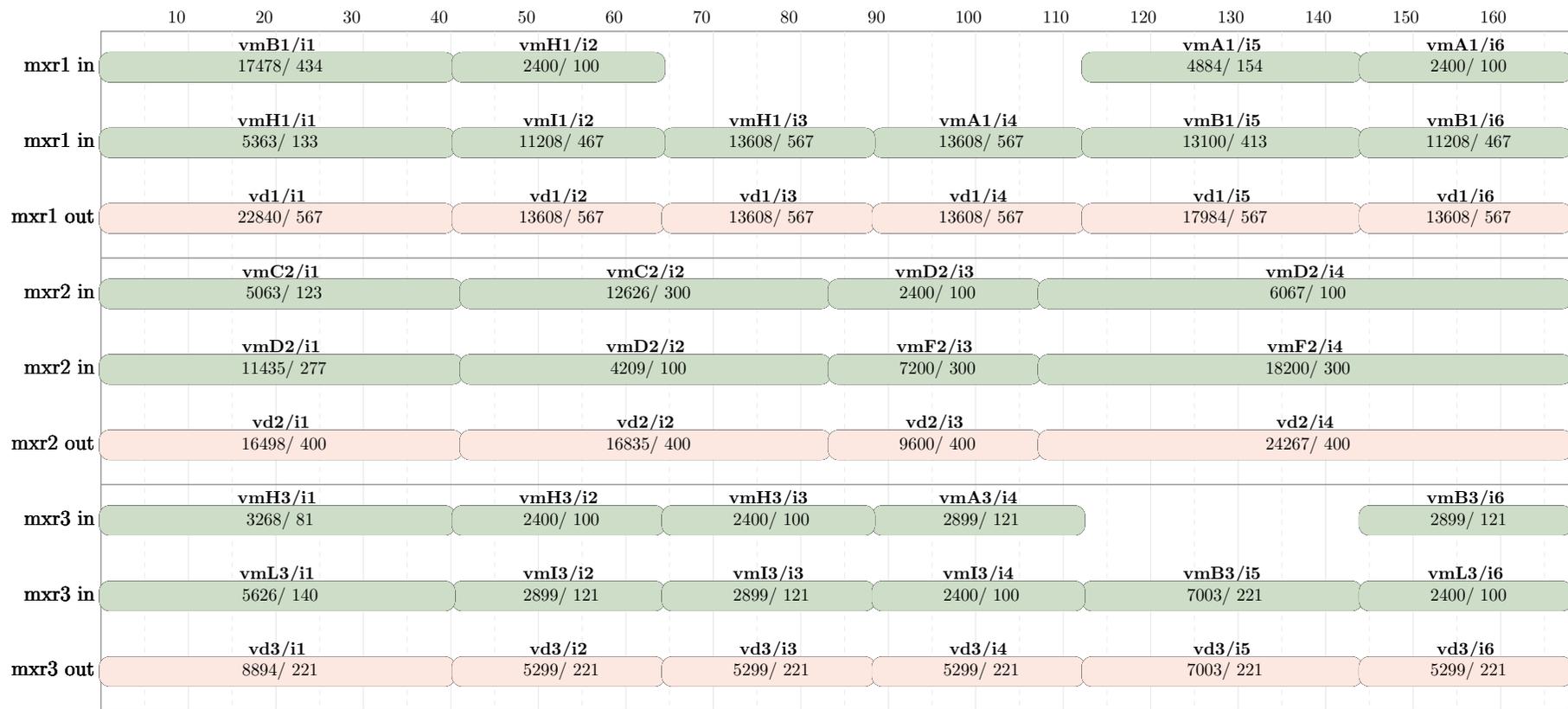


Figura C.3: Programação nos misturadores com informações de volume e vazão para o cenário dois com descarregamentos não-sobrepostos ($n = 6$).

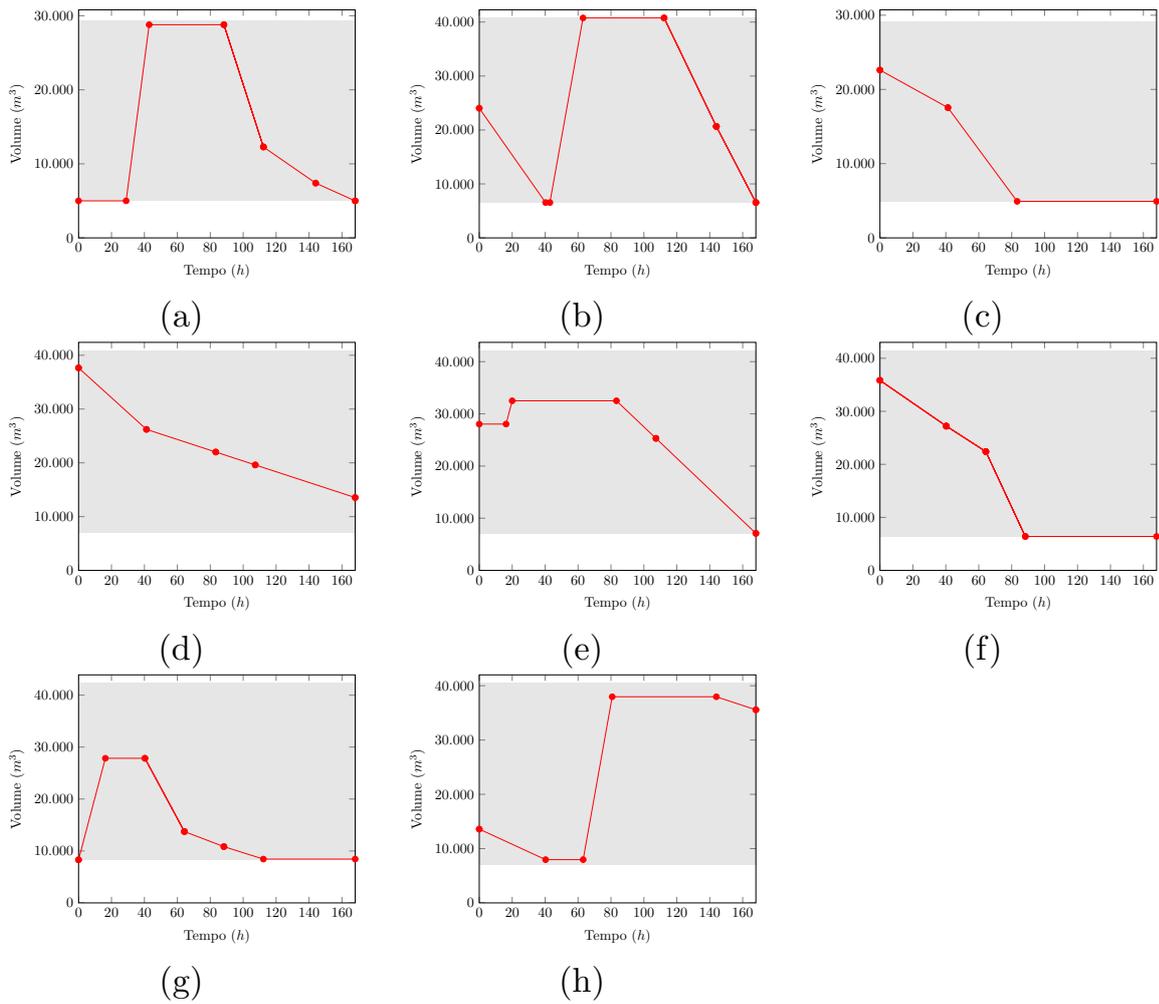


Figura C.4: Inventário nos tanques de carga (a) AA, (b) BB, (c) CC, (d) DD, (e) FF, (f) HH, (g) II, e (h) LL no cenário dois com descarregamentos não-sobrepostos.

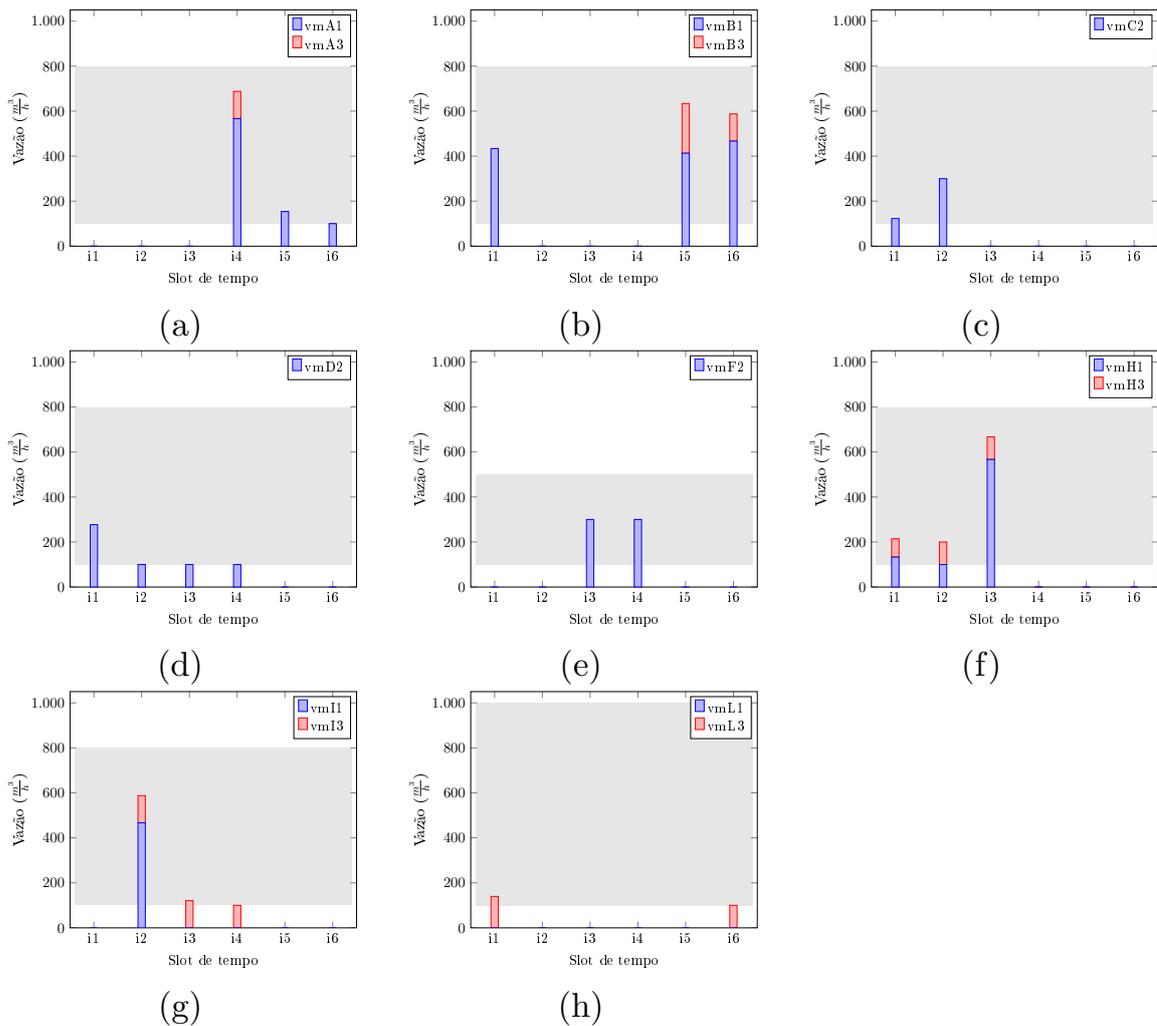


Figura C.5: Vazão de saída dos tanques de carga (a) AA, (b) BB, (c) CC, (d) DD, (e) FF, (f) HH, (g) II, e (h) LL no cenário dois com descarregamentos não-sobrepostos.

Cenário três

O melhor programa para o cenário três com descarregamentos não-sobrepostos na bateria de testes Bat-1 foi obtido com $n = 8$.

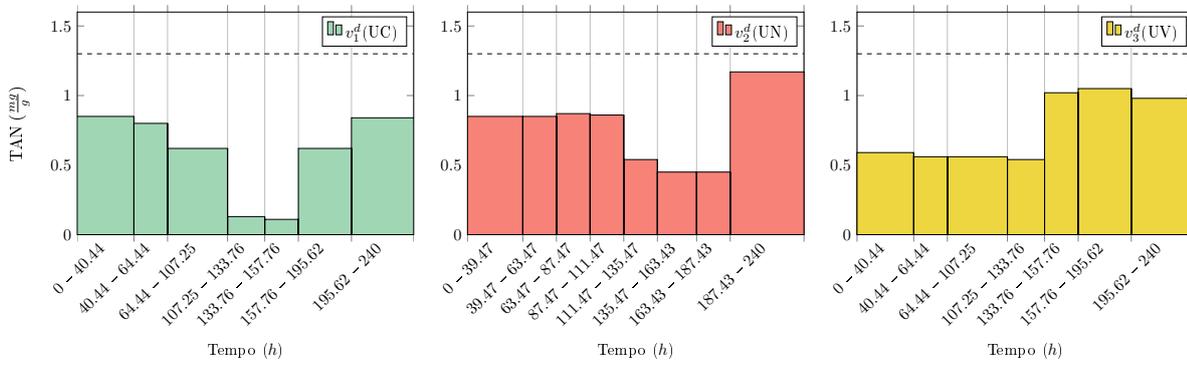


Figura C.6: Perfil de acidez nas operações de destilação W_D no cenário três com descarregamentos não-sobrepostos.

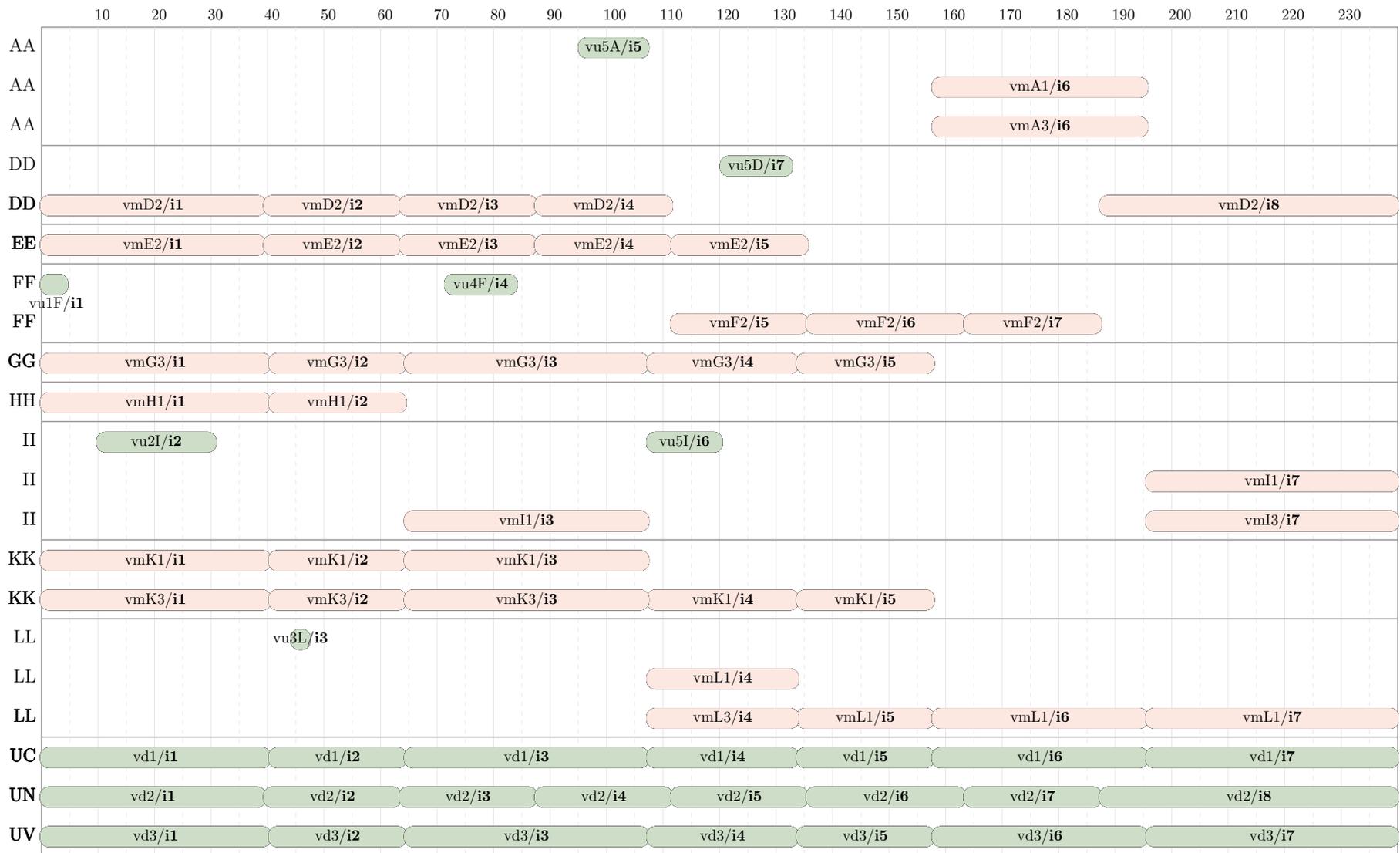


Figura C.7: Gráfico de Gantt para o cenário três com descarregamentos não-sobrepostos ($n = 8$).

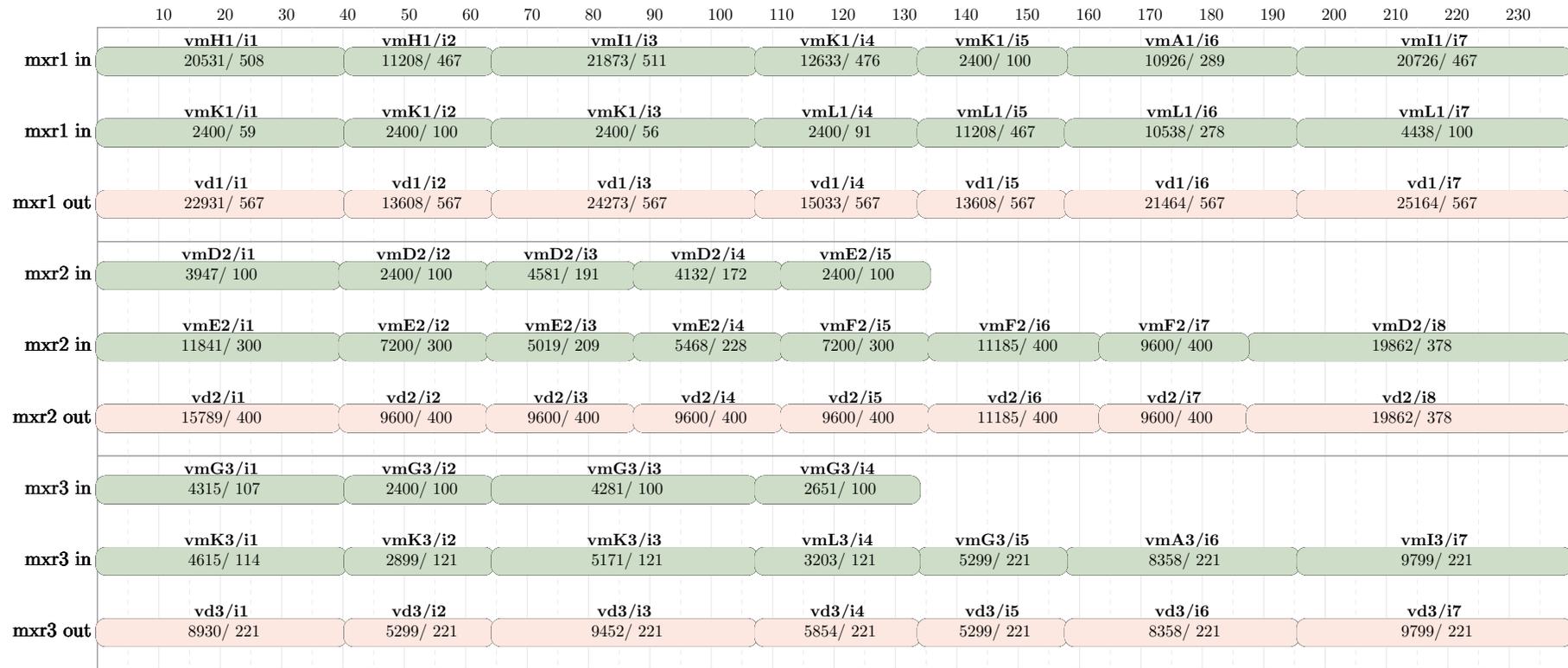


Figura C.8: Programação nos misturadores com informações de volume e vazão para o cenário três com descarregamentos não-sobrepostos ($n = 8$).

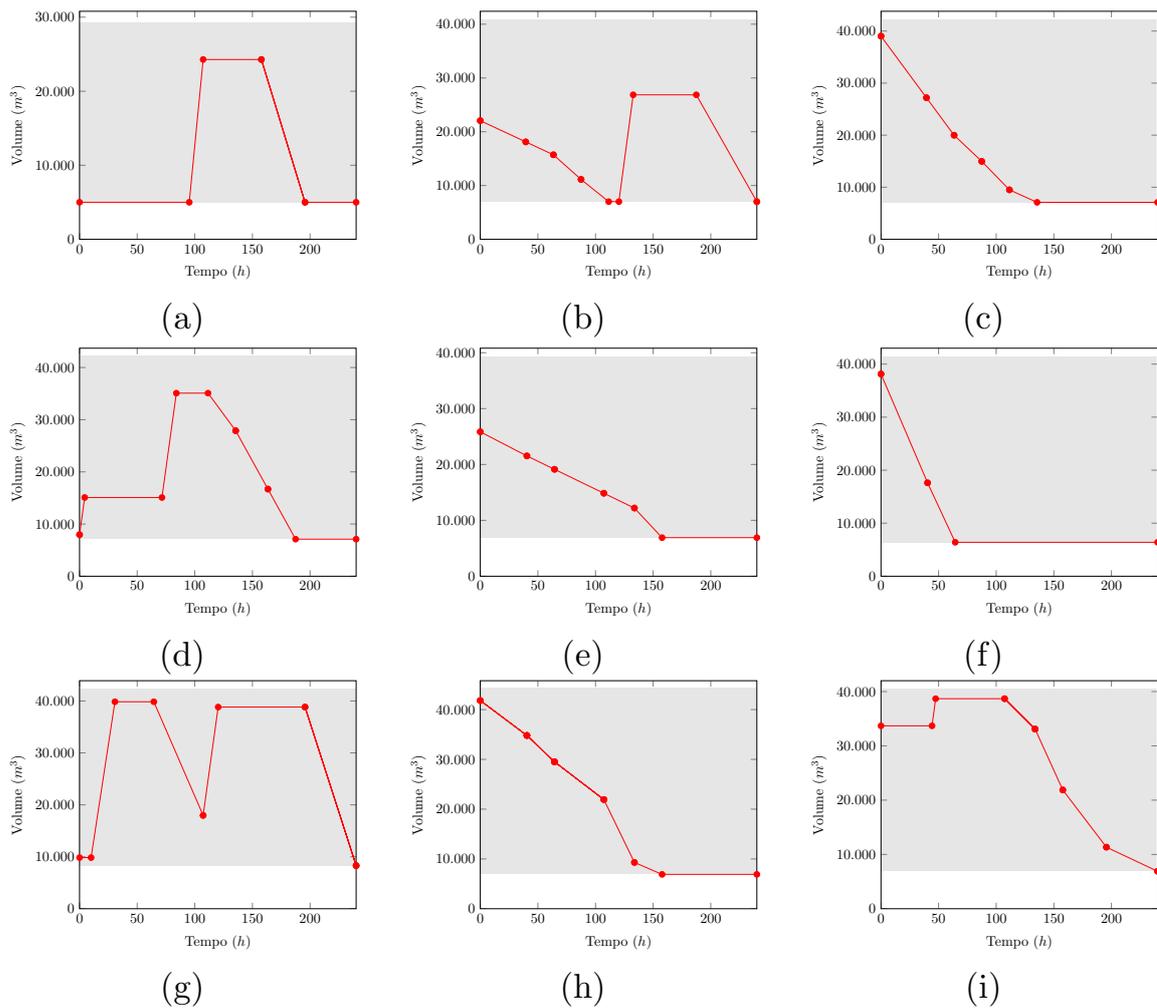


Figura C.9: Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário três com descarregamentos não-sobrepostos.

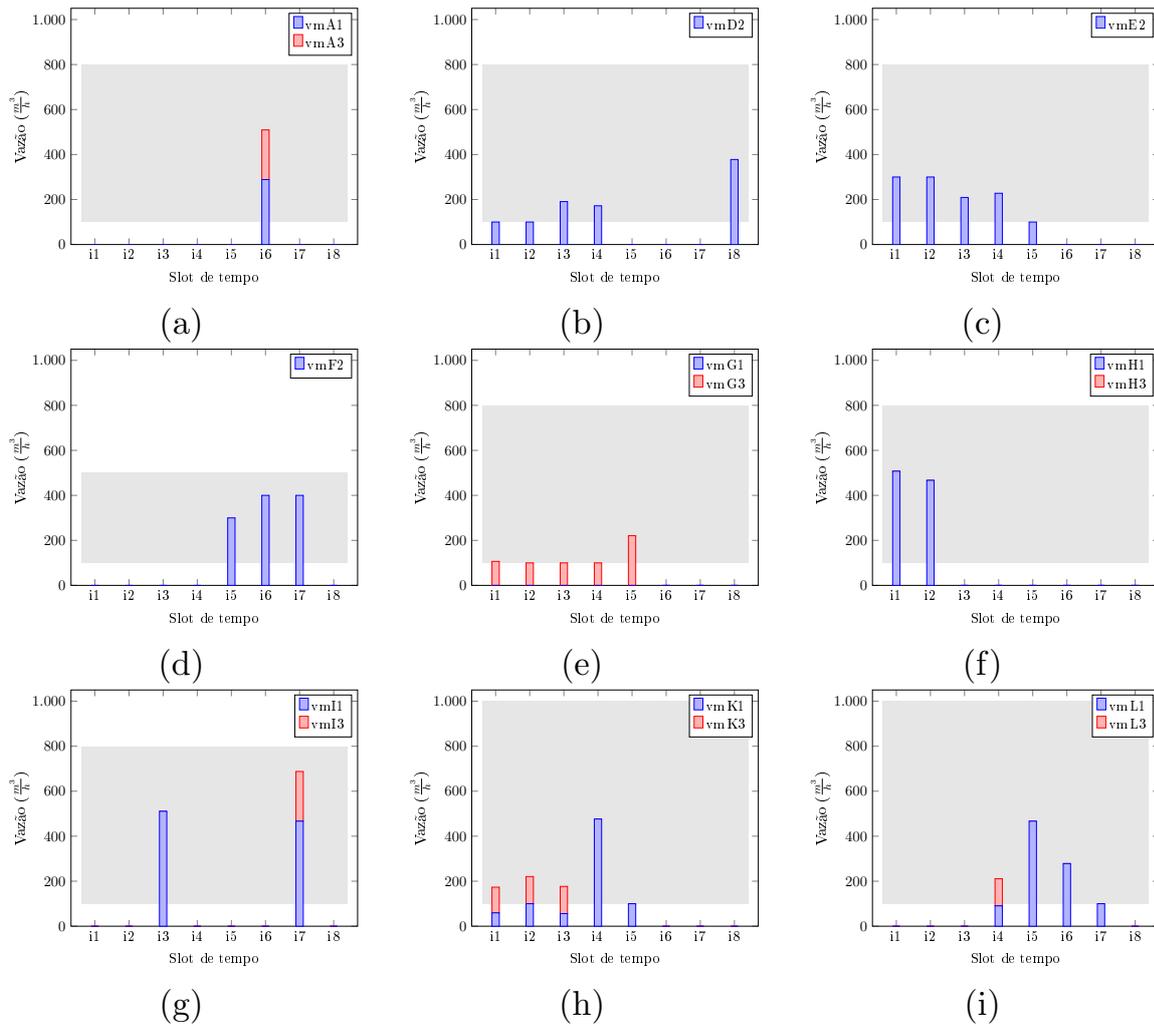


Figura C.10: Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário três com descarregamentos não-sobrepostos.

Cenário quatro

O melhor programa para o cenário quatro com descarregamentos não-sobrepostos na bateria de testes Bat-1 foi obtido com $n = 7$.

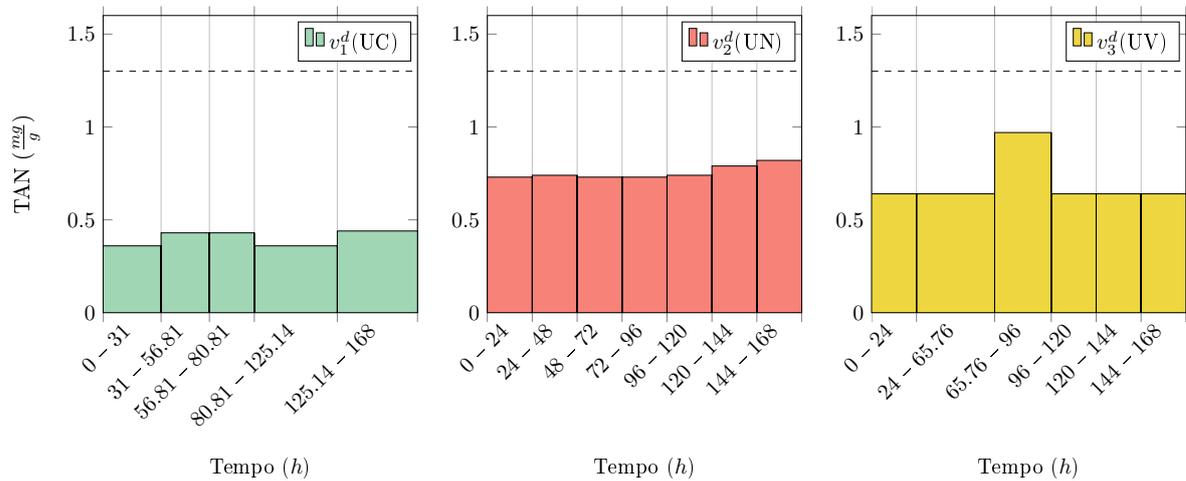


Figura C.11: Perfil de acidez nas operações de destilação W_D no cenário quatro com descarregamentos não-sobrepostos.

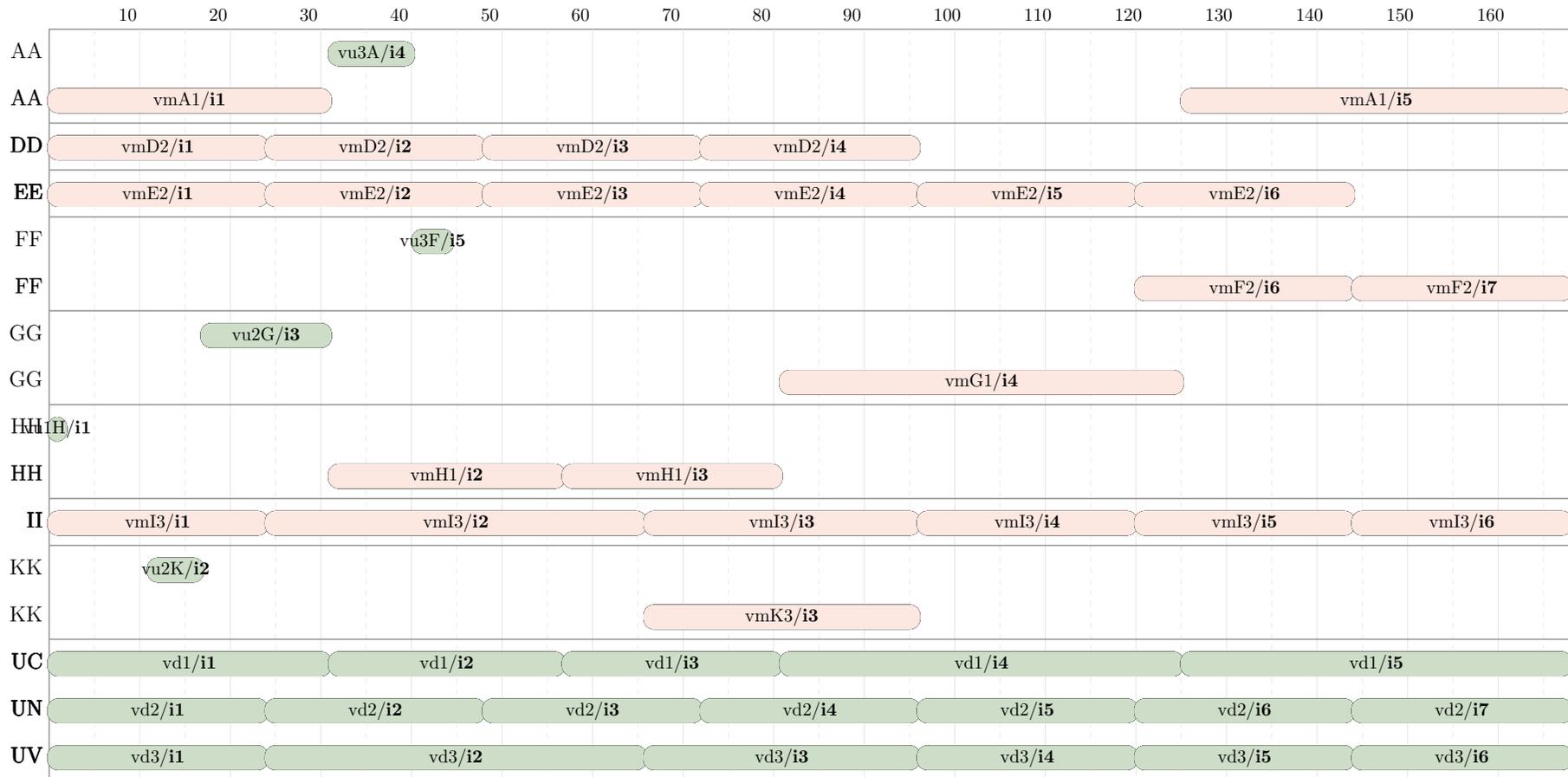


Figura C.12: Gráfico de Gantt para o cenário quatro com descarregamentos não-sobrepostos ($n = 7$).

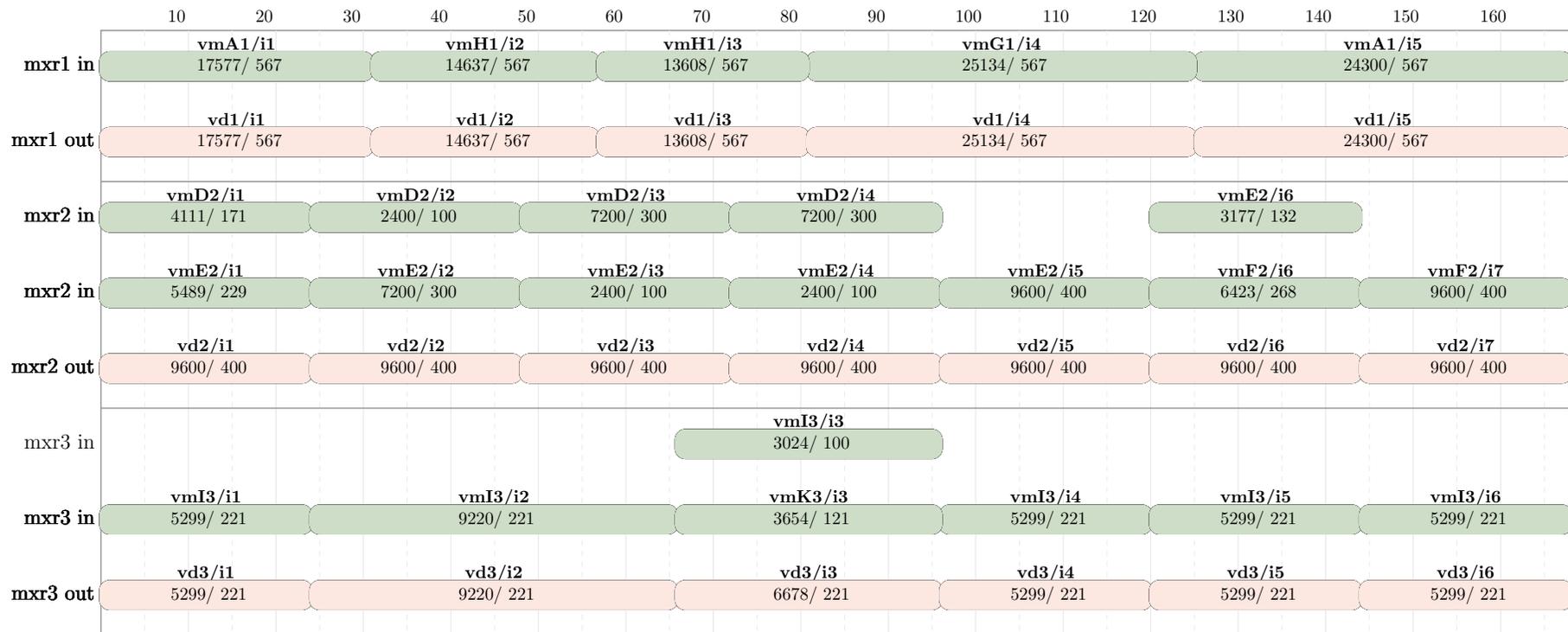


Figura C.13: Programação nos misturadores com informações de volume e vazão para o cenário quatro com descarregamentos não-sobrepostos ($n = 7$).

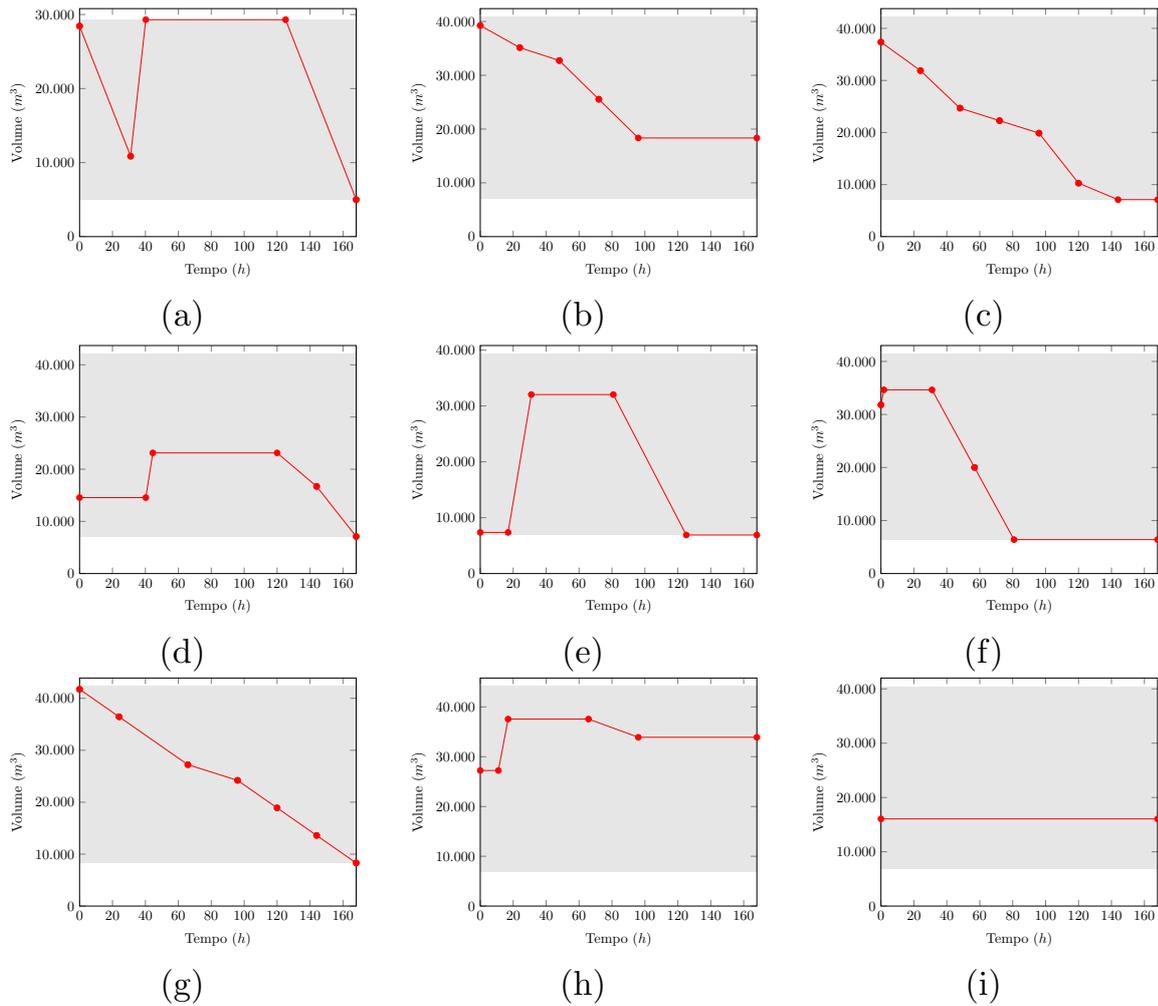


Figura C.14: Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário quatro com descarregamentos não-sobrepostos.

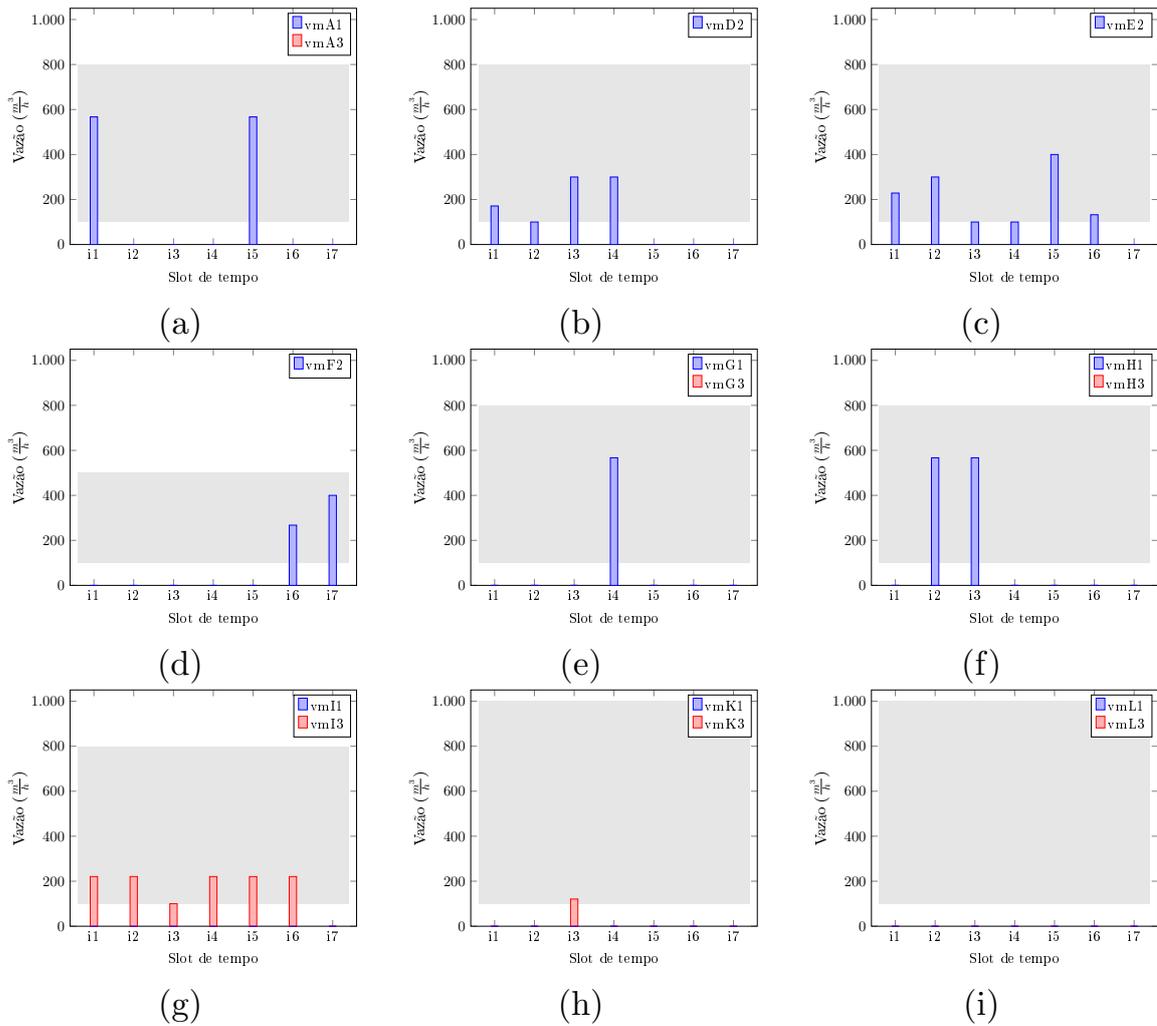


Figura C.15: Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário quatro com descarregamentos não-sobrepostos.

Cenário cinco

O melhor programa para o cenário cinco com descarregamentos não-sobrepostos na bateria de testes Bat-1 foi obtido com $n = 6$.

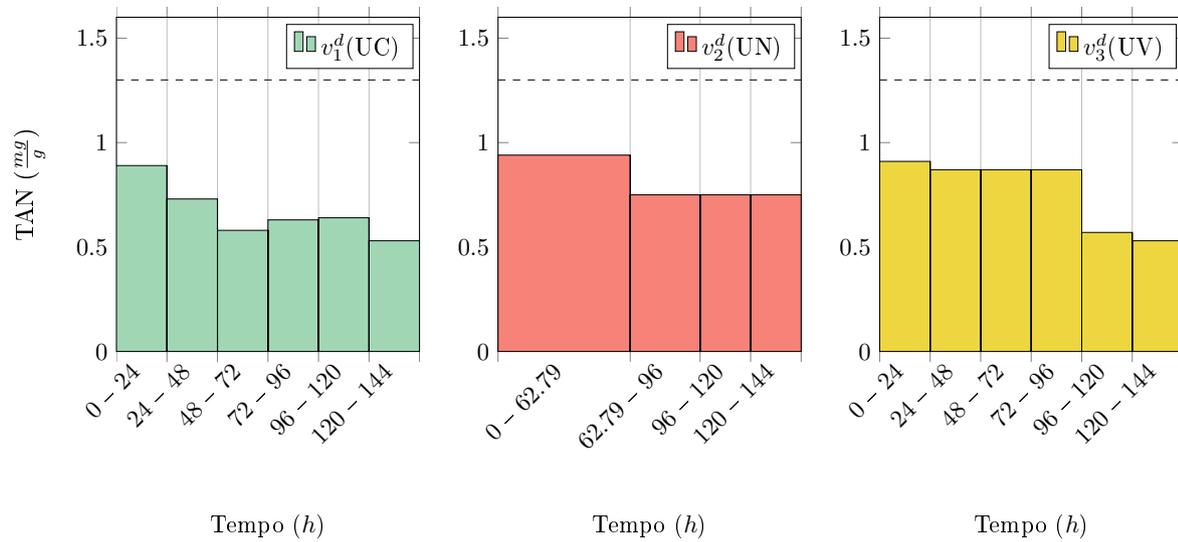


Figura C.16: Perfil de acidez nas operações de destilação W_D no cenário cinco com descarregamentos não-sobrepostos.

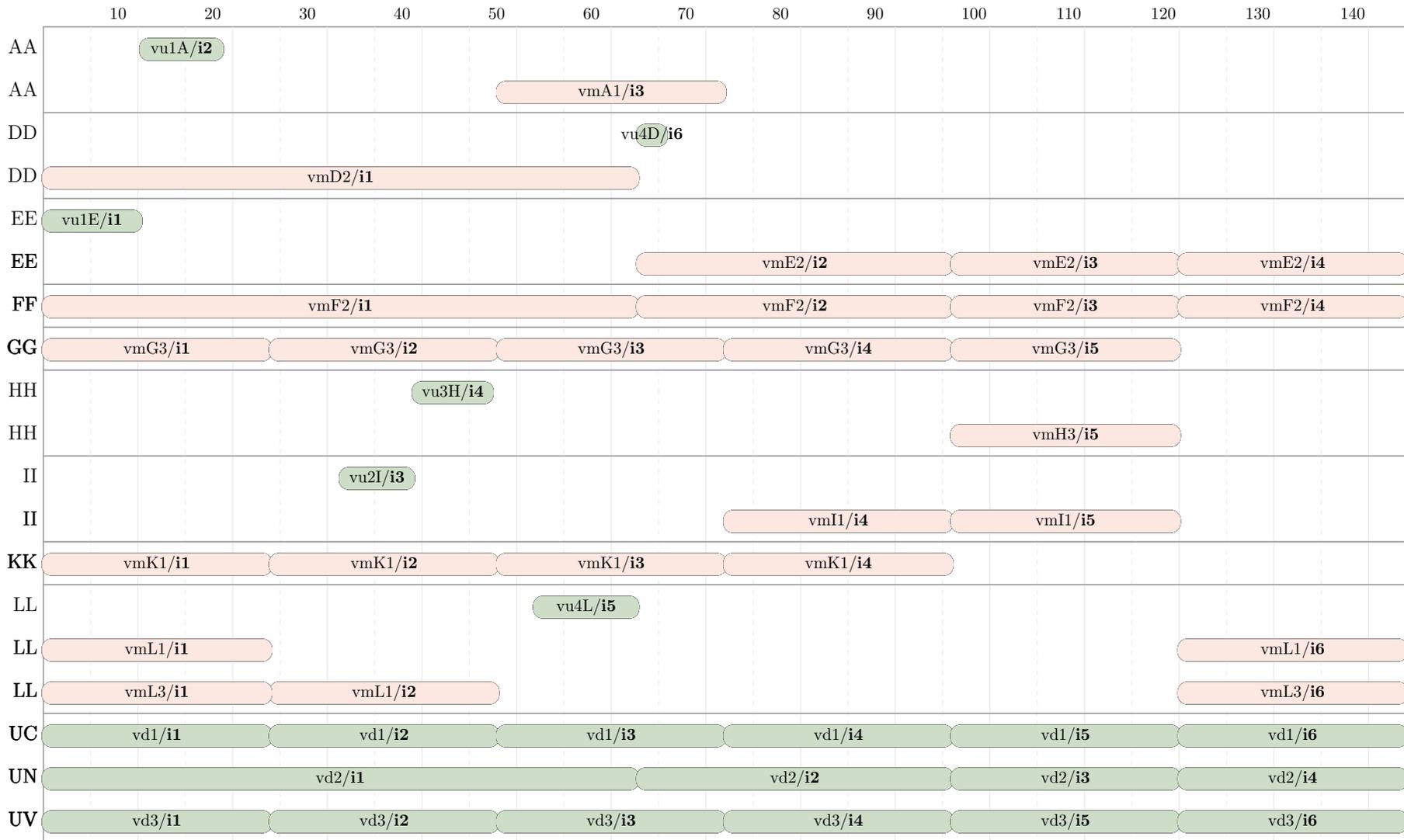


Figura C.17: Gráfico de Gantt para o cenário cinco com descarregamentos não-sobrepostos ($n = 6$).

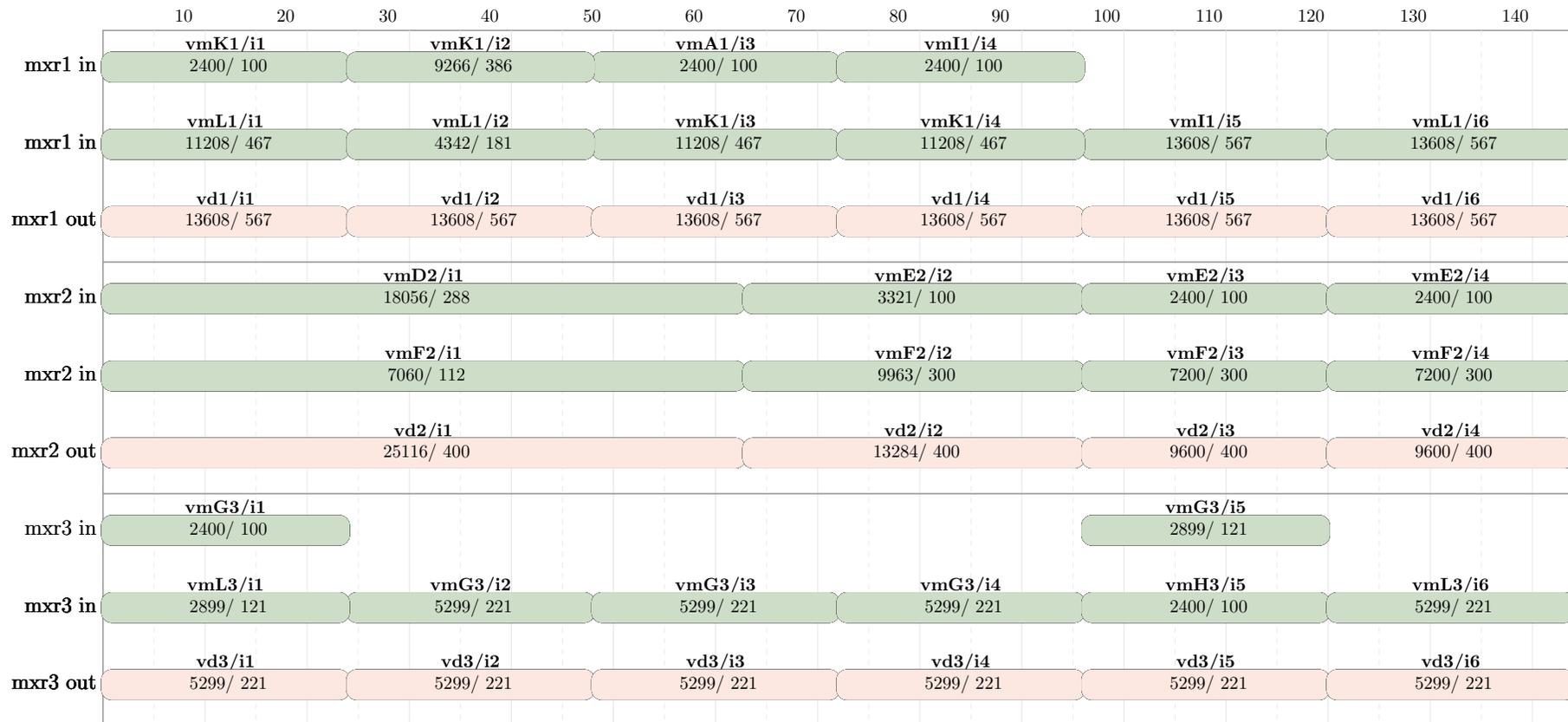


Figura C.18: Programação nos misturadores com informações de volume e vazão para o cenário cinco com descarregamentos não-sobrepostos ($n = 6$).

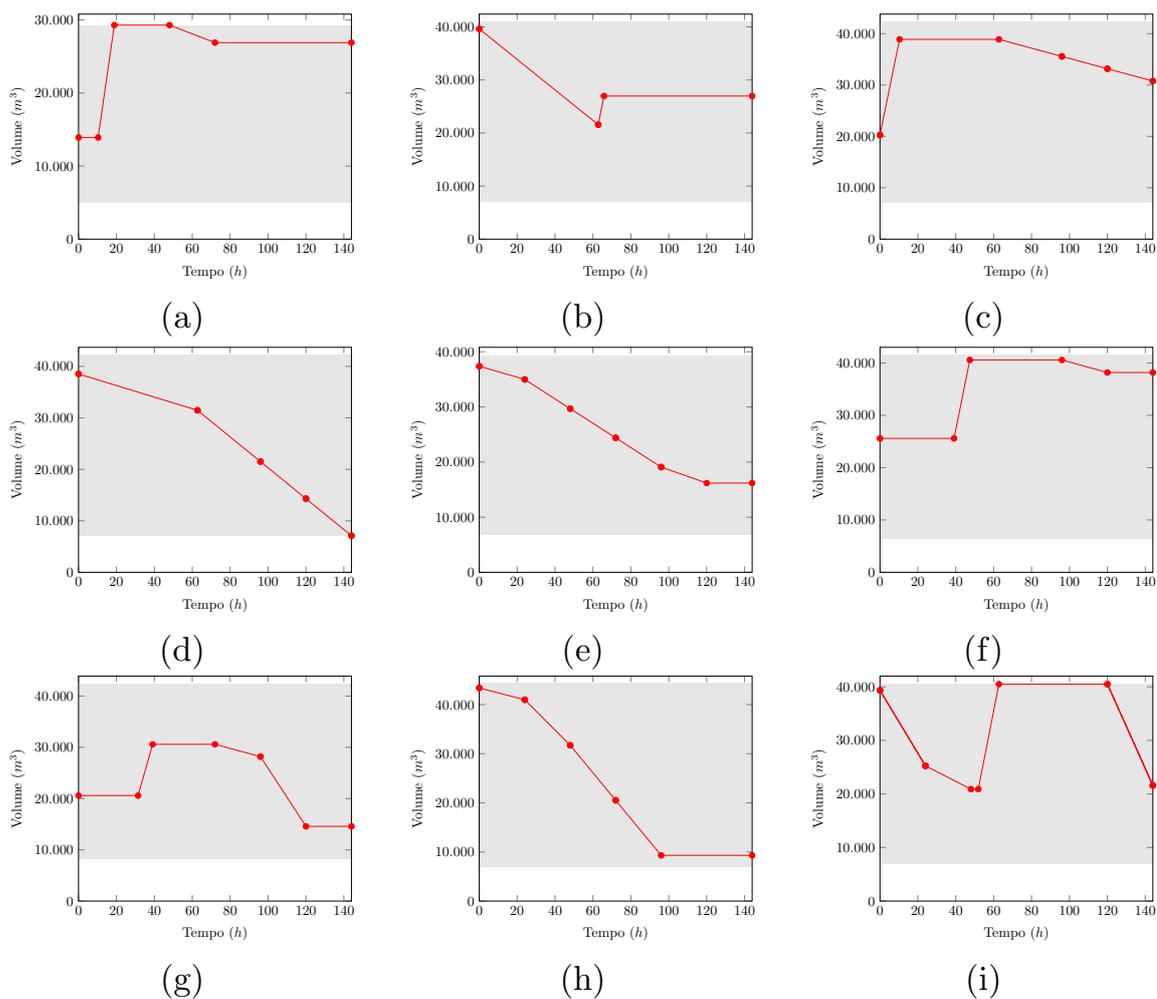


Figura C.19: Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário cinco com descarregamentos não-sobrepostos.

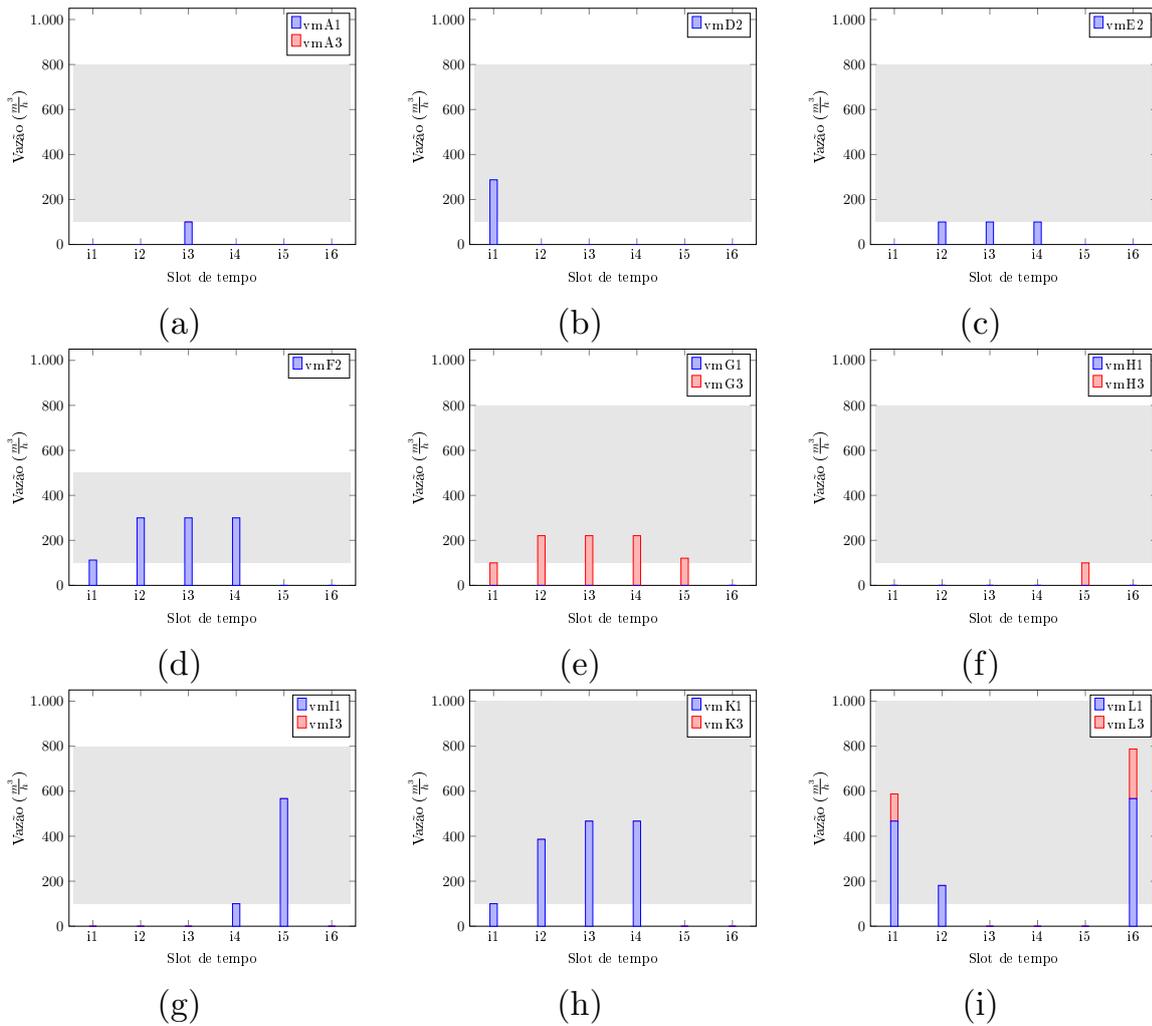


Figura C.20: Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário cinco com descarregamentos não-sobrepostos.

Cenário seis

O melhor programa para o cenário seis com descarregamentos não-sobrepostos na bateria de testes Bat-1 foi obtido com $n = 5$.

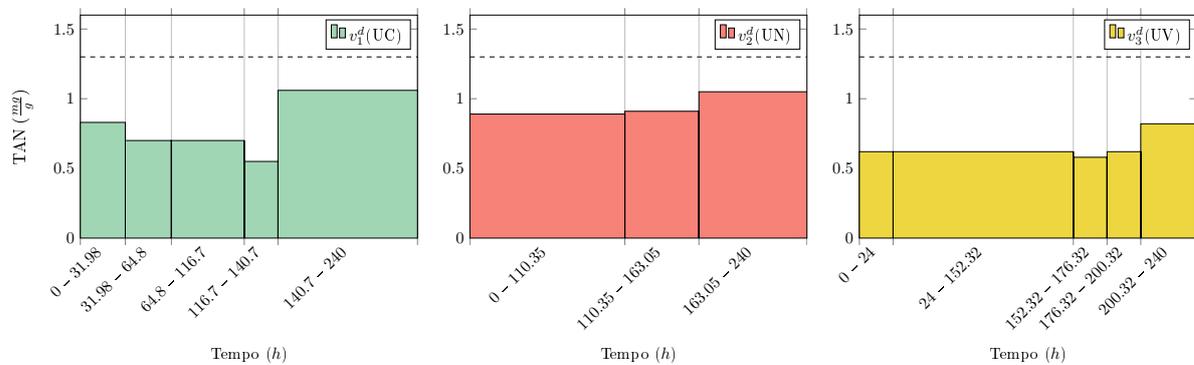


Figura C.21: Perfil de acidez nas operações de destilação W_D no cenário seis com descarregamentos não-sobrepostos.

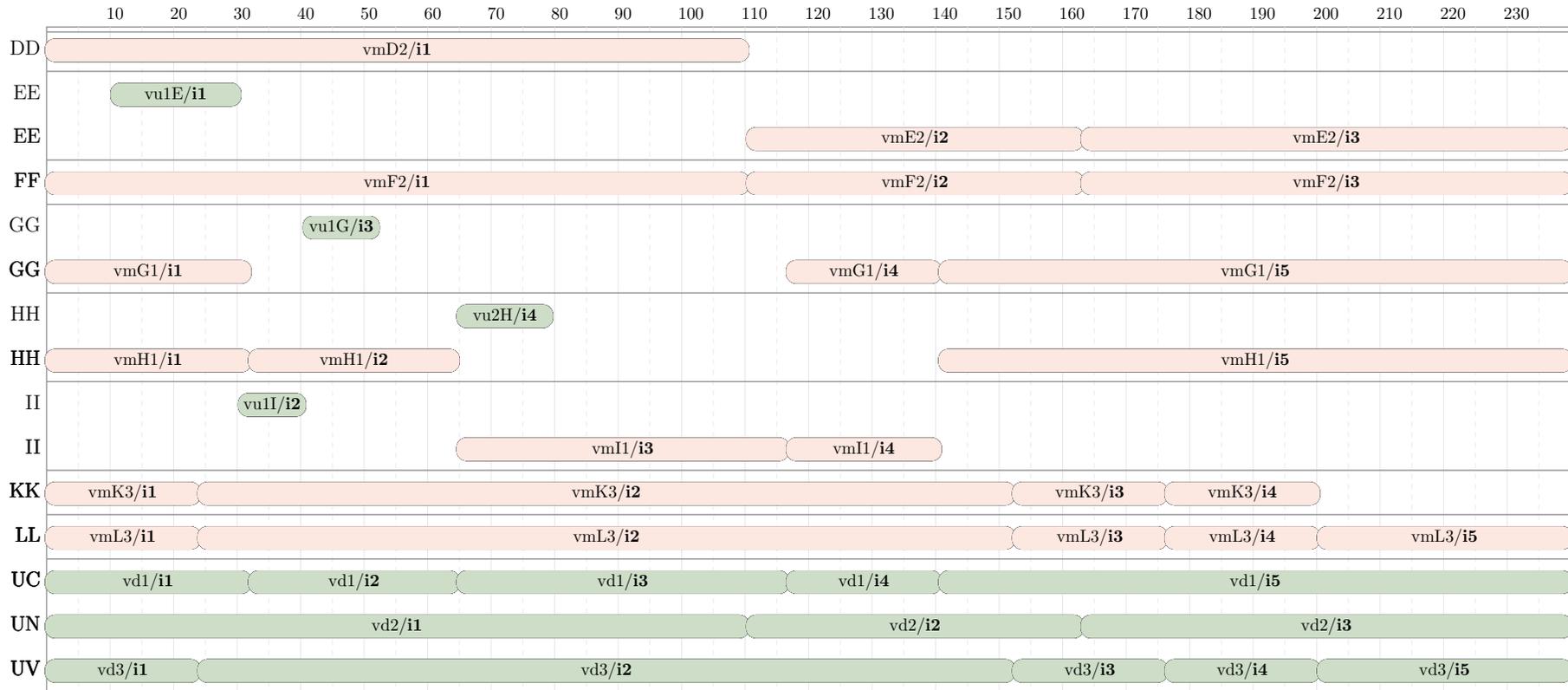


Figura C.22: Gráfico de Gantt para o cenário seis com descarregamentos não-sobrepostos ($n = 5$).

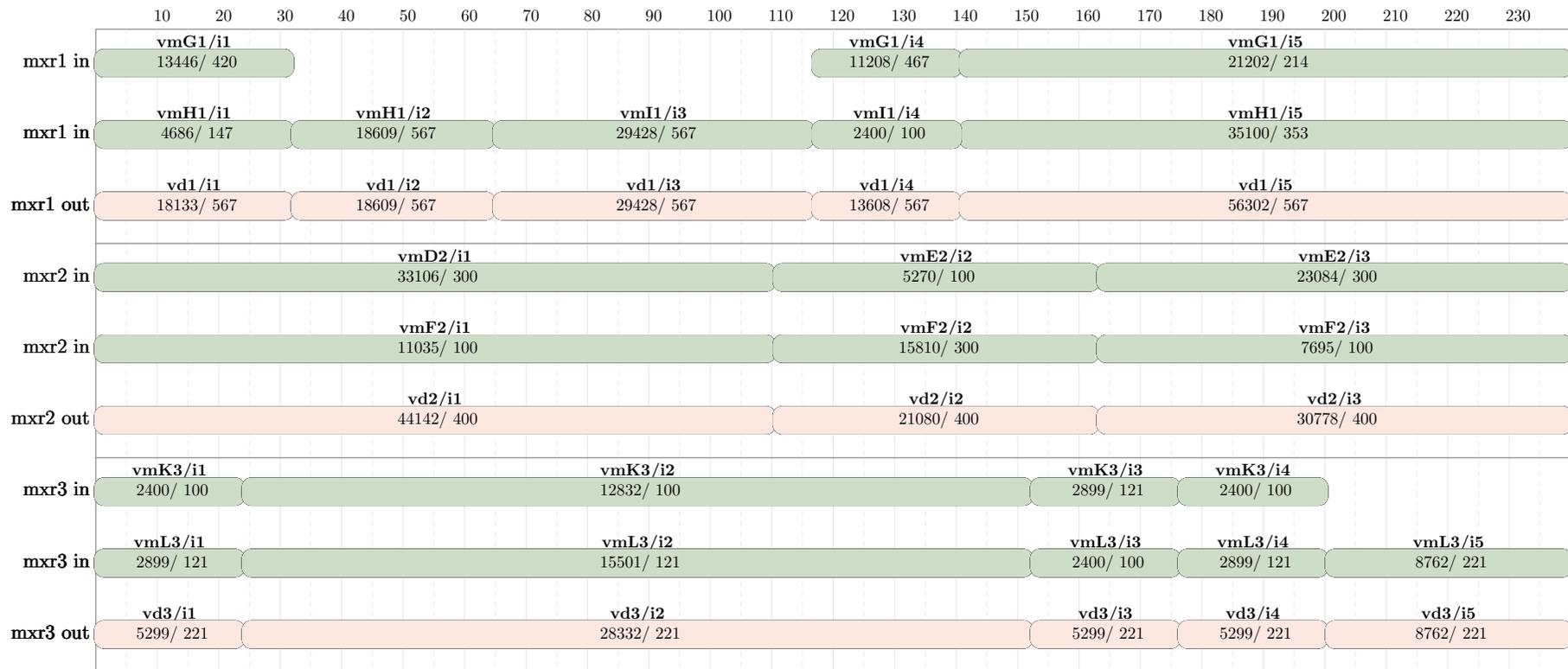


Figura C.23: Programação nos misturadores com informações de volume e vazão para o cenário seis com descarregamentos não-sobrepostos ($n = 5$).

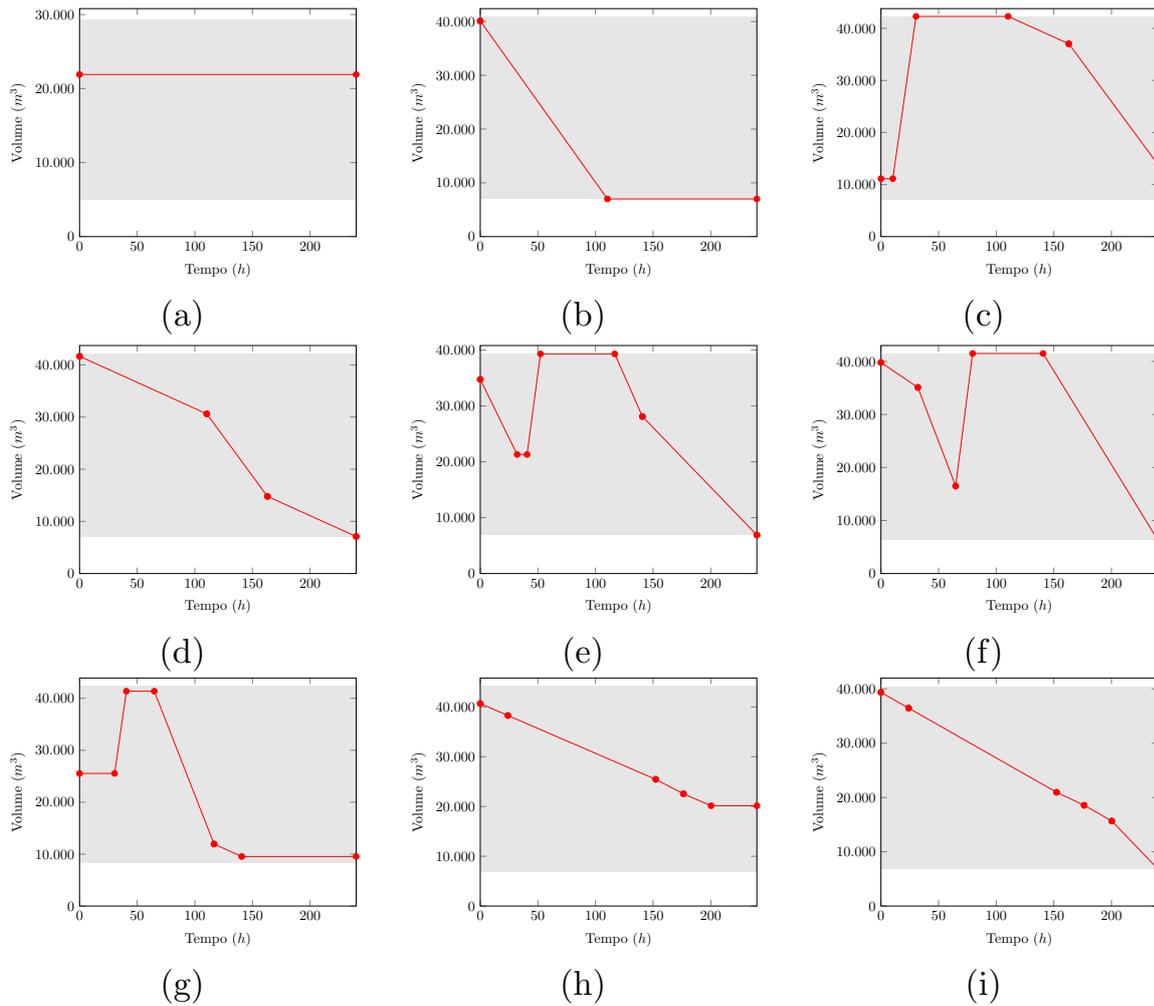


Figura C.24: Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário seis com descarregamentos não-sobrepostos.

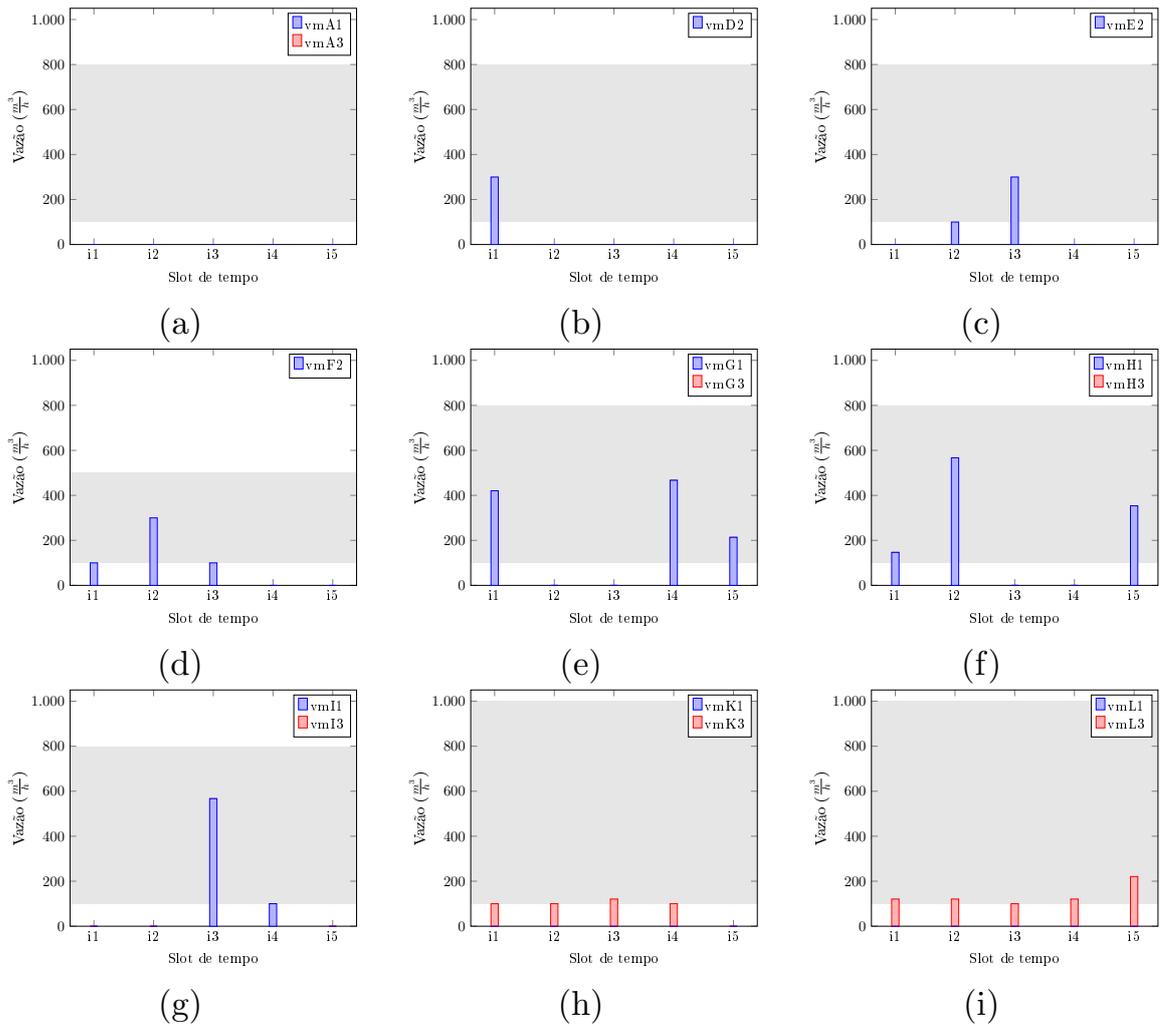


Figura C.25: Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário seis com descarregamentos não-sobrepostos.

Modelagem com descarregamentos sobrepostos

Cenário dois

O melhor programa para o cenário dois com descarregamentos sobrepostos na bateria de testes Bat-1 foi obtido com $n = 3$.

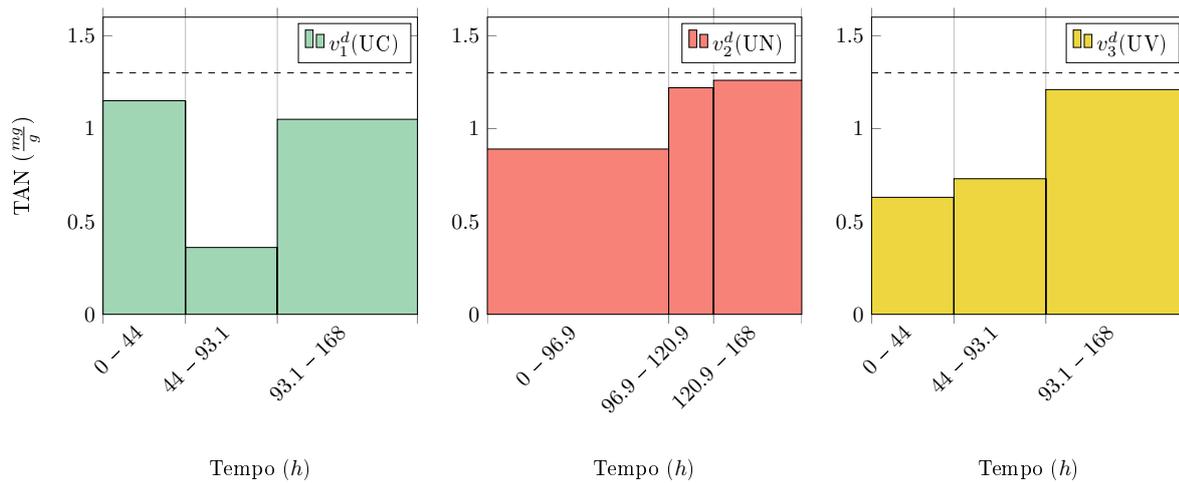


Figura C.26: Perfil de acidez nas operações de destilação W_D no cenário dois com descarregamentos sobrepostos.

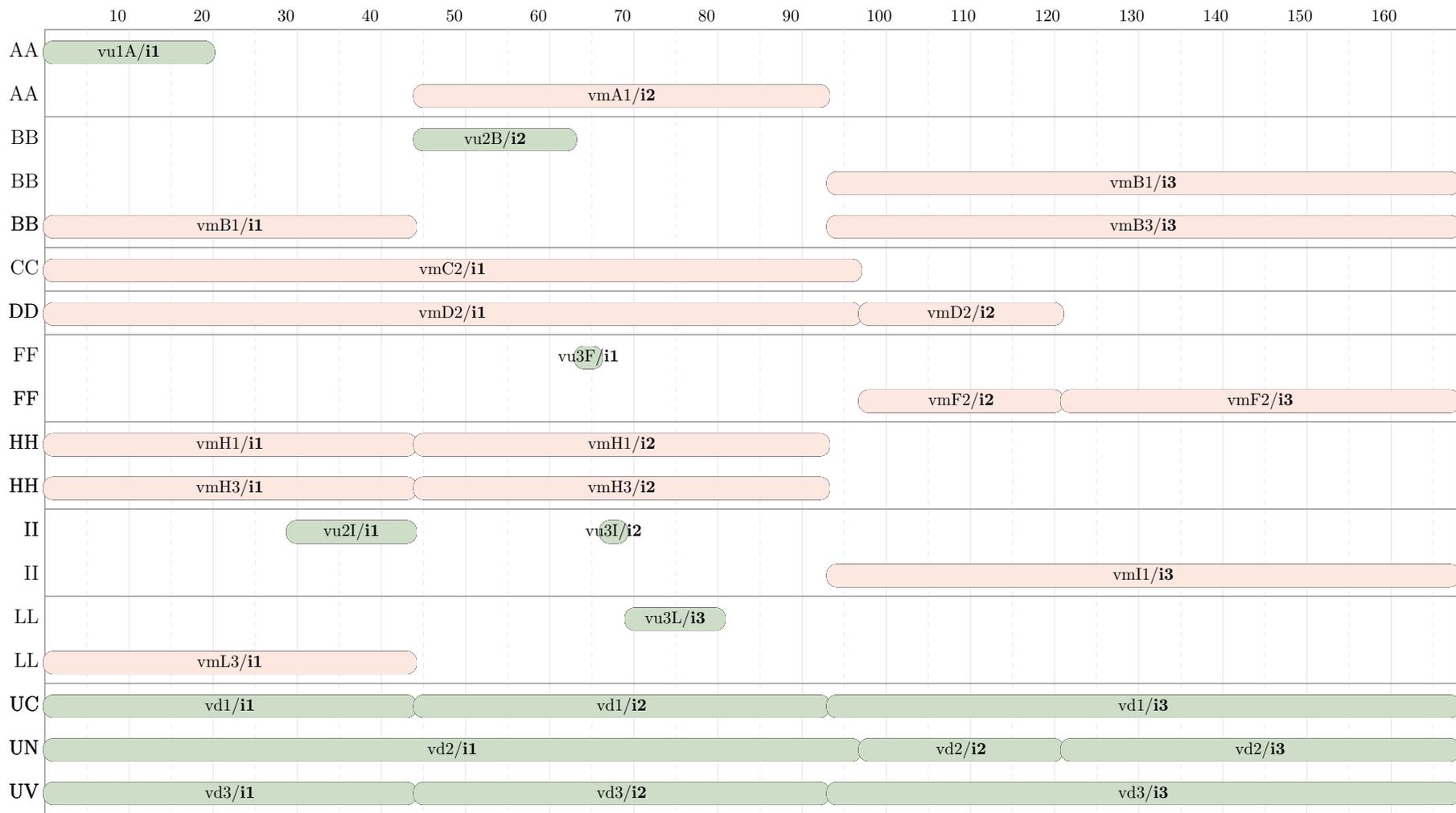


Figura C.27: Gráfico de Gantt para o cenário dois com descarregamentos sobrepostos ($n = 3$).

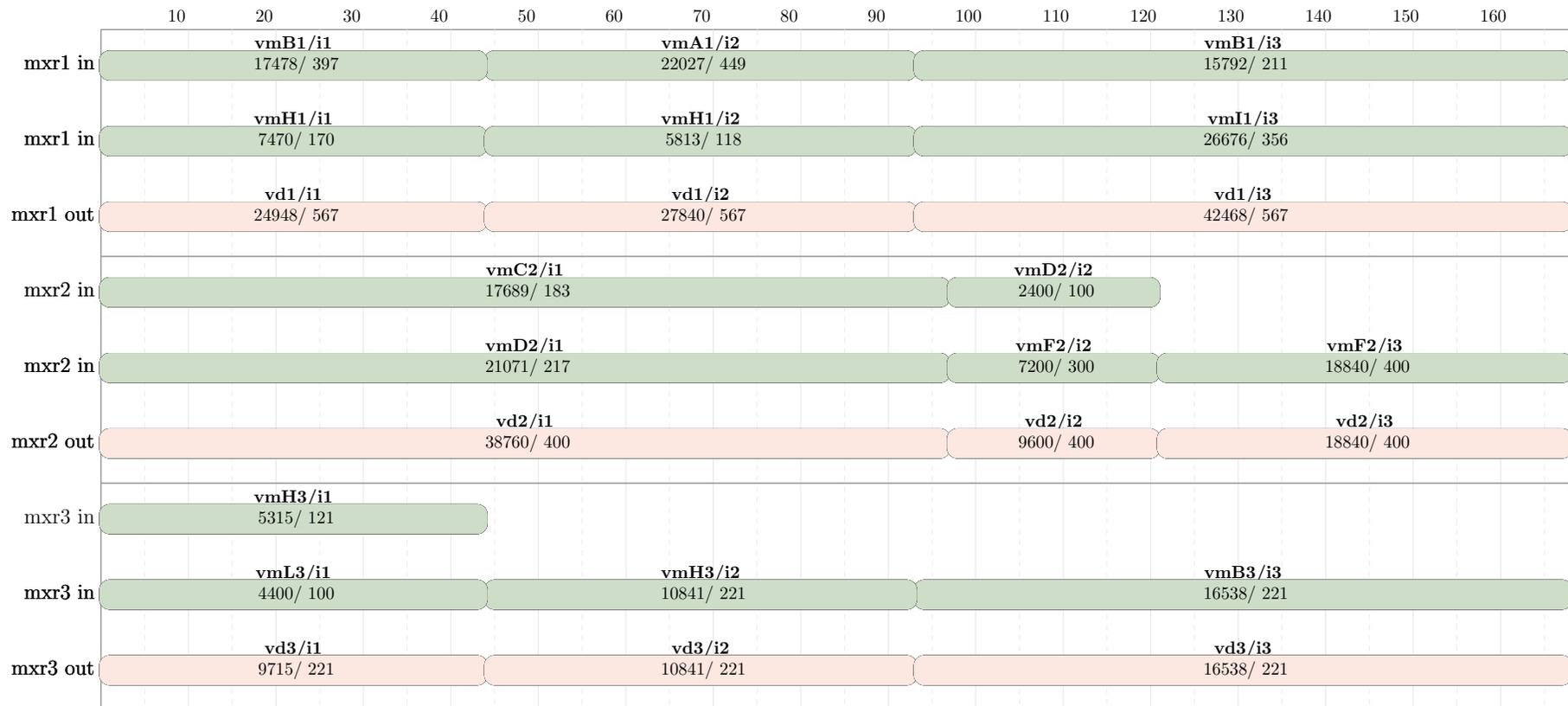


Figura C.28: Programação nos misturadores com informações de volume e vazão para o cenário dois com descarregamentos sobrepostos ($n = 3$).

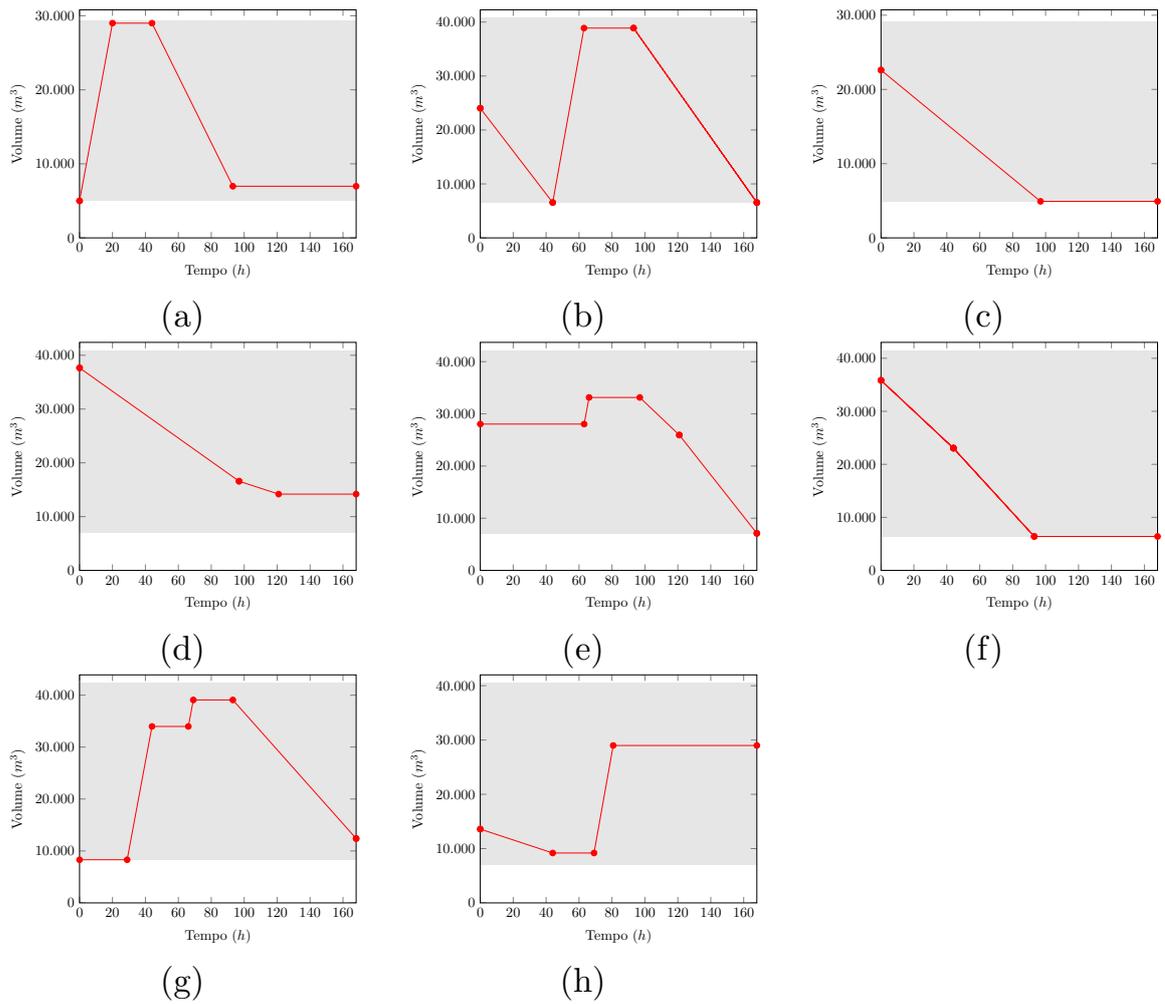


Figura C.29: Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário dois com descarregamentos sobrepostos.

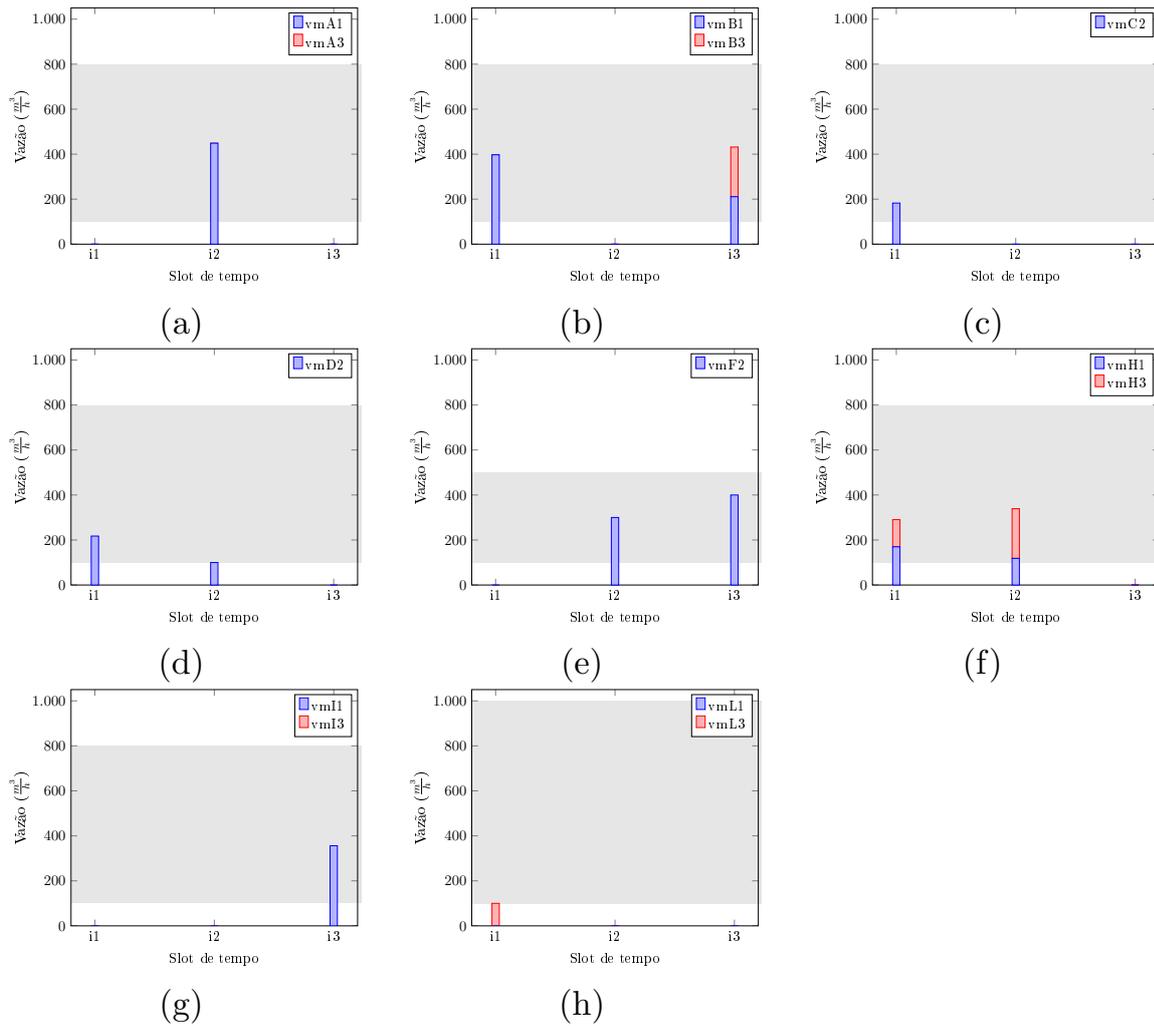


Figura C.30: Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário dois com descarregamentos sobrepostos.

Cenário três

O melhor programa para o cenário três com descarregamentos sobrepostos na bateria de testes Bat-1 foi obtido com $n = 6$.

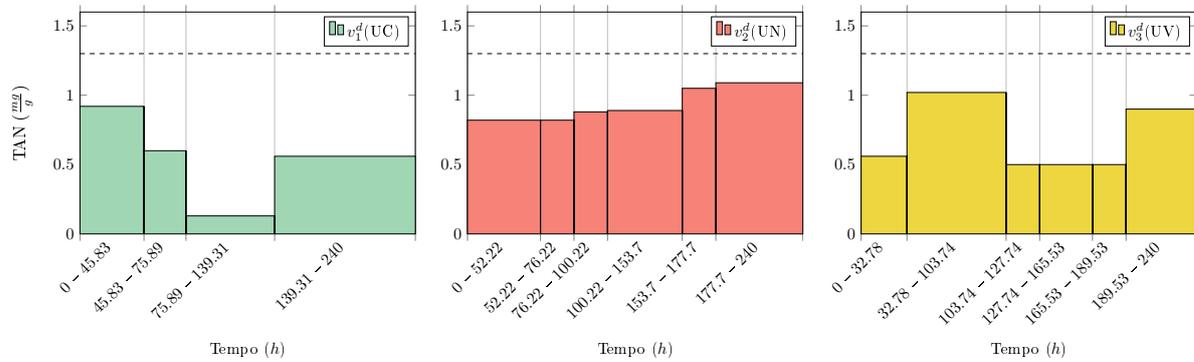


Figura C.31: Perfil de acidez nas operações de destilação W_D no cenário três com descarregamentos sobrepostos.

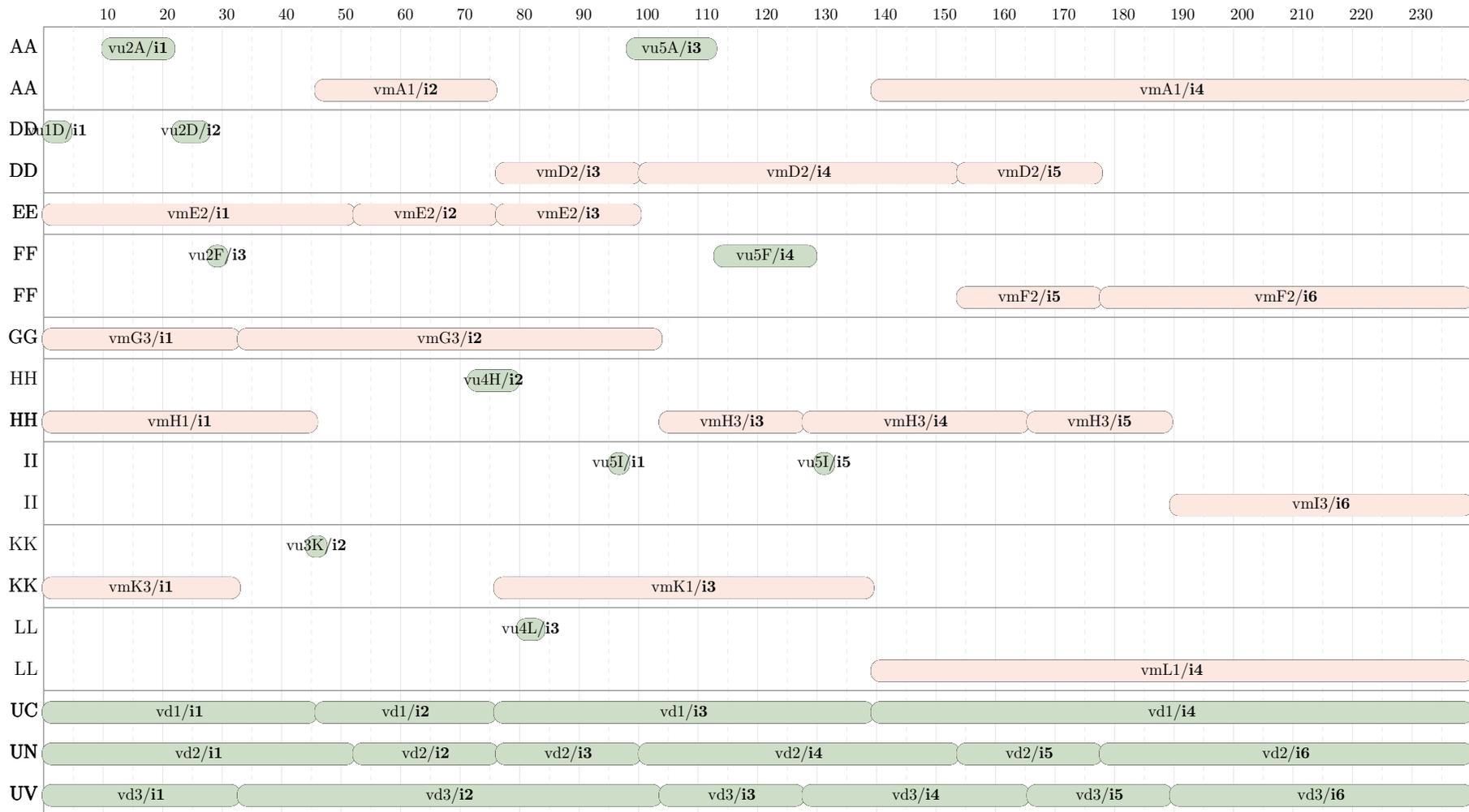


Figura C.32: Gráfico de Gantt para o cenário três com descarregamentos sobrepostos ($n = 6$).

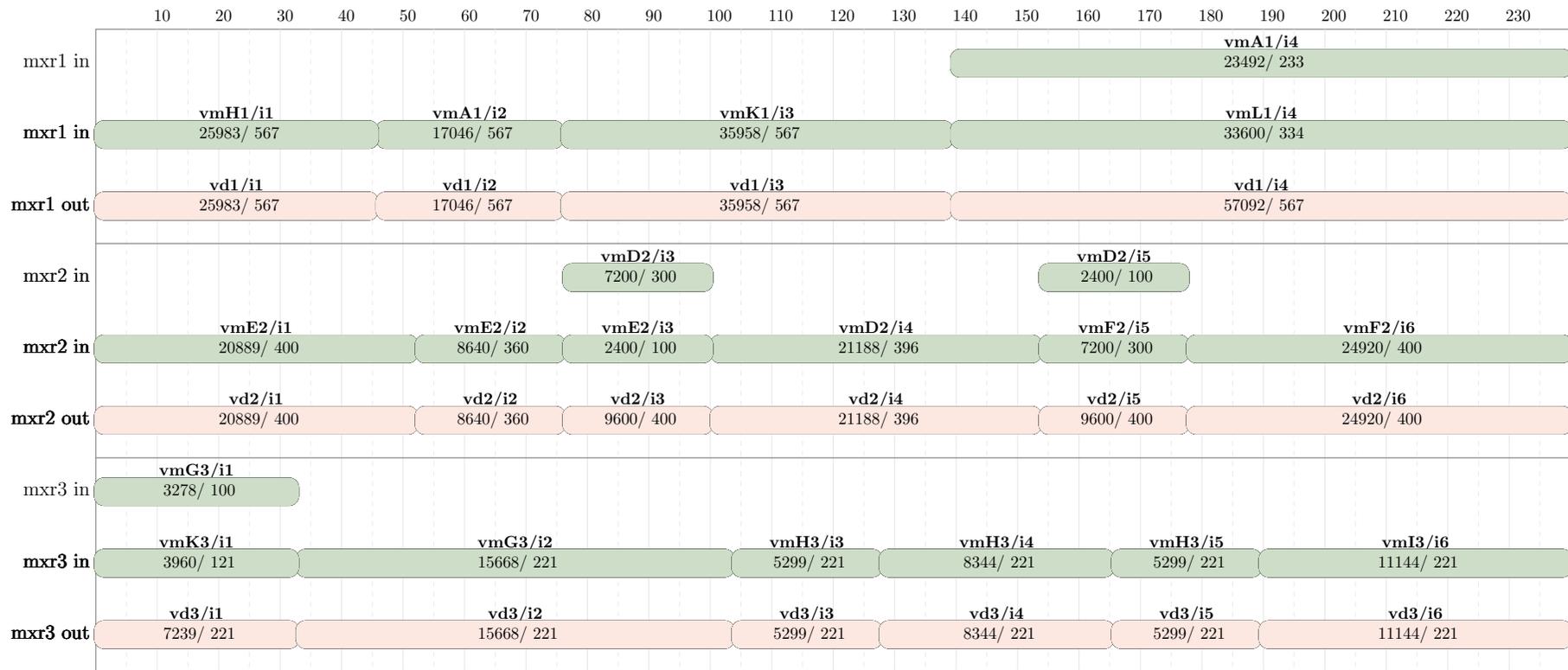
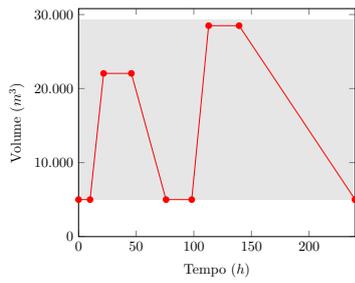
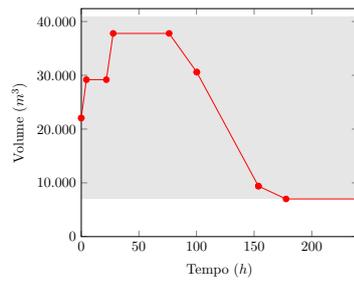


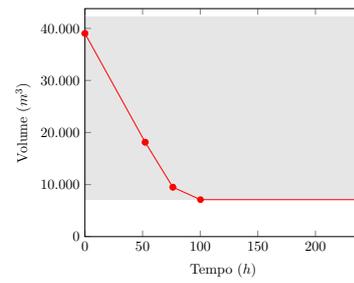
Figura C.33: Programação nos misturadores com informações de volume e vazão para o cenário três com descarregamentos sobrepostos ($n = 6$).



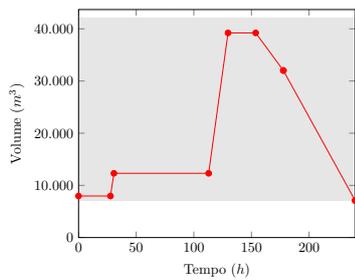
(a)



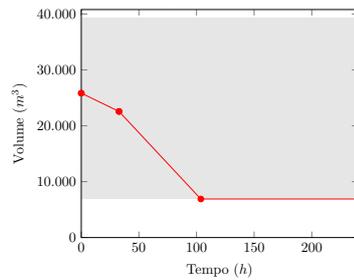
(b)



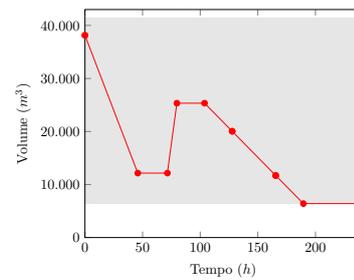
(c)



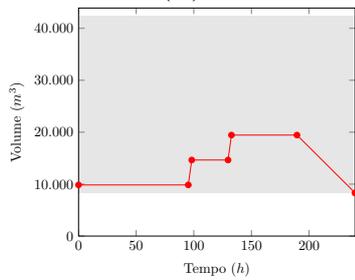
(d)



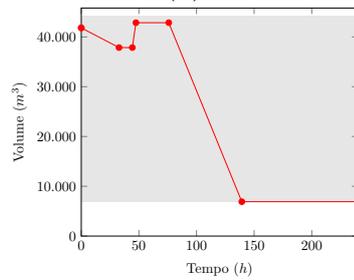
(e)



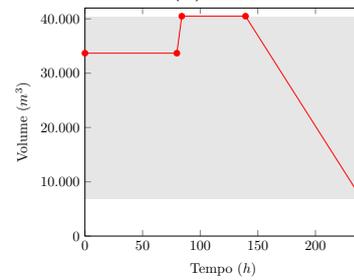
(f)



(g)



(h)



(i)

Figura C.34: Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário três com descarregamentos sobrepostos.

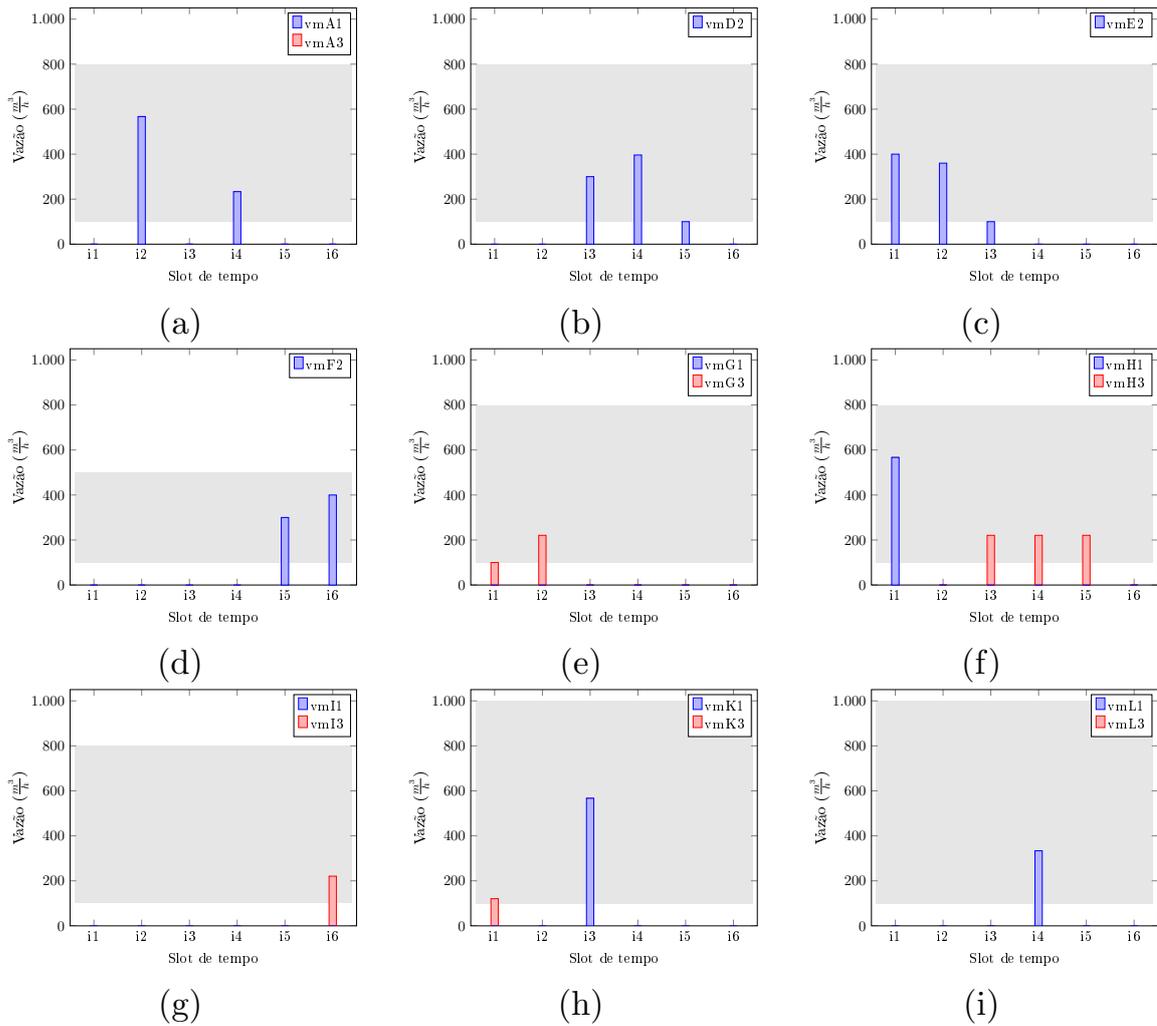


Figura C.35: Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário três com descarregamentos sobrepostos.

Cenário quatro

O melhor programa para o cenário quatro com descarregamentos sobrepostos na bateria de testes Bat-1 foi obtido com $n = 6$.

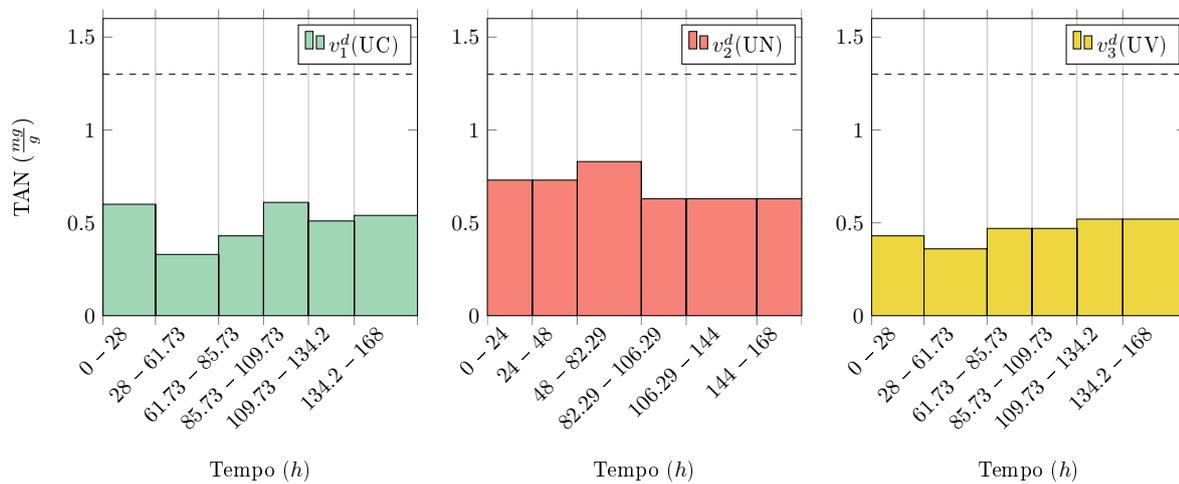


Figura C.36: Perfil de acidez nas operações de destilação W_D no cenário quatro com descarregamentos sobrepostos.

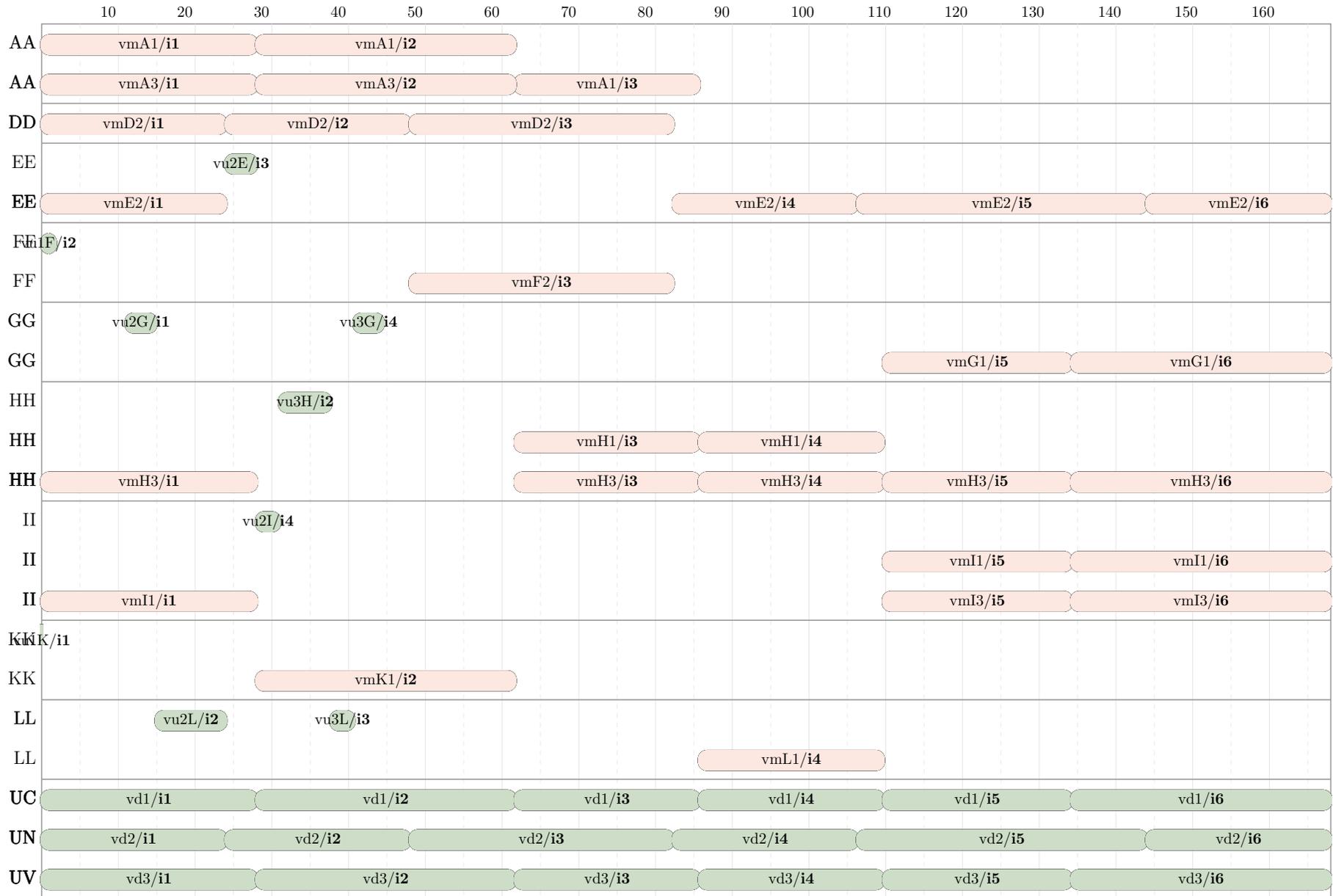


Figura C.37: Gráfico de Gantt para o cenário quatro com descarregamentos sobrepostos ($n = 6$).

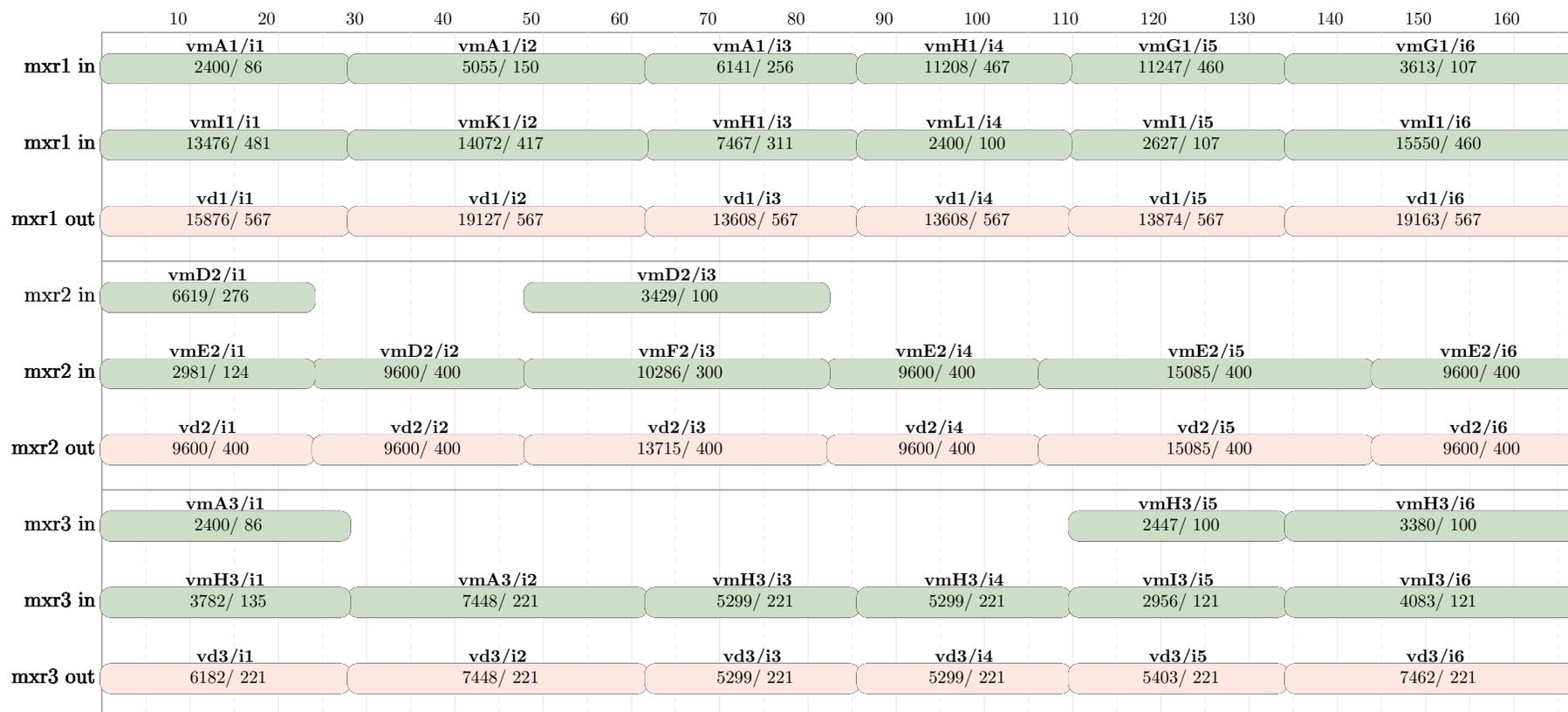


Figura C.38: Programação nos misturadores com informações de volume e vazão para o cenário quatro com descarregamentos sobrepostos ($n = 6$).

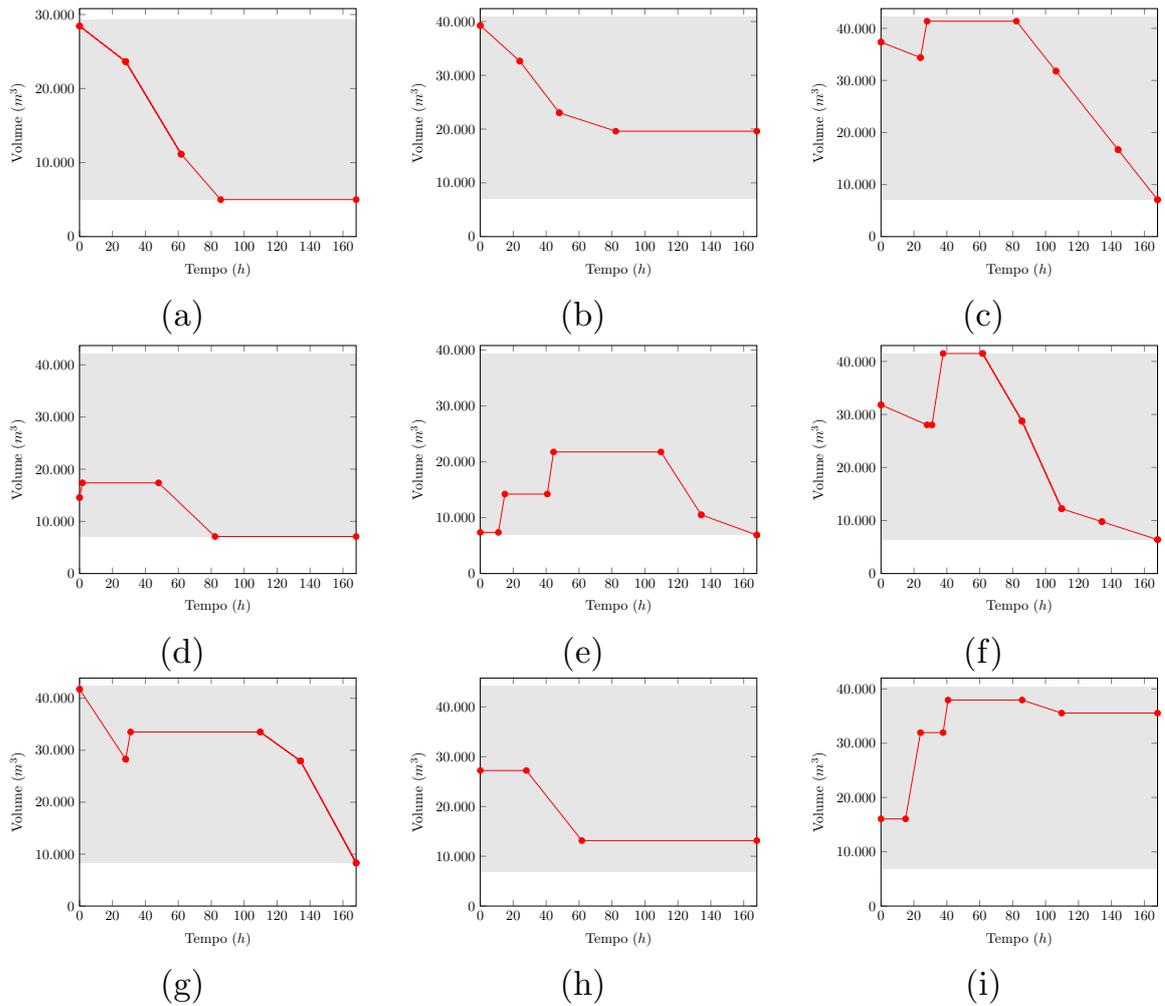


Figura C.39: Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário quatro com descarregamentos sobrepostos.

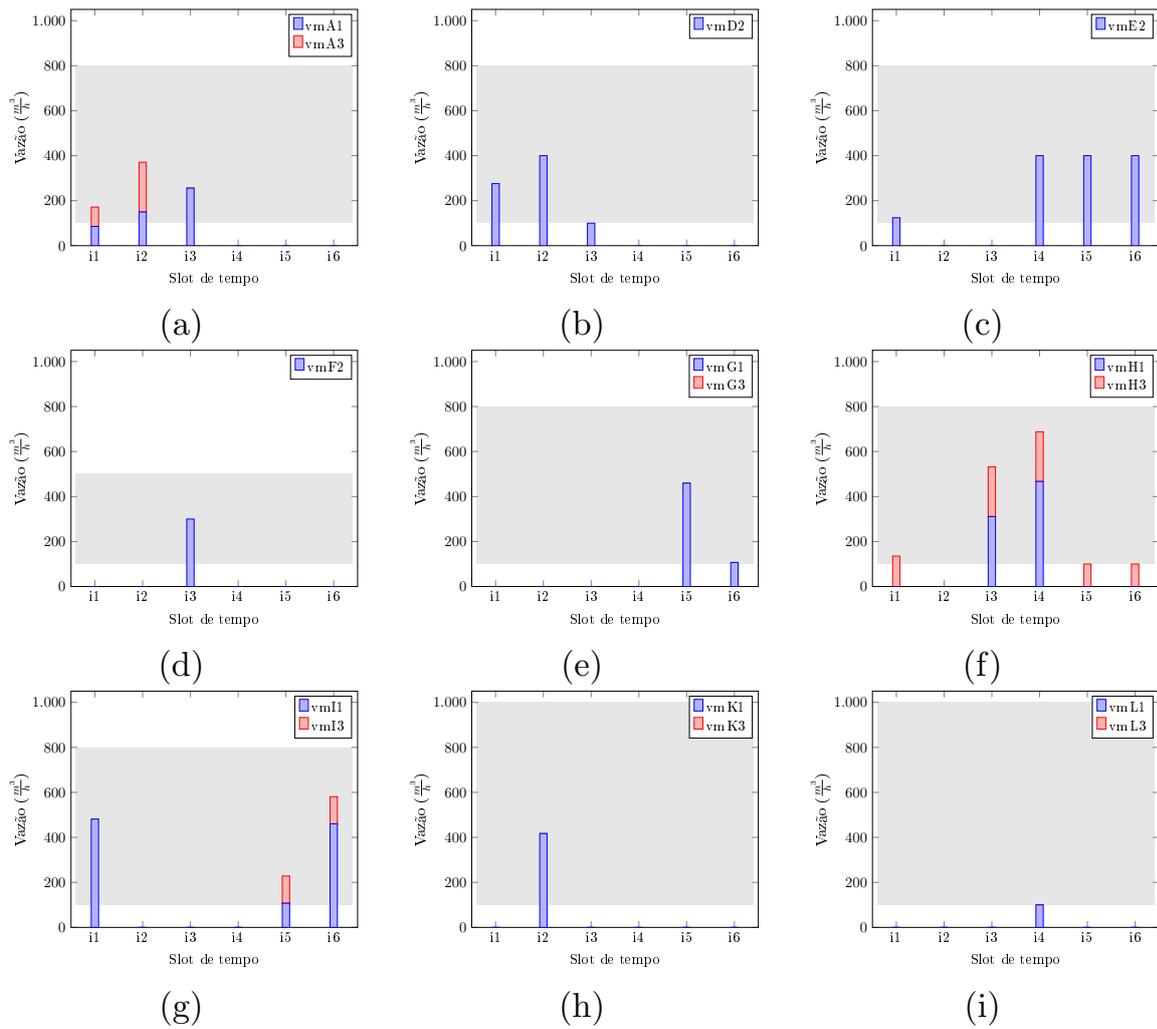


Figura C.40: Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário quatro com descarregamentos sobrepostos.

Cenário cinco

O melhor programa para o cenário cinco com descarregamentos sobrepostos na bateria de testes Bat-1 foi obtido com $n = 4$.

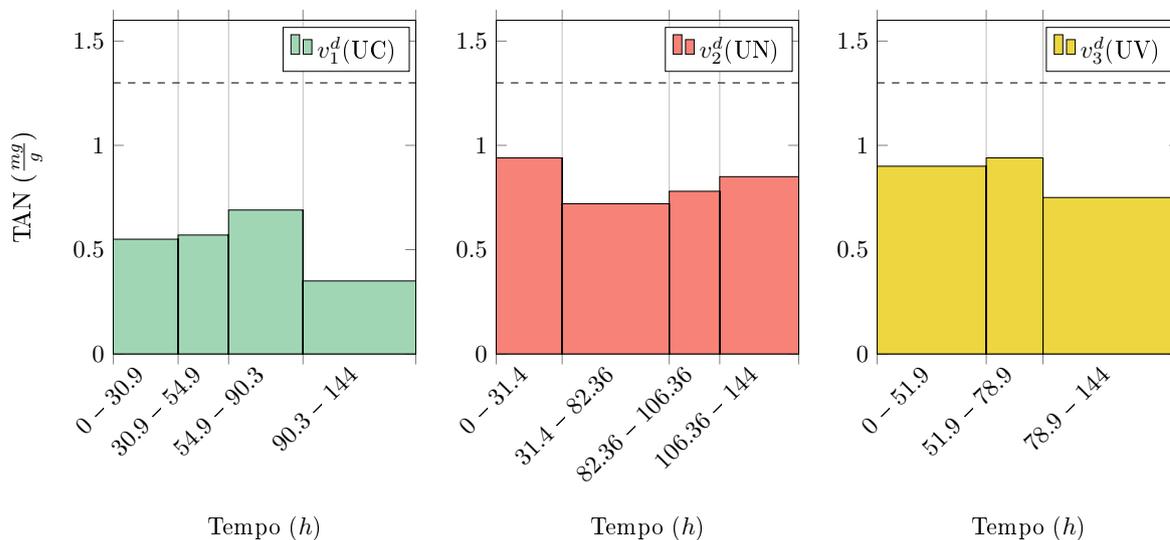


Figura C.41: Perfil de acidez nas operações de destilação W_D no cenário cinco com descarregamentos sobrepostos.

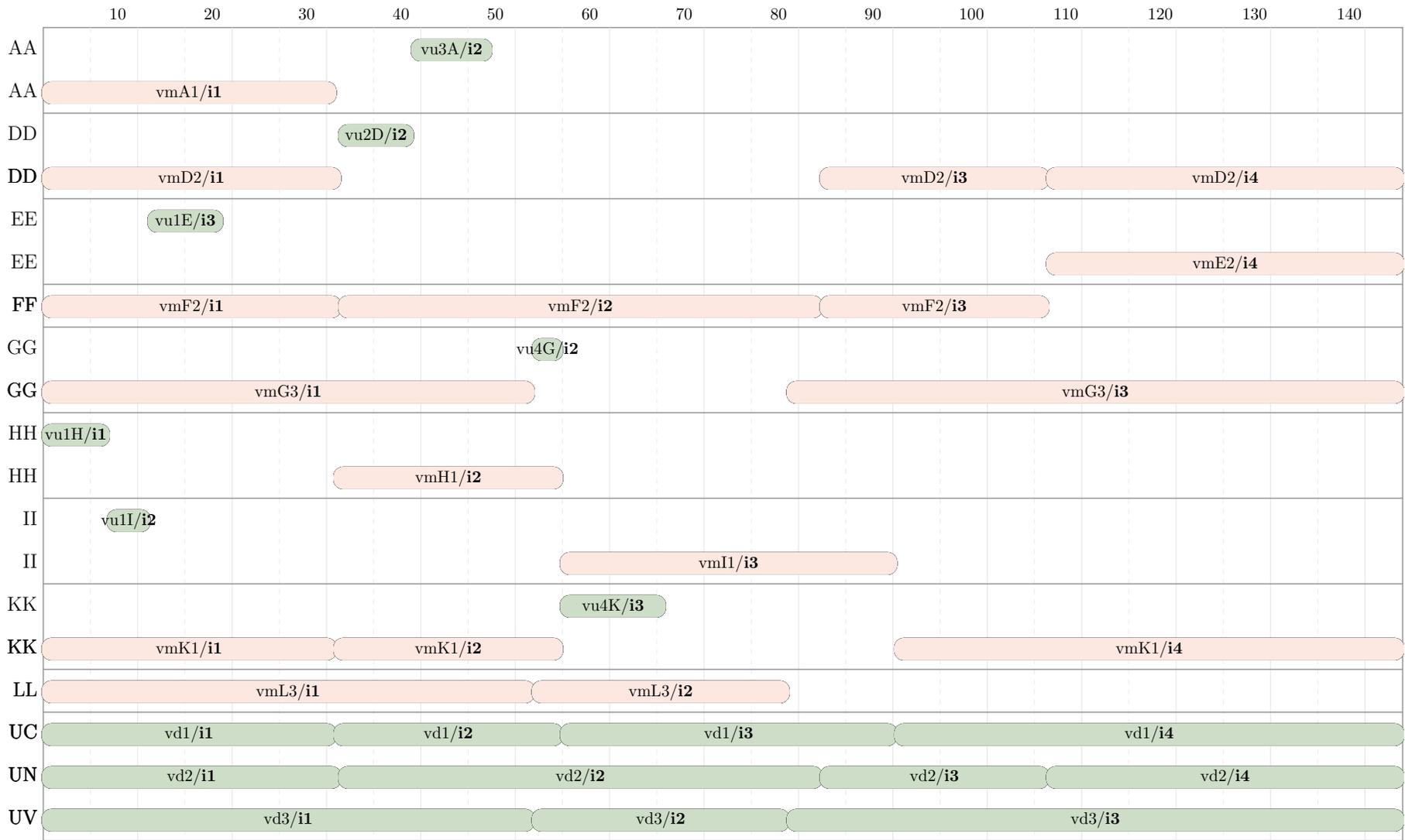


Figura C.42: Gráfico de Gantt para o cenário cinco com descarregamentos sobrepostos ($n = 4$).

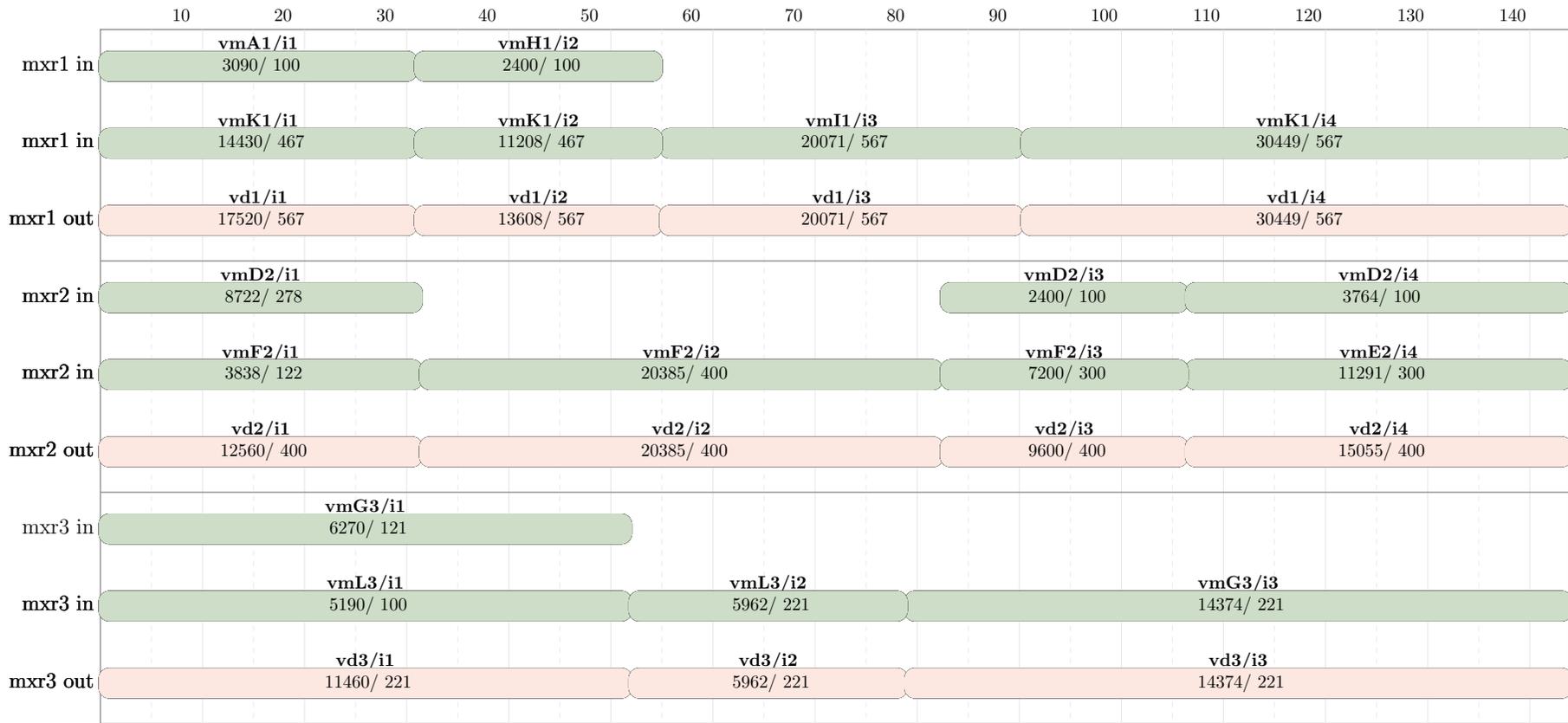


Figura C.43: Programação nos misturadores com informações de volume e vazão para o cenário cinco com descarregamentos sobrepostos ($n = 4$).

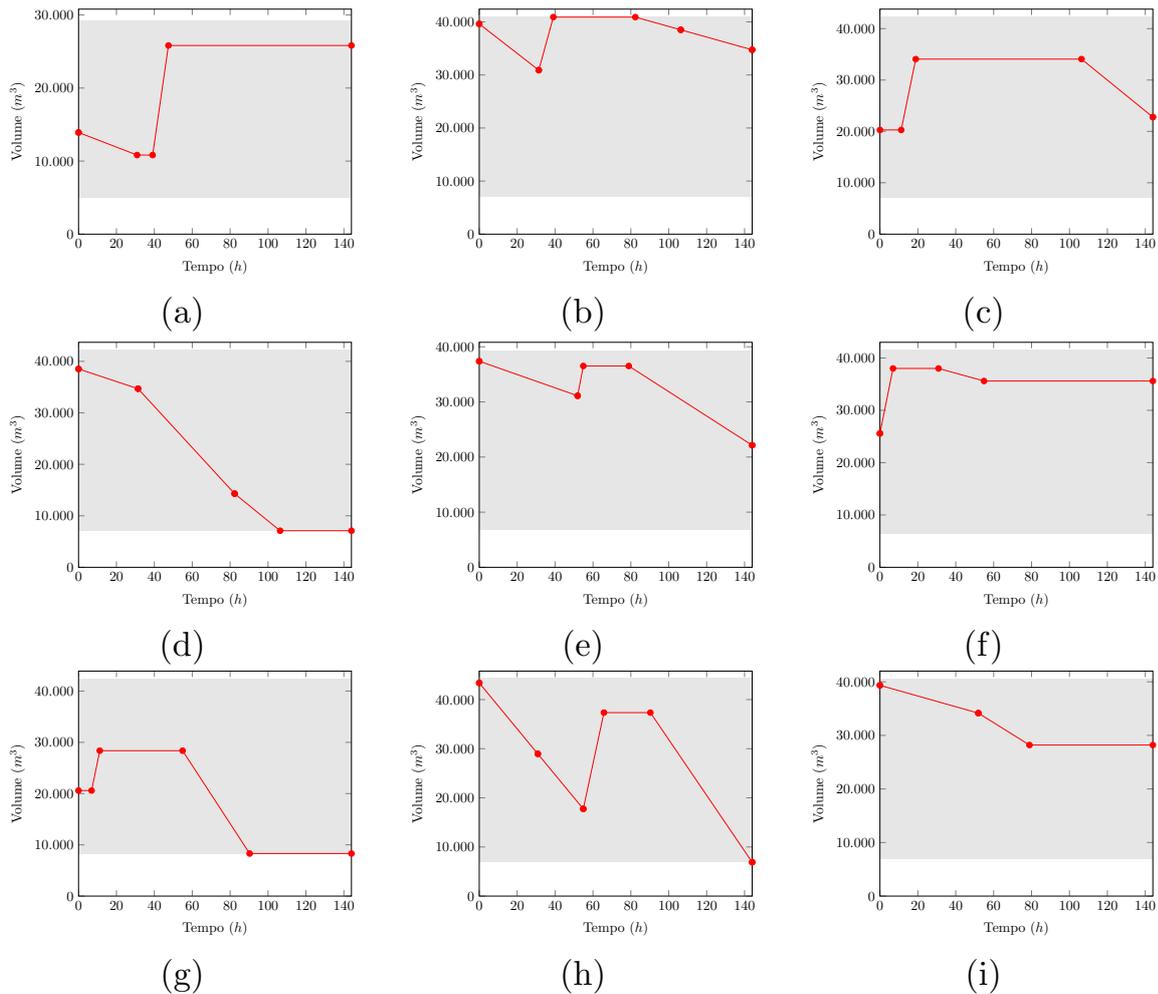


Figura C.44: Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário cinco com descarregamentos sobrepostos.

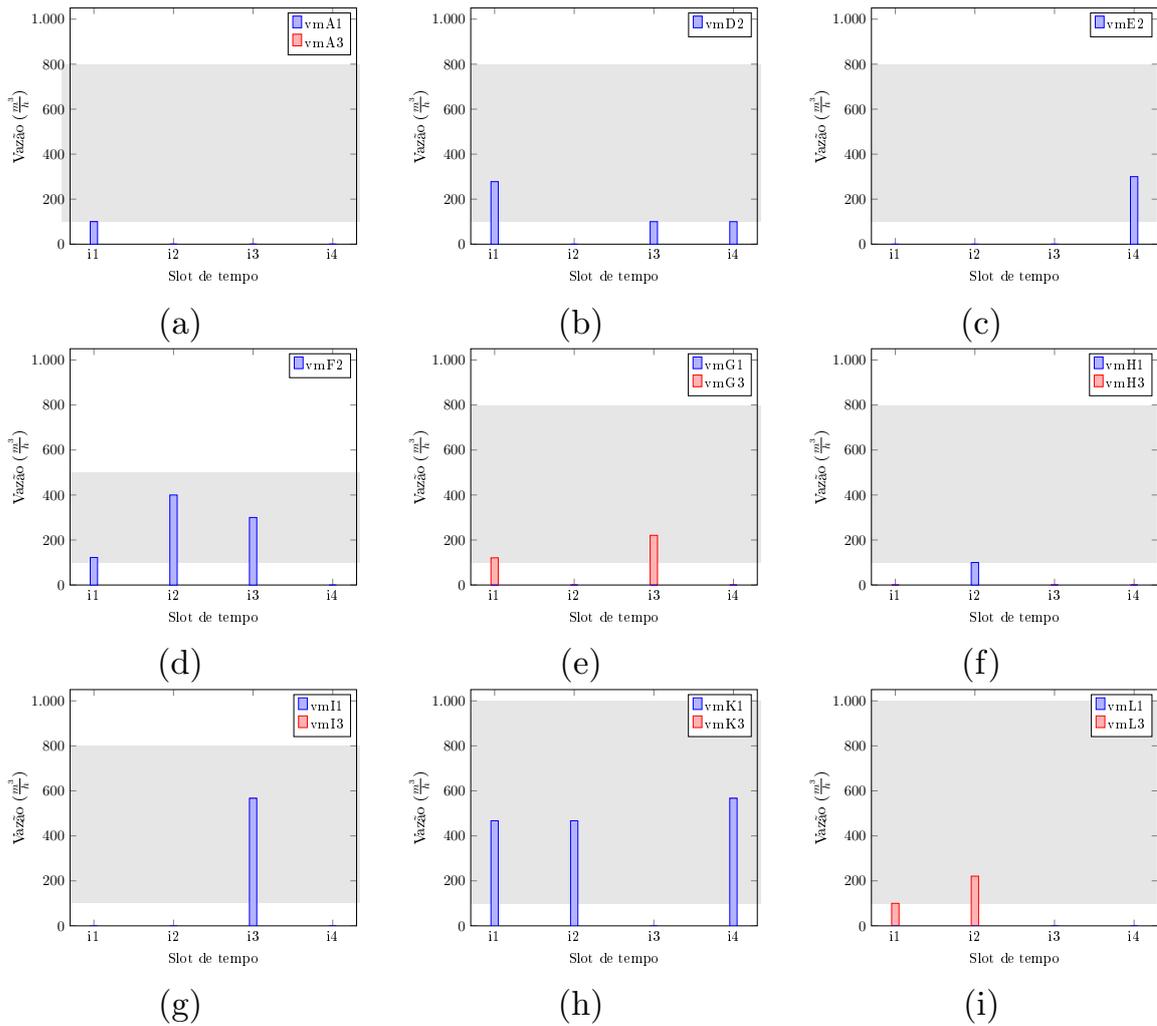


Figura C.45: Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário cinco com descarregamentos sobrepostos.

Cenário seis

O melhor programa para o cenário seis com descarregamentos sobrepostos na bateria de testes Bat-1 foi obtido com $n = 5$.

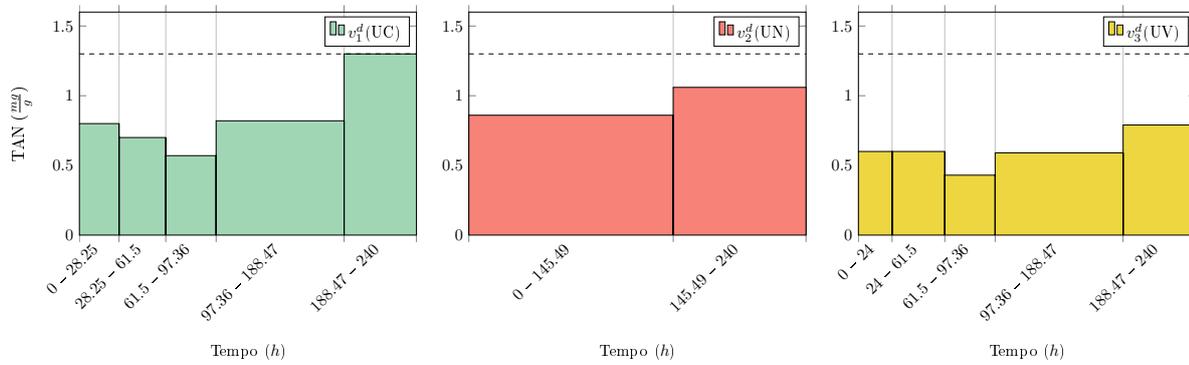


Figura C.46: Perfil de acidez nas operações de destilação W_D no cenário seis com descarregamentos sobrepostos.

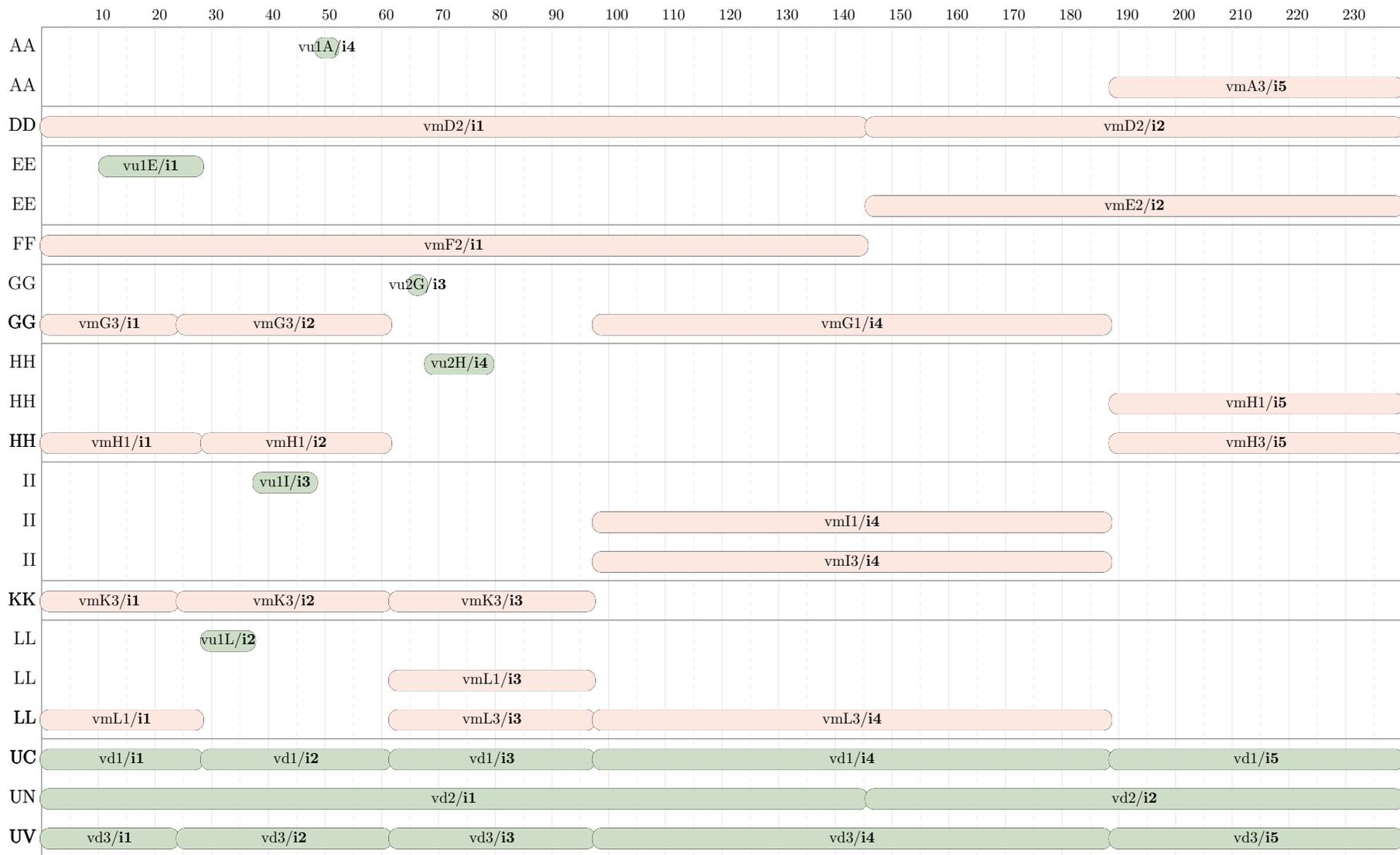


Figura C.47: Gráfico de Gantt para o cenário seis com descarregamentos sobrepostos ($n = 5$).

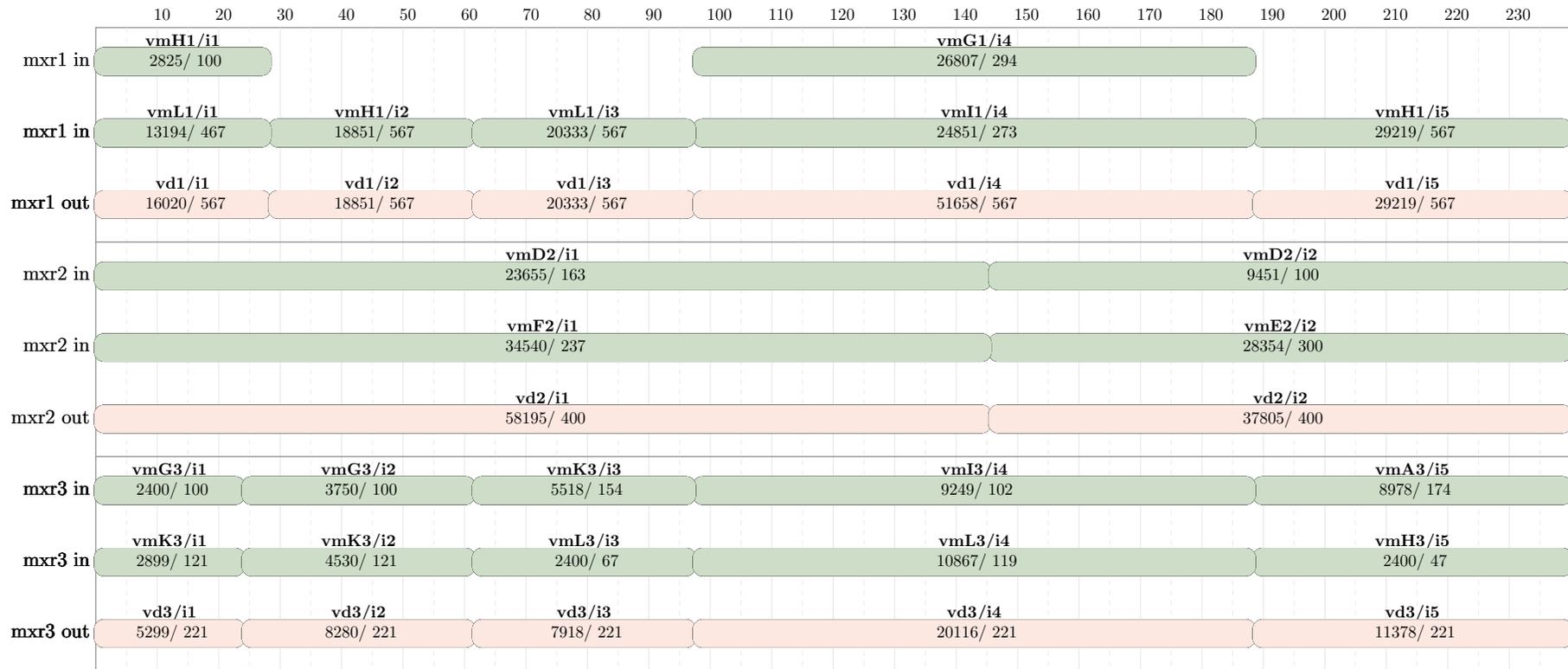


Figura C.48: Programação nos misturadores com informações de volume e vazão para o cenário seis com descarregamentos sobrepostos ($n = 5$).

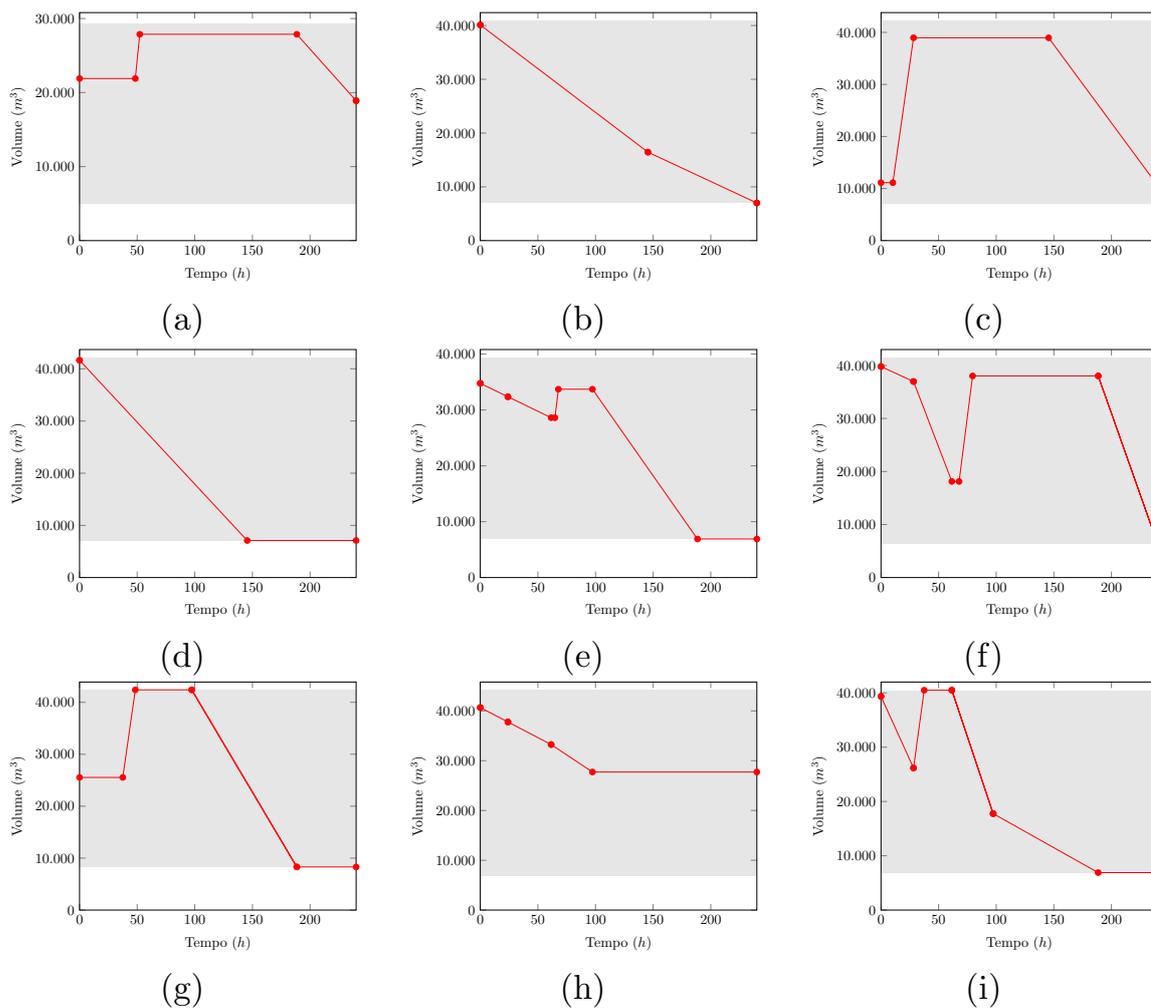


Figura C.49: Inventário nos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário seis com descarregamentos sobrepostos.

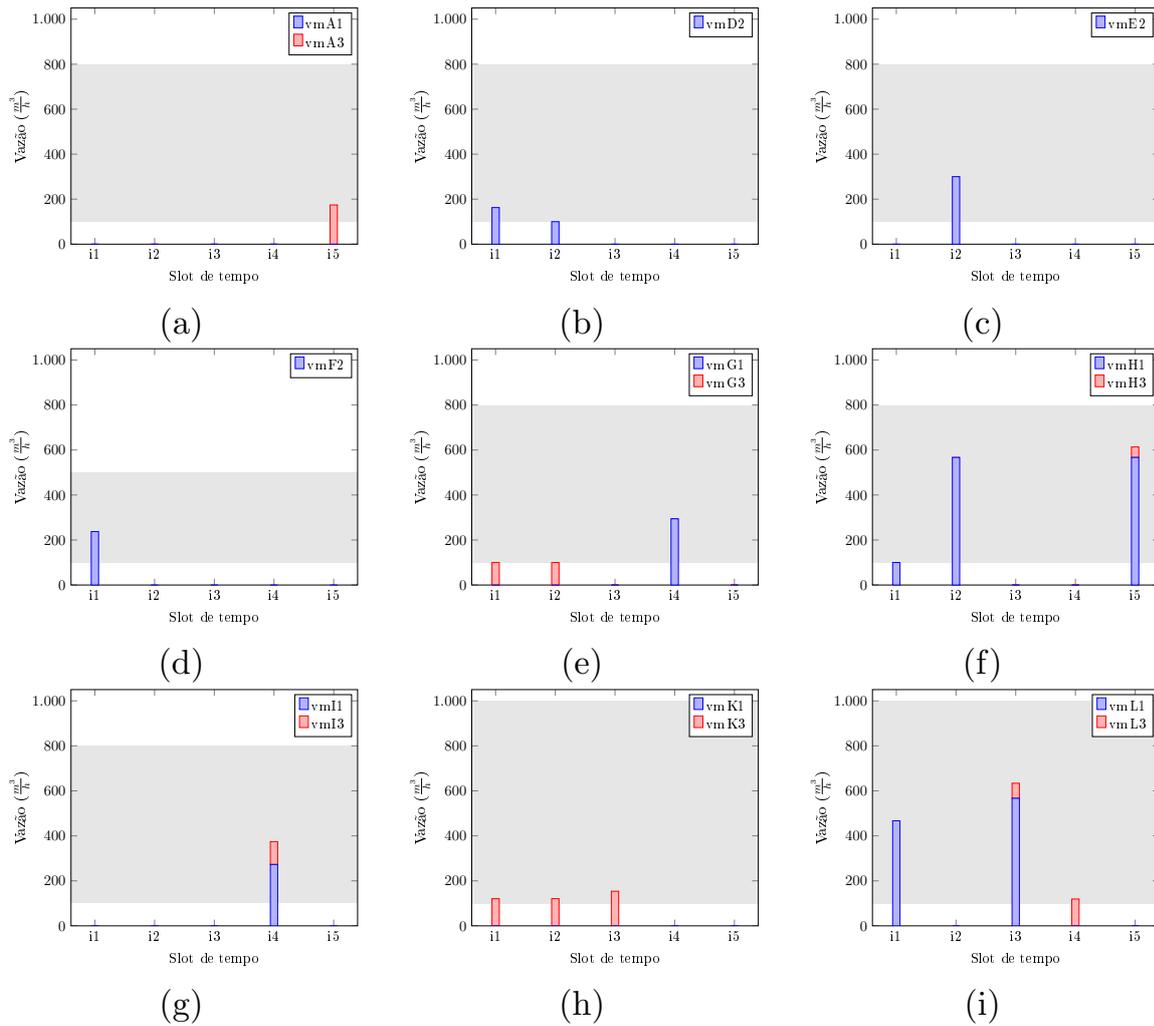


Figura C.50: Vazão de saída dos tanques de carga (a) AA, (b) DD, (c) EE, (d) FF, (e) GG, (f) HH, (g) II, (h) KK e (i) LL no cenário seis com descarregamentos sobrepostos.

RESULTADOS COMPUTACIONAIS

A notação usada para apresentar os resultados computacionais é:

- **status do modelo MILP principal (SM)** de acordo com Rosenthal (2014):
 - 8: solução factível encontrada;
 - 10: não existe solução factível;
 - 13: busca interrompida por falta de memória;
 - 14: solução não encontrada;
- **número de variáveis (NV)**;
- **número de variáveis binárias (NVB)**;
- **número de equações (NE)**;
- **número de equações não-lineares (NENL)**;
- **número de nós do programa inteiro-misto linear principal (NN-MIP)**;
- **número de elementos não-zero na matriz de coeficientes do modelo MILP (NZ-MIP)**;
- **número de elementos não-lineares não-zero na matriz de coeficientes do modelo NLP (NZ-NL)**;
- **melhor limite superior da função objetivo no programa inteiro-misto linear em 1×10^{-3} \$ (FO-LS)**;
- **função objetivo do programa inteiro-misto principal em 1×10^{-3} \$ (FO-MIP)**;
- **função objetivo do programa inteiro-misto associado em 1×10^{-3} \$ (FO-MIPa)¹**;

¹A FO-MIPa é o programa MILP de mesma solução inteira do NLP reportado.

- *gap* de otimalidade do programa inteiro-misto linear associado em % (GAPa) de acordo com a Equação D.1;

$$\text{GAPa} = 100 \cdot \frac{\text{FO-LS} - \text{FO-MIPa}}{\text{FO-LS}} \quad (\text{D.1})$$

- função objetivo do programa não-linear em 1×10^{-3} \$ (FO-NLP);
- estimativa do *gap* de otimalidade com estratégia de decomposição MILP-NLP em % (GAPd) de acordo com a Equação 5.43 ²;
- número de soluções inviáveis no *pool* de soluções (NSIP);
- tempo computacional do programa inteiro-misto linear em minutos (t-MIP);
- tempo computacional do programa não-linear linear em minutos (t-NLP);
- tempo computacional total em minutos (t-TOTAL);

²A função objetivo MILP usada como limite superior para estimar o GAPd foi a do MILP principal.

Modelagem com descarregamentos não-sobrepostos

Tabela D.1: Resultados computacionais da modelagem com descarregamentos não-sobrepostos.

cenário	<i>n</i>	SM	NV	NVB	NE	NENL	NN-MIP	NZ-MIP	NZ-NL	FO-LS	FO-MIP	FO-MIPa	GAPa	FO-NLP	GAPd	NSIP	t-MIP	t-NLP	t-TOTAL
2	i5	8	7.911	200	7.205	1.690	15.470	58.572	6.760	46.500,09	46.500,09	46.476,23	0,05	45.597,09	1,94	3	2,30	1,31	3,61
	i6	8	9.493	240	8.583	2.028	61.156	76.828	8.112	46.662,99	46.662,99	46.510,37	0,33	45.680,23	2,11	12	26,75	1,56	28,32
	i7	8	11.075	280	9.978	2.366	49.053	97.402	9.464	46.705,07	46.705,07	46.541,85	0,35	45.605,85	2,35	17	24,14	3,00	27,14
	i8	8	12.657	320	11.390	2.704	528.328	120.373	10.816	46.749,83	46.705,07	46.583,19	0,36	45.624,43	2,31	11	89,10	2,79	91,89
	i9	8	14.239	360	12.819	3.042	73.766	145.820	12.168	46.751,26	46.609,96	45.577,19	2,51	44.810,03	3,86	10	69,15	3,04	72,18
3	i5	10	15.851	315	8.499		84										0,11		0,11
	i6	8	19.021	378	13.298	3.240	17.380	165.451	12.960	60.873,47	60.873,47	60.783,89	0,15	59.823,50	1,72	0	6,13	10,22	16,35
	i7	8	22.191	441	15.416	3.780	27.500	210.320	15.120	66.920,14	66.920,14	66.920,14	0,00	65.891,75	1,54	0	31,26	20,52	51,77
	i8	8	25.361	504	17.553	4.320	4.270	260.391	17.280	66.921,98	66.921,98	66.921,98	0,00	65.899,90	1,53	1	3,49	24,36	27,85
	i9	8	28.531	567	19.709	4.860	3.015	315.802	19.440	66.921,98	66.921,98	66.921,98	0,00	65.850,89	1,60	0	1,37	45,39	46,76
4	i5	8	11.701	225	9.699	2.700	5.576	85.180	10.800	48.514,79	48.514,79	48.471,11	0,09	46.590,43	3,49	1	0,99	2,98	3,97
	i6	8	14.041	270	11.524	3.240	7.841	111.715	12.960	48.559,41	48.559,41	48.546,37	0,03	46.732,09	3,72	0	4,41	6,40	10,81
	i7	8	16.381	315	13.368	3.780	9.583	141.568	15.120	48.569,13	48.569,13	48.428,86	0,29	47.031,54	3,17	4	6,86	8,47	15,33
	i8	8	18.721	360	15.231	4.320	62.539	174.829	17.280	48.348,31	48.275,22	48.272,04	0,16	46.722,62	3,22	8	120,00	8,42	128,43
	i9	14	21.061	405	12.253		86.061			48.407,92							120,00		120,00
5	i5	8	13.776	270	10.494	2.700	2.946	105.179	10.800	41.646,25	41.646,25	41.506,69	0,34	39.942,48	4,09	0	0,80	5,19	5,99
	i6	8	16.531	324	12.465	3.240	7.665	138.286	12.960	42.156,75	42.156,75	42.156,75	0,00	40.207,97	4,62	2	2,43	8,12	10,55
	i7	8	19.286	378	14.455	3.780	11.037	175.584	15.120	42.036,04	42.036,04	42.032,90	0,01	40.053,95	4,72	2	3,60	9,69	13,29
	i8	8	22.041	432	16.464	4.320	102.109	217.187	17.280	41.883,85	41.873,11	41.835,25	0,12	39.974,59	4,53	4	50,14	14,63	64,77
	i9	8	24.796	486	18.492	4.860	58.877	263.209	19.440	42.029,65	41.454,70	41.444,44	1,39	39.823,70	3,93	6	47,84	12,79	60,63
6	i5	8	9.626	180	9.084	2.700	17.780	66.234	10.800	69.267,31	69.267,31	69.267,31	0,00	67.830,30	2,06	2	6,14	1,33	7,47
	i6	8	11.551	216	10.799	3.240	13.666	86.440	12.960	69.400,94	69.400,94	69.400,94	0,00	67.827,06	2,27	8	5,55	2,61	8,16
	i7	8	13.476	252	12.533	3.780	2.897	109.091	15.120	69.400,94	69.400,94	69.400,94	0,00	67.768,63	2,35	0	1,89	3,86	5,75
	i8	8	15.401	288	14.286	4.320	5.948	134.253	17.280	69.400,94	69.400,94	69.400,94	0,00	67.727,13	2,41	1	5,46	5,49	10,96
	i9	8	17.326	324	16.058	4.860	4.844	161.992	19.440	69.400,94	69.400,94	69.400,94	0,00	67.596,66	2,60	1	6,78	9,82	16,60

Modelagem com descarregamentos sobrepostos

Tabela D.2: Resultados computacionais da modelagem com descarregamentos sobrepostos.

cenário	<i>n</i>	SM	NV	NVB	NE	NENL	NN-MIP	NZ-MIP	NZ-NL	FO-LS	FO-MIP	FO-MIPa	GAPa	FO-NLP	GAPd	NSIP	t-MIP	t-NLP	t-TOTAL
2	i3	8	4.747	120	4.833	1.014	24.928	29.485	4.056	46.488,32	46.488,32	46.488,32	0,00	45.610,22	1,89	10	1,49	0,20	1,69
	i4	8	6.329	160	6.438	1.352	26.291	44.685	5.408	46.727,98	46.727,98	46.078,25	1,39	45.421,01	2,80	11	2,61	0,48	3,10
	i5	8	7.911	200	8.135	1.690	57.092	62.876	6.760	46.750,23	46.750,23	46.583,54	0,36	45.555,02	2,54	5	10,82	1,48	12,30
	i6	8	9.493	240	9.924	2.028	65.611	84.273	8.112	46.758,41	46.758,41	46.758,41	0,00	45.486,89	2,72	12	23,45	1,49	24,93
	i7	8	11.075	280	11.805	2.366	116.045	109.091	9.464	46.756,37	46.756,37	46.694,23	0,13	45.591,83	2,49	12	103,21	2,58	105,79
3	i3	8	9.511	189	7.721	1.620	52.485	60.535	6.480	66.640,70	66.640,70	66.640,70	0,00	65.480,87	1,74	0	3,33	1,06	4,40
	i4	8	12.681	252	10.299	2.160	124.423	92.423	8.640	66.921,98	66.921,98	66.920,14	0,00	65.815,85	1,65	0	11,50	2,67	14,18
	i5	8	15.851	315	13.044	2.700	8.979	130.731	10.800	66.921,98	66.921,98	66.921,98	0,00	65.875,13	1,56	0	1,56	6,31	7,86
	i6	8	19.021	378	15.956	3.240	15.807	175.861	12.960	66.921,98	66.921,98	66.921,98	0,00	65.876,19	1,56	0	3,74	10,20	13,94
	i7	8	22.191	441	19.035	3.780	12.532	228.215	15.120	66.921,98	66.921,98	66.901,53	0,03	65.732,48	1,78	0	6,96	19,97	26,92
4	i3	8	7.021	135	6.487	1.620	4.716	42.517	6.480	48.555,34	48.555,34	48.081,00	0,98	46.835,03	3,54	1	0,27	0,75	1,01
	i4	8	9.361	180	8.573	2.160	3.333	64.183	8.640	48.569,13	48.569,13	48.507,54	0,13	46.872,49	3,49	1	0,38	1,89	2,27
	i5	8	11.701	225	10.764	2.700	4.633	89.941	10.800	48.569,13	48.569,13	48.079,64	1,01	46.878,36	3,48	0	1,19	4,24	5,43
	i6	8	14.041	270	13.060	3.240	2.942	120.037	12.960	48.569,13	48.569,13	48.425,18	0,30	46.945,64	3,34	0	1,02	7,15	8,17
	i7	8	16.381	315	15.461	3.780	23.614	154.717	15.120	48.569,13	48.569,13	48.559,41	0,02	46.907,02	3,42	1	15,53	13,02	28,55
5	i3	8	8.266	162	7.131	1.620	5.923	51.580	6.480	42.325,87	42.325,87	42.242,48	0,20	40.492,95	4,33	1	0,25	1,29	1,54
	i4	8	11.021	216	9.472	2.160	6.205	78.375	8.640	42.327,64	42.327,64	42.182,46	0,34	40.514,40	4,28	0	0,63	3,11	3,73
	i5	8	13.776	270	11.949	2.700	4.633	110.426	10.800	42.327,64	42.327,64	42.302,52	0,06	40.234,79	4,94	0	1,08	6,59	7,67
	i6	8	16.531	324	14.562	3.240	11.747	148.057	12.960	42.327,64	42.327,64	42.274,87	0,12	40.254,88	4,90	1	3,55	11,44	14,99
	i7	14	19.286	378	13.531		309.339			42.010,96							120,00		120,00
6	i3	10	5.776	108	4.331		9.067										1,41		1,41
	i4	8	7.701	144	7.818	2.160	82.219	50.315	8.640	69.400,94	69.400,94	69.400,94	0,00	67.753,87	2,37	1	10,37	0,82	11,19
	i5	8	9.626	180	9.759	2.700	7.624	69.861	10.800	69.400,94	69.400,94	69.208,29	0,28	67.806,80	2,30	0	2,38	1,44	3,82
	i6	8	11.551	216	11.774	3.240	15.452	92.503	12.960	69.400,94	69.400,94	69.258,80	0,20	67.803,43	2,30	1	5,97	2,85	8,82
	i7	8	13.476	252	13.863	3.780	6.266	118.409	15.120	69.400,94	69.400,94	69.400,94	0,00	67.674,54	2,49	0	5,73	5,10	10,82

REFERÊNCIAS BIBLIOGRÁFICAS

- ACHTERBERG, T.; WUNDERLING, R. Mixed integer programming: Analyzing 12 years of progress. Em: *Facets of Combinatorial Optimization*. Springer, 2013, pp. 449–481. DOI: https://doi.org/10.1007/978-3-642-38189-8_18.
- ANDREI, N. Nonlinear Optimization Applications Using the GAMS Technology. Em: Springer, 2013. Cap. 7.7 – Pooling: Blending with two ingredients, one tank and two finite products (Pool1), pp. 178–181. DOI: <https://doi.org/10.1007/978-1-4614-6797-7>.
- ARTIBA, A.; ELMAGHRABY, S. *The Planning and Scheduling of Production Systems. Methodologies and applications*. Chapman & Hall, 1997.
- BEBER, M.; BECKER, T. Towards an Understanding of the Relation between Topological Characteristics and Dynamic Behavior in Manufacturing Networks. *Procedia CIRP*, vol. 19. 2nd CIRP Robust Manufacturing Conference (RoMac 2014), pp. 21–26, 2014. ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2014.05.005>.
- BLACKSTONE, J., ed. *APICS Dictionary*. 14th edition. APICS, 2013.
- BLĄŻEWICZ, J.; ECKER, K.; PESCH, E.; SCHMIDT, G.; WEGLARZ, J. *Handbook on scheduling: from theory to applications*. Springer Science & Business Media, 2007.
- BROOKE, A.; KENDRICK, D.; MEERAUS, A.; RAMAN, R. GAMS: the solver manuals. *Washington, DC, GAMS Development Corporation*, 1998.
- BRUCKER, P.; KNUST, S. Complex Scheduling. Em: Springer, 2012. Cap. 2 – Algorithms and Complexity, pp. 65–82. DOI: <https://doi.org/10.1007/978-3-642-23929-8>.
- CASTRO, P.; BARBOSA-PÓVOA, A.; MATOS, H. An Improved RTN Continuous-Time

- Formulation for the Short-term Scheduling of Multipurpose Batch Plants. *Industrial & Engineering Chemistry Research*, vol. 40, nº 9, pp. 2059–2068, 2001. DOI: <https://doi.org/10.1021/ie000683r>.
- CHEN, X.; GROSSMANN, I.; ZHENG, L. A comparative study of continuous-time models for scheduling of crude oil operations in inland refineries. *Computers & Chemical Engineering*, vol. 44, pp. 141–167, 2012. DOI: <https://doi.org/10.1016/j.compchemeng.2012.05.009>.
- CHENG, J.; ZHU, L.; KE, Y.; CHU, S. Fast Algorithms for Maximal Clique Enumeration with Limited Memory. Em: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '12. Beijing, China: ACM, 2012, pp. 1240–1248. ISBN: 978-1-4503-1462-6. DOI: <https://doi.org/10.1145/2339530.2339724>.
- EXTENSIONS, O. S. GAMS-The Solver Manuals, 2001.
- FLOUDAS, C. Nonlinear and mixed-integer optimization: fundamentals and applications. Em: Oxford University Press, 1995. Cap. 3 - Fundamentals of Nonlinear Optimization, pp. 45–73.
- FLOUDAS, C. Nonlinear and mixed-integer optimization: fundamentals and applications. Em: Oxford University Press, 1995. Cap. 2 - Convex Analysis, pp. 17–44.
- FLOUDAS, C. Nonlinear and mixed-integer optimization: fundamentals and applications. Em: Oxford University Press, 1995. Cap. 7 - Process Synthesis, pp. 225–257.
- GANESH, B. Scheduling of single-stage noncontinuous processes. Diss. de mestrado. National University of Singapore, 2004.
- GEISLER, B.; MARTIN, A.; MORSI, A.; SCHEWE, L. Mixed Integer Nonlinear Programming. Em: ed. por LEE, J.; LEYFFER, S. Springer, 2012. Cap. Part V: Convexification and Linearization, pp. 286–369.
- HAMACHER, H.; KLAMROTH, K. *Linear Optimierung und Netzwerkoptimierung – Zweisprachige Ausgabe Deutsch Englisch*. Vieweg, 2006.

- HARJUNKOSKI, I.; MARAVELIAS, C.; BONGERS, P.; CASTRO, P.; ENGELL, S.; GROSSMANN, I.; HOOKER, J.; M'ENDEZ, C.; SAND, G.; WASSICK, J. Scope for industrial applications of production scheduling models and solution methods. *Computers & Chemical Engineering*, vol. 62, pp. 161–193, 2014. ISSN: 0098-1354. DOI: <https://doi.org/10.1016/j.compchemeng.2013.12.001>.
- HEIDRICH, P. Contribuição do MRP na gestão estratégica da manufatura. *II Simpósio de Excelência em Gestão de Tecnologia*, pp. 969–977, 2005.
- HILLIER, F.; LIEBERMAN, G. *Introduction to operations research*. McGraw-Hill Education, 2001.
- IERAPETRITOU, M.; FLOUDAS, C. Effective Continuous-Time Formulation for Short-Term Scheduling. 1. Multipurpose Batch Processes. *Industrial & Engineering Chemistry Research*, vol. 37, nº 11, pp. 4341–4359, 1998. DOI: <https://doi.org/10.1021/ie970927g>.
- IERAPETRITOU, M.; FLOUDAS, C. Effective Continuous-Time Formulation for Short-Term Scheduling. 2. Continuous and Semicontinuous Processes. *Industrial & Engineering Chemistry Research*, vol. 37, nº 11, pp. 4360–4374, 1998. DOI: <https://doi.org/10.1021/ie9709289>.
- JIA, Z.; IERAPETRITOU, M. Efficient short-term scheduling of refinery operations based on a continuous time formulation. *Computers & chemical engineering*, vol. 28, nº 6, pp. 1001–1019, 2004. DOI: <https://doi.org/10.1016/j.compchemeng.2003.09.007>.
- JIA, Z.; IERAPETRITOU, M.; KELLY, J. Refinery Short-Term Scheduling Using Continuous Time Formulation: Crude-Oil Operations. *Industrial & Engineering Chemistry Research*, vol. 42, nº 13, pp. 3085–3097, 2003.
- KALLRATH, J. Planning and scheduling in the process industry. English. *OR Spectrum*, vol. 24, nº 3, pp. 219–250, 2002. ISSN: 0171-6468. DOI: <https://doi.org/10.1007/s00291-002-0101-7>.
- KONDILI, E.; PANTELIDES, C.; SARGENT, R. A general algorithm for short-term scheduling of batch operations-I. MILP formulation. *Computers & Chemical Engi-*

neering, vol. 17, nº 2. An International Journal of Computer Applications in Chemical Engineering, pp. 211–227, 1993. DOI: [https://doi.org/10.1016/0098-1354\(93\)80015-F](https://doi.org/10.1016/0098-1354(93)80015-F).

LASSCHUIT, W.; THIJSSSEN, N. Supporting supply chain planning and scheduling decisions in the oil and chemical industry. *Computers & chemical engineering*, vol. 28, nº 6, pp. 863–870, 2004. DOI: <https://doi.org/10.1016/j.compchemeng.2003.09.026>.

LEE, H.; PINTO, J.; GROSSMANN, I.; PARK, S. Mixed-Integer Linear Programming Model for Refinery Short-Term Scheduling of Crude Oil Unloading with Inventory Management. *Industrial & Engineering Chemistry Research*, vol. 35, nº 5, pp. 1630–1641, 1996. DOI: <https://doi.org/10.1021/ie950519h>.

LIBERTI, L.; PANTELIDES, C. An exact reformulation algorithm for large nonconvex NLPs involving bilinear terms. *Journal of Global Optimization*, vol. 36, pp. 161–189, 2005. DOI: <https://doi.org/10.1007/s10898-006-9005-4>.

LOPES, J. Modelagem da injeção simultânea de misturas durante transição de cargas nas unidades de destilação de refinarias de petróleo para minimização da perturbação nas composições. Diss. de mestrado. Universidade Federal de Uberlândia, 2014.

MARAVELIAS, C. General framework and modeling approach classification for chemical production scheduling. *AIChE Journal*, vol. 58, nº 6, pp. 1812–1828, 2012. DOI: <https://doi.org/10.1002/aic.13801>.

MARAVELIAS, C.; GROSSMANN, I. New General Continuous-Time State-Task Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants. *Industrial & Engineering Chemistry Research*, vol. 42, nº 13, pp. 3056–3074, 2003. DOI: <https://doi.org/10.1021/ie020923y>.

MÉNDEZ, C.; CERDÁ, J.; GROSSMANN, I.; HARJUNKOSKI, I.; FAHL, M. Logistic Optimization of Chemical Production Processes. Em: ed. por ENGELL, S. Wiley-VCH Verlag, 2008. Cap. 8 – MILP Optimization Models for Short-term Scheduling of Batch Processes, pp. 163–182.

MÉNDEZ, C.; CERDÁ, J.; GROSSMANN, I.; HARJUNKOSKI, I.; FAHL, M. State-of-

- the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering*, vol. 30, n° 6–7, pp. 913–946, 2006. ISSN: 0098-1354. DOI: <https://doi.org/10.1016/j.compchemeng.2006.02.008>.
- MOURET, S.; GROSSMANN, I.; PESTIAUX, P. A Novel Priority-Slot Based Continuous-Time Formulation for Crude-Oil Scheduling Problems. *Industrial & Engineering Chemistry Research*, vol. 48, n° 18, pp. 8515–8528, 2009. DOI: <https://doi.org/10.1021/ie8019592>.
- MOURET, S.; GROSSMANN, I.; PESTIAUX, P. Time representations and mathematical models for process scheduling problems. *Computers & Chemical Engineering*, vol. 35, n° 6, pp. 1038–1063, 2011. DOI: <https://doi.org/10.1016/j.compchemeng.2010.07.007>.
- PANTELIDES, C. Unified frameworks for optimal process planning and scheduling. Em: *Proceedings on the second conference on foundations of computer aided operations*. Cache Publications New York. 1994, pp. 253–274.
- PINEDO, M. Scheduling. Theory, Algorithms, and Systems. Em: 4th. Springer, 2012. Cap. 2 - Deterministic Models: Preliminaries, pp. 13–33.
- PINEDO, M. Scheduling. Theory, Algorithms, and Systems. Em: 4th. Springer, 2012. Cap. 1 - Introduction, pp. 1–10.
- PINEDO, M. Scheduling. Theory, Algorithms, and Systems. Em: 4th. Springer, 2012. Cap. Apêndice D - Complexity Theory, pp. 589–602.
- PINTO, J. Métodos contínuos de otimização de processos. 2003.
- PINTO, J.; GROSSMANN, I. A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Industrial & Engineering Chemistry Research*, vol. 34, n° 9, pp. 3037–3051, 1995. DOI: <https://doi.org/10.1021/ie00048a015>.
- PINTO, J.; JOLY, M.; MORO, L. Planning and scheduling models for refinery operations. *Computers & Chemical Engineering*, vol. 24, n° 9, pp. 2259–2276, 2000. DOI: [https://doi.org/10.1016/S0098-1354\(00\)00571-8](https://doi.org/10.1016/S0098-1354(00)00571-8).

- POCHET, Y.; WOLSEY, L. Production Planning by Mixed Integer Programming. Em: Springer, 2006. Cap. 3 – Mixed-Integer Programming Algorithms, pp. 77–113.
- POCHET, Y.; WOLSEY, L. Production Planning by Mixed Integer Programming. Em: Springer, 2006. Cap. 1 – Introduction, pp. 3–36.
- QUESADA, I.; GROSSMANN, I. Global optimization of bilinear process networks with multicomponent flows. *Computers & Chemical Engineering*, vol. 19, nº 12, pp. 1219–1242, 1995. DOI: [https://doi.org/10.1016/0098-1354\(94\)00123-5](https://doi.org/10.1016/0098-1354(94)00123-5).
- ROSENTHAL, R. *GAMS - A User's Guide*. 2014.
- SCHMIDT, M. C.; SAMATOVA, N. F.; THOMAS, K.; PARK, B.-H. A scalable, parallel algorithm for maximal clique enumeration. *Journal of Parallel and Distributed Computing*, vol. 69, nº 4, pp. 417–428, 2009. ISSN: 0743-7315. DOI: <https://doi.org/10.1016/j.jpdc.2009.01.003>.
- SHAH, N. Mathematical programming techniques for crude oil scheduling. *Computers & Chemical Engineering*, vol. 20, S1227–S1232, 1996. DOI: [https://doi.org/10.1016/0098-1354\(96\)00212-8](https://doi.org/10.1016/0098-1354(96)00212-8).
- SILVA, A. Programação da produção para o problema de suprimento de petróleo com a eliminação da discrepância da composição nos tanques e a especificação dos produtos da destilação. Diss. de mestrado. Universidade Federal de Uberlândia, 2013.
- SRINIVASAN, A. The value of strong inapproximability results for clique. Em: *Proceedings of the thirty-second annual ACM symposium on Theory of computing*. ACM. 2000, pp. 144–152. DOI: <https://doi.org/10.1145/335305.335322>.
- SUNDARAMOORTHY, A.; KARIMI, I. A simpler better slot-based continuous-time formulation for short-term scheduling in multipurpose batch plants. *Chemical Engineering Science*, vol. 60, nº 10, pp. 2679–2702, 2005. DOI: <https://doi.org/10.1016/j.ces.2004.12.023>.