

Alyne Toscano Martins

**Problema da programação da produção de
bebidas à base de frutas: modelos de
otimização e métodos heurísticos**

São Carlos

2017

Alyne Toscano Martins

Problema da programação da produção de bebidas à base de frutas: modelos de otimização e métodos heurísticos

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de São Carlos como parte dos requisitos para a obtenção do Título de Doutora Engenharia de Produção.

Universidade Federal de São Carlos
Departamento de Engenharia de Produção
Programa de Pós-Graduação em Engenharia de Produção

Orientador: Prof. Dr. Reinaldo Morabito
Coorientador: Profa. Dra. Deisemara Ferreira

São Carlos
2017



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Tese de Doutorado da candidata Alyne Toscano Martins, realizada em 03/07/2017:

Prof. Dr. Reinaldo Morabito Neto
UFSCar

Prof. Dra. Deisemara Ferreira
UFSCar/Sorocaba

Prof. Dr. Martín Gómez Ravetti
UFMG

Prof. Dra. Maria do Socorro Nogueira Rangel
IBILCE/UNESP

Prof. Dr. Maristela Oliveira dos Santos
USP

Prof. Dr. Pedro Augusto Munari Junior
UFSCar

*Aos meus pais, Eurides e Vera, e ao meu irmão José,
Dedico.*

Agradecimentos

Como eu sempre digo, ninguém chega a lugar nenhum sozinho, e para chegar a conclusão desse trabalho muitas pessoas foram essenciais ao longo dessa jornada. Gostaria de expressar minha eterna gratidão a todas elas.

Agradeço ao meu orientador Prof. Dr. Reinaldo Morabito. Obrigada Reinaldo, primeiramente por ter aceitado ser meu orientador, pela confiança depositada em mim, e depois por ter me ensinado tanto sendo um exemplo de pessoa, pesquisador e orientador. Obrigada por toda a tranquilidade e confiança passadas a mim quando os problemas apareciam e eu achava que seria impossível resolvê-los. Adoro dizer para os meus alunos que fui orientada pelo autor do livro que eles estudam, e como esse professor é inteligente, competente e ao mesmo tempo humano. Aprendi muito com você Reinaldo e tenho muita honra em ter sido sua orientada.

O agradecimento para minha coorientadora, Profa. Dra. Deisemara Ferreira é duplo, como orientadora e como amiga. Primeiramente gostaria de agradecer-lá por saber separar tão bem esses dois papéis. Você foi a primeira pessoa que abraçou a ideia desse doutorado e que esteve comigo ao longo de toda essa caminhada. Obrigada por tentar e conseguir sempre extrair o máximo e o melhor de mim (nem eu acreditava tanto em mim como você), obrigada pela confiança, pelas broncas, pelos conselhos e pelo exemplo de profissional que você é. Aprendi muito com a Profa Deisemara Ferreira. Ah... mas com a minha amiga Deise então, quantas histórias para contar ao longo desses anos. Sua amizade me ajudou a ser mais consciente ao longo de todo o processo de doutorado e a trazer leveza a essa jornada como muitas conversas filosóficas, risadas, cafés e cervejas. Obrigada amiga, esse doutorado não seria o mesmo sem você.

Um agradecimento super especial ao GPO (Grupo de Pesquisa Operacional) da UFSCar, vocês fizeram meu doutorado e minha vida em São Carlos muito mais divertidos. A primeira integrante do GPO que me acolheu (literalmente) foi a Gabi. Obrigada Gabi por me hospedar tantas e tantas noites na sua casa, nunca conseguirei retribuir toda sua hospitalidade. Sua amizade foi um grande presente desse doutorado. Em seguida, a irmã gêmea que o doutorado me trouxe, Amélia. Obrigada irmã, por iniciar e terminar essa jornada juntas, por tantas risadas, bolos e cafés em casa, conversas e força durante esse doutorado. No GPO preciso agradecer ainda aos cachacos, os colombianos que deixavam a vida mais leve e divertida naquele laboratório: Aldair, Alfredo, César, Jonathan, Juan, Karim e Pedro. Obrigada pelas muitas risadas, cervejas e churrascos. Um agradecimento especial ao Aldair, sempre disposto a me ajudar e ao César e Juan, por terem ligado tantas vezes o computador para mim. Agradeço também aos integrantes do GPO: Admilson, José

Ferreira, Mateus, Letícia, Tamara e Vinícius.

Aos meus pais Vera e Eurides, ao meu irmão José, agradeço pelo amor, incentivo, torcida, pelo apoio incondicional, mesmo sem entenderem muito bem o que de fato é um doutorado, entenderem a importância dessa conquista para mim. Obrigada por me ensinarem humildade, força e perseverança. Vocês são o motivo e a razão de tudo na minha vida. Agradeço ainda a minha cunhada Natália e ao meu sobrinho José Filho, que aumentaram essa família durante o doutorado. Aproveito para agradecer a todos os meus familiares pelo amor e por sempre torcerem pelas minhas conquistas. Um agradecimento especial aos meus tios Isabel (Tia Nenem) e Marcos por estarem sempre tão presentes apesar da distância.

Agradeço ao Eduardo pelo carinho, admiração, apoio, ajuda com algumas figuras e referências dessa tese, passeios com o Elvis e louças lavadas. Obrigada pela parceria Edu. Agradeço também ao meu cachorro Elvis, por estar sempre do meu lado em dias e noites que passei a fio no computador.

As amigas do DEP: Raquel, Luciana, Mariana, Cintia e Fabiane. Um agradecimento especial a Raquel com quem dividi o início do doutorado, a primeira casa que morei em São Carlos, as noites acordadas, algumas angústias e a amizade, que mesmo a distância, dura até hoje.

Agradeço ainda aos amigos de longa data que mesmo longe torcem por mim e vibram com as minhas conquistas: Angélica Belasco, Ana Paula Ximenes, Eduardo Bernardes, Cris Pereira, Cris Defalque, Nanna e Manu. Um agradecimento especial ao Marcos e a Amanda, por toda a força e ajuda disponibilizada quando eu precisei. Agradeço também aos amigos de Uberaba e UFTM: Bruno e Gislene, Flávio e Priscila, Victor, Daniel e Priscila Pereira.

Agradeço a Universidade Federal do Triângulo Mineiro pelo apoio financeiro durante o afastamento de dois anos para conclusão do doutorado. Ainda no âmbito da UFTM gostaria de agradecer aos meus alunos por entenderem os momentos em que eu não estava totalmente dedicada, e aos meus colegas de departamento, por me apoiarem e me cobrirem, quando necessário, durante esses anos de doutorado. Um agradecimento especial ao meu amigo Wellington pelo incentivo no início desse projeto e pela torcida e apoio durante todo o processo. Agradeço aos meus amigos de grupo de pesquisa Flávio e Victor também por todo apoio e incentivo.

Aos professores da banca: Prof. Dr. Martin Gómez Ravetti, Profa. Dra. Socorro Rangel, Profa. Dra. Maristela Oliveira dos Santos e Prof. Dr. Pedro Munari agradeço todas as sugestões feitas na qualificação e na defesa que contribuíram para a melhoria e amadurecimento desse trabalho. Um agradecimento especial a Profa. Socorro que me apresentou a Pesquisa Operacional ainda no mestrado e despertou o meu gosto por fazer

pesquisa nessa área.

A todos os professores e técnicos do DEP (Departamento de Engenharia de Produção) da UFSCar que foram essenciais no desenvolvimento desse trabalho. Um agradecimento especial ao Robson e a Regilene, por serem sempre prestativos e solícitos.

Agradeço ainda a todos os colaboradores das empresas de bebidas que eu visitei durante o desenvolvimento desse trabalho. Não consigo citar todos, e em algumas situações esse nomes nem podem ser citados, mas foram essenciais no desenvolvimento dessa pesquisa.

Enfim, agradeço a todos que torceram, incentivaram e vibraram com essa conquista.

Gratidão ao amadurecimento intelectual e pessoal alcançados nesse ciclo.

*“As invenções são, sobretudo, o resultado
de um trabalho teimoso.”*
Santos Dumont

“É preciso, antes de mais nada, querer.”
Amyr Klink

Resumo

Nesta tese é estudado o problema de otimização da programação da produção de bebidas à base de frutas, como néctares e refrescos. Trata-se de um problema de dimensionamento e sequenciamento de lotes em dois estágios de produção principais (preparo da bebida e envase) encontrado em empresas típicas do setor. As principais características desse processo de produção, que o difere de outros problemas de planejamento e programação da produção de bebidas, como por exemplo refrigerantes e cerveja, é a existência de limpezas temporais obrigatórias no processo após determinado tempo sem nenhuma limpeza, e a existência de estoques pulmão (*buffers*) no segundo estágio, que influenciam na realização da sincronia de produção entre os dois estágios. Até onde se pesquisou, este problema de otimização ainda não foi abordado nas literaturas de engenharia de produção e pesquisa operacional. O objetivo desta tese é propor abordagens de solução efetivas para esse problema. Com essas abordagens, avalia-se como otimizar a programação da produção praticada em fábricas características do setor, e com isso, pretende-se contribuir para a literatura de problemas de dimensionamento e sequenciamento de lotes. Propõem-se um novo modelo matemático de otimização para representar o problema integrado de dimensionamento e sequenciamento de lotes dois estágios com limpezas temporais, para a produção de bebidas à base de frutas. Métodos heurísticos baseados na decomposição do problema em estágios e em modelos aproximados para o problema também são propostos. Além disso, são utilizados o método *branch-and-cut* do *solver* CPLEX, e heurísticas *relax-and-fix* e *fix-and-optimize* para resolver o modelo integrado. Explora-se também heurísticas *fix-and-optimize* com busca em vizinhança. Testes computacionais são realizados com instâncias baseadas em dados reais de uma empresa típica do setor. Os resultados mostram que as abordagens de otimização propostas fornecem boas soluções para o problema e que são efetivas para resolvê-lo em situações reais.

Palavras-chave: Programação da produção. Dimensionamento e sequenciamento de lotes. Bebidas à base de frutas. Limpezas temporais. Programação inteira mista. Heurísticas baseadas em modelo. Heurísticas de decomposição. Heurísticas *relax-and-fix*. Heurísticas *fix-and-optimize*. *Fix-and-optimize* com busca em vizinhança.

Abstract

The present work aims to study the problem of the production scheduling optimization of fruit-based beverage, such as nectars and fruit flavoured drinks. This is a lot sizing and scheduling problem in two main production stages (beverage preparation and bottling) found in typical companies of the sector. The main characteristics from this production process, that differs it from other lot sizing and scheduling the production of beverages problems, such as soft drinks and beer, is the existence of mandatory temporal cleanings in the process after a determined time without any cleaning and the presence of inventory buffers in the second stage, which impacts the synchronization between the two production stages. As far as the research is concerned, this optimization problem has not yet been addressed in industrial engineering and operational research literature. The objective of this thesis is to propose approaches with effective solutions for this problem. With these approaches, it is evaluated how to optimize the production scheduling in common factories at the sector, and with that, it is intended to contribute to the lot sizing and scheduling problems literature. A new mathematical model of optimization is proposed to represent the integrated two-stage lot sizing and scheduling problem with temporal cleaning for the fruit-based beverage production. Heuristic methods based on the decomposition of the problem into stages and on approximated models for the problem are also proposed. In addition, the branch-and-cut method from CPLEX solver and the relax-and-fix and fix-and-optimize heuristics are used to solve the integrated model. It also explores fix-and-optimize heuristics with neighborhood search. Computational tests are performed with instances based on real data from a typical company at the sector. The results show that the proposed optimization approaches provide good solutions to the problem and are effective in solving it in real situations.

Keywords: Production scheduling. Lot sizing and scheduling. Fruit-based beverage. Temporal cleaning. Mixed integer programming. Heuristics based in model. Decomposition heuristics. Relax-and-fix heuristics. Fix-and-optimize heuristics. Fix-and-optimize with neighborhood search heuristics.

Lista de ilustrações

Figura 1 – Diferenças entre a sincronia: refrigerantes e cervejas comparados com bebidas à base de frutas. Fonte: a autora.	49
Figura 2 – Exemplo de embalagem de refresco. Fonte: a autora.	52
Figura 3 – Exemplo de embalagem de néctar. Fonte: a autora.	52
Figura 4 – Exemplo de embalagem de suco. Fonte: a autora.	53
Figura 5 – Fluxograma do processamento via envase asséptico de bebida à base de fruta. Fonte: Adaptado de (VENTURINI, 2010).	53
Figura 6 – Processo geral de produção de bebidas à base de frutas. Fonte: a autora.	56
Figura 7 – Processo de preparo da bebidas à base de frutas. Fonte: a autora. . . .	56
Figura 8 – Processo simplificado de produção de bebidas à base de frutas. Fonte: a autora.	59
Figura 9 – Exemplar ilustrativo de uma programação da produção factível. Fonte: a autora.	60
Figura 10 – Comparação dos subperíodos do GLSP com os lotes do MDSL-2E-LT.	72
Figura 11 – Representação de como as variáveis Y_{mjto} e Z_{mijt} capturam o sequenciamento dos itens e a produção dos lotes. Fonte: a autora.	75
Figura 12 – Limpezas do início do período. Restrições (4.15) e (4.16). Fonte: a autora.	76
Figura 13 – Ordenação e não sobreposição dos lotes. Restrições (4.19) e (4.20). Fonte: a autora.	77
Figura 14 – Troca entre itens. Restrições (4.21) e (4.22). Fonte: a autora.	77
Figura 15 – Sincronia entre os estágios. Restrições (4.23) e (4.24). Fonte: a autora. .	78
Figura 16 – Ilustração da identificação da necessidade de inclusão da primeira limpeza temporal $k = 1$ do tanque preparatório m para os itens i e j . Fonte: a autora.	80
Figura 17 – Programação infactível do tanque preparatório representando a necessidade de uma limpeza temporal k . Fonte: a autora.	82
Figura 18 – Representação de uma programação factível com a inclusão da limpeza temporal. Fonte: a autora.	83
Figura 19 – Teste 1 - Gráfico de Gantt da solução do exemplar 1 para $t = 1$ e $t = 2$ para $m = 1$. Fonte: a autora.	90
Figura 20 – Teste 4 - Gráfico de Gantt da solução do exemplar 2 para $t = 1$ e $t = 2$ para $m = 1$. Fonte: a autora.	92
Figura 21 – Fluxograma das heurísticas de decomposição. Fonte: a autora.	96
Figura 22 – Exemplo de como as heurísticas HE1 e HE2 resolvem o problema. Fonte: a autora.	97
Figura 23 – Resumo das variações das heurísticas HE1 e HE2. Fonte: a autora. . . .	103

Figura 24 – Representação do tempo de produção contínua do item j . Fonte: a autora.	113
Figura 25 – Representação de uma vizinhança nas heurísticas <i>fix-and-optimize</i> .	121
Figura 26 – Liberação da partição S_k e definição do subproblema.	122
Figura 27 – Solução após otimização do subproblema obtido com a partição S_k .	123
Figura 28 – Liberação da partição S_{k+1} e definição do subproblema.	123
Figura 29 – Solução após otimização do subproblema obtido com a partição S_{k+1} .	124
Figura 30 – Liberação da partição S_{k+2} e definição do subproblema.	125
Figura 31 – Perfis de desempenho das estratégias CPLEX. Fonte: a autora.	147
Figura 32 – Valores dos limitantes ao longo do tempo durante a resolução da instância S2 pela estratégia CPX12. Fonte: a autora.	149
Figura 33 – Valores dos limitantes ao longo do tempo durante a resolução da instância R1 pela estratégia CPX12. Fonte: a autora.	149
Figura 34 – Perfil de desempenho da função objetivo das heurísticas <i>relax-and-fix</i> . Fonte: a autora.	155
Figura 35 – Perfil de desempenho da função objetivo nas heurísticas <i>fix-and-optimize</i> com solução inicial da CPX12. Fonte: a autora.	157
Figura 36 – Perfil de desempenho da função objetivo nas heurísticas <i>fix-and-optimize</i> com solução inicial da RFX4. Fonte: a autora.	158
Figura 37 – Perfil de desempenho da função objetivo nas heurísticas <i>fix-and-optimize</i> com solução inicial da HE1B-DS. Fonte: a autora.	159
Figura 38 – Perfil de desempenho da função objetivo dos melhores métodos de solução para o problema. Fonte: a autora.	168
Figura 39 – Perfil de desempenho do tempo dos melhores métodos de solução para o problema. Fonte: a autora.	168
Figura 40 – Comparação das limpezas temporais dos tanques preparatórios nas soluções obtidas pela FXOV2 com $Cl_t = 1$ e $Cl_t = 100$.	172
Figura 41 – Comparação das limpezas temporais das linhas nas soluções obtidas pela FXOV2 com $Cl_t = 1$ e $Cl_t = 100$.	172
Figura 42 – Comparação dos atrasos, estoques e trocas obtidos com a FXOV2 quando $Cl_t = 1$ e $Cl_t = 100$.	173
Figura 43 – Comparação dos estoques, atrasos e trocas obtidos com os modelos MDSL-2E-LT e MDSL-2E.	177
Figura 44 – Processo de produção de bebida mista de fruta. Fonte: a autora.	201
Figura 45 – Processo produtivo de refrigerantes. Fonte: Adaptado de Ferreira <i>et al.</i> (2011).	220
Figura 46 – Exemplo de programação sincronizada e não sincronizada. Fonte: Adaptado de (FERREIRA; MORABITO; RANGEL, 2009).	221

Lista de tabelas

Tabela 1 – Resumo das características dos trabalhos revisados e discutidos nesta seção.	41
Tabela 2 – Abreviação da nomenclatura dos métodos de solução apresentados na Tabela 1	43
Tabela 6 – Dimensão de cada conjunto de restrição do modelo (4.4)-(4.43).	85
Tabela 7 – Dados dos exemplares ilustrativos.	87
Tabela 8 – Demanda em unidades do exemplar ilustrativo.	87
Tabela 9 – Lotes máximo e mínimo em litros do exemplar ilustrativo.	87
Tabela 10 – Custo de troca do item i para o item j do exemplar ilustrativo.	87
Tabela 11 – Tempo de troca em minutos do item i para o item j do exemplar ilustrativo (TC_{ij}^I / TC_{ij}^{II}).	87
Tabela 12 – Sequência, lotes de produção e limpezas temporais da solução da instância 1.	88
Tabela 13 – Variáveis de tempo referentes ao Gantt da Figura 19.	89
Tabela 14 – Sequência, lotes de produção e limpezas temporais da solução da instância 2.	91
Tabela 15 – Variáveis de tempo referentes ao Gantt da Figura 20	93
Tabela 16 – Sumário das heurísticas propostas - HE1.	110
Tabela 17 – Sumário das heurísticas propostas - HE2.	110
Tabela 18 – Resumo dos modelos desenvolvidos para o problema de programação da produção de bebidas à base de frutas.	115
Tabela 19 – Características das heurísticas <i>relax-and-fix</i> propostas.	118
Tabela 20 – Características das heurísticas <i>fix-and-optimize</i> propostas.	120
Tabela 21 – Métodos de solução avaliados nos experimentos computacionais.	137
Tabela 22 – Quantidade de elementos dos conjuntos M , T , J , O_{mt} , Q_{mt}^I e Q_{mt}^{II} para cada instância.	140
Tabela 24 – Dimensão das instâncias utilizadas no modelo MDSL-2E-LT.	146
Tabela 25 – Resultados com o CPLEX no modo <i>default</i> e com a estratégia CPX12.	148
Tabela 26 – Detalhes dos resultados da Tabela 25 para as instâncias R1, R2, R3 e R4.	148
Tabela 27 – Detalhes das soluções obtidas pela estratégia CPX12.	150
Tabela 28 – Resultados obtidos pelas heurísticas HE1-DS.	151
Tabela 29 – Resultados obtidos pelas heurísticas HE2-DS.	151
Tabela 30 – Resultado das heurísticas baseadas em modelo: Subestimada (H-SUB) e Superestimada (H-SUP)	152
Tabela 31 – Soluções obtidas com a H1EB-DS e com a H-SUP com tempo limite de 20 minutos.	153

Tabela 32 – Resultados do modelo MDSL-2E-LT via CPX12 com solução inicial dada pelas heurísticas H1EB-DS e H-SUP.	154
Tabela 33 – Resultados da heurística <i>relax-and-fix</i> RFX4 e estratégia CPX12.	156
Tabela 34 – Resultados das heurísticas <i>fix-and-optimize</i> com aleatoriedade e da <i>fix-and-optimize</i> com princípios de VNDS.	161
Tabela 35 – Dimensão das instâncias geradas utilizadas no modelo MDSL-2E-LT.	164
Tabela 36 – Detalhes das soluções obtidas pela estratégia CPX12 para as instâncias geradas aleatoriamente.	165
Tabela 37 – Valores das funções objetivos (u.m.) obtidas pelos melhores métodos para todas as instâncias.	169
Tabela 38 – Tempos despendidos pelos melhores métodos para obtenção da solução.	170
Tabela 39 – Custos de atraso e estoque obtidos pela FXOV2 para as instâncias com capacidade original e capacidade reduzida.	174
Tabela 40 – Comparação do número de variáveis e restrições para os modelos MDSL-2E e MDSL-2E-LT.	175
Tabela 41 – Comparação entre os resultados obtidos por MDSL-2E e MDSL-2E-LT.	176
Tabela 42 – Resultados da heurística HE2B-DS+FXOV2 comparados com a solução da empresa.	177
Tabela 43 – Programação da produção obtida com a heurística HE2B-DS+FXOV2 para o primeiro período da instância R1.	179
Tabela 44 – Programação da produção utilizada pela empresa para o primeiro período da instância R1.	179
Tabela 45 – Descrição dos valores em litros dos <i>kits</i> e dos lotes máximo e mínimo.	202
Tabela 46 – Velocidade de produção dos pasteurizadores.	202
Tabela 47 – Velocidades nominais das máquinas de envase na primeira visita realizada a fábrica.	203
Tabela 48 – Velocidades nominais das máquinas de envase na última visita realizada a fábrica.	203
Tabela 49 – Resumo dos trabalhos relacionados ao dimensionamento e sequenciamento de lotes na indústria de bebidas.	227
Tabela 50 – Estratégias CPLEX testadas.	235
Tabela 51 – Instâncias Testadas.	238
Tabela 52 – Soluções das heurísticas HE1-IS e HE2-IS e do modelo MDSL-2E-LT.	239
Tabela 53 – Resultados da heurística HE2A-IS comparados com a empresa.	242
Tabela 54 – Soluções estratégias CPLEX - função objetivo (u.m.).	243
Tabela 55 – Soluções estratégias CPLEX - <i>gap</i> (%).	243
Tabela 56 – Soluções heurísticas <i>relax-and-fix</i> - função objetivo (u.m.).	244
Tabela 57 – Soluções heurísticas <i>relax-and-fix</i> - <i>gap</i> (%).	244
Tabela 58 – Tempos computacionais heurísticas <i>relax-and-fix</i> - tempo em segundos.	245

Tabela 59 – Soluções heurísticas <i>fix-and-optimize</i> com solução inicial fornecida por CPX12 - função objetivo (u.m.).	245
Tabela 60 – Soluções heurísticas <i>fix-and-optimize</i> com solução inicial fornecida por CPX12 - <i>gap</i> (%).	245
Tabela 61 – Tempos computacionais heurísticas <i>fix-and-optimize</i> com solução inicial fornecida por CPX12 - tempo em segundos.	246
Tabela 62 – Soluções heurísticas <i>fix-and-optimize</i> com solução inicial fornecida por RFX4 - função objetivo (u.m.).	246
Tabela 63 – Soluções heurísticas <i>fix-and-optimize</i> com solução inicial fornecida por RFX14 - <i>gap</i> (%).	246
Tabela 64 – Tempos computacionais heurísticas <i>fix-and-optimize</i> com solução inicial fornecida por RFX4 - tempo em segundos.	247
Tabela 65 – Soluções heurísticas <i>fix-and-optimize</i> com solução inicial fornecida por HE1B-DS - função objetivo (u.m.).	247
Tabela 66 – Soluções heurísticas <i>fix-and-optimize</i> com solução inicial fornecida por HE1B-DS - <i>gap</i> (%).	247
Tabela 67 – Tempos computacionais heurísticas <i>fix-and-optimize</i> com solução inicial fornecida por HE1B-DS - tempo em segundos.	248
Tabela 68 – Detalhes das soluções obtidas pela estratégia CPX12 na resolução do modelo MDSL-2E-LT para todas as instâncias.	249
Tabela 69 – Detalhes das soluções obtidas pela estratégia RFX4 para todas as instâncias.	250
Tabela 70 – Detalhes das soluções obtidas pela estratégia HE1B-DS para todas as instâncias.	251
Tabela 71 – Detalhes das soluções obtidas pela estratégia H-SUP para todas as instâncias.	252
Tabela 72 – Detalhes das soluções obtidas pela estratégia CPX12+FXO12 para todas as instâncias.	253
Tabela 73 – Detalhes das soluções obtidas pela estratégia RFX4+FXO3 para todas as instâncias.	254
Tabela 74 – Detalhes das soluções obtidas pela estratégia HE1B-DS+FXO3 para todas as instâncias.	255
Tabela 75 – Detalhes das soluções obtidas pela estratégia FXOV1_T-M-J para todas as instâncias.	256
Tabela 76 – Detalhes das soluções obtidas pela estratégia FXOV2 para todas as instâncias.	257
Tabela 77 – Detalhes das soluções obtidas pela estratégia FXO+VNDS para todas as instâncias.	258

Lista de abreviaturas e siglas

ABIR	Associação Brasileira das Indústrias de Refrigerantes e Bebidas não Alcoólicas
CIP	<i>Clear in Place</i>
CLSD	<i>Capacitated lot sizing with sequence-dependent setups</i>
CLSP	<i>Capacitated Lot Sizing Problem</i>
CPX	Estratégia CPLEX
CSLP	<i>Continuous Setup Lot Sizing Problem</i>
DLSP	<i>Discrete Lot Sizing and Scheduling Problem</i>
DS	Dependente da Sequência
EOQ	<i>Economic Order Quantity</i>
ELSP	<i>Economic Lot Scheduling Problem</i>
FXO	Heurística <i>fix-and-optimize</i>
GLSP	<i>General Lot Sizing and Scheduling Problem</i>
HE1	Heurísticas Estágio I
HE2	Heurísticas Estágio II
HE1A	Heurísticas Estágio I tipo A
HE1A-DS	Heurísticas Estágio I tipo A caso dependente da sequência
HE1A-IS	Heurísticas Estágio I tipo A caso independente da sequência
HE1B	Heurísticas Estágio I tipo B
HE1B-DS	Heurísticas Estágio I tipo B caso dependente da sequência
HE1B-IS	Heurísticas Estágio I tipo B caso independente da sequência
HE2A	Heurísticas Estágio II tipo A
HE2A-DS	Heurísticas Estágio II tipo A caso dependente da sequência
HE2A-IS	Heurísticas Estágio II tipo A caso independente da sequência

HE2B	Heurísticas Estágio II tipo B
HE2B-DS	Heurísticas Estágio II tipo B caso dependente da sequência
HE2B-IS	Heurísticas Estágio II tipo B caso independente da sequência
H-SUB	Heurística Subestimada
H-SUP	Heurística Superestimada
IBGE	Instituto Brasileiro de Geografia e Estatística
IS	Independente da Sequência
MDL-L	Modelo de Dimensionamento de Lotes para a Linha
MDL-P	Modelo de Dimensionamento de Lotes do Preparo
MDSL-L	Modelo de Dimensionamento e Sequenciamento de Lotes da Linha
MDSL-P	Modelo de Dimensionamento e Sequenciamento de Lotes do Preparo
MDSL-2E	Modelo Integrado de Dimensionamento e Sequenciamento de Lotes Dois Estágios
MDSL-2E-LT	Modelo Integrado de Dimensionamento e Sequenciamento de Lotes Dois Estágios com Limpezas Temporais
MIP	<i>Mixed Integer Programming</i>
MR1-O	Modelo Relaxado 1 - Otimista
MS2-P	Modelo Relaxado 2 - Pessimista
PIA	Pesquisa Industrial Anual
PCP	Planejamento e Controle da Produção
PLSP	<i>Proportional Lot Sizing and Scheduling Problem</i>
PCV	Problema do Caixeiro Viajante
PCVA	Problema do Caixeiro Viajante Assimétrico
RFX	Heurística <i>relax-and-fix</i>
INS	iterative neighborhood search
FXOV1	<i>fix-and-optimize</i> com busca em vizinhança 1
FXOV2	<i>fix-and-optimize</i> com busca em vizinhança 2

FXO-VNDS *fix-and-optimize* com princípios de VNDS

VNS *Variable Neighborhood Search*

VNDS *Variable Neighborhood Decomposition Search*

Sumário

1	INTRODUÇÃO	29
1.1	Objetivos	32
1.2	Justificativas e contribuições	32
1.3	Método de pesquisa	33
1.4	Organização do texto	34
2	REVISÃO DA LITERATURA RELACIONADA	35
2.1	Problemas de dimensionamento e sequenciamento de lotes: características, aplicações e métodos de solução	36
2.2	Planejamento e programação na indústria de bebidas	47
3	DESCRIÇÃO DO PROCESSO DE PRODUÇÃO DE BEBIDA À BASE DE FRUTAS	51
3.1	Processo de fabricação de bebidas à base de frutas	54
3.1.1	Limpezas	58
3.1.2	Sincronia entre os estágios de produção	61
3.2	Definição do problema	63
4	MODELAGEM MATEMÁTICA	67
4.1	Modelo Integrado de Dimensionamento e Sequenciamento de Lotes Dois Estágios com Limpezas Temporais (MDSL-2E-LT)	67
4.1.1	Exemplos Ilustrativos	86
4.1.2	Modelo Integrado de Dimensionamento e Sequenciamento de Lotes Dois Estágios (MDSL-2E)	93
5	MÉTODOS HEURÍSTICOS DE PROGRAMAÇÃO MATEMÁTICA PARA O PROBLEMA	95
5.1	Heurísticas de decomposição do problema em estágios	96
5.1.1	Heurísticas Estágio I - HE1A-DS e HE1B-DS	103
5.1.2	Heurísticas Estágio I - HE1A-IS e HE1B-IS	105
5.1.3	Heurísticas Estágio II - HE2A-DS e HE2B-DS	106
5.1.4	Heurísticas Estágio II - HE2A-IS e HE2B-IS	108
5.2	Heurísticas baseadas em modelos relaxados	110
5.2.1	Modelo Relaxado 1 - Otimista (MR1-O)	111
5.2.2	Modelo Relaxado 2 - Pessimista (MR2-P)	113
5.3	Heurísticas <i>relax-and-fix</i>	115

5.4	Heurísticas de melhoria <i>fix-and-optimize</i>	118
5.4.1	Heurísticas <i>fix-and-optimize</i>	118
5.4.2	Heurísticas <i>fix-and-optimize</i> com busca em vizinhanças	121
6	EXPERIMENTOS COMPUTACIONAIS	137
6.1	Descrição das instâncias	138
6.1.1	Instâncias baseadas em dados reais	138
6.1.2	Instâncias geradas aleatoriamente	140
6.2	Crítérios de avaliação dos resultados	144
6.3	Testes computacionais: parte I	145
6.3.1	Ajuste de parâmetros CPLEX no modelo MDSL-2E-LT	145
6.3.2	Avaliação das heurísticas de decomposição	150
6.3.3	Avaliação das heurísticas baseadas nos modelos relaxados	152
6.3.4	Avaliação dos métodos H1EB-DS+CPX12 e H-SUP+CPX12	153
6.3.5	Avaliação das heurísticas <i>relax-and-fix</i>	154
6.3.6	Avaliação das heurísticas <i>fix-and-optimize</i>	156
6.3.7	Avaliação das heurísticas <i>fix-and-optimize</i> com busca em vizinhança	160
6.3.8	Conclusão	162
6.4	Testes computacionais: parte II	163
6.5	Avaliação de cenários	171
6.5.1	Cenário 1: Alto custo da limpeza temporal	171
6.5.2	Cenário 2: Capacidade restrita	172
6.5.3	Cenário 3: Modelo MDSL-2E sem consideração da limpeza temporal	174
6.6	Comparação do melhor método de solução com a solução da empresa	177
7	CONSIDERAÇÕES FINAIS	181
	REFERÊNCIAS	187
	APÊNDICE A – FÁBRICA A	201
	APÊNDICE B – TRABALHOS SOBRE DIMENSIONAMENTO E SEQUENCIAMENTO DE LOTES	207
	APÊNDICE C – PROBLEMAS DE PROGRAMAÇÃO DA PRODUÇÃO NA INDÚSTRIA DE BEBIDAS	219
C.1	Processo de Produção de Bebidas	219
C.2	Trabalhos encontrados da literatura	222
	APÊNDICE D – MODELO MDSL-2E-LT COMPLETO	229

APÊNDICE E – AJUSTE DE PARÂMETROS DO CPLEX	233
APÊNDICE F – EXPERIMENTOS COMPUTACIONAIS PARA AS HEURÍSTICAS DE DECOMPOSIÇÃO PARA CASO INDEPENDENTE DA SEQUÊNCIA	237
APÊNDICE G – TABELAS DOS RESULTADOS COMPUTACIO- NAIS	243

1 Introdução

De acordo com informações da PIA (Pesquisa Industrial Anual) Instituto Brasileiro de Geografia e Estatística (IBGE, 2015), o setor de fabricação de bebidas responde por aproximadamente 4% do valor adicionado da indústria de transformação brasileira, e emprega cerca de 144 mil pessoas no mercado formal, o que corresponde a 2,2% do pessoal ocupado na indústria de transformação do Brasil. Nos últimos cinco anos, uma fatia do setor de bebidas que vem crescendo é a de bebidas não alcoólicas (refrigerantes, águas minerais, néctares, sucos concentrados, energéticos, isotônicos, chás e refrescos). Segundo a ABIR (2015), de 2010 para 2013 houve um aumento de 9,5% no consumo de bebidas não alcoólicas no Brasil.

O crescimento no setor de bebidas não alcoólicas no país é explicado, em parte, porque o Brasil vem seguindo uma tendência mundial de crescimento do consumo de alimentos funcionais (alimentos ou ingredientes que produzem efeitos benéficos à saúde, além de suas funções nutricionais básicas). A venda de produtos considerados saudáveis movimentou cerca de U\$ 27 bilhões em 2015 e só as bebidas devem movimentar U\$ 11 milhões até 2020 (EUROMONITOR, 2015). Tais produtos começaram a ganhar atenção dos consumidores com a crescente preocupação com bem-estar e saúde. Na área de bebidas, a categoria dos funcionais cresce a um ritmo cinco vezes maior que a dos refrigerantes tradicionais. Com base nessa tendência de valorização de alimentos funcionais, a categoria de sucos prontos foi beneficiada. O mercado de sucos e néctares foi um dos que mais cresceu e continua em ascensão, com alta de 100% entre os anos de 2004 a 2008. Este segmento também influenciou no aumento do consumo de refrescos prontos para beber (refrescos com sabores de frutas, bebidas à base de soja, bebidas à base de guaraná e água de coco) (VENTURINI, 2010). A tendência é que o consumo de néctares e refrescos continue aumentando. Além do crescimento desse nicho de mercado, cresce também a competitividade entre as empresas desse setor.

A composição do custo de um produto para uma empresa envolve diversas parcelas, como custos com matéria-prima, impostos, logísticos (custos de transporte), *marketing* e custos relacionados ao processo de produção em si, entre outros. Assim, produzir de forma otimizada é essencial para se manter no mercado, sobretudo em épocas de crise financeira, com inflação e juros altos, como é o caso do Brasil recentemente (IBGE, 2015). Por conseguinte, um desafio para essas empresas é utilizar programas de produção efetivos. Isto porque estes programas, muitas vezes estabelecidos manualmente, envolvem a produção de diversos itens, em várias máquinas, com recursos escassos para a produção, a fim de atender às demandas dinâmicas dos produtos, com o objetivo de reduzir os custos de produção, como, por exemplo, custos de preparação, estoque e atraso da produção.

De maneira geral, a produção de bebidas possui dois estágios principais que são: o preparo da bebida e seu envase. No estágio de preparo, são misturados os ingredientes à água para produzir a bebida. Depois de pronta, a bebida é enviada para o setor de envase, onde é colocada em embalagens específicas. A produção de cada tipo de bebida possui características distintas. Por exemplo, na produção de refrigerantes e de cervejas, os tempos e custos de *setups* de produção (tempos e custos despendidos durante a troca de produtos na produção) são dependentes da sequência dos lotes de produção. Porém, nos refrigerantes, o xarope pronto já pode ser enviado imediatamente para as linhas de envase, enquanto que na produção de cervejas, a bebida pode ficar até 40 dias armazenada em tanques, para maturação, antes de ser enviada para as linhas de envase. Devido aos dois estágios de produção existentes na fabricação de bebidas, a consideração da sincronia entre esses estágios na programação da produção é imprescindível para obtenção de um programa de produção factível. Para mais detalhes sobre esses processos de produção, veja, por exemplo, Toledo et al. (2009), Ferreira, Morabito e Rangel (2009) e Baldo et al. (2014).

Na literatura, alguns autores vem tratando de problemas de planejamento e programação da produção na indústria de bebidas principalmente nos níveis tático e operacional, em especial de refrigerantes e cervejas (TOLEDO et al., 2007; FERREIRA; MORABITO; RANGEL, 2009; FERREIRA; MORABITO; RANGEL, 2010; DEFALQUE; RANGEL; FERREIRA, 2011; FERREIRA et al., 2012; TOLEDO et al., 2014; MALDONADO; RANGEL; FERREIRA, 2014; BALDO et al., 2014; TOLEDO et al., 2015) e nas referências neles citadas. Um dos principais focos desses trabalhos é o desenvolvimento de modelos matemáticos e métodos de soluções para os problemas de dimensionamento e sequenciamento de lotes de produção dessas indústrias. Alguns trabalhos também tratam a programação da produção de bebidas no nível estratégico e tático (GUIMARÃES; KLABJAN; ALMADA-LOBO, 2012) e associados com custos logísticos (SEL; BILGEN, 2014), mas são menos frequentes.

Estudando o setor de produção de bebidas à base de frutas (sucos, néctares e refrescos), percebe-se que existem características diferentes da produção dessas bebidas em relação à produção de refrigerantes e cervejas. Na tecnologia de produção de bebidas à base de frutas existem, por exemplo, além da limpeza para troca de item, a exigência de limpezas no maquinário após determinado período de tempo decorrido desde a última limpeza (limpeza temporal), e a presença de tanques pulmão (*buffers*) no segundo estágio (PAGLIARUSSI; MORABITO; SANTOS, 2017; LEITE, 2008). As limpezas temporais são obrigatórias em intervalos de tempo sem limpeza em ambos os estágios, independente se haverá ou não troca de sabor. Tanques pulmão garantem que um novo lote pode ser preparado enquanto o lote atual está sendo envasado, ou seja, garantem um lote de folga entre os estágios. No Capítulo 3 esse processo de produção está descrito em detalhes.

Essas características próprias e específicas da produção de bebidas à base de frutas influenciam diretamente na sincronia entre os estágios de produção, o que faz com que a programação da produção, nesse caso, não seja simplesmente uma extensão dos trabalhos de refrigerantes e cervejas. Dois trabalhos da literatura (PAGLIARUSSI; MORABITO; SANTOS, 2017; LEITE, 2008) trataram esse problema de programação da produção de bebidas à base de frutas. Porém, eles consideram apenas um estágio de produção (o estágio considerado o gargalo de produção), o que pode ser razoável para o caso de certas empresas, mas não para o processo de produção de empresas deste setor em geral.

O presente trabalho foca na otimização da programação da produção de bebidas à base de frutas em processos de produção de dois estágios com limpezas temporais no nível operacional. O problema se resume em definir os tamanhos dos lotes de produção e o sequenciamento desses lotes para atender a demanda previamente conhecida, otimizando custos, como por exemplo, custos de falta e de excesso de produção, e levando em consideração as limpezas temporais e os tanques pulmão, pois estes afetam diretamente na sincronia entre os dois estágios de produção. Neste trabalho, esse problema de programação da produção é abordado sob o ponto de vista de problemas integrados de dimensionamento e sequenciamento de lotes, pois a determinação do tamanho dos lotes influencia diretamente nas limpezas temporais, que por sua vez influenciam na sincronia entre os estágios e consequentemente, no consumo da capacidade de produção disponível. Uma revisão bibliográfica sobre problemas integrados de dimensionamento e sequenciamento de lotes é apresentada no Capítulo 2.

Convém observar que se os tamanhos dos lotes desse problema fossem pré-definidos e não influenciassem nas limpezas temporais, o problema poderia também ser abordado como um *flowshop* permutacional, em que todos os *jobs* (lotes) passam por todas as máquinas e seguem a mesma sequência (PINEDO, 2008). Mas esse não é o caso do problema de programação da produção de bebidas à base de frutas.

Durante essa pesquisa foram visitadas 5 fábricas produtoras de bebidas (3 de uma mesma franquia multinacional), para melhor entendimento e contextualização do problema. Uma dessas empresas é especialista na produção de sucos, néctares e refrescos. As outras produzem também outros tipos de bebidas, além de bebidas à base de frutas. Dados da produção de uma dessas empresas, que aceitou colaborar com essa pesquisa, foram coletados para avaliação em testes computacionais das abordagens aqui propostas. Nesta tese, investiga-se a possibilidade de otimização de programas de produção encontrados na realidade deste setor. Desta maneira, esta pesquisa contribui para o desenvolvimento de ferramentas de apoio ao tomador de decisão, que tem o desafio de definir semanalmente os programas de produção de um nicho de mercado importante, e que torna-se cada vez mais crescente e competitivo.

1.1 Objetivos

O principal objetivo desse trabalho é pesquisar, desenvolver e aplicar abordagens de otimização para o problema de programação da produção de bebidas à base de frutas em processos de produção de dois estágios com limpezas temporais. Para alcançar esse objetivo principal, as atividades/etapas a serem desenvolvidas são:

- Estudar o processo de produção de bebidas à base de frutas em dois estágios de produção. Para um estudo aprofundado desse processo produtivo, foram realizadas diversas visitas técnicas a fábricas deste setor.
- Propor novos modelos matemáticos de otimização combinatória para o problema de programação da produção das bebidas à base de frutas no nível operacional. Isso significa desenvolver modelos de dimensionamento e sequenciamento de lotes de produção em dois estágios, preparo e envase da bebida, com sincronia entre os dois estágios de produção e que capturem todas as características da tecnologia de produção de bebidas à base de frutas que influenciam nas decisões da programação da produção.
- Desenvolver métodos de solução heurísticos, tais como, heurísticas construtivas de decomposição que resolvam o problema desagregando os dois estágios, e heurísticas baseadas em MIP (*mixed integer programming model*) do tipo *relax-and-fix* e *fix-and-optimize*, capazes de tratar problemas de tamanhos realistas em tempos computacionais aceitáveis.
- Mostrar que essas abordagens de otimização propostas são efetivas para resolver o problema de programação da produção de bebidas à base de frutas na prática.

1.2 Justificativas e contribuições

Dado o cenário atual do mercado de bebidas à base de frutas, uma justificativa para este trabalho é colaborar para o desenvolvimento de sistemas de apoio à tomada de decisão para a programação da produção do setor de bebidas e para outros processos de produção que envolvam, por exemplo, sincronia entre dois estágios de produção e limpezas temporais obrigatórias. Nas visitas realizadas a empresas do setor foi possível observar que nenhuma delas possui um sistema baseado em técnicas de otimização para apoio a elaboração da programação da produção. Essa programação é realizada manualmente em planilhas eletrônicas, com base na experiência do tomador de decisão e sem o auxílio de nenhum sistema computacional de otimização. Os modelos e métodos de solução aqui propostos têm potencial para colaborar na otimização dos custos envolvidos na produção, agilizar e sistematizar a programação da produção e ainda permitir análises de cenários

e simulações com diferentes conjuntos de dados (previsões de demandas, limitações de capacidade, números variados de máquinas, etc), para que o tomador de decisão possa avaliar melhor as decisões a serem tomadas.

Analisando a literatura científica de problemas de programação da produção de bebidas, percebe-se que para refrigerantes existem vários trabalhos. Nesta pesquisa foram encontrados 12 artigos sobre o assunto (vide Apêndice C). Porém, para as bebidas à base de frutas, a literatura ainda é escassa, tendo sido encontrados apenas dois trabalhos que consideram o problema de maneira simplificada, com apenas um estágio de produção. Portanto, este trabalho pode contribuir para o enriquecimento do estado da arte da literatura relacionada.

Por outro lado, os problemas de dimensionamento e sequenciamento de lotes vêm sendo bem estudados nos últimos anos, como mostram alguns trabalhos recentes de revisão de literatura: Jans e Degraeve (2008), Glock, Grosse e Ries (2013) e Copil et al. (2017). Tratam-se de problemas pertencentes à classe NP-difícil (BITRAN; YANASSE, 1992) que vêm sendo cada vez mais aplicados à solução de situações práticas. Na revisão de literatura, não foram encontrados trabalhos que explorem todas as características encontradas no problema aqui estudado (limpezas temporais e lote de folga entre os estágios), e que podem ser adaptáveis ou aplicáveis a outros processos de produção diferentes de bebidas à base de frutas, como na produção de leite, por exemplo. Estas observações reforçam a originalidade e relevância científica do presente estudo.

Desta maneira, as contribuições desse trabalho são de sentido prático e teórico. Na prática, com a pesquisa e proposta de abordagens de otimização da programação da produção de bebidas à base de frutas, que pode auxiliar os tomadores de decisão em empresas do setor. No sentido teórico, com o estudo de um novo problema de dimensionamento e sequenciamento de lotes com características específicas, com a formulação de um novo modelo matemático de otimização combinatória que representa fielmente o problema, e com o desenvolvimento de métodos de solução heurísticos para resolução do problema e do modelo matemático proposto.

1.3 Método de pesquisa

O método de pesquisa utilizado nesse trabalho é a Modelagem/Simulação, que envolve técnicas matemáticas e computacionais para descrever ou simular o funcionamento de um sistema. Esse método é fundamentado no pressuposto de que podemos construir modelos que expliquem total ou parcialmente o comportamento dos processos operacionais de ambientes reais, e que podemos também capturar os problemas de tomada de decisões envolvidos nestes processos.

O processo de produção estudado nesse trabalho está definido em detalhes no

Capítulo 3 e o problema de pesquisa envolve determinar programas de produção para esse processo da indústria de bebidas à base de frutas, de maneira a otimizar a programação da produção. A modelagem foi escolhida como método de pesquisa, pois um objetivo é desenvolver um modelo matemático que apresente apropriadamente o problema real, além de propor e aplicar métodos de solução efetivos para obter programas de produção otimizados e factíveis na prática. Além disso, segundo [Bertrand e Fransoo \(2002\)](#) e [Morabito e Pureza \(2012\)](#), trata-se de uma pesquisa empírica quantitativa normativa, pois propõe principalmente criar modelos matemáticos que se ajustem às observações de um sistema real e desenvolver políticas, estratégias e ações que melhorem a situação corrente.

O primeiro passo de desenvolvimento desse trabalho envolve a definição do problema abordado com a apresentação dos dados e informações relevantes relacionadas e a caracterização desse problema segundo a literatura ([MORABITO; PUREZA, 2012](#)). Em seguida, deve-se propor um modelo matemático para representar o problema, que deve se adequar corretamente às considerações particulares do sistema de produção em questão. Após esta etapa é utilizado um pacote de otimização para a verificação e validação do modelo e resolução de instâncias, o que permite verificar a necessidade de efetuar ajustes no modelo e a consideração de métodos de solução mais eficazes. De posse do modelo validado e de métodos de solução para resolvê-lo, testes com dados reais da fábrica são realizados. Além disso, são realizados também testes com dados gerados aleatoriamente para verificar a robustez do modelo e dos métodos de solução propostos, além de se explorar diversos cenários alternativos para o problema. Por fim, os resultados destes testes realizados são apresentados, analisados e comparados com as informações relacionadas às práticas da fábrica estudada.

1.4 Organização do texto

Este texto está estruturado da seguinte maneira: no Capítulo 2 é apresentada uma revisão bibliográfica sobre problemas integrados de dimensionamento e sequenciamento de lotes. No Capítulo 3 é apresentada a descrição do processo de produção de bebidas à base de frutas, as peculiaridades desse processo e o delineamento do problema aqui estudado. O modelo matemático proposto para resolver o problema é apresentado no Capítulo 4. Métodos de solução heurísticos para o problema de programação da produção de bebidas à base de frutas e métodos de soluções heurísticos para resolver o modelo matemático integrado proposto são apresentados no Capítulo 5. No Capítulo 6 são apresentados os resultados de experimentos computacionais para o modelo matemático e para os métodos heurísticos. Por fim, no Capítulo 7 são apresentadas as conclusões e perspectivas de trabalhos futuros.

2 Revisão da literatura relacionada

As atividades de Planejamento e Controle da Produção (PCP) envolvem uma série de decisões com o objetivo de definir o que, quanto e quando produzir, comprar e entregar, além de quem e/ou onde e/ou como produzir. Essas decisões são imprescindíveis para o funcionamento de qualquer empresa e estão diretamente ligadas a decisões cruciais, tais como: subsidiar decisões na elaboração de contratos de fornecimento, planejamento da capacidade de médio prazo, política de horas extras e de bancos de horas, contratações e demissões no médio prazo, aquisição de novos equipamentos, dentre outras (FERNANDES; GODINHO, 2010).

A divisão entre as decisões de planejamento e controle da produção não são muito claras na teoria e na prática. Entretanto há algumas características gerais que ajudam a distinguir os dois. Planejamento é a formalização do que se pretende que aconteça em determinado momento do futuro. Um plano não garante que um evento realmente vá acontecer. Embora os planos sejam baseados em expectativas, durante a sua implementação os eventos nem sempre acontecem como esperado. Assim, Controle é o processo de lidar com essas variações. Pode significar que os planos precisem ser refeitos a curto prazo. O controle faz os ajustes que permitem que a operação atinja os objetivos que o plano estabeleceu, mesmo que os pressupostos assumidos pelo plano não se confirmem (SLACK; CHAMBERS; JOHNSTON, 2009).

As decisões de planejamento e controle da produção são divididas em três níveis de acordo com os tipos de decisões que serão tomadas: estratégico (decisões de longo prazo), tático (decisões de médio prazo) e operacional (decisões de curto prazo) (JOHNSON; MONTGOMERY, 1974). O planejamento estratégico está relacionado ao mais alto nível de tomada de decisões, em que são definidos os objetivos globais da empresa e a forma como implementar essas metas ao longo de vários anos. Exemplos de decisões tomadas no nível estratégico são desenvolvimento e produção de novos produtos, abertura de novas unidades da empresa, determinação da capacidade de produção e armazenagem nos próximos anos, entre outros.

No nível tático são tomadas e detalhadas decisões para fazer cumprir as metas estabelecidas pelo planejamento estratégico. O planejamento detalhado da produção é realizado nesse nível em um horizonte de planejamento de médio prazo, por exemplo, nos próximos meses. São tomadas as decisões, por exemplo, do que será produzido em cada período (dimensionamento de lotes), quais os recursos necessários para execução do plano de produção, etc. As decisões deste nível podem se tornar mais difíceis à medida que processo de produção envolve múltiplos estágios, muitos produtos, múltiplas máquinas,

demandas sazonais, etc.

No planejamento operacional são tomadas as decisões de mais baixo nível (*eg.* nível chão de fábrica) e de curto prazo, por exemplo nos próximos dias ou semanas. Essas decisões estão mais ligadas ao controle para execução do planejamento que foi realizado no nível tático. Atividades como sequenciar os lotes nos centros de trabalho, administrar estoques em processo, fazer controle de qualidade, entre outras, são típicas do planejamento operacional.

Neste capítulo o objetivo é fazer uma revisão bibliográfica sobre os problemas de planejamento e programação da produção em um nível mais tático/operacional. O objetivo é revisar trabalhos que tratam o problema de dimensionamento e sequenciamento de lotes integrados. Trabalhos voltados à decisões de um nível mais estratégico, tais como planejamento hierárquico da produção, planejamento agregado e otimização da cadeia de suprimentos, não serão tratados aqui em detalhes. Para mais detalhes sobre esses problemas, veja por exemplo, [Christou, Lagodimos e Lycopoulou \(2007\)](#) e [Jans e Degraeve \(2008\)](#).

2.1 Problemas de dimensionamento e sequenciamento de lotes: características, aplicações e métodos de solução

Os problemas de dimensionamento de lotes são problemas de planejamento da produção com *setups* entre os lotes de produção ([BRAHIMI et al., 2006](#)). Devido a esses *setups*, muitas vezes torna-se muito oneroso produzir um determinado produto em todos os períodos. Por outro lado, realizar poucos *setups* para produzir grandes quantidades para atender demandas futuras pode gerar estoques, cujos custos podem ser tão onerosos quanto os *setups*. Ou seja, existe um *trade off* entre fazer *setups* e gerar estoques. Então, o objetivo é determinar os períodos onde a produção deve acontecer e as quantidades para satisfazer a demanda, enquanto se minimiza custos de produção, *setup* e estoque. Outros custos também podem ser considerados, tais como custos de atrasar o atendimento da demanda, custos de troca, entre outros.

O primeiro modelo para problemas de dimensionamento de lotes é o conhecido *Economic Order Quantity* (EOQ) de [Harris \(1913\)](#). Nesse modelo é assumido uma escala de tempo contínua, uma taxa de demanda constante e um horizonte de planejamento infinito. Devido às restrições desta abordagem na modelagem de situações mais realistas, surgiram outros modelos, como o *Economic Lot Scheduling Problem* (ELSP) ([ELMAGHRABY, 1978](#); [ZIPKIN, 1991](#)), onde o problema é programar a produção de mais de um produto em uma única máquina. A capacidade disponível é limitada, e as taxas de produção e demanda são constantes no horizonte de tempo considerado infinito. Desde o surgimento desse trabalho, os problemas de dimensionamento de lotes tem recebido uma ampla atenção tanto da

comunidade acadêmica quanto na prática. [Glock, Grosse e Ries \(2013\)](#) apresentam uma revisão da literatura de dimensionamento de lotes com o objetivo de mostrar como as linhas de pesquisa emergem do modelo proposto por [Harris \(1913\)](#) e quais as maiores contribuições que foram realizadas nessas respectivas áreas.

Existem várias formas de classificar os problemas de dimensionamento de lotes. Em relação ao número de itens: único item ou multi-itens; em relação ao número de máquinas: mono ou multimáquinas; em relação aos parâmetros: estático, se não varia ao longo do tempo, ou dinâmico, caso contrário; ainda com relação aos parâmetros: determinístico ou estocástico; em relação ao número de níveis: único nível ou multinível; em relação a estrutura dos itens produzidos que também podem ser: único nível ou multinível; em relação a restrições de capacidade e recurso: capacitado ou não capacitado, entre outros. Mais detalhes sobre as principais características dos problemas de dimensionamento de lotes e a classificação destes podem ser encontrados em [Karimi, Ghomi e Wilson \(2003\)](#) e [Suerie \(2005\)](#).

Uma característica que merece ser aqui mais discutida é o horizonte de planejamento. O horizonte de planejamento é o intervalo de tempo pelo qual o planejamento da produção se estende, podendo ser finito ou infinito. Além disso, a escala do tempo dentro do horizonte de planejamento pode ser contínua ou discreta. Se for discreta, o número de períodos (*buckets*) são limitados, onde cada um pode ter tamanho grande ou pequeno e, ainda, os *buckets* podem ser classificados como uniforme, se todos possuem o mesmo tamanho, ou não uniforme, se são de tamanhos variáveis. Nos problemas do tipo *small bucket*, o período é relativamente curto e somente um item, ou no máximo dois, podem ser produzidos em cada período. Problemas do tipo *big bucket* são aqueles em que o período de tempo é longo o suficiente para produção de múltiplos itens. Quando a escala de tempo é contínua, não há limitações e definições de períodos pré-definidos.

Os modelos EOQ e ELSP consideram problemas de dimensionamento de lotes não capacitados. Modelos com restrição de capacidade foram discutidos pela primeira vez por [Wagner e Whitin \(1958\)](#), porém abordavam ainda um único item. As empresas geralmente possuem capacidade limitada e além disso, fabricam em geral mais do que um item. Sendo assim, o CLSP (*Capacitated Lot Sizing Problem*) é um modelo típico *big bucket* que considera o dimensionamento de lotes de múltiplos itens com restrições de capacidade. [Florian, Lenstra e Kan \(1980\)](#) mostraram que resolver o CLSP otimamente é considerado um problema NP-difícil. Mais tarde, [Bitran e Yanasse \(1992\)](#) mostraram que alguns casos especiais com um único item é possível resolver o problema em tempo polinomial, mas que a partir de dois itens o problema torna-se NP-difícil. [Trigeiro, Thomas e McClain \(1989\)](#) mostraram ainda que se tempos de *setup* positivos são incorporados ao modelo, o problema de encontrar uma solução factível já é NP-completo.

Revisões sobre problemas de dimensionamento de lotes podem ser encontradas em:

Karimi, Ghomi e Wilson (2003), Brahimí et al. (2006), Quadat e Kuhn (2008), Robinson, Narayanan e Sahin (2009), Glock, Grosse e Ries (2013) e Díaz-Madroño, Mula e Peidro (2014).

Jans e Degraeve (2008) observam que os limites entre trabalhos de dimensionamento e sequenciamento de lotes vem diminuindo cada vez mais, principalmente com a consideração de tempos e custos de *setup* dependentes da sequência, e que a tendência que vem ocorrendo nos últimos anos é considerar esses problemas de maneira integrada. Uma estratégia para incorporar e determinar a sequência dos itens é dividir o horizonte de planejamento em períodos menores (dias, turnos, etc.) e permitir que em cada período seja feito apenas um item. Assim, o sequenciamento fica determinado pela ordem dos itens nos períodos. Como já mencionado anteriormente, esses são os modelos do tipo *small bucket* (DREXL; KIMMS, 1997).

O DLSP (*Discrete Lot Sizing and Scheduling Problem*) é um problema do tipo *small bucket* cuja restrição de capacidade é uma restrição do tipo “tudo-ou-nada”, ou seja, se um item é produzido, ele deve ocupar toda a capacidade do período. Esse modelo foi proposto por Fleischmann (1990). No CSLP (*Continuous Setup Lot Sizing Problem*) proposto por Karmarkar e Schrage (1985), a diferença com o DLSP é que a capacidade não precisa ser totalmente usada, mas ainda continua sendo produzido um único item por período. O DLSP tem complexidade NP-difícil, sendo que uma solução factível pode ser encontrada em tempo polinomial. Se são considerados tempos de preparo ou máquinas paralelas, o problema de factibilidade do DLSP passa a ser um problema NP-completo. A complexidade do DLSP foi estudada por Salomon et al. (1991).

Uma outra variação destes modelos é o modelo PLSP (*Proportional lot sizing and scheduling problem*), apresentada por Haase (1994), em que é permitida a utilização de parte da capacidade. Neste caso é permitida, além da utilização parcial da capacidade, sua utilização na produção de um segundo item dentro de um mesmo período. Assim, podem ocorrer até dois preparos dentro de um período.

Uma crítica aos modelos do tipo *small bucket* é que nos problemas reais, o número de períodos pode ser muito grande, fazendo com que os modelos possam se tornar intratáveis computacionalmente. Para contornar esse problema, Fleischmann e Meyr (1997) propõem o GLSP (*General Lot Sizing and Scheduling Problem*), que trata-se de um híbrido entre *big* e *small buckets*, pois os períodos, denominados aqui macro-períodos, são divididos em um número fixo de períodos menores, denominados subperíodos ou micro-períodos. Em cada subperíodo apenas um item pode ser produzido. Esses subperíodos possuem tamanho variável de acordo com o tamanho do lote. No final, os subperíodos determinarão os tamanhos dos lotes de itens e o sequenciamento dos itens dentro dos macro-períodos.

Em Drexl e Kimms (1997) encontra-se uma revisão de alguns dos modelos que integram dimensionamento e sequenciamento de lotes descritos acima. Note que esses

modelos apresentam o controle do tempo de maneira discreta. Drexl e Kimms (1997) apresentam também uma revisão da consideração do tempo contínuo em problemas de dimensionamento e sequenciamento de lotes. Além disso, tratam dos problemas integrados de dimensionamento e sequenciamento de lotes multinível.

Um outro modelo *big bucket* para formular o problema integrado de dimensionamento e sequenciamento de lotes é o CLSD (*Capacitated lot sizing with sequence-dependent setups*) proposto por Haase (1996). Comparado ao GLSP, o sequenciamento dos lotes dentro de um macro-período não é modelado através de micro-períodos. Ao invés disso, o autor faz uma adaptação do CLSP com a inclusão do sequenciamento dos itens produzidos dentro de um período através do problema do caixeiro viajante (PCV). Para representar o sequenciamento usando o PCVA (Problema do Caixeiro Viajante Assimétrico), para cada período, toma-se um grafo $G = (V, A)$, em que V corresponde a um conjunto de itens a serem produzidos no período e as arestas em A correspondem às possíveis trocas entre cada par de itens. O problema então consiste em encontrar um circuito hamiltoniano saindo de uma origem e passando por todos os vértices (LAWLER et al., 1986). Para eliminar *subrotas* que podem ser geradas na resolução do PCVA, são utilizadas restrições de eliminação de *subrotas* (ÖNCAN; ALTINEL; LAPORTE, 2009). Alguns trabalhos que tratam o dimensionamento de lotes integrado ao sequenciamento de lotes via PCV são: Fleischmann (1994), Salomon, Wassenhove L.N.V.; Dumas e Dauzère-pères (1997), Gupta e Magnusson (2005), Almada-Lobo, Oliveira e Carravilla (2008a), Almada-Lobo, Oliveira e Carravilla (2008b), Toso, Morabito e Clark (2009), Almeder e Almada-Lobo (2011), Defalque, Rangel e Ferreira (2011), Ferreira et al. (2012), Maldonado, Rangel e Ferreira (2014) e Guimarães, Klabjan e Almada-Lobo (2014).

Uma outra maneira de considerar o sequenciamento dos lotes é através da consideração de problemas contínuos, que tratam do tempo de maneira contínua. Suerie (2005) destacam que esse tipo de abordagem é interessante para tratar casos que consideram o carregamento do estado de *setup* (*carry-over*) entre os períodos, e que esse tipo de consideração é importante para indústrias de processo. Alguns trabalhos que abordam o dimensionamento e sequenciamento dos lotes considerando o tempo de maneira contínua são: Suerie (2005), Almeder e Almada-Lobo (2011), Camargo, Toledo e Almada-Lobo (2012), Amorim et al. (2013), Almeder et al. (2015) e Ramezani, Sanami e Nikabadi (2017).

Recentemente Copil et al. (2017) apresentam uma revisão da literatura estruturada para os problemas integrados de dimensionamento e sequenciamento de lotes. Os autores fazem uma classificação dos trabalhos encontrados nos últimos 20 anos, revisando mais de 160 trabalhos. A divisão inicial é de acordo com o modelo do qual o trabalho é derivado, por exemplo, se as restrições clássicas do modelo vêm do GLSP, ou do PLSP, entre outros. Os trabalhos também são classificados de acordo com características do problema que são

levadas em consideração como, por exemplo, estágios de produção, tipos de *setup*, número de máquinas, etc. Também são classificados pelos métodos de solução utilizados e por suas aplicações em problemas reais.

Dada a vasta literatura de problemas de dimensionamento e sequenciamento de lotes, o foco dessa revisão são os trabalhos mais atuais que tratam problemas com algumas das características parecidas com o problema em estudo: dois estágios, com máquinas paralelas, sincronia, limpezas temporais, manutenções periódicas ou aplicação em problema real. Com exceção, também são analisados trabalhos que não possuem nenhuma dessas características, mas que são considerados relevantes. Com o intuito de classificar, encontrar características de tendências de pesquisa e métodos de solução dos problemas de dimensionamento e sequenciamento de lotes integrados, os trabalhos aqui revisados são sumarizados na Tabela 1. Da mesma forma que Copil et al. (2017), os trabalhos são agrupados através das características dos modelos dos quais foram derivados: CLSD, PLSP, CSLP, DLSP e GLSP. Os trabalhos em que os modelos não se enquadram em nenhum desses 5 modelos clássicos são denominados de “OUTRO”. Na primeira coluna é apresentada a referência. Na segunda coluna (Est) apresenta-se o número de estágios de produção; quando os estágios são classificados em “livre” significa que mais de 2 estágios estão sendo abordados. A estrutura dos produtos finais é descrita na coluna “Prod”. Se a estrutura de produto é determinada por um único nível, este é denotado por “1”, se for determinado por mais de um nível (múltiplos níveis), denota-se por “m”. Na coluna “*Setup*” identifica-se se o *setup* é dependente (ds) ou independente da sequência (is). A coluna “Modelo” descreve a estrutura do modelo usado no trabalho. A quinta coluna (Var. Tempo) indica se o modelo têm variáveis contínuas de controle de tempo, como por exemplo, uma variável indicando o tempo de início de um *setup*. A penúltima coluna indica se o trabalho é inspirado e/ou aplicado em alguma situação prática. E, por fim, a última coluna determina o método de solução utilizado para resolver o problema. Os métodos de solução estão abreviados, e a lista com as siglas estão na Tabela 2. Todos os trabalhos apresentados nesta tabela são melhor discutidos no Apêndice B, pois o objetivo dessa seção é analisar as principais características observadas nos trabalhos revisados e não se estender em discutir os detalhes de um a um.

Os trabalhos com aplicação no setor de bebidas são discutidos mais a frente na Seção 2.2 e não estão sumarizados na Tabela 1. Uma revisão detalhada desses trabalhos é apresentada no Apêndice C. Cabe ressaltar que a presente revisão da literatura não tem nenhuma pretensão de esgotar todo o assunto e abordar todos os trabalhos existentes sobre esses problemas. O objetivo é discutir, com base nos trabalhos pesquisados, como os problemas de dimensionamento e sequenciamento de lotes integrado vêm sendo abordados na literatura: as características clássicas, as aplicações e os métodos de solução, a fim de identificar similaridades com o problema tratado nesse trabalho.

Tabela 1 – Resumo das características dos trabalhos revisados e discutidos nesta seção.

Referência	Est	Prod	Setup	Modelo Base	Var. Tempo	Aplicação	Método de Solução
Toso, Morabito e Clark (2009)	1	1	ds	GLSP		Nutrição Animal	MIP-solver, R&F
Mohammadi (2010)	livre	m	ds	GLSP		-	HR+R&F
Almeder e Almada-Lobo (2011)	1	1	ds	GLSP e CLSD	✓	Semicondutor	MIP-solver
Transchel et al. (2011)	1	m	ds	GLSP	✓	Indústria de processo	MIP-solver
Camargo, Toledo e Almada-Lobo (2012)	2	m	ds	GLSP, CLSD e OUTRO	✓	Fiação	MIP-solver
Santos e Almada-Lobo (2012)	2	m	ds	GLSP		Papel e Celulose	F&O estocástico
Figueira, Santos e Almada-Lobo (2013)	2	m	ds	GLSP		Papel e Celulose	VNS
Meyr e Mann (2013)	1	1	ds	GLSP		-	Decomposição
Seeanner e Meyr (2013)	livre	m	ds	GLSP	✓	-	R&F e F&O
Seeanner, Almada-Lobo e Meyr (2013)	livre	m	ds	GLSP	✓	-	VNDS+F&O
Camargo, Toledo e Almada-Lobo (2014)	1	m	ds	GLSP	✓	Fiação	HOPS
Mohammadi e Poursabzi (2014)	livre	m	ds	GLSP		-	HR
Furlan et al. (2015)	2	m	ds	GLSP		Papel e Celulose	AG
Rohaninejad, Kheirkhah e Fattahi (2015)	livre	m	ds	GLSP	✓	-	AG+EP
Martínez, Toso e Morabito (2016)	1	1	ds	GLSP		Polpa moldada	MIP-solver
Clark e Clark (2000)	1	1	ds	CLSD		-	HR, R&F
Clark (2003b)	livre	m	-	CLSD		-	HR, R&F
Gupta e Magnusson (2005)	1	1	ds	CLSD		Lixa	Inicialização, sequenciamento e melhoria
Almada-Lobo et al. (2007)	1	1	ds	CLSD		Recipiente de vidro	MIP-solver; Heurística
Almada-Lobo e James (2010)	1	1	ds	CLSD		Recipiente de vidro	VNS e BT
Almada-Lobo, Oliveira e Carravilla (2008b)	1	m	ds	CLSD		Recipiente de vidro	VNS iterativa com R&F
James e Almada-Lobo (2011)	1	1	ds	CLSD		-	INS e R&F
Lang e Shen (2011)	1	1	ds	CLSD		Folhas de Plástico/Pára-brisa	R&F e F&O
Guimarães, Klabjan e Almada-Lobo (2013)	1	1	ds	CLSD		-	Decomposição, R&F e F&O
Xiao et al. (2013)	1	1	ds	CLSD		Semicondutor	R&F+F&O
Clark, Mahdih e Rangel (2014)	1	1	ds	CLSD		-	MIP-solver
Guimarães, Klabjan e Almada-Lobo (2014)	1	1	ds	CLSD		-	MIP-solver
Tempelmeier e Copil (2015)	1	1	ds	CLSD	✓	Alimentícia	MIP-solver
Almeder et al. (2015)	livre	m	ds	CLSD	✓	-	MIP-solver
Xiao et al. (2015)	1	1	ds	CLSD		Semicondutor	Heurística híbrida com RL e SA
Toledo et al. (2016)	1	1	ds	CLSD		Recipiente de vidro	F&O e AG
Fachini, Esposito e Camargo (2016)	1	1	ds	CLSD		Recipiente de vidro	VNS
Mahdih, Clark e Bijari (2017)	1	1	ds	CLSD		-	MIP-solver

Tempelmeier e Buschkuhl L. (2008)	1	1	is	PLSP	✓	Automobilística	MIP-solver
Stadtler (2011)	1	m	is	PLSP		Farmacêutica	MIP-solver
Stadtler e Sahling (2013)	livre	m	is	PLSP	✓	-	R&F+F&O
Vanderbeck (1998)	1	1	is	CSLP		-	GC+inequações válidas
Araujo, Arenales e Clark (2007)	1	m	ds	CSLP		Fundição	HR+R&F+BL
Marinelli, Nenni e Sforza (2007)	2	m	is	CSLP+CLSP		Iogurte	Decomposição
Araujo, Arenales e Clark (2008)	1	m	is	CSLP		Fundição	HR+R&F+BL
Almada-Lobo et al. (2010)	1	1	ds	CSLP		Recipiente de vidro	RL
Araujo e Clark (2013)	1	m	ds	CSLP		-	HR+R&F+BL;
Toledo, Arantes e Almada-Lobo (2013)	1	1	ds	CSLP		Recipiente de vidro	AG
Salomon, Wassenhove L.N.V.; Dumas e Dauzère-pères (1997)	1	1	ds	DLSP		-	Reformulação PCVJT + PD
Göthe-Lundgren, Lundgren e Persson (2002)	livre	m	is	DLSP		Refinaria	Inequações válidas, BT
Jans e Degraeve (2004)	1	1	is	DLSP		Pneu	GC
Gicquel e Minoux (2015)	1	1	ds	DLSP		-	Inequações Válidas
Dauzère-Pères e Lasserre (1994)	livre	1	is	OUTRO	✓	-	Decomposição
Haase e Kimms (2000)	1	1	ds	OUTRO		Máquina de Pintura	Branch-&-Bound
Kovács, Brown e Tarim (2009)	1	1	ds	OUTRO		-	MIP-solver e PD
Mateus et al. (2010)	1	1	is	OUTRO	✓	-	Decomposição, GRASP
Mohammadi et al. (2010a)	1	1	ds	OUTRO		-	HR e R&F
Mohammadi et al. (2010b)	livre	1	ds	OUTRO		-	HR e R&F
Mohammadi et al. (2010)	livre	1	ds	OUTRO		-	HR
Mohammadi, Ghomi e Jafari (2011)	livre	1	ds	OUTRO		-	AG
Mohammadi e Ghomi (2011)	livre	1	ds	OUTRO		-	AG e HR
Kopanos, Puigjaner e Maravelias (2011)	1	1	ds	OUTRO		Indústria de processo	MIP-solver
Kopanos, Puigjaner e Georgiadis (2011)	1	1	ds	OUTRO		Iogurte	MIP-solver
Belo-Filho, Santos e Menezes (2012)	livre	1	ds	OUTRO		-	A-team
Babaei, Mohammadi e Ghomi (2014)	livre	1	ds	OUTRO		-	AG
Karimi-Nasab e Seyedhoseini (2013)	livre	m	ds	OUTRO	✓	-	Inequações válidas; MIP-solver
Karimi-Nasab et al. (2013)	livre	m	ds	OUTRO	✓	-	Algoritmos meméticos
Ramezani, Saidi-Mehrabad e Fattahi (2013)	livre	1	ds	OUTRO	✓	-	HR
Ramezani, Saidi-Mehrabad e Teimoury (2013)	livre	1	ds	OUTRO	✓	-	HR
Gómez-Urrutia, Aggoune e Dauzère-Pères (2014)	livre	1	is	OUTRO	✓	-	Decomposição, RL e BT
Karimi-Nasab e Modarres (2015)	livre	1	ds	OUTRO	✓	-	Inequações válidas e <i>cut-and-branch</i>
Karimi-Nasab, Modarres e Seyedhoseini (2015)	livre	1	ds	OUTRO	✓	-	EP
Wolosewicz, Dauzère-Pères e Aggoune (2015)	livre	1	is	OUTRO	✓	-	Decomposição, RL
Ramezani, Sanami e Nikabadi (2017)	livre	1	is	OUTRO	✓	Telha	HR e EP
Boormee e Sethanan (2016)	livre	1	is	OUTRO		Ovo de galinha	Variante do EP

Tabela 2 – Abreviação da nomenclatura dos métodos de solução apresentados na Tabela 1

Método	Abreviação
Pacotes de otimização	MIP-solver
Horizonte Rolante	HR
<i>Relax-and-fix</i>	R&F
<i>Fix-and-optimize</i>	F&O
<i>Variable Neighborhood Search</i>	VNS
<i>Variable Neighborhood Decomposition Search</i>	VNDS
<i>Hamming-Oriented Partition Search</i>	HOPS
Algoritmo Genético	AG
Enxame de partículas	EP
Relaxação Lagrangeana	RL
<i>Simulated Annealing</i>	SA
Geração de Colunas	GC
Busca Local	BL
Programação dinâmica	PD
Problema do caixeiro viajante com janela de tempo	PCVJT
Busca Tabu	BT

Jans e Degraeve (2008) destacam que a resolução de problemas reais é uma das áreas que apresenta maior crescimento, sendo uma direção próspera para a pesquisa. Nesse problemas as características próprias de cada sistema de produção são os diferenciais dos modelos. Isso também pode ser observado na Tabela 1 com o grande número de trabalhos inspirados em situações reais. Dentre os 70 trabalhos sumarizados nesta tabela, 33 deles são aplicados ou inspirados em problemas reais. As aplicações são nos seguintes processos de produção: nutrição animal, semicondutor, alimentícia, indústria de processo, fiação, papel e celulose, polpa moldada, lixa, recipiente de vidro, folhas de plástico para pára-brisa, automobilística, farmacêutica, fundição, iogurte, refinaria, pneu, máquina de pintura, telha e ovo de galinha.

O número de modelos considerando múltiplos estágios de produção e produtos multinível tem aumentado nos últimos anos, como pode ser visto na Tabela 1. Apesar disso, a maioria dos trabalhos aplicados a problemas reais apresentam modelos com um único estágio, que em geral é o estágio considerado gargalo da produção. Dentre os 32 trabalhos aplicados a casos reais, somente 3 deles consideram mais de dois estágios: Göthe-Lundgren, Lundgren e Persson (2002), Ramezani, Sanami e Nikabadi (2017) e Boonmee e Sethanan (2016). Dentre as aplicações práticas em sistemas multiestágios, os modelos dois estágios ainda são a maioria, encontrados em 5 trabalhos: Marinelli, Nenni e Sforza (2007), Santos e Almada-Lobo (2012), Camargo, Toledo e Almada-Lobo (2012), Figueira, Santos e Almada-Lobo (2013) e Furlan et al. (2015). A aplicação na indústria de bebidas também é em geral com dois estágios, como pode ser visto em Ferreira, Morabito e Rangel (2009), Baldo et al. (2014) e Toledo et al. (2015). Almada-Lobo et al. (2015) fazem uma breve revisão sobre problemas de dimensionamento e sequenciamento de lotes em dois

estágios aplicados à problemas reais.

Nota-se que existe um crescimento do número de trabalhos que tratam sistema de produção do tipo *job-shop*, por exemplo, [Mohammadi e Poursabzi \(2014\)](#) e [Rohaninejad, Kheirkhah e Fattahi \(2015\)](#), estreitando ainda mais as fronteiras entre problemas de dimensionamento e sequenciamento de lotes. Esse tipo de problema também tem sido abordado via modelos diferentes dos clássicos ([DAUZÈRE-PÉRÈS; LASSERRE, 1994](#); [WOLOSEWICZ; DAUZÈRE-PÉRÈS; AGGOUNE, 2015](#); [GÓMEZ-URRUTIA; AGGOUNE; DAUZÈRE-PÉRÈS, 2014](#)).

Quando os problemas considerados envolvem múltiplos estágios e produtos multitéveis, a sincronia entre predecessores e sucessores deve ser levada em consideração para obtenção de programações da produção viáveis na prática ([ALMEDER et al., 2015](#)). Isto significa que um produto sucessor pode ser processado em uma máquina sucessora somente depois que uma peça, uma quantidade de um lote ou ainda um lote inteiro tenha terminado de ser processado na máquina predecessora. Uma tendência de modelagem que tem sido utilizada para garantir essa sincronia é a utilização de variáveis contínuas para controlar o início e final de operação de cada produto em cada máquina como pode ser visto, por exemplo, em [Camargo, Toledo e Almada-Lobo \(2012\)](#), [Seeanner e Meyr \(2013\)](#), [Stadtler e Sahling \(2013\)](#), [Seeanner, Almada-Lobo e Meyr \(2013\)](#), [Karimi-Nasab e Seyedhoseini \(2013\)](#), [Karimi-Nasab et al. \(2013\)](#), [Karimi-Nasab et al. \(2013\)](#), [Ramezani, Saidi-Mehrabad e Fattahi \(2013\)](#), [Gómez-Urrutia, Aggoune e Dauzère-Pérès \(2014\)](#) e [Karimi-Nasab e Modarres \(2015\)](#).

A consideração de problemas reais em ambientes multiestágios também chamam a atenção para a consideração da oscilação do gargalo entre os estágios. Segundo [Copil et al. \(2017\)](#), o desenvolvimento de *hardwares* cada vez mais potentes e *solvers* mais robustos permite que possam ser incorporadas cada vez mais especificidades de problemas reais nos modelos matemáticos. Assim, segundo os autores a pesquisa em ambientes em que o gargalo é inconstante é provavelmente uma das características que tende a ser mais abordada em pesquisas futuras.

É possível perceber na revisão de literatura também que estão surgindo as mais diversas formas de extensão dos problemas de dimensionamento e sequenciamento de lotes baseados em características reais: a consideração de recursos escassos ([JANS; DEGRAEVE, 2004](#); [TEMPELMEIER; BUSCHKÜHL L., 2008](#); [ALMEDER; ALMADA-LOBO, 2011](#); [TEMPELMEIER; COPIL, 2015](#)), como, por exemplo, um operador de *setup*; a perecibilidade dos itens ([TEMPELMEIER; COPIL, 2015](#)); a inclusão da programação de manutenções periódicas ([RAMEZANIAN; SAIDI-MEHRABAD; FATTAHI, 2013](#)), entre outros ([JANS; DEGRAEVE, 2008](#)).

Como já mencionado, sabe-se que os problemas de dimensionamento de lotes pertencem a classe NP-difícil. Além disso, as restrições acrescentadas aos modelos para

consideração de extensões e detalhes técnicos de problemas reais, por exemplo, podem acabar dificultando a obtenção de uma solução ótima até para problemas pequenos (*toys*). Sendo assim, além do interesse da comunidade científica pela modelagem desses problemas, uma questão bastante estudada é a maneira de resolver esses problemas, ou seja, os métodos de solução. Os mais diversos métodos tem sido utilizados nos últimos anos, conforme pode ser percebido na Tabela 1.

É sabido que modelos de dimensionamento e sequenciamento de lotes apresentam, em geral, limitantes inferiores fracos. Os modelos do tipo CLSD apresentam, na maioria das vezes, limitantes melhores do que os do tipo GLSP (ALMEDER; ALMADA-LOBO, 2011). Além disso, o conhecimento existente sobre reformulações, inequações válidas, técnicas de eliminação de subrotas, etc, para o problema do caixeiro viajante pode ser utilizado para tornar as formulações baseadas no CLSD mais fortes (MALDONADO; RANGEL; FERREIRA, 2014). Técnicas para obter formulações mais fortes, tais como geração de colunas (VANDERBECK, 1998; JANS; DEGRAEVE, 2004), relaxação Lagrangeana (ALMADA-LOBO et al., 2010), reformulações (SALOMON; WASSENHOVE L.N.V.; DUMAS; DAUZÈRE-PÉRÈS, 1997), etc, também são utilizadas em alguns trabalhos, especialmente nos baseados no CSLP e DLSP, como pode ser observado na Tabela 1. Porém, segundo Copil et al. (2017), nem sempre é fácil a aplicação de métodos de decomposição ou relaxação Lagrangeana, por exemplo, para modelos baseados em problemas reais. O subproblema resultante de uma decomposição, por exemplo, pode ainda continuar sendo muito difícil de ser resolvido.

Heurística é o método mais utilizado para a maioria dos trabalhos apresentados na Tabela 1; aproximadamente 68% dos trabalhos utilizam algum tipo de método heurístico, em especial para a resolução de instâncias reais. Como pode ser observado na Tabela 1, MIP-heurísticas do tipo *relax-and-fix*, *fix-and-optimize* e de horizonte rolante são utilizadas com frequência até os dias de hoje (MOHAMMADI, 2010; LANG; SHEN, 2011; XIAO et al., 2013; STADTLER; SAHLING, 2013; MOHAMMADI; POURSABZI, 2014) e com várias variações, como por exemplo, a *fix-and-optimize* estocástica utilizada em Santos e Almada-Lobo (2012). Essas heurísticas também têm sido combinadas com outros métodos, como metaheurísticas (ALMADA-LOBO; OLIVEIRA; CARRAVILLA, 2008b; MOHAMMADI; GHOMI, 2011; SEEANNER; ALMADA-LOBO; MEYR, 2013). No recente trabalho de Chen (2015), por exemplo, os autores aplicam a VNS juntamente com a *fix-and-optimize* para o problema de dimensionamento de lotes capacitado multinível. Esse trabalho não está listado na Tabela 1 por ser um trabalho que trata somente dimensionamento de lotes. Dentre os trabalhos da Tabela 1, 23 utilizam alguma dessas três MIP-heurísticas, sozinhas ou combinadas com outros métodos. Percebe-se que metaheurísticas puras (sem hibridização com outros métodos) também têm sido bastante utilizadas para resolver esses problemas, como pode ser visto, por exemplo nos trabalhos: Toledo, Arantes e Almada-Lobo (2013), Karimi-Nasab et al. (2013), Furlan et al. (2015) e Rohaninejad,

Kheirkhah e Fattahi (2015). Decompor o problema também tem sido uma estratégia bem utilizada, como pode ser notado na Tabela 1; veja, por exemplo, Marinelli, Nenni e Sforza (2007), Mateus et al. (2010) e Meyr e Mann (2013). Menezes, Mateus e Ravetti (2016) também resolvem um problema de planejamento e programação da produção através de uma abordagem hierárquica, resolvendo inicialmente um problema de dimensionamento de lotes, seguido do sequenciamento. Quando a programação da produção não for possível, informações sobre a capacidade são encaminhadas para o planejamento da produção para realização de ajustes. Este trabalho não está classificado na Tabela 1, pois também envolve custos logísticos.

Ambos Copil et al. (2017) e Jans e Degraeve (2008) destacam que a pesquisa em algoritmos híbridos que combinam modelos matemáticos e métodos heurísticos é uma área em ascensão para problemas de dimensionamento e sequenciamento de lotes. Segundo os autores, a inclusão de características de problemas reais nos modelos levam a problemas cada vez mais complexos, e uma alternativa para solução desses modelos é explorar as estruturas e propriedades de subproblemas específicos contidos nesses modelos. As matheurísticas são abordagens que combinam decisões dos modelos matemáticos com (meta) heurísticas (MANIEZZO; STÜTZLE; VOSS, 2010). Trabalhos que utilizam as matheurísticas para resolução de problemas de dimensionamento de lotes são: Seeanner, Almada-Lobo e Meyr (2013), Guimarães, Klabjan e Almada-Lobo (2013), Camargo, Toledo e Almada-Lobo (2014) e Xiao et al. (2015).

O problema que está sendo investigado nesse trabalho possui especificidades que, segundo Copil et al. (2017), Jans e Degraeve (2008) e a revisão aqui realizada, são tendências de pesquisa atualmente: controle da sincronia entre os estágios de produção e presença de limpezas temporais (mais detalhes no Capítulo 3). No caso, as limpezas temporais se assemelham às manutenções periódicas (RAMEZANIAN; SAIDI-MEHRABAD; FATTAHI, 2013). Não são idênticas, pois essas manutenções acontecem periodicamente em um determinado instante de tempo pré-estabelecido, ou dentro de um intervalo de tempo. As limpezas temporais não tem momentos exatos para ocorrer; a única obrigatoriedade é que não podem ser ultrapassadas TP_{max} (TL_{max}) horas ou minutos de produção contínua do mesmo sabor sem a realização dessas limpezas temporais. Ou seja, dependendo da maneira como os lotes são distribuídos ao longo do horizonte de planejamento, essas limpezas temporais podem ser evitadas, diferentemente das manutenções periódicas referidas acima, que sempre tem que ocorrer. Manutenções periódicas são mais encontradas em trabalhos no contexto de *scheduling* (ANGEL-BELLO et al., 2011; PACHECO; BELLO; ÁLVAREZ, 2013; YU; ZHANG; STEINER, 2014).

Convém observar que se os tamanhos dos lotes fossem conhecidos, o problema aqui estabelecido poderia ser visto como um *flowshop* permutacional, em que todos os *jobs* passam por todas as máquinas e seguem a mesma sequência nas máquinas (SEEANNER;

ALMADA-LOBO; MEYR, 2013; RAMEZANIAN; SAIDI-MEHRABAD; TEIMOURY, 2013). Nesse caso, os lotes seriam considerados como *jobs*, o primeiro estágio de produção seria a primeira máquina que os *jobs* passariam e o segundo estágio, a segunda e última máquina. Ramezaniyan, Saidi-Mehrabad e Fattahi (2013) utilizam variáveis de controle do tempo para um modelo de dimensionamento e sequenciamento de lotes para sistemas de produção do tipo *flow shop* com manutenções preventivas. Além desse, existem outros trabalhos na literatura que tratam os problemas de dimensionamento e sequenciamento de lotes considerando os subperíodos de tamanho variável e controlados por variáveis que contam o tempo, conforme por ser observado na Tabela 1. (ALMEDER; ALMADA-LOBO, 2011; CAMARGO; TOLEDO; ALMADA-LOBO, 2012; ALMEDER et al., 2015).

2.2 Planejamento e programação na indústria de bebidas

Muitos trabalhos são encontrados na literatura sobre planejamento e programação da produção de bebidas. Guimarães, Klabjan e Almada-Lobo (2012), Christou, Lagodimos e Lycopoulou (2007) e Gunther (2008) consideram o planejamento da produção de refrigerantes e cerveja em nível tático/estratégico. Sel e Bilgen (2014) abordam o planejamento da produção sob o ponto de vista estocástico. Ferreira, Morabito e Rangel (2009) e Ferreira, Morabito e Rangel (2010) introduzem um MIP baseado no GLSP para resolver o problema de dimensionamento e sequenciamento de lotes de refrigerantes. Dada a dificuldade de resolução desses problemas, MIP heurísticas são utilizadas para resolvê-los. Ferreira et al. (2012) apresentam uma formulação monoestágio para o problema de dimensionamento e sequenciamento de lotes de refrigerantes e consideram reformulações baseadas no caixeiro viajante assimétrico. Modelos que também utilizam restrições do caixeiro viajante na modelagem são: Defalque, Rangel e Ferreira (2011) e Maldonado, Rangel e Ferreira (2014). Toledo et al. (2015) consideram um modelo mais geral do que o abordado por Ferreira, Morabito e Rangel (2009), e apresentam um MIP que também mostra-se difícil de ser resolvido. Baldo et al. (2014) propõem um MIP para dimensionamento e sequenciamento de lotes na indústria cervejeira. Para resolver o modelo os autores propõem MIP heurísticas. Os únicos trabalhos encontrados na literatura que são mais parecidos com problema abordado nessa tese são Leite (2008) e Pagliarussi, Morabito e Santos (2017). Ambos tratam o planejamento e programação da produção de bebidas à base de frutas, porém, nestes apenas o segundo estágio foi modelado e resolvido, por ser considerado o gargalo da produção. Como o primeiro estágio não foi abordado, a presença do tanque pulmão e a sincronia entre os dois estágios também não foram levadas em consideração nesses trabalhos. As limpezas temporais foram tratadas simplesmente como lotes máximos de produção. O problema é resolvido via MIPs baseados no GLSP e CLSP. Uma revisão detalhada sobre problemas de dimensionamento e sequenciamento de lotes de bebidas é apresentada no Apêndice C. Além dos trabalhos acima citados, no Apêndice C também

são discutidos os seguintes trabalhos: Clark (2003a), Rangel e Ferreira (2003), Toledo et al. (2007), Ferreira, Morabito e Rangel (2008), Toledo et al. (2009) e Toledo et al. (2014).

Conforme mostrado no Apêndice C, na revisão de literatura realizada até o momento não foram encontrados trabalhos que abordem o planejamento e programação da produção de bebidas à base de frutas considerando os dois estágios de produção. Porém, em visitas realizadas às empresas do setor, foi possível observar situações em que o gargalo de produção pode variar entre o primeiro e segundo estágio nestas empresas. Portanto, é importante a consideração dos dois estágios de produção no planejamento e programação da produção de bebidas à base de frutas em empresas em geral, como descrito no Capítulo 3.

Apesar dos trabalhos científicos existentes sobre problemas de programação da produção de bebidas (20 trabalhos encontrados nessa pesquisa), convém salientar que esses trabalhos não são diretamente aplicáveis e adaptáveis para o processo produtivo abordado nessa tese de doutorado, descrito em detalhes no Capítulo 3. Nota-se que para cada tipo de produção de bebida, existem particularidades nos processos de preparo e envase da bebida que os diferenciam. A seguir são explicadas as diferenças com relação aos *setups*, a sincronia, as limpezas temporais e os tempos de espera:

- *Setups*: na produção de refrigerantes, os tempos e custos de *setup* são dependentes da sequência de produção nos dois estágios (FERREIRA; MORABITO; RANGEL, 2009). Na produção de cervejas, os tempos e custos de *setup* são dependentes da sequência somente no segundo estágio (BALDO et al., 2014). Na produção de bebidas à base de frutas, os tempos e custos de troca (*setups*) podem ou não ser dependentes da sequência e são considerados nos dois estágios de produção.
- Sincronia: nos três processos de produção é necessário considerar a sincronia entre os estágios para obtenção de planos de produção factíveis na prática. Porém, a forma como essa sincronia é considerada é diferente para cada processo de produção. Na produção de refrigerantes, os tanques preparatórios ficam acoplados as linhas de envase durante todo o processo de envase, até a finalização do tanque. Os tanques são liberados para preparar o próximo lote em minutos ou horas. No processo de produção de cervejas, os tanques ficam ocupados por dias com os processos de maturação/fermentação. Após a bebida pronta, o tanque fica ligado à linha durante o envase (BALDO et al., 2014). Na produção de cerveja podem ainda haver tanques pulmão entre os dois estágios, mas que ainda não foram abordados na literatura, até onde temos conhecimento. Nas bebidas à base de frutas, existe um tanque pulmão em cada linha. Quando um lote está pronto no primeiro estágio, a bebida é enviada para o segundo estágio instantaneamente, e enquanto o segundo estágio envasa este lote, o próximo lote pode ser preparado. Um exemplo de comparação da forma como a sincronia é abordada na literatura de refrigerantes e cerveja, e de como é realizada

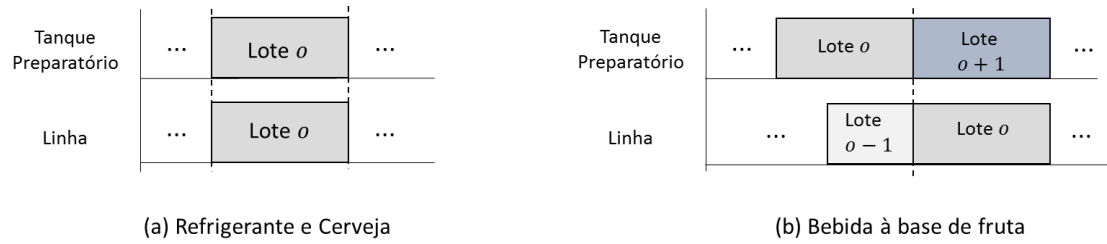


Figura 1 – Diferenças entre a sincronia: refrigerantes e cervejas comparados com bebidas à base de frutas. Fonte: a autora.

para bebidas à base de frutas, está representado na Figura 1.

- **Limpezas temporais:** somente o processo de produção de bebidas à base de frutas exigem a parada da produção para realização de limpezas temporais. As limpezas temporais são obrigatórias depois de determinado tempo decorrido desde o instante da realização da última limpeza. Essas limpezas não podem ser consideradas simplesmente como lote máximo, pois nesse tempo decorrido estão incluídos tempos de produção da bebida e também tempos de espera entre os estágios. Além disso, também devido as esperas que podem acontecer no tempo de duração da limpeza temporal, não é possível prever ou estimar exatamente a quantidade de limpezas temporais para simplesmente subtrair esses valores da capacidade.
- **Tempos de espera:** devido à exigência de limpezas temporais e à existência do tanque pulmão, a maneira como os tempos de espera acontecem em cada estágio de produção para as bebidas à base de frutas se difere da forma como ocorrem as esperas em refrigerantes e cervejas. Como consequência, afeta a modelagem da sincronia. Mais detalhes sobre a programação da produção e a sincronia entre os estágios são apresentados na Seção 3.1.

Em virtude dessas características apresentadas (*setups*, sincronia, limpezas temporais e tempos de espera), os modelos encontrados na literatura sobre problemas de programação da produção de refrigerante e cerveja não são simplesmente adaptáveis, ou facilmente estendíveis, ao problema de bebidas à base de frutas. Essas características afetam significativamente a abordagem de modelagem, sendo necessária a criação de um novo modelo matemático para representar esse problema. Até onde se pesquisou na revisão de literatura apresentada neste capítulo, não foi encontrado nenhum trabalho na literatura que aborde um problema de dimensionamento e sequenciamento de lotes com todas essas características aqui consideradas.

3 Descrição do processo de produção de bebida à base de frutas

Nesse capítulo é descrito em detalhes o processo de fabricação de bebidas à base de frutas. Segundo [Venturini \(2010\)](#), de acordo com a legislação brasileira, as bebidas de frutas, não alcoólicas em geral são classificadas da seguinte forma:

- **Suco:** bebida não fermentada, não concentrada e não diluída, destinada ao consumo, obtida da fruta sã, ou parte do vegetal de origem, por processamento tecnológico adequado, submetida a tratamento que assegure sua apresentação e conservação até o momento do consumo. Em sucos é proibida a adição de aromas e corantes artificiais;
- **Suco integral:** designação privativa do suco sem adição de açúcar e na sua concentração natural, sendo vedado o uso de tal designação para o suco reconstituído. Por exemplo, sucos de laranja derivado da fruta espremida;
- **Suco misto:** obtido pela mistura de duas ou mais frutas e das partes comestíveis de dois ou mais vegetais, ou dos seus respectivos sucos.
- **Suco reconstituído:** obtido pela diluição de suco concentrado ou desidratado, até a concentração original do suco integral ou ao teor de sólidos solúveis mínimo estabelecido nos respectivos padrões de identidade e qualidade para cada tipo de suco integral. Por exemplo, sucos em garrafas de vidro que devem ser misturados a água antes do consumo;
- **Néctar:** bebida não fermentada, obtida da diluição em água potável da parte comestível do vegetal e açúcares ou de extratos vegetais e açúcares, podendo ser adicionada de ácidos, e destinada ao consumo direto, devendo conter um percentual mínimo do suco ou polpa estabelecido;
- **Refresco ou bebida de fruta ou de vegetal:** é a bebida não gaseificada, não fermentada, obtida pela diluição, em água potável, do suco de fruta, polpa ou extrato vegetal de sua origem, com ou sem açúcar.
- **Refresco misto ou bebida mista de frutas ou de extratos vegetais:** é a bebida obtida pela diluição, em água potável, da mistura de dois ou mais sucos de frutas ou de extratos vegetais, devendo o somatório do teor de sucos e extratos vegetais ser estabelecido em ato administrativo;
- **Refrigerante:** é a bebida gaseificada, obtida pela dissolução, em água potável, de suco ou extrato vegetal de sua origem, adicionada de açúcares. Os refrigerantes de laranja, tangerina e uva deverão conter, no mínimo, 10% em volume do respectivo suco na sua concentração natural.

Esse texto explica de maneira detalhada apenas o processamento de bebidas à base de frutas (néctares e refrescos), pois a programação da produção desses produtos é o foco dessa tese. Para mais informações sobre o processamento de sucos naturais, integrais e refrigerantes, veja, por exemplo, [Venturini \(2010\)](#) e [Ferreira, Morabito e Rangel \(2009\)](#).

De forma resumida, no Brasil, para ser chamada de suco, a bebida deve conter 100% da fruta. No caso do néctar, sua composição deve ter um percentual mínimo do suco ou polpa estabelecido pelas Instruções Normativas nº 17, 18, 19 e 42 publicadas em 2013 pelo

Ministério da Agricultura, Pecuária e Abastecimento¹. O néctar de pêsego, por exemplo, deve conter no mínimo 40% de fruta. Caso a bebida tenha um valor menor de fruta do que o estabelecido pelas Instruções Normativas para ser néctar, essa bebida é classificada como refresco ou bebida de fruta. Também segundo essas instruções, a indústria deverá informar no painel principal do rótulo das bebidas não alcoólicas o percentual de polpa da fruta ou suco utilizado nos ingredientes. As Figuras 2, 3 e 4 apresentam as embalagens e as especificações da quantidade de fruta em um refresco, néctar e suco, respectivamente.



Figura 2 – Exemplo de embalagem de refresco. Fonte: a autora.



Figura 3 – Exemplo de embalagem de néctar. Fonte: a autora.

A tecnologia de fabricação vai depender: (i) do tipo de produto que se deseja (suco reconstituído, néctar, refresco ou refrigerante); (ii) da proporção de suco no produto; (iii) da utilização ou não de agentes conservantes; (iv) do ajuste do pH através da adição de ácidos orgânicos; (v) da capacidade de produção requerida; do tipo de processamento (contínuo ou batelada); da embalagem e do sistema de estocagem (ambiente, refrigerado ou congelado) (VENTURINI, 2010). De maneira geral, os néctares e refrescos seguem basicamente o mesmo processo de produção: preparo, pasteurização e envase. O processo pode se diferenciar na etapa de envase. Para o envase dessas bebidas, existem três tipos de processamentos: (a) o envase asséptico, (b) o envase a quente e o (c) envase a frio.

¹ Em:

<<http://www.agricultura.gov.br/legislacao>>. Acesso: 19 de março de 2015



Figura 4 – Exemplo de embalagem de suco. Fonte: a autora.

A Figura 5 ilustra o processo de produção de bebidas à base de frutas com envase asséptico.

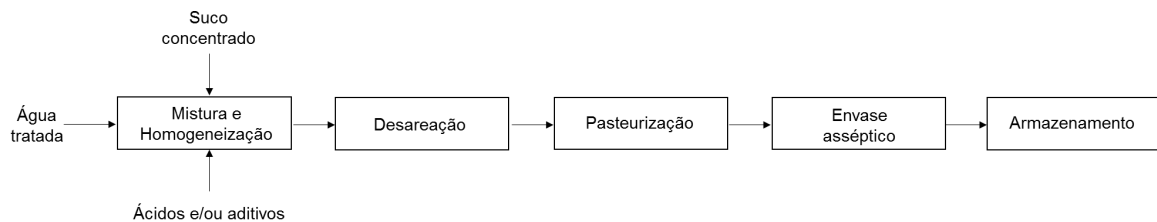


Figura 5 – Fluxograma do processamento via envase asséptico de bebida à base de fruta. Fonte: Adaptado de (VENTURINI, 2010).

Para a obtenção de bebidas de frutas (refrescos), normalmente se parte do suco integral ou concentrado, iniciando-se o processo na etapa de mistura e homogeneização, conforme ilustra a Figura 5. A adição de água tratada e demais ingredientes, como açúcar, ácidos e/ou aditivos, depende do padrão definido para o produto e das características do lote de matéria-prima. Os parâmetros fundamentais para a formulação de cada produto são: o *brix*, a acidez, ou a relação entre *brix* e acidez, o teor de polpa, aromas e óleos essenciais (extraídos da própria fruta) que, adequadamente combinados, conferirão a característica sensorial desejada do suco em termos de sabor, aroma, doçura, acidez e textura. O *brix* é uma escala numérica utilizada para determinar a quantidade de sólidos solúveis em uma solução de sacarose, ou seja, a quantidade de sólidos solúveis é a soma de todos os sólidos - açúcar, sal, proteínas, ácidos, etc. - dissolvidos em água. A acidez é dada pela porcentagem de ácidos, expressa em ácido cítrico. A relação entre o teor de sólidos solúveis e a acidez é padronizada para cada tipo de bebida. Essa formulação final em geral é um grande segredo das empresas, pois é o que vai garantir a preferência do consumidor (VENTURINI, 2010).

Após a formulação por meio do processo de mistura e homogeneização, a bebida passa pelo processo de desaeração. A desaeração consiste na passagem da bebida através de uma câmara sob vácuo para a remoção do oxigênio oriundo dos tratamentos anteriores, ou da agitação na diluição dos ingredientes que compõem o suco. Esse procedimento é

importante, pois a presença de oxigênio dificulta a etapa de pasteurização e promove reações oxidativas de compostos de sabor.

Em seguida, a bebida passa pela pasteurização. Esse procedimento é essencial para reduzir a carga microbiana e inativar enzimas, que podem causar alterações físicas e bioquímicas indesejáveis ao suco. O tratamento térmico deve eliminar microorganismos patogênicos e garantir a redução de carga microbiana em níveis aceitáveis com a vida de prateleira desejada. A pasteurização é uma combinação de tempo e temperatura que é realizada, geralmente, em trocadores de calor tubulares ou de placas a temperaturas entre 85°C e 95°C. Temperaturas inferiores são incapazes de cumprir os objetivos já descritos, e temperaturas superiores poderão comprometer o sabor do suco, adicionando-lhe um sabor indesejável de “cozido”. O pasteurizador possui uma zona de preaquecimento, uma zona de aquecimento para atingir a temperatura de pasteurização, uma zona de retenção onde o líquido é mantido pelo tempo necessário na temperatura de pasteurização e uma zona de regeneração onde o alimento que está entrando é preaquecido e o alimento quente é pré-resfriado. Nessa etapa de regeneração o suco pasteurizado é resfriado até cerca de 7°C a 10°C (VENTURINI, 2010).

Atualmente, a maneira mais comum para o envase de néctar e refresco para venda ao consumidor final é o envase asséptico, com o uso de embalagens cartonadas (pré tratadas com agentes antissépticos físicos e/ou químicos), em máquinas de alta capacidade e ambientes que assegurem as condições de higiene requeridas. Em geral, uma única máquina realiza a assepsia, sela, deposita o volume desejado, remove o ar interno e lacra a embalagem cartonada. Depois de envasada, a bebida é enviada ao setor de armazenamento onde fica até ser entregue aos clientes.

Alternativamente, em sistemas de baixo custo pode-se empregar o envase a quente, seguido da inversão da embalagem, para que o líquido quente atinja a tampa e o espaço vazio sendo assim, estes também submetidos à elevada temperatura. Ou ainda, realizar o envase a frio e tratar termicamente a bebida após o envase, empregando-se sistemas por imersão ou túneis por aspersão de água quente. Nesse caso, com a desvantagem do longo tempo de exposição à elevada temperatura, podem ocasionar mudanças de aroma e até caramelização do produto (VENTURINI, 2010). A vida útil dos produtos depende da matéria-prima, dos ingredientes, do uso de agentes de conservação, do processo empregado, da embalagem, do sistema de envase e da forma e temperatura de estocagem.

3.1 Processo de fabricação de bebidas à base de frutas

Para um melhor entendimento do processo de fabricação de bebidas à base de frutas, foram realizadas algumas visitas técnicas a cinco fábricas do ramo. Em todas estas, o tipo de envase utilizado é o envase asséptico, conforme detalhado na seção anterior. Em

uma delas, localizada no interior do estado de São Paulo, dados reais foram coletados e estão mostrados e discutidos em detalhes no Apêndice A. Essa fábrica é denotada ao longo desse texto por Fábrica A. De acordo com o estudo realizado e conforme discutido na seção anterior de introdução desse capítulo, o processo geral de fabricação de bebidas à base de frutas com envase asséptico consiste basicamente de quatro etapas básicas: preparo da bebida (mistura e homogeneização), pasteurização, envase e empacotamento. Além disto é considerado que cada item é composto apenas pelo sabor da bebida, uma vez que as embalagens são iguais.

Na Figura 6 estão representados os dois estágios de produção considerados principais do ponto de vista da programação da produção e encontrados nesse tipo de processo produtivo. O gargalo de produção desse processo pode oscilar entre um e outro estágio, ao longo dos períodos do horizonte, dependendo do mix de produção e lotes envolvidos para atender a demanda dos produtos. O maquinário necessário para produção de bebidas à base de frutas é composto por mixers, tanques preparatórios, tanques pulmões, pasteurizadores e máquinas de envase. Na Figura 6 também estão representadas por meio de flechas as possíveis ligações entre o maquinário. No Estágio I estão os K mixers e os N tanques preparatórios; no Estágio II estão os M tanques pulmões e pasteurizadores e as L máquinas de envase. Em geral $K \leq N$, $N = M$ e $L \geq M$.

As etapas de produção são divididas em dois estágios. O primeiro estágio, denominado Xaroparia, é composto pelo preparo da bebida. Os processos de pasteurização, envase e empacotamento compõem o segundo estágio, denominado Linha, conforme representado na Figura 6.

O preparo da bebida na xaroparia é dividido em três etapas. Na primeira etapa, os insumos necessários para o preparo da bebida são separados e pesados em quantidades pré determinadas pela receita de preparo de cada sabor. Esses insumos são compostos por partes líquidas e sólidas que, quando misturadas com água, resultarão na bebida pronta para envasar. Na segunda etapa (vide Figura 7), esses ingredientes são dissolvidos em água, um a um no Mixer e, após esse pré-processamento, são enviados, um a um, para o tanque preparatório, onde acontece a mistura desses ingredientes pré processados com água e açúcar. A terceira etapa de preparo refere-se a uma análise química para checar se a bebida está atendendo todas as especificações necessárias. Caso seja necessário, mais água é adicionada à bebida para garantir a qualidade exigida. O tempo gasto para preparar um lote de bebida (um tanque de bebida) é fixo e independe da quantidade de bebida que está sendo preparada.

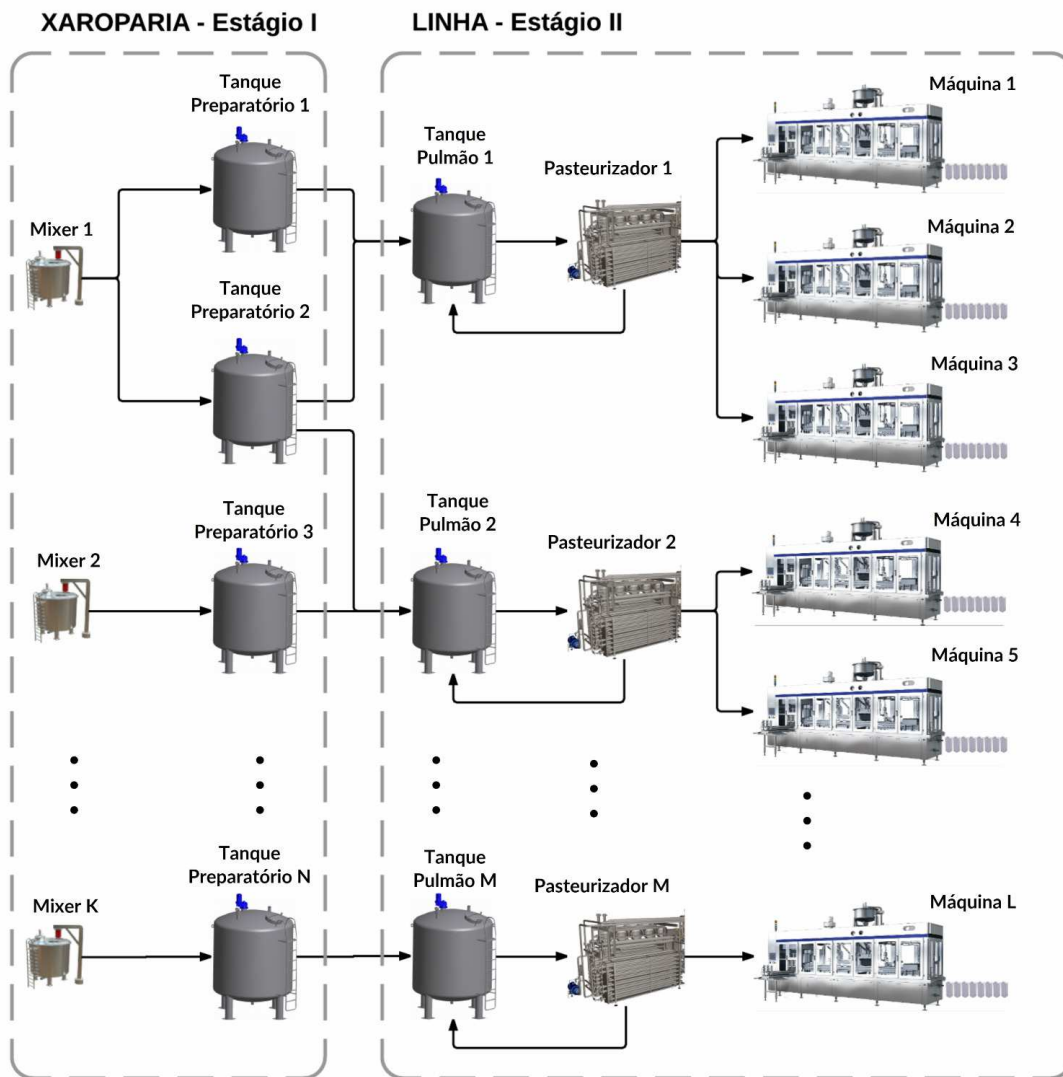


Figura 6 – Processo geral de produção de bebidas à base de frutas. Fonte: a autora.

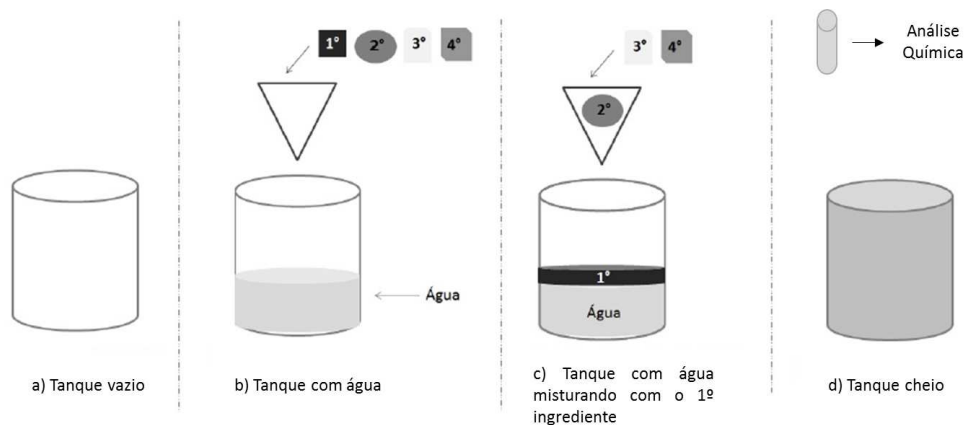


Figura 7 – Processo de preparo da bebidas à base de frutas. Fonte: a autora.

Os insumos chegam às empresas geralmente em *kits* de partes líquidas e sólidas. Para produzir a bebida, os *kits* devem ser utilizados em uma quantidade mínima pré-determinada pela receita da bebida, ou seja, essa quantidade define lotes mínimos para produção de cada sabor. Esse lote mínimo também deve garantir que as hélices que agitam a bebida no tanque fiquem completamente cobertas pela bebida. Por outro lado, o tamanho do lote máximo de bebida que pode ser preparada é limitado pela capacidade física do tanque.

Na maioria das vezes, existe um mixer dedicado para cada tanque preparatório, mas o recurso mixer pode ser algumas vezes compartilhado entre dois tanques preparatórios. É possível ver na Figura 6, por exemplo, que o mixer 1 pode atender os tanques preparatórios 1 e 2. Porém, esse atendimento não ocorre em paralelo, ou seja, de forma simultânea.

Depois de pronto, o lote de bebida pode ser enviado do tanque preparatório para os tanques pulmões do segundo estágio (linha) via tubulações próprias, liberando esses tanques de preparo para iniciar um novo lote de bebida. Os tanques pulmões, além de estocar a bebida enquanto ela é envasada, são necessários pois recebem de volta a bebida que é pasteurizada e ainda não foi envasada. Isto pode ocorrer uma vez que a capacidade de pasteurização é maior que a de envase. Nota-se então que essa capacidade do segundo estágio é determinada pela capacidade das máquinas de envase. Essa volta de bebida está representada na Figura 6 pelas setas que saem do pasteurizador de volta para o tanque pulmão.

Um tanque preparatório pode enviar bebida para mais de um tanque pulmão por vez, e o tanque pulmão pode receber bebida de mais de um tanque preparatório por vez. Nesses casos, a bebida recém-preparada pode ser misturada com a bebida que já está no tanque preparatório. Veja nas flechas da Figura 6 que o tanque preparatório 2 envia bebida para os tanques pulmões 1 e 2, e o tanque pulmão 1 está recebendo suco dos tanques preparatórios 1 e 2. No entanto, a ligação entre tanque pulmão e pasteurizador é biunívoca, ou seja, um tanque pulmão se liga a um único pasteurizador e cada pasteurizador é ligado a um único tanque pulmão. Por isso, nesse tipo de processo produtivo os tanques pulmões e pasteurizadores sempre devem aparecer em números iguais, conforme ilustrado na Figura 6.

As máquinas de envase variam apenas na velocidade de envase e a capacidade de produção dessas máquinas determina a velocidade de produção. Os pasteurizadores podem ser ligados a qualquer uma delas e a várias, ao mesmo tempo. Porém uma máquina só recebe bebida de um pasteurizador por vez. Note que cada pasteurizador possui uma velocidade de produção, logo a soma das velocidades das máquinas ligadas a este não pode ultrapassar sua velocidade. Contudo, a soma das capacidades pode ser inferior a capacidade do pasteurizador. Por exemplo, na Figura 6, caso o pasteurizador 1 tenha velocidade de processamento de 5.000 litros por minuto (l/min), a soma das velocidades das máquinas 1,

2 e 3 que estão ligadas a ele não pode ultrapassar 5.000 l/min. Suponha que a velocidade da máquina 1 seja de 800 l/min, e das máquinas 2 e 3 seja 1500 l/min cada; então a capacidade de envase das três máquinas juntas é de 3.800 l/min. Isso significa que estão retornando para o tanque pulmão 1.200 l/min. Portanto, a velocidade de envase é considerada como a soma das velocidades das máquinas de envase ligadas a um determinado pasteurizador.

Nas empresas visitadas o número de tanques preparatórios é o mesmo que de tanques pulmões, e em geral estão dedicados. Logo, o processo de produção a ser considerado nesta pesquisa considera:

- 1) na xaroparia apenas os tanques preparatórios. Os mixers podem ser ignorados no problema, pois não são considerados como recursos escassos e não interferem nas decisões de planejamento e controle da produção;
- 2) existe um tanque preparatório dedicado à cada linha.

A Figura 8 representa a configuração a ser estudada. Essa simplificação é adotada pois, apesar de todas as possíveis ligações que podem ser feitas entre o maquinário, nas empresas visitadas encontramos esse tipo de configuração. Além do que o problema tratado com essa simplificação já é muito difícil de ser modelado e resolvido. Daqui em diante no texto, o primeiro estágio será referido como xaroparia ou tanque preparatório, e o segundo estágio como linha.

A bebida do tanque preparatório é transferida para a linha quando o tanque pulmão está vazio. Ou seja, a transferência é realizada lote a lote. Assim, o tempo em que o tanque preparatório fica ocupado, transferindo a bebida para o tanque pulmão, é considerado como parte do tempo de produção. Dessa forma, o tanque pulmão faz parte da linha. Ao receber o lote que está pronto na xaroparia, suponhamos um lote s , o tanque pulmão fica abastecendo a linha com esse lote s . Logo, a xaroparia é liberada para a produção do próximo lote ($s + 1$), enquanto a linha está envasando o lote s . Uma linha m (destacada na Figura 8) é composta por tanque pulmão, pasteurizador e um conjunto de máquinas de envase previamente ligadas ao pasteurizador.

3.1.1 Limpezas

Para garantir a segurança alimentar e a higiene industrial, um sistema de gestão da qualidade é implantado na produção de bebidas à base de frutas. Além da pasteurização, pela qual a bebida passa para realização de controle microbiológico, algumas práticas de fabricação são adotadas pelas empresas para garantir a qualidade da bebida. Esse é o caso do controle microbiológico realizado a cada determinado período de tempo de produção, ou ainda o controle da temperatura das máquinas de envase. Uma característica desse controle de qualidade encontrada nos processos de produção de bebidas à base de frutas

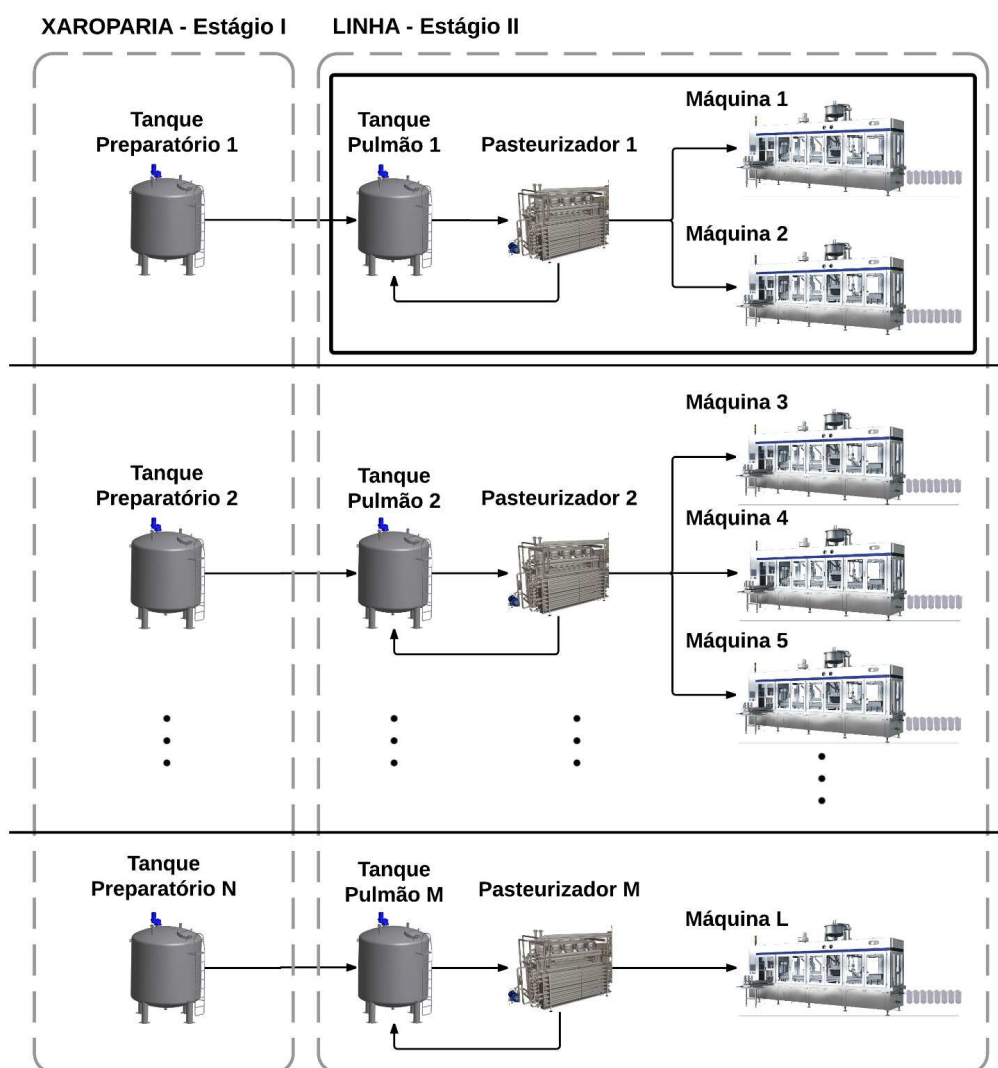


Figura 8 – Processo simplificado de produção de bebidas à base de frutas. Fonte: a autora.

com envase asséptico é a obrigatoriedade de limpezas. Essas limpezas nesse processo de produção são denominadas CIP (*Clear in Place*).

Para um melhor entendimento das limpezas (e das esperas; veja Seção 3.1.2), considere a Figura 9. Nela é apresentada uma programação da produção de um período de um exemplar ilustrativo com três sabores (itens a, b e c), um tanque preparatório e uma linha. Note que não é preciso considerar explicitamente o tanque pulmão, pois ele faz parte da linha, como explicado acima na Seção 3.1. A influência do tanque pulmão na programação da produção está implícita ao se considerar que a bebida pronta no primeiro estágio é transferida para a linha de maneira imediata. Esse tempo de transferência é pequeno e pode ser considerado no tempo de preparo da bebida, ou seja, o tempo de produção de bebida no primeiro estágio é o tempo de misturar os ingredientes (preparo) mais o tempo de transferir a bebida pronta para o tanque pulmão. Logo, enquanto a linha envasa um lote pronto, o tanque preparatório já está disponível para iniciar o preparo

de um novo lote. Na Figura 9, o primeiro estágio (xaroparia) está representado pela primeira linha e é referido como tanque preparatório, uma vez que estamos considerando que cada tanque preparatório está dedicado à uma linha. E o segundo estágio (linha) está representado pela segunda linha da Figura 9. Os quadrados rotulados por a1 a a7 referem-se ao tempo de preparo dos lotes de 1 a 7 do sabor “a”, ou seja, no total foram preparados 7 lotes de sabor “a”. Para o sabor “b” foram preparados 5 lotes, de b1 a b5, e para o sabor “c” apenas dois lotes, c1 e c2. No segundo estágio, cada quadrado dos lotes representa o tempo de envase.

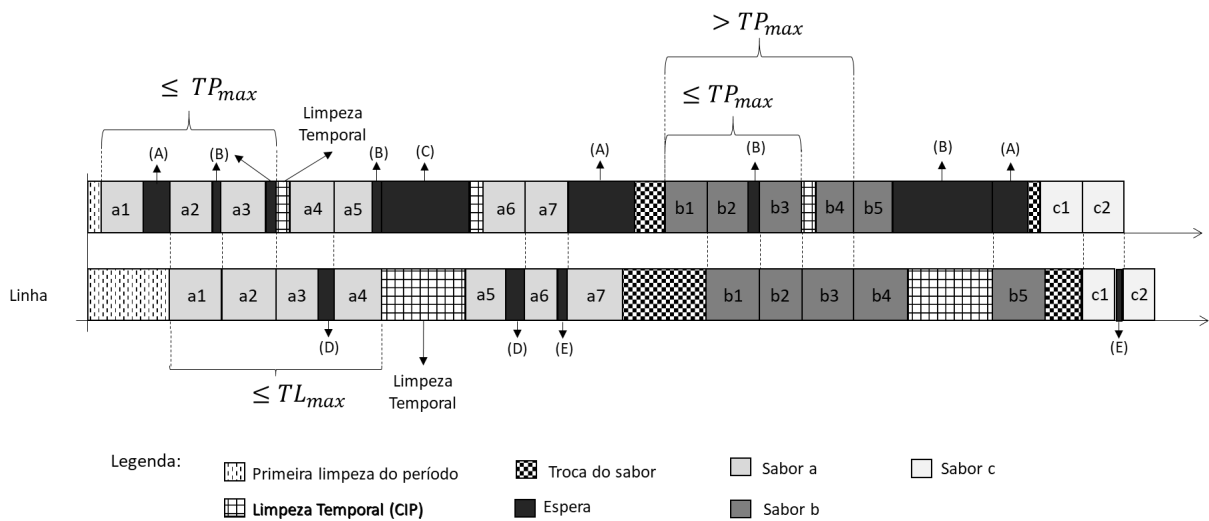


Figura 9 – Exemplar ilustrativo de uma programação da produção factível. Fonte: a autora.

O CIP acontece em três momentos diferentes:

- (I) A cada troca de sabor é necessária uma limpeza. O tempo e o custo da limpeza, nesse caso, pode ou não ser dependente da sequência de produção. Neste trabalho foram observadas as duas situações nas visitas às empresas. Isto porque, dependendo dos tipos de produtos que são produzidos, são adotados procedimentos diferentes para a limpeza. Na presente pesquisa são abordadas as duas situações: tempos e custos dependentes e independentes da sequência.
- (II) No início de cada período ocorre uma limpeza, cujo tempo de duração é, em geral, maior do que das limpezas de troca de sabor. Além disso, o tempo dessa limpeza para xaroparia é diferente do para linha. Não é possível aproveitar uma limpeza ou um preparo do período anterior, isto é, o *setup* não é *carryover*.

Realizar CIPs nesses dois momentos (I) e (II) descritos acima, é comum na produção de outros tipos de bebida. No entanto, na produção de bebidas à base de frutas, há uma terceira situação que exige limpezas na xaroparia e/ou nas linhas:

(III) Passadas TP_{max} (TL_{max}) unidades de tempo (horas ou minutos, por exemplo) desde a realização da última limpeza no tanque preparatório (linha), é necessária a realização de uma nova limpeza. Essas limpezas obrigatórias podem ser denominadas de *limpezas temporais* ou *limpezas periódicas*, pois dependem do tempo decorrido desde a finalização da última limpeza. Na Figura 9 estão representadas as limpezas periódicas após os tempos TP_{max} e TL_{max} decorridos desde a primeira limpeza realizada no período, no tanque preparatório e na linha, respectivamente. A primeira limpeza representada nessa figura é do mesmo tamanho de uma limpeza temporal, o que acontece nas empresas visitadas. A realização da limpeza temporal pode acontecer antes de TP_{max} ou de TL_{max} , mas não pode ultrapassar esses limites. No intervalo desde a última limpeza até TP_{max} (TL_{max}), podem ocorrer a produção contínua do mesmo sabor e/ou esperas. Isto significa que ainda que o tanque preparatório (linha) fique ocioso entre a produção de dois lotes, será necessário uma nova limpeza após TP_{max} (TL_{max}) unidades de tempo. Na Figura 9, por exemplo, observa-se que a sequência de lotes a1, a2 e a3, juntamente com as esperas (A) e (B), atingem o tempo TP_{max} . De forma similar, TL_{max} é alcançado pelo envase dos lotes a1, a2, a3 e a4 e pela espera (D). Por outro lado, os lotes b1, b2, b3 e b4 atingem na linha o tempo TL_{max} sem nenhuma espera, isto é, com a produção contínua de sabor b. Os tempos gastos para a realização da limpeza temporal na xaroparia e na linha são diferentes. Em geral, o tempo de realização de um CIP na linha é sempre maior do que na xaroparia. Em geral, também, tem-se que: $TL_{max} > TP_{max}$.

A troca de embalagens despende um tempo muito menor do que a troca de sabor. Sendo assim, essas trocas são realizadas juntamente com a troca de sabor. E portanto, uma troca de item é na verdade uma troca de sabor.

3.1.2 Sincronia entre os estágios de produção

Nas visitas realizadas às empresas do setor, constatou-se que pode haver gargalo no processo de produção de bebida à base de frutas e este pode variar entre os estágios I e II, dependendo das velocidades dos pasteurizadores, das máquinas de envase e dos tempos de processamento dos lotes no primeiro estágio. Além disso, tanto os tanques preparatórios quanto as linhas de produção são recursos com restrição de capacidade¹, portanto, podem ocorrer esperas da xaroparia (bebida) pela linha (embalagem), e da linha (embalagem) pela xaroparia (bebida). Para que seja possível encontrar uma programação da produção factível para esse problema, é preciso considerar a sincronia entre os estágios, ou seja, é preciso considerar os tempos em que o tanque preparatório (Xaroparia - Estágio I) fica

¹ Recurso com restrição de capacidade (*Capacity-Constrained Resource* - CCR) é aquele cuja utilização é próxima da capacidade e pode ser um gargalo se não for programado corretamente (CHASE; JACOB; AQUILANO, 2006).

ocioso, esperando para enviar a bebida pronta para o tanque pulmão, e os tempos em que o tanque pulmão (Linha - Estágio II) fica vazio, e portanto, a linha se mantém ociosa, esperando o tanque preparatório enviar a bebida para ser pasteurizada e envasada.

Para um melhor entendimento das esperas que podem ocorrer, considere a Figura 9. As linhas pontilhadas representam o instante em que o lote está sendo transferido do tanque preparatório para o tanque pulmão. Os tempos de preparo da bebida são sempre os mesmos, independente do tamanho do lote (fração de tanque ou tanque cheio) que está sendo preparado. As esperas de tipo (A) a (E) que aparecem na Figura 9 são explicadas a seguir.

Tipo (A): Tanto no tanque preparatório quanto na linha de produção, a programação do início do período é sempre iniciada com uma limpeza. Essa limpeza despende, em geral, o mesmo tempo de uma limpeza temporal, independentemente de qual foi o último sabor preparado no período anterior. Tendo em vista que a limpeza da linha no início do período é maior do que no tanque, acontece a primeira espera do tanque pela linha (A), vide situação na Figura 9. Apesar do lote a1 estar pronto no tanque preparatório, ele não pode ser enviado para a linha, pois ainda está acontecendo a limpeza nesta. Essa espera também acontece nas trocas entre os sabores de “a” para “b”, e de “b” para “c”. Observe que essa espera pode ser adicionada antes da limpeza do tanque preparatório para, por exemplo, evitar ser contada no ciclo de duração da limpeza, mas também poderia ter ocorrido após a limpeza.

Tipo (B): Essas esperas acontecem quando o tempo de envase de um lote s é maior do que o tempo de preparo do próximo lote, $s + 1$. Na Figura 9, por exemplo, o lote a2 ficou pronto no tanque preparatório antes do término do envase do lote a1 pela linha.

Tipo (C): Note na Figura 9 que a espera tipo (C) ocorre em seguida da espera tipo (B). Elas estão apresentadas separadamente, pois a espera tipo (C) ocorreu devido à uma limpeza temporal na linha, isto é, o tanque continuou esperando a linha.

As esperas de (A) a (C) descritas acima são esperas do tanque preparatório pela linha. As esperas de (D) e (E), descritas a seguir, são esperas da linha pelo tanque preparatório.

Tipo (D): Quando ocorre a limpeza temporal na xaroparia (tanque preparatório), o preparo do lote seguinte é “atrasado”, então a linha espera o término do preparo. Na Figura 9 a espera da linha após o lote a3 ocorre devido à limpeza temporal no tanque realizada após o preparo do lote a3. Essa espera nem sempre acontece. Note, por exemplo, que enquanto o lote b3 é envasado, ocorre uma limpeza temporal no tanque preparatório e o preparo do lote b4, e portanto não há nenhuma espera da linha pelo tanque preparatório.

Tipo (E): Quando o tempo para a máquina envasar um lote s é menor do que o tempo de preparo do próximo lote, $s + 1$, seja porque o lote é pequeno ou porque a máquina é veloz,

acontece uma espera da linha pelo preparo do lote $s + 1$, (E). Na Figura 9, o envase do lote a6 terminou antes do final do preparo do lote a7. O mesmo aconteceu no envase do lote c1 e c2.

As capacidades limitadas da xaroparia e da linha de produção fazem com que as esperas oscilem entre os estágios. Se a linha tem menos capacidade (é mais lenta), há mais esperas no estágio I. Se o tanque preparatório tem menos capacidade com relação à linha, então as linhas esperam pelo preparo dos lotes. Essas esperas acontecem também porque o sistema de produção em questão tem a característica de que o segundo estágio depende do primeiro estágio, e não existe um estoque intermediário entre os dois estágios que seja suficiente para manter o segundo estágio sempre suprido. Assim, a sincronia entre os estágios de produção deve considerar, além das restrições usuais de um problema de dimensionamento e sequenciamento de lotes, as esperas descritas anteriormente.

3.2 Definição do problema

Após o estudo e descrição do processo de produção de bebidas à base de frutas da seção 3.1 anterior, conclui-se que o problema a ser resolvido consiste na determinação, de forma integrada, do dimensionamento e do sequenciamento dos lotes de produção nos dois estágios. O objetivo é utilizar de maneira adequada a capacidade disponível para atender toda a demanda em cada período, minimizando os custos de produção. Estes custos são decorrentes do não atendimento da demanda (falta) e da antecipação da produção (além da demanda) no final de cada período. O não atendimento da demanda é chamado daqui em diante de atraso, e a produção antecipada é referida como estoque.

Nesse tipo de empresa, o horizonte de planejamento é geralmente entre 1 e 3 meses, e cada período é considerado como uma semana. Em geral, um planejamento é feito com os três (ou dois) meses e mensalmente a demanda é ajustada para realização da programação da produção nos períodos daquele mês. Logo, a programação da produção é realizada considerando essa demanda (mensal, bimestral ou trimestral) como determinística.

Nas empresas visitadas a prioridade é sempre satisfazer a demanda do cliente. Uma venda não atendida, ou atendida com atraso, incorre em um custo alto para a empresa, pois dependendo do contrato e da insatisfação do cliente, esta venda pode até tornar-se perdida e a confiança desse cliente na empresa pode diminuir. Portanto, o custo de atraso para o problema aqui abordado, é considerado o mais alto entre todos os custos considerados.

Em empresas de bebidas existe uma preocupação com estoques semanais devido ao alto valor agregado dos produtos e sua perecibilidade. Além disso, existe um conflito frequente entre a produção e a logística. Se existe muita produção além da demanda, esse excesso de produção é enviado para o setor de logística e os custos de estocar esses produtos são empurrados para esse setor. Portanto, é relevante e justificável a penalização

dos estoques no final de cada período (semana).

Nas empresas visitadas foram encontrados entre 5 e 20 itens diferentes. Assim, em situações onde a capacidade é ociosa, as trocas e limpezas temporais devem ser penalizadas com o objetivo de forçar um maior aproveitamento da capacidade disponível. Essa penalização é menor do que os custos de estoque e atraso, pois essas limpezas não são tão custosas para a empresa. O intuito dessa penalização é evitá-las para possibilitar um aumento de produtividade, mas sem permitir que sejam gerados atraso ou estoque em detrimento de uma limpeza.

Assim, o objetivo desse problema de programação da produção é encontrar um programa factível (determinar quanto produzir e em que ordem para um horizonte finito dividido em períodos, que nesse caso são semanas), minimizando custos de atraso, estoque e penalizando trocas e limpezas temporais. Logo, esse problema é abordado nesta tese como um problema de dimensionamento e sequenciamento dos lotes de produção, nos dois estágios, minimizando custos de estoque e atraso, as trocas e as limpezas temporais, tendo em vista as seguintes características e restrições a serem satisfeitas:

- o horizonte de planejamento é finito e dividido em períodos. Esses períodos possuem mesmo tamanho e correspondem a uma semana;
- a demanda é dinâmica, pois pode variar de período para período, e determinística, pois admite-se que seja previamente conhecida, ou bem estimada;

Estágio I

- no primeiro estágio, formado pelos tanques preparatórios, acontece o preparo da bebida;
- o tempo de preparo dos lotes é fixo e independente da quantidade e do sabor preparados;
- o tempo de transferência de um lote do tanque preparatório para a linha é considerado no tempo de preparo;
- no início de cada período é necessária a realização de uma limpeza, que em geral tem a mesma duração de uma limpeza temporal;
- limpezas temporais devem ser realizadas a cada TP_{max} unidades de tempo (horas ou minutos) desde a realização da última limpeza.

Estágio II

- o segundo estágio é definido pelas linhas de produção, responsáveis pelo envase e compostas de tanque pulmão, pasteurizador e máquinas de envase;
- a velocidade de envase é determinada pela soma das velocidades das máquinas de envase;
- no início de cada período é necessária a realização de uma limpeza, que em geral tem a mesma duração de uma limpeza temporal;
- limpezas temporais devem ser realizadas a cada TL_{max} unidades de tempo (horas ou minutos) desde a realização da última limpeza.

Sincronia entre os estágios

- os lotes têm o mesmo tamanho nos dois estágios. O lote mínimo é determinado pela quantidade mínima que cada *kit* de bebida é capaz de fazer. O lote máximo é determinado pela capacidade física dos taques preparatórios e pela quantidade que pode ser envasada de maneira contínua em TL_{max} unidades de tempo;
- depois de pronto no tanque preparatório, o lote inteiro é transferido para o segundo estágio, desde que a linha tenha terminado de envasar o lote anterior, pois o tanque pulmão é considerado parte da linha.
- na linha, o tanque pulmão abastece as máquinas de envase com o lote pronto, enquanto um novo lote é preparado no tanque preparatório.
- a sincronia entre os estágios deve ser garantida com a consideração das esperadas descritas na Seção 3.1.2.

Neste capítulo foi detalhado o problema de pesquisa desse trabalho. Dadas as características e especificidades desse problema e a revisão de literatura apresentada no Capítulo 2, constata-se que esse problema é original, tem relevância científica e ainda não foi abordado na literatura. No próximo capítulo é proposta uma modelagem matemática para representação desse problema de programação da produção de bebidas à base de frutas.

4 Modelagem matemática

Neste capítulo é apresentado o Modelo Integrado de Dimensionamento e Sequenciamento de Lotes Dois Estágios com Limpezas Temporais (MDSL-2E-LT). Trata-se de uma nova abordagem de modelagem, diferente dos modelos encontrados para o planejamento e programação de bebidas à base de frutas e outros produtos. Devido à existência de tanques pulmão e das limpezas temporais, a sincronia entre os estágios difere da sincronização modelada em problemas de programação da produção de bebidas encontradas na literatura científica relacionada, conforme detalhado no Capítulo 2. Além disso, apresenta-se também aqui uma simplificação do modelo, desconsiderando as limpezas temporais (MDSL-2E), que pode ser aplicado a problemas com processos industriais que tenham um tanque pulmão entre os dois estágios, mas em que não existe a exigência de limpezas temporais.

4.1 Modelo Integrado de Dimensionamento e Sequenciamento de Lotes Dois Estágios com Limpezas Temporais (MDSL-2E-LT)

A partir da descrição do problema apresentada no Capítulo 3, as premissas de modelagem utilizadas para o modelo proposto são elencadas a seguir:

- Cada tanque preparatório da xaroparia está dedicado a uma linha, conforme Figura 8.
- As designações das máquinas de envase aos pasteurizadores são pré-determinadas no início da produção.
- Estoques e atrasos de produto final são permitidos.
- Os lotes de bebida no tanque preparatório são os mesmos da linha de envase. Isto é, um lote é considerado como a quantidade de bebida produzida (preparada) em um tanque. A produção é discretizada em número de tanques produzidos.
- Cada lote de bebida deve satisfazer uma quantidade mínima e máxima de produção, ou seja, devem satisfazer lotes máximos e mínimos. A quantidade mínima que define o lote mínimo é pré-estabelecida de acordo com os *kits* para fazer a bebida e com a quantidade mínima para cobrimento das hélices nos tanques. O lote máximo fica definido pela capacidade física dos tanques e por quantidades que não ultrapassem TL_{max} unidades de tempo (horas ou minutos) de produção na linha.

- O tempo de produção de cada lote no primeiro estágio é fixo, enquanto os tempos de envase no segundo estágio são variáveis, pois dependem do tamanho do lote e da velocidade da linha que irá envasá-lo.
- O tempo despendido para o envio da bebida do primeiro para o segundo estágio é pequeno. Esse tempo é considerado no tempo de produção do primeiro estágio. Ou seja, o lote pronto no primeiro estágio está instantaneamente disponível para utilização no segundo estágio.
- Devido a existência do tanque pulmão, assim que o primeiro estágio termina a produção de um lote e este é enviado para a linha, o tanque preparatório já está disponível para iniciar a produção de um novo lote. Não é necessário esperar o término do envase como acontece em outros processos de produção de bebidas.
- Os tempos de troca (*setup*) são dependentes da sequência de produção nos dois estágios. O caso independente da sequência pode ser tratado mudando-se apenas os dados de entrada do problema.
- Limpezas temporais são obrigatórias e devem ser realizadas no máximo a cada TL_{max} unidades de tempo na linha, e a cada TP_{max} unidades de tempo no tanque preparatório, desde a realização da última limpeza, seja essa última limpeza realizada por limpeza temporal ou por troca de sabor ou por início de um novo período.
- No início de cada período é obrigatória a realização de uma limpeza nos dois estágios, independentemente do último item que foi produzido no período anterior. Essa limpeza tem a mesma duração de uma limpeza temporal.
- A demanda para os produtos ocorre no final de cada período.
- A função objetivo considera custos de produção além da demanda (estoque), custos de não atendimento da demanda (atraso), e uma penalização pelo tempo despendido com limpezas para troca de sabor e com limpezas temporais.

Todas essas premissas são consistentes e realistas com o processo produtivo das empresas desse setor, conforme apresentado no Capítulo 3. O modelo matemático apresentado a seguir considera cada uma delas.

Considere a seguir os conjuntos, parâmetros e variáveis utilizados no modelo.

Conjuntos

- J conjunto de itens (sabores) (i e $j \in J$).
- M conjunto de tanques preparatórios/linhas ($m \in M$).
- T conjunto de períodos ($t \in T$).
- O_{mt} conjunto de lotes que podem ser produzidos pelo par tanque preparatório

e linha, denotado por m , no período t ($o \in O_{mt}$).

Q_{mt}^I conjunto das possíveis limpezas temporais no tanque preparatório m no período t ($k \in Q_{mt}^I$).

Q_{mt}^{II} conjunto das possíveis limpezas temporais na linha m no período t ($l \in Q_{mt}^{II}$).

Parâmetros

- d_{jt} demanda em unidades do item j no período t .
- h_j^+ custo de estoque de uma unidade do item j .
- h_j^- custo de atraso de uma unidade do item j .
- I_{j0}^+ estoque (produção em excesso) em unidades do item j no início do primeiro período do horizonte de planejamento. Considera-se $I_{j0}^+ = 0$.
- I_{j0}^- atraso (não atendimento da demanda) em unidades do item j no início do primeiro período do horizonte de planejamento. Considera-se $I_{j0}^- = 0$.
- i_0 produto fictício para o qual o tanque preparatório e a linha são preparados no início de cada período (item fantasma), $i_0 \in J$.
- LB_j quantidade mínima de produção, em litros, do item j em um tanque (lote mínimo de cada batelada).
- UB_j quantidade máxima de produção, em litros, do item j em um tanque (lote máximo de cada batelada).
- ρ_j quantidade de bebida em litros para produzir uma unidade de um item do tipo j .
- Tp tempo de produção (mistura/preparo) de um lote de bebida no tanque, independente de sabor e quantidade.
- LT^I tempo de realização de uma limpeza temporal no tanque preparatório.
- LT^{II} tempo de realização de uma limpeza temporal na linha.
- Cl_t custo de uma limpeza temporal no tanque preparatório e na linha.
- TC_{ij}^I tempo de uma limpeza no tanque preparatório para troca do item i para o item j .
- TC_{ij}^{II} tempo de uma limpeza na linha para troca do item i para o item j .
- Cap_{mt} capacidade total disponível em tempo do tanque preparatório/linha m no período t .
- S_m velocidade de envase da linha m em litros por hora.
- TP_{max} tempo máximo permitido sem limpeza temporal desde a realização da última limpeza no tanque preparatório, durante a produção de um mesmo sabor.
- TL_{max} tempo máximo permitido sem limpeza temporal desde a realização da última limpeza na linha, durante a produção de um mesmo sabor.
- $Mgde$ número suficientemente grande (em geral, utilizado como 15% maior do que Cap_{mt}).
- C_{ij} penalização da troca do item i para o item j .

α penalização das variáveis ω_{mjtok}^I e ω_{mjtok}^{II} na função objetivo.

A cardinalidade do conjunto O_{mt} é uma estimativa da quantidade máxima de lotes que podem ser produzidos na linha m no período t e é dada pela equação (4.1).

$$|O_{mt}| = \left\lceil \frac{\left[Cap_{mt} - \left(\left\lfloor \frac{Cap_{mt}}{TL_{max}} \right\rfloor + 1 \right) \cdot LT^{II} \right] \cdot S_m}{0,75 \cdot \max_j \{UB_j\}} \right\rceil \quad (4.1)$$

Supondo que haja a produção apenas do sabor j durante todo o período, desconta-se da capacidade disponível Cap_{mt} uma estimativa das possíveis limpezas temporais que poderiam acontecer $\left(\left\lfloor \frac{Cap_{mt}}{TL_{max}} \right\rfloor + 1 \right) \cdot LT^{II}$. O resultado é multiplicado pela velocidade da linha para estimar a quantidade em litros que seria possível envasar. Esse valor é dividido por 75% do maior lote máximo, que também é apenas uma estimativa. Esse valor foi estimado baseando-se na observação de que nas empresas visitadas é comum a utilização do tanque um pouco abaixo da capacidade total (capacidade real e nominal).

Os números de possíveis limpezas temporais que podem ocorrer no tanque preparatório e na linha, ou seja, as cardinalidades dos conjuntos Q_{mt}^I e Q_{mt}^{II} , são dados pelas equações (4.2) e (4.3) a seguir. Eles são dadas pela divisão da capacidade total de tempo disponível da linha e do tanque preparatório pelo tempo máximo permitido sem a realização de limpeza temporal.

$$|Q_{mt}^I| = \left\lfloor \frac{Cap_{mt}}{TP_{max}} \right\rfloor. \quad (4.2)$$

$$|Q_{mt}^{II}| = \left\lfloor \frac{Cap_{mt}}{TL_{max}} \right\rfloor. \quad (4.3)$$

Variáveis

- I_{jt}^+ estoque em unidades do item j no final do período t .
- I_{jt}^- atraso em unidades do item j no final do período t .
- X_{mjto} quantidade de itens do tipo j produzida pelo par tanque preparatório/linha m no período t no lote o .
- Y_{mjto} = 1 se há produção pelo conjunto tanque preparatório/linha m do item j no lote o no período t ; 0, caso contrário.
- Z_{mijt} = 1 se existe uma troca do item i para o item j no tanque preparatório/linha m do período t ; 0 caso contrário.
- Z_{mjjt} = 0 para todo item j .
- V_{mjt} variável auxiliar para eliminação de subrota envolvendo o item j no conjunto tanque preparatório/linha m no período t .
- μ_{mjto}^{Is} instante de início do preparo do lote o do item j no tanque preparatório m no período t .
- μ_{mjto}^{Ie} instante de término do preparo do lote o para o item j no tanque preparatório

	m no período t .
μ_{mjto}^{IIs}	instante de início do envase do lote o para o item j na linha m no período t .
μ_{mjto}^{IIE}	instante de término do envase do lote o para o item j na linha m no período t .
ψ_{mjtok}^I	instante de início da limpeza temporal k no tanque preparatório m que ocorre imediatamente antes do início do preparo do lote o do sabor j no período t .
ψ_{mjtol}^{II}	instante de início da limpeza temporal l na linha m que ocorre imediatamente antes do início do envase do lote o do sabor j no período t .
W_{mjtok}^I	$= 1$, se é realizada a limpeza temporal k no tanque preparatório m imediatamente antes do início do lote o do sabor j no período t ; 0 , caso contrário.
W_{mjtol}^{II}	$= 1$, se é realizada a limpeza temporal l na linha m imediatamente antes do início do lote o do sabor j no período t ; 0 , caso contrário.
ω_{mjtok}^I	$= 1$, se os lotes, a partir do lote o (inclusive), $o > 1$, do sabor j no tanque preparatório m no período t , ocorrem após a limpeza temporal k ; 0 , caso contrário.
ω_{mjtol}^{II}	$= 1$, se os lotes, a partir do lote o (inclusive), $o > 1$, do sabor j na linha m no período t , ocorrem após a limpeza temporal k ; 0 , caso contrário.

Uma vez que tanque preparatório e linha estão dedicados um ao outro, no modelo proposto o índice m representa ambos. Os lotes de bebida no tanque preparatório são os mesmos das linhas de envase, e são designados pelo índice o . Cabe ressaltar que cada lote de bebida refere-se a produção de um tanque de bebida com uma quantidade variando entre os lotes mínimo e máximo permitidos. Ou seja, a produção é discretizada em número de tanques preparados. Esse tipo de discretização da produção em lotes foi inspirada nos trabalhos de [Almeder e Almada-Lobo \(2011\)](#) e [Camargo, Toledo e Almada-Lobo \(2012\)](#). Essa discretização foi utilizada para facilitar a modelagem da sincronia entre os estágios. É preciso controlar as limpezas temporais que acontecem entre lotes do mesmo sabor e como essas limpezas podem resultar em esperas em ambos os estágios. A discretização dos lotes em o facilita o controle dos momentos exatos de início, término e espera de cada lote, além do momento exato em que as limpezas devem ser realizadas, permitindo assim a sincronia entre os estágios.

É importante evidenciar que os lotes o utilizados no modelo MDSL-2E-LT não tem a mesma função que os subperíodos dos modelos do tipo GLSP, pois os subperíodos nesse tipo de modelagem tem o objetivo de controlar o sequenciamento dentro de cada período, o que não é o caso do modelo proposto. Considerando m e t fixos, a Figura 10 ilustra a maneira como o GLSP considera os subperíodos e como o MDSL-2E-LT utiliza os lotes. Na Figura 10(a) os subperíodos são numerados de 1 a 4, já na Figura 10(b) para cada item diferente a contagem de lotes é reiniciada.

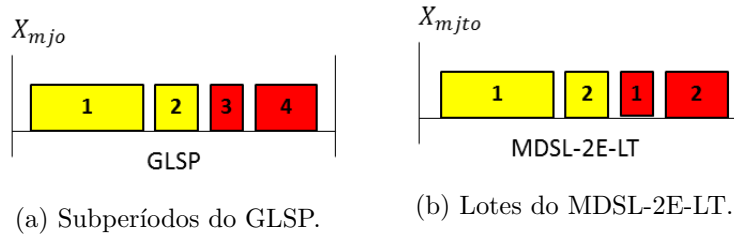


Figura 10 – Comparação dos subperíodos do GLSP com os lotes do MDSL-2E-LT.

Também para modelar a sincronia entre os estágios e controlar as limpezas temporais, variáveis contínuas de controle de tempo são utilizadas. Variáveis desse tipo vêm sendo usadas em diversos trabalhos nos último anos (RAMEZANIAN; SANAMI; NIKABADI, 2017; ALMEDER et al., 2015; TEMPELMEIER; COPIL, 2015; SEEANNER; MEYR, 2013), conforme apresentado no Capítulo 2.

No modelo proposto, o sequenciamento dos lotes é realizado através de restrições de fluxo e de restrições de eliminação de subrotas do caixeiro viajante, assim como em modelos do tipo CLSD (HAASE, 1996).

O Modelo MDSL-2E-LT é dado pelas equações (4.4)-(4.43) apresentadas a seguir.

Função objetivo.

Como discutido na Seção 3.2, a função objetivo proposto em (4.4) tem como intuito impor ao modelo que apresente soluções que aproveitem ao máximo a capacidade, em qualquer situação, seja com capacidade folgada ou restrita. Esse é o objetivo das empresas, um melhor aproveitamento possível dos recursos disponíveis. Para isso são minimizados os estoques, os atrasos, a realização de trocas e de limpezas temporais. As variáveis ω_{mitok}^I e ω_{mitol}^{II} também são penalizadas nessa função objetivo devido às restrições (4.27), (4.28), (4.30), (4.31), (4.33) e (4.35), descritas mais adiante.

$$\begin{aligned}
 \text{Min } Z = & \sum_{t \in T} \sum_{j \in J} (h_j^+ I_{jt}^+ + h_j^- I_{jt}^-) + \sum_{m \in M} \sum_{t \in T} \sum_{j \in J} \sum_{\substack{i \in J \\ i \neq j}} C_{ij} Z_{mijt} \\
 & + \sum_{m \in M} \sum_{j \in J} \sum_{t \in T} \sum_{o \in O_{mt}} \sum_{k \in Q_{mt}^I} (Clt W_{mitok}^I + \alpha \omega_{mitok}^I) \\
 & + \sum_{m \in M} \sum_{j \in J} \sum_{t \in T} \sum_{o \in O_{mt}} \sum_{l \in Q_{mt}^{II}} (Clt W_{mitol}^{II} + \alpha \omega_{mitol}^{II}) \quad (4.4)
 \end{aligned}$$

Os pesos dados a cada um desses objetivos refletem o impacto de cada um desses fatores na melhor otimização possível dos recursos para as empresas desse setor. A prioridade é o atendimento da demanda sem estoques ou atrasos, ou seja, os custos das variáveis I_{jt}^+ e I_{jt}^- são sempre maiores do que os custos das outras variáveis, sendo que o atraso tem um custo ainda mais alto do que o estoque. Com o intuito de ter uma

programação da produção otimizada ainda que a capacidade seja folgada, as trocas e limpezas temporais também são penalizadas, entretanto com custos menores do que os de estoque e atraso, contemplando assim um *trade-off* entre produzir estoque e fazer trocas e limpezas, mas não deixando de fazer limpezas em detrimento do atendimento exato da demanda. A penalização de ω_{mitok}^I e ω_{mitol}^{II} é muito pequena em relação aos outros custos.

Restrições de dimensionamento e sequenciamento de lotes e sincronia.

Balanceamento de estoque e demanda. As restrições de balanceamento de estoque e demanda são dadas por (4.5). As demandas são dadas por período. Uma vez que as variáveis de produção são dadas por linha m e lote o , é preciso somar essas quantidades para obter o total produzido.

$$I_{j(t-1)}^+ + I_{jt}^- + \sum_{m \in M} \sum_{o \in O_{mt}} X_{mjto} = d_{jt}^I + I_{jt}^+ + I_{j(t-1)}^-, \quad \forall j \in J, \forall t \in T. \quad (4.5)$$

Lotes mínimo e máximo. Os lotes mínimo e máximo são definidos, respectivamente, por (4.6) e (4.7). O lote mínimo tem que ser maior ou igual a LB_j , que é a o máximo entre a quantidade mínima de um *kit* de bebida e a quantidade mínima para cobrir as hélices dos tanques. O lote máximo é o mínimo entre UB_j e o tempo máximo disponível de envase antes da realização de uma limpeza temporal. Com isso garante-se que lotes com o tempo de envase ultrapassando TL_{max} não sejam preparados. As restrições (4.7) também garantem que só haverá produção do item j na linha m no período t no lote o se o conjunto tanque preparatório/linha estiver preparado para a produção desse item ($Y_{mjto} = 1$).

$$\rho X_{mjto} \geq LB_j Y_{mjto}, \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in O_{mt}. \quad (4.6)$$

$$\rho X_{mjto} \leq \min\{UB_j, S_m TL_{max}\} Y_{mjto}, \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in O_{mt}. \quad (4.7)$$

Lotes ociosos e item fantasma. As restrições (4.8) eliminam simetrias nas soluções do modelo, empurrando todos os lotes ociosos para o final do período. Logo, só pode haver produção no lote o se tiver havido produção no lote $o - 1$. Para garantir que o item fantasma seja produzido uma única vez no período, são utilizadas as restrições (4.9). Pela estrutura do modelo, essas restrições poderiam ser eliminadas, porém testes computacionais preliminares mostraram que (4.9) acelera a resolução do modelo. O item fantasma é utilizado como a origem do caminho hamiltoniano que definirá o sequenciamento dos itens no período. Desta forma, não há necessidade de definir *a priori* o primeiro item que será produzido em cada período, e além disso, a troca do item fantasma i_0 para qualquer outro item garante que sempre seja realizada a primeira limpeza no início de cada período. Para isso, define-se que o tempo e custo de troca de i_0 para qualquer j são o tempo e custo da primeira limpeza do período. Uma estratégia que poderia ser utilizada é descontar da capacidade o tempo dessa limpeza, porém uma vez que o item fantasma já

está sendo usado para garantir o sequenciamento dos itens, este também é utilizado para garantir a primeira limpeza do período.

$$Y_{mjt(o-1)} \geq Y_{mjto}, \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in O_{mt}, o > 1. \quad (4.8)$$

$$\sum_{o \in O_{mt}} Y_{mioto} \leq 1, \quad \forall m \in M, \forall t \in T. \quad (4.9)$$

Conservação de fluxo e sequenciamento dos lotes. As restrições de conservação de fluxo são dadas por (4.10). Para que o item fantasma seja sempre o primeiro item produzido no período, são acrescentadas as restrições (4.11). As restrições (4.12) impedem que haja mais de uma troca para o mesmo sabor no mesmo período. Para garantir a eliminação de subrotas (no caso do sequenciamento dos lotes, subsequências desconexas), são acrescentadas as restrições do tipo MTZ (*Miller, Tucker e Zemlin*) (4.13) (ÖNCAN; ALTINEL; LAPORTE, 2009).

$$\sum_{\substack{i \in J \\ i \neq j}} Z_{mijt} = \sum_{\substack{i \in J \\ i \neq j}} Z_{mjit}, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (4.10)$$

$$\sum_{\substack{j \in J \\ j \neq i_0}} Z_{mi_0jt} \geq \sum_{\substack{i \in J \\ i \neq k}} Z_{mikt}, \quad \forall m \in M, \forall k \in J, \forall t \in T. \quad (4.11)$$

$$\sum_{\substack{j \in J \\ j \neq i}} Z_{mijt} \leq 1, \quad \forall m \in M, \forall i \in J, \forall t \in T. \quad (4.12)$$

$$V_{mjt} \geq (V_{mit} + 1) - (|J| - 1)(1 - Z_{mijt}), \quad \forall t \in T, \forall m \in M, \forall i, j \in J \setminus i_0, i \neq j. \quad (4.13)$$

O sequenciamento da produção é determinado pelas variáveis Z_{mijt} . Uma vez que o item j é produzido no período t , para cada lote o ($o = 1, \dots, O_{mt}$) as variáveis binárias Y_{mjot} captam a informação se há ou não a produção do item j no lote o . A Figura 11 exemplifica o funcionamento dessas variáveis, sendo que as setas que saem e voltam para um mesmo nó (item) indicam o número de lotes produzidos para o item correspondente. Logo, para cada item j podem ser produzidos de 1 à, no máximo, $|O_{mt}|$ lotes em cada período. A garantia de que todos os lotes de um mesmo sabor são produzidos em sequência é dada pelas restrições (4.15), (4.19) e (4.20) (mais detalhes adiante na explicação das restrições (4.20)).

Setup. As restrições (4.14) garantem que o tanque preparatório/linha m só estará preparado para produzir o item j se tiver havido alguma troca de um item i para j , com $i \neq j$. Além disso, essas restrições limitam em $|O_{mt}|$ o número de lotes do item j no tanque preparatório/linha m no período t .

$$\sum_{o \in O_{mt}} Y_{mjto} \leq |O_{mt}| \sum_{\substack{i \in J \\ i \neq j}} Z_{mijt}, \quad \forall m \in M, \forall j \in J, \forall t \in T, j \neq i_0. \quad (4.14)$$

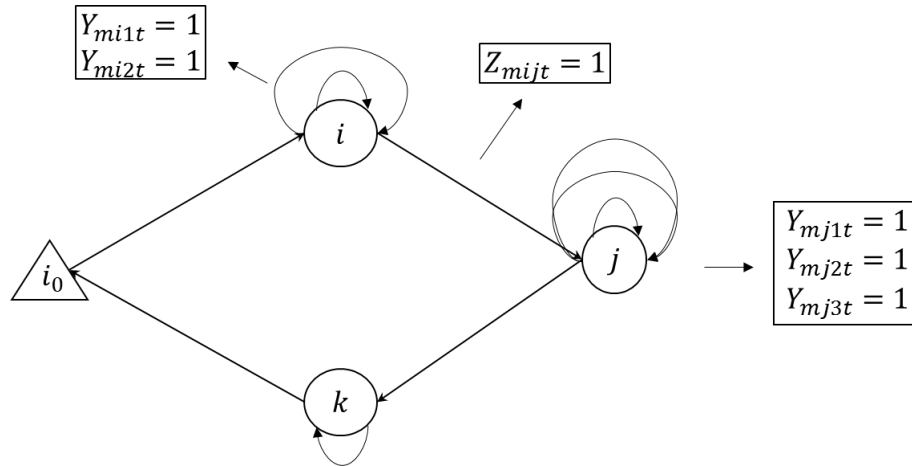


Figura 11 – Representação de como as variáveis Y_{mjt} e Z_{mijt} capturam o sequenciamento dos itens e a produção dos lotes. Fonte: a autora.

A partir das restrições (4.15) a seguir, são apresentadas as restrições que controlam os tempos de início e de final dos lotes, além de garantirem a sincronia entre os dois estágios.

Limpezas no início de cada período. As restrições (4.15) asseguram que o primeiro lote do período t , que pode ser do sabor j , será produzido no estágio I após a primeira limpeza do período. Nesse caso, considera-se que o tempo da primeira limpeza do período é o mesmo de uma limpeza temporal, LT^I .

$$\mu_{mj1t}^{Is} \geq LT^I Z_{mij0t}, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (4.15)$$

O início do envase do primeiro lote do período ocorre após o máximo entre o término da limpeza na linha e o tempo de preparo do lote no estágio I. As restrições (4.16) modelam este início. Na linha também considera-se que o tempo da primeira limpeza do período é idêntico ao de uma limpeza temporal LT^{II} .

$$\mu_{mj1t}^{IIs} \geq \max\{LT^{II} Z_{mij0t}, \mu_{mj1t}^{Ie}\}, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (4.16)$$

Observe que está implícita a possibilidade de espera da linha pelo lote a ser envasado, ou do lote preparado pela linha. Veja a Figura 12 que representa as esperas do tipo (A) e (E) descritas no Capítulo 3. As esperas representadas em (a) e (b) são do tipo (A) e podem ocorrer quando o tempo de limpar a linha é maior do que o tempo de limpar o tanque preparatório mais o tempo de preparo do primeiro lote. A espera pode acontecer no primeiro estágio depois do preparo do primeiro lote, situação (a), ou ser antecipada para antes da limpeza, como representado na situação (b). A espera da linha pelo tanque, do tipo (E) é representada na Figura 12 pela situação (c). Caso o tempo de limpar a linha seja menor do que o tempo de limpar o tanque preparatório e preparar o primeiro lote, a linha fica esperando por esse lote. Nesse exemplo a espera está adiantada antes da limpeza da linha, mas poderia também estar entre a limpeza e o envase do lote 1.

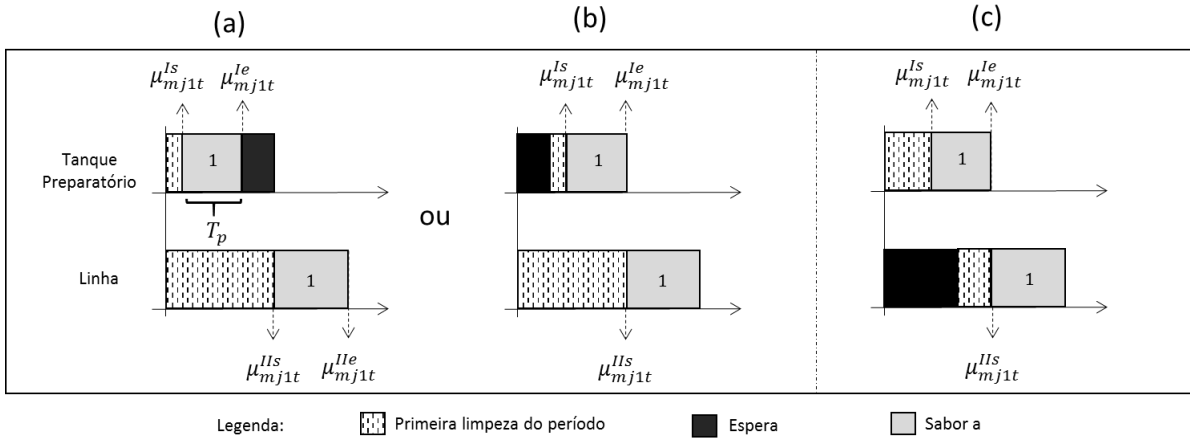


Figura 12 – Limpezas do início do período. Restrições (4.15) e (4.16). Fonte: a autora.

Da maneira que o problema foi modelado, as esperas não são contadas de forma explícita, e são garantidas pelas restrições (4.16) e pelas restrições (4.23) e (4.24) definidas adiante. Existem trabalhos na literatura que consideram esperas de forma explícita por meio de uma variável de espera, como por exemplo (FERREIRA; MORABITO; RANGEL, 2009).

Dimensionamento dos lotes. O instante que o lote o é finalizado no tanque preparatório m é a soma do instante de início dele (μ_{mjto}^{Is}) com o tempo de produzi-lo (preparar a bebida) (Tp), conforme restrições (4.17). Veja na situação (a) da Figura 12 que ilustra o início e término do lote 1, por exemplo. Da mesma forma, na linha, o tempo final do lote é o instante de início de envase desse lote (μ_{mjto}^{IIs}) mais o tempo de envasá-lo (X_{mjto}/S_m), garantido pelas restrições (4.18).

$$\mu_{mjto}^{Ie} = \mu_{mjto}^{Is} + Tp Y_{mjto}, \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in O_{mt}. \quad (4.17)$$

$$\mu_{mjto}^{IIe} = \mu_{mjto}^{IIs} + (X_{mjto}/S_m), \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in O_{mt}. \quad (4.18)$$

Ordenação e não sobreposição dos lotes. As restrições (4.19) e (4.20) garantem, respectivamente, que para o estágio I e para o estágio II um lote o só pode ser iniciado depois da finalização do lote anterior $o-1$, conforme representado na Figura 13. Com isso, é garantida a ordenação dos lotes e a não sobreposição entre eles com relação ao início do lote o e final do lote $o-1$.

$$\mu_{mjto}^{Is} \geq \mu_{mj(o-1)t}^{Ie}, \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in O_{mt}, o > 1. \quad (4.19)$$

$$\mu_{mjto}^{IIs} \geq \mu_{mj(o-1)t}^{IIe}, \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in O_{mt}, o > 1. \quad (4.20)$$

Troca entre itens. As restrições (4.21) determinam que se houver uma troca do item i para o item j no estágio I, o tempo de início do primeiro lote do item j deve ser posterior ao final do último lote $O = |O_{mt}|$ do item i , mais o tempo de troca, conforme representado na Figura 14. Analogamente, as restrições (4.22) funcionam para o estágio II.

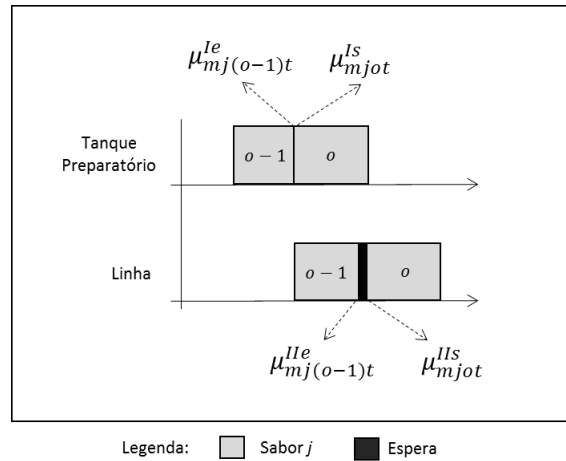


Figura 13 – Ordenação e não sobreposição dos lotes. Restrições (4.19) e (4.20). Fonte: a autora.

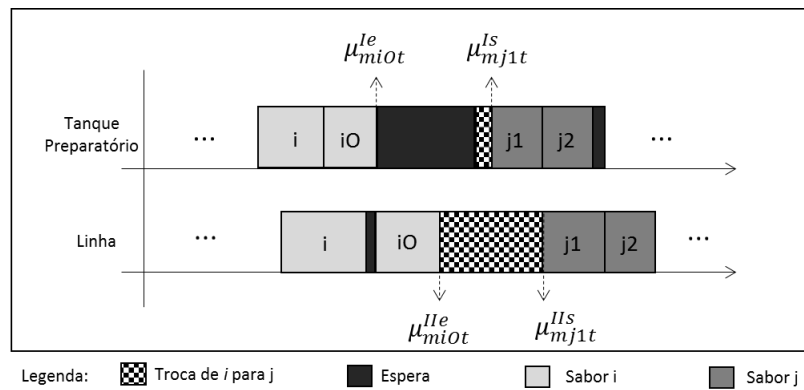


Figura 14 – Troca entre itens. Restrições (4.21) e (4.22). Fonte: a autora.

Note que essas restrições (4.21) e (4.22) serão ativas somente se $Z_{mijt} = 1$, caso contrário ficam redundantes. Em ambas as restrições utiliza-se o instante final do último lote O do item j . Nem todos os lotes de 1 a O são utilizados para cada item na solução final. Supondo que o' foi o último lote em que houve produção, para todos os outros lotes $o > o'$ ($o, o + 1, \dots, O$) em que não há produção, os instantes de início e final desses lotes serão idênticos ao instante de final do lote o' devido às restrições (4.17), (4.18), (4.19) e (4.20).

$$\mu_{mj1t}^{Is} \geq \mu_{miOt}^{Ie} + TC_{ij}^I - Mgde(1 - Z_{mijt}), \quad \forall m \in M, \forall i, j \in J, \forall t \in T, \quad (4.21)$$

$$i \neq j, j \neq i_0.$$

$$\mu_{mj1t}^{IIs} \geq \mu_{miOt}^{IIe} + TC_{ij}^{II} - Mgde(1 - Z_{mijt}), \quad \forall m \in M, \forall i, j \in J, \forall t \in T, \quad (4.22)$$

$$i \neq j, j \neq i_0.$$

Sincronia entre os estágios. As restrições (4.23) asseguram que no período t , o instante de início do envase do lote o do item j na linha m (estágio II) deve ser posterior ao instante final da produção da bebida desse lote no tanque preparatório (estágio I). Em outras palavras, o envase do lote o é posterior ao término da produção desse lote no tanque

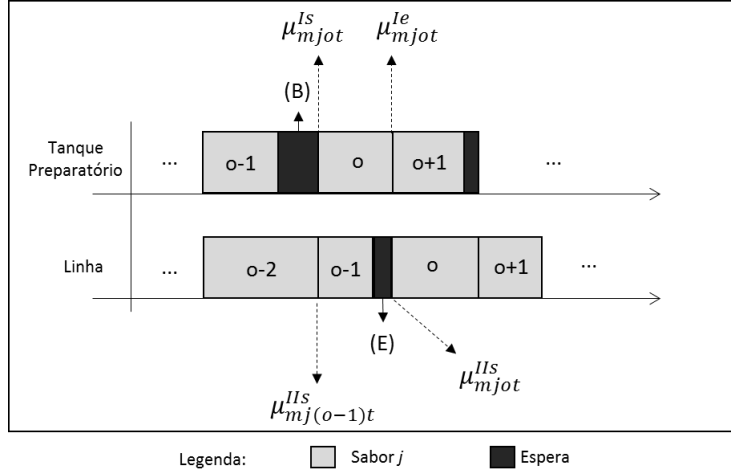


Figura 15 – Sincronia entre os estágios. Restrições (4.23) e (4.24). Fonte: a autora.

preparatório. Assim, são garantidas as esperas do tipo (E) na linha, conforme representado na Figura 15. Da mesma forma, as restrições (4.24) garantem que a produção do lote o no tanque preparatório só pode ser iniciada depois que o lote anterior $o - 1$ foi transferido para a linha e, portanto, o tanque preparatório está vazio para iniciar esse novo lote. Essas restrições são ativadas somente quando há produção do lote o , ou seja, quando $Y_{mjto} = 1$, e garantem as esperas do tipo (B) representadas na Figura 15. Se não houver produção do lote o no estágio I, não existem esperas do tipo (B) no final do lote $o - 1$ do tanque preparatório, pois este será o último lote produzido. Por esse motivo, as restrições (4.24) são ativas somente se existe produção no lote o , isto é, $Y_{mjto} = 1$ e $Mgde(1 - Y_{mjto}) = 0$. É possível associar a variável μ_{mjot}^{IIs} ao instante em que o tanque preparatório está vazio após ter transferido o lote o para a linha.

$$\mu_{mjto}^{IIs} \geq \mu_{mjto}^{Ie}, \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.23)$$

$$\forall o \in O_{mt}, o > 1.$$

$$\mu_{mjto}^{Is} \geq \mu_{mj(o-1)t}^{IIs} + Mgde(1 - Y_{mjto}), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.24)$$

$$\forall o \in O_{mt}, o > 1.$$

Capacidade de produção. A capacidade é garantida impondo que o instante final do último lote O no estágio II, de qualquer sabor, seja menor do que a capacidade disponível, conforme as restrições (4.25).

$$\mu_{mjto}^{IIe} \leq Cap_{mt}, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (4.25)$$

A capacidade do estágio I fica garantida pela estrutura do problema. Devido as restrições (4.23), o lote o sempre é finalizado primeiro no tanque preparatório e depois na linha. Portanto, se estiver garantido que o final do último lote O na linha não ultrapassa o tempo disponível de produção (Cap_{mt}), então esse tempo não será ultrapassado no tanque

preparatório. Logo, as restrições (4.26) podem ser inserida como inequações válidas.

$$\mu_{mjtO}^{Ie} \leq Cap_{mt}, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (4.26)$$

Se o problema considerado não envolvesse limpezas temporais, as restrições (4.5)-(4.26), juntamente com uma modificação da função objetivo (4.4), já definiriam um modelo de dimensionamento e sequenciamento de lotes com as outras considerações descritas no Capítulo 3. Esse modelo é discutido em mais detalhes na Seção 4.1.2.

As restrições que identificam e definem as limpezas temporais são descritas a seguir. Não foi encontrada na literatura nenhuma restrição para abordar essas limpezas temporais da maneira como são definidas nesta pesquisa. Da forma como a identificação dessas limpezas foi modelada, é necessário diferenciar as restrições em dois conjuntos: as que identificam a primeira limpeza temporal de cada item no período (4.27) e (4.30), e as que identificam as demais limpezas para cada item no período (4.33) e (4.35).

Restrições de identificação das limpezas temporais.

Identificação da primeira limpeza temporal do item j no período t . A consideração da primeira limpeza temporal ($k = 1$) do período t que acontece antes do preparo do lote o para o item j no tanque preparatório m é feita pelo conjunto de restrições (4.27). Note que essas restrições referem-se a primeira limpeza temporal do período durante a produção do item j . Note ainda que essas restrições não se referem a primeira limpeza do período.

A Figura 16 é uma adaptação da Figura 9 e ilustra a identificação da necessidade de inclusão da primeira limpeza temporal ($k = 1$) do tanque preparatório m para os itens i e j . Nessa figura, por coincidência ambas as limpezas temporais para os itens i e j acontecem antes do lote $o = 4$. Note que a limpeza deve ocorrer antes de TP_{max} unidades de tempo (minutos ou horas) desde a última limpeza.

As restrições (4.27) só são ativas quando existe produção no lote o , ou seja, quando $Y_{mjto} = 1$, pois anula o termo $Mgde(1 - Y_{mjto})$. A parcela $\mu_{mjtO}^{IIs} - \mu_{mjt1}^{Is}$ desse conjunto de restrições calcula o tempo decorrido desde o início da produção do primeiro lote ($o = 1$) do item j até o instante em que o tanque preparatório está disponível por ter transferido o lote $o - 1$ para a linha (μ_{mjtO}^{IIs}). Enquanto a diferença $\mu_{mjtO}^{IIs} - \mu_{mjt1}^{Is}$ é menor do que TP_{max} , a minimização da variável ω_{mjtO1}^I garante que $\omega_{mjtO1}^I = 0$. Quando essa diferença ultrapassa TP_{max} , tem-se que $\omega_{mjtO1}^I = 1$ e o conjunto de restrições (4.28) torna-se ativa fazendo com que $W_{mjtO1}^I = 1$. Dessa forma, é registrado o lote o em que é necessário incluir a primeira limpeza temporal do item j antes desse lote. A partir desse lote, as restrições (4.28) não estarão mais ativas para $k = 1$. Note que para os lotes consecutivos a o ($o + 1, o + 2, \dots, |O_{mt}|$), ou seja, para $o' > o$, a variável $\omega_{mjtO'1}^I$ deve assumir o valor 1 para que as restrições sejam satisfeitas. Consequentemente, para os lotes consecutivos a o , $W_{mjtO'1}^I = 0$.

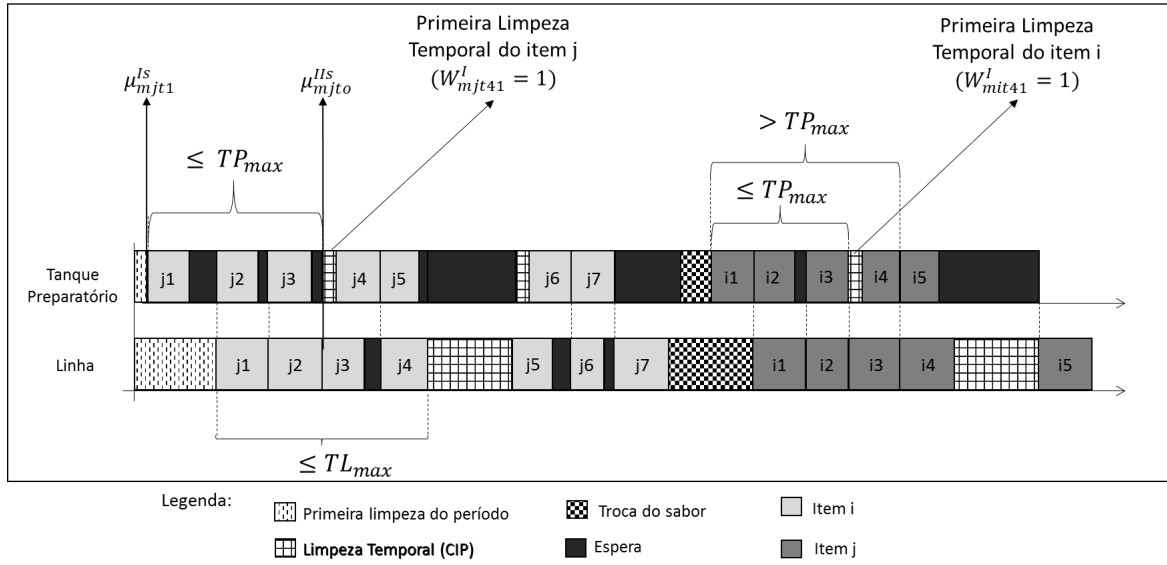


Figura 16 – Ilustração da identificação da necessidade de inclusão da primeira limpeza temporal $k = 1$ do tanque preparatório m para os itens i e j . Fonte: a autora.

As restrições (4.29) representam o fato de que antes do início de um novo item sempre existe uma limpeza (referente a troca de sabor ou ao início do período), ou seja, nunca existe uma limpeza temporal antes do primeiro lote do item j .

$$\mu_{mjt0}^{II_s} - \mu_{mjt1}^{I_s} - Mgde (1 - Y_{mjt0}) \leq Mgde \omega_{mjt01}^I + TP_{max}, \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.27)$$

$$\forall o \in O_{mt}, o > 1.$$

$$W_{mjt0k}^I \geq \omega_{mjt0k}^I - \omega_{mjt(o-1)k}^I, \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.28)$$

$$\forall o \in O_{mt}, \forall k \in Q_{mt}^I, o > 1.$$

$$\omega_{mjt1k}^I = 0, \quad \forall m \in M, \forall j \in J, \quad (4.29)$$

$$\forall t \in T, \forall k \in Q_{mt}^I.$$

As restrições (4.30), (4.31) e (4.32) funcionam de maneira idêntica às (4.27), (4.28) e (4.29), respectivamente, para identificação da primeira limpeza temporal da linha.

A inclusão das variáveis ω_{mjt01}^I e ω_{mjt01}^{II} na função objetivo (4.4) é necessária para a funcionamento das restrições (4.28) e (4.31). De fato, as restrições (4.27), por exemplo, garantem que ω_{mjt01}^I assumam valor 1 quando $\mu_{mjt0}^{II_s} - \mu_{mjt1}^{I_s}$ ultrapassa TP_{max} , mas não garante que ω_{mjt01}^I seja 0 quando $\mu_{mjt0}^{II_s} - \mu_{mjt1}^{I_s} < TP_{max}$. Assim, poderiam acontecer limpezas temporais antes do tempo máximo permitido ser atingido, e dessa maneira a busca pela solução ótima poderia ser dificultada. A minimização dessas variáveis na função objetivo implica que estas assumam o valor 1 somente quando $\mu_{mjt0}^{II_s} - \mu_{mjt1}^{I_s}$ ultrapassa

TP_{max} .

$$\begin{aligned} \mu_{m_j t o}^{IIe} - \mu_{m_j t 1}^{II_s} - Mgde (1 - Y_{mito}) \leq Mgde \omega_{m_j t o 1}^{II} + TP_{max}, \quad \forall m \in M, \forall j \in J, \forall t \in T, \\ \forall o \in O_{mt}, o > 1. \end{aligned} \quad (4.30)$$

$$\begin{aligned} W_{m_j t o l}^{II} \geq \omega_{m_j t o l}^{II} - \omega_{m_j t (o-1)l}^{II}, \quad \forall m \in M, \forall j \in J, \forall t \in T, \\ \forall o \in O_{mt}, \forall l \in Q_{mt}^{II}, o > 1. \end{aligned} \quad (4.31)$$

$$\begin{aligned} \omega_{m_j t 1 k}^{II} = 0, \quad \forall m \in M, \forall j \in J, \\ \forall t \in T, \forall k \in Q_{mt}^I. \end{aligned} \quad (4.32)$$

Identificação das demais limpezas temporais. Uma vez que $\omega_{m_j t o' 1}^I = 1$, tem-se que $\omega_{m_j t o 1}^I = 1$ para todos os outros lotes consecutivos $o > o', \forall o' \in O_{mt}$. Para identificação das próximas, $k = 2, \dots, Q_{mt}^I$, limpezas temporais que podem surgir para o item j , utiliza-se o conjunto de restrições (4.33) que funciona de forma similar às restrições (4.27). Suponha que para o item j no tanque preparatório m no período t antes do lote o' , tenha sido realizada a limpeza temporal $k - 1$, ou seja, $W_{m_j t o' (k-1)}^I = 1$. Neste caso, as restrições (4.33) devem estar ativas quando $W_{m_j t o' (k-1)}^I = 1$ e quando há produção em um lote o ($Y_{m_j t o} = 1$), tal que $o > o'$. Caso $W_{m_j t o' (k-1)}^I = 0$ ou $Y_{m_j t o} = 0$, as restrições são inativa. As restrições também são inativas se $W_{m_j t o' (k-1)}^I = 1$ e $Y_{m_j t o} = 0$, ou se $W_{m_j t o' (k-1)}^I = 0$ e $Y_{m_j t o} = 1$. A parcela $\mu_{m_j t o}^{II_s} - (\psi_{m_j t o' (k-1)}^I + LT^I)$ calcula o tempo de preparo contínuo, incluindo esperas do item j desde o início da produção do primeiro lote o' após a limpeza $k - 1$, até a produção do lote o . Novamente, assim como para o conjunto de restrições (4.27), quando esse valor ultrapassa TP_{max} , a variável binária $\omega_{m_j t o k}^I$ assume o valor 1. A Figura 17 ilustra uma situação infactível, em que ao produzir o lote o de um determinado item j , o tempo desde a última limpeza temporal $k - 1$ ultrapassa o valor TP_{max} , ou seja, é necessário uma limpeza temporal antes do lote o .

$$\begin{aligned} \mu_{m_j t o}^{II_s} - (\psi_{m_j t o' (k-1)}^I + LT^I) - Mgde (2 - W_{m_j t o' (k-1)}^I - Y_{m_j t o}) \leq Mgde \omega_{m_j t o k}^I + TP_{max}, \\ \forall m \in M, \forall j \in J, \forall t \in T, \forall o, o' \in O_{mt}, \forall k \in Q_{mt}^I, o > o', k > 1. \end{aligned} \quad (4.33)$$

O conjunto de restrições (4.34) garante que se não há limpeza temporal, $W_{m_j t o k}^I = 0$, então o instante de início dessa limpeza é nulo, $\psi_{m_j t o k}^I = 0$.

$$\begin{aligned} \psi_{m_j t o k}^I \leq Mgde W_{m_j t o k}^I, \quad \forall m \in M, \forall j \in J, \forall t \in T, \\ \forall o \in O_{mt}, \forall k \in K_{mt}^I. \end{aligned} \quad (4.34)$$

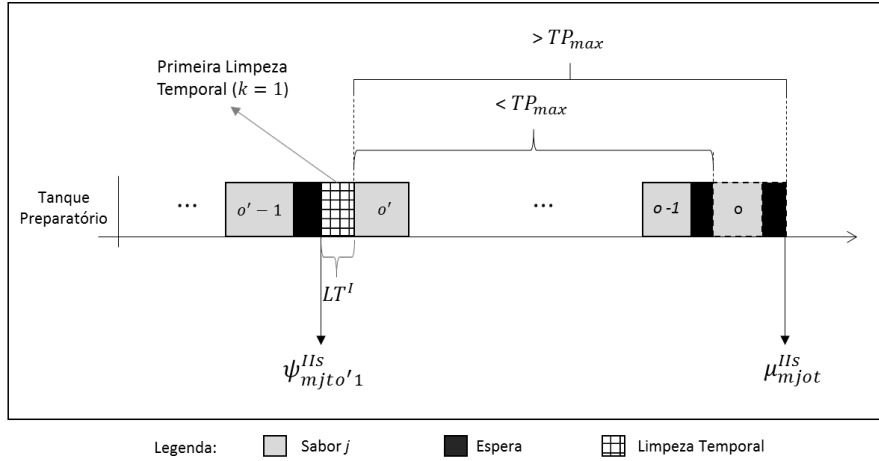


Figura 17 – Programação infactível do tanque preparatório representando a necessidade de uma limpeza temporal k . Fonte: a autora.

As restrições (4.35) e (4.36) referem-se às limpezas temporais do estágio II e funcionam de maneira análoga às restrições (4.33) e (4.34), respectivamente.

$$\begin{aligned} \mu_{mjto}^{IIe} - (\psi_{mjt o' (l-1)}^{II} + LT^{II}) - Mgde (2 - W_{mjt o' (l-1)}^{II} - Y_{mjto}) \leq Mgde \omega_{mjt ol}^{II} + TL_{max}, \\ \forall m \in M, \forall j \in J, \forall t \in T, \forall o, o' \in O_{mt}, \forall l \in Q_{mt}^{II}, o > o', l > 1. \end{aligned} \quad (4.35)$$

$$\begin{aligned} \psi_{mjt ol}^{II} \leq Mgde W_{mjt ol}^{II}, \quad \forall m \in M, \forall j \in J, \forall t \in T, \\ \forall o \in O_{mt}, \forall l \in Q_{mt}^{II}. \end{aligned} \quad (4.36)$$

Restrições de inclusão das limpezas temporais.

Inclusão das limpezas temporais no estágio I. Diferentemente da Figura 17, a Figura 18 representa uma programação factível com uma limpeza temporal k que acontece antes do lote o . Note que essa limpeza temporal é iniciada somente depois que o lote $o - 1$ foi enviado para o estágio II, e assim o estágio I está liberado. As restrições (4.37) garantem que o instante de início da limpeza temporal k , que deve acontecer antes do lote o , sabor j que está sendo preparado no tanque preparatório m no período t , é anterior ao instante de início do lote o , menos o tempo de realização da limpeza temporal. Da mesma forma, as restrições (4.38) garantem que o início da limpeza temporal k é posterior ao início do lote o , menos o tempo da limpeza temporal. De fato, as restrições (4.37) e (4.38) juntas asseguram que o instante de início do lote o (μ_{mjto}^{Is}) acontece exatamente no instante em que a limpeza temporal $\psi_{mjt ok}^I$ termina. Ambas são ativas somente quando $W_{mjt ok}^I = 1$. As restrições (4.39) garantem que a sincronia seja mantida.

$$\mu_{mjto}^{Is} \geq \psi_{mjtok}^I + LT^I W_{mjtok}^I - Mgde (1 - W_{mjtok}^I), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.37)$$

$$\forall o \in O_{mt}, \forall k \in Q_{mt}^I.$$

$$\mu_{mjto}^{Is} \leq \psi_{mjtok}^I + LT^I W_{mjtok}^I - Mgde (1 - W_{mjtok}^I), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.38)$$

$$\forall o \in O_{mt}, \forall k \in Q_{mt}^I.$$

$$\psi_{mjtok}^I \geq \mu_{mjt(o-1)}^{IIs} - Mgde (1 - W_{mjtok}^I), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.39)$$

$$\forall o \in O_{mt}, \forall k \in K_{mt}^I, o > 1.$$

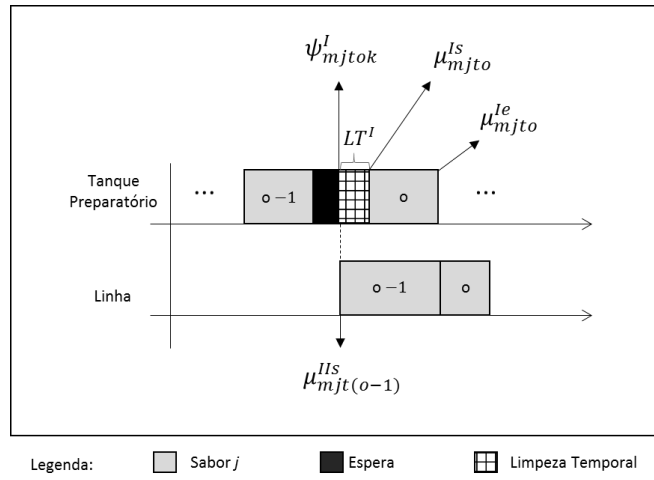


Figura 18 – Representação de uma programação factível com a inclusão da limpeza temporal. Fonte: a autora.

Inclusão das limpezas temporais no estágio II. As restrições (4.40) e (4.41) funcionam de maneira análoga às restrições (4.37) e (4.38), porém para o estágio II. As restrições (4.42) garantem a não sobreposição entre o final do lote $o-1$ e o início da limpeza temporal k , antes do lote o na linha.

$$\mu_{mjto}^{IIs} \geq \psi_{mjtol}^{II} + LT^{II} W_{mjtol}^{II} - Mgde (1 - W_{mjtol}^{II}), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.40)$$

$$\forall o \in O_{mt}, \forall l \in Q_{mt}^{II}.$$

$$\mu_{mjto}^{IIs} \leq \psi_{mjtol}^{II} + LT^{II} W_{mjtol}^{II} - Mgde (1 - W_{mjtol}^{II}), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.41)$$

$$\forall o \in O_{mt}, \forall l \in Q_{mt}^{II}.$$

$$\psi_{mjtok}^{II} \geq \mu_{mjt(o-1)}^{IIe} - Mgde (1 - W_{mjtok}^{II}), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.42)$$

$$\forall o \in O_{mt}, \forall l \in Q_{mt}^{II}, o > 1.$$

Domínio das variáveis.

O domínio das variáveis é dado pela restrição (4.43).

$$X_{mjto}, I_{jt}^+, I_{jt}^-, V_{mjt}, \mu_{mjto}^{Is}, \mu_{mjto}^{Ie}, \mu_{mjto}^{IIs}, \mu_{mjto}^{IIe}, \psi_{mjtok}^I, \psi_{mjtol}^{II} \geq 0;$$

$$W_{mjtok}^I, W_{mjtol}^{II} \in \mathbb{Z}_+; \quad Z_{mijt}, Y_{mjto}, \omega_{mjtok}^I, \omega_{mjtol}^{II} \in \{0, 1\}$$

$$\forall m \in M, \forall i, j \in J, \forall t \in T, \forall o \in O_{mt}, \forall k \in Q_{mt}^I, \forall l \in Q_{mt}^{II}. \quad (4.43)$$

O modelo matemático completo (4.4)-(4.43) é rerepresentado no Apêndice D, para conveniência do leitor.

Embora as variáveis W_{mjtok}^I e W_{mjtol}^{II} tenham sido definidas como inteiras, o domínio dessas poderia ser ainda binário ou real. De fato, devido às variáveis ω_{mjtok}^I e ω_{mjtol}^{II} serem binárias, à minimização das variáveis W_{mjtok}^I e W_{mjtol}^{II} na função objetivo e às restrições (4.28) e (4.31), essas variáveis sempre assumirão valores 0 ou 1 para qualquer um desses domínios. Entretanto, testes computacionais preliminares mostraram que a melhor opção é que W_{mjtok}^I e W_{mjtol}^{II} sejam definidas como inteiras.

O número de variáveis reais (Var_Real) presentes no modelo (4.4)-(4.43) é dado pela equação (4.44) e o número de variáveis binárias (Var_Bin) pela equação (4.45).

$$\begin{aligned} Var_Real = & |M| * |J| * |T| * \left[1 + \sum_t \sum_m |O_{mt}| * \left(5 + \sum_t \sum_m (|Q_{mt}^I| + |Q_{mt}^{II}|) \right) \right] \\ & + 2 * |J| * |T|. \end{aligned} \quad (4.44)$$

$$Var_Bin = |M| * |J| * |T| * \left[|J| + \sum_t \sum_m |O_{mt}| * \left(1 + 2 * \sum_t \sum_m (|Q_{mt}^I| + |Q_{mt}^{II}|) \right) \right]. \quad (4.45)$$

O número de restrições do modelo (4.4)-(4.43) é dado pela equação (4.46) abaixo. Na Tabela 6 está explicitada a parcela com que cada conjunto de restrição colabora para o número total de restrições. Essa equação é obtida através da soma de todas essas parcelas.

$$\begin{aligned} Total_Rest = & |J| * |T| + |M| * |T| * |J - 1|^2 + |M| * |J| * |T| * \left\{ 6 + \sum_t \sum_m |O_{mt}| \right. \\ & * \left[1 + \left(\sum_t \sum_m \left(\frac{|O_{mt}|^2 - |O_{mt}|}{2} \right) * \sum_t \sum_m (|Q_{mt}^I| + |Q_{mt}^{II}| - 2) \right) \right. \\ & + \sum_t \sum_m (|Q_{mt}^I| + |Q_{mt}^{II}|) \left. \right] + \left(\sum_t \sum_m (|O_{mt}| - 1) * \sum_t \sum_m (|Q_{mt}^I| + |Q_{mt}^{II}|) \right) \\ & \left. + \sum_t \sum_m |Q_{mt}^{II}| \right\}. \end{aligned} \quad (4.46)$$

O modelo MDSSL-2E-LT (4.4)-(4.43) é difícil de ser resolvido por pacotes de otimização, até para instâncias pequenas. No Capítulo 6, testes computacionais realizados com esse modelo são apresentados em detalhes. Alguns testes iniciais com esse modelo também são apresentados em Toscano, Ferreira e Morabito (2016). Note que o número de restrições é significativo e que a cardinalidade do conjunto O_{mt} tem um peso importante no número de restrições e de variáveis.

Tabela 6 – Dimensão de cada conjunto de restrição do modelo (4.4)-(4.43).

Restrição	Dimensão
(4.5)	$ J * T $
(4.6), (4.7)	$ M * J * T * \sum_t \sum_m O_{mt} $
(4.8), (4.19), (4.20) (4.23), (4.24), (4.27) (4.30)	$ M * J * T * \sum_t \sum_m (O_{mt} - 1)$
(4.9)	$ M * T $
(4.10), (4.11), (4.12) (4.15), (4.16), (4.16) (4.18), (4.25), (4.26)	$ M * J * T $
(4.13), (4.21), (4.22)	$ M * T * [J - 1 * J - 1 - (J - 1)]$
(4.14)	$ M * (J - 1) * T $
(4.28), (4.39)	$ M * J * T * \sum_t \sum_m (O_{mt} - 1) * \sum_t \sum_m Q_{mt}^I $
(4.29)	$ M * J * T * \sum_t \sum_m Q_{mt}^I $
(4.31), (4.42)	$ M * J * T * \sum_t \sum_m (O_{mt} - 1) * \sum_t \sum_m Q_{mt}^{II} $
(4.32)	$ M * J * T * \sum_t \sum_m Q_{mt}^{II} $
(4.33)	$ M * J * T * \sum_t \sum_m O_{mt} * \sum_t \sum_m (Q_{mt}^I - 1) * \sum_t \sum_m \left(\frac{ O_{mt} ^2 - O_{mt} }{2} \right)$
(4.34), (4.37), (4.38)	$ M * J * T * \sum_t \sum_m O_{mt} * \sum_t \sum_m Q_{mt}^I $
(4.35)	$ M * J * T * \sum_t \sum_m O_{mt} * \sum_t \sum_m (Q_{mt}^{II} - 1) * \sum_t \sum_m \left(\frac{ O_{mt} ^2 - O_{mt} }{2} \right)$
(4.36), (4.40), (4.41)	$ M * J * T * \sum_t \sum_m O_{mt} * \sum_t \sum_m Q_{mt}^{II} $

Restrições adicionais.

Algumas inequações válidas podem ser adicionadas ao modelo com o objetivo de melhorar o desempenho dos *solvers* ao resolvê-lo. É sabido que antes da produção de cada lote o , pode ocorrer uma única limpeza k em cada tanque preparatório m do período t . Logo, as restrições (4.47) representam essa condição. As restrições (4.48) representam a mesma situação para o estágio II.

$$\sum_{k \in Q_{mt}^I} W_{mjtok}^I \leq 1, \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in O_{mt}, o > 1. \quad (4.47)$$

$$\sum_{l \in Q_{mt}^{II}} W_{mjtol}^{II} \leq 1, \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in O_{mt}, o > 1. \quad (4.48)$$

Da mesma forma, se houver uma limpeza k , ela ocorre antes de um único lote o , em cada tanque preparatório m do período t . O mesmo ocorre para cada linha m . As restrições (4.49) e (4.50) configuram essas situações para tanque preparatório e linha, respectivamente.

$$\sum_{\substack{o \in O_{mt} \\ o > 1}} W_{m_j t o k}^I \leq 1, \quad \forall m \in M, \forall j \in J, \forall t \in T, k \in Q_{mt}^I. \quad (4.49)$$

$$\sum_{\substack{o \in O_{mt} \\ o > 1}} W_{m_j t o l}^{II} \leq 1, \quad \forall m \in M, \forall j \in J, \forall t \in T, l \in Q_{mt}^{II}. \quad (4.50)$$

De acordo com as restrições (4.27), (4.28), (4.29) e (4.33), durante a produção do item j no tanque preparatório m no período t , se o tempo decorrido desde a realização da última limpeza temporal $k-1$ ultrapassa TP_{max} com a produção do lote o , então a variável $\omega_{m_j t o k}^I$ vale 1. Logo, antes da produção do lote o , o tempo desde a última limpeza temporal não ultrapassa TP_{max} , e portanto $\omega_{m_j t (o-1)k}^I = \omega_{m_j t (o-2)k}^I = \dots = \omega_{m_j t 1k}^I = 0$. Da mesma forma, tem-se que $\omega_{m_j t (o+1)k}^I = \omega_{m_j t (o+2)k}^I = \dots = \omega_{m_j t |O|k}^I = 1$. Portanto as restrições (4.51) e (4.52) cortam soluções que não tenham esse comportamento. A situação é a mesma para o segundo estágio, e assim as restrições (4.53) e (4.54) são para as variáveis da linha.

$$\omega_{m_j t o k}^I \leq \omega_{m_j t (o+1)k}^I \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in 1, \dots, (|O_{mt}| - 1). \quad (4.51)$$

$$\omega_{m_j t o k}^I \geq \omega_{m_j t (o-1)k}^I \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in O_{mt}, o > 1. \quad (4.52)$$

$$\omega_{m_j t o k}^{II} \leq \omega_{m_j t (o+1)k}^{II} \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in 1, \dots, (|O_{mt}| - 1). \quad (4.53)$$

$$\omega_{m_j t o k}^{II} \geq \omega_{m_j t (o-1)k}^{II} \quad \forall m \in M, \forall j \in J, \forall t \in T, \forall o \in O_{mt}, o > 1. \quad (4.54)$$

Testes iniciais foram realizados para avaliação das inequações válidas (4.47)-(4.54) e as que permitiram um melhor desempenho do *solver* foram: (4.51), (4.52), (4.53) e (4.54). Essas inequações foram adicionadas ao modelo MDSL-2E-LT.

4.1.1 Exemplos Ilustrativos

Para um melhor entendimento do modelo MDSL-2E-LT (4.4)-(4.43), são apresentadas as soluções de dois exemplos ilustrativos (pequenos) nesta seção. Na Tabela 7 estão descritos alguns valores de parâmetros dos exemplos ilustrativos. Na Tabela 8 está apresentada a demanda em unidades para cada item j em cada período t e na Tabela 9, os lotes máximo e mínimo em litros para cada item. Nessas tabelas o item fantasma está representado por i_0 . Os custos de troca do item i para o item j são mostrados na Tabela 10. Os tempos de troca em minutos de i para j , para os estágios I e II, estão na Tabela 11. Na Tabela 10, a penalização da troca de i_0 para qualquer outro item vale 1, a mesma penalização de uma limpeza temporal, no caso a primeira do período. O custo de troca de qualquer item para i_0 é sempre nulo. A mesma observação pode ser feita com os tempos de troca apresentados na Tabela 11.

O primeiro exemplar foi gerado com os dados reais apresentados nas Tabelas de 7 a 11. No segundo exemplar, o lote máximo dos tanques UB_j foi reduzido em 50%, o

tempo de preparo foi tomado como 110 minutos, o valor de TP_{max} modificado para 250 minutos e TL_{max} alterado para 350 minutos. Essas alterações foram realizadas para o segundo exemplar com o intuito de observar uma situação em que a capacidade é restrita no estágio I (acontecem esperas no estágio II), uma vez que para o primeiro exemplar há capacidade restrita no estágio II.

Esses exemplares foram resolvidos pelo *solver* CPLEX 12.6.1, sob a linguagem de modelagem AMPL. O critério de parada foi o valor do *gap* menor ou igual a 1%. Para os ambos os testes, esse valor de *gap* foi atingido em menos de 10 minutos. O computador utilizado para esses testes ilustrativos é o mesmo utilizado nos testes computacionais no Capítulo 6, um Intel Core i7 com dois núcleos de 3.6GHz cada e memória de 16GB.

Tabela 7 – Dados dos exemplares ilustrativos.

Parâmetro	Valor	Parâmetro	Valor
$ M $ (tanques preparatórios/ máquinas)	2	TP_{max}	150 minutos
$ J $ (itens)	4 (3 itens + i_0)	TL_{max}	300 minutos
$ T $ (períodos)	2	LT^I	10 minutos
$ O_{1t} , \forall t \in T$	11	LT^{II}	50 minutos
$ O_{2t} , \forall t \in T$	6	$h_j^+, \forall j \in J$	10 u.m.
$ Q_{mt}^I , \forall m \in M, \forall t \in T$	6	$h_j^-, \forall j \in J$	100 u.m.
$ Q_{mt}^{II} , \forall m \in M, \forall t \in T$	3	Clt	1 u.m.
Tp	20 minutos	α	0,001
$Cap_{mt}, \forall m \in M, \forall t \in T$	1.000 minutos	ρ	2,4 litros
		S_1	120 litros/minuto
		S_2	60 litros/minuto

Tabela 8 – Demanda em unidades do exemplar ilustrativo.

j/t	1	2
2	11.000	16.100
3	12.500	21.250
4	17.500	0

Tabela 9 – Lotes máximo e mínimo em litros do exemplar ilustrativo.

j	UB_j	LB_j
2	10.000	2.500
3	12.000	3.000
3	12.000	3.000

Tabela 10 – Custo de troca do item i para o item j do exemplar ilustrativo.

i/j	i_0	2	3	4
i_0	0	1	1	1
2	0	0	2	2
3	0	2	0	4
4	0	4	2	0

Tabela 11 – Tempo de troca em minutos do item i para o item j do exemplar ilustrativo (TC_{ij}^I / TC_{ij}^{II}).

i/j	i_0	2	3	4
i_0	0/0	10/50	10/50	10/50
2	0/0	0/0	10/50	10/50
3	0/0	20/100	0/0	20/100
4	0/0	20/100	10/50	0/0

Exemplar 1: Dados originais.

A Tabela 12 apresenta os valores das variáveis de troca (Z_{mijt}), produção em unidades (X_{mjto}) e de limpezas temporais (W_{mjto}^I e W_{mjto}^{II}) não nulas da solução ótima (com menos de 1% de *gap*) para o exemplar 1. O valor da função objetivo da solução é $Z = 14,006$. Essa solução não tem estoque e nem atraso. Limpezas temporais acontecem nos dois estágios apenas para o tanque preparatório/linha (dedicados) $m = 1$ nos dois períodos.

Tabela 12 – Sequência, lotes de produção e limpezas temporais da solução da instância 1.

	$t = 1$			$t = 2$		
$m = 1$	$Z_{1131} = 1$	$X_{1311} = 1.250$	$W_{12131}^I = 1$	$Z_{1122} = 1$	$X_{1221} = 1.041,67$	$W_{13251}^I = 1$
	$Z_{1321} = 1$	$X_{1312} = 1.250$	$W_{13141}^I = 1$	$Z_{1232} = 1$	$X_{1222} = 3.375$	$W_{13251}^{II} = 1$
	$Z_{1241} = 1$	$X_{1313} = 5.000$		$Z_{1312} = 1$	$X_{1223} = 1.041,67$	
	$Z_{1411} = 1$	$X_{1314} = 5.000$			$X_{1224} = 1.041,67$	
		$X_{1211} = 2.666,67$			$X_{1225} = 4.166,67$	
		$X_{1212} = 4.166,67$			$X_{1321} = 4.000$	
		$X_{1213} = 4.166,67$			$X_{1322} = 1.250$	
		$X_{1411} = 1.250$			$X_{1323} = 1.250$	
		$X_{1412} = 5.000$			$X_{1324} = 4.750$	
		$X_{1413} = 3.750$			$X_{1325} = 5.000$	
				$X_{1326} = 5.000$		
$m = 2$	$Z_{2141} = 1$	$X_{2411} = 1.250$		$Z_{2122} = 1$	$X_{2221} = 3.250$	
	$Z_{2411} = 1$	$X_{2412} = 1.250$		$Z_{2212} = 1$	$X_{2222} = 2.183,33$	
		$X_{2413} = 5.000$				

A Figura 19 apresenta o gráfico de Gantt para os dois períodos do exemplar ($t = 1$ e $t = 2$) para o tanque preparatório e linha $m = 1$. O gráfico de Gantt para $m = 2$ não foi exibido por ser parecido com o da Figura 19. Nessa figura é possível ver que existem períodos ociosos (esperas em que o tanque preparatório está vazio) na linha do tempo do tanque preparatório. Por exemplo, no período 1 existe um tempo ocioso antes da troca do item 3 para o item 2 e antes da troca do item 2 para o item 4. No período 2, existe um tempo ocioso antes da troca do item 2 para o item 3, e também antes da limpeza temporal realizada para o item 3. Nesta solução vê-se que é melhor deixar tempo ocioso do que se adiantar a realização do CIP de troca de sabor ou de limpeza temporal. Vale lembrar que se esses CIPs fossem adiantados no tanque preparatório, o tempo de validade dessa limpeza (TP_{max}) seria alcançado antes e talvez mais limpezas temporais fossem necessárias. O tempo ocioso pode ser visto como uma espera, porém com o tanque vazio. Tanto esses tempos ociosos quanto as esperas não são definidos explicitamente por variáveis, mas podem ser calculados através das variáveis que definem os inícios e finais dos lotes. Na Tabela 13, por exemplo, estão os instantes de início e final de cada lote, para $t = 2$ e $m = 1$ apresentados pelas variáveis de controle de tempo. Os valores da Tabela 13 foram utilizados para fazer o período $t = 2$ do gráfico representado na Figura 19. No

Tabela 13 – Variáveis de tempo referentes ao Gantt da Figura 19.

	Estágio I		Estágio II	
Item 2	$\mu_{1221}^{Is} = 30,00$	$\mu_{1221}^{Ie} = 50,00$	$\mu_{1221}^{IIs} = 50,00$	$\mu_{1221}^{IIe} = 70,83$
	$\mu_{1222}^{Is} = 50,83$	$\mu_{1222}^{Ie} = 70,83$	$\mu_{1222}^{IIs} = 70,83$	$\mu_{1222}^{IIe} = 138,33$
	$\mu_{1223}^{Is} = 70,83$	$\mu_{1223}^{Ie} = 90,83$	$\mu_{1223}^{IIs} = 138,33$	$\mu_{1223}^{IIe} = 159,17$
	$\mu_{1224}^{Is} = 139,17$	$\mu_{1224}^{Ie} = 159,17$	$\mu_{1224}^{IIs} = 159,17$	$\mu_{1224}^{IIe} = 180,00$
	$\mu_{1225}^{Is} = 159,17$	$\mu_{1225}^{Ie} = 179,17$	$\mu_{1225}^{IIs} = 180,00$	$\mu_{1225}^{IIe} = 263,33$
Item 3	$\mu_{1321}^{Is} = 293,33$	$\mu_{1321}^{Ie} = 313,33$	$\mu_{1321}^{IIs} = 313,33$	$\mu_{1321}^{IIe} = 393,33$
	$\mu_{1322}^{Is} = 373,33$	$\mu_{1322}^{Ie} = 393,33$	$\mu_{1322}^{IIs} = 393,33$	$\mu_{1322}^{IIe} = 418,33$
	$\mu_{1323}^{Is} = 393,33$	$\mu_{1323}^{Ie} = 413,33$	$\mu_{1323}^{IIs} = 418,33$	$\mu_{1323}^{IIe} = 443,33$
	$\mu_{1324}^{Is} = 418,33$	$\mu_{1324}^{Ie} = 438,33$	$\mu_{1324}^{IIs} = 443,33$	$\mu_{1324}^{IIe} = 538,33$
	$\psi_{13251}^{Is} = 558,33$		$\psi_{13251}^{IIs} = 538,33$	
	$\mu_{1325}^{Is} = 568,33$	$\mu_{1325}^{Ie} = 588,33$	$\mu_{1325}^{IIs} = 588,33$	$\mu_{1325}^{IIe} = 688,33$
	$\mu_{1326}^{Is} = 588,33$	$\mu_{1326}^{Ie} = 608,33$	$\mu_{1326}^{IIs} = 688,33$	$\mu_{1326}^{IIe} = 788,33$

gráfico os valores estão adicionados em 1.000 unidades (capacidade do primeiro período) por estarem no segundo período. Por exemplo, note que o instante de início do primeiro lote $o = 1$ do item 2 no tanque preparatório é logo após a primeira limpeza do período, $\mu_{1221}^{Is} = 30,00$. No instante que o lote $o = 2$ do item 2 está pronto no tanque preparatório, a linha também já está preparada para recebê-lo, $\mu_{1222}^{Ie} = \mu_{1222}^{IIs} = 70,83$, portanto não há espera. A espera entre os lotes $o = 3$ e $o = 4$ do item 2 no tanque preparatório é dada pela diferença $\mu_{1223}^{IIs} - \mu_{1223}^{Ie} = 138,33 - 90,83 = 47,5$ minutos. O tempo ocioso (espera do tanque preparatório pela linha) depois do lote $o = 5$ é calculada pela diferença $\mu_{1321}^{Is} - \mu_{1225}^{IIs} - TC_{23}^I = 293,33 - 180,00 - 10 = 103,33$. Vale observar também que ocorre uma limpeza temporal antes do lote $o = 5$ do item 3, tanto na linha, quanto no tanque preparatório. No tanque preparatório é necessário pois $\mu_{1324}^{IIs} - \mu_{1321}^{Is} = TP_{max} = 150$ minutos. Na linha, a limpeza aconteceu com $\mu_{1324}^{IIe} - \mu_{1321}^{IIs} = 270$ minutos passados desde a última limpeza (troca de sabor), ou seja, um valor menor do que $TL_{max} = 300$. Convém observar que o tanque preparatório sempre termina a produção antes da linha, porém depois de terminar de preparar o último lote do sabor 3 ($o = 6$), a bebida ainda ficou por $\mu_{1326}^{IIs} - \mu_{1326}^{Ie} = 88$ minutos esperando para ser envasada e o tanque preparatório terminar de operar, de fato, nesse período.

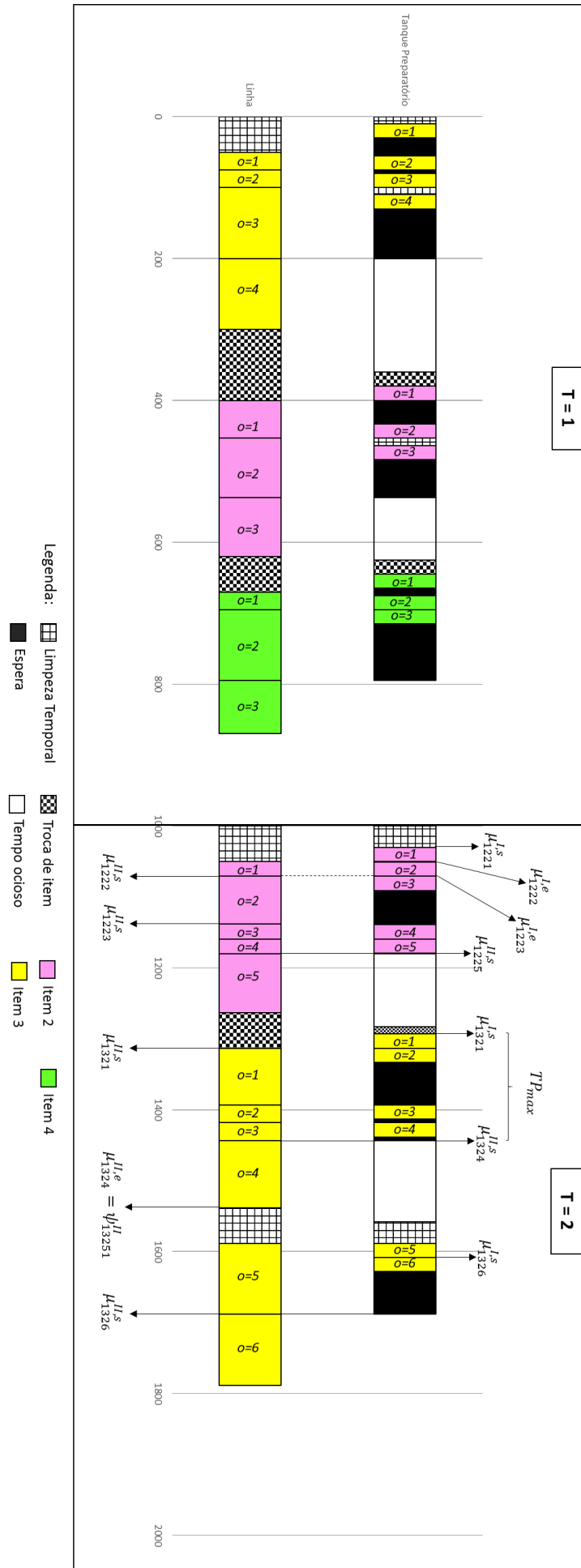


Figura 19 – Teste 1 - Gráfico de Gantt da solução do exemplar 1 para $t = 1$ e $t = 2$ para $m = 1$. Fonte: a autora.

Exemplar 2: A capacidade restrita está no tanque com UB_j reduzido em 50%, $Tp=110$, $TP_{max} = 250$ e $TL_{max} = 350$.

O valor da função objetivo para esse teste é $Z = 1.026.694,67$. O valor da função objetivo é alto em decorrência de atrasos, que ocorrem devido às modificações de parâmetros. Na Tabela 14 estão apresentados os valores das variáveis não nulas de troca, produção, estoque, atraso e limpezas temporais da solução apresentada para a instância 2. O gráfico de Gantt para $t = 1$ e $t = 2$ para o tanque preparatório e linha $m = 1$ é mostrado na Figura 20. As variáveis de controle de tempo do período $t = 1$ utilizadas para o desenho do primeiro período desse gráfico estão apresentadas na Tabela 15.

Tabela 14 – Sequência, lotes de produção e limpezas temporais da solução da instância 2.

	$t = 1$			$t = 2$		
$m = 1$	$Z_{1141} = 1$	$X_{1411} = 2.500$	$W_{14131}^I = 1$	$Z_{1132} = 1$	$X_{1321} = 2.500$	$W_{13231}^I = 1$
	$Z_{1431} = 1$	$X_{1412} = 2.500$	$W_{14152}^I = 1$	$Z_{1322} = 1$	$X_{1322} = 2.500$	$W_{13252}^I = 1$
	$Z_{1311} = 1$	$X_{1413} = 2.500$	$W_{14141}^{II} = 1$	$Z_{1242} = 1$	$X_{1323} = 2.500$	$W_{13241}^{II} = 1$
		$X_{1414} = 2.500$		$Z_{1412} = 1$	$X_{1324} = 2.500$	
		$X_{1415} = 2.500$			$X_{1325} = 2.500$	
		$X_{1416} = 2.500$			$X_{1326} = 2.500$	
		$X_{1311} = 2.500$			$X_{1221} = 2.083,33$	
	$X_{1312} = 2.500$			$X_{1421} = 2.500$		
$m = 2$	$Z_{2121} = 1$	$X_{2211} = 2.083,33$	$W_{22131}^I = 1$	$Z_{2122} = 1$	$X_{2221} = 2.083,33$	$W_{22231}^I = 1$
	$Z_{2231} = 1$	$X_{2212} = 1.750$	$W_{22152}^I = 1$	$Z_{2232} = 1$	$X_{2222} = 2.083,33$	$W_{22252}^I = 1$
	$Z_{2311} = 1$	$X_{2213} = 2.083,33$	$W_{22131}^{II} = 1$	$Z_{2312} = 1$	$X_{2223} = 2.083,33$	$W_{22231}^{II} = 1$
		$X_{2214} = 2.083,33$	$W_{23131}^I = 1$		$X_{2224} = 2.083,33$	$W_{23231}^I = 1$
		$X_{2215} = 2.083,33$			$X_{2225} = 1.916,67$	
		$X_{2311} = 2.500$			$X_{2321} = 2.500$	
		$X_{2312} = 2.500$			$X_{2322} = 2.500$	
	$X_{2313} = 1.416,67$			$X_{2323} = 1.250$		
Atraso	$I_{21}^- = 916,67$			$I_{22}^- = 4.683,33$		
	$I_{31}^- = 1.083,33$			$I_{32}^- = 1.083,33$		
	$I_{41}^- = 2.500$					

É possível notar que neste exemplar o tanque preparatório tem capacidade restrita, pois esperas aparecem na linha. Por exemplo, existe uma espera na linha antes de começar a limpeza do início do período. Uma vez que $\mu_{1411}^{II_s} = 120$ e $LT^{II} = 50$, existe uma espera de 70 minutos. Depois do envase do lote $o = 1$, do item 4, a linha fica esperando o lote $o = 2$, que está sendo preparado no tanque preparatório. Essa espera é calculada pela subtração $\mu_{1412}^{I_e} - \mu_{1411}^{II_e} = 230 - 170 = 60$ minutos. A espera na linha no final do lote $o = 3$ do item 4, acontece antes da limpeza temporal, para aumentar a validade dessa próxima limpeza, e pode ser calculada pela subtração $\psi_{14141}^{II} - \mu_{1413}^{II_e} = 410 - 400 = 10$ minutos. As outras esperas podem ser calculadas de maneira similar a essas e as discutidas para o exemplar 1.

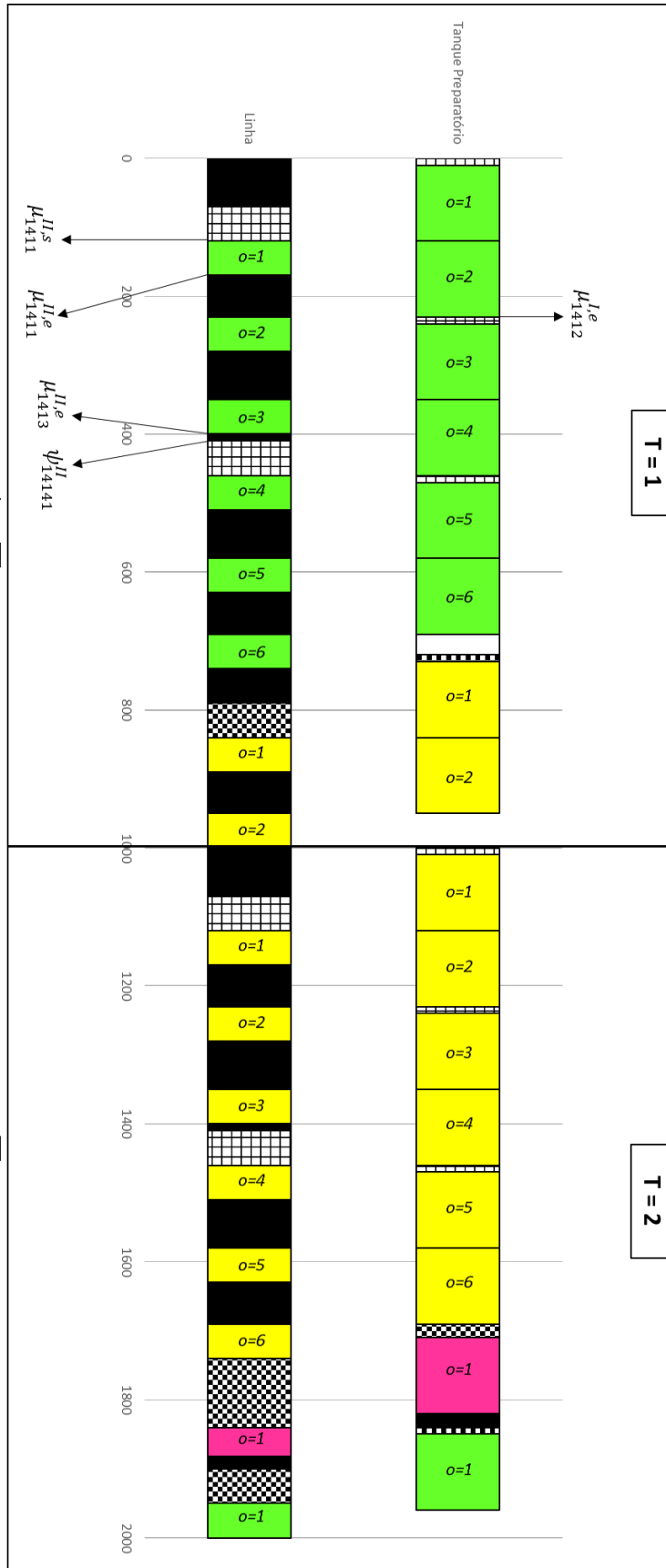


Figura 20 – Teste 4 - Gráfico de Gantt da solução do exemplar 2 para $t = 1$ e $t = 2$ para $m = 1$. Fonte: a autora.

Tabela 15 – Variáveis de tempo referentes ao Gantt da Figura 20

	Estágio I		Estágio II	
Item 4	$\mu_{1411}^{Is} = 10$	$\mu_{1411}^{Ie} = 120$	$\mu_{1411}^{IIs} = 120$	$\mu_{1411}^{IIE} = 170$
	$\mu_{1412}^{Is} = 120$	$\mu_{1412}^{Ie} = 230$	$\mu_{1412}^{IIs} = 230$	$\mu_{1412}^{IIE} = 280$
	$\psi_{14131}^I = 230$			
	$\mu_{1413}^{Is} = 240$	$\mu_{1413}^{Ie} = 350$	$\mu_{1413}^{IIs} = 350$	$\mu_{1413}^{IIE} = 400$
			$\psi_{14141}^{II} = 410$	
	$\mu_{1414}^{Is} = 350$	$\mu_{1414}^{Ie} = 460$	$\mu_{1414}^{IIs} = 460$	$\mu_{1414}^{IIE} = 510$
	$\psi_{14152}^I = 460$			
Item 3	$\mu_{1415}^{Is} = 470$	$\mu_{1415}^{Ie} = 580$	$\mu_{1415}^{IIs} = 580$	$\mu_{1415}^{IIE} = 630$
	$\mu_{1416}^{Is} = 580$	$\mu_{1416}^{Ie} = 690$	$\mu_{1416}^{IIs} = 690$	$\mu_{1416}^{IIE} = 740$
	$\mu_{1311}^{Is} = 730$	$\mu_{1311}^{Ie} = 840$	$\mu_{1311}^{IIs} = 840$	$\mu_{1311}^{IIE} = 890$
	$\mu_{1312}^{Is} = 840$	$\mu_{1312}^{Ie} = 950$	$\mu_{1312}^{IIs} = 950$	$\mu_{1312}^{IIE} = 1000$

4.1.2 Modelo Integrado de Dimensionamento e Sequenciamento de Lotes Dois Estágios (MDSL-2E)

Desconsiderando as limpezas temporais do modelo MDSL-2E-LT, obtém-se o modelo MDSL-2E. Trata-se de um modelo para o problema de dimensionamento e sequenciamento de lotes dois estágios com a existência de um *buffer* (tanque pulmão) no segundo estágio. Ainda é mantida a exigência de uma limpeza no início de cada período. Esse modelo é diferente dos modelos dois estágios para bebidas, como cervejas e refrigerantes (BALDO et al., 2014; FERREIRA; MORABITO; RANGEL, 2009). De fato, nesse modelo a sincronia é considerada de maneira diferente, pois, assim que um tanque de bebida está pronto, este pode ser imediatamente enviado para o segundo estágio, liberando o tanque preparatório para iniciar a produção de um novo lote, enquanto a máquina está envasando o lote anterior. Uma interpretação que pode ser dada a esse modelo é que sempre existem dois tanques dedicados a uma única linha de produção e esses tanques ficam revezando o abastecimento da linha.

Considere as mesmas variáveis e mesmos parâmetros do modelo MDSL-2E-LT, mas desconsidere as variáveis ω_{mjtok}^I , ω_{mjtol}^{II} , W_{mjtok}^I , W_{mjtol}^{II} , ψ_{mjtok}^I e ψ_{mjtol}^{II} . A seguir, o modelo MDSL-2E é definido pela função objetivo (4.55), pelas restrições de (4.5) a (4.24) e pelo domínio das variáveis (4.56).

$$\text{Min } Z = \sum_{t \in T} \sum_{j \in J} (h_j^+ I_{jt}^+ + h_j^- I_{jt}^-) + \sum_{m \in M} \sum_{t \in T} \sum_{j \in J} \sum_{\substack{i \in J \\ i \neq j}} C_{ij} Z_{mijt} \quad (4.55)$$

s.a.:

$$(4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), (4.13), (4.14), (4.15), \\ (4.16), (4.17), (4.18), (4.19), (4.20), (4.21), (4.22), (4.23), (4.24).$$

$$\begin{aligned}
X_{mjt}, I_{jt}^+, I_{jt}^-, V_{mjt}, \mu_{mjt}^{Is}, \mu_{mjt}^{Ie}, \mu_{mjt}^{IIs}, \mu_{mjt}^{IIe} &\geq 0; & W_{mjt}^I, W_{mjt}^{II} &\in \mathbb{Z}^+; \\
Z_{mijt}, Y_{mjt} &\in \{0, 1\}, & \forall m \in M, \forall i, j \in J, \forall t \in T, \forall o \in O_{mt}. & \quad (4.56)
\end{aligned}$$

O modelo MDSL-2E (4.55)-(4.56) é analisado e comparado com o modelo MDSL-2E-LT (4.4)-(4.43) na Seção 6.5.3 do Capítulo 6. O objetivo é investigar o impacto da inclusão das limpezas temporais nesse problema.

5 Métodos heurísticos de programação matemática para o problema

Diversos autores tem proposto métodos heurísticos, métodos de decomposição e métodos híbridos para resolver os problemas de dimensionamento e sequenciamento de lotes nos últimos anos, conforme discutido na revisão bibliográfica do Capítulo 2 (GUIMARÃES; KLABJAN; ALMADA-LOBO, 2013; SEEANNER; MEYR, 2013; XIAO et al., 2015). Isso porque os problemas de dimensionamento de lotes pertencem à classe NP-difícil (BITRAN; YANASSE, 1992) e também por causa dos tamanhos desses problemas em situações reais. Além disso, para representação adequada de situações reais, também são adicionadas aos modelos restrições que em geral complicam ainda mais os problemas, mesmo para a resolução de instâncias relativamente pequenas.

Este é o caso do modelo integrado MDSL-2E-LT (4.4)-(4.43) apresentado no Capítulo 4. Além disso, testes computacionais preliminares (veja Seção 6.3.1) mostraram a dificuldade de um *solver* de última geração (CPLEX) de encontrar soluções de boa qualidade para o modelo MDSL-2E-LT dentro de um tempo computacional razoável para propósitos práticos. Até mesmo para instâncias de pequeno porte, o modelo é difícil de ser resolvido otimamente. Além disso, para instâncias muito grandes o *solver* tem dificuldade até de resolver a relaxação linear do modelo. Por essa razão, neste trabalho estão sendo propostas heurísticas para tentar encontrar boas soluções para o problema em tempos razoáveis, e que sejam melhores do que as encontradas pelo CPLEX para o modelo MDSL-2E-LT dentro de certos tempos limites.

As heurísticas propostas nesse capítulo podem ser classificadas em heurísticas de programação matemática, pois em todas existem interações com modelos matemáticos (ZANAKIS; EVANS; VAZACOPOULOS, 1989). A primeira classe de heurísticas, apresentada na Seção 5.1, explora a decomposição do problema nos dois estágios de produção. Um estágio é resolvido via modelo matemático, e a partir do resultado desse modelo, a solução final sincronizada é obtida de maneira construtiva. Na Seção 5.2, propõem-se heurísticas construtivas em que uma solução factível é obtida pela solução modelos relaxados para o problema, que consideram os dois estágios de produção, mas não considera a sincronia entre eles. A sincronia nessas heurísticas é realizada de forma construtiva. Na Seção 5.3 o modelo MDSL-2E-LT é resolvido de maneira decomposta através de heurísticas construtivas do tipo *relax-and-fix*. Na Seção 5.4 são abordadas heurísticas de melhoria do tipo *fix-and-optimize* para resolver o modelo integrado MDSL-2E-LT. A partir de uma solução inicial, obtida por alguma das heurísticas propostas nas Seções 5.1, 5.2 ou 5.3, as heurísticas *fix-and-optimize* buscam melhorar essa solução.

5.1 Heurísticas de decomposição do problema em estágios

Nesta seção são apresentadas heurísticas de decomposição para solução do problema de programação da produção de bebidas à base de frutas. Dois tipos de heurísticas são propostas: Heurísticas Estágio I (HE1) e Heurísticas Estágio II (HE2). Nessas heurísticas, o problema é decomposto nos dois estágios de produção. Em ambas, um modelo matemático inteiro misto (MIP) é utilizado para resolver a programação da produção de um dos estágios de produção; a solução obtida pelo modelo é sincronizada com o outro estágio através de um passo construtivo que insere na programação da produção os tempos de espera. Logo ao final da construção da sincronia é necessário verificar se a capacidade dos períodos foi excedida ou não. Se a capacidade é satisfeita, a solução é factível e a heurística termina. Caso contrário, as capacidades dos períodos são reduzidas e o MIP é resolvido novamente. O intuito de reduzir a capacidade no modelo é retornar uma solução com mais tempo disponível para a inserção das esperas na fase de sincronia. Esse processo é repetido até que uma solução factível seja encontrada, ou até um limite de tempo. Note que o MIP é sempre factível devido a permissão de atraso. O fluxograma apresentado na Figura 21 representa o funcionamento das heurísticas HE1 e HE2.

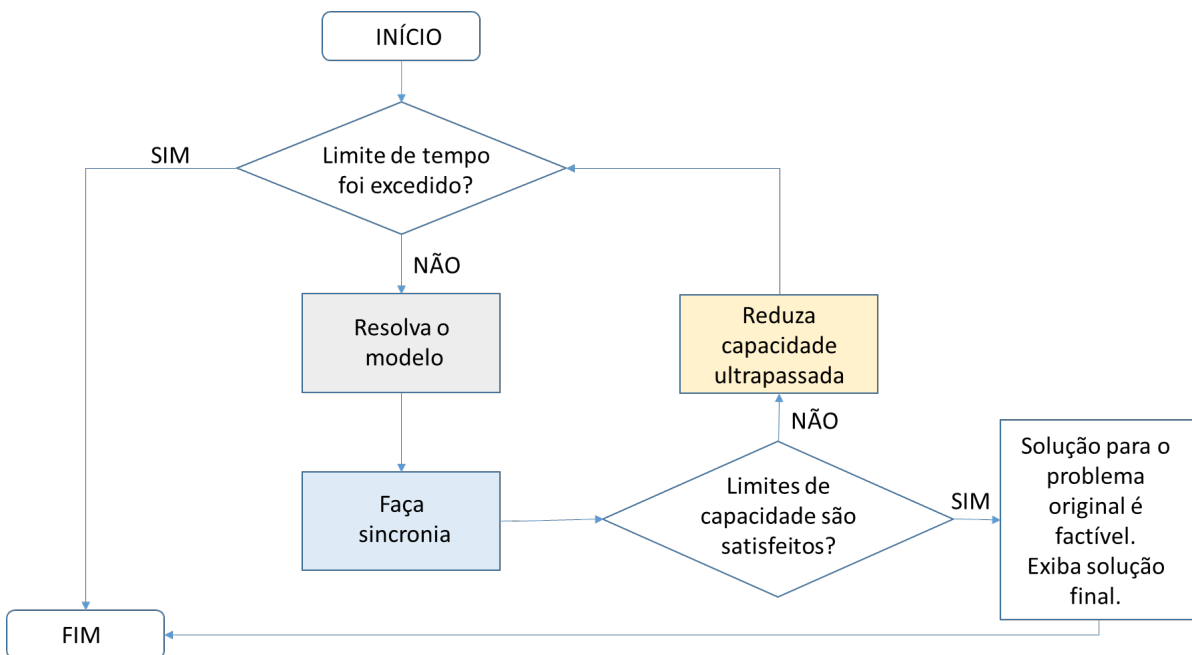


Figura 21 – Fluxograma das heurísticas de decomposição. Fonte: a autora.

Na heurística HE1 o modelo matemático é definido para otimizar o estágio I e na HE2, para otimizar o estágio II. Mais a frente são apresentados esses dois modelos. A Figura 22 representa o funcionamento das heurísticas HE1 e HE2.

Após resolver o modelo é preciso desagregar os lotes de produção apresentados pelo modelo em lotes do tipo o , como do modelo MDSL-2E-LT, para realização da sincronia. Esses lotes o são limitados pelo tamanho do tanque preparatório, conforme explicado no

Capítulo 4. Como visto mais adiante, os modelos para o primeiro estágio (MDSL-P) e (MDL-P) têm uma variável (ξ_{mjt}^I) que já contabiliza o número de lotes o que deve ser utilizado por cada tanque preparatório/linha m para produzir o sabor j no período t . Entretanto, para os modelos do segundo estágio, em que só a linha é levada em consideração, os lotes ainda não estão divididos em tanques preparatórios e essa desagregação precisa ser feita para a etapa de sincronia.

Uma vez desagregados esses lotes em lotes menores o , eles juntamente com suas respectivas sequências de produção são utilizados na fase construtiva. Nessa fase é construída a programação da produção, lote a lote. Na HE1, os lotes obtidos com o modelo para os tanques preparatórios (estágio I) são inseridos na linha, as limpezas temporais inseridas em ambos tanque preparatório e linha, juntamente com os tempos de espera que podem ocorrer entre os dois estágios de produção. O mesmo ocorre com a heurística HE2, porém, nesse caso os lotes e a sequência são definidos pelo modelo da linha. Note que a programação da produção na solução sincronizada depende mais tempo do que na solução dos modelos de cada estágio.

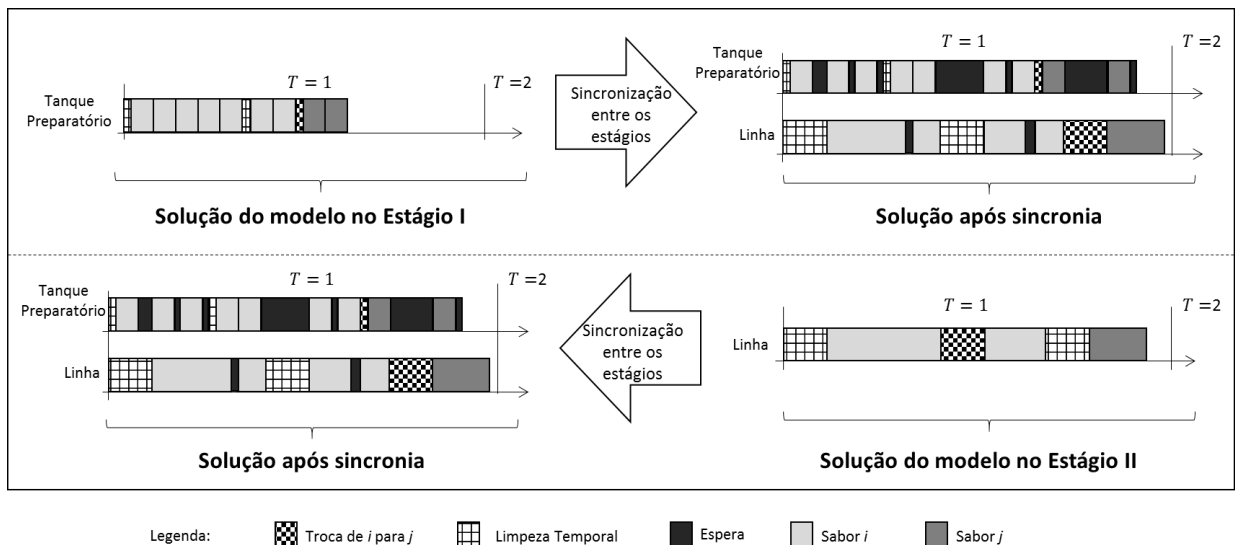


Figura 22 – Exemplo de como as heurísticas HE1 e HE2 resolvem o problema. Fonte: a autora.

A construção da sincronia entre os estágios é dada pelo Algoritmo 1. Considere os parâmetros utilizados para o modelo MDSL-2E-LT no Capítulo 4 e também os seguintes parâmetros:

- TD_{mt}^I e TD_{mt}^{II} são os tempos totais decorridos desde o instante de término da realização da última limpeza (que pode ser a primeira limpeza do período, uma limpeza temporal ou ainda uma limpeza realizada por troca de sabor) para cada par (m, t) , nos estágios I e II respectivamente;
- $CapUtil_{mt}$ é a capacidade utilizada na linha m e no período t após a sincronia;

- $N\text{Lotes}_{mt}$ é o número total de lotes que devem ser produzidos por cada conjunto tanque preparatório/linha m de cada período t (esse número é obtido na desagregação dos lotes obtidos pelo modelo em lotes o , como citado anteriormente);
- Lt_o é a quantidade em litros de cada lote o ;
- SB_o é o sabor de cada lote o ;
- θ_o^I e θ_o^{II} são os tempos de espera que podem ocorrer no final de cada lote o , no tanque preparatório e na linha, respectivamente.

Para realizar a sincronia entre os dois estágios, é necessário controlar os tempos de produção de cada lote, os tempos de limpezas temporais, os tempos de trocas de sabor e os tempos das esperas. A descrição do Algoritmo 1 é feita a seguir:

- Na linha 4 o valor de $N\text{Lotes}_{mt}$ é calculado de acordo com a solução do modelo do estágio que foi otimizado via MIP. A desagregação dos lotes e a sincronia são realizados somente se $N\text{Lotes}_{mt} > 0$ (linha 5), ou seja, se houve produção no tanque preparatório/linha m no período t .
- Na linha 6 são inicializados os parâmetros $CapUtil_{mt}$, TD_{mt}^I e TD_{mt}^{II} com valor 0. Uma vez que no início do horizonte de planejamento sempre é necessária a realização de uma limpeza nos tanques preparatórios e nas linhas (ver Figura 9), um novo ciclo desde a última limpeza é iniciado.
- Na linha 7 inicia-se o processo de sincronia, isto é, os lotes de bebida no estágio I são convertidos em lotes de produto final nas linhas e os tempos despendidos com as limpezas e as esperas são considerados.
- Se existe uma troca de item (linha 8), ocorre uma limpeza nos dois estágios. Os parâmetros TD_{mt}^I e TD_{mt}^{II} são atualizados (linha 9) e um novo ciclo sem limpezas temporais se inicia e é considerado até que TD_{mt}^I atinja o valor TP_{max} e/ou TD_{mt}^{II} atinja no máximo TL_{max} , ou até que uma nova troca de sabor seja realizada.
- Senão, se não há uma troca de sabor (linha 13), o tempo decorrido desde a realização da última limpeza (primeira do período, temporal ou troca de sabor) é checado nas linhas 14, 19 e 24. Quando existe alguma situação que infringe o tempo máximo aceitável, é necessário incluir uma limpeza temporal (no tanque preparatório e/ou na linha). Então, os parâmetros TD_{mt}^I e TD_{mt}^{II} são reiniciados, um novo lote do mesmo sabor é adicionado, os tempos de espera são calculados e $CapUtil_{mt}$ é atualizada.

A maneira como as heurísticas decompõem o problema e realizam a sincronia estão definidas no fluxograma apresentado na Figura 21 e pelo Algoritmo 1. No passo de

Algoritmo 1: Pseudocódigo da construção da sincronia.

```

1 INÍCIO
2 para  $t \in T$  faça
3   para  $m \in M$  faça
4     Calcule  $NLotes_{mt}$ .
5     se  $NLotes_{mt} > 0$  então
6       Inicialização:  $CapUtil_{mt} \leftarrow 0$ ;  $TD_{mt}^I \leftarrow 0$ ;  $TD_{mt}^{II} \leftarrow 0$ ;  $SB_0 \leftarrow SB_1$ .
7       para  $o$  de 1 até  $NLotes_{mt}$  faça
8         se  $SB_o \neq SB_{o-1}$  então
9            $TD_{mt}^I \leftarrow T_p$  e  $TD_{mt}^{II} \leftarrow \frac{Lt_o}{S_m}$ .
10          Calcule as esperas  $\theta_o^I$  ( (A), (B), (C) ou (D) ) e  $\theta_o^{II}$  (E).
11           $TD_{mt}^I \leftarrow TD_{mt}^I + \theta_o^I$  e  $TD_{mt}^{II} \leftarrow TD_{mt}^{II} + \theta_o^{II}$ .
12           $CapUtil_{mt} \leftarrow CapUtil_{mt} + TC_{ij}^{II} + TD_{mt}^{II}$ .
13        senão
14          se  $TD_{mt}^I > TP_{max}$  e  $TD_{mt}^{II} > TL_{max}$  então
15             $TD_{mt}^I \leftarrow T_p$  e  $TD_{mt}^{II} \leftarrow \frac{Lt_o}{S_m}$ .
16            Calcule as esperas  $\theta_o^I$  e  $\theta_o^{II}$ .
17             $TD_{mt}^I \leftarrow TD_{mt}^I + \theta_o^I$  e  $TD_{mt}^{II} \leftarrow TD_{mt}^{II} + \theta_o^{II}$ .
18             $CapUtil_{mt} \leftarrow CapUtil_{mt} + LT^{II} + TD_{mt}^{II}$ .
19          senão se  $TD_{mt}^I > TP_{max}$  e  $TD_{mt}^{II} < TL_{max}$  então
20             $TD_{mt}^I \leftarrow T_p$  e  $TD_{mt}^{II} \leftarrow TD_{mt}^{II} + \frac{Lt_o}{S_m}$ .
21            Calcule as esperas  $\theta_o^I$  e  $\theta_o^{II}$ .
22             $TD_{mt}^I \leftarrow TD_{mt}^I + \theta_o^I$  e  $TD_{mt}^{II} \leftarrow TD_{mt}^{II} + \theta_o^{II}$ .
23             $CapUtil_{mt} := CapUtil_{mt} + TD_{mt}^{II}$ .
24          senão se  $TD_{mt}^I < TP_{max}$  e  $TD_{mt}^{II} > TL_{max}$  então
25             $TD_{mt}^I \leftarrow TD_{mt}^I + T_p$  e  $TD_{mt}^{II} \leftarrow TD_{mt}^{II} + \frac{Lt_o}{S_m}$ .
26            Calcule as esperas  $\theta_o^I$  e  $\theta_o^{II}$ .
27             $TD_{mt}^I \leftarrow TD_{mt}^I + \theta_o^I$  e  $TD_{mt}^{II} \leftarrow TD_{mt}^{II} + \theta_o^{II}$ .
28             $CapUtil_{mt} \leftarrow CapUtil_{mt} + LT^{II} + TD_{mt}^{II}$ .
29          senão
30             $TD_{mt}^I \leftarrow TD_{mt}^I + T_p$  e  $TD_{mt}^{II} \leftarrow \frac{Lt_o}{S_m}$ .
31            Calcule as esperas  $\theta_o^I$  e  $\theta_o^{II}$ .
32             $TD_{mt}^I \leftarrow TD_{mt}^I + \theta_o^I$  e  $TD_{mt}^{II} \leftarrow TD_{mt}^{II} + \theta_o^{II}$ .
33             $CapUtil_{mt} \leftarrow CapUtil_{mt} + TD_{mt}^{II}$ .
34 FIM

```

redução de capacidade (veja fluxograma na Figura 21), duas diferentes estratégias podem ser utilizadas dando origem a duas variações (tipo A e tipo B) de cada heurística HE1 e HE2. Assim, tem-se as heurísticas: HE1A, HE2A, HE1B e HE2B. Nas heurísticas do tipo A, todas as capacidades excedidas do horizonte de planejamento são reduzidas antes da reotimização. Nas heurísticas do tipo B, é reduzida apenas a primeira capacidade excedida detectada pelo algoritmo.

Os passos das heurísticas do tipo A e B são descritos, respectivamente, nos Algoritmos 2 e 3 a seguir. Considere os seguintes parâmetros:

- $CapDisp_{mt}$ é a capacidade disponível (tempo disponível de produção) do tanque

preparatório/linha m no período t no problema original;

- Cap_{mt} é a capacidade disponível do tanque preparatório/linha m no período t utilizada no MIP;
- $CapUtil_{mt}$ quantidade de tempo utilizada na programação da produção para tanque preparatório/linha m no período t após a sincronia;
- $Tempo_Exec$ é o tempo decorrido desde o início do algoritmo;
- $Tempo_Lim$ é o tempo limite de execução do algoritmo;
- δ é a quantia que falta de capacidade na linha m e período t para a sincronia ser factível. Se $CapUtil_{mt} > CapDisp_{mt}$, então $\delta = CapUtil_{mt} - CapDisp_{mt}$;
- β é uma porcentagem a ser utilizada na redução de capacidade.
- β_1 é o valor atribuído a β para $Tempo_Exec < \frac{1}{3}Tempo_Lim$; β_2 é o valor atribuído a β para $\frac{1}{3}Tempo_Lim < Tempo_Exec \leq \frac{2}{3}Tempo_Lim$; $\beta_3 = 1$ é o valor atribuído quando $Tempo_Exec > \frac{2}{3}Tempo_Lim$, definidos *a priori*.
- ε valor mínimo de redução da capacidade. Se a capacidade tem que ser reduzida, deve ser reduzida em no mínimo ε ;
- φ porcentagem para redução das capacidades de todos os tanques preparatórios/linhas m .

Nas heurísticas do tipo A é avaliado se o tempo disponível de produção ($CapDisp_{mt}$) foi excedido para todo tanque preparatório/linha m e período t . Para todos os pares (m, t) , para os quais a capacidade foi violada, reduz-se a capacidade Cap_{mt} no modelo e o processo se repete. No início da heurística tem-se $Cap_{mt} = CapDisp_{mt}$, pois ainda não houve nenhuma redução de capacidade, mas, ao longo do Algoritmo 2, o valor de Cap_{mt} vai sendo modificado. Esta rotina é repetida até que uma solução factível seja obtida, ou até que o tempo limite de execução seja atingido.

Para cada uma das heurísticas HE1 e HE2 são resolvidos modelos diferentes de dimensionamento e sequenciamento de lotes na linha 3 do Algoritmo 2. Esses modelos são descritos em detalhes nas seções 5.1.1, 5.1.2, 5.1.4 e 5.1.4. Os modelos resolvidos são extensões de modelos de dimensionamento e sequenciamento de lotes clássicos da literatura, alguns deles, por exemplo, são baseados no CLSP. Trigeiro, Thomas e McClain (1989) aponta que para o CLSP (*Capacitated Lot Sizing Problem*) com tempos de *setup*, o problema de checar se existe uma solução factível é NP-completo. Assim, como no problema resolvido são utilizadas instâncias reais de grande porte, é esperado que os modelos utilizados nas heurísticas sejam também difíceis de serem resolvidos. De fato, dependendo do tamanho dos exemplares, nem sempre esses modelos são resolvidos até

Algoritmo 2: Pseudocódigo das heurísticas do tipo A.

```

1 INÍCIO
2 Inicialização: Defina Tempo_Lim,  $\beta_1, \beta_2, \beta_3, \varphi, \delta, \varepsilon$ .
3 enquanto Tempo_Exec < Tempo_Lim faça
4     Resolva o modelo matemático.
5     se Existe uma solução factível então
6         Sincronize os estágios. (Use Algoritmo 1).
7         se Tempo_Exec  $\leq \frac{1}{3} \cdot$ Tempo_Lim então
8              $\beta \leftarrow \beta_1$ 
9         senão se  $\frac{1}{3} \cdot$ Tempo_Lim < Tempo_Exec  $\leq \frac{2}{3} \cdot$ Tempo_Lim então
10             $\beta \leftarrow \beta_2$ 
11            para todo  $t \in T$  e  $m \in M$  faça
12                 $Cap_{mt} \leftarrow \varphi \cdot Cap_{mt}$ 
13            senão
14                 $\beta \leftarrow \beta_3$ 
15            para todo  $t \in T$  e  $m \in M$  faça
16                se  $CapUtil_{mt} > CapDisp_{mt}$  então
17                     $\delta \leftarrow CapUtil_{mt} - CapDisp_{mt}$ .
18                     $Cap_{mt} \leftarrow Cap_{mt} - \max\{\beta \cdot \delta, \varepsilon\}$ .
19                     $Aux \leftarrow 1$ .
20                se  $Aux=0$  então
21                    Pare e retorne a melhor solução inteira.
22            senão
23                Pare. O algoritmo não retorna uma solução factível.
24 FIM

```

a otimalidade pelo *solver* dentro de um tempo razoável para a execução das heurísticas. Assim, a otimalidade ou um tempo limite são os critérios de parada na solução desses modelos. Conforme as premissas de modelagem apresentadas no Capítulo 4, são permitidos atrasos, logo o modelo sempre é factível.

Toda redução de capacidade no modelo matemático implica em mais tempo disponível para inclusão das limpezas temporais e dos tempos de espera. No entanto, essas reduções geram soluções com altos níveis de atraso e, portanto, soluções com qualidade ruim. Por outro lado, reduzir pouco a capacidade implica em pouco tempo para inclusão das esperas e limpezas temporais, o que pode levar a soluções infactíveis para o problema original. Ou seja, grandes reduções podem levar a soluções com muito atraso e reduções muito pequenas podem impedir que o algoritmo convirja em um tempo razoável, isto é, que a heurística termine sem solução factível dentro do tempo máximo de execução permitido. Vemos então que existe um *trade-off* na redução de capacidade.

A redução da capacidade é realizada conforme descrito entre as linhas 12 e 18 do Algoritmo 2. Quando a capacidade utilizada após a sincronia ($CapUtil_{mt}$) é um valor

próximo da capacidade disponível ($CapDisp_{mt}$), temos que o valor de δ (que é a diferença entre as capacidades) é muito pequeno. Assim, se somente o valor δ fosse utilizado na redução de capacidade, essas reduções seriam muito pequenas, logo a solução do novo modelo com a capacidade reduzida seria muito próxima da solução da última iteração ou até mesmo idêntica, fazendo com que a convergência do algoritmo para uma solução factível fosse muito lenta. Assim, propõem-se aqui diferentes formas de redução para contornar essa situação. Na linha 17 a capacidade é atualizada com o maior valor entre $\beta \cdot \delta$ e ε . Tomar o valor máximo entre uma porcentagem da capacidade ultrapassada ($\beta \cdot \delta$) e uma tolerância definida (ε) é uma maneira de equilibrar as reduções de capacidade. No início do algoritmo, os valores de δ podem ser muito grandes e portanto utiliza-se a porcentagem β . À medida que δ vai se tornando muito pequeno, a tolerância ε é um número pequeno que supera esse valor pequeno de δ e vai ser dominante nas iterações finais do algoritmo, para melhorar e garantir a convergência do algoritmo.

A ideia proposta para lidar com a infactibilidade apresentada no Algoritmo 2 é que as reduções de capacidade comecem de forma mais sutil e se tornem mais agressivas conforme o tempo vai passando, ou seja, o valor de β é atualizado ao longo da heurística. Os parâmetros que influenciam nessa redução são calculados nas linhas de 6 a 13. O valor de β começa pequeno e aumenta à medida que o tempo de execução vai se aproximando do tempo limite, assim: $0 < \beta_1 < \beta_2 < 1$ e $\beta_3 = 1$. Uma outra estratégia utilizada também para tentar encontrar uma solução factível dentro do tempo limite é reduzir a capacidade de todos os períodos e máquinas por um fator φ quando o tempo decorrido está entre $1/3$ e $2/3$ do tempo limite. Testes computacionais mostraram que essas reduções são efetivas uma vez que sempre é encontrada uma solução factível para todas as instâncias testadas no Capítulo 6. As definições dos valores de β_1 , β_2 , φ e ε são feitas empiricamente através de testes iniciais.

Essa estratégia utilizada aqui para reduzir a capacidade de acordo com o tempo disponível da heurística é diferente da estratégia proposta em [Toscano, Ferreira e Morabito \(2015\)](#) e [Toscano, Ferreira e Morabito \(2017\)](#), que apresentam uma heurística parecida com a HE1 e HE2. Os autores utilizam uma estratégia de redução de capacidade com o foco para obtenção de soluções com mais qualidade e portanto, acabam por não encontrar solução factível para algumas instâncias dentro de um tempo de execução definido. A diferença no Algoritmo 2 para os trabalhos [Toscano, Ferreira e Morabito \(2015\)](#) e [Toscano, Ferreira e Morabito \(2017\)](#) é a maneira com que β é calculado ao longo da heurística. Em [Toscano, Ferreira e Morabito \(2015\)](#) e [Toscano, Ferreira e Morabito \(2017\)](#) o valor de β é definido *a priori* e mantido durante toda a heurística.

A diferença entre as heurísticas do tipo A e do tipo B é o passo de redução de capacidade. Na heurística tipo B, a capacidade é reduzida apenas para o primeiro par (m, t) identificado com capacidade excedida. Quando o par (m, t) é identificado, a redução

de capacidade é aplicada e o modelo matemático é resolvido novamente. Assim, as linhas de 15 a 18 no Algoritmo 2 são substituídas pelas linhas de 14 a 19 descritas no Algoritmo 3.

Nas próximas seções são descritos os modelos matemáticos utilizados nessas heurísticas.

Uma vez que neste trabalho foram encontradas casos de tempos e custos dependentes e independentes da sequência, são propostos modelos matemáticos para os estágios I e II para os dois casos: dependente da sequência (DS) e independente da sequência (IS). Logo, a partir das heurísticas HE1 e HE2, são obtidas oito variações definidas na Figura 23.

Algoritmo 3: Pseudocódigo das heurísticas do tipo B.

```

14 para  $t \in T$  e  $m \in M$  faça
15     se  $CapUtil_{mt} > CapDisp_{mt}$  então
16          $\delta \leftarrow CapUtil_{mt} - CapDisp_{mt}$ .
17          $Cap_{mt} \leftarrow Cap_{mt} - \max\{\beta \cdot \delta, \varepsilon\}$ .
18          $Aux \leftarrow 1$ .
19     Interrompa o laço.

```

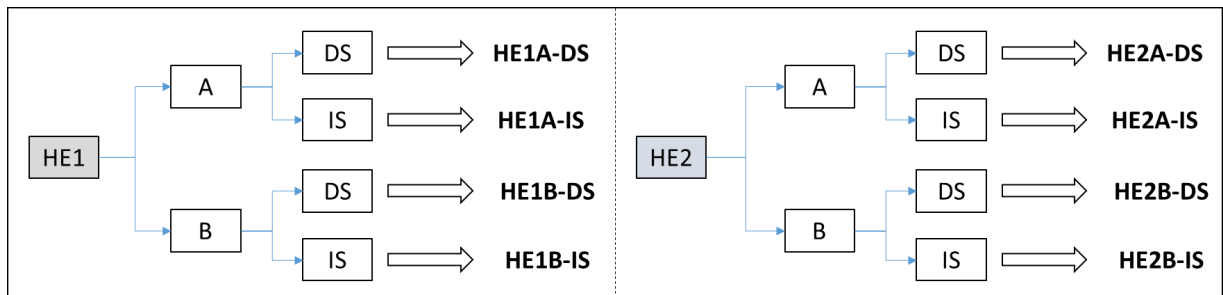


Figura 23 – Resumo das variações das heurísticas HE1 e HE2. Fonte: a autora.

5.1.1 Heurísticas Estágio I - HE1A-DS e HE1B-DS

As heurísticas HE1 resolvem o problema de dimensionamento e sequenciamento de lotes para o estágio I via modelo matemático. Assim, são tomadas as decisões do número de lotes, da quantidade de litros que deve ser produzida pelos tanques preparatórios em cada lote, e da sequência de produção. Isto é feito por um modelo de dimensionamento e sequenciamento de lotes capacitado, multiperíodo, multi-item e multimáquina. O modelo é descrito a seguir pelas expressões (5.1)-(5.12). Esse modelo foi inspirado pelo modelo CLSD de Haase (1996) e nos modelos aplicados a refrigerantes de Defalque, Rangel e Ferreira (2011) e Ferreira et al. (2012).

Considere os parâmetros apresentados no Capítulo 4 para o modelo MDSL-2E-LT. Considere também d_{jt}^I como a demanda em litros do item j no período t e $MaxTqs_m$ como um limite superior estimado para a quantidade de tanques que podem ser preparados pelo

tanque preparatório m ao longo de um período. As variáveis do Modelo de Dimensionamento e Sequenciamento de Lotes do Preparo (MDSL-P) são:

- I_{jt}^{I+} estoque em litros do item j no final do período t ;
 I_{jt}^{I-} atraso em litros do item j no final do período t ;
 X_{mjt}^I quantidade produzida em litros no tanque preparatório m do item j no período t ;
 Z_{mijt} = 1 se no tanque preparatório m existe uma troca do item i para o item j no período t ;
0 caso contrário, $Z_{mijt} = 0 \quad \forall j$;
 U_{mjt}^I número de limpezas temporais ocorridas no tanque preparatório m para o item j
no período t ;
 ξ_{mjt}^I número de lotes produzidos pelo tanque preparatório m do item j no período t .

O modelo MDSL-P é dado a seguir pelas equações (5.1)-(5.12).

$$\begin{aligned} \text{Min } Z^I = & \sum_{t \in T} \sum_{j \in J} (h_j^+ I_{jt}^{I+} + h_j^- I_{jt}^{I-}) / \rho_j + \sum_{m \in M} \sum_{t \in T} \sum_{j \in J} \sum_{\substack{i \in J \\ i \neq j}} C_{ij} Z_{mijt} \\ & + \sum_{m \in M} \sum_{j \in J} \sum_{t \in T} Clp U_{mjt}^I \end{aligned} \quad (5.1)$$

s.a.:

$$I_{j(t-1)}^{I+} + I_{jt}^{I-} + \sum_{m \in M} X_{mjt}^I = d_{jt}^I + I_{jt}^{I+} + I_{j(t-1)}^{I-}, \quad \forall t \in T, \forall j \in J. \quad (5.2)$$

$$\sum_{j \in J} Tp \xi_{mjt}^I + \sum_{j \in J} \sum_{\substack{i \in J \\ i \neq j}} TC_{ij}^I Z_{mijt} + \sum_{j \in J} LT^I U_{mjt}^I \leq Cap_{mt}, \quad \forall m \in M, \forall t \in T. \quad (5.3)$$

$$U_{mjt}^I \geq \frac{Tp \xi_{mjt}^I}{TP_{max}} - 1, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.4)$$

$$X_{mjt}^I \geq LB_j \xi_{mjt}^I, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.5)$$

$$X_{mjt}^I \leq UB_j \xi_{mjt}^I, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.6)$$

$$\xi_{mjt}^I \leq |T| MaxTq_m \sum_{i \in J} Z_{mijt}, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.7)$$

$$\sum_{\substack{i \in J \\ i \neq j}} Z_{mijt} = \sum_{\substack{i \in J \\ i \neq j}} Z_{mij t}, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.8)$$

$$\sum_{\substack{j \in J \\ j \neq i_0}} Z_{mij t} \geq \sum_{\substack{i \in J \\ i \neq k}} Z_{mik t}, \quad \forall m \in M, \forall k \in J, \quad \forall t \in T. \quad (5.9)$$

$$\sum_{\substack{j \in J \\ i \neq j}} Z_{mijt} \leq 1, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.10)$$

$$V_{mjt} \geq (V_{mit} + 1) - (|J| - 1)(1 - Z_{mijt}), \quad \forall m \in M, \forall t \in T, \quad \forall i, j \in J \setminus i_0, i \neq j. \quad (5.11)$$

$$Z_{mijt} \in \{0, 1\}, U_{mjt}^I, \xi_{mjt}^I \in \mathbb{Z}_+; X_{mjt}^I, I_{jt}^{I,+}, I_{jt}^{I,-}, V_{mjt} \geq 0, \quad \forall m \in M, \forall t \in T, \quad \forall i, j \in J. \quad (5.12)$$

A função objetivo (5.1) minimiza a soma dos custos de estoque, atraso, trocas e penaliza o número de limpezas temporais. A restrição (5.2) representa o balanceamento de estoque e demanda para cada item em cada período. A restrição (5.3) garante que a capacidade não será excedida. Nessa restrição, estão somados os tempos despendidos com preparo, com as trocas entre itens e com as limpezas temporais. O número de limpezas temporais é contado através da restrição (5.4). A restrição (5.4), combinada com a minimização da variável U_{mjt}^I na função objetivo, conta quantas limpezas temporais deverão ser realizadas para cada item j no período t através da função piso $\left\lfloor \frac{T_p \xi_{mjt}^I}{TP_{max}} \right\rfloor$. Isto é, a cada TP_{max} tempos de preparação de lotes de bebida j , é necessária a realização de uma limpeza temporal. Os lotes mínimo e máximo são garantidos pelas restrições (5.5) e (5.6), respectivamente. A restrição (5.7) é de *setup*. Essa restrição associada com a minimização da variável Z_{mijt} garante que se há preparo do item j , tem que ter havido uma troca de i para j ; e o número máximo de preparos que podem ser realizados está limitado por $MaxTqs_m$. As restrições (5.8) a (5.11) funcionam da mesma forma que (4.10)-(4.13). A restrição de fluxo é dada em (5.8). A restrição (5.9) impõe que o primeiro item produzido seja o item fantasma i_0 e a restrição (5.10), que não haja mais do que uma troca de i para j no tanque m no período t . Para eliminação de subrotas são utilizadas as restrições (5.11). Finalmente, a restrição (5.12) define o domínio das variáveis.

Assim, as heurísticas H1EA-DS e H1EB-DS seguem os Algoritmos 2 e 3, respectivamente. Na linha 3 do Algoritmo 2 o modelo resolvido é o MDSL-P (5.1)-(5.12). No Algoritmo 1 de construção da sincronia tem-se que $NLots_{mt} = \sum_{j \in J} \xi_{mjt}^I$.

5.1.2 Heurísticas Estágio I - HE1A-IS e HE1B-IS

Para o caso em que os tempos e custos de troca são independentes da sequência, a heurística HE1A-IS continua seguindo o Algoritmo 2 e a heurística HE1B-IS o Algoritmo 3. A única diferença é que na linha 3, o modelo que passa a ser resolvido é dado pelas expressões (5.13)-(5.17).

No problema considerado independente da sequência (IS), sempre que há uma troca entre itens, é realizada uma limpeza temporal, como foi descrito no Capítulo 3. Assim, os custos e tempos de troca entre os itens serão os mesmos custos e tempos de realização

considerados para limpeza temporal. Considere os parâmetros e variáveis utilizados pelo modelo MDSL-P. Considere ainda a variável:

$$P_{mjt} = 1 \text{ se existe produção no tanque preparatório } m \text{ do item } j \text{ no período } t; \\ 0 \text{ caso contrário.}$$

O Modelo de Dimensionamento de Lotes do Preparo (MDL-P) é dado pelas expressões (5.13), (5.2), (5.4), (5.5), (5.6) e (5.14)-(5.17).

$$\text{Min } Z^I = \sum_{t \in T} \sum_{j \in J} (h_j^+ I_{jt}^{I,+} + h_j^- I_{jt}^{I,-}) / \rho_j + \sum_{m \in M} \sum_{t \in T} \sum_{j \in J} Clp P_{mjt} + \sum_{m \in M} \sum_{j \in J} \sum_{t \in T} Clp U_{mjt}^I. \quad (5.13)$$

s.a.:

$$\begin{aligned} & (5.2), (5.4), (5.5), (5.6) \\ \sum_{j \in J} Tp \xi_{mjt}^I + \sum_{j \in J} LT^I Z_{mjt} + \sum_{j \in J} LT^I U_{mjt}^I & \leq Cap_{mt}, \quad \forall m \in M, \forall t \in T. \end{aligned} \quad (5.14)$$

$$\xi_{mjt}^I \leq MaxTq_m P_{mjt}, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.15)$$

$$P_{mjt} \leq \xi_{mjt}^I, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.16)$$

$$P_{mjt} \in \{0, 1\}, U_{mjt}^I, \xi_{mjt}^I \in \mathbb{Z}_+; X_{mjt}, I_{jt}^{I,+}, I_{jt}^{I,-}, V_{mjt} \geq 0, \quad \forall t \in T, \forall m \in M, \quad \forall i, j \in J. \quad (5.17)$$

A função objetivo (5.13) minimiza os custos de estoque, atraso e penaliza as trocas e limpezas temporais. As restrições (5.2), (5.4), (5.5), (5.6) determinam o balanceamento de estoque e demanda, o número de limpezas temporais, os lotes máximo e mínimo e já foram explicadas anteriormente. A restrição (5.15) é similar à (5.7) e garante que quando existe preparo do item j , deve ter havido uma limpeza temporal antes do início da produção de j . A restrição (5.16) assegura que se não houver preparo do item j não é realizada uma limpeza temporal. O domínio da variáveis é definido em (5.17).

Note que o modelo MDL-P é uma simplificação do modelo MDSL-P (5.1)-(5.12). A variável binária Z_{mijt} é substituída pela variável P_{mjt} , a restrição (5.7) é adaptada para a restrição (5.15) e as restrições que determinam o sequenciamento, (5.8), (5.9), (5.10) e (5.11), são eliminadas.

5.1.3 Heurísticas Estágio II - HE2A-DS e HE2B-DS

As heurísticas H2E seguem basicamente os mesmos passos das heurísticas H1E. A principal diferença entre elas é que o modelo resolvido na HE2 é para o estágio II, conforme

explicado na Seção 5.1.

Nas heurísticas HE2A-DS e HE2B-DS, o modelo de dimensionamento e sequenciamento de lotes capacitado, multiperíodo, multi-item e multimáquina é para as linhas (estágio II). Considere o conjunto de parâmetros apresentados no Capítulo 4 para o modelo MDSL-2E-LT. Considere também o parâmetro: a_m o tempo de produção de uma unidade de um item na linha m . As variáveis utilizadas no modelo para a linha são:

- I_{jt}^+ estoque em unidades do item j no final do período t ;
- I_{jt}^- atraso em unidades do item j no final do período t ;
- X_{mjt}^{II} quantidade produzida em unidades na linha m do item j no período t ;
- U_{mjt}^{II} número de limpezas temporais ocorridas na linha m para o item j no período t ;

O Modelo de Dimensionamento e Sequenciamento de Lotes da Linha (MDSL-L) é dado pelas equações (5.18)-(5.23), (5.8)-(5.11) e (5.24).

$$\begin{aligned} \text{Min } Z^{II} = & \sum_{t \in T} \sum_{j \in J} (h_j^+ I_{jt}^+ + h_j^- I_{jt}^-) + \sum_{m \in M} \sum_{t \in T} \sum_{j \in J} \sum_{\substack{i \in J \\ i \neq j}} C_{ij} Z_{mijt} \\ & + \sum_{m \in M} \sum_{j \in J} \sum_{t \in T} Clp U_{mjt}^{II} \end{aligned} \quad (5.18)$$

s.a.:

$$I_{j(t-1)}^+ + I_{jt}^- + \sum_{m \in M} X_{mjt}^{II} = d_{jt} + I_{jt}^+ + I_{j(t-1)}^-, \forall t \in T, \forall j \in J. \quad (5.19)$$

$$\sum_{j \in J} a_m X_{mjt}^{II} + \sum_{j \in J} \sum_{\substack{i \in J \\ i \neq j}} TC_{ij}^{II} Z_{mijt} + \sum_{j \in J} LP^{II} U_{mjt}^{II} \leq Cap_{mt}, \forall m \in M, \forall t \in T. \quad (5.20)$$

$$U_{mjt}^{II} \geq \frac{a_m X_{mjt}^{II}}{TL_{max}} - 1, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.21)$$

$$X_{mjt}^{II} \geq LB_j \sum_{\substack{i \in J \\ i \neq j}} Z_{mijt}, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.22)$$

$$X_{mjt}^{II} \leq Cap_{mt} S_m \sum_{\substack{i \in J \\ i \neq j}} Z_{mijt}, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.23)$$

$$(5.8), (5.9), (5.10), (5.11)$$

$$Z_{mijt} \in \{0, 1\}, U_{mjt}^{II} \in \mathbb{Z}_+; X_{mjt}^{II}, I_{jt}^+, I_{jt}^-, V_{mjt} \geq 0, \quad \forall m \in M, \forall t \in T, \quad \forall i, j \in J. \quad (5.24)$$

A função objetivo (5.18) minimiza a soma dos custos de estoque, atraso, trocas e penaliza o número de limpezas temporais na linha. A restrição (5.19) representa o

balanceamento de estoque e demanda e (5.20) garante que a capacidade não será excedida. Nessa restrição estão somados os tempos de envase, os tempos de troca e das limpezas temporais da linha. A restrição (5.21), combinada com a minimização da variável U_{mjt}^{II} na função objetivo, conta quantas limpezas temporais deverão ser realizadas na linha para cada item j no período t . Logo, a cada TL_{max} tempos de envase da bebida j , é necessária a realização de uma limpeza temporal na linha. Os lotes mínimo e máximo são garantidos pela restrições (5.22) e (5.23), respectivamente. Nesse caso, o lote máximo é limitado pela capacidade máxima de produção da linha. As restrições de (5.8) a (5.11) determinam o sequenciamento e já foram explicadas anteriormente. A restrição (5.24) define o domínio das variáveis.

Note que o MDSL-L segue a mesma ideia do modelo MDSL-P, porém nesse modelo não é necessário uma variável para contabilizar o número de tanques preparados (ξ_{mjt}). No modelo MDSL-L, os tempos de produção são determinados pelo parâmetro a_m , enquanto que no modelo MDSL-P o tempo de produção de cada tanque é fixo e dado por T_p . Outra diferença é que no modelo modelo MDSL-P, existe um lote máximo de limitação da capacidade física do tanque preparatório na restrição (5.6), enquanto que no modelo MDSL-L a limitação é a capacidade do período e da máquina na restrição (5.23), juntamente com TL_{max} na restrição (5.21).

As heurísticas HE2A-DS e HE2B-DS seguem os Algoritmos 2 e 3, respectivamente, e na linha 3 de ambos os algoritmos o modelo resolvido é o MDSL-L (5.18)-(5.24). No modelo MDSL-P as variáveis ξ_{mjt}^I contam o número de tanques que devem ser preparados para atender a demanda. O modelo MDSL-L define os lotes em unidades, então para proceder com a sincronia, o número total de lotes em litros, ou seja, o número de tanques que deverão ser preparados, é calculado por $NLotes_{mt} = \sum_{j \in J} \frac{\rho_j \cdot X_{mjt}^{II}}{UB_j}$ (linha 4 do Algoritmo 1). O termo $\rho_j \cdot X_{mjt}^{II}$ é a produção em litros, e UB_j é a quantidade máxima de litros do item j que é possível preparar em um tanque.

5.1.4 Heurísticas Estágio II - HE2A-IS e HE2B-IS

As heurísticas H2EA-IS e HE2B-IS seguem os mesmos passos das heurísticas HE2A-DS e HE2B-DS, respectivamente, exceto pelo modelo, resolvido para as linhas. Neste caso, o modelo resolvido na linha 3 do Algoritmo 2 é de um Modelo de Dimensionamento de Lotes para a Linha (MDL-L). Da mesma forma que no modelo MDL-P, os tempos e custos de troca entre os itens são os mesmos da limpeza temporal, ou seja, independentes da sequência.

Os parâmetros e variáveis são os mesmos utilizados para os modelos apresentados anteriormente neste capítulo. Além desses, é considerada a variável:

ξ_{mjt}^{II} número de lotes do item j envasados pela linha m no período t ;

O modelo MDL-L é dado pelas equações (5.25)-(5.30).

$$\text{Min } Z^{II} = \sum_{t \in T} \sum_{j \in J} (h_j^+ I_{jt}^+ + h_j^- I_{jt}^-) + \sum_{m \in M} \sum_{j \in J} \sum_{t \in T} Clp \xi_{mjt}^{II} \quad (5.25)$$

s.a.:

$$I_{j(t-1)}^+ + I_{jt}^- + \sum_{m \in M} X_{mjt}^{II} = d_{jt} + I_{jt}^+ + I_{j(t-1)}^- \quad \forall j \in J, \forall t \in T. \quad (5.26)$$

$$\sum_{j \in J} a_m X_{mjt}^{II} + LT^{II} \sum_{j \in J} \xi_{mjt}^{II} \leq Cap_{mt}, \quad \forall m \in M, t \in T. \quad (5.27)$$

$$a_m X_{mjt}^{II} \leq TL_{max} \xi_{mjt}^{II}, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (5.28)$$

$$X_{mjt}^{II} \geq \rho_j LB_j \xi_{mjt}^{II}, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (5.29)$$

$$I_{j,t} \geq 0, X_{mjt} \geq 0, \xi_{mjt}^{II} \in \mathbb{Z}_+, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (5.30)$$

A função objetivo (5.25) minimiza os custos de estoque, atraso e limpezas temporais que acontecem a cada troca de item, ou a cada tempo TL_{max} de produção contínua do item j . A restrição (5.26) garante o balanceamento de estoque e demanda e (5.27) limita a produção de acordo com a capacidade disponível. A restrição (5.28) restringe o lote máximo a um tempo TL_{max} e contabiliza o número de limpezas temporais ocorridas por tempo ou por troca de item. O lote mínimo é garantido pela restrição (5.29) e o domínio das variáveis é dado em (5.30).

Note que o modelo MDL-P ((5.13), (5.2)-(5.6),(5.14)-(5.17)) é obtido por uma simplificação do modelo MDSL-P (5.1)-(5.12) para o caso não dependente da sequência. O modelo MDSL-L ((5.18)-(5.23), (5.8)-(5.11),(5.24)) é obtido por uma adaptação do modelo MDSL-P para o segundo estágio. E o modelo MDL-L (5.26)-(5.30) é obtido por uma simplificação do modelo MDSL-L. Uma vez que a produção na linha não está discretizada em tanques, como no modelo MDL-P, é possível determinar as limpezas temporais através de lotes máximos de TL_{max} na linha, assim como foi feito em (PAGLIARUSSI; MORABITO; SANTOS, 2017). Logo, não são necessárias as variáveis Z_{mjt} , pois a restrição (5.28) indica que a cada lote máximo produzido é realizada uma limpeza temporal, e isso garante inclusive as limpezas temporais no início de cada período. Assim, a restrição de controle da capacidade (5.27) é diferente da restrição (5.20), pois em (5.27) o primeiro termo determina o tempo de produção e o segundo termo contabiliza todas as limpezas temporais que ocorrem, por troca ou por tempo máximo sem limpeza.

Resumo das heurísticas de decomposição

Nas seções anteriores foram apresentadas heurísticas construtivas baseadas em decomposição para o problema de programação da produção de bebidas à base de frutas. Dois tipos de heurísticas foram propostas: HE1 e HE2, que diferem pelo estágio de produção resolvido via modelo. Para cada uma dessas foram propostas duas variações na forma de tratamento da capacidade, gerando as heurísticas HE1A, HE1B, HE2A e HE2B. E para cada uma dessas variações foram considerados modelos para os casos DS e IS. As Tabelas 16 e 17 resumam as heurísticas HE1 e HE2, respectivamente. Nestas tabelas são apresentados os estágios cuja programação da produção é resolvida via modelo, o modelo utilizado e a maneira que a capacidade é reduzida ao longo das iterações.

Tabela 16 – Sumário das heurísticas propostas - HE1.

Heurística	Estágio/Modelo	Procedimento de Sincronia
HE1A-DS	Estágio I MDSL-P (5.1)-(5.12)	Reduz Cap_{mt} para todo (m, t) em que há excesso e reotimiza o modelo
HE1B-DS	Estágio I MDSL-P (5.1)-(5.12)	Reduz Cap_{mt} no primeiro par (m, t) detectado e reotimiza o modelo
HE1A-IS	Estágio I MDL-P (5.13);(5.2)-(5.6);(5.14)-(5.17)	Reduz Cap_{mt} para todo (m, t) em que há excesso e reotimiza o modelo
HE1B-IS	Estágio I MDL-P (5.13);(5.2)-(5.6);(5.14)-(5.17)	Reduz Cap_{mt} no primeiro par (m, t) detectado e reotimiza o modelo

Tabela 17 – Sumário das heurísticas propostas - HE2.

Heurística	Estágio/Modelo	Procedimento de Sincronia
HE2A-DS	Estágio II MDSL-L (5.18)-(5.23);(5.8)-(5.11);(5.24)	Reduz Cap_{mt} para todo (m, t) em que há excesso e reotimiza o modelo
HE2B-DS	Estágio II MDSL-L (5.18)-(5.23);(5.8)-(5.11);(5.24)	Reduz Cap_{mt} no primeiro par (m, t) detectado e reotimiza o modelo
HE2A-IS	Estágio II MDL-L (5.25)-(5.30)	Reduz Cap_{mt} para todo (m, t) em que há excesso e reotimiza o modelo
HE2B-IS	Estágio II MDL-L (5.25)-(5.30)	Reduz Cap_{mt} no primeiro par (m, t) detectado e reotimiza o modelo

5.2 Heurísticas baseadas em modelos relaxados

Nessa seção apresenta-se outras duas heurísticas baseadas em modelos matemáticos: Heurística Subestimada (H-SUB) e Heurística Superestimada (H-SUP). O funcionamento dessas heurísticas segue os mesmos passos das heurísticas de decomposição apresentadas na Seção 5.1, ou seja, essas heurísticas também seguem o fluxograma apresentado na Figura 21. A diferença é que os modelos matemáticos consideram restrições dos dois estágios, entretanto não consideram a sincronia entre os estágios como o modelo MDSL-2E-LT. Por esse motivo, esses modelos serão chamados, a partir daqui, de modelos relaxados.

Os modelos relaxados têm menos variáveis e restrições do que o modelo MDSL-2E-LT. O modelo é considerado relaxado (aproximado), pois não calcula de maneira exata o número de limpezas temporais e as esperas que acontecem entre os dois estágios de produção devido à essas limpezas, como é feito no modelo MDSL-2E-LT. Assim, na solução retornada pelo modelo relaxado, é necessário inserir as limpezas temporais e a sincronia entre os estágios, assim como nas heurísticas de decomposição HE1 e HE2. Na fase de construção da sincronia o tamanho dos lotes e a sequência de produção, obtidos com o modelo relaxado, não são alterados. Somente os tempos de início e final dos lotes são ajustados para inclusão das limpezas temporais e das esperas. Após essa fase construtiva a solução pode ou não ser factível. Se a solução não for factível, a capacidade é diminuída e o modelo é resolvido novamente, da mesma forma que nas heurísticas HE1 e HE2. Espera-se obter melhores soluções com as heurísticas H-SUB e H-SUP em menos iterações do que com as heurísticas HE1 e HE2, uma vez que os modelos estão considerando informações dos dois estágios de produção e estão mais próximos da representação exata do problema.

O modelo aproximado utilizado na heurística H-SUB é denominado Modelo Relaxado 1 - Otimista (MR1-O), e o modelo utilizado na H-SUP é denominado Modelo Relaxado 2 - Pessimista (MR2-P). A única diferença entre os modelos MR1-O e MR2-P é a maneira de calcular o número de limpezas temporais. Por serem modelos relaxados, no modelo MR1-O a capacidade de produção é superestimada, isto significa que a estimativa do número de limpezas temporais é feita considerando menos limpezas temporais do que podem existir de fato, e portanto, ele é chamado de Otimista. No modelo MR2-P, a capacidade é subestimada, ou seja, considera que possam ocorrer mais limpezas temporais do que o Otimista e do que podem existir de fato, e por isso ele é chamado de Pessimista. Os parâmetros e variáveis são os mesmos utilizados no modelo MDSL-2E-LT e também as variáveis U_{mjt}^I e U_{mjt}^{II} utilizadas na Seção 5.1. Cabe lembrar que essas variáveis representam o número de limpezas temporais que serão realizadas no tanque preparatório m (U_{mjt}^I) e na linha (U_{mjt}^{II}) durante a produção do sabor j no período t .

5.2.1 Modelo Relaxado 1 - Otimista (MR1-O)

O Modelo Relaxado 1 - Otimista (MR1-O) é dado pela função objetivo (5.31) e pelas equações (4.5)-(4.24) e (5.32)-(5.36).

$$\begin{aligned} \text{Min } Z^S = & \sum_{t \in T} \sum_{j \in J} (h_j^+ I_{jt}^+ + h_j^- I_{jt}^-) + \sum_{m \in M} \sum_{t \in T} \sum_{j \in J} \sum_{\substack{i \in J \\ i \neq j}} C_{ij} Z_{mijt} \\ & + \sum_{m \in M} \sum_{j \in J} \sum_{t \in T} Clp(U_{mjt}^I + U_{mjt}^{II}). \end{aligned} \quad (5.31)$$

s.a.:

(4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), (4.13), (4.14), (4.15).

(4.16), (4.17), (4.18), (4.19), (4.20), (4.21), (4.22), (4.23), (4.24).

$$U_{mjt}^I \geq \frac{(\mu_{mjtO}^{II,s} - \mu_{mjt1}^{I,s}) - Mgde(1 - Y_{mjtO})}{TP_{max}} - 1, \quad \begin{array}{l} \forall m \in M, \forall j \in J, \\ \forall t \in T, \forall o \in O_{mt}. \end{array} \quad (5.32)$$

$$U_{mjt}^{II} \geq \frac{(\mu_{mjtO}^{II,e} - \mu_{mjt1}^{II,s})}{TL_{max}} - 1, \quad \begin{array}{l} \forall m \in M, \forall j \in J, \\ \forall t \in T. \end{array} \quad (5.33)$$

$$\mu_{mjtO}^{I,e} \leq Cap_{mt} - LT^I \sum_{i \in J} U_{mit}^I, \quad \begin{array}{l} \forall m \in M, \forall j \in J, \\ \forall t \in T. \end{array} \quad (5.34)$$

$$\mu_{mjtO}^{II,e} \leq Cap_{mt} - LT^{II} \sum_{i \in J} U_{mit}^{II}, \quad \begin{array}{l} \forall m \in M, \forall j \in J, \\ \forall t \in T. \end{array} \quad (5.35)$$

$$\begin{array}{l} X_{mjtO}, I_{jt}^+, I_{jt}^-, V_{mjt}, \mu_{mjtO}^{I,s}, \mu_{mjtO}^{I,e}, \mu_{mjtO}^{II,s}, \mu_{mjtO}^{II,e} \geq 0; \quad U_{mjt}^I, U_{mjt}^{II} \in \mathbb{Z}_+; \\ Z_{mijt}, Y_{mjtO} \in \{0, 1\}, \quad \forall m \in M, \forall i, j \in J, \forall t \in T, \forall o \in O_{mt}. \end{array} \quad (5.36)$$

A função objetivo (5.31) minimiza os custos de estoque, atraso, troca e o número de limpezas temporais.

A restrição (5.32) estima o número de limpezas temporais para o estágio I. Essa restrição é ativada somente quando há produção no lote o ($Y_{mjtO} = 1$). Tendo em vista que a variável $\mu_{mjtO}^{II,s}$ determina o momento em que o tanque preparatório ficou vazio (pois este é o momento em que o lote o é transferido para a linha), o termo $\mu_{mjtO}^{II,s} - \mu_{mjt1}^{I,s}$ calcula o tempo decorrido, desde o instante de início de preparo do primeiro lote $o = 1$ do item j , até o instante que o lote o é enviado para a linha, conforme ilustrado na Figura 24. A restrição (5.32), juntamente com integralidade e minimização da variável U_{mjt}^I na função objetivo, determina a função piso: $\left\lfloor \frac{\mu_{mjtO}^{II,s} - \mu_{mjt1}^{I,s}}{TL_{max}} \right\rfloor$, em que o é o último lote em que houve produção. Por exemplo, supondo que $TL_{max} = 1.000$ minutos e que o tempo total de ocupação do tanque preparatório com o sabor j tenha sido 5.620 minutos, tem-se que $U_{mjt}^I = 5$, ou seja, estima-se que seja necessário a realização de 5 limpezas temporais, que serão descontadas da capacidade. Embora a restrição seja criada para todo lote $o \in O_{mt}$, a restrição que irá dimensionar o número de limpezas temporais no tanque preparatório é a restrição com maior limitante, ou seja, maior $\mu_{mjtO}^{II,s}$, em que houve produção no lote o ($Y_{mjtO} = 1$). É necessário colocar essa restrição para todo lote o para contabilizar a espera no final de cada lote do tanque preparatório, assim como é feito com a restrição (4.24) do Modelo MDSL-2E-LT.

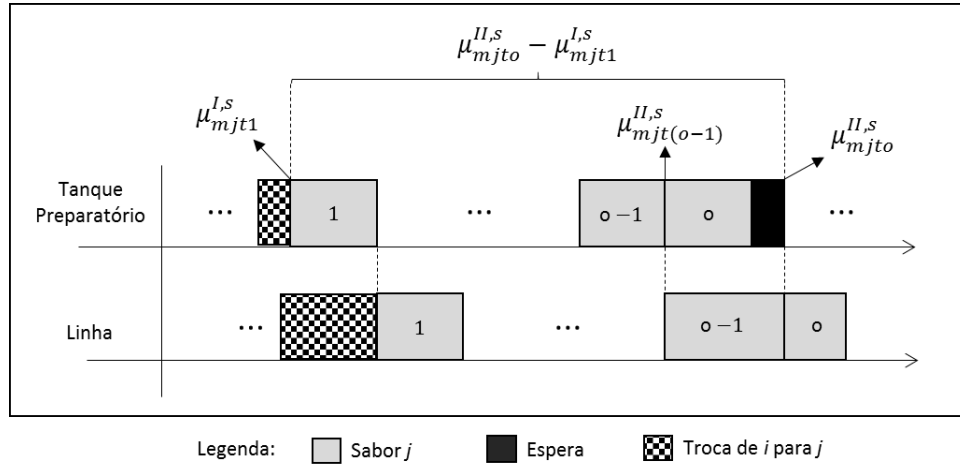


Figura 24 – Representação do tempo de produção contínua do item j . Fonte: a autora.

Os tempos despendidos para realização das limpezas temporais são subtraídos da capacidade através das restrições (5.34) e (5.35). O domínio das variáveis é definido por (5.36).

Conforme mencionado, o modelo MR1-O é um modelo aproximado, pois não considera as esperas que podem ocorrer no estágio II devido à limpeza temporal no estágio I, como exemplificado pela espera (D) na Figura 9. Também não são consideradas as esperas que ocorrem no estágio I devido à limpeza temporal no estágio II, como a espera (C) na Figura 9. Por não considerar essas esperas, a capacidade disponível nesse modelo fica super estimada, e por isso esse modelo é chamado de Otimista.

5.2.2 Modelo Relaxado 2 - Pessimista (MR2-P)

O Modelo Relaxado 2 - Pessimista (MR2-P) é dado pela função objetivo (5.31) e pelas expressões (4.5)-(4.24), (5.37)-(5.40) e (5.36).

$$\begin{aligned} \text{Min } Z^P = & \sum_{t \in T} \sum_{j \in J} (h_j^+ I_{jt}^+ + h_j^- I_{jt}^-) + \sum_{m \in M} \sum_{t \in T} \sum_{j \in J} \sum_{\substack{i \in J \\ i \neq j}} C_{ij} Z_{mijt} \\ & + \sum_{m \in M} \sum_{j \in J} \sum_{t \in T} Clp(U_{mjt}^I + U_{mjt}^{II}). \end{aligned} \quad (5.31)$$

s.a.:

$$(4.5), (4.6), (4.7), (4.8), (4.9), (4.10), (4.11), (4.12), (4.13), (4.14), (4.15), \\ (4.16), (4.17), (4.18), (4.19), (4.20), (4.21), (4.22), (4.23), (4.24).$$

$$U_{mjt}^I \geq \frac{(\mu_{mjt}^{II,s} - \mu_{mjt1}^{I,s}) - Mgde(1 - Y_{mjt}) + LP^{II}U_{mjt}^{II}}{TP_{max}} - 1, \quad \forall m \in M, \forall j \in J, \\ \forall t \in T, \forall o \in O_{mt}. \quad (5.37)$$

$$U_{mjt}^{II} \geq \frac{(\mu_{mjt0}^{II,e} - \mu_{mjt1}^{II,s}) + LP^I U_{mjt}^I}{TL_{max}} - 1, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.38)$$

$$\mu_{mjt0}^{I,e} \leq Cap_{mt} - LT^I \sum_{i \in J} U_{mit}^I - LT^{II} \sum_{i \in J} U_{mit}^{II}, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.39)$$

$$\mu_{mjt0}^{II,e} \leq Cap_{mt} - LT^{II} \sum_{i \in J} U_{mit}^{II} - LT^I \sum_{i \in J} U_{mit}^I, \quad \forall m \in M, \forall j \in J, \quad \forall t \in T. \quad (5.40)$$

$$X_{mjto}, I_{jt}^+, I_{jt}^-, V_{mjt}, \mu_{mjto}^{I,s}, \mu_{mjto}^{I,e}, \mu_{mjto}^{II,s}, \mu_{mjto}^{II,e} \geq 0; \quad U_{mjt}^I, U_{mjt}^{II} \in \mathbb{Z}_+; \quad (5.36)$$

$$Z_{mijt}, Y_{mjto} \in \{0, 1\}, \quad \forall m \in M, \forall i, j \in J, \forall t \in T, \forall o \in O_{mt}.$$

A função objetivo (5.31) é idêntica à do modelo MR1-O anterior.

As restrições (5.37) e (5.38) são parecidas com as restrições (5.32) e (5.33), respectivamente. Contudo, nelas é suposto que sempre que há uma limpeza temporal no estágio II, acontece uma espera no estágio I (espera (D) - Figura 9). Assim, essa espera entra na contagem do tempo total decorrido desde o início da produção do item j , com a parcela $LP^{II} U_{mjt}^{II}$ na restrição (5.37). Da mesma forma, supõe-se que sempre há uma espera no estágio II quando acontece uma limpeza temporal no estágio I (espera (C) - Figura 9), e o tempo despendido com essas limpezas é estimado na restrição (5.38) por meio da parcela $LP^I U_{mjt}^I$.

Os tempos decorridos com as limpezas temporais, juntamente com as esperas estimadas, são subtraídos da capacidade por meio das restrições (5.39) e (5.40). O domínio das variáveis é idêntico ao do modelo MR1-O, definido por (5.36).

Resumidamente, as esperas geradas em um estágio devido à realização de limpezas temporais no outro estágio nem sempre acontecem, mas no modelo MR2-P elas sempre são consideradas. Logo, a capacidade nesse caso fica subestimada, pois podem estar sendo descontados valores de espera que na realidade não existem.

A sincronia das soluções obtidas pelos modelos MR1-O e MR2-P também é realizada pelo Algoritmo 1. O parâmetro $NLotes_{mt}$ é calculado por $\sum_{o \in O_{mt}} \sum_{j \in J} Y_{mjto}$. Os parâmetros que definem o sequenciamento (SB_o) e a quantidade produzida por cada lote (Lt_o) também são determinados de acordo com as variáveis Z_{mijt} e X_{mjto} . A partir desses valores, a inserção das limpezas temporais e das esperas seguem de acordo com o Algoritmo 1.

O critério de parada na resolução dos modelos MR1-O e MR2-P é a otimalidade ou um tempo limite. Por serem modelos relaxados e não levarem em consideração as

restrições de determinação de limpeza temporal (4.27)-(4.42), são modelos com menos variáveis e restrições do que o MDSL-2E-LT. Nota-se, assim, que o esforço computacional para resolvê-los é bem menor do que para resolver o modelo integrado MDSL-2E-LT, possibilitando, por exemplo, a resolução de instâncias de maior porte, conforme pode ser visto em testes computacionais apresentados no Capítulo 6.

Resumo e relação entre os modelos propostos nas estratégias de solução do problema

Além do modelo integrado MDSL-2E-LT apresentado no Capítulo 4, foram apresentados diversos modelos utilizados nas heurísticas de resolução do problema de programação da produção de bebidas à base de frutas nas Seções 5.1 e 5.2. Com o intuito de resumir todos os modelos apresentados e comparar as decisões abordadas por cada um, apresenta-se um resumo destes na Tabela 18. Na primeira coluna é apresentada a abreviação do nome do modelo, juntamente com as expressões que o define. Na segunda coluna são indicados os estágios considerados na modelagem. Dado que o modelo aborda os dois estágios de produção, a consideração da sincronia na modelagem é mostrada na terceira coluna: sincronizado (S) ou não sincronizado (NS). E na última coluna é descrito se o modelo representa os custos e tempos de trocas dependentes (DS) ou independentes da sequência (IS). As características preenchidas com “-” não se aplicam ao modelo em questão. Cabe lembrar que a estrutura de modelagem do MDSL-2E-LT implica que esse modelo seja aplicado para ambos os casos (DS ou IS), e como os modelos MR1-O, MR2-P são derivados do MDSL-2E-LT, a mesma consideração é estendida a estes.

Tabela 18 – Resumo dos modelos desenvolvidos para o problema de programação da produção de bebidas à base de frutas.

Modelo		Estágio		Sincronia		Trocas	
Nome	Expressões	I	II	S	NS	DS	IS
MDSL-P	(5.1)-(5.12)	X		-	-	X	
MDL-P	(5.13);(5.2)-(5.6);(5.14)-(5.17)	X		-	-		X
MDSL-L	(5.18)-(5.23);(5.8)-(5.11);(5.24)		X	-	-	X	
MDL-L	(5.25)-(5.30)		X	-	-		X
MR1-O	(5.31);(4.5)-(4.24);(5.32)-(5.36)	X	X		X	X	X
MR2-P	(5.31);(4.5)-(4.24);(5.37)-(5.40);(5.36)	X	X		X	X	X
MDSL-2E-LT	(4.4)-(4.43)	X	X	X		X	X

5.3 Heurísticas *relax-and-fix*

As heurísticas baseadas em programação matemática apresentadas nas Seções 5.1 e 5.2 utilizam modelos próprios em sua composição, que independem do modelo integrado MDSL-2E-LT. Como mencionado anteriormente, esse modelo é difícil de ser resolvido por *solvers*, como o CPLEX, até para exemplares pequenos, como poder ser observado no

Capítulo 6. Para um melhor aproveitamento da eficiência do *solver* CPLEX, um ajuste de parâmetros foi realizado para resolução do modelo MDSL-2E-LT com as instâncias baseadas em dados reais. Uma descrição completa dos parâmetros que foram testados está apresentada no Apêndice E. Cada configuração de parâmetros testada é chamada de estratégia CPLEX ao longo do texto.

A dificuldade em resolver o modelo é o alto número de variáveis inteiras, binárias e restrições, que, dependendo do tamanho da instâncias, pode passar de dezenas de milhões. Devido à estrutura de modelagem utilizada, o parâmetro $|O_{mt}|$ tem grande influência no número de variáveis e restrições, que são em grande parte indexadas por $o \in O_{mt}$, como observado no Capítulo 4 pelas expressões (4.44), (4.45) e (4.46), e é difícil de ser estimado de forma precisa.

Assim, nessa seção são propostas heurísticas construtivas *relax-and-fix* de programação matemática baseadas no modelo MDSL-2E-LT. A heurística *relax-and-fix* é um método de decomposição que tenta construir iterativamente uma solução para o modelo por meio da resolução de diversos modelos menores derivados do modelo original (POCHET; WOLSEY, 2006). A ideia é explorar a estrutura do modelo, tentando resolvê-lo por partes, de maneira mais fácil. Essas heurísticas tem sido utilizadas com sucesso em diferentes classes de problemas de otimização (TOLEDO et al., 2015; MORENO; ALEM; FERREIRA, 2016), em especial nos problemas de dimensionamento e sequenciamento de lotes de grande porte (FERREIRA; MORABITO; RANGEL, 2010; SANTOS; ALMADA-LOBO, 2012; BALDO et al., 2014; STEINRÜCKEA, 2015).

De maneira geral, nas heurísticas *relax-and-fix* o conjunto de variáveis inteiras é particionado em P conjuntos disjuntos: $S_1, S_2, \dots, S_k, \dots, S_P$. O procedimento possui P iterações. A cada iteração k , apenas as variáveis pertencentes ao conjunto corrente S_k são mantidas como inteiras; as demais variáveis inteiras são relaxadas ou fixadas e o problema menor resultante (subproblema) é resolvido. Se a solução do subproblema for factível, as variáveis do conjunto, ou parte delas, são fixadas em seu valor atual.

Os passos das heurísticas *relax-and-fix* são apresentados no Algoritmo 4.

Como a cada iteração apenas um subconjunto de variáveis é mantido como inteiro, a ideia é que esses subproblemas sejam menores, mais fáceis e mais rápidos de serem resolvidos. Porém, pode ocorrer dos subproblemas ainda não serem resolvidos rapidamente. Neste caso, a cada iteração pode ser determinado um tempo limite para resolução do subproblema. Caso o subproblema seja resolvido até a otimalidade e ainda sobre tempo em uma iteração, esse tempo pode ser transferido para as próximas iterações.

Os diferentes critérios utilizados para particionar o conjunto de variáveis e para estabelecer a fixação das variáveis é que definem as mais diversas heurísticas do tipo *relax-and-fix*. Na heurística clássica, a partição é feita por períodos e eles são percorridos

Algoritmo 4: Algoritmo das heurísticas *relax-and-fix*.

```

1 INÍCIO
2 Defina a partição do conjunto de variáveis inteiras:  $S_1, S_2, \dots, S_P$ .
3 Defina o critério de fixação das variáveis.
4 Relaxe todas as variáveis inteiras do modelo.
5 para  $k \leftarrow 1$  até  $P$  faça
6   | Retorne a integralidade das variáveis da partição  $S_k$ .
7   | Resolva o MIP resultante.
8   | se a solução encontrada for factível então
9   |   | Fixe as variáveis da partição  $S_k$  de acordo com o critério de fixação.
10  | senão
11  |   | Pare. Não foi possível encontrar uma solução factível.
12 FIM

```

do primeiro até o último. Nas heurísticas *relax-and-fix* com *overlapping*, são utilizadas duas partições diferentes de variáveis, cada uma com P conjuntos disjuntos (BALDO et al., 2014). A partição $S_1^i, S_2^i, \dots, S_P^i$ é definida pelas variáveis que se tornarão inteiras a cada iteração k , $k = 1, \dots, P$, e a partição $S_1^f, S_2^f, \dots, S_P^f$ é definida pelas variáveis que serão fixadas a cada iteração k . Em cada iteração k , a integralidade das variáveis do conjunto S_k^i é reestabelecida, o problema é resolvido e as variáveis do conjunto S_k^f são fixadas. Na iteração seguinte, $k + 1$, as variáveis de S_{k+1}^i tornam-se inteiras e as variáveis de S_k^i se mantêm inteiras. Assim, as variáveis do conjunto S_k^i que não estavam no conjunto S_k^f e, portanto, não foram fixadas na iteração k , podem assumir outros valores durante a resolução do novo subproblema, acontecendo assim a sobreposição (*overlapping*). Para que a sobreposição funcione, deve-se ter que $S_1^f \subseteq S_1^i$, $S_2^f \subseteq S_1^i \cup S_2^i$, \dots , $S_P^f \subseteq S_1^i \cup S_1^i \cup \dots \cup S_P^i$.

Nesse trabalho considera-se as heurísticas *relax-and-fix* sem *overlapping*. Os subproblemas sempre são factíveis, uma vez que no modelo MDSL-2E-LT existem variáveis de atraso da produção. As partições são feitas por períodos. Duas estratégias para percorrer os períodos foram utilizadas: do primeiro para o último (*forward*) e do último para o primeiro (*backward*). Com o intuito de deixar os subproblemas ainda menores, também foram testadas partições por período combinadas com partições por máquinas (RFX 11 e RFX12).

Várias combinações de critérios de fixação foram utilizadas, gerando um total de 16 heurísticas do tipo *relax-and-fix*, resumidas na Tabela 19. Na primeira coluna é apresentada a sigla que representa a heurística, na segunda coluna está descrita a partição usada e como os conjuntos são percorridos e na última coluna são mostradas as variáveis que são relaxadas e fixadas. Nas heurísticas RFX13 a RFX16, as variáveis são fixadas a cada período somente quando existe produção para o lote o , do item j , no período t , no tanque preparatório/máquina m . Se não existe produção, essas variáveis ficam livres, ou seja, não são fixadas em zero. Essa técnica de fixar somente se existe produção foi usada por

Ferreira, Morabito e Rangel (2010) e gera bons resultados, e por esse motivo são aqui testadas.

Tabela 19 – Características das heurísticas *relax-and-fix* propostas.

Heurística <i>relax-and-fix</i>	Conjunto de partição variáveis	Variáveis fixadas
RFX1	Período (<i>forward</i>)	Y_{mjto} e Z_{mijt}
RFX2	Período (<i>backward</i>)	Y_{mjto} e Z_{mijt}
RFX3	Período (<i>forward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtok}^{II}
RFX4	Período (<i>backward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtok}^{II}
RFX5	Período (<i>forward</i>)	Y_{mjto} , Z_{mijt} , ω_{mjtok}^I e ω_{mjtok}^{II}
RFX6	Período (<i>backward</i>)	Y_{mjto} , Z_{mijt} , ω_{mjtok}^I e ω_{mjtok}^{II}
RFX7	Período (<i>forward</i>)	Y_{mjto} , Z_{mijt} e X_{mjto}
RFX8	Período (<i>backward</i>)	Y_{mjto} , Z_{mijt} e X_{mjto}
RFX9	Período (<i>forward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I , W_{mjtok}^{II} , ω_{mjtok}^I e ω_{mjtok}^{II}
RFX10	Período (<i>backward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I , W_{mjtok}^{II} , ω_{mjtok}^I e ω_{mjtok}^{II}
RFX11	Período (<i>forward</i>) Linha (<i>forward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtok}^{II}
RFX12	Período (<i>backward</i>) Linha (<i>forward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtok}^{II}
RFX13	Período (<i>forward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtok}^{II} , se $Y_{mjto} > 0$
RFX14	Período (<i>backward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtok}^{II} , se $Y_{mjto} > 0$
RFX15	Período (<i>forward</i>)	Y_{mjto} , Z_{mijt} e X_{mjto} , se $Y_{mjto} > 0$
RFX16	Período (<i>forward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I , W_{mjtok}^{II} , ω_{mjtok}^I e ω_{mjtok}^{II} , se $Y_{mjto} > 0$

5.4 Heurísticas de melhoria *fix-and-optimize*

Nas Seções 5.1, 5.2 e 5.3, foram apresentadas heurísticas para obtenção de soluções factíveis para o problema de programação da produção de bebidas à base de frutas. Nesta seção são propostas heurísticas que melhoraram uma solução factível para o problema, ou seja, heurísticas de melhoria. Na Seção 5.4.1 são apresentadas heurísticas *fix-and-optimize* clássicas e na Seção 5.4.2 heurísticas *fix-and-optimize* com busca em vizinhança. A partir de uma solução factível para o modelo MDSL-2E-LT, essas heurísticas decompõem esse modelo resolve apenas para parte dele.

5.4.1 Heurísticas *fix-and-optimize*

As heurísticas *fix-and-optimize* são heurísticas de melhoria, logo, precisam ser inicializadas com uma solução factível, diferentemente das heurísticas *relax-and-fix* que constroem uma solução factível para o problema. Entretanto, assim como as heurísticas *relax-and-fix*, as *fix-and-optimize* também exploram a estrutura dos modelos e resolvem subproblemas menores. A heurística *fix-and-optimize* surgiu como uma melhoria da *relax-and-fix* (POCHET; WOLSEY, 2006) e também têm sido muito utilizada em problemas de

dimensionamento e sequenciamento de lotes (HELBER; SAHLING, 2010; TOLEDO et al., 2015; GÖREN; TUNALI, 2015).

A abordagem para resolver o problema da heurística *fix-and-optimize* é similar com a da *relax-and-fix*. Enquanto na *relax-and-fix* o problema inicia relaxado, com apenas uma partição das variáveis considerada inteira, na *fix-and-optimize* todas as variáveis inteiras das partições estão inicialmente fixas no valor da solução incumbente. A cada iteração da *fix-and-optimize*, as variáveis de uma partição são deixadas livres para serem otimizadas, diferente da *relax-and-fix*, em que as variáveis de uma partição se tornam inteiras. Dizer que uma partição de variáveis está “livre” ou “liberada” para ser otimizada significa que essas variáveis não estão fixadas em nenhum valor, mas o domínio sob o qual elas devem ser otimizadas se mantém o mesmo.

Os passos seguidos pela heurística *fix-and-optimize* abordada estão apresentados no Algoritmo 5. Considere novamente a partição do conjunto de variáveis inteiras em P conjuntos disjuntos: S_1, \dots, S_P . A solução inicial (SOL^{ini}) é tomada como solução incumbente (SOL^{inc}). A solução corrente (SOL^{cor}) é inicialmente é vazia. A cada iteração, todas as variáveis inteiras do problema são fixadas no valor da solução incumbente, e apenas as variáveis inteiras do conjunto S_k ficam livres para serem otimizadas. Se o valor da solução do subproblema resultante ($FO(SOL^{cor})$) for melhor do que o valor da solução incumbente ($FO(SOL^{inc})$), esta é substituída. O processo é repetido enquanto o tempo decorrido ($Tempo_Decor$) não excede o tempo máximo disponível (Lim_Tempo) e estão sendo encontradas soluções melhores. Se todas as partições já tiverem sido percorridas pelo menos uma vez, o algoritmo é interrompido se não há mais melhoria, antes de atingir o limite de tempo. Convém observar que a busca por uma solução melhor é realizada de forma ordenada, pois as partições são sempre percorridas de 1 até P (*forward*) ou de P até 1 (*backward*).

Neste trabalho, as partições e as variáveis que são fixadas nas heurísticas *fix-and-optimize* estão apresentadas na Tabela 20. Com o intuito de testar a influência da solução inicial nas heurísticas *fix-and-optimize*, três estratégias para obtenção de solução inicial são testadas:

- 1) a solução da melhor estratégia CPLEX, gerando assim a heurística CPX+FXO (estratégia CPLEX+*fix-and-optimize*);
- 2) a solução da melhor heurística *relax-and-fix*, produzindo heurísticas RFX+FXO (*relax-and-fix* + *fix-and-optimize*);
- 3) a solução da melhor heurística dentre as de decomposição e as baseadas em modelos relaxados, gerando assim as heurísticas H+FXO (heurística + *fix-and-optimize*).

Algoritmo 5: Algoritmo das heurísticas *fix-and-optimize*.

```

1 INÍCIO
2 Obtenha uma solução inicial  $SOL^{ini}$  para o problema com função objetivo de valor  $FO(SOL^{ini})$ .
3 Defina a partição do conjunto de variáveis inteiras:  $S_1, S_2, \dots, S_P$ .
4 Defina  $Lim\_Tempo$  o tempo máximo disponível para a heurística fix-and-optimize.
5  $SOL^{inc} \leftarrow SOL^{ini}$ .
6  $SOL^{cor} \leftarrow \emptyset$ .
7 enquanto ( (  $Tempo\_Decor < Lim\_Tempo$  ) e (  $FO(SOL^{cor}) > FO(SOL^{inc})$  ) ) ou (
  iter  $\leq P$  ) faça
8    $k \leftarrow k + 1$  e  $iter \leftarrow iter + 1$ .
9   Fixe o conjunto escolhido de variáveis inteiras nos valores da solução incumbente  $SOL^{inc}$ .
10  Libere as variáveis do conjunto  $S_k$  para serem reotimizadas.
11  Resolva o MIP resultante (resolva até que atinja a otimalidade ou que alcance um tempo
12  determinado).
13   $SOL^{cor} \leftarrow$  solução do MIP-resultante.
14  se  $FO(SOL^{cor}) < FO(SOL^{inc})$  então
15     $SOL^{inc} \leftarrow SOL^{cor}$ .
16  se  $k = P$  então
17     $k \leftarrow 0$ 
18 FIM

```

Tabela 20 – Características das heurísticas *fix-and-optimize* propostas.

Heurística <i>fix-and-optimize</i>	Conjunto de partição variáveis	Variáveis fixadas
FXO1	Período (<i>forward</i>)	Y_{mjto} e Z_{mijt}
FXO2	Período (<i>backward</i>)	Y_{mjto} e Z_{mijt}
FXO3	Período (<i>forward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtok}^{II}
FXO4	Período (<i>backward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtok}^{II}
FXO5	Período (<i>forward</i>)	Y_{mjto} , Z_{mijt} , ω_{mjtok}^I e ω_{mjtok}^{II}
FXO6	Período (<i>backward</i>)	Y_{mjto} , Z_{mijt} , ω_{mjtok}^I e ω_{mjtok}^{II}
FXO7	Período (<i>forward</i>)	Y_{mjto} , Z_{mijt} e X_{mjto}
FXO8	Período (<i>backward</i>)	Y_{mjto} , Z_{mijt} e X_{mjto}
FXO9	Período (<i>forward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I , W_{mjtok}^{II} , ω_{mjtok}^I e ω_{mjtok}^{II}
FXO10	Período (<i>backward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I , W_{mjtok}^{II} , ω_{mjtok}^I e ω_{mjtok}^{II}
FXO11	Período (<i>forward</i>) Linha (<i>forward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtok}^{II}
FXO12	Período (<i>backward</i>) Linha (<i>forward</i>)	Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtok}^{II}

Além das combinações das heurísticas *relax-and-fix* e *fix-and-optimize* (BALDO et al., 2014; TOLEDO et al., 2015) para resolver problemas de dimensionamento de lotes, a *fix-and-optimize* também tem sido utilizada em abordagens híbridas com diversos outros métodos de solução, como, por exemplo, com busca local (JAMES; ALMADA-LOBO, 2011), com metaheurísticas (SEEANNER; ALMADA-LOBO; MEYR, 2013; CHEN, 2015; GÖREN; TUNALI, 2015; TOLEDO et al., 2016) e com *branch-and-cut* (CAMARGO; TOLEDO; ALMADA-LOBO, 2014).

5.4.2 Heurísticas *fix-and-optimize* com busca em vizinhanças

As heurísticas *fix-and-optimize* clássicas abordadas na literatura tentam melhorar uma solução inicial resolvendo subproblemas gerados a partir dessa solução. A forma com que esses subproblemas são determinados se deve a escolha *a priori* das partições de variáveis que serão liberadas para otimização. Essas partições são percorridas de forma ordenada e a ordem em que essas serão percorridas também é definida *a priori* (*forward* ou *backward*), como explicado na seção anterior. Logo, a *fix-and-optimize* explora o espaço de soluções factíveis do problema, porém fica limitada a exploração somente das partições definidas *a priori*. Além disso, ao percorrer as partições de maneira ordenada, a *fix-and-optimize* não explora outros subproblemas que talvez possam apresentar melhores soluções. Por exemplo, suponha que a partição do conjunto de variáveis seja por períodos, suponha também que o último período com as variáveis liberadas tenha sido o período $T = 1$ e que o próximo período a ter as variáveis liberadas é o período $T = 2$. Não existem garantias de que se, por exemplo, as variáveis do período $T = 4$ fossem liberadas para otimização após o período $T = 1$, esse subproblema não encontraria uma solução melhor do que o subproblema obtido ao liberar as variáveis do período $T = 2$.

Com o intuito de explorar melhorar o espaço de soluções em busca de soluções melhores, explora-se os conceitos de geração em vizinhança e busca de vizinhança juntamente com a *fix-and-optimize*. Uma vizinhança de um ponto x no espaço de soluções factíveis E , denotada por $N(x)$, é o conjunto de todos os pontos do espaço de solução E que estão próximos de x de acordo com alguma medida de proximidade. No caso do modelo MDSL-2E-LT, o ponto do espaço de solução é definido pelo valor das variáveis que determinam uma solução. Para a solução incumbente, por exemplo, o ponto do espaço é dado pela matriz de variáveis: $SOL^{inc} = (X, Y, Z, I^+, I^-, V, \mu^{Is}, \mu^{Ie}, \mu^{IIs}, \mu^{IIe}, \psi^I, \psi^{II}, W^I, W^{II}, \omega^I, \omega^{II})$.

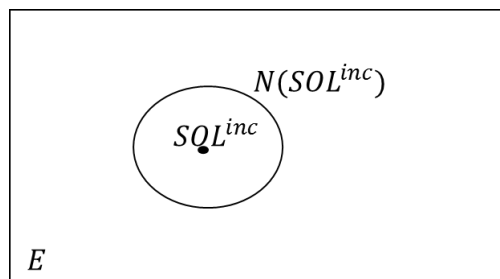


Figura 25 – Representação de uma vizinhança nas heurísticas *fix-and-optimize*.

Fazendo um paralelo com a *fix-and-optimize*, suponha que uma parte das variáveis de todas as partições, exceto da partição S_k de SOL^{inc} , estão fixadas. Todos os possíveis valores que as variáveis de S_k podem assumir definem uma vizinhança de SOL^{inc} , ou seja, definem $N(SOL^{inc})$, conforme representado na Figura 25. Se a partição é por período e S_k representa o período 4, por exemplo, tem-se que o ponto é

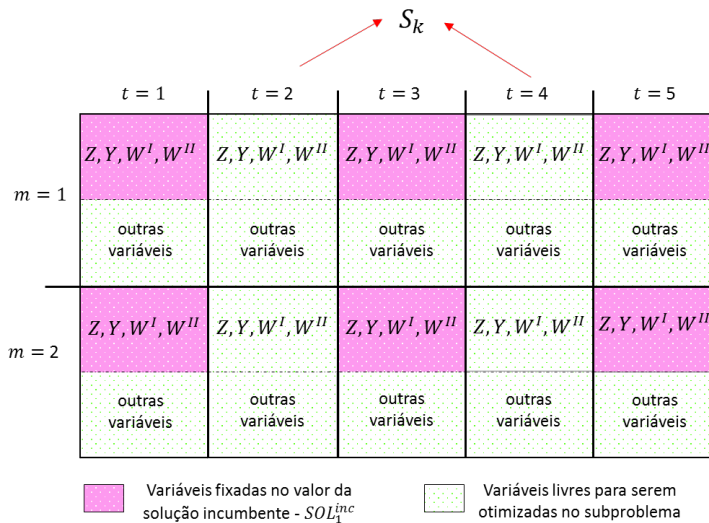
$(X_{mj4o}, Y_{mj4o}, Z_{mij4}, I_{j4}^+, I_{j4}^-, V_{mj4}, \mu_{mj4o}^{Is}, \mu_{mj4o}^{Ie}, \mu_{mj4o}^{IIs}, \mu_{mj4o}^{IIe}, \psi_{mj4ok}^I, \psi_{mj4ol}^{II}, W_{mj4ok}^I, W_{mj4ol}^{II}, \omega_{mj4ok}^I, \omega_{mj4ol}^{II})$ e a vizinhança são as soluções que se diferem pelos valores dessas variáveis. Considerando que as variáveis escolhidas para serem fixadas são $Z_{mijt}, Y_{mjto}, W_{mjto}^I$ e W_{mjto}^{II} , note que as variáveis $Z_{mijt}, Y_{mjto}, W_{mjto}^I$ e W_{mjto}^{II} , estão fixadas para $t \neq 4$, ou seja, todas as soluções vizinhas a essa possuem esses valores em comum, enquanto que as outras podem assumir qualquer valor.

O critério de definição das partições S_1, S_2, \dots, S_p estabelece a estrutura de vizinhança. Isto significa que se a partição S_k corresponde a dois períodos, por exemplo, a estrutura de vizinhança utilizada é do tipo período, com a liberação de dois períodos e denotada por $N_2^T(SOL^{inc})$. Os dois períodos dessa vizinhança são escolhidos aleatoriamente. Isso implica que o índice t deve ser variado para a definição de outras vizinhanças com a estrutura de período.

Nas heurísticas *fix-and-optimize* com busca em vizinhança pode haver mais de um tipo de estrutura de vizinhança. Nas Figuras de 26 a 30 está representado um exemplo em que são utilizadas duas estruturas de vizinhança: por período e por tanque preparatório/linha, e a busca por soluções nessas vizinhanças para um problema com 5 períodos e 2 tanques preparatórios/linhas.

Na Figura 26 está representada a vizinhança do tipo período para a solução incumbente SOL^{inc} , em que dois períodos estão livres para otimização. Note que as variáveis que não foram escolhidas para serem fixadas também estão livres para serem modificadas nessa vizinhança.

- Libere a partição S_k (períodos $t = 2$ e $t = 4$) e defina o subproblema.




- Otimize o subproblema.

Figura 26 – Liberação da partição S_k e definição do subproblema.

Ao resolver o subproblema MIP mantendo as variáveis escolhidas fixadas nos dois períodos escolhidos aleatoriamente, obtém-se uma outra solução representada na Figura 27. Suponha que essa solução seja melhor que a anterior, logo ela se torna a solução incumbente SOL^{inc} . Suponha ainda que a vizinhança explorada continue sendo $N_2^T(SOL^{inc})$, e que a partição S_{k+1} seja agora dada pelos períodos $t = 1$ e $t = 5$, escolhidos aleatoriamente.

- Solução otimizada.

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
$m = 1$	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}
	outras variáveis	outras variáveis	outras variáveis	outras variáveis	outras variáveis
$m = 2$	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}
	outras variáveis	outras variáveis	outras variáveis	outras variáveis	outras variáveis



 Valores das variáveis da solução incumbente - SOL^{inc}

- Se a solução encontrada é melhor do que a incumbente, substitua a incumbente e libere a próxima partição S_{k+1} .

Figura 27 – Solução após otimização do subproblema obtido com a partição S_k .

- Libere a partição S_{k+1} (períodos $t = 1$ e $t = 5$) e defina o subproblema.

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
$m = 1$	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}
	outras variáveis	outras variáveis	outras variáveis	outras variáveis	outras variáveis
$m = 2$	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}
	outras variáveis	outras variáveis	outras variáveis	outras variáveis	outras variáveis


 Variáveis fixadas no valor da solução incumbente - SOL^{inc} Variáveis livres para serem otimizadas no subproblema

- Otimize o subproblema.


Figura 28 – Liberação da partição S_{k+1} e definição do subproblema.

Na Figura 28 está representado o subproblema MIP que será resolvido com a

liberação dessa partição S_{k+1} , ou seja, os valores que podem ser atribuídos as variáveis livres formam a vizinhança da solução incumbente. A solução desse subproblema MIP está representada na Figura 29. Suponha que a solução não tenha sido melhorada, e que a estrutura de vizinhança a ser utilizada agora seja de um tanque preparatório/linha, ou seja, são liberadas para otimização as variáveis indexadas a um único tanque preparatório/linha. Isso significa que a vizinhança a ser explorada é $N_1^M(SOL^{inc})$. Supondo que $m = 1$ seja escolhido aleatoriamente e que essa seja a partição S_{k+2} , a Figura 30 representa o subproblema MIP obtido com essa estrutura de vizinhança. Esse subproblema é novamente resolvido para buscar uma solução melhor e o processo de busca em vizinhança é repetido.

- Solução otimizada.

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
$m = 1$	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}
	outras variáveis	outras variáveis	outras variáveis	outras variáveis	outras variáveis
$m = 2$	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}	Z, Y, W^I, W^{II}
	outras variáveis	outras variáveis	outras variáveis	outras variáveis	outras variáveis


 Valores das variáveis da solução incumbente - SOL^{inc}

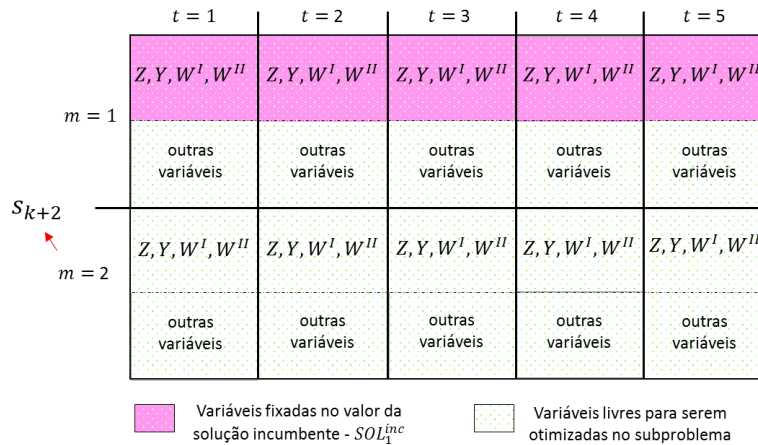
- Se solução encontrada é melhor do que a incumbente, substitua a incumbente e libere a próxima partição S_{k+2} .

Figura 29 – Solução após otimização do subproblema obtido com a partição S_{k+1} .

Fazendo um paralelo com as heurísticas *fix-and-optimize*, cujos passos estão apresentados no Algoritmo 5, a partição das variáveis definida na linha 3 representa as vizinhanças de S_1 a S_P , porém pertencentes a uma mesma estrutura, ou seja, de um mesmo tipo, por exemplo, períodos. A forma com que essas vizinhanças são percorridas é definida *a priori* e controlada pelo contador de iterações nas linhas 8, 15 e 16 do algoritmo. As heurísticas *fix-and-optimize* com busca em vizinhança diferenciam-se das heurísticas *fix-and-optimize* tradicionais justamente nesses passos. A maneira com que as vizinhanças são percorridas e a forma como são escolhidas as estruturas de vizinhança a cada iteração da heurística podem ser realizadas de diversas formas possíveis, inclusive podem ser determinadas por princípios de metaheurísticas de busca local, tais como a VNS (*Variable Neighborhood Search*) e a VNDS (*Variable Neighborhood Decomposition Search*).

Neste trabalho explorou-se a *fix-and-optimize* com busca em vizinhança e com princípios da metaheurística VNDS para o modelo MDSL-2E-LT. São propostas três heurísticas *fix-and-optimize* com busca em vizinhança:

- Libere a partição S_{k+2} ($m = 2$) e defina o subproblema.



- Otimize o subproblema.

Figura 30 – Liberação da partição S_{k+2} e definição do subproblema.

- *fix-and-optimize* com busca em vizinhança 1 (FXOV1);
- *fix-and-optimize* com busca em vizinhança 2 (FXOV2);
- *fix-and-optimize* com princípios de VNDS (FXO-VNDS).

As estruturas de vizinhança utilizadas nas heurísticas FXOV1, FXOV2 e FXO+VNDS são denotadas pela letra N , com o sobrescrito indicando o tipo de estrutura de vizinhança e o índice indicando quantos elementos devem ser escolhidos aleatoriamente:

- N_τ^T : nesse tipo de vizinhança estão as soluções encontradas com a resolução do subproblema MIP que libera para otimização τ períodos escolhidos aleatoriamente. Ou seja, nessa estrutura de vizinhança a partição das variáveis é realizada por períodos, e todas as variáveis indexadas ao período liberado são desfixadas e liberadas para serem otimizadas. Por exemplo, na vizinhança N_2^T , dois períodos devem ser escolhidos de forma aleatória para serem liberados.
- N_ζ^M : esta vizinhança é formada pelas soluções obtidas quando ζ tanques preparatórios/linhas são liberados para serem otimizados no subproblema MIP, ou seja, quando as variáveis pertencentes as partições formadas por esses ζ tanques preparatórios/linhas estão livres para serem otimizadas.
- N_ι^J : nessa vizinhança estão todas as possíveis soluções que podem ser encontradas com todas as variáveis fixadas exceto as variáveis inteiras indexadas aos ι itens.

Essas vizinhanças foram estabelecidas com base nos trabalhos de Helber e Sahling (2010) e Seannner, Almada-Lobo e Meyr (2013).

A partir daqui, quando ao longo do texto for mencionado que períodos, tanques preparatórios/linhas e/ou itens foram liberados para otimização, significa que a partição S_k da *fix-and-optimize* (Algoritmo 5) é com relação a esse conjunto. Ou seja, significa que as variáveis indexadas aos períodos, tanques preparatórios/linhas e/ou itens “liberados” não estão fixadas em nenhum valor (estão liberadas) e podem ser otimizadas no subproblema MIP na fase de busca local.

A forma com que essas vizinhanças são definidas (valores τ , ζ e ι) e a sequência em que serão exploradas varia dependendo da heurística (FXOV1, FXOV2 e FXO+VNDS). Para facilitar, a partir daqui considera-se como \mathcal{R} o conjunto das variáveis binárias e inteiras liberadas para otimização e \mathcal{F} o conjunto das variáveis que se mantém fixas no valor da solução incumbente. Note que a cada vizinhança percorrida as variáveis pertencentes a esses conjuntos são modificadas. As heurísticas FXOV1, FXOV2 e FXO-VNDS são descritas a seguir em detalhes.

Heurística *fix-and-optimize* com busca em vizinhança 1 (FXOV1)

Nas heurísticas *fix-and-optimize* com busca em vizinhança, resolve-se uma série de subproblemas MIP selecionados de forma aleatória em um algoritmo iterativo de troca de subproblemas (JAMES; ALMADA-LOBO, 2011). Os passos da heurística FXOV1 são apresentados no Algoritmo 6. Os parâmetros utilizados nesse algoritmo são:

- k identifica a vizinhança. As vizinhanças variam de $k = 1 \dots k_{max}$, conforme explicado mais adiante;
- it contabiliza o número de iterações sem melhoria com a utilização de uma mesma vizinhança;
- $Lim_it_viz_k$ é o número limite permitido de iterações sem melhoria utilizando uma mesma vizinhança, definido *a priori*;
- $it_sem_melhoria$ conta o número total de iterações sem melhoria do algoritmo;
- $Lim_sem_melhoria$ é o número limite permitido de iterações do algoritmo sem melhoria definido *a priori*;
- $Flag$ é um parâmetro auxiliar;
- $Hist_N^T$, $Hist_N^M$ e $Hist_N^J$ armazenam, respectivamente, os períodos, tanques preparatórios/linhas e itens utilizados nas iterações anteriores, para que eles não se repitam nas escolhas aleatórias de T , M e J que acontecem nas vizinhanças;
- SOL^{ini} , SOL^{inc} e SOL^{cor} são as soluções inicial, incumbente e corrente, respectivamente;
- $Tempo_Decor$ controla o tempo decorrido no algoritmo;

- Lim_Tempo é o limite máximo de tempo disponível para a heurística.

Após a inicialização dos parâmetros e a obtenção de uma solução inicial, inicia-se a busca em vizinhança pela primeira vizinhança $k = 1$. Se uma solução melhor do que a incumbente é encontrada após a resolução do subproblema MIP (SOL^{corr}) com a vizinhança k , a solução incumbente (SOL^{inc}) é substituída e os contadores de iteração sem melhoria $it_sem_melhoria$ e it são reinicializados. A busca reinicia com a mesma vizinhança k . Caso contrário, se a solução SOL^{corr} não é melhor do que a incumbente SOL^{inc} , o parâmetro it é incrementado. Se it atingir o número máximo de iterações sem melhoria com uma mesma vizinhança ($Lim_it_viz_k$), a busca é transferida para a próxima vizinhança $k + 1$. Se todas as k_{max} vizinhanças tiverem sido exploradas, a busca retorna para a primeira vizinhança $k = 1$. O processo é repetido enquanto houver tempo disponível e enquanto o número limite de iterações sem melhoria ($Lim_sem_melhoria$) não é alcançado.

Para a heurística FXOV1, as variáveis binárias e inteiras que definem a vizinhança são: Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtol}^{II} , respectivamente, a existência de produção, o sequenciamento e a existências de limpezas temporais nos estágios I e II. Essas variáveis foram escolhidas pois, dentre as variáveis testadas nas heurísticas *fix-and-optimize* resumidas na Tabela 20, as que utilizam essas variáveis foram as que obtiveram melhor desempenho nos testes computacionais apresentados no Capítulo 6.

As estruturas de vizinhança utilizadas na FXOV1 são três diferentes combinações entre as estruturas de vizinhança N_τ^T , N_ζ^M e N_ι^J descritas anteriormente, gerando as heurísticas FXOV1_T, FXOV1_T-M e FXOV1_T-M-J explicadas a seguir. Seja k_{max} o número total de vizinhanças utilizadas na heurística. As vizinhanças são exploradas de 1 até k_{max} .

FXOV1_T → Nessa heurística a única estrutura de vizinhança utilizada é N_τ^T . A ideia é explorar a liberação de períodos escolhidos aleatoriamente para que as variáveis desses períodos sejam otimizadas. O parâmetro τ deve variar de 1 até $\left\lfloor \frac{|T|}{2} \right\rfloor$. Por exemplo, em uma instância com 5 períodos, são exploradas $k_{max} = \left\lfloor \frac{5}{2} \right\rfloor = 2$ estruturas de vizinhança: N_1^T e N_2^T . Na vizinhança N_1^T , um período é escolhido aleatoriamente para ser liberado e na N_2^T são liberados dois períodos escolhidos de forma aleatória. Nesse caso a vizinhança $k = 1$ é a N_1^T e a vizinhança $k = 2$ é N_2^T .

FXOV1_T-M → Para essa heurística são exploradas as estruturas de vizinhança N_τ^T e N_ζ^M . Explora-se a otimização das variáveis por períodos e por tanques preparatórios/linhas, nessa ordem. O parâmetro τ varia da mesma forma que na heurística FXOV1_T e o parâmetro ζ segue o mesmo padrão variando de 1 até $\left\lfloor \frac{|M|}{2} \right\rfloor$. Por exemplo, em uma instância com $|T| = 5$ e $|M| = 2$, são exploradas $k_{max} = \left\lfloor \frac{5}{2} \right\rfloor + \left\lfloor \frac{2}{2} \right\rfloor = 3$ estruturas

de vizinhança: N_1^T , N_2^T e N_1^M . Os períodos τ e tanques preparatórios/linhas ζ que serão liberados são escolhidos aleatoriamente.

FXOV1_T-M-J → Nessa heurística são exploradas as três estruturas de vizinhança N_ι^J , N_τ^T e N_ζ^M . Entretanto, os parâmetros ι , τ e ζ e a ordem em que essas vizinhanças serão visitadas são definidos da mesma forma que em [Seeanner, Almada-Lobo e Meyr \(2013\)](#), explicados a seguir.

A ideia do trabalho de [Seeanner, Almada-Lobo e Meyr \(2013\)](#) é que o número de variáveis que serão fixadas a cada iteração aumente a cada troca de vizinhança. Na heurística FXOV1_T-M-J utiliza-se o mesmo princípio, entretanto, a ordem em que as vizinhanças N_ι^J , N_τ^T e N_ζ^M serão percorridas e os valores de ι , τ e ζ serão determinados de modo que somente o número de variáveis Z_{mijt} liberadas para otimização aumente a cada troca de vizinhança. Como os conjuntos O_{mt} , Q_{mt}^I e Q_{mt}^{II} são variáveis para cada m e t , e esses por sua vez são sorteados a cada iteração, o controle do aumento do número do número de variáveis de \mathcal{R} a cada troca de vizinhança não será feito para as variáveis Y_{mjto} , W_{mjtok}^I e W_{mjtol}^{II} nessa heurística FXOV1.

Note que a cada ι itens sorteados da vizinhança N_ι^J , o número de variáveis Z_{mijt} liberadas para otimização é $\iota \cdot |M| \cdot |J| \cdot |T|$; a cada τ períodos liberados para otimização na vizinhança N_τ^T , o número de variáveis Z_{mijt} liberadas é $\tau \cdot |M| \cdot |J| \cdot |J|$; e, o número de variáveis Z_{mijt} liberadas para otimização na vizinhança N_ζ^M é $\zeta \cdot |J| \cdot |J| \cdot |T|$.

Na aplicação estudada nessa tese tem-se que $|T| \leq |J|$ e que $|M| \leq |J|$. Em geral, $|M| < |T|$, mas podem existir casos em que $|M| > |T|$. Portanto, a forma como as vizinhanças N_ι^J , N_τ^T e N_ζ^M serão percorridas na FXOV1_T-M-J é dividida em dois casos: $|M| \leq |T|$ e $|M| > |T|$.

Caso 1: $|M| \leq |T|$. A ordem das vizinhanças é: $N_\iota^J \rightarrow N_\tau^T \rightarrow N_\zeta^M$. De fato, a ordem de grandeza dos conjuntos é dada por:

$$|M| < |T| < |J| \Rightarrow |M| \cdot |T| < |M| \cdot |J| < |T| \cdot |J|. \quad (5.41)$$

Multiplicando por $|J|$ a expressão (5.41), tem-se que $|M| \cdot |J| \cdot |T| < |M| \cdot |J| \cdot |J| < |T| \cdot |J| \cdot |J|$. Logo, quando um item $i \in J$ é liberado para otimização, isto implica que $|M| \cdot |J| \cdot |T|$ variáveis pertencerão a \mathcal{R} . Depois, em seguida, quando um período $t \in T$ é liberado para otimização, $|M| \cdot |J| \cdot |J|$ pertencerão a \mathcal{R} , com a garantia de que $|M| \cdot |J| \cdot |T| < |M| \cdot |J| \cdot |J|$. Logo, a partir da desigualdade (5.41) define-se as variações de ι , τ e ζ :

- para que $\iota \cdot |M| \cdot |J| \cdot |T| \leq |M| \cdot |J| \cdot |J|$, temos que $\iota \leq \left\lfloor \frac{|J|}{|T|} \right\rfloor$;
- para que $\tau \cdot |M| \cdot |J| \cdot |J| \leq |T| \cdot |J| \cdot |J|$, temos que $\tau \leq \left\lfloor \frac{|T|}{|M|} \right\rfloor$;
- para ζ a única limitação é o próprio $|M|$, ou seja, $\zeta \leq |M|$.

Caso 2: $|M| > |T|$. A ordem das vizinhanças é: $N_\iota^J \rightarrow N_\zeta^M \rightarrow N_\tau^T$. A explicação é idêntica ao caso 1. A cardinalidade dos conjuntos pode ser ordenada por:

$$|T| < |M| < |J| \Rightarrow |M| \cdot |T| < |T| \cdot |J| < |M| \cdot |J|. \quad (5.42)$$

Logo, a partir da desigualdade (5.42) define-se as variações de ι , τ e ζ :

- para que $\iota \cdot |M| \cdot |J| \cdot |T| \leq |T| \cdot |J| \cdot |J|$, temos que $\iota \leq \left\lfloor \frac{|J|}{|M|} \right\rfloor$;
- para que $\zeta \cdot |T| \cdot |J| \cdot |J| \leq |M| \cdot |J| \cdot |J|$, temos que $\zeta \leq \left\lfloor \frac{|M|}{|T|} \right\rfloor$;
- para τ a única limitação é o próprio $|T|$, ou seja, $\tau \leq |T|$.

Por exemplo, suponha um problema em que $|T| = 4$, $|M| = 2$ e $|J| = 5$. Suponha ainda que após os cálculos descritos acima, tem-se que $\iota \leq 1$, $\tau \leq 2$ e $\zeta \leq 2$. Logo as vizinhanças utilizadas serão N_1^J , N_1^T , N_2^T , N_1^M e N_2^M , sendo percorridas nessa ordem no Algoritmo 6 com $k = 1, \dots, k_{max} = 5$.

Heurística *fix-and-optimize* com busca em vizinhança 2 (FXOV2)

Na heurística *fix-and-optimize* com busca em vizinhança 2 (FXOV2), as variáveis utilizadas para serem fixadas ou liberadas para otimização são as mesmas que na FXOV1, Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtol}^{II} . As estruturas de vizinhanças utilizadas, a ordenação das vizinhanças e os valores de ι , τ e ζ também são definidos da mesma forma que para a FXOV1_T-M-J. A diferença entre a FXOV2 e a FXOV1_T-M-J é com relação ao critério com que as vizinhanças são percorridas e com relação ao critério de parada. Na FXOV2 as vizinhanças são percorridas de acordo com o mesmo critério que a VNDS.

Para um melhor entendimento da FXOV2 e da FXO-VNDS descrita mais adiante, apresenta-se a seguir, de forma breve, a metaheurística VNDS. A VNDS surge a partir da metaheurística VNS. A VNS é baseada em um princípio simples: mudança sistemática de vizinhança na busca por uma solução dentro do espaço de soluções factíveis (HANSEN; MLADENOVIC, 2001). Segundo Hansen e Mladenović (2003), embora a VNS seja muito eficiente para obter soluções para muitos problemas, ela ainda encontra dificuldade para resolver grandes instâncias. Assim, a VNDS surge como uma tentativa de melhoria da VNS para essas circunstâncias. Ao invés de fazer buscas em todo o espaço de solução, a VNDS limita-se a um espaço menor, e por isso é denominada decomposição, o que pode ajudar a resolver problemas maiores do que a VNS resolve. Os passos da VNDS são apresentados a seguir:

Inicialização. Selecione o conjunto de estruturas de vizinhança N_k , para $k = 1, \dots, k_{max}$, que serão usadas na busca; encontre uma solução inicial x ; escolha um critério de parada;

Algoritmo 6: Algoritmo das heurísticas *fix-and-optimize* com aleatoriedade.

```

1 INÍCIO
2 Inicialização:  $k \leftarrow 1$ ;  $it \leftarrow 0$ ;  $it\_sem\_melhoria \leftarrow 0$ ;  $Flag \leftarrow 0$ ;  $Hist\_N^T \leftarrow \emptyset$ ; e/ou
    $Hist\_N^M \leftarrow \emptyset$ ; e/ou  $Hist\_N^J \leftarrow \emptyset$ .
3 Obtenha uma solução inicial  $SOL^{ini}$  para o problema com valor  $FO(SOL^{ini})$ .
4  $SOL^{inc} \leftarrow SOL^{ini}$ .
5 enquanto  $\{ (Tempo\_Decor < Lim\_Tempo) \text{ e } (Flag = 0) \}$  faça
6     Fixe o conjunto de variáveis  $\mathcal{F}$  inteiras nos valores da solução incumbente  $SOL^{inc}$ .
7     Determine  $\mathcal{R}$  de acordo com a vizinhança  $k$  (determine o conjunto levando em consideração os
   conjuntos  $Hist\_N^T$ ,  $Hist\_N^M$  e  $Hist\_N^J$ ).
8     Libere as variáveis  $\mathcal{R}$  para serem reotimizadas.
9     Resolva o MIP resultante (resolva até que atinja a otimalidade ou até que alcance um tempo
   determinado).
10     $SOL^{cor} \leftarrow$  Solução do MIP resultante.
11    se  $FO(SOL^{cor}) < FO(SOL^{inc})$  então
12         $SOL^{inc} \leftarrow SOL^{cor}$ .
13         $it \leftarrow 0$ .
14         $it\_sem\_melhoria \leftarrow 0$ .
15    senão
16         $it\_sem\_melhoria \leftarrow it\_sem\_melhoria + 1$ .
17        se  $it = Lim\_it\_viz\_k$  então
18            se  $k = k_{max}$  então
19                 $k \leftarrow 1$ .
20            senão
21                 $k \leftarrow k + 1$ .
22        senão
23             $it \leftarrow it + 1$ .
24    se  $it\_sem\_melhoria = Lim\_sem\_melhoria$  então
25         $Flag \leftarrow 1$ .
26 FIM

```

Repita a seguinte sequência até que o critério de parada seja satisfeito:

(1) Faça $k \leftarrow 1$

(2) Repita os seguintes passos até que $k = k_{max}$:

(a) Agitação. Gere de forma aleatória um ponto x' utilizando a k -ésima vizinhança de x ($x' \in N_k(x)$); considere y como um conjunto de k características de solução presentes em x' , mas não presentes em x , ou seja, $y = x' \setminus x$.

(b) Busca Local. Encontre um ótimo local no espaço de y ou por inspeção ou por alguma heurística; denote a melhor solução encontrada como y' , e como x'' a correspondente solução em todo o espaço de solução E ($x'' = (x' \setminus y) \cup y'$).

(c) Mover ou não. Se este ótimo local é melhor do que a solução incumbente, mova para lá ($x \leftarrow x''$) e continue a busca com N_1 ($k \leftarrow 1$); caso contrário, faça $k \leftarrow k + 1$.

Para uma dada solução x , todas exceto k características (que podem ser variáveis,

por exemplo) são fixadas na fase de busca local. Ou seja, no conjunto y mencionado acima em (b), estão fixadas características de x' . Todas as possíveis formas de fixação (ou liberação) de características definem a vizinhança $N_k(x)$. O algoritmo inicia com uma solução aleatória x' de $N_1(x)$. Resolve-se o problema considerando somente as variáveis que não estão fixadas em y e um novo valor de solução é obtido. Se a nova solução obtida não é melhor do que a solução incumbente, a busca muda de vizinhança $k = k + 1$, para uma vizinhança maior do que a anterior, ou seja, com mais características livres do que fixadas, com o intuito de explorar um pouco mais o espaço E . Caso a nova solução seja melhor do que a incumbente, a busca volta para a primeira vizinhança ($k \leftarrow 1$) a partir da nova solução.

A única diferença entre a VNS e a VNDS é no passo 2(b): na VNS aplica-se um método de busca local em todo o espaço de solução E (iniciando de $x' \in N_k(x)$), enquanto que, na VNDS, a cada iteração é resolvido um subproblema em algum subespaço $V_k \subseteq N_k(x)$ com $x' \in V_k$. Na VNDS, a medida que as vizinhanças k são percorridas os subconjuntos V_k aumentam. Os passos da VNDS apresentados acima foram extraídos de Hansen e Mladenović (2003). Para mais detalhes sobre essas metaheurísticas, veja Hansen e Mladenović (2001), Hansen, Mladenović e Perez-Britos (2001) e Hansen e Mladenović (2003).

Na FXOV2 proposta, parte-se de uma solução inicial factível, ou seja, não há o passo de agitação. Uma vez que as estruturas de vizinhança estão bem definidas, a FXOV2 inicia otimizando o subproblema que possui as variáveis que estão liberadas (não fixadas) na primeira vizinhança. O conjunto de variáveis liberadas nas vizinhanças é escolhido de forma aleatória. Essa otimização é a fase de busca local. A nova solução obtida com essa otimização difere da solução incumbente apenas pelas variáveis que foram otimizadas, pois as fixadas são as mesmas da solução incumbente. Depois da obtenção da nova solução é preciso decidir para onde mover a busca, que refere-se ao passo (c) mover ou não. A cada nova solução melhor encontrada, a VNDS retorna para a primeira vizinhança e explora essa vizinhança novamente a partir da nova solução. Caso contrário, o número de variáveis não fixadas é aumentado com a mudança para a próxima vizinhança $k = k + 1$, para permitir passos mais longos e assim, mover a busca para outra parte do espaço de solução. Esses passos são repetidos até todas as k_{max} vizinhanças definidas tenham sido percorridas, ou até que algum outro critério de parada seja alcançado.

A FXOV2 não pode ser considerada uma FXO com princípios de VNDS pelo fato de que não há garantias que os subproblemas resolvidos aumentam de tamanho a cada vizinhança percorrida, ou seja, não existem garantias que o conjunto \mathcal{R} aumenta de cardinalidade conforme k aumenta. Isso é garantido na FXO-VNDS descrita mais adiante.

Os passos da FXOV2 são descritos no Algoritmo 7. Considere os mesmos parâmetros utilizados para o algoritmo 6. Assim como para FXOV1, as vizinhanças começam a ser

percorridas a partir de $k = 1$, após a inicialização dos parâmetros. São permitidas diversas iterações com a mesma vizinhança k controladas pelo contador it , que inicialmente vale zero. O parâmetro $Flag$ sinaliza o critério de parada por ter percorrido todas as k_{max} vizinhanças pré-definidas. Encontrada uma solução inicial para o problema, a solução incumbente recebe essa solução inicial, os conjuntos \mathcal{R} e \mathcal{F} são definidos de acordo com a vizinhança k e o subproblema MIP é otimizado. Se a solução obtida (SOL^{corr}) for melhor do que a solução incumbente (SOL^{inc}), a solução incumbente é substituída, o algoritmo reinicia com a primeira vizinhança $k = 1$ e o contador it é zerado. Caso contrário, o contador it é incrementado até um limite $Lim_it_viz_k$. Quando it alcança o limite $Lim_it_viz_k$, se ainda não foram percorridas todas as vizinhanças, o histórico da vizinhança k é apagado e uma nova vizinhança passa a ser explorada com o incremento de k . O algoritmo termina quando todas k_{max} vizinhanças tiverem sido percorridas, ou quando um tempo limite (Lim_Tempo) é alcançado. Como já mencionado anteriormente,

Algoritmo 7: Algoritmo das heurísticas *fix-and-optimize* com busca em vizinhança 2 (FXOV2) e das heurísticas *fix-and-optimize* com princípios de VNDS (FXO-VNDS).

```

1 INÍCIO
2 Inicialização:  $k \leftarrow 1$ ;  $it \leftarrow 0$ ;  $Flag \leftarrow 0$ ;  $Hist\_N^T \leftarrow \emptyset$ ;  $Hist\_N^M \leftarrow \emptyset$  e  $Hist\_N^J \leftarrow \emptyset$ .
3 Obtenha uma solução inicial  $SOL^{ini}$  para o problema com valor  $FO(SOL^{ini})$ .
4  $SOL^{inc} \leftarrow SOL^{ini}$ .
5 enquanto  $\{ (Tempo\_Decor < Lim\_Tempo) \text{ e } (Flag = 0) \}$  faça
6     Fixe o conjunto de variáveis  $\mathcal{F}$  inteiras nos valores da solução incumbente  $SOL^{inc}$ .
7     Determine  $\mathcal{R}$  de acordo com a vizinhança  $k$  (determine o conjunto levando em consideração os conjuntos  $Hist\_N^T$ ,  $Hist\_N^M$  e  $Hist\_N^J$ ).
8     Libere as variáveis de  $\mathcal{R}$  para serem reotimizadas.
9     Resolva o MIP resultante (resolva até que atinja a otimalidade ou até que alcance um tempo determinado).
10     $SOL^{cor} \leftarrow$  Solução do MIP resultante.
11    se  $FO(SOL^{cor}) < FO(SOL^{inc})$  então
12         $SOL^{inc} \leftarrow SOL^{cor}$ .
13         $k \leftarrow 1$ .
14         $it \leftarrow 0$ .
15    senão
16         $it \leftarrow it + 1$ .
17    se  $it = Lim\_it\_viz\_k$  então
18        se  $k = k_{max}$  então
19             $Flag \leftarrow 1$ .
20        senão
21             $k \leftarrow k + 1$ .
22            Limpa o conjunto histórico ( $Hist$ ) da vizinhança  $k$ .
23             $it \leftarrow 0$ .
24 FIM

```

as variáveis pertencentes a \mathcal{R} são escolhidas de acordo com cada uma das estruturas de vizinhança. Os ι itens, τ períodos e ζ tanques preparatórios/linhas são selecionados de forma aleatória, respeitando o histórico de utilização dos mesmos: $Hist_N^J$, $Hist_N^T$ e $Hist_N^M$.

Note que a diferença entre a heurística FXOV1 e a FXOV2 é que na segunda, sempre que existe uma melhoria, a busca retorna para a primeira vizinhança $k = 1$, e que o critério de parada, além de um tempo limite, é a busca ter percorrido todas as k_{max} vizinhanças.

Heurística *fix-and-optimize* com princípios de VNDS (FXO-VNDS)

A heurística *fix-and-optimize* com princípios de VNDS proposta segue os mesmos conceitos do trabalho de Seeanner, Almada-Lobo e Meyr (2013), que combina princípios de VNDS com a *fix-and-optimize* para resolver um modelo do tipo GLSP proposto para o problema de dimensionamento e sequenciamento de lotes multinível. A heurística proposta é denominada por FXO-VNDS.

A FXO-VNDS segue o mesmo algoritmo que a FXOV2 (Algoritmo 7). Entretanto, na heurística FXO-VNDS a cardinalidade do conjunto de variáveis que são liberadas para otimização deve crescer a medida que as vizinhanças são percorridas. Ou seja, a cardinalidade de \mathcal{R} para a vizinhança $k + 1$ deve ser maior do que a cardinalidade de \mathcal{R} para a vizinhança k . Para isso, na FXO-VNDS, as variáveis utilizadas para serem liberadas ou fixadas são somente as variáveis que indicam se existe produção Y_{mjto} e as variáveis de troca Z_{mijt} . Ou seja, a cardinalidade de \mathcal{R} , é definida pela cardinalidade dos conjuntos M , J , T e O_{mt} .

Na FXO-VNDS também são utilizadas as três vizinhanças N_ι^J , N_τ^T e N_ζ^M . A ordenação em que essas vizinhanças são percorridas e os valores de ι , τ e ζ são definidos de tal forma que a cardinalidade de \mathcal{R} aumente a cada vizinhança percorrida. Da mesma forma que na FXOV2, a ordem em que as vizinhanças são percorridas é dividida em dois casos, explicados a seguir.

Caso 1: $|M| \leq |T|$. A ordem das vizinhanças é: $N_\iota^J \rightarrow N_\tau^T \rightarrow N_\zeta^M$.

De fato, partindo da desigualdade (5.41), o número de variáveis Z_{mijt} liberadas para otimização aumenta à medida que essas vizinhanças são percorridas nessa ordem. Os valores de ι , τ e ζ devem ser definidos como explicado anteriormente na FXOV2, isto é, $\iota \leq \left\lfloor \frac{|J|}{|T|} \right\rfloor$, $\tau \leq \left\lfloor \frac{|T|}{|M|} \right\rfloor$ e $\zeta \leq |M|$.

A partir da desigualdade (5.41), conclui-se que o número de variáveis Y_{mjto} liberados para otimização a cada estrutura de vizinhança percorrida respeita a desigualdade:

$$\underbrace{\sum_{m \in M} \sum_{t \in T} O_{mt}}_{\text{libera o item } i} < \underbrace{|J| \sum_{m \in M} O_{mt'}}_{\text{libera o período } t'} < \underbrace{|J| \sum_t O_{m't'}}_{\text{libera } m'} \quad (5.43)$$

em que t' representa o período escolhido aleatoriamente na vizinhança N_τ^T e m' representa o tanque preparatório/linha escolhido aleatoriamente na vizinhança N_ζ^M . Note que essa desigualdade depende de m' e t' escolhidos aleatoriamente. Como o valor de O_{mt} depende

de m e t , para que a desigualdade (5.43) permaneça válida, os valores de ι , τ e ζ são definidos conforme descrito a seguir:

- (i) A variação do número de itens ι que podem ser liberados para otimização na vizinhança N_ι^J fica determinada por:

$$\iota \cdot \sum_{m \in M} \sum_{t \in T} O_{mt} < |J| \cdot \sum_{m \in M} O_{mt'} \Rightarrow \iota \leq \min_{t' \in T} \left\{ \left\lfloor \frac{|J| \cdot \sum_{m \in M} O_{mt'}}{\sum_{m \in M} \sum_{t \in T} O_{mt}} \right\rfloor \right\}. \quad (5.44)$$

Para o cálculo de ι em (5.44) são considerados somente os valores de $t' \in T$ que mantém a desigualdade (5.43) verdadeira, ou seja, somente os valores de $t' \in T$ que satisfazem a equação:

$$|J| > \frac{\sum_{m \in M} \sum_{t \in T} O_{mt}}{\sum_{m \in M} O_{mt'}} \quad (5.45)$$

Os valores de $t' \in T$ que não satisfazem (5.45) também não são considerados para a escolha aleatória dos períodos na vizinhança N_τ^T .

- (ii) A partir de (5.43) define-se também a variação de τ :

$$\tau \cdot |J| \sum_{m \in M} O_{mt'} < |J| \sum_{t \in T} O_{m't} \Rightarrow \tau \leq \min_{m' \in M, t' \in T} \left\{ \left\lfloor \frac{\sum_t O_{m't}}{\sum_{m \in M} O_{mt'}} \right\rfloor \right\}. \quad (5.46)$$

Os pares (m', t') que não satisfazem a primeira desigualdade de (5.46) não são considerados na escolha aleatória de N_ζ^M .

- (iii) A variação de ζ é de 1 a $|M|$, ou seja, $\zeta \leq M$. Os valores de m' são escolhidos de forma que a inequação (5.43) mantenha-se válida.

Os valores de ι , τ e ζ definidos acima garantem que a medida que as vizinhanças são percorridas, o número de variáveis Y_{mjto} pertencentes a \mathcal{R} aumente. Entretanto, como na FXO-VNDS são utilizadas as variáveis Y_{mjto} e Z_{mijt} , é preciso garantir que \mathcal{R} sempre aumente à medida que as vizinhanças $N_\iota^J \rightarrow N_\tau^T \rightarrow N_\zeta^M$ são percorridas, nessa ordem, para o caso 1 ($|M| \leq |T|$). Considerando os valores de ι , τ e ζ definidos para Z_{mijt} na heurística FXOV1-T-M-J, para a FXO-VNDS esse valores são definidos como segue:

- $\iota \leq \min \left\{ \left\lfloor \frac{|J|}{|T|} \right\rfloor; \min_{t' \in T} \left\{ \left\lfloor \frac{|J| \cdot \sum_{m \in M} O_{mt'}}{\sum_{m \in M} \sum_{t \in T} O_{mt}} \right\rfloor \right\} \right\};$
- $\tau \leq \min \left\{ \left\lfloor \frac{|T|}{|M|} \right\rfloor; \min_{m' \in M, t' \in T} \left\{ \left\lfloor \frac{\sum_t O_{m't}}{\sum_{m \in M} O_{mt'}} \right\rfloor \right\} \right\};$
- $\zeta \leq |M|.$

Portanto as vizinhanças para a FXO-VNDS para o caso $|M| \leq |T|$ estão bem definidas e podem ser utilizadas no Algoritmo 7.

Caso 2: $|M| > |T|$. A ordem das vizinhanças é: $N_\iota^J \rightarrow N_\zeta^M \rightarrow N_\tau^T$. A análise para a definição dos valores ι , ζ e τ , de tal forma que a cardinalidade de \mathcal{R} aumente à medida que as vizinhanças são percorridas, é feita da mesma forma que para o caso 1.

Analisa-se inicialmente os valores de ι , ζ e τ considerando somente as variáveis Y_{mjt} . A partir da desigualdade (5.42), o número de variáveis Y_{mjto} liberadas para otimização a cada estrutura de vizinhança percorrida respeita a desigualdade:

$$\underbrace{\sum_{m \in M} \sum_{t \in T} O_{mt}}_{\text{libera o item } i} < \underbrace{|J| \sum_t O_{m't}}_{\text{libera } m'} < \underbrace{|J| \sum_{m \in M} O_{mt'}}_{\text{libera o período } t'} \quad , \quad (5.47)$$

lembrando que t' representa o período escolhido aleatoriamente na vizinhança N_τ^T e m' representa o tanque preparatório/linha escolhido aleatoriamente na vizinhança N_ζ^M . Logo, define-se limites para ι , ζ e τ considerando que (5.47) não pode ser violada:

- (i) A variação do número de itens ι que podem ser liberados para otimização na vizinhança N_ι^J fica determinada por:

$$\iota \cdot \sum_{m \in M} \sum_{t \in T} O_{mt} < |J| \cdot \sum_{t \in T} O_{m't} \Rightarrow \iota \leq \min_{m' \in M} \left\{ \left\lfloor \frac{|J| \cdot \sum_{t \in T} O_{m't}}{\sum_{m \in M} \sum_{t \in T} O_{mt}} \right\rfloor \right\}. \quad (5.48)$$

Para o cálculo de ι em (5.48) são considerados somente os valores de $m' \in M$ que mantém a desigualdade (5.49) verdadeira, ou seja, somente os valores de $m' \in M$ que satisfazem a equação:

$$|J| > \frac{\sum_{m \in M} \sum_{t \in T} O_{mt}}{\sum_{t \in T} O_{m't}} \quad (5.49)$$

Os valores de $m' \in M$ que não satisfazem (5.49) também não são considerados para a escolha aleatória dos tanques preparatórios/linhas na vizinhança N_ζ^T .

- (ii) A partir de (5.47) define-se também a variação de ζ :

$$\zeta \cdot |J| \sum_{t \in T} O_{m't} < |J| \sum_{m \in M} O_{mt'} \Rightarrow \zeta \leq \min_{m' \in M, t' \in T} \left\{ \left\lfloor \frac{\sum_t O_{mt'}}{\sum_{m \in M} O_{m't}} \right\rfloor \right\}. \quad (5.50)$$

Os pares (m', t') que não satisfazem a primeira desigualdade de (5.50) não são considerados na escolha aleatória de N_τ^T .

- (iii) A variação de τ é de 1 a $|T|$, ou seja, $\tau \leq T$. Os valores de t' são escolhidos de forma que a inequação (5.43) mantenha-se válida.

Para garantir que \mathcal{R} sempre aumente à medida que as vizinhanças $N_\iota^J \rightarrow N_\tau^T \rightarrow N_\zeta^M$ são percorridas, nessa ordem, para o caso 2 ($|M| > |T|$), define-se ι , τ e ζ considerando Y_{mjto} e Z_{mijt} , ou seja, para a FXO-VNDS esse valores são definidos da seguinte forma:

- $\iota \leq \min \left\{ \left\lfloor \frac{|J|}{|M|} \right\rfloor ; \min_{m' \in M} \left\{ \left\lfloor \frac{|J| \cdot \sum_{t \in T} O_{m't}}{\sum_{m \in M} \sum_{t \in T} O_{mt}} \right\rfloor \right\} \right\};$
- $\zeta \leq \min \left\{ \left\lfloor \frac{|M|}{|T|} \right\rfloor ; \min_{m' \in M, t' \in T} \left\{ \left\lfloor \frac{\sum_t O_{mt'}}{\sum_{m \in M} O_{m't}} \right\rfloor \right\} \right\};$
- $\tau \leq |T|.$

Logo as vizinhanças para a FXO-VNDS para o caso $|M| \leq |T|$ também estão bem definidas e são utilizadas no Algoritmo 7.

Resumidamente, nesta seção foram apresentadas heurísticas *fix-and-optimize* clássicas, dois tipos de heurísticas *fix-and-optimize* com busca em vizinhança (FXOV1 e FXOV2) e uma heurística *fix-and-optimize* com princípios de VNDS (FXO-VNDS).

Neste capítulo apresentou-se diversas heurísticas para resolver o problema de programação da produção de bebidas à base de frutas. Foram apresentadas estratégias de decomposição do problema em estágios, heurísticas utilizando modelos relaxados, heurísticas *relax-and-fix* para o modelo integrado MDSL-2E-LT e heurísticas de melhoria do tipo *fix-and-optimize*. No próximo capítulo esses métodos são avaliados computacionalmente.

6 Experimentos computacionais

Para avaliar o modelo matemático e o desempenho dos métodos de solução propostos, neste trabalho foram realizados diversos experimentos computacionais, descritos nesse capítulo. O foco dos testes é resolver o problema de programação da produção considerando tempos e custos de troca dependentes da sequência. As heurísticas de decomposição do tipo IS foram avaliadas separadamente no Apêndice F com um conjunto de instâncias diferente dos exemplares utilizados nesse capítulo. Mais detalhes sobre as heurísticas do tipo IS também podem ser encontrados em [Toscano, Ferreira e Morabito \(2015\)](#) e [Toscano, Ferreira e Morabito \(2017\)](#).

A Tabela 21 resume as estratégias propostas para resolução do problema de programação da produção de bebidas à base de frutas que foram apresentados nos capítulos anteriores, e que são avaliadas nesse capítulo.

Tabela 21 – Métodos de solução avaliados nos experimentos computacionais.

<i>Branch-and-cut</i> do CPLEX	Ajuste dos parâmetros do CPLEX para resolver instâncias com o modelo MDSL-2E-LT
Heurísticas de decomposição do problema	HE1A-DS e HE1B-DS HE2A-DS e HE2B-DS
Heurísticas baseadas em modelos relaxados	H-SUP H-SUB
<i>Branch-and-cut</i> do CPLEX	Resolução de instâncias com o modelo MDSL-2E-LT usando como solução inicial a heurística com melhor desempenho dentre HE1A-DS, HE1B-DS, HE2A-DS e HE2B-DS
<i>Branch-and-cut</i> do CPLEX	Resolução de instâncias com o modelo MDSL-2E-LT usando como solução inicial a heurística com melhor desempenho dentre H-SUP e H-SUB
Heurísticas <i>relax-and-fix</i>	Resolução de instâncias com o modelo MDSL-2E-LT e <i>Branch-and-cut</i> do CPLEX
Heurísticas de melhoria	⇒ Heurísticas <i>fix-and-optimize</i> aplicadas ao modelo MDSL-2E-LT com solução inicial: <ul style="list-style-type: none"> – Estratégia CPLEX com melhor desempenho – <i>Relax-and-fix</i> com melhor desempenho – Heurística com melhor desempenho dentre: HE1, HE2, H-SUP, H-SUB ⇒ <i>fix-and-optimize</i> com busca em vizinhança: FXOV1_T, FXOV1_T-M, FXOV1_T-M-J e FXOV2 ⇒ <i>fix-and-optimize</i> com princípios de VNDS: FXO-VNDS

Foram realizados testes para ajuste dos melhores parâmetros do CPLEX na solução de instâncias do modelo MDSL-2E-LT. Avaliou-se o desempenho dos métodos de decomposição (HE1A-DS, HE1B-DS, HE2A-DS e HE2B-DS) e das heurísticas baseadas nos modelos relaxados (H-SUP e H-SUB). Analisou-se as soluções obtidas a resolução

do modelo MDSL-2E-LT pelo *solver* quando são fornecidas soluções iniciais através de heurísticas. Dois testes desse tipo foram realizados: iniciando o *solver* com a solução obtida pela heurística com melhor desempenho dentre HE1A-DS, HE1B-DS, HE2A-DS e HE2B-DS e com a solução obtida pela heurística com melhor desempenho dentre H-SUP e H-SUB. Todas as estratégias das heurísticas *relax-and-fix* aplicadas ao modelo integrado MDSL-2E-LT também foram testadas. Todas as 12 estratégias *fix-and-optimize* descritas na Tabela 20 foram testadas com três soluções iniciais diferentes: a parametrização do CPLEX com o melhor desempenho, a *relax-and-fix* com melhor desempenho e a heurística com melhor desempenho dentre HE1A-DS, HE1B-DS, HE2A-DS, HE2B-DS, H-SUP e H-SUB. As heurísticas *fix-and-optimize* com busca em vizinhança e com princípios de VNDS foram testadas com a solução inicial dada pela heurísticas mais rápida dentre todas as heurísticas de decomposição (HE1A-DS, HE1B-DS, HE2A-DS e HE2B-DS), as baseadas em modelos relaxados (H-SUP e H-SUB) e as *relax-and-fix*.

No total 80 estratégias de solução para o problema foram testadas, sendo 17 estratégias CPLEX, 4 heurísticas de decomposição do problema em estágios, 2 heurísticas baseadas em modelos relaxados, 16 estratégias *relax-and-fix*, 36 heurísticas *fix-and-optimize* (12 estratégias com 3 soluções iniciais diferentes), 4 heurísticas *fix-and-optimize* com busca em vizinhança e uma heurística *fix-and-optimize* com princípios de VNDS. Esses testes se dividiram em duas partes, descritas nas Seções 6.3 e 6.4.

Os modelos e heurísticas foram programados na linguagem de modelagem AMPL (FOURER; GAY; KERNIGHAN, 2003). O *software* de otimização CPLEX 12.6.1 foi o *solver* utilizado para resolução de instâncias dos modelos. Os testes foram realizados em um computador Intel Core i7 com dois núcleos de 3.6GHz cada e memória de 16GB. Os critérios de parada estabelecidos para todos os métodos foi de 3.600 segundos, ou um *gap* menor ou igual a 1% de otimalidade.

6.1 Descrição das instâncias

Os testes computacionais foram conduzidos com dois tipos de conjuntos de instâncias: o primeiro baseado em dados reais coletados em uma das empresas visitadas durante o desenvolvimento desse trabalho, e o segundo com instâncias realistas geradas aleatoriamente com base nestes dados coletados. A seguir esses dois conjuntos de instâncias são descritos.

6.1.1 Instâncias baseadas em dados reais

As instâncias reais são nomeadas por R1, R2, R3 e R4. As instâncias R1, R2 e R3 são baseadas em dados reais e cada uma corresponde a um mês de produção da Fábrica A (Apêndice A). A instância R4 corresponde a 13 semanas (períodos) e foi utilizada para

avaliar o comportamento dos métodos de solução em instâncias cujo modelo MDSL-2E-LT apresenta um grande número de variáveis e restrições, uma vez que neste trabalho o objetivo é otimizar a programação da produção em nível operacional (no máximo 1 mês, 4 ou 5 períodos). A partir dessas instâncias foram criadas duas instâncias menores (S1 e M1) com base nos dados reais, que podem ser resolvidas até a otimalidade em tempo hábil.

Os dados reais utilizados nessas instâncias são a demanda em cada período, o tempo de execução de uma limpeza temporal, o tempo de troca entre itens, as capacidades dos tanques preparatórios e tanques pulmões, o tempo de preparo de um lote, as velocidades dos pasteurizadores e das máquinas de envase, os parâmetros TP_{max} e TL_{max} e a capacidade Cap_{mt} . Cada período refere-se a uma semana, logo as instâncias R1, R2 e R3 têm entre 4 e 5 períodos. Conforme discutido no Apêndice A, a programação da produção na empresa é realizada para um mês, porém revista semanalmente. A ideia é gerar um plano de produção maior, tendo a previsão da demanda ao longo de um mês, e rever a programação da produção semanalmente, sempre que necessário. Os custos de estoque, atraso, troca e limpeza temporal não foram disponibilizados pela empresa por motivos de confidencialidade, o que limitou a comparação das soluções das abordagens aqui propostas com as soluções da empresa do ponto de vista dos custos da solução. Esses valores foram gerados aleatoriamente para refletir a prioridade da empresa em atender a demanda dos consumidores sem atraso, provendo o maior nível de serviço possível. Segundo os programadores da produção ouvidos em visitas realizadas a fábricas do setor, a principal meta das empresas deste setor é sempre atender os consumidores e estocar o menos possível. Trocas e limpezas temporais desnecessárias ou exageradas também não são desejadas pelos tomadores de decisão. Porém, o maior objetivo é diminuir atrasos e estoques, mesmo que isso implique na realização de diversas e limpezas temporais. Assim, o custo de atraso de uma unidade foi estimado em 100, o custo de estoque de uma unidade em 10, os custos de troca variando no intervalo $[1,10]$ e o custo de uma limpeza temporal 1. Como discutido no Capítulo 3, esses pesos visam somente estabelecer uma prioridade entre os objetivos de uma programação da produção efetiva.

A Tabela 22 a seguir apresenta a quantidade de elementos dos conjuntos M , T , J , O_{mt} , Q_{mt}^I e Q_{mt}^{II} . A quantidade de elementos dos conjuntos O_{mt} , Q_{mt}^I e Q_{mt}^{II} são calculadas conforme descrito no Capítulo 4. Cada par m e t envolve valores diferentes para $|O_{mt}| \in \mathbb{Z}_+$, logo, na tabela os valores para $|O_{mt}|$ são apresentados em notação de intervalo.

Além das 6 instâncias descritas acima (S1, M1, R1, R2, R3 e R4), nos testes computacionais do modelo MDSL-2E-LT também foi utilizada uma variação da instância S1, em que o valor de O_{mt} é reduzido pela metade, denotada por S2. Essa instância foi criada com o intuito de avaliar o impacto no tempo computacional do número de lotes (cardinalidade de O_{mt}) na solução do problema. A redução do valor de $|O_{mt}|$ pela metade não causa nenhum prejuízo à qualidade da solução, uma vez que em testes preliminares

Tabela 22 – Quantidade de elementos dos conjuntos M , T , J , O_{mt} , Q_{mt}^I e Q_{mt}^{II} para cada instância.

	S1	M1	R1	R2	R3	R4
$ M $	2	2	2	2	2	2
$ T $	2	3	4	5	4	13
$ J $	3	4	5	5	5	5
$ O_{mt} $	12	[5,10]	[44,54]	[42,51]	[42,51]	[44,54]
$ Q_{mt}^I $	6	6	5	5	5	5
$ Q_{mt}^{II} $	3	3	2	2	2	2

com a instância S1 percebeu-se que nem todos os lotes disponíveis $o \in O_{mt}$ eram utilizados.

6.1.2 Instâncias geradas aleatoriamente

Para uma investigação mais detalhada do desempenho e das soluções obtidas com a resolução do modelo MDSL-2E-LT e dos demais métodos de solução propostos para resolver o problema de programação da produção de bebidas à base de frutas, apresentados nos capítulos anteriores, considerou-se mais instâncias além das apresentadas na Seção 6.1. O intuito é que essas instâncias representem diversas situações para o problema de programação da produção de bebidas à base de frutas, ainda de forma realista. A partir de uma análise das informações coletadas durante as visitas às empresas do setor, foi elaborado um gerador de dados aleatórios que representa as características dos problemas resolvidos pelas empresas do setor. Cada instância será nomeada por J_M_T_-E, em que J indica o número de itens, M o número de tanques preparatórios/linhas, T o número de períodos e E o número do exemplar da classe J_M_T_. Por exemplo, a instância J10M2T4-2 é o exemplar de número 2 de uma classe com 10 itens, 2 tanques preparatórios/linhas e 4 períodos.

Na empresa em que os dados foram coletados são produzidos somente 5 itens, porém nas outras empresas que foram visitadas verificou-se que esse número pode ser maior. Assim, propõem-se que o número de itens, $|J|$, de uma instância varie dentro do conjunto $\{5, 10, 15, 20\}$. Nas empresas visitadas foram encontrados até 6 tanques preparatórios e linhas dedicados, logo o número de tanques preparatórios/linhas, $|M|$ deve ser escolhido do conjunto $\{2, 4, 6\}$. Como o objetivo deste trabalho é avaliar a programação da produção em um nível operacional, o número de períodos foi escolhido do conjunto $\{2, 4, 6\}$. A partir da combinação desses parâmetros, foram determinados 6 classes para a geração das instâncias a saber:

J3M2T2	J10M2T4	J15M6T4
J5M2T4	J10M4T4	J20M6T6.

Para cada uma dessas classes foram geradas 3 instâncias.

A seguir é descrito como os parâmetros do modelo MDSL-2E-LT são obtidos para cada instância.

- **Quantidade de bebida em cada item.** De acordo com as informações coletadas, a quantidade de bebida em um item pode variar entre 0,9 e 12 litros, pois um item é considerado um fardo com 6 frascos de determinada bebida. Assim, o parâmetro ρ_j é escolhido através de uma distribuição uniforme: $\rho_j \sim U[0, 9; 12]$.

- **Velocidade das máquinas de envase.** A velocidade das máquinas de envase pode variar entre de 40 a 200 litros por minuto. Logo, o parâmetro S_m é obtido através de uma distribuição uniforme: $S_m \sim U[40; 200]$.

- **Lote máximo.** O lote máximo depende da capacidade dos tanques preparatórios. Foram encontrados tanques de 10.000 até 20.000 litros, logo o lote máximo é definido nesse intervalo: $UB_j \sim U[10.000; 20.000]$.

- **Lote mínimo.** O lote mínimo é definido como uma porcentagem do lote máximo variando entre 20% e 60%, dependendo do sabor e da fábrica. Assim, o lote mínimo é estabelecido por $LB_j = \theta \cdot UB_j$ com $\theta \sim U[0, 20; 0, 60]$.

- **Tempo para realização das limpezas temporais.** Os tempos para realização das limpezas temporais nos tanques preparatórios e nas linhas são idênticos para todas as empresas visitadas. Assim, esses parâmetros são mantidos no valor original para todas as instâncias: $LT^I = 50$ minutos e $LT^{II} = 300$ minutos.

- **Tempo máximo sem realização de limpeza.** O tempo máximo decorrido sem a realização de uma limpeza também é fixo nas empresas que foram visitadas: $TP_{max} = 1445$ minutos (aproximadamente 24 horas) e $TL_{max} = 2885$ minutos (aproximadamente 48 horas).

- **Tempo de preparo.** O tempo de preparo gira em torno de 100 minutos (uma hora e meia), podendo ter uma variação de alguns minutos para mais ou para menos. Assim, esse parâmetro é definido por uma distribuição normal com média 100 e desvio padrão 10: $Tp \sim N(100, 10^2)$.

- **Estoque e Atraso iniciais.** Considera-se que não existem estoques e atrasos no início dos períodos: $I_{jt}^+ = 0$ e $I_{jt}^- = 0$.

- **Demanda.** A limpeza temporal é uma característica presente nas fábricas de bebidas à base de frutas. Assim, para gerar as demandas para o problema de programação da produção é necessário considerar demandas que para serem atendidas, necessitem de limpezas temporais na programação. A geração da demanda é dada pelos seguintes passos:

- 1) Define-se o menor e o maior número de litros que podem ser envasados em TL_{max}

minutos entre todas as linhas m :

$$menor = \min_{\forall m \in M} \{[S_m \cdot TL_{max}]\} \quad e \quad maior = \max_{\forall m \in M} \{[S_m \cdot TL_{max}]\}.$$

- 2) Para cada período $t \in T$, são escolhidos aleatoriamente dois itens i e j que terão suas demandas geradas da seguinte forma:

$$d_{it} \sim U \left[\frac{menor}{\rho_i}; \frac{maior}{\rho_i} \right] \quad e \quad d_{jt} \sim U \left[\frac{maior}{\rho_j}; \frac{2 \cdot maior}{\rho_j} \right].$$

Com isso, garante-se que, em cada período, pelos menos dois itens têm demandas que podem gerar limpezas temporais. Considere J_t^* o conjunto dos itens i e j que tiveram a capacidade calculada no período t , conforme explicado acima.

- 3) O cálculo da demanda dos outros itens em cada período é baseado na capacidade real de produção do período (8.550 minutos) e em quanto dessa capacidade já foi utilizada pelas demandas geradas para os itens $i, j \in J_t^*$, como explicado acima em 2). Calcula-se o quanto do tempo disponível foi utilizado pelos itens $i, j \in J_t^*$:

$$cap_disp_t = \left(\sum_{m \in M} 8.550 \cdot S_m \right) - \sum_{j \in J_t^*} \rho_j \cdot d_{jt}.$$

Apesar da referência para o cálculo das demandas ser a capacidade real, a capacidade de cada instância é calculada mais adiante.

- 4) O valor de cap_disp_t é dividido entre os outros itens para gerar um limite máximo de geração de demanda:

$$lim_{jt} = \frac{cap_disp_t}{|J| - |J_t^*| - 1} \quad \forall t \in T, \forall j \in J \setminus J_t^*.$$

- 5) As demandas dos outros itens são geradas por meio de uma distribuição uniforme:

$$d_{j,t} \sim U \left[0; \frac{lim_{jt}}{\rho_j} \right], \quad \forall t \in T, j \in J \setminus J_t^*.$$

• **Tempo de troca no primeiro estágio.** O tempo de troca do item fantasma para qualquer item é o tempo de uma limpeza temporal, pois no início de cada período existe uma limpeza obrigatória com o mesmo tempo da limpeza temporal. Dependendo do sabor produzido anteriormente ao próximo item que será produzido, é necessário realizar uma limpeza ou simplesmente um enxágue. Assim, para os demais itens, os tempos de troca entre os itens no primeiro estágio estão em torno da metade do tempo da limpeza temporal, podendo variar em torno desse valor. Assim, esse parâmetro é definido através de uma distribuição normal com média 30 e desvio-padrão 5: $TC_{ij}^I \sim N(30, 5^2)$.

• **Tempo de troca no segundo estágio.** Da mesma forma que no primeiro estágio, o tempo de troca de qualquer item para o item fantasma no segundo estágio é o

tempo de uma limpeza temporal, uma vez que a primeira limpeza do período depende o mesmo tempo que a limpeza temporal. Para os demais itens, os tempos de troca entre os itens são em torno da metade da limpeza temporal e portanto, define-se: $TC_{ij}^{II} \sim N(150, 30^2)$. Ambos os tempos de troca do primeiro e segundo estágios são gerados respeitando a desigualdade triangular.

- **Custo de limpeza temporal.** O custo de limpeza temporal é definido como uma unidade, ou seja, $Cl_t = 1$. Cabe lembrar que o objetivo é penalizar as limpezas temporais apenas para garantir que a demanda seja atendida otimizando a capacidade.

- **Custos de troca.** Os custos de troca levam em consideração as trocas realizadas nos dois estágios de produção. Como não foi possível obter os custos reais de troca nas empresas visitadas, esses custos são gerados também com o intuito de melhor aproveitamento da capacidade, assim como o custo da limpeza temporal. Os custos de troca são calculados por:

- $C_{j_{i_0}} = 0, \forall j \in J$, ou seja, o custo de troca do item fantasma para qualquer item é sempre nulo;

- $C_{i_0j} = Cl_t, \forall j \in J$, pois o custo de troca do item fantasma para o primeiro item do período é o custo de uma limpeza temporal;

- $C_{ij} = 10 * \left(\frac{TC_{ij}^I + TC_{ij}^{II}}{\max_{\forall i, j \in J \setminus i_0} \{TC_{ij}^I + TC_{ij}^{II}\}} \right)$, ou seja, esse valor normaliza a soma dos tempos de troca dos dois estágios pelo maior tempo de troca, e é multiplicado por 10 para ser um pouco maior do que o custo de uma limpeza temporal.

- **Custos de estoque e atraso.** Para os exemplares baseados em dados reais, usou-se custo de atraso igual a 100 e custo de estoque igual a 10. Seguindo a mesma lógica, porém com o intuito de variar um pouco os custos entre os itens, os custos de estoque e atraso são gerados da seguinte forma: $h_j^+ \sim U[10; 20]$ e $h_j^- = 10 * h_j^-, \forall j \in J$.

- **Capacidade.** O cálculo da capacidade para cada tanque preparatório/linha m em cada período t foi baseado nos trabalhos de Toledo e Armentano (2006) e Baldo et al. (2014). Inicialmente, estima-se em quanto tempo a linha m envasa determinada parte da demanda do período. A esse valor é somado o maior tempo de troca entre todos os itens multiplicado pelo número de itens, exceto o item fantasma, ou seja, trata-se de uma estimativa dos tempos de troca nessa linha. Soma-se ainda o tempo de uma limpeza temporal, que é o tempo da primeira limpeza do período. Obtém-se então:

$$Cap'_{mt} = \left(\sum_{j \in J} \frac{d_{jt} \rho_j}{\sum_{m \in M} S_m} \right) + (|J| - 1) * \max_{\forall i, j \in J \setminus i_0} \{TC_{ij}^{II}\} + LT^{II}.$$

Ao valor Cap'_{mt} ainda é preciso adicionar uma estimativa das possíveis limpezas temporais

que podem acontecer ao longo do período e assim, obtém-se o valor final da capacidade:

$$Cap_{mt} = Cap'_{mt} + \left\lfloor \frac{Cap'_{mt}}{TL_{max}} \right\rfloor LT^{II}.$$

Note que para o cálculo das capacidades disponíveis de produção usou-se somente os tempos de troca e de limpeza temporal referentes as linhas, pois conforme explicado no Capítulo 4, se as linhas respeitam a capacidade disponível (em tempo) os tanques preparatórios também respeitam, uma vez que eles estão dedicados e a produção termina na linha.

6.2 Critérios de avaliação dos resultados

Para avaliar os resultados obtidos com os experimentos computacionais, são utilizados dois critérios: o tempo computacional medido em segundos, e a qualidade da solução medida pelo valor da função objetivo e/ou *gap*. Nas estratégias heurísticas, o *gap* é calculado por:

$$gap = \frac{FO^{est} - Limitante_Inferior}{FO^{est}}, \quad (6.1)$$

em que FO^{est} é o valor da função objetivo da melhor solução factível da estratégia corrente e *Limitante_Inferior* é um limitante inferior para o problema que pode ser obtido com a resolução do modelo MDSL-2E-LT, ou com o valor da relaxação linear desse modelo.

Para fazer a comparação entre várias estratégias de solução, utiliza-se os perfis de desempenho de Dolan e Moré (2002). Esses perfis são baseados na função de distribuição de probabilidade cumulativa $P(f, q) : \mathbb{R} \rightarrow [0, 1]$ da estratégia de solução f . A função $P(f, q)$ é definida por:

$$P(f, q) = \frac{|\{p \in P : \log_2(v(p, f)) \leq q\}|}{n_P}, \quad q \geq 0, \quad (6.2)$$

em que P é o conjunto de instâncias, com $p = 1, \dots, |P|$, $n_P = |P|$ é o número total de instâncias, o conjunto das estratégias testadas é F , com $f = 1, \dots, |F|$, e $v(p, f)$ é a razão de melhoria, definida como a razão entre o valor da função objetivo (ou tempo, ou *gap*) obtido quando a instância p é resolvida pela estratégia f sobre a estratégia com o melhor desempenho. Seja $TC_{p,f}$ o valor da função objetivo (ou tempo, ou *gap*) da solução da instância p resolvida pela estratégia f , então:

$$v(p, f) = \frac{TC_{p,f}}{\min\{TC_{p,f} : f \in F\}}. \quad (6.3)$$

A função $P(f, q)$ indica a probabilidade da estratégia f estar dentro de um fator $q \in \mathbb{R}$ da melhor razão possível. Esta ferramenta mede, assim, a robustez da estratégia. As melhores estratégias são aquelas obtidas com o menor valor de q para $P(f, q) = 1$, ou seja, as estratégias em que o valor de $P(f, q)$ se aproxima mais rapidamente do valor máximo 1.

6.3 Testes computacionais: parte I

Nesta seção são realizados testes computacionais com o modelo MDSL-2E-LT e com os métodos de solução apresentados no Capítulo 5.4 e resumidos na Tabela 21. Nesses testes iniciais (parte I), são utilizadas somente as instâncias baseadas em dados reais descritas na Seção 6.1.1. O objetivo é determinar a melhor configuração de parâmetros do CPLEX para resolução do modelo MDSL-2E-LT, a melhor heurística construtiva *relax-and-fix*, a melhor heurística de decomposição do problema, a melhor heurística entre H-SUP e H-SUB, e as melhores heurísticas *fix-and-optimize* para essas instâncias. Os melhores métodos determinados nessa seção são utilizados na Seção 6.4 para resolver todas as instâncias (geradas aleatoriamente e reais), com o intuito de determinar o melhor método para resolver o problema de programação da produção de bebidas à base de frutas, e para analisar como esse método poderia ajudar na tomada de decisão nas empresas do setor.

É sabido que o tipo de instâncias têm influência nos resultados, entretanto, as instâncias aqui utilizadas refletem os dados das empresas visitadas durante esse trabalho. Logo, o intuito é encontrar efetivas soluções para instâncias representantes desse tipo de empresa.

Inicialmente, na Seção 6.3.1, são descritos os resultados obtidos com a calibração do CPLEX para o modelo MDSL-2E-LT.

6.3.1 Ajuste de parâmetros CPLEX no modelo MDSL-2E-LT

O modelo MDSL-2E-LT foi resolvido com todas as variações de parâmetros apresentadas na Tabela 50 do Apêndice E para todas as instâncias, a fim de avaliar a melhor combinação de parâmetros do CPLEX para esse modelo com essas instâncias.

Na Tabela 24 estão apresentadas as dimensões das instâncias baseadas em dados reais utilizadas no modelo MDSL-2E-LT (4.4)-(4.43): o número total de variáveis, o número de variáveis inteiras, binárias e o número de restrições. É possível perceber, dependendo da instância, que o número de variáveis e restrições cresce consideravelmente. A influência da dimensão do conjunto O_{mt} no tamanho das instâncias e no tempo computacional para resolvê-las é percebido comparando as instâncias S1 e S2, em que ao reduzir a cardinalidade de O_{mt} pela metade, o número de variáveis e restrições também é reduzido aproximadamente pela metade. Isto porque as variáveis X_{mjto} , Y_{mjto} , $\mu_{mjto}^{I,s}$, $\mu_{mjto}^{I,e}$, $\mu_{mjto}^{II,s}$, $\mu_{mjto}^{II,e}$, ψ_{mjtok}^I , W_{mjtok}^I , W_{mjtok}^{II} , ω_{mjtok}^I , ω_{mjtok}^{II} , e as restrições (4.6), (4.7), (4.8), (4.17), (4.18), (4.19), (4.20), (4.23), (4.24), (4.27), (4.28), (4.30), (4.31), (4.33), (4.34), (4.35), (4.36), (4.37), (4.38), (4.39), (4.40), (4.41), (4.42) dependem de O_{mt} . Nas equações (4.44), (4.45) e (4.46) é mostrada a parcela de $|O_{mt}|$ no número de variáveis e restrições.

Para nenhuma das 5 instâncias testadas (S1, M1, R1, R2, R3 e R4) a solução ótima foi encontrada e o critério de parada alcançado foi o tempo limite de 3.600 segundos.

Para a instância S2 algumas estratégias terminaram com o *gap* de 1% ou menos e o tempo computacional foi em média 1.274,31 segundos. Isso mostra a influência de $|O_{mt}|$ na dificuldade de resolução do problema. Sendo assim, a comparação entre as estratégias é realizada apenas com relação à função objetivo e *gap*. As Figuras 31a e 31b mostram os perfis de desempenho para os valores obtidos de *gap* e função objetivo, respectivamente, pelas 17 estratégias apresentadas no Apêndice E (Tabela 50). No gráfico de perfil de desempenho 31a, o valor do eixo $P(f, q)$ está variando de 0,95 a 1 para facilitar a visualização da ordem em que as estratégias atingem $P(f, q) = 1$. As Tabelas 54 e 55 no Apêndice G detalham os valores de função objetivo e *gap* encontrados pelas estratégias CPLEX e utilizados para fazer os gráficos da Figura 31.

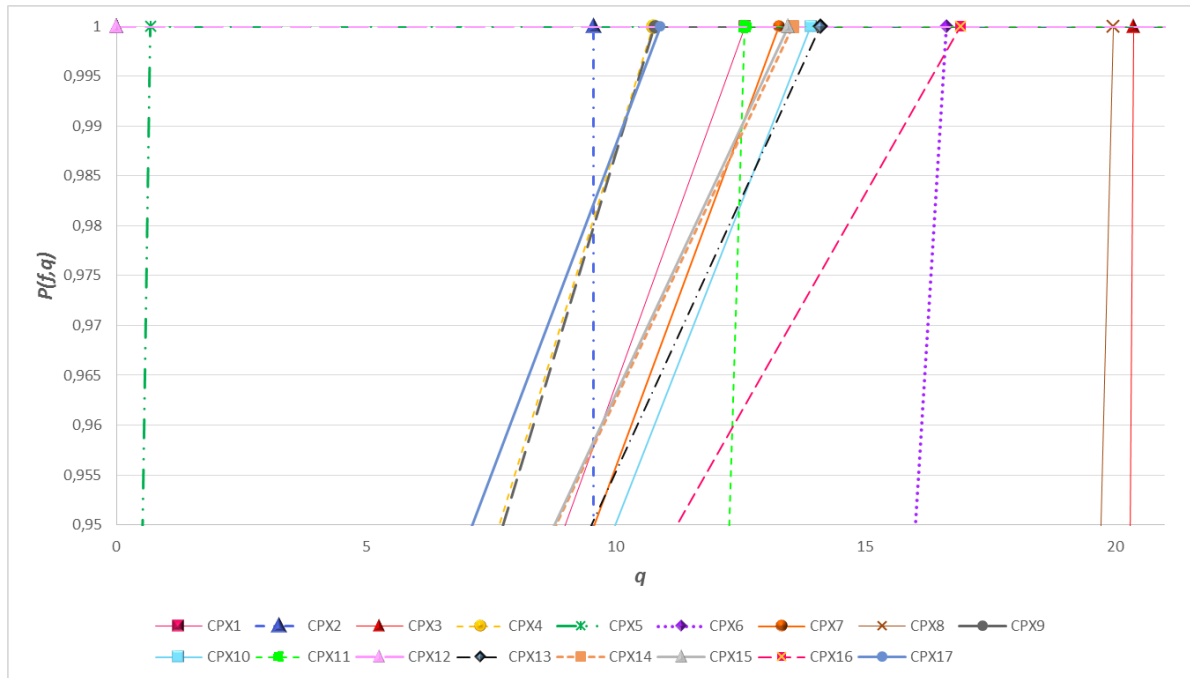
	Variáveis	Variáveis Inteiras	Variáveis Binárias	Restrições
S1	6.220	3.276	276	18.288
S2	3.076	1.668	60	7.032
M1	23.904	13.470	1.020	90.507
R1	63.104	32.680	2.504	398.424
R2	74.855	38.775	2.955	454.655
R3	59.884	31.020	2.364	363.724
R4	205.088	106.210	8.138	1.294.878

Tabela 24 – Dimensão das instâncias utilizadas no modelo MDSL-2E-LT.

Na Figura 31a é possível ver que a estratégia CPX12 foi a que teve o melhor desempenho no quesito valor função objetivo atingindo $P(f, q) = 1$ com $q = 0$; em segundo lugar ficou a estratégia CPX5 com $q = 0,68$ quando $P(f, q) = 1$. As piores estratégias foram CPX8 e CPX3 que atingiram $P(f, q) = 1$ com $q = 19,96$ e $q = 20,37$, respectivamente.

Com relação ao *gap*, as melhores estratégias foram a CPX7 e a CPX10, que atingiram $P(f, q) = 1$ com $q = 3,20$ e $q = 4,16$, respectivamente (Figura 31b). A pior foi a estratégia CPX13 que atingiu $P(f, q) = 1$ com $q = 9,37$. Como melhor estratégia foi então escolhida a CPX12, que tem o melhor valor de função objetivo. Nessa estratégia são alterados a seleção de nós, que passa a ser pelo melhor limitante, e as heurísticas RINS são aplicadas a cada 10 nós da árvore (Tabela 50). Durante a resolução dos problemas por meio da estratégia CPX12, foram explorados em média 31.312 nós. Como o número de nós explorados é alto, deixar a heurística RINS ligada a cada 10 nós é uma boa estratégia para obter boas soluções de função objetivo, como foi comprovado por CPX12 em que a RINS é aplicada a cada 10 nós.

Uma vez que a estratégia CPX12 é a que apresenta o melhor desempenho de função objetivo entre as estratégias CPLEX, na Tabela 25 as soluções obtidas com essa estratégia são comparadas com o CPLEX no modo *default*. Essa tabela apresenta os valores da função objetivo e do *gap* encontrados pelo CPLEX no modo *default* e com a estratégia



(a) Perfil de desempenho - função objetivo.

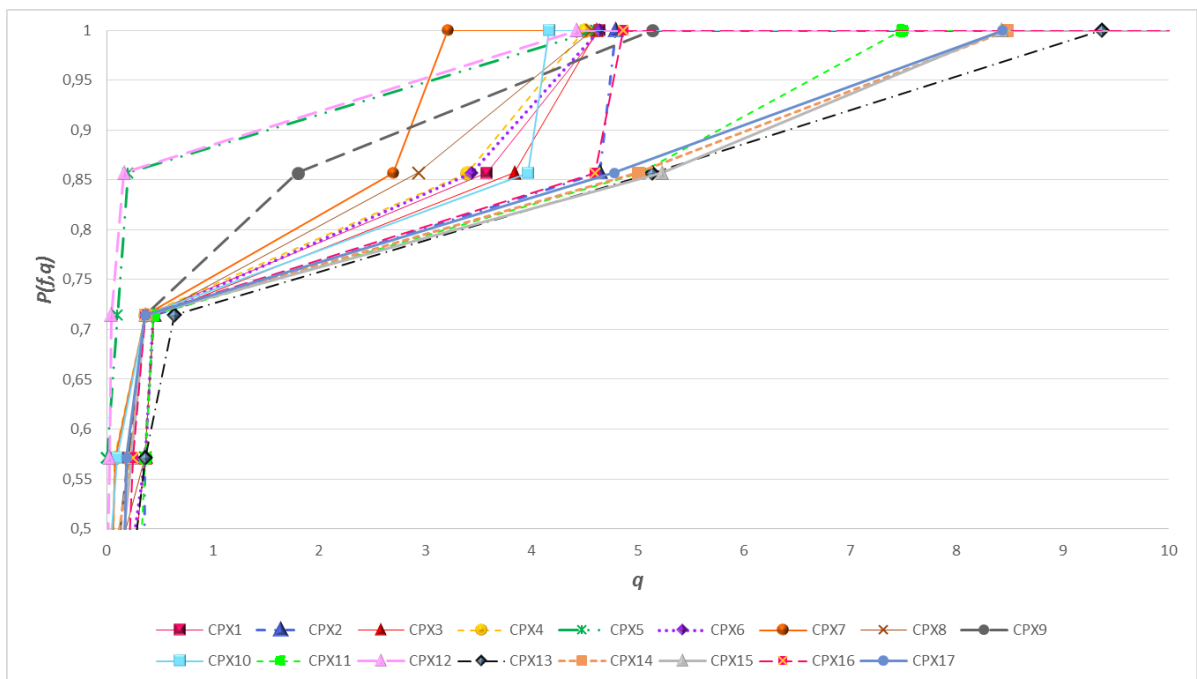
(b) Perfil de desempenho - *gap*.

Figura 31 – Perfis de desempenho das estratégias CPLEX. Fonte: a autora.

CPX12. Na média, a estratégia CPX12 apresenta função objetivo 6 vezes menor do que o *default* do CPLEX. No *gap* essa diferença não é tão grande. Entretanto, cabe observar que para a instância R1 a estratégia CPX12 encontra uma solução com *gap* 24,29% a menos do que no *default*, e para R2, com 20,66% a menos.

Os valores da função objetivo que aparecem na Tabela 25 são altos devido a ocorrência de atrasos e estoques que possuem custos unitários de 100 e 10, respectivamente.

Tabela 25 – Resultados com o CPLEX no modo *default* e com a estratégia CPX12.

Instância	<i>Default</i>		CPX12	
	Função objetivo (u.m.)	<i>gap</i> * (%)	Função objetivo (u.m.)	<i>gap</i> * (%)
S1	14,01	10,47	14,01	21,40
S2	14,01	0,01	14,01	0,93
M1	25,02	37,36	25,02	41,72
R1	25.266.143,00	100,00	41,18	75,71
R2	751.177,83	100,00	50,18	79,34
R3	12.199.625,33	100,00	1.237.545,21	100,00
R4	1.060.327.115,67	100,00	173.182.854,35	100,00
Média	156.934.873,55	63,98	24.917.220,43	59,79

* *gap* retornado pelo CPLEX.

Para os resultados que não atingem nem a casa das centenas, não houveram atrasos e estoques, somente custos de trocas e de limpezas temporais. A Tabela 26 apresenta o detalhamento dos lotes estocados e atrasados para as instâncias R1, R2, R3 e R4. Nela é possível observar a grande quantidade de atraso e estoque que acarretam os altos valores de função objetivo.

Tabela 26 – Detalhes dos resultados da Tabela 25 para as instâncias R1, R2, R3 e R4.

Inst.	<i>Default</i>				CPX12			
	Atraso (unid.)	Estoque (unid.)	Trocas (u.m.)	Limp. temp.	Atraso (unid.)	Estoque (unid.)	Trocas (u.m.)	Limp. temp.
R1	246.062	65.989	26	18	0	0	19	22
R2	0	75.111	28	87	0	0	26	24
R3	121.995	0	20	66	12.375	0	18	27
R4	10.406.077	10.000	69	54	171.999.292	1.183.458	62	42

Os *gaps* altos mostrados na Tabela 25 se dão devidos ao fato dos limitantes inferiores serem muito ruins para esse modelo. Por exemplo, a Figura 32 apresenta o gráfico de convergência dos limitantes (solução inteira e limitante inferior) durante a resolução do exemplar S2 pela estratégia CPX12. Para esse exemplar, a melhor solução encontrada tem valor $Z = 14,01$ com *gap* de 0,01%. Porém, pela Figura 32 percebe-se que o *branch-and-cut* encontrou a solução inteira de valor $Z = 14,01$ quando o limitante inferior (LI) estava com o valor $LI = 6,28$, ou seja, com o *gap* em 55,14%. Isso mostra a dificuldade do *solver* em encontrar bons limitantes inferiores. Por esse motivo, a estratégia CPX7 se saiu melhor com relação ao *gap*. Lembrando que essa estratégia enfatiza fortemente a prova da otimalidade pelo *branch-and-cut*.

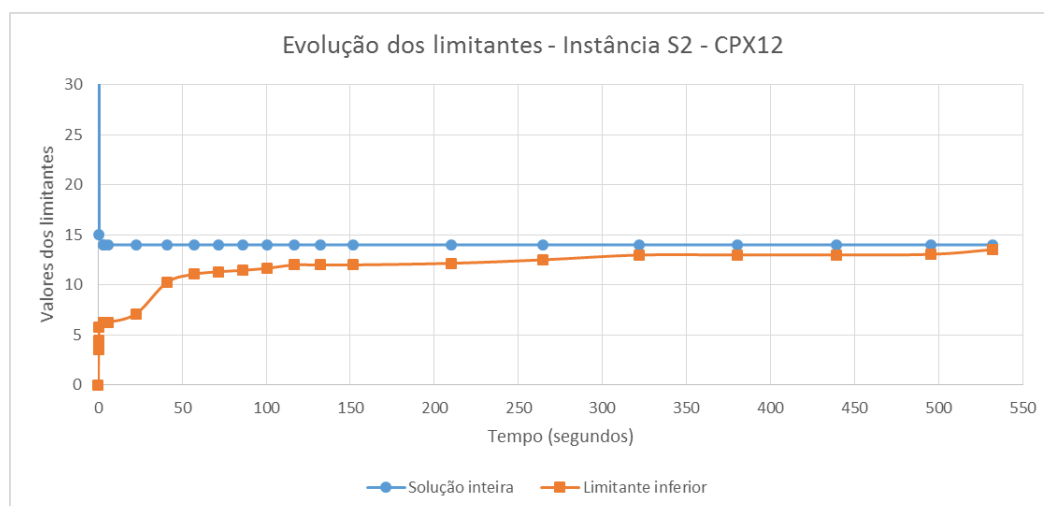


Figura 32 – Valores dos limitantes ao longo do tempo durante a resolução da instância S2 pela estratégia CPX12. Fonte: a autora.

Para as instâncias maiores, encontrar limitantes superiores (soluções inteiras) também pode ser uma tarefa difícil, como pode ser visto pela Figura 33 que representa a evolução dos limitantes ao longo do tempo na resolução da instância R1 pela estratégia CPX12. É possível perceber que a melhor solução foi encontrada depois de 1.500 segundos e que depois de 2.000 segundos não há mais nenhuma evolução relevante dos limitantes nos próximos 1.600 segundos.

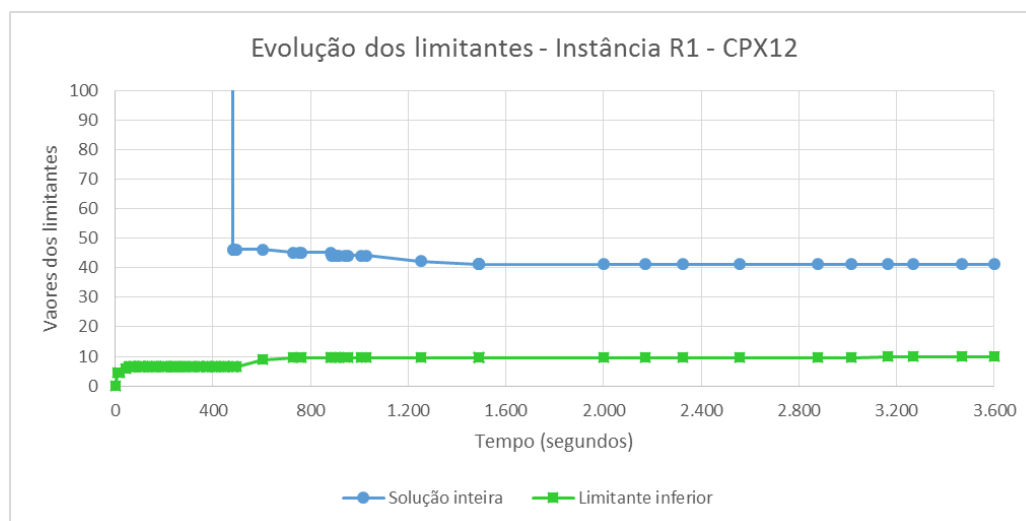


Figura 33 – Valores dos limitantes ao longo do tempo durante a resolução da instância R1 pela estratégia CPX12. Fonte: a autora.

Na Tabela 27 estão descritos alguns detalhes das soluções obtidas pela CPX12. O limitante inferior apresentado nessa tabela é usado para calcular o *gap* dos resultados obtidos pelos métodos de solução e apresentados nesse capítulo. Na quinta coluna está apresentado o valor obtido com a relaxação linear do modelo MDSL-2E-LT; nota-se que

a relaxação é muito fraca. Nessa tabela é apresentado também o tempo para obter a primeira solução factível. Observa-se que esse tempo é menos do que 1 segundo para todas as instâncias, pois uma solução trivial para esse modelo é atrasar toda a demanda, o que é computado rapidamente. Observando o tempo para encontrar a solução incumbente, percebe-se que além de ser difícil obter bons limitantes inferiores, para as instâncias maiores, bons limitantes superiores também exigem bastante tempo computacional, como já discutido anteriormente.

Tabela 27 – Detalhes das soluções obtidas pela estratégia CPX12.

Instância	Função objetivo (u.m.)	gap (%)	Limitante inferior (u.m.)	Relaxação linear (u.m.)	Tempo para 1ª solução factível (seg.)	Tempo para solução incumbente (seg.)	Tempo total (seg.)
S1	14,01	21,40	11,00	1,40	0,00	7,00	3.600,00
S2	14,01	0,93	13,87	2,79	0,00	2,94	531,89
M1	25,02	41,72	14,57	1,74	0,00	171,45	3.600,00
R1	41,18	75,71	9,96	4,78	0,02	1998,94	3.600,00
R2	50,18	79,34	10,33	6,14	0,03	3577,01	3.600,00
R3	1.237.545,21	100,00	8,62	5,25	0,02	2857,88	3.600,00
R4	173.182.854,35	100,00	22,56	16,92	0,05	3595,23	3.600,00

Todos os subproblemas MIP derivados do MDSL-2E-LT e utilizados pelas estratégias *relax-and-fix* e *fix-and-optimize* foram resolvidos utilizando a melhor configuração selecionada nestes testes pelo critério função objetivo que é a CPX12.

6.3.2 Avaliação das heurísticas de decomposição

Nesta seção são analisados os resultados obtidos para as instâncias baseadas em dados reais com as heurísticas de decomposição do modelo em dois estágios: HE1A-DS, HE1B-DS, HE2A-DS E HE2B-DS apresentadas na Seção 5.1. Os parâmetros que são definidos *a priori* nas heurísticas de decomposição foram configurados empiricamente para os seguintes valores: $\beta = 0,25$, $\varepsilon = 0,025$. Cap_{mt} , $\forall m \in M$, $\forall t \in T$, $\eta_1 = 0,50$, $\eta_2 = 1$ e $\varphi = 0,90$. O tempo limite para resolução dos modelos em cada iteração é de 60 segundos.

A Tabela 28 mostra as soluções obtidas com as heurísticas do estágio I HE1-DS e a Tabela 29, os resultados da heurística HE2-DS. O *gap* é calculado em relação ao melhor limitante encontrado com o modelo MDSL-2E-LT e descrito na Tabela 27, de acordo com a equação (6.1).

Na Tabela 28, as heurísticas HE1A-DS e HE1B-DS apresentam valores de função objetivo idênticos para as instâncias S1 e M1. Para a instância R1, a diferença entre as soluções é de apenas uma unidade. Para a instância R3, HE1B-DS apresenta um resultado quase 3 vezes melhor do que a HE1A-DS, porém para essa instância R3 a HE1A-DS se sai um pouco melhor. Para a instância R4, a função objetivo obtida pela HE1B-DS é quase 40 vezes menor do que a função objetivo apresentada pela HE1A-DS. Uma explicação para

Tabela 28 – Resultados obtidos pelas heurísticas HE1-DS.

HE1-DS						
Instância	HE1A-DS			HE1B-DS		
	Função objetivo (u.m.)	gap (%)	Tempo (seg.)	Função objetivo (u.m.)	gap (%)	Tempo (seg.)
S1	24,00	42,21	1,73	24,00	42,21	1,80
M1	31,00	53,00	0,84	31,00	53,00	5,16
R1	48,00	79,25	7,28	47,00	78,80	15,59
R2	612.671,33	100,00	26,91	195.962,00	99,99	61,88
R3	4.928.219,67	100,00	55,36	5.094.801,00	100,00	78,20
R4	159.816.045,50	100,00	1.226,62	4.068.931,00	100,00	1.216,05
Média	27.559.506,58	77,85	219,79	1.559.966,00	77,74	229,78

isso é que como essa instância é maior do que as outras e usa mais tempo, foram realizadas reduções mais bruscas em todos os períodos, enquanto que para a HE1B-DS, apesar da redução ser brusca, é realizada apenas para alguns períodos. A heurística HE1A-DS é 10 segundos mais rápida, em média, do que a HE1B-DS.

Tabela 29 – Resultados obtidos pelas heurísticas HE2-DS.

HE2-DS						
Instância	HE2A-DS			HE2B-DS		
	Função objetivo (u.m.)	gap (%)	Tempo (seg.)	Função objetivo (u.m.)	gap (%)	Tempo (seg.)
S1	23,00	39,70	0,56	23,00	39,70	249,02
M1	30,00	53,77	0,12	30,00	53,77	0,14
R1	53,00	72,51	1,11	52,00	71,98	1,81
R2	599.916,77	100,00	6,12	479.664,11	100,00	18,33
R3	26.142.122,66	100,00	35,33	26.944.737,87	100,00	29,34
R4	3.069.057,23	100,00	140,50	2.787.596,60	100,00	270,02
Média	4.968.533,78	77,66	30,62	5.035.350,60	77,57	94,78

Conforme apresentado na Tabela 29, os resultados obtidos com HE2B-DS são melhores dos que os encontrados com HE2A-DS para a maioria das instâncias (R1, R2 e R4). Quando comparadas em termos do tempo computacional, as heurísticas HE2-DS são mais rápidas do que as heurísticas HE1-DS. Enquanto que a HE1A-DS demora em média 219,79 segundos para ser executada, a HE2A-DS leva apenas 30,62 segundos em média; a HE1B-DS também demora em média 229,78 segundos contra 94,78 segundos da HE2B-DS.

Uma vez que HE1B-DS e HE2B-DS têm desempenhos melhores do que HE1A-DS e HE2A-DS, respectivamente, essas duas heurísticas são comparadas. A HE1B-DS tem melhores valores de função objetivo para R1, R2 e R3, enquanto que HE2B-DS apresenta melhores valores para S1, M1 e R4. Entretanto, o valor obtido pela HE2B-DS para a instância R3 é muito alto, enquanto que para a S1 e M1 a diferença é só de uma unidade. Portanto, define-se que a heurística que tem o melhor desempenho entre as construtivas é a HE1B-DS. Comparando essa estratégia HE1B-DS com a CPX12, percebe-se que houve melhoria para a maior instância, R4, e que os tempos de solução são menores.

6.3.3 Avaliação das heurísticas baseadas nos modelos relaxados

A fim de avaliar o desempenho das heurísticas Subestimada (H-SUB) e Superestimada (H-SUP), definidas na Seção 5.2, a Tabela 30 apresenta para ambas os valores da função objetivo e *gap* com relação ao melhor limitante encontrado para o modelo MDSL-2E-LT e tempo computacional.

Tabela 30 – Resultado das heurísticas baseadas em modelo: Subestimada (H-SUB) e Superestimada (H-SUP)

Instância	H-SUB (Modelo MR1-O)			H-SUP (Modelo MR2-P)		
	Função objetivo (u.m.)	<i>gap</i> (%)	Tempo (seg.)	Função objetivo (u.m.)	<i>gap</i> (%)	Tempo (seg.)
S1	15,00	7,53	71,27	15,00	7,53	61,11
S2	15,00	7,53	21,47	15,00	7,53	22,06
M1	28,00	47,96	600,20	29,00	49,76	600,17
R1	47,00	78,81	1.800,92	47,00	78,81	600,42
R2	805.571,58	100,00	1.803,67	20.680,00	99,95	600,50
R3	8.788.905,17	100,00	1.802,51	2.154.214,67	100,00	600,50
R4	8.457.036,33	100,00	1.802,92	2.979.833,67	100,00	601,17
Média	2.578.802,58	63,12	1.128,99	736.404,91	63,37	440,85

A heurística H-SUP obteve os melhores resultados em menor tempo. Para essas heurísticas, o tempo máximo disponibilizado para resolução dos modelos MR1-O (5.31);(4.5)-(4.24);(5.32)-(5.36) e MR2-P ((5.31);(4.5)-(4.24);(5.37)-(5.40);(5.36)) foi de 600 segundos em cada iteração. Assim, percebe-se que para as instâncias S1 e S2, o modelo foi resolvido até a otimalidade. Para a instância M1, as duas heurísticas conseguiram uma solução factível logo na primeira iteração. Porém, para as instâncias de R1 até R4, somente a heurística H-SUP obteve solução sem precisar reduzir a capacidade do modelo, ou seja, na primeira iteração. Por esse motivo, as soluções de H-SUP são melhores e obtidas mais rapidamente do que de H-SUB.

Observa-se que os resultados obtidos pela heurística H-SUP são ainda melhores do que os resultados apresentados pela heurística HE1B-DS, porém o tempo despendido pela H-SUP é quase 3 vezes maior do que pela HE1B-DS. De fato, o modelo MR2-P está mais próximo da representação do problema integrado em dois estágios do que o modelo MDSL-P (5.1)-(5.12), que representa apenas o primeiro estágio e é utilizado na HE1B-DS. Isso faz com que menos iterações de redução de capacidade sejam realizadas na heurística H-SUP, entretanto o modelo MR2-P é computacionalmente mais caro de ser resolvido.

Quando compara-se a H-SUP com o MDSL-2E-LT resolvido pela estratégia CPX12, nota-se que os valores de função objetivo são piores. Mas de certa forma, para as instâncias S1, S2, M1 e R1 não são tão ruins, dado que para encontrar esses valores foi gasto em torno de 1/6 do tempo despendido pela CPX12. Para a maior instância, R4, a H-SUP apresenta uma solução melhor do que CPX12.

6.3.4 Avaliação dos métodos H1EB-DS+CPX12 e H-SUP+CPX12

Nesta seção analisa-se o comportamento da resolução do modelo MDSL-2E-LT pela estratégia CPX12 com solução inicial fornecida pelas heurísticas H1EB-DS e H-SUP. Esses métodos são nomeados de H1EB-DS+CPX12 e H-SUP+CPX12. Aliás, daqui para frente ao longo do texto será utilizada a notação “SI+M” para indicar que “SI” (solução inicial) é o método utilizado para obter uma solução inicial para ser fornecida para o método de solução “M”. O objetivo desse teste é avaliar se fornecer uma solução inicial com uma qualidade razoável para o CPLEX pode ajudar na busca de boas soluções para o modelo MDSL-2E-LT.

As heurísticas H1EB-DS e H-SUP foram resolvidas com tempo limite de 20 minutos (1/3 do tempo total de 3.600 segundos) e a solução do modelo pelo CPLEX foi limitada por 1 hora menos o tempo despendido pelas heurísticas. Os valores de função objetivo e os tempos despendidos pelas heurísticas estão mostrados na Tabela 31. Observe que as heurísticas retornam uma solução factível em no máximo 435,58 segundos (instância R4).

Os resultados obtidos pelas estratégias H1EB-DS+CPX12 e H-SUP+CPX12 estão apresentados na Tabela 32. As duas estratégias em todas as instâncias atingiram o tempo limite de 3.600 segundos, exceto a instância S2 resolvida pelo método H1EB-DS+CPX12, que encontrou a solução ótima em 289,18 e com H-SUP+CPX12 em 504,61, ambas em tempos menores do que o CPX12 que demorou 531,89. Ambos os métodos H1EB-DS+CPX12 e H-SUP+CPX12 melhoram significativamente as soluções iniciais em média em 55,62% e 37,27%, respectivamente, como é possível ver na Tabela 32.

Tabela 31 – Soluções obtidas com a H1EB-DS e com a H-SUP com tempo limite de 20 minutos.

Instância	H1EB-DS		H-SUP	
	Função objetivo (u.m.)	Tempo (seg.)	Função objetivo (u.m.)	Tempo (seg.)
S1	24,00	1,64	15,00	61,06
S2	24,00	1,64	15,00	21,92
M1	31,00	4,97	28,00	300,17
R1	47,00	17,55	47,00	300,44
R2	195.962,00	49,25	20.678,00	300,47
R3	5.094.801,00	15,30	2.154.216,67	300,36
R4	395.532.538,67	435,58	6.845.494,33	301,14
Média	57.260.489,67	75,13	1.288.642,00	226,51

Quando comparados os resultados com CPX12, ambos H-SUP+CPX12 e H1EB-DS+CPX12 melhoram o resultado da instância R4 de maneira significativa. Para a instância R3, o método H-SUP+CPX12 encontra uma solução com valor de função objetivo aproximadamente 2 vezes menor. Entretanto, para as outras instâncias a melhoria com relação a CPX12 não é muito significativa. Com relação aos *gaps*, na média, H-SUP+CPX12 apresenta um *gap* pior do que a estratégia CPX12. Sendo assim, conclui-se

Tabela 32 – Resultados do modelo MDSL-2E-LT via CPX12 com solução inicial dada pelas heurísticas H1EB-DS e H-SUP.

Instância	H1EB-DS+CPX12			H-SUP+CPX12			MDSL-2E-LT	
	Função Objetivo (u.m.)	gap (%)	Melhoria da sol. inicial (%)	Função Objetivo (u.m.)	gap (%)	Melhoria da sol. inicial (%)	Função Objetivo (u.m.)	gap (%)
S1	14,02	20,53	41,58	14,02	46,62	6,53	14,01	21,40
S2	14,01	0,03	41,63	14,01	0,96	6,60	14,01	0,93
M1	25,07	40,01	19,13	25,07	49,15	10,46	25,02	41,72
R1	40,67	70,90	13,47	42,57	71,03	9,43	41,18	75,71
R2	49,71	72,09	99,97	51,79	76,57	99,75	50,18	79,34
R3	1.237.545,78	100,00	75,71	612.546,91	100,00	71,57	1.237.545,21	100,00
R4	8.457.036,33	100,00	97,86	2.972.833,67	100,00	56,57	173.182.854,35	100,00
Média	1,384,960.80	57,65	55,62	512,218.29	63,48	37,27	24,917,220.57	59,87

* *gap* retornado pelo CPLEX.

que fornecer uma solução inicial para o CPLEX não altera seu desempenho, exceto para instâncias grandes em que a solução inicial do CPLEX é geralmente a trivial (toda a demanda é atrasada), como no caso da instância R4.

6.3.5 Avaliação das heurísticas *relax-and-fix*

Nesta seção são avaliadas as heurísticas *relax-and-fix* resumidas na Tabela 19 da Seção 5.3. A Figura 34 mostra o perfil de desempenho do valor da função objetivo das heurísticas *relax-and-fix* (RFX1, RFX2, ..., RFX16) testadas com o modelo MDSL-2E-LT. Os valores das funções objetivos obtidas por cada uma das heurísticas estão apresentados na Tabela 56 no Apêndice G.

As estratégias RFX4 e RFX16 ficaram praticamente empatadas em primeiro lugar, pois ambas retornaram os menores valores de função objetivo para 57,14% das instâncias ($P(f, q) = 0, 57$ quando $q = 0$). Porém, a RFX4 ainda se saiu um pouco melhor, pois atingiu $P(f, q) = 1$ com $q = 0, 97$ enquanto que a RFX16 atingiu $P(f, q) = 1$ com $q = 1, 01$. Isso significa que para as demais instâncias (42,86%), as soluções da RFX4 estão dentro de um fator $2^{0,96} \approx 1, 94$ vezes o menor valor de função objetivo obtido dentre todas e para a RFX16, essas soluções estão dentro de um fator $2^{1,01} \approx 2, 02$ vezes o melhor valor de função objetivo. Ou seja, os valores das funções objetivos obtidas pela RFX4 estão mais próximos dos melhores valores quando comparadas as soluções obtidas pela RFX16. Nas heurísticas RFX4, são fixadas as variáveis binárias que determinam se existe produção (Y_{mjto}), a sequência de produção (Z_{mijt}) e a existência das limpezas temporais (W_{mjtok}^I e W_{mjtok}^{II}). Na RFX16, as variáveis fixadas são as mesmas da RFX4 e também as variáveis ω_{mjtok}^I e ω_{mjtok}^{II} , porém essas variáveis são fixadas somente quando existe produção no lote o , ou seja, somente se $Y_{mjto} > 0$. Em segundo lugar ficaram também empatadas as estratégias RFX13 e RFX15.

Os tempos computacionais e *gap* estão discriminados, respectivamente, nas Tabelas

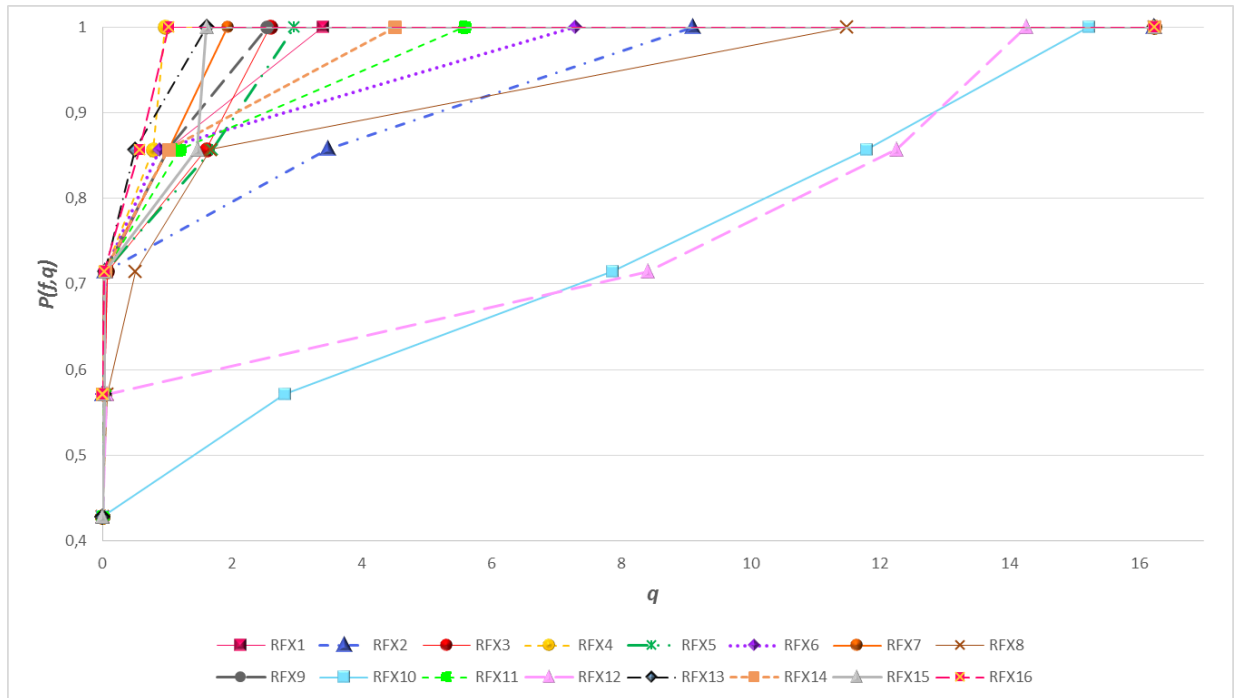


Figura 34 – Perfil de desempenho da função objetivo das heurísticas *relax-and-fix*. Fonte: a autora.

57 e 58 no Apêndice G. Os critérios tempos computacionais e *gap* não foram avaliados com perfil de desempenho, pois apresentam soluções muito parecidas. Além disso, neste trabalho o maior interesse é avaliar os valores de função objetivo, ou seja, deseja-se determinar o método que apresenta o melhor valor de função objetivo.

Para as instâncias S1, S2, M1, R3 e R4, os *gaps* apresentados por todas as heurísticas *relax-and-fix* foram iguais. Para a instância R1 o menor *gap* foi de 74,46% retornado pelas heurísticas RFX7, RFX9 e RFX16. Para a instância R2 o menor *gap* foi de 78,48% para as estratégias RFX13, RFX16. O menor tempo computacional foi, em média, 1.980,33 segundos, apresentado pela RFX12, e o pior foi de 2.770,11 segundos, em média, dado pelo RFX1. A RFX4 e a RFX16 obtiveram os melhores desempenhos com relação à função objetivo, com tempos médios de 2.532,32 e 2.355,61 segundos, respectivamente. Para as instâncias menores S1 e S2, a média de tempo de execução de todas as estratégias *relax-and-fix* foi de 501,16 e 588,32, respectivamente. Note que nesse caso o valor de $|O_{mt}|$ não fez tanta diferença nos tempos de solução.

A Tabela 33 apresenta uma comparação entre a melhor heurística RFX4 e a melhor estratégia CPLEX, CPX12. Percebe-se que houve uma melhoria da RFX4 com relação a CPX12 para as instâncias de R1 a R4, sendo que o maior ganho foi para a instância R4. Além disso, houve uma melhoria nos tempos computacionais para as instâncias S1 e M1, sendo que a RFX4 obteve a mesma solução que a CPX12 para a instância S1 com 3201,53 segundos a menos, e para a M1 com 600,3 segundos a menos.

Tabela 33 – Resultados da heurística *relax-and-fix* RFX4 e estratégia CPX12.

Inst.	RFX4			CPX12		
	Função objetivo (u.m.)	gap (%)	Tempo (seg.)	Função objetivo (u.m.)	gap (%)	Tempo (seg.)
S1	14,01	0,93	398,47	14,01	0,93	3.600,00
S2	14,01	0,93	459,05	14,01	0,93	531,89
M1	25,02	41,72	2.991,70	25,02	41,72	3.600,00
R1	40,20	75,10	3.600,00	41,18	75,71	3.600,00
R2	49,18	78,92	3.600,00	50,18	79,34	3.600,00
R3	1.196.377,33	100,00	3.600,00	1.237.545,21	100,00	3.600,00
R4	11.001.671,00	100,00	3.600,00	173.182.854,35	100,00	3.600,00

6.3.6 Avaliação das heurísticas *fix-and-optimize*

Nesta seção são analisadas as heurísticas *fix-and-optimize* descritas na Seção 5.4.1 do Capítulo 5.4. Para cada uma delas foram utilizados três soluções iniciais diferentes:

- 1) a melhor estratégia CPLEX (CPX12), gerando a heurística denotada por CPX12+FXO;
- 2) a melhor heurística *relax-and-fix* (RFX4), determinando a heurística RFX4+FXO;
- 3) e, por fim, o método heurístico que apresenta as melhores soluções em menos tempo, a heurística HE1B-DS, apresentando a heurística HE2B-DS+FXO.

O objetivo de testar três soluções iniciais diferentes é avaliar a interferência da solução inicial nas heurísticas *fix-and-optimize*. Para resolver as estratégias que determinam as soluções iniciais, o tempo disponível é de no máximo 1/3 do tempo total determinado para resolução da *fix-and-optimize*. Uma vez que o tempo máximo disponível para cada instância é de 3.600 segundos, para os testes realizados nessa seção foi estabelecido um tempo máximo de 1.200 segundos para obtenção de uma solução inicial. Depois de percorrer todas as partições por pelo menos uma vez, um outro critério de parada é a passagem por uma partição sem melhoria. Ou seja, se ao otimizar as variáveis de uma partição já visitada a solução encontrada não for melhor do que a solução incumbente, o método é encerrado. O tempo máximo disponível para a otimização do modelo em cada partição foi definido como o tempo total (3.600 segundos) menos o tempo utilizado para encontrar a primeira solução, dividido pelo número de partições (número de períodos, por exemplo) e multiplicado por 2. Caso esse tempo não seja todo utilizado para otimizar o modelo, o tempo que não foi utilizado é somado ao tempo disponível na próxima partição.

A Figura 35 mostra o perfil de desempenho para o valor da função objetivo das heurísticas CPX12+FXO. As melhores heurísticas foram CPX12+FXO12, CPX12+FXO4 e CPX12+FXO9, que mostraram o melhor perfil convergindo para $P(f, q) = 1$ em $q = 1, 38$, $q = 1, 72$ e $q = 2, 29$, respectivamente. A CPX12+FXO retornou o menor valor para 42,86%

das instâncias ($P(f, q) = 0,4286$ com $q = 0$), e para as outras 57,14% das instâncias as soluções da CPX12+FXO12 estão dentro de um fator $2^{1,38} \approx 2,6$ vezes o menor valor da função objetivo obtido dentre todas. O pior desempenho foi da heurística CPX12+FXO7, que atingiu $P(f, q) = 1$ com $q = 15,02$. Nessa heurística as variáveis fixadas são as binárias Y_{mjto} e Z_{mijt} e as de produção X_{mjto} . Os valores das funções objetivos obtidos pelas 12 estratégias (CPX12+FXO1 a CPX12+FXO12) estão detalhados na Tabela 59 do Apêndice G.

Os valores de *gap* e tempo computacional das heurísticas CPX12+FXO estão no Apêndice G, nas Tabelas 60 e 61, respectivamente. Na média, os menores valores de *gap* são de 56,92% e 56,97% obtidos, respectivamente, pelas heurísticas CPX12+FXO9 e CPX12+FXO4. O pior valor de *gap*, na média, é 63,42%, dado pelas estratégias CPX12+FXO7. Com relação ao tempo computacional, CPX+FXO7 obteve o menor tempo médio, 1.880,72 segundos, e CPX+FXO12 o pior com valor de 2.556,00 segundos. O perfil de desempenho para a função objetivo das heurísticas RFX4+FXO é mostrado na Figura 36. Neste caso, a heurística que teve o melhor desempenho foi a RFX4+FXO3, cuja FXO3 fixa as mesmas variáveis que a RFX4 que fornece a solução inicial.

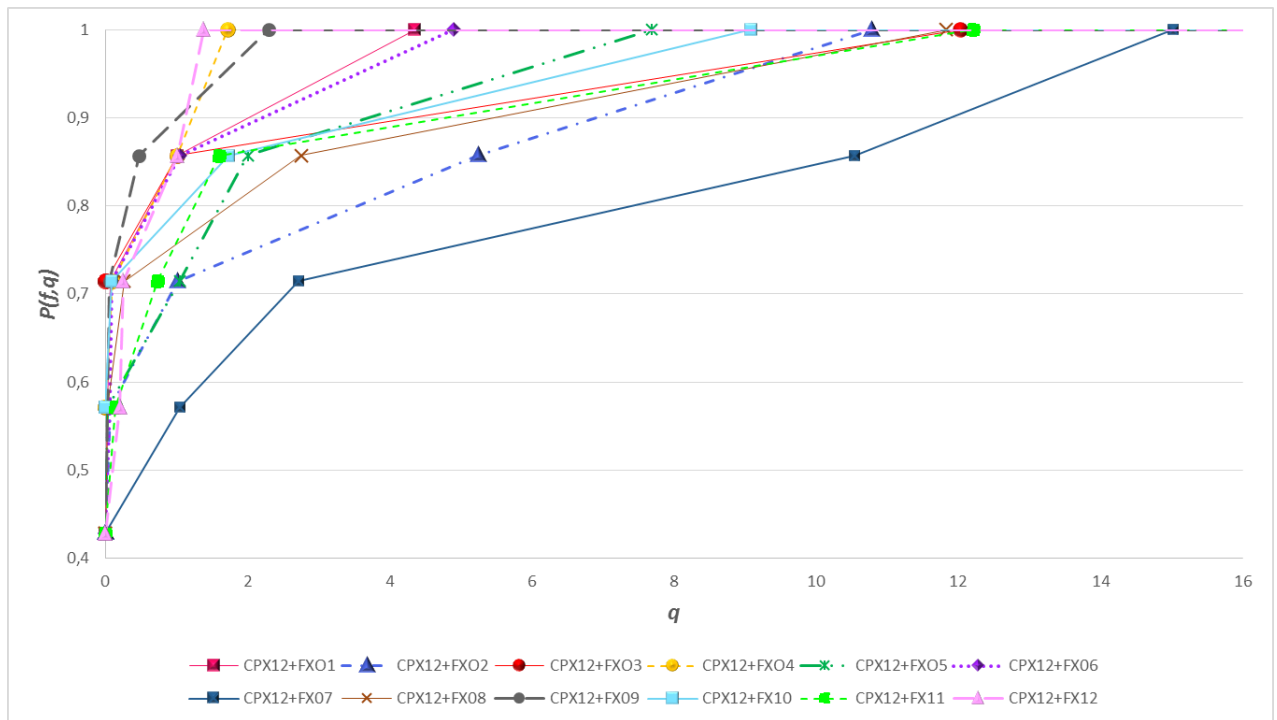


Figura 35 – Perfil de desempenho da função objetivo nas heurísticas *fix-and-optimize* com solução inicial da CPX12. Fonte: a autora.

A heurística RFX4+FXO3 obteve a melhor solução para 85,71% das instâncias, e para as demais instâncias apresentou soluções que estão dentro de um fator $2^{0,07} \approx 1,05$ vezes do melhor valor de função objetivo obtido. Ou seja, para as demais instâncias o valor da função objetivo pode ser até 1,05 vezes pior do que o melhor valor de função

objetivo dentre todos. Houve praticamente um empate no segundo lugar entre as estratégias RFX4+FXO10 e RFX4+FXO10, sendo que a RFX4+FXO4 é ligeiramente melhor, pois alcança $P(f, q) = 1$ com $q = 1,06$, enquanto que para a RFX4+FXO10 $q = 1,10$ quando $P(f, q) = 1$. A RFX4+FXO4 fixa as mesmas variáveis que a RFX4+FXO3 (Y_{mjto} , Z_{mijt} , W_{mjtok}^I e W_{mjtok}^{II}), porém percorre os períodos do último para o primeiro (*backward*). A RFX4+FXO10 também percorre os períodos *backward*, e fixa as mesmas variáveis que a RFX4+FXO4, mas fixa também ω_{mjtok}^I e ω_{mjtok}^{II} . Em terceiro lugar também estão praticamente empatadas as heurísticas RFX4+FXO2, RFX4+FXO5, RFX4+FXO9 e RFX4+FXO12. O pior desempenho foi da heurística RFX4+FXO8 que fixa as variáveis de produção X_{mjto} .

A RFX4+FXO2 foi a que apresentou o maior tempo computacional médio, 2.468,95 segundos. O menor tempo computacional médio foi apresentado pela estratégia RFX4+FXO6, 1.919,65 segundos. A estratégia RFX4+FXO3 gastou em média 2.280,80 segundos para apresentar uma solução. O melhor *gap* médio é 56,80% e foi obtido por RFX4+FXO1, RFX4+FXO10 e RFX4+FXO11, enquanto o pior *gap* médio foi da RFX4+FXO8, 59,82%. Os dados de *gap* e tempo computacional para as estratégias RFX4+FXO estão apresentados nas Tabelas 63 e 64, respectivamente, no Apêndice G.

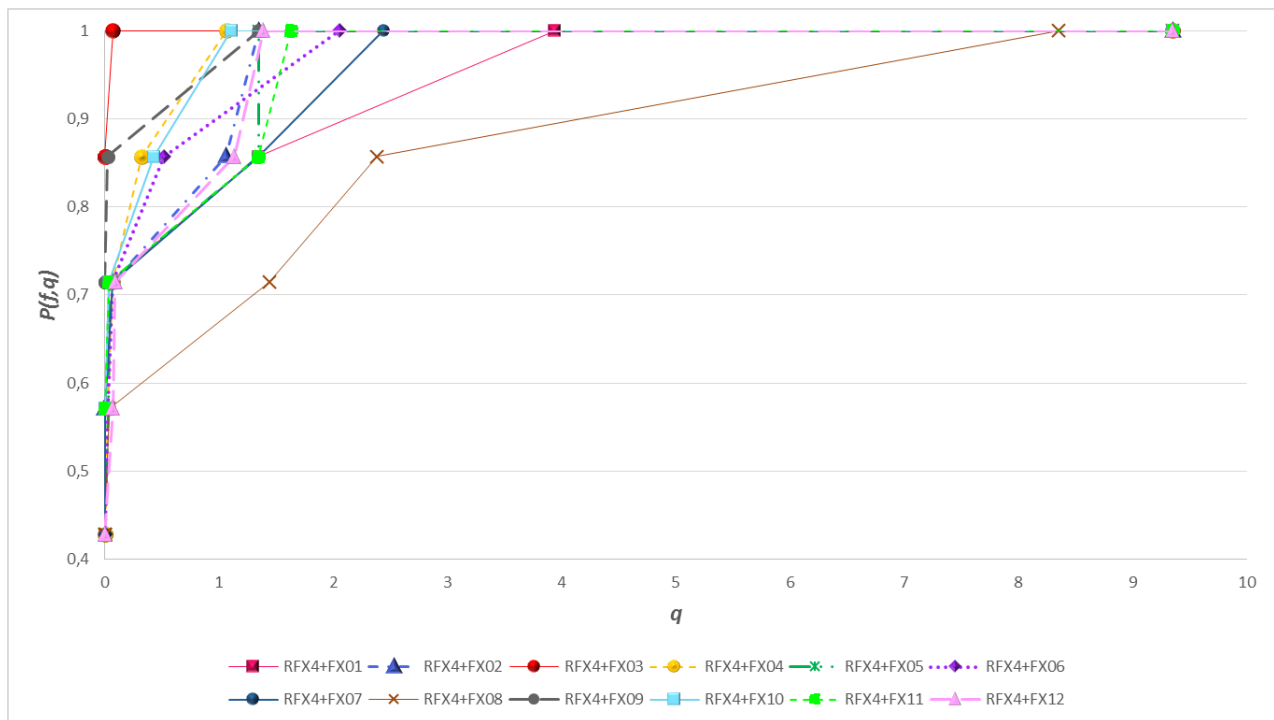


Figura 36 – Perfil de desempenho da função objetivo nas heurísticas *fix-and-optimize* com solução inicial da RFX4. Fonte: a autora.

Para as heurísticas *fix-and-optimize* com solução inicial da heurística construtiva HE1B-DS, o gráfico de perfil de desempenho da função objetivo é apresentado na Figura 37. Os perfis de desempenho são apresentados para o eixo $P(f, q)$ variando de 0,9 a 1,

para facilitar a visualização da ordem em que as estratégias atingem $P(f, q) = 1$. Note na Figura 37 que as estratégias HE1B-DS+FXO5, HE1B-DS+FXO11 e HE1B-DS+FXO12 estão praticamente empatadas como as melhores estratégias. Porém a HE1B-DS+FXO3 é ligeiramente melhor do que as outras duas, pois esta atinge $P(f, q) = 1$ com $q = 0,33$, enquanto HE1B-DS+FXO11 atinge com $q = 0,38$ e HE1B-DS+FXO12 com $q = 0,40$. A pior estratégia foi a HE1B-DS+FXO8, seguida de HE1B-DS+FXO1 e HE1B-DS+FXO2. Todas as outras estratégias tiveram desempenho similar.

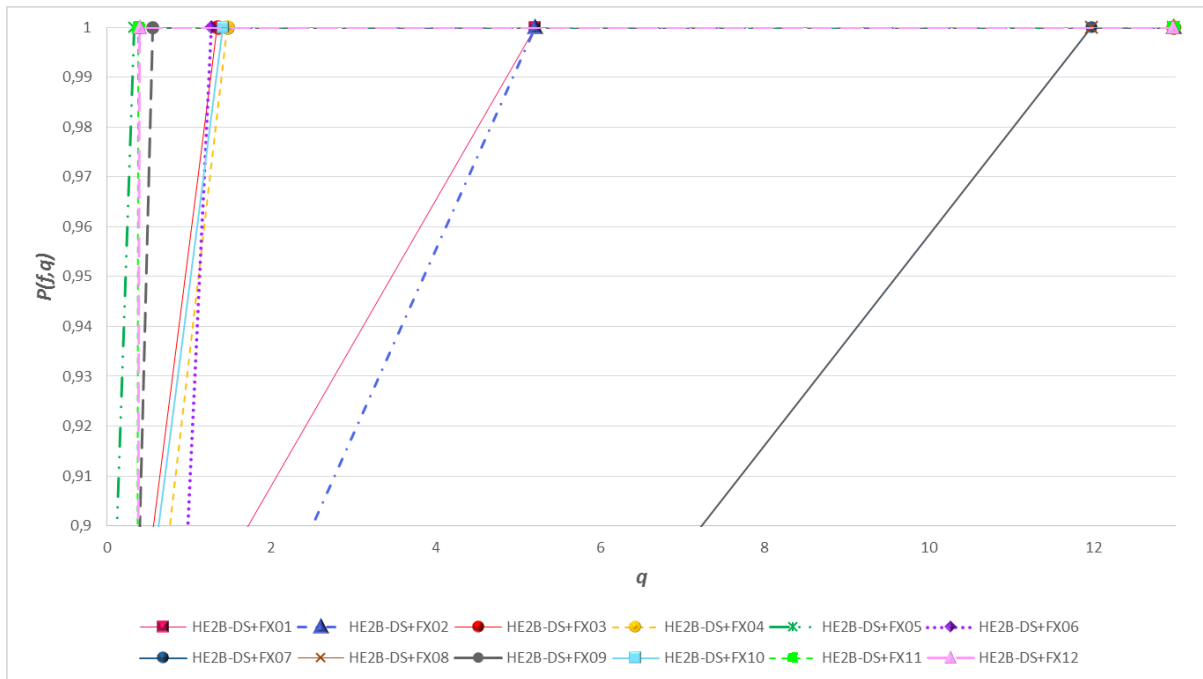


Figura 37 – Perfil de desempenho da função objetivo nas heurísticas *fix-and-optimize* com solução inicial da HE1B-DS. Fonte: a autora.

Os valores de função objetivo obtidos por todas as estratégias HE1B-DS+FXO para todas as instâncias estão detalhados na Tabela 65 no Apêndice G. Os valores de *gap* e tempo computacional obtidos pelas heurísticas do tipo HE1B-DS+FXO para todas as instâncias são detalhadas nas Tabelas 66 e 67, respectivamente, no Apêndice G. O menor tempo computacional médio foi obtido com a estratégia HE1B-DS+FXO8, 1.328,49 segundos, porém com pior valor de função objetivo. O pior tempo computacional médio foi da heurística HE1B-DS+FXO5, 2.039,15 segundos, que foi a estratégia com a melhor função objetivo. É interessante notar que, para a instância R4, todo o tempo de 3.600 segundo foi utilizado, indicando que se houvesse mais tempo a função objetivo poderia ter sido ainda mais melhorada.

Em todas as estratégias *fix-and-optimize*, as melhores estratégias fixam as variáveis Y_{mjto} e Z_{mijt} , e se diferenciam em fixar ou não as variáveis W_{mjtok}^I , W_{mjtok}^{II} , ω_{mjtok}^I e ω_{mjtok}^I . Fixar a variável de produção X_{mjto} sempre é uma das piores estratégias.

Na Seção 6.4 as heurísticas *fix-and-optimize* que tiveram o melhor desempenho para as instâncias baseadas nos dados reais (CPX12+FXO12, RFX4+FXO3 e HE1B-DS+FXO3) serão avaliadas com as instâncias geradas aleatoriamente.

6.3.7 Avaliação das heurísticas *fix-and-optimize* com busca em vizinhança

Os resultados dos testes computacionais com as heurísticas *fix-and-optimize* com busca em vizinhança para as instâncias baseadas em dados reais são analisados nessa seção. Entre as estratégias FXOV1_T, FXOV1_T-M e FXOV1_T-M-J, a que apresenta os melhores resultados é avaliada na Seção 6.4 para o conjunto de instâncias geradas aleatoriamente. As estratégias FXOV2 e FXO-VNDS são analisadas na presente seção e também na Seção 6.4 para as instâncias geradas.

Nos testes realizados para as heurísticas *fix-and-optimize* clássicas na Seção 6.3.6, notou-se um bom desempenho quando utiliza-se a solução da heurística HE1B-DS como solução inicial, pois essas heurísticas conseguem encontrar uma solução razoável com baixos tempos de execução. Assim sobra mais tempo para a resolução dos subproblemas MIP ao longo da heurística *fix-and-optimize*. Portanto, a solução inicial para as heurísticas FXOV1_T, FXOV1_T-M e FXOV1_T-M-J, FXOV2 e FXO-VNDS foi definida como a solução da HE1B-DS nos testes dessa seção e da Seção 6.4. Alguns testes iniciais foram realizados com a solução inicial dada pelo CPLEX e mostraram desempenho pior do que com a solução inicial dada pela HE1B-DS. O tempo para cada subproblema MIP nas heurísticas FXOV1_T, FXOV1_T-M e FXOV1_T-M-J foi determinado como:

$$\max \left\{ \frac{Lim_Tempo - Tempo_Decor}{2 \cdot k_{max} \cdot it}, 100 \right\}, \quad (6.4)$$

ou seja, valor máximo entre 100 segundos e o tempo total subtraído do tempo utilizado pela HE1B-DS, dividido pelo número total de estruturas de vizinhanças e multiplicado pelo número de iterações sem melhoria e por 2, com o intuito de garantir que o algoritmo possa percorrer pelo menos duas vezes cada estrutura de vizinhança. Porém, em testes iniciais percebeu-se que menos de 100 segundos é pouco tempo para resolução dos subproblemas MIP, e portanto, o mínimo de tempo para resolução de cada subproblema MIP foi 100 segundos. Nesse caso, não se aproveitou o tempo não utilizado na resolução dos subproblemas de iterações anteriores. Para as heurísticas FXOV2 e FXO-VNDS, o tempo para cada subproblema foi limitado em 300 segundos. Com alguns testes preliminares, percebeu-se que esse tempo era suficiente para obter no mínimo soluções factíveis para os subproblemas e, em alguns casos, resolvê-los de forma ótima.

Os resultados obtidos com os métodos FXOV1_T, FXOV1_T-M, FXOV1_T-M-J, FXOV2 e FXO-VNDS estão apresentados na Tabela 34.

Tabela 34 – Resultados das heurísticas *fix-and-optimize* com aleatoriedade e da *fix-and-optimize* com princípios de VNDS.

Instância	FXOV1_T			FXOV1_T-M		
	Função Objetivo (u.m.)	gap 1 (%)	Tempo (seg.)	Função Objetivo (u.m.)	gap (%)	Tempo (seg.)
S1	14,02	1,09	617,56	14,02	1,09	756,02
S2	14,01	0,97	87,02	14,01	0,97	120,62
M1	27,12	46,28	3.600,00	25,07	41,87	3.600,00
R1	40,59	75,46	3.586,42	40,59	75,46	3.600,00
R2	48,72	78,80	3.552,92	48,66	78,77	3.600,00
R3	612.549,91	100,00	3.165,05	1.237.548,74	100,00	3.600,00
R4	4.991.933,78	100,00	3.307,11	4.555.926,66	100,00	3.600,00
Média	800.661,16	57,51	2.559,44	827.659,68	56,88	2.696,66

Instância	FXOV1_T-M-J			FXOV2		
	Função Objetivo (u.m.)	gap (%)	Tempo (seg.)	Função Objetivo (u.m.)	gap (%)	Tempo (seg.)
S1	14,02	1,09	2.907,13	14,02	1,09	3.600,00
S2	14,01	0,97	851,61	14,01	0,97	412,28
M1	25,07	41,87	3.600,00	25,07	41,87	3.600,00
R1	39,65	74,88	3.600,00	40,59	75,46	3.600,00
R2	48,66	78,77	3.600,00	53,86	80,82	3.600,00
R3	487.547,78	100,00	3.600,00	487.544,75	100,00	3.600,00
R4	2.576.476,70	100,00	3.600,00	17.734.013,20	100,00	3.600,00
Média	437.737,98	56,80	3.108,39	2.603.100,78	57,17	3.144,61

Instância	FXO-VNDS		
	Função Objetivo (u.m.)	gap (%)	Tempo (seg.)
S1	14,02	1,09	2.735,14
S2	14,01	0,97	785,06
M1	25,07	41,89	3.600,00
R1	42,62	76,63	3.600,00
R2	56,91	81,85	3.600,00
R3	487.547,74	100,00	3.600,00
R4	14.690.612,56	100,00	3.600,00
Média	2.168.330,42	57,49	3.074,32

Comparando-se as três estruturas de vizinhanças utilizadas nas heurísticas FXOV1, o menor valor de função objetivo obtido em média foi para a FXOV1_T-M-J, que é quase metade do que foi obtido pelos métodos FXOV1_T e FXOV1_T-M. Entretanto, a heurística mais rápida é a FXOV2_T, isso porque nessa heurística apenas uma vizinhança é explorada, e assim acaba ficando presa em um mínimo local mais rapidamente do que as outras duas.

Quando compara-se os resultados da FXOV1_T-M-J com a FXOV2, as instâncias S1, S2 e M1 apresentam exatamente os mesmos resultados, enquanto que para a R1, R2 e R4 a FXO-VNDS consegue soluções melhores. A FXOV1_T-M-J comparada com a FXO-VNDS só não tem uma solução melhor para a instância R3, apesar de que para essa instância a solução é praticamente idêntica. Lembrando que a FXO-VNDS e a FXOV2

seguem o mesmo algoritmo de busca em vizinhança (VNDS) e se diferem apenas pela escolha do conjunto de variáveis fixados a cada iteração. Note que para a instância R4 a solução obtida pela FXOV1_T-M-J é mais de 6 vezes melhor do que as soluções obtidas pelas heurísticas FXO-VNDS e FXOV2.

Comparando as heurísticas FXOV1_T-M-J, FXOV2 e FXO-VNDS com a estratégia CPX12, cujos resultados estão na Tabela 27, percebe-se que a FXOV1_T-M-J apresenta valores idênticos ou melhores para todas as instâncias. A FXOV2 só não apresenta valor melhor do que a CPX12 para a instância R2 e a FXO-VNDS não obtém melhores resultados para as instâncias R1 e R2. Os tempos médios de solução para as heurísticas FXOV1_T-M-J, FXOV2 e FXO-VNDS são menores do que a CPX12, que tem tempo médio de 3.161,69.

Para a instância R3 que tem a capacidade restrita, as estratégias FXOV1_T-M-J, FXOV2 e FXO-VNDS apresentam soluções com valor de função objetivo quase três vezes menor do que a solução encontrada por CPX12. O mesmo ocorre para a instância R4, em que as soluções obtidas com essas três estratégias pode chegar a ter função objetivo com valor 62 vezes menor do que a solução obtida com a CPX12. Isso dá indícios de que essas três estratégias FXOV1_T-M-J, FXOV2 e FXO-VNDS conseguem explorar de forma satisfatória o espaço de solução, mostrando-se promissoras para encontrar boas soluções para instâncias difíceis de serem resolvidas pelo CPLEX.

As estratégias *fix-and-optimize* com busca em vizinhança, inclusive a *fix-and-optimize* com princípios de VNDS, escolhem as partições que serão percorridas pelo método com base nas estruturas de vizinhança e nas escolhas aleatórias dos conjuntos serão liberados para otimização em cada uma dessas estruturas. Sendo assim, se a semente do AMPL não for fixa, cada execução do método para uma mesma instância pode liberar conjuntos diferentes para otimização, o que pode acarretar em um valor diferente de solução a cada rodada, sendo que essa solução pode ser melhor ou pior que o anterior. Cabe ressaltar, que nesse trabalho a semente utilizada foi fixa, e portanto os resultados encontrados são sempre os mesmos, independente do número de vezes que a heurística seja executada.

6.3.8 Conclusão

Nesta seção, 6.3, foram testadas todas as estratégias de solução propostas para resolução do problema de programação da produção de bebidas à base de frutas para as instâncias baseadas em dados reais. O objetivo foi escolher, de cada grupo de estratégias, a estratégia com o o melhor desempenho de função objetivo quando aplicada as instâncias baseadas em dados reais. Conclui-se que as estratégias que apresentaram o melhor desempenho para essas instâncias e que serão avaliadas em todas as instâncias na próxima seção (Seção 6.4) são:

- ajuste de parâmetros CPLEX: CPX12;
- heurística de decomposição do problema: HE1B-DS;
- heurística baseada em modelo aproximado: H-SUP;
- heurística *relax-and-fix*: RFX4;
- heurísticas *fix-and-optimize*: CPX12+FXO12, RFX4+FXO3, HE1B-DS+FXO3;
- heurísticas *fix-and-optimize* com busca em vizinhança: FXOV1_T-M-J, FXOV2, FXO+VNDS.

6.4 Testes computacionais: parte II

Após serem selecionados os melhores métodos para as instâncias baseadas em dados reais na Seção 6.3 anterior, estes foram aplicados para resolução das instâncias geradas aleatoriamente e nessa seção estão descritos esses resultados computacionais. O objetivo é avaliar dentre essas estratégias qual tem melhor desempenho na resolução do problema de programação da produção de bebidas à base de frutas. Investiga-se as melhores estratégias com relação ao valor da função objetivo (quantidade de estoques, atrasos, trocas e limpezas temporais) e com relação ao tempo para obtenção de uma solução. Para chegar a essa conclusão utiliza-se todas as instâncias abordadas nessa tese, as baseadas em dados reais e as geradas aleatoriamente.

Inicialmente são apresentados os resultados encontrados com a resolução do modelo MDSL-2E-LT pela estratégia CPX12 para todas as instâncias geradas aleatoriamente. A Tabela 35 apresenta a dimensão do modelo para as instâncias geradas aleatoriamente. Como era esperado, observa-se nessa tabela que a medida que o porte das instâncias crescem, o número de restrições e variáveis também aumentam (equações (4.46), (4.44) e (4.45)). Para as instâncias pertencentes as classes J15M6T4 e J20M6T6, o número total de variáveis chega a ultrapassar 2 milhões e o número de restrições ultrapassa 30 milhões para algumas instâncias. Devido ao número de variáveis e restrições das instâncias, elas foram divididas em três classes: pequenas, médias e grandes, conforme apresentado na Tabela 35. Para essas instâncias, os *softwares* AMPL/CPLEX não conseguiram nem ao menos construir os modelos com os parâmetros da CPX12. Assim, seguindo as recomendações contidas em ILOG (2014), para as instâncias das classes J15M6T4 e J20M6T6, utilizou-se o CPLEX *default* com processamento em paralelo desativado e com a habilitação do parâmetro *nodefile*, para permitir que a memória física do computador seja utilizada para salvar informações ao longo da árvore *branch-and-bound*.

Os resultados obtidos pelo CPLEX na resolução do modelo MDSL-2E-LT para as instâncias geradas aleatoriamente estão apresentados na Tabela 36. Nesta tabela estão

apresentados os valores de função objetivo, o gap^1 que refere-se ao gap calculado através da fórmula (6.1) com o melhor limitante inferior obtido ao longo da árvore *branch-and-bound*; o melhor limitante inferior encontrado; o gap^2 que é calculado com o limitante inferior dado pela relaxação linear do modelo MDSL-2E-LT; os tempos despendidos pelo CPLEX para obtenção das soluções incumbente, primeira factível, e o tempo total. A relaxação linear do modelo para todas as instâncias também foi resolvida, entretanto os resultados foram muito ruins e não serão apresentados nessa tabela. Para as instâncias do tipo J15M6T4 e para as instâncias J20M6T6-1 e J20M6T6-2, o tempo limite foi alcançado pelo *solver* antes mesmo de iniciar a busca na árvore *branch-and-bound* e, portanto, nenhum limitante inferior foi obtido. Por outro lado, os tempos para obtenção de uma solução factível são muito pequenos, desde que sempre é possível atrasar toda a produção, isto é, obter uma solução trivial. Embora, em alguns casos, o CPLEX não consiga nem ao menos resolver a relaxação linear, as heurísticas internas desse *solver* são capazes de encontrar uma solução trivial. Assim como observado para as instâncias baseadas em dados reais, existe uma dificuldade em encontrar bons limitantes inferiores e superiores para o problema. Na Tabela 68 do Apêndice G, os resultados obtidos com a estratégia CPX12 do CPLEX estão detalhados para todas as instâncias.

Tabela 35 – Dimensão das instâncias geradas utilizadas no modelo MDSL-2E-LT.

Instâncias	Variáveis	Variáveis	Variáveis	Restrições	
	Total	Inteiras	Binárias		
Pequenas	J3M2T2-1	47.468	13.584	14.584	536.372
	J3M2T2-2	61.652	17.616	18.964	907.704
	J3M2T2-3	36.556	10.136	11.068	359.208
	J5M2T4-1	50.176	12.816	14.694	298.604
	J5M2T4-2	60.970	15.498	17.844	421.772
	J5M2T4-3	66.772	17.556	19.800	599.468
Médias	J10M2T4-1	117.681	31.867	35.607	911.954
	J10M2T4-2	183.439	48.653	55.033	1.715.306
	J10M2T4-3	145.720	39.974	44.297	1.168.639
	J10M4T4-1	263.258	71.676	79.816	2.070.572
	J10M4T4-2	762.064	217.712	235.565	11.427.722
	J10M4T4-3	545.628	156.530	168.828	6.285.266
Grandes	J15M6T4-1	2.015.368	594.080	632.768	36.058.928
	J15M6T4-2	2.163.160	631.056	676.288	40.812.288
	J15M6T4-3	1.218.856	349.888	378.800	15.406.368
	J20M6T6-1	2.565.870	734.076	799.344	31.772.136
	J20M6T6-2	3.425.442	991.494	1.070.685	53.391.300
	J20M6T6-3	1.834.440	513.996	568.344	18.336.714

A partir daqui, compara-se as melhores estratégias definidas na Seção 6.3: RFX4, H1EB-DS, H-SUP, CPX12+FXO12, RFX4+FXO3, HE1B-DS+FXO3, FXOV1_T-M-J, FXOV2 e FXO-VNDS. A Tabela 37 apresenta o valor da função objetivo obtida por esses métodos. Em negrito estão destacados os menores valores encontrados para cada instância

Tabela 36 – Detalhes das soluções obtidas pela estratégia CPX12 para as instâncias geradas aleatoriamente.

Instância	Função objetivo (u.m.)	gap^1 (%)	Limitante inferior (u.m.)	gap^2 (%)	Tempo para sol. incumbente (seg.)	Tempo para 1ª sol. factível (seg.)	Tempo (seg.)
J3M2T2-1	4.570.081,36	100,00	11,25	100,00	3.278,88	15,39	3.600,00
J3M2T2-2	80,80	78,21	17,60	95,06	2.411,58	42,92	3.600,00
J3M2T2-3	64,52	46,94	34,24	94,96	600,59	9,22	3.600,00
J5M2T4-1	128,73	36,10	82,26	95,29	1.708,89	0,02	3.600,00
J5M2T4-2	4.668.067,93	100,00	27,19	100,00	2.590,47	0,03	3.600,00
J5M2T4-3	1.660.217,03	100,00	27,27	100,00	3.084,91	0,06	3.600,00
J10M2T4-1	44,26	49,26	22,46	88,88	2.571,72	0,08	3.600,00
J10M2T4-2	5.384.043,59	100,00	57,47	100,00	3.570,58	0,14	3.600,00
J10M2T4-3	47,45	66,12	16,08	88,30	3.370,31	0,11	3.600,00
J10M4T4-1	3.491.312,01	100,00	16,54	100,00	3.554,86	0,17	3.600,00
J10M4T4-2	1.702.094.355,50	100,00	11,48	100,00	2.936,05	1,03	3.600,00
J10M4T4-3	1.331.390.668,00	100,00	11,67	100,00	3.600,00	0,55	3.600,00
J15M6T4-1	3.722.444.221,00	*	*	100,00	10,72	10,72	3.600,00
J15M6T4-2	2.780.035.092,00	*	*	*	4,39	4,39	3.600,00
J15M6T4-3	2.321.584.616,00	*	*	100,00	1,44	1,44	3.600,00
J20M6T6-1	3.227.228.020,18	*	*	100,00	3,66	3,66	3.600,00
J20M6T6-2	4.657.845.415,15	*	*	*	5,44	5,44	3.600,00
J20M6T6-3	2.512.697.057,76	100,00	14,36	100,00	2631,02	2,27	3.600,00

entre todos os métodos. Dado que as heurísticas H1EB-DS e H-SUP não utilizam o modelo MDSL-2E-LT, os valores de função objetivo obtidos com elas não apresentam o termo referente a penalização das variáveis $\omega_{m,jtok}^I$ e $\omega_{m,jtol}^{II}$. Logo, para uma comparação adequada, nos valores de função objetivo apresentados na Tabela 37 não está somado esse termo para os métodos baseados no modelo integrado. Na Tabela 38 estão apresentados os tempos computacionais para obtenção das soluções apresentadas na Tabela 37. Os resultados detalhados obtidos com essas estratégias de solução estão apresentados nas Tabelas 69, 70, 71, 72, 73, 74, 75, 76 e 77 do Apêndice G.

Na Tabela 37 também é apresentado um resumo do desempenho dos métodos analisados nessa seção. Na linha “# Melhores soluções” o número de instâncias, dentre as 25 instâncias testadas (reais e geradas), em que cada método obteve o melhor resultado. Por exemplo, a heurística RFX4 obteve o menor valor de solução para 5 das 25 instâncias. Note que os métodos que encontram o maior número de melhores soluções são HE1B-DS+FXO3 e FXOV1_T-M-J; ambos encontram melhores soluções para 11 das 25 instâncias. A linha indicada por “# Soluções melhores que a solução inicial” indica em quantas instâncias os métodos de melhoria melhoraram suas respectivas as soluções iniciais. E por fim, a última linha apresenta a porcentagem de melhoria de cada método com relação a solução obtida com a estratégia CPX12. As estratégias que obtiveram a maior melhoria, em média, foram as heurísticas *fix-and-optimize*: HE1B-DS+FXO3 FXOV1_T-M-J FXOV2 FXO+VNDS, com uma melhoria em torno de 94%. Observando os resultados da Tabela 37, essa porcentagem alta de melhoria com relação ao modelo MDSL-2E-LT é devida às grandes instâncias das classes J10M4T4, J15M6T4 e J20M6T6, para as quais o modelo

encontra somente uma solução trivial, com toda a demanda atrasada, enquanto que a heurística H1EB-DS consegue soluções melhores nesse sentido.

Conforme discutido anteriormente, as instâncias grandes das classes J15M6T4 e J20M6T6 são as mais difíceis de serem resolvidas. As heurísticas *fix-and-optimize*, cujas soluções iniciais são dadas pela heurística HE1B-DS, isto é, HE1B-DS+FXO3, FXOV1_T-M-J, FXOV2 e FXO+VNDS, não foram capazes de melhorar suas soluções iniciais. As heurísticas CPX12+FXO12 e RFX4+FXO3 também não conseguiram melhorar a solução inicial. Isso porque, mesmo fixando parte das variáveis, os subproblemas ainda são grandes e com os tempos disponíveis em cada iteração o CPLEX não consegue encontrar uma solução melhor.

A heurística RFX4 não consegue obter nenhuma solução factível para as instâncias J20M6T6-1 e J20M6T6-2, o mesmo ocorre com a heurística RFX4+FXO3. Como na primeira iteração da *relax-and-fix* o problema resolvido é idêntico à relaxação linear do modelo, no tempo disponível para resolver cada partição não é possível encontrar nenhuma solução, uma vez que mesmo com o tempo total de 3.600 segundos o CPLEX não resolve a relaxação linear para algumas instâncias das classes J15M6T4 e J20M6T6. Note que para as instâncias grandes, as melhores solução são as obtidas pela heurística H1EB-DS, tanto com o tempo de uma hora, como com o tempo de 20 minutos, que são as soluções apresentadas nas colunas HE1B-DS+FXO3, FXOV1_T-M-J, FXOV2 e FXO+VNDS (cabe lembrar que a solução inicial desses métodos é a solução da HE1B-DS em um tempo máximo de 20 minutos).

Para as instâncias pequenas, baseadas em dados reais (S1 e S2), todos os métodos encontram a solução ótima, que é sabido ser a solução com valor 14,00. A otimalidade da instância M1 não foi provada, mas acredita-se que a solução 25,00 também seja a solução ótima para essa instância, e todos os métodos obtiveram esses valores, exceto as heurísticas H1EB-DS e H-SUP. Para as instâncias de R1 a R4, todos os métodos apresentam soluções melhores ou próximas da estratégia CPX12, exceto as heurísticas H1EB-DS e H-SUP, que apresentam soluções com altos valores de estoque e atraso para as instâncias R2 e R3. Cabe destacar que os métodos RFX4+FXO3, FXOV1_T-M-J, FXOV2 e FXO+VNDS apresentam as melhores soluções para a instância R3 e que a heurística HE1B-DS+FXO3 apresenta o menor valor de solução para a instância R4.

O gráfico de perfil de desempenho dos valores de função objetivo da Tabela 37 está mostrado na Figura 38. Observa-se que o melhor método para resolver o problema é a FXOV2, que converge mais rapidamente para $P(f, q) = 1$ quando $q = 2,99$, seguido do método FXOV1_T-M-J que atinge $P(f, q) = 1$ quando $q = 9,72$. Em terceiro lugar estão empatados os métodos HE1B-DS+FXO3 e FXO-VNDS que atingem $P(f, q) = 1$ quando $q = 13,02$. Em quarto lugar está o método HE1B-DS, seguido de H-SUP, CPX12+FXO12, CPX12 e RFX4. Cabe notar mais uma vez que a comparação com relação à função objetivo

é no sentido de que busca-se o método que obtém soluções com a menor quantidade de itens estocados e atrasados e o menor número de trocas e limpezas temporais para as instâncias avaliadas. Obviamente, ao buscar as soluções os métodos utilizam funções objetivos diferentes, uma vez que temos heurísticas que decompõem o modelo em estágios, heurísticas com modelos relaxados e heurísticas baseadas no MDSL-2E-LT, e logo não seria fidedigna a comparação sob outro ponto de vista.

Era esperado que as heurísticas *fix-and-optimize* encontrassem as melhores soluções, uma vez que são heurísticas de melhoria. As heurísticas FXOV2, FXOV1_T-M-J e FXO+VNDS, em particular, já partem de uma solução razoavelmente boa, obtida de maneira rápida com a heurística H1EB-DS, e o fato de explorarem de maneira aleatória os conjuntos que são liberados para a otimização favorecem-as a obterem soluções ainda melhores. Essas três estratégias tem em comum a liberação das variáveis de produção Y_{mjto} , de troca Z_{mijt} e de definição das limpezas temporais W_{mjto}^I e W_{mjto}^{II} e as partições divididas entre períodos (T), itens (J) e tanques preparatórios/linhas (M).

Encontrar uma boa solução inicial a partir da resolução do MDSL-2E-LT é muito difícil, especialmente para instâncias grandes, como discutido anteriormente. Obviamente não era esperado que as heurísticas *relax-and-fix* encontrassem boas soluções para os problemas, quando comparadas aos outros métodos. A mesma observação se aplica a estratégia CPLEX CPX12. As heurísticas HE1B-DS e H-SUP são mais vantajosas do que as *relax-and-fix* e a estratégia CPX12, pois essas duas últimas não conseguem resolver instâncias de grande porte pelo tamanho do modelo MDSL-2E-LT.

O perfil de desempenho para o tempo computacional é mostrado na Figura 39. Os métodos mais rápidos são aqueles que não utilizam o MDSL-2E-LT, ou seja, HE1B-DS, que obtém soluções factíveis em 717,21 segundos em média e H-SUP com a média de 856,87 segundos. Em terceiro lugar está o método HE1B-DS+FXO3 com em média 2.745,90 segundos.

Portanto, para os tipos de instâncias testadas, numa situação em que existe tempo disponível para obter a programação da produção, conclui-se que usar o método FXOV2 é a melhor opção. Para uma situação em que o tempo disponível não é muito grande e deseja-se uma solução com qualidade razoável, o método HE1B-DS+FXO3 deve ser usado. Já para uma situação em que pretende-se obter uma solução factível rapidamente, indica-se o uso do método H1EB-DS, mas sua qualidade pode não ser tão boa como a dos métodos anteriores.

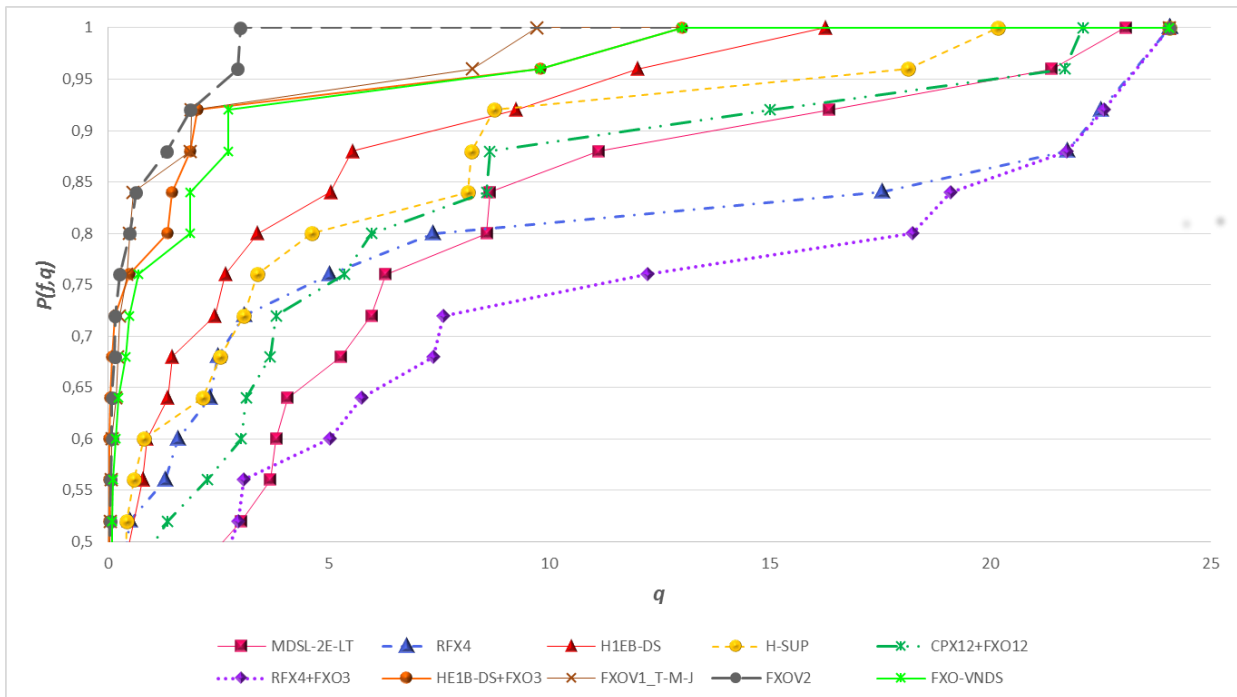


Figura 38 – Perfil de desempenho da função objetivo dos melhores métodos de solução para o problema. Fonte: a autora.

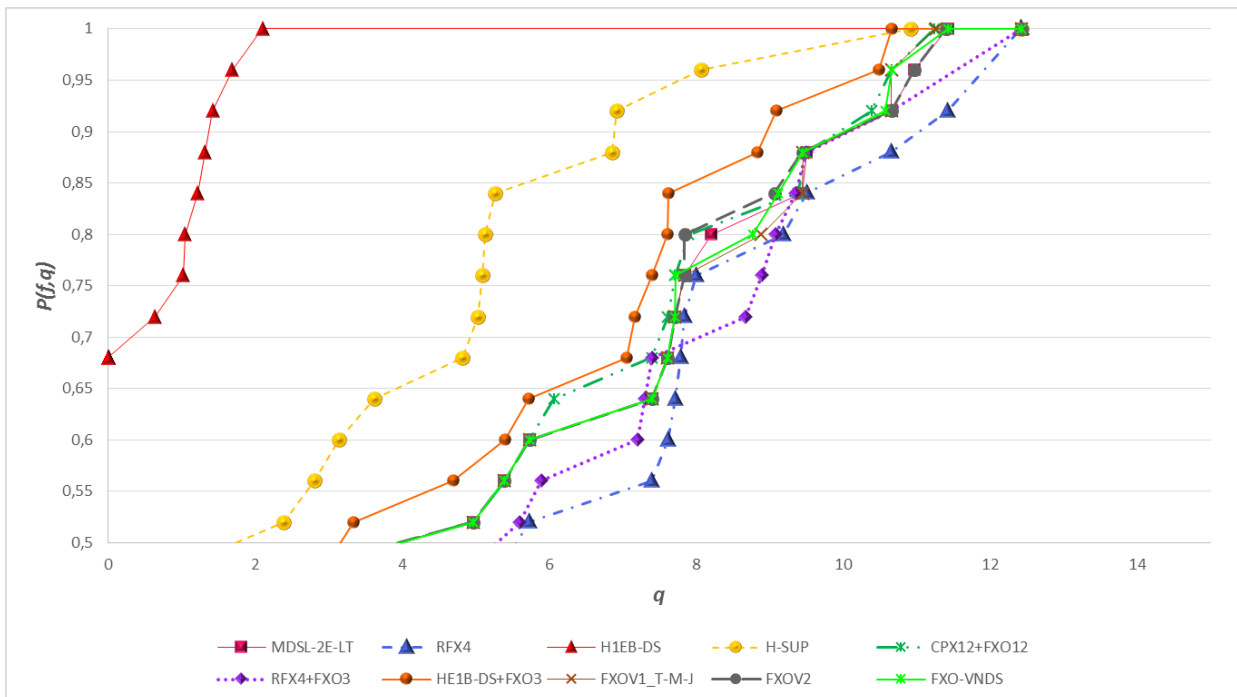


Figura 39 – Perfil de desempenho do tempo dos melhores métodos de solução para o problema. Fonte: a autora.

Tabela 37 – Valores das funções objetivos (u.m.) obtidas pelos melhores métodos para todas as instâncias.

Instâncias	CPX12	RFX4	HEIB-DS	H-SUP	CPX12+FXO12	RFX4+FXO3	HEIB-DS+FXO3	FXOV1	T-M-J	FXOV2	FXO+VNDS	Menor Valor
S1	14,00	14,00	24,00	15,00	14,00	14,00	14,00	14,00	14,00	14,00	14,00	14,00
S2	14,00	14,00	24,00	15,00	14,00	14,00	14,00	14,00	14,00	14,00	14,00	14,00
M1	25,00	25,00	31,00	29,00	25,00	25,00	25,00	25,00	25,00	25,00	25,00	25,00
R1	41,00	40,00	47,00	47,00	44,00	40,00	39,00	39,00	39,00	40,00	42,00	39,00
R2	50,00	49,00	195,962,00	20,680,00	57,00	48,00	49,00	48,00	48,00	53,00	56,00	48,00
R3	1.237.545,00	1.196.377,33	5.094.801,00	2.154.214,67	1.237.545,00	487.546,00	1.237.548,00	487.547,00	487.547,00	487.544,00	487.547,00	487.544,00
R4	173.182.854,00	11.001.671,00	4.068.931,00	2.979.833,67	19.474.138,17	13.737.475,83	2.227.572,67	2.576.474,33	17.734,010,00	14.690.610,50	2.227.572,67	2.227.572,67
J3M2T2-1	4.570,080,40	70,80	58,90	57,90	87,20	64,80	55,90	54,90	55,90	55,90	54,90	54,90
J3M2T2-2	79,60	63,70	60,60	58,60	98,00	33.009,832,43	60,60	60,60	60,60	60,60	60,60	58,60
J3M2T2-3	63,80	62,80	4.470,994,22	56,60	62,80	62,80	62,80	63,80	63,80	62,80	63,80	56,60
J5M2T4-1	128,40	113,90	118,10	117,50	128,40	116,40	114,10	114,10	114,10	114,10	118,10	113,90
J5M2T4-2	4.668,067,29	1.027.426,64	47.420,730,87	1.805,573,62	4.868,703,16	1.169,294,83	4.176,008,83	3.782,028,46	7.835,850,49	6.748,601,21	1.027.426,64	1.027.426,64
J5M2T4-3	1.660,216,14	109,447,11	3.264,063,37	99.245,69	104.406,50	761,948,24	270,541,28	132.737,84	248.707,63	358.836,45	99.245,69	99.245,69
J10M2T4-1	43,80	37,40	32,00	39,40	36,40	5.840,39	30,00	30,00	30,00	30,00	30,00	30,00
J10M2T4-2	5.384,042,38	413,461,16	351,485,31	207,981,58	5.668,628,59	7.444,647,16	138.819,90	204.231,49	142.605,78	223.821,48	138.819,90	138.819,90
J10M2T4-3	46,80	48,80	45,70	44,70	54,00	168.579,60	35,00	41,70	41,70	45,70	45,70	35,00
J10M4T4-1	1.702,094,355,50	36.811,924,61	192,70	238,20	6.350,284,54	59.900,742,58	196,20	202,60	203,30	203,30	203,30	192,70
J10M4T4-2	1.331.390,668,00	1.693.417,888,18	291,490,11	574,870,779,97	1.632,078,219,46	1.693,417,888,18	428,696,09	407,108,97	482,10	428,694,41	482,10	482,10
J10M4T4-3	473,827,23	1.265,837,225,77	212,70	61.594,158,20	959,318,013,61	1.331.390,668,00	1.767,050,07	64.923,96	328,00	1.767,050,07	212,70	212,70
J15M6T4-1	3.722.444,221,00	2.587.771,272,00	460.742.743,66	2.655.564,912,90	3.722.444,221,00	2.587.771,272,00	485.843,568,94	485.843,568,94	485.843,568,94	485.843,568,94	485.843,568,94	460.742.743,66
J15M6T4-2	2.780,035,092,00	1.835,181,425,91	217.601.920,71	1.818,783,551,55	2.780,035,092,00	1.835,181,425,91	303,871,618,31	303,871,618,31	303,871,618,31	303,871,618,31	303,871,618,31	217.601.920,71
J15M6T4-3	2.321.584,616,00	1.204.192,427,82	232.987,274,03	906.508,710,24	2.321.584,616,00	1.204.192,427,82	37.204.906,85	37.204.906,85	37.204.906,85	37.204.906,85	37.204.906,85	37.204.906,85
J20M6T6-1	3.227,228,020,18	**	8.401.247,60	2.406,508,444,90	3.227,228,020,18	**	30.504,781,92	30.504,781,92	30.504,781,92	30.504,781,92	30.504,781,92	8.401.247,60
J20M6T6-2	4.657,845,415,15	**	1.774,291,352,15	3.449,992,588,20	4.657,845,415,15	**	332.387.316,23	332.387.316,23	332.387.316,23	332.387.316,23	332.387.316,23	332.387.316,23
J20M6T6-3	2.512,697,057,76	1.026,276,931,76	16.958,575,48	1.888,898,294,70	2.512,697,057,76	1.026,276,931,76	6.237.371,10	6.237.371,10	6.237.371,10	6.237.371,10	6.237.371,10	6.237.371,10
Média	897.859,863,38	420.140,783,42	111.045,696,73	550.799,587,55	874.037,399,28	425.865,952,42	48.251,859,83	48.148,215,32	48.899,992,23	48.830,218,08	48.148,215,32	48.148,215,32
Desvio Padrão	1.442.583,024,14	768.095,082,45	362.653,931,62	1.024,556,969,78	1.441,715,200,45	769,936,316,93	126.332,595,99	126.372,804,39	126.130,372,04	126.140,173,19	126.130,372,04	126.130,372,04
# Melhores soluções	3/25	5/25	5/25	3/25	3/25	4/25	11/25	11/25	9/25	8/25	8/25	18
# Soluções melhores que a solução inicial	-	-	-	-	15	14	16	18	18	18	18	18
Melhoria na média com relação a CPX12	53,21%	87,63%	87,63%	38,65%	2,65%	52,57%	94,63%	94,64%	94,55%	94,55%	94,56%	94,56%

Tabela 38 – Tempos despendidos pelos melhores métodos para obtenção da solução.

Instância	MDSL-2E-LT	REF4	HIEB-DS	H-SUP	CPX12+FXO12	RFX4+FXO3	HEIB-DS+FXO3	FXOV1	T-M-J	FXOV2	FXO+VNS	Menor valor
S1	3.600,00	398,47	1,80	61,11	1.203,31	732,61	353,08	2.907,13	3.600,00	2.735,14	1,80	
S2	531,89	459,05	1,80	22,06	2.404,09	853,97	75,95	851,61	412,28	785,06	1,80	
M1	3.600,00	2.991,70	5,16	600,17	1.230,94	2.800,95	2.345,74	3.600,00	3.600,00	3.600,00	5,16	
R1	3.600,00	3.600,00	17,19	600,42	3.600,00	2.703,66	2.280,97	3.600,00	3.600,00	3.600,00	17,19	
R2	3.600,00	3.600,00	67,95	600,50	3.600,00	3.300,78	3.584,91	3.600,00	3.600,00	3.600,00	67,95	
R3	3.600,00	3.600,00	85,72	600,50	3.600,00	2.706,22	2.212,96	3.600,00	3.600,00	3.600,00	85,72	
R4	3.600,00	3.600,00	1.216,05	601,17	3.456,97	3.600,00	1.641,05	3.600,00	3.600,00	3.600,00	601,17	
J3M2T2-1	3.600,00	3.600,00	2,23	600,27	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	2,23	
J3M2T2-2	3.600,00	3.600,00	4,97	600,34	3.600,00	3.600,00	2.704,56	3.479,99	2.664,85	2.756,78	4,97	
J3M2T2-3	3.600,00	3.600,00	21,33	601,50	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	21,33	
J5M2T4-1	3.600,00	3.600,00	18,41	600,34	1.228,33	2.703,47	3.600,00	3.600,00	3.600,00	3.600,00	18,41	
J5M2T4-2	3.600,00	3.600,00	1.236,39	601,08	3.600,00	2.417,34	3.600,00	3.600,00	3.600,00	3.600,00	601,08	
J5M2T4-3	3.600,00	3.600,00	772,13	1.201,55	2.028,89	2.418,98	3.600,00	3.600,00	3.600,00	3.600,00	772,13	
J10M2T4-1	3.600,00	3.600,00	1,31	2.533,22	3.184,17	853,50	1.875,48	3.219,85	3.529,65	3.600,00	1,31	
J10M2T4-2	3.600,00	3.600,00	115,48	602,81	3.600,00	3.600,00	1.166,49	3.600,00	3.600,00	3.600,00	115,48	
J10M2T4-3	3.600,00	3.600,00	15,70	603,39	3.066,58	936,00	2.259,55	3.600,00	3.600,00	3.317,49	15,70	
J10M4T4-1	3.600,00	3.600,00	457,33	601,16	3.600,00	3.014,59	3.600,00	3.600,00	3.600,00	3.600,00	457,33	
J10M4T4-2	3.600,00	3.600,00	1.406,50	607,52	3.600,00	2.766,14	2.447,64	3.600,00	3.600,00	3.600,00	607,52	
J10M4T4-3	3.600,00	3.600,00	866,05	1.818,77	3.600,00	2.734,48	2.499,09	3.510,69	3.600,00	3.600,00	866,05	
J15M6T4-1	3.600,00	3.600,00	1.704,27	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	1.704,27	
J15M6T4-2	3.600,00	3.600,00	1.584,64	638,62	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	638,62	
J15M6T4-3	3.600,00	3.600,00	1.877,61	1.208,08	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	1.208,08	
J20M6T6-1	3.600,00	**	2.604,83	608,88	3.600,00	**	3.600,00	3.600,00	3.600,00	3.600,00	608,88	
J20M6T6-2	3.600,00	**	2.081,95	649,53	3.600,00	**	3.600,00	3.600,00	3.600,00	3.600,00	649,53	
J20M6T6-3	3.600,00	3.600,00	1.763,42	658,70	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	658,70	
Média	3.477,28	3.297,79	717,21	856,87	3.160,13	2.754,03	2.745,90	3.438,77	3.432,27	3.407,78	717,21	
Desvio Padrao	613,62	914,15	838,61	762,04	829,68	992,18	1.076,90	560,91	656,31	596,48	560,91	

6.5 Avaliação de cenários

Nessa seção são analisados alguns cenários alternativos para o problema de programação da produção de bebidas à base de frutas. Na Seção 6.4 anterior definiu-se as melhores estratégias para resolver esse problema. Assim, utilizando a estratégia FXOV2, nessa seção são analisados 2 diferentes cenários que poderiam ser encontrados em situações práticas. Os cenários são testados utilizando as instâncias baseadas em dados reais e as instâncias geradas aleatoriamente, com a modificação de alguns parâmetros. Além disso, realiza-se uma análise do modelo MDSL-2E, que é uma simplificação do modelo MDSL-2E-LT para uma situação em que não exista limpezas temporais, e uma comparação do mesmo com o modelo MDSL-2E-LT. A análise do modelo MDSL-2E é considerada como um terceiro cenário.

6.5.1 Cenário 1: Alto custo da limpeza temporal

A primeira situação a ser analisada é o com alto custo da limpeza temporal, aumentando de $Cl_t = 1$ para $Cl_t = 100$. Suponha que exista um custo alto para a realização de uma limpeza temporal, por exemplo, porque os produtos utilizados para a realização da limpeza são muito caros, ou porque para a realização da limpeza temporal seja necessária a utilização de um recurso muito escasso. A ideia é observar esse impacto no comportamento das limpezas temporais, das trocas e do número de instâncias com atraso e estoque.

Na Figura 40 apresenta-se um gráfico de comparação do número de limpezas temporais nos tanques preparatórios apresentados nas soluções obtidas com a FXOV2 para os custos de limpeza temporal $Cl_t = 1$ e $Cl_t = 100$, e na Figura 41 o número de limpezas temporais nas linhas. Como era esperado, para a maioria das instâncias o número de limpezas temporais diminuiu quando o custo de realização dessas limpezas aumenta. O número de limpezas temporais nos tanques preparatórios e nas linhas não diminuiu somente para 5 e 4 instâncias, respectivamente, de um total de 25 instâncias.

Os custos de troca para todas as instâncias são apresentados na Figura 42(a) e o número de instâncias com atraso e estoque não nulos em algum período estão exibidos na Figura 42(b). Nota-se que quando o custo da limpeza temporal aumenta ($Cl_t = 100$), o número de limpezas temporais diminuem, entretanto, mais trocas são realizadas. Exceto para as instâncias J15M6T4-2 e J20M6T6-1, para todas as outras instâncias os custos de troca foram maiores (ou iguais) quando $Cl_t = 100$. O número de instâncias com estoque diferente de zero em algum período se manteve o mesmo, e apenas uma instância passou a ter atraso diferente de zero quando o custo da limpeza temporal aumentou.

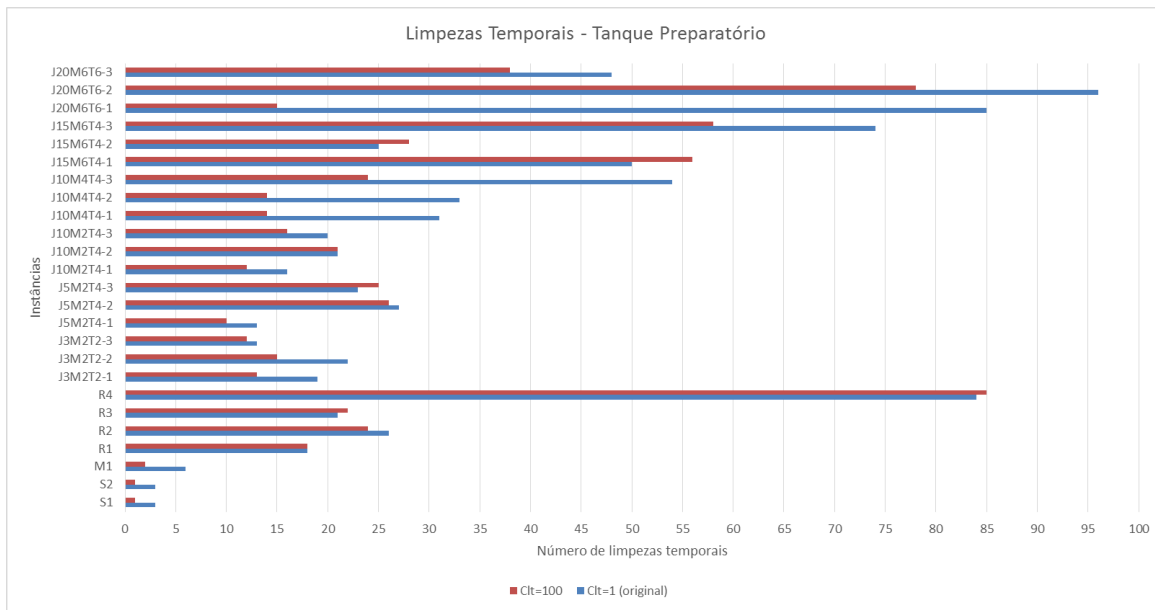


Figura 40 – Comparação das limpezas temporais dos tanques preparatórios nas soluções obtidas pela FXOV2 com $Cl_t = 1$ e $Cl_t = 100$.

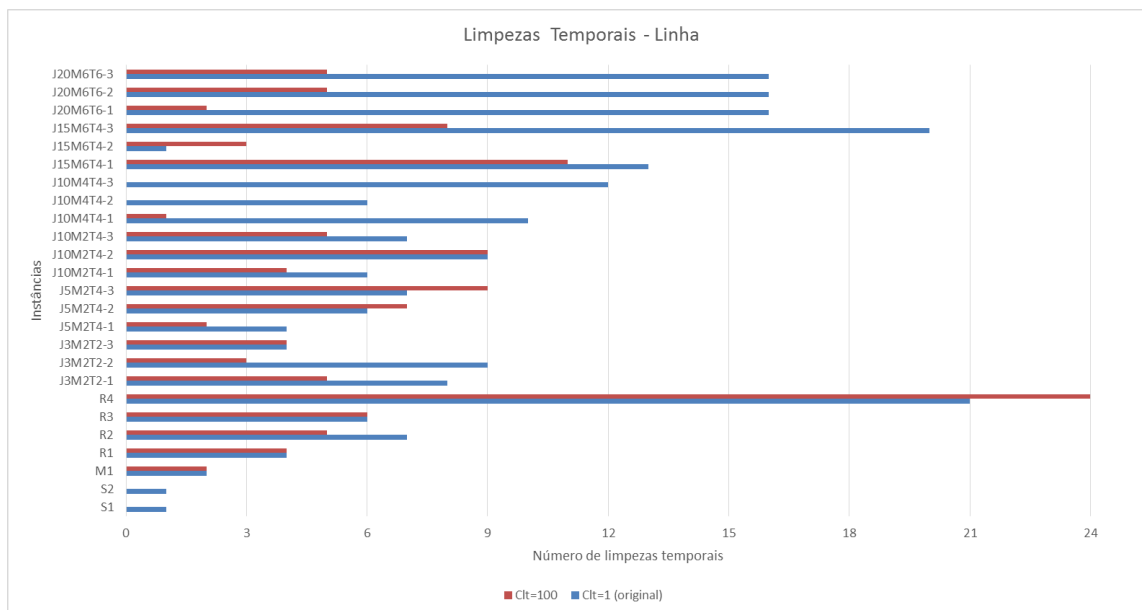
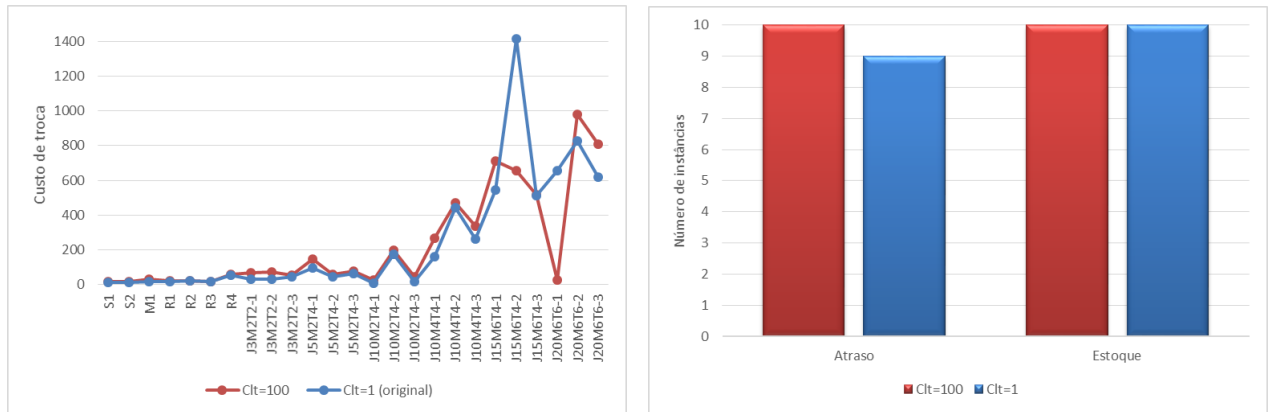


Figura 41 – Comparação das limpezas temporais das linhas nas soluções obtidas pela FXOV2 com $Cl_t = 1$ e $Cl_t = 100$.

6.5.2 Cenário 2: Capacidade restrita

Suponha que a capacidade esteja restrita, ou que haja uma redução imprevista no tempo de capacidade disponível em cada período. Essa redução pode ser referente a alguma quebra de máquina, ou manutenção de alguma máquina, ou ainda alguma parada programada devido a feriados, por exemplo. Nesse cenário, o teste realizado nessa seção



(a) Comparação dos custos de troca para cada instância. (b) Número de instâncias com estoque e atraso não nulos.

Figura 42 – Comparação dos atrasos, estoques e trocas obtidos com a FXOV2 quando $Clt = 1$ e $Clt = 100$.

considera uma redução de 20% das capacidades de todos os períodos e máquinas em relação às instâncias anteriores, ou seja, uma redução de 20% em Cap_{mt} , com as mesmas demandas anteriores.

Espera-se, que mantendo a mesma demanda e com menos tempo disponível, menos itens serão produzidos, ou seja, aumentarão os atrasos, por exemplo. Por outro lado, espera-se que o número de limpezas e trocas diminua, uma vez que existe menos tempo para produzir. De fato, na média, o número de limpezas temporais nos tanques preparatórios reduziu de cerca de 33 nos resultados obtidos com a capacidade original para cerca de 27 quando a capacidade é reduzida, e o número de limpezas temporais nas linhas diminuiu de cerca de 8 para cerca de 6. Uma diminuição também acontece nos custos de troca de 243,55 na média para 211,06 quando a capacidade é reduzida.

Na Tabela 39 são apresentados os custos de estoque e atraso obtidos pela FXOV2 para as instâncias com capacidade original e capacidade reduzida. O número de instâncias com estoque passa de 10 para 16 e de instâncias com atraso de 9 para 19 quando a capacidade é reduzida. Apesar disso, para as instâncias J15M6T4-1, J15M6T4-2 e J20M6T6-3, os custos de atraso diminuíram quando a capacidade foi reduzida. Isso pode acontecer pois essas não são as soluções ótimas. Além disso, para as instâncias pertencentes as classes J15M6T4 e J20M6T6, a FXOV2 não consegue melhorar a solução inicial obtida pela HE1B-DS por serem problemas muito grandes, em que o CPLEX não consegue nem resolver a relaxação linear. De forma geral, percebe-se que para todas as instâncias a redução de 20% da capacidade implica em muitos atrasos, inclusive para as instâncias reais R1, R2 e R3.

Tabela 39 – Custos de atraso e estoque obtidos pela FXOV2 para as instâncias com capacidade original e capacidade reduzida.

Instância	Custos de atraso		Custos de estoque	
	Capacidade original	Capacidade reduzida	Capacidade original	Capacidade reduzida
S1	0,00	0,00	0,00	0,00
S2	0,00	0,00	0,00	0,00
M1	0,00	0,00	0,00	0,00
R1	0,00	14.861.000,00	0,00	735,00
R2	0,00	58.299.916,67	0,00	0,00
R3	487.500,00	54.475.000,00	0,00	0,00
R4	8.929.666,67	51.144.666,67	8.804.183,33	42.000,00
J3M2T2-1	0,00	16.724.546,25	0,00	4.285,00
J3M2T2-2	0,00	27.967.579,34	0,00	900,43
J3M2T2-3	0,00	13.559.743,80	0,00	0,00
J5M2T4-1	0,00	18.363.519,65	0,00	3.048,70
J5M2T4-2	6.126.123,64	58.108.686,81	1.709.647,85	172.403,85
J5M2T4-3	0,00	34.183.078,08	248.615,03	20.891,46
J10M2T4-1	0,00	0,00	0,00	4.849,12
J10M2T4-2	0,00	15.231.393,46	142.400,18	22.798,29
J10M2T4-3	0,00	0,00	0,00	0,00
J10M4T4-1	0,00	89.238.543,36	0,00	28.082,40
J10M4T4-2	0,00	102.442.584,99	0,00	45.626,02
J10M4T4-3	0,00	97.003.509,10	0,00	11.658,37
J15M6T4-1	483.708.290,62	306.522.345,57	2.134.668,42	26.217,03
J15M6T4-2	301.328.460,27	350.529.279,58	2.541.715,84	0,00
J15M6T4-3	36.693.365,45	117.629.653,89	510.933,70	19.966,85
J20M6T6-1	28.531.604,18	1.351.742.111,50	1.972.419,34	23.301,87
J20M6T6-2	331.725.622,48	501.726.955,62	663.420,58	1.087,28
J20M6T6-3	4.066.798,55	0,00	2.169.890,05	39.763,52

6.5.3 Cenário 3: Modelo MDSL-2E sem consideração da limpeza temporal

Na Seção 4.1.2 foi apresentado o modelo MDSL-2E, que é uma simplificação do modelo MDSL-2E-LT por não considerar as limpezas temporais. A fim de analisar os efeitos das limpezas temporais na programação da produção das bebidas à base de fruta, foram realizados testes computacionais para as mesmas instâncias do modelo MDSL-2E-LT. Objetiva-se também analisar o impacto das variáveis e restrições adicionadas ao modelo MDSL-2E para consideração das limpezas temporais.

A Tabela 40 apresenta o número de variáveis e restrições de ambos modelos, MDSL-2E-LT e MDSL-2E, e o número de variáveis e restrições que são reduzidas ao desconsiderar as limpezas temporais. O modelo MDSL-2E não possui variáveis inteiras e por esse motivo a quantidade desse tipo de variável não aparece na tabela. Para o modelo MDSL-2E, percebe-se que o número de variáveis e restrições não chega a um milhão para nenhuma das instâncias, nem mesmo para as instâncias maiores das classes J15M6T4 e J20M6T6. Como já explicitado na Tabela 6 do Capítulo 4, esses números mostram o quão são expressivas

as restrições para determinação das limpezas temporais e as variáveis utilizadas nessas restrições.

Tabela 40 – Comparação do número de variáveis e restrições para os modelos MDSL-2E e MDSL-2E-LT.

Instância	MDSL-2E-LT			MDSL-2E			Redução	
	Variáveis	Variáveis Binárias	Restrições	Variáveis	Variáveis Binárias	Restrições	Variáveis	Restrições
S1	6.220	276	18.288	1.236	240	2.096	-4.984	-16.192,00
S2	3.076	60	7.032	660	144	1.136	-2.416	-5.896,00
M1	23.904	1.020	90.507	2.884	570	4.909	-21.020	-85.598,00
R1	63.104	2.504	398.424	14.452	2.592	24.308	-48.652	-374.116,00
R2	74.855	2.955	454.655	17.162	3.090	28.882	-57.693	-425.773,00
R3	59.884	2.364	363.724	13.732	2.472	23.108	-46.152	-340.616,00
R4	205.088	8.138	1.294.878	46.942	8.424	78.974	-158.146	-1.215.904,00
J3M2T2-1	47.468	14.584	536.372	6.900	1.184	11.536	-40.568	-524.836
J3M2T2-2	61.652	18.964	907.704	8.988	1.532	15.016	-52.664	-892.688
J3M2T2-3	36.556	11.068	359.208	6.300	1.084	10.536	-30.256	-348.672
J5M2T4-1	50.176	14.694	298.604	12.040	2.190	20.288	-38.136	-278.316
J5M2T4-2	60.970	17.844	421.772	14.776	2.646	24.848	-46.194	-396.924
J5M2T4-3	66.772	19.800	599.468	14.452	2.592	24.308	-52.320	-575.160
J10M2T4-1	117.681	35.607	911.954	22.850	4.510	39.038	-94.831	-872.916
J10M2T4-2	183.439	55.033	1.715.306	38.162	7.062	64.558	-145.277	-1.650.748
J10M2T4-3	145.720	44.297	1.168.639	26.612	5.137	45.308	-119.108	-1.123.331
J10M4T4-1	263.258	79.816	2.070.572	49.814	9.724	85.050	-213.444	-1.985.522
J10M4T4-2	762.064	235.565	11.427.722	110.996	19.921	187.020	-651.068	-11.240.702
J10M4T4-3	545.628	168.828	6.285.266	78.194	14.454	132.350	-467.434	-6.152.916
J15M6T4-1	2.015.368	632.768	36.058.928	238.984	44.544	405.312	-1.776.384	-35.653.616
J15M6T4-2	2.163.160	676.288	40.812.288	275.368	50.608	465.952	-1.887.792	-40.346.336
J15M6T4-3	1.218.856	378.800	15.406.368	173.992	33.712	296.992	-1.044.864	-15.109.376
J20M6T6-1	2.565.870	799.344	31.772.136	372.966	74.592	640.392	-2.192.904	-31.131.744
J20M6T6-2	3.425.442	1.070.685	53.391.300	461.040	89.271	787.182	-2.964.402	-52.604.118
J20M6T6-3	1.834.440	568.344	18.336.714	300.642	62.538	519.852	-1.533.798	-17.816.862

Na Tabela 41 são apresentadas as soluções obtidas com os modelos MDSL-2E e MDSL-2E-LT. A primeira observação que pode ser feita é com relação ao tempo computacional, que em geral diminui substancialmente, como era esperado. Os tempos para o modelo MDSL-2E-LT não estão apresentados na tabela pois, exceto para a instância S2 que é resolvida até a otimalidade em 531,89 segundos, para todas as outras instâncias foram utilizados os 3.600 segundos disponíveis. Para o modelo MDSL-2E são encontradas soluções ótimas para 13 das 25 instâncias em um tempo médio de 45,09 segundos. Para as instâncias maiores das classes J15M6T4 e J15M6T4, a *gap* é de 100%, porém foi possível encontrar um limitante inferior para todos os problemas dessa classe, diferentemente do modelo MDSL-2E-LT, que para algumas instâncias não é possível resolver nem a relaxação linear no nó raiz da árvore *branch-and-cut*.

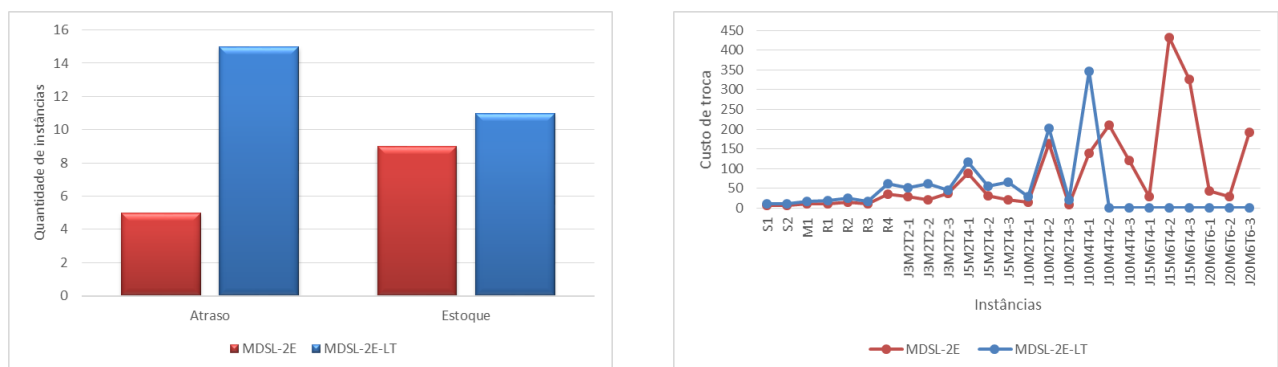
Tabela 41 – Comparação entre os resultados obtidos por MDSL-2E e MDSL-2E-LT.

Instância	MDSL-2E			MDSL-2E-LT	
	Função objetivo (u.m.)	gap (%)	Tempo (seg.)	Função objetivo (u.m.)	gap (%)
S1	6,00	0,00	0,36	14,01	21,40
S2	6,00	0,00	0,11	14,01	0,93
M1	10,00	0,00	0,38	25,02	41,72
R1	10,00	0,00	4,77	41,18	75,71
R2	15,00	0,00	121,06	50,18	79,34
R3	11,00	0,00	124,89	1.237.545,21	100,00
R4	730.036,00	100,00	3.600,00	173.182.854,35	100,00
J3M2T2-1	28,90	0,98	89,38	4.570.081,36	100,00
J3M2T2-2	20,50	0,00	8,34	80,80	78,21
J3M2T2-3	36,60	0,80	4,39	64,52	46,94
J5M2T4-1	88,20	0,98	221,43	128,73	36,10
J5M2T4-2	69.344,71	99,95	3.600,00	4.668.067,93	100,00
J5M2T4-3	21,70	0,00	3,58	1.660.217,03	100,00
J10M2T4-1	15,40	0,00	3,43	44,26	49,26
J10M2T4-2	3.002,04	94,61	3.600,00	5.384.043,59	100,00
J10M2T4-3	8,00	0,00	4,07	47,45	66,12
J10M4T4-1	140,00	16,35	3.600,00	3.491.312,01	100,00
J10M4T4-2	1.963.830,90	100,00	3.600,00	1.702.094.355,50	100,00
J10M4T4-3	120,00	43,45	3.600,00	1.331.390.668,00	100,00
J15M6T4-1	373.981.028,37	100,00	3.600,00	3.722.444.221,00	*
J15M6T4-2	171.892.434,42	100,00	3.600,00	2.780.035.092,00	*
J15M6T4-3	326,40	94,41	3.600,00	2.321.584.616,00	*
J20M6T6-1	404.128.478,86	100,00	3.600,00	3.227.228.020,18	*
J20M6T6-2	532.365.962,38	100,00	3.600,00	4.657.845.415,15	*
J20M6T6-3	146.056.231,98	100,00	3.600,00	2.512.697.057,76	100,00

Percebe-se também na Tabela 41 a redução do valor da função objetivo para todas as instâncias. Para as instâncias que não apresentam atraso e nem estoque, como por exemplo a instância J5M2T4-1, o valor da função objetivo do modelo MDSL-2E refere-se somente aos custos de troca, enquanto que para essa mesma instância a solução do modelo MDSL-2E-LT envolve os custos de troca e de limpezas temporais. A retirada da exigência de limpezas temporais implicou para a instância R3, por exemplo, em não gerar atrasos; e para a instância R4, implicou em diminuir estoques. Na Figura 43(a) é apresentado o número de instâncias com atraso ou estoque não nulos para os dois modelos. Nota-se que para o modelo MDSL-2E-LT, 15 instâncias têm soluções com atrasos em algum dos períodos, enquanto que para o modelo MDSL-2E, apenas 5 instâncias têm soluções com atrasos. Isso mostra o quanto de recurso de capacidade é consumido pelas limpezas temporais e, conseqüentemente, a importância em dimensionar e sequenciar os lotes com a consideração dessas limpezas da forma mais otimizada possível, como foi apresentado nesse trabalho.

A Figura 43(b) apresenta um gráfico relacionando os custos de troca apresentados pelos dois modelos para cada uma das 25 instâncias, sendo 1 a notação para primeira instância (S1) da Tabela 41, e 25 a notação para a última instância (J20M6T6-3). Neste caso

nota-se que os custos de troca são mais altos para as instâncias de 1 (S1) a 17 (J10M4T4-1). Para as instâncias J10M4T4-2, J10M4T4-3 e das classes J15M6T4 e J20M6T6 (de 18 a 25), o modelo MDSL-2E-LT apresenta custos de troca nulos, pois para essas instâncias conseguiu-se apenas soluções triviais, com toda a demanda atrasada.



(a) Número de instâncias com atraso e estoque não nulos.

(b) Comparação dos custos de troca para cada instância.

Figura 43 – Comparação dos estoques, atrasos e trocas obtidos com os modelos MDSL-2E-LT e MDSL-2E

6.6 Comparação do melhor método de solução com a solução da empresa

Nesta seção realiza-se uma comparação entre as soluções encontradas na prática da Fábrica A com as soluções obtidas pela melhor estratégia de solução para o problema de programação da produção de bebidas à base de frutas, a FXOV2. Cabe lembrar que os custos de atraso, estoque, troca e limpezas temporais não foram fornecidos pela empresa, conforme explicado anteriormente na Seção 6.1, e assim essa comparação tem limitações.

Na Tabela 42 são apresentadas de forma detalhada as soluções obtidas com a FXOV2 e as soluções apresentadas pela Fábrica A para as instâncias reais R1, R2 e R3.

Tabela 42 – Resultados da heurística HE2B-DS+FXOV2 comparados com a solução da empresa.

Inst.	HE2B-DS+FXO2					Empresa				
	Atraso (unid.)	Estoque (unid.)	Trocas (u.m.)	Limpezas temp.	Função obj. (u.m.)	Atraso (unid.)	Estoque (u.m.)	Trocas (u.m.)	Limpezas temp.	Função obj. (u.m.)
R1	0	0	18,00	22	40,00	0	0	20,00	31	51,00
R2	0	0	20,00	33	53,00	0	0	26,00	38	64,00
R3	4.875	0	17,00	27	487.544,75	12.250	0	18,00	34	1.225.052,00

Para as instâncias R1 e R2, em que não há estoque e nem atraso, a heurística conseguiu reduzir o número de limpezas temporais com relação às soluções da empresa. Os

custos de troca também foram reduzidos para essas instâncias em duas unidades para a R1 e em seis unidades para R2. Na instância R3, o valor das trocas diminuiu em apenas uma unidade. Por outro lado, o número de limpezas temporais foi reduzido em sete unidades, permitindo um melhor aproveitamento da capacidade e uma diminuição de quase três vezes a quantidade atrasada pela empresa.

Nas Tabelas 43 e 44, são apresentadas as programações da produção para o período $T = 1$ para a instância R1 obtidas pela heurística HE2B-DS+FXO2 e pela empresa, respectivamente. Nessas tabelas estão apresentados os itens demandados (uva e morango) o número de lotes que devem ser produzidos e a quantidade em litros de cada lote, em cada uma das máquinas. O intuito dessas tabelas é analisar a diferença entre os valores de solução da empresa e os valores obtidos com a heurística HE2B-DS+FXOV2, como apresentados na Tabela 42.

A primeira observação a ser feita é que a empresa tem uma estratégia de sempre fazer lotes máximos (tanques cheios) e apenas o último lote com a quantidade que falta para completar a demanda, como pode ser visto na Tabela 44. Outra estratégia adotada pela empresa tentar produzir ao mesmo tempo o mesmo sabor nos dois tanques preparatórios/linhas. Veja que em $M = 1$ são produzidos 25 lotes de Uva e ao mesmo tempo em $M = 2$ são produzidos 24 lotes. Nesse exemplo, a velocidade de envase da linha 2 é de 64 litros/min, que é um pouco menor do que na linha 1, com 52 litros/min. Esse tipo de programação não prevê a grande quantidade de tempo despendido com as esperas. Essa programação é feita manualmente, e com certeza humanamente impossível prever todas as esperas e tamanhos de lotes que poderiam se adequar melhor as esperas. Em situações em que a capacidade está folgada, como no caso das instâncias R1 e R2, a única perda é com relação ao número de limpezas e trocas que devem ser feitas, mas que não gera nenhum custo muito oneroso a empresa. No entanto, em situações em que a capacidade está mais apertada, como no caso da instância R3, não fazer uma programação da produção otimizada e minimizando os tempos despendidos com as esperas e com as limpezas pode acarretar em atrasos na entrega, o que gera prejuízos à empresa.

Na programação obtida com a heurística (Tabela 43), estão descritos os lotes exatamente na ordem em que foram produzidos. Nota-se que são produzidos muito mais lotes do que na empresa, entretanto pouquíssimos deles são produzidos com a quantidade máxima permitida (no caso do item uva o lote máximo é de 10.000 litros). A heurística não tenta sincronizar os mesmos itens em $M = 1$ e $M = 2$. Veja que a quantidade do item Uva produzido no tanque preparatório/linha $M = 1$ é muito maior do que a quantidade produzida em $M = 2$. Com a produção dividida em mais lotes com quantidades menores, a heurística juntamente com o modelo MDSL-2E-LT conseguem adequar melhor a sincronia entre os estágios, minimizando assim as esperas e consequentemente as limpezas temporais. Isso pode fazer toda a diferença no aproveitamento da capacidade, como pode ser observado

na solução da instância R3.

Tabela 43 – Programação da produção obtida com a heurística HE2B-DS+FXOV2 para o primeiro período da instância R1.

	Heurística HE2B-DS+FXOV2			
	$M = 1$		$M = 2$	
Sabor	Lotes	Quantidade (Litros)	Lotes	Quantidade (Litros)
Uva	13	6.400	11	5.200
	1	9.600	1	7.224
	13	6.400	1	5.200
	1	8.640	1	7.800
	1	9.280	3	5.200
	12	6.400	2	10.000
	1	9.600	1	5.000
	3	10.000		
	1	8.952		
	5	10.000		
Total	51	369.272	20	118.024
Morango	7	6.400	13	5.200
	1	9280	1	7.800
	5	6400	11	5.200
	1	9900	1	5.420
	2	12000	1	12.000
Total	16	119.980	27	150.020

Tabela 44 – Programação da produção utilizada pela empresa para o primeiro período da instância R1.

	Empresa			
	$M = 1$		$M = 2$	
Sabor	Lotes	Quantidade (Litros)	Lotes	Quantidade (Litros)
Uva	24	10.000	24	10.000
	1	7.296		
Total	25	247.296	24	240.000
Morango	11	12.000	11	12.000
	1	6.000		
Total	12	138.000	11	132.000

O tempo para obtenção de uma programação da produção factível não está detalhado na Tabela 42, mas na empresa o programador da produção demora 2 horas ou mais para obter uma programação da produção factível, enquanto que a heurística FXOV2 depende menos de 1 hora (3.432,27 segundos, em média). Esses resultados ilustram o potencial de aplicação de uma abordagem de otimização para resolver esse problema na prática.

7 Considerações finais

Nesta tese estudou-se o problema de programação da produção de bebidas à base de frutas. Esse setor de bebidas vem crescendo de maneira significativa nos últimos anos, tanto no Brasil quanto mundialmente. Para definir a programação da produção desse processo produtivo, é preciso dimensionar e sequenciar os lotes de produção em dois estágios, levando em consideração a sincronia desses estágios, a existência de *buffers* no segundo estágio (tanques pulmão), e a necessidade de realizar limpezas temporais obrigatórias nos dois estágios e em momentos diferentes para cada um deles. Essas características e restrições específicas desse processo de produção dificultam a elaboração de programas de produção efetivos para o problema.

O problema de programação da produção de bebidas à base de frutas ainda foi pouco abordado na literatura na linha de pesquisa explorada nesta tese. Apesar do grande número de trabalhos que existem na literatura de dimensionamento e sequenciamento de lotes (COPIL et al., 2017), inclusive de bebidas (FERREIRA et al., 2012; BALDO et al., 2014; PAGLIARUSSI; MORABITO; SANTOS, 2017) nenhum trabalho foi encontrado abordando o problema de bebidas à base de frutas com todas as suas características. Para um melhor entendimento do problema, nesta pesquisa foram realizadas visitas técnicas a cinco fábricas do setor (sendo três delas de uma mesma rede multinacional) e diversos dados e informações foram coletados. Com base na literatura, nas observações realizadas nas empresas e nos dados coletados, definiu-se o problema objeto deste estudo e objetivos a serem atingidos de desenvolver e propor uma abordagem de otimização efetiva para tratar esse problema.

Realizou-se uma cuidadosa revisão bibliográfica da literatura relacionada para um melhor entendimento do estado da arte de problemas de dimensionamento e sequenciamento de lotes, suas aplicações em diversos setores, especialmente nas indústrias de bebidas. Com base na descrição do problema e na revisão de literatura, foi desenvolvido um novo modelo, o modelo MDSL-2E-LT, que representa adequadamente as decisões envolvidas neste problema de maneira integrada. Apesar de algumas características de modelagem utilizadas para o MDSL-2E-LT serem baseadas em trabalhos da literatura, o modelo é inovador para tratar apropriadamente as limpezas temporais, os tanques pulmão e gerar programas de produção factíveis para esse processo de produção.

O modelo integrado MDSL-2E-LT, e os métodos de solução propostos nessa tese, foram implementados na linguagem de modelagem AMPL e resolvidos através do *solver* CPLEX. Uma tentativa inicial de melhorar o desempenho do *solver* ao resolver o modelo MDSL-2E-LT foi a modificação de alguns parâmetros do CPLEX para calibrar o método

Branch-and-Cut. Testes computacionais iniciais mostraram a dificuldade do *solver* CPLEX para obtenção de soluções ótimas, e até mesmo de boas soluções, ao resolver as instâncias para o modelo MDSL-2E-LT, inclusive para as de pequeno porte. Assim, além do modelo MDSL-2E-LT, foram também propostos nesta tese métodos de soluções heurísticas para resolver o problema de programação da produção de bebidas à base de frutas.

Foram desenvolvidas heurísticas de decomposição do problema (HE1, HE2 e variações). Decompor o problema e resolvê-lo por estágio foi inspirado na prática da empresa, que programa a produção visando quantos lotes podem ser produzidos nos tanques preparatórios por semana, ou olhando para quantos litros podem ser envasados nas linhas. Nas heurísticas propostas, a decomposição é realizada por estágios e cada estágio é resolvido por meio de um MIP específico. Para essas heurísticas, foram desenvolvidos modelos para os casos de custos e tempos de troca dependentes e independentes da sequência de produção, ambos observados nas situações reais das empresas visitadas. Testes computacionais iniciais mostraram que essas heurísticas são relativamente rápidas para obtenção de soluções factíveis, mas ainda não muito satisfatórias, do ponto de vista de qualidade, especialmente para as instâncias de grande porte.

Devido à dificuldade de solução de instâncias para o MDSL-2E-LT, e sabendo que as restrições de sincronia são onerosas ao modelo, do ponto de vista de dificultar a resolução do CPLEX, foram propostas duas heurísticas (H-SUB e H-SUP) baseadas em modelos relaxados para o problema (MR1-O e MR2-P), com estratégias otimista e pessimista do ponto de vista da capacidade de produção. Essas heurísticas utilizam modelos, baseados no MDSL-2E-LT, porém mais simplificados, que trazem informações dos dois estágios de produção, mas que não estão totalmente sincronizados; logo, no final da resolução de cada modelo é preciso realizar a sincronia e verificar se a solução é factível. Os testes computacionais iniciais realizados com essas estratégias mostraram que os métodos obtêm soluções em tempo mais rápido do que a resolução do MDSL-2E-LT, mas ao mesmo tempo, de forma mais lenta do que as heurísticas de decomposição. A qualidade das soluções é equiparável nessas estratégias com a heurística de decomposição.

Como já mencionado, ao contabilizar os modelos propostos pelas heurísticas de decomposição juntamente com as heurísticas H-SUB e H-SUP foram propostos seis modelos diferentes, utilizados nas diferentes estratégias de solução, considerando cada um dos estágios separadamente e também com informações dos dois estágios, mas sem sincronia. Esses modelos foram resumidos na Tabela 18.

Com o intuito de utilizar o modelo MDSL-2E-LT para resolução do problema, tentou-se resolver esse modelo por partes com as heurísticas *relax-and-fix*. Foram testadas dezesseis estratégias diferentes de escolha e partição de variáveis nas heurísticas *relax-and-fix*. Entretanto, para as instâncias de grande porte, o CPLEX não consegue bom desempenho nem quando a maioria das variáveis estão relaxadas.

Uma vez que as heurísticas de decomposição obtêm soluções razoáveis de maneira rápida, foram exploradas estratégias de melhoria para essas soluções. Implementou-se heurísticas *fix-and-optimize* no modelo MDSL-2E-LT. Foram testadas três soluções iniciais diferentes: a solução dada pelo CPLEX calibrado, a solução obtida com a heurística *relax-and-fix* de melhor desempenho e o melhor método heurístico entre as heurísticas de decomposição e as heurísticas baseadas no modelo relaxado. Para cada uma dessas soluções iniciais, doze estratégias *fix-and-optimize*. Como era esperado, as heurísticas *fix-and-optimize* apresentaram melhorias significativas para as soluções iniciais, sendo que a solução inicial dada pela heurística mais rápida, no caso a HE2B-DS, apresentou os melhores resultados.

Dado o bom desempenho das heurísticas *fix-and-optimize*, propôs-se também heurísticas *fix-and-optimize* com busca em vizinhança. Três estratégias diferentes foram propostas: a FXOV1, em que são testadas três estruturas de vizinhança diferentes; a FXOV2, em que a busca segue os princípios da VNDS, entretanto os conjuntos de variáveis fixadas e liberadas para otimização não seguem os mesmos princípios que a VNDS; e a FXO-VNDS, em que ambos, a busca nas vizinhanças e a escolha dos subproblemas resolvidos, seguem os princípios da metaheurística VNDS.

Para avaliar o desempenho de todos os métodos de solução propostos nesta tese para o problema estudado, utilizou-se instâncias baseadas em dados reais coletados de uma das empresas visitadas. Além disso, com base nesses dados, também foi proposto um gerador de dados para o problema, para geração de instâncias aleatórias realistas, de tal forma que essas instâncias representem o problema estudado.

Os testes foram realizados em duas etapas. Na primeira, foram definidas as estratégias com melhor desempenho para resolver as instâncias baseadas em dados reais. Assim, foram escolhidas as estratégias que apresentaram menor valor de estoques, atrasos, limpezas temporais e trocas. Através das instâncias baseadas em dados reais definiu-se: a melhor heurística de decomposição, a melhor heurística baseada nos modelos relaxados, a melhor heurística *relax-and-fix* e as melhores estratégias *fix-and-optimize*. Uma vez que no total foram propostas 80 estratégias de solução, seria inviável fazer testes computacionais com todos esses métodos para todas as instâncias. Obviamente que nesse tipo de escolha existe uma perda por não haver um teste estatístico que garanta que de fato essas são as melhores estratégias para qualquer tipo de instância. Entretanto, como o objetivo é definir o melhor método de solução para instâncias similares às coletadas, optou-se por definir as melhores estratégias dessa forma. Na segunda etapa foram realizados testes com as melhores estratégias para todas as instâncias (reais e geradas aleatoriamente), com o objetivo de definir o melhor método de solução para o problema de programação da produção de bebidas à base de frutas para as instâncias tratadas nessa tese e comparar a solução obtida com esse melhor método com a solução real da empresa.

Na primeira parte dos testes chegou-se a conclusão que as heurísticas de decomposição são os métodos de solução mais rápidos, e que as MIP-heurísticas são os métodos mais efetivos para encontrar boas soluções para as instâncias reais. As heurísticas H-SUP, HE1B-DS, RFX4, CPX12+FXO12, RFX4+FXO3, HE1B-DS+FXO3, FXOV1-T_M_J, FXOV2 e FXO-VNDS são os métodos que apresentam melhores soluções na resolução dessas instâncias baseadas em dados reais.

Para todas as instâncias (reais e geradas aleatoriamente) os resultados mostram a dificuldade de resolução do modelo MDSL-2E-LT, tanto para a obtenção de boas soluções factíveis, como para a prova de otimalidade de solução devido aos fracos limitantes inferiores. Para instâncias de grande porte, o CPLEX não consegue nem ao menos resolver a relaxação linear.

Na segunda parte dos testes computacionais, considerando todas as instâncias, conclui-se que o método com melhor desempenho é o FXOV2, seguido de FXOV1-T_M_J. Em terceiro lugar há um empate entre os métodos FXO-VNDS e HE1B-DS+FXO3. Os métodos com o pior desempenho são aqueles que utilizam a heurística *relax-and-fix*, RFX4 e RFX4+FXO3. Isso porque o *solver* apresenta dificuldade para resolver os modelos a cada iteração da heurística, devido ao pouco tempo disponível, que as vezes não é suficiente nem para obtenção do limitante inferior. Com relação ao tempo computacional, o método mais rápido é a heurística de decomposição HE1B-DS, seguida da HE1B-DS+FXO3 e da H-SUP. Portanto, em uma situação em que existe tempo disponível para se gerar programas de produção, recomenda-se a utilização da heurística FXOV2 com a solução inicial dada pela heurística HE1B-DS. Para uma situação em que é necessária a obtenção de uma solução rapidamente, indica-se a utilização da heurística HE1B-DS. Para o caso em que o tempo disponível não é muito grande e espera-se uma solução de qualidade razoável, o método recomendado é o HE1B-DS+FXO3.

Ao comparar as soluções obtidas pela heurística HE1B-DS+FXOV2 com a solução da empresa, conclui-se que a heurística apresenta soluções, em média, 33,8% melhores do que a empresa, com relação a quantidade de itens estocados, itens atrasados, número de limpezas temporais e trocas. Percebeu-se que a heurística apresenta programas de produção com lotes de valores muito variados, enquanto que a empresa sempre tenta fazer lotes máximos. O modelo MDSL-2E-LT, utilizado na heurística HE1B-DS+FXOV2, consegue prever que ao fazer lotes menores, menos esperas são geradas e conseqüentemente menos limpezas temporais, melhorando o aproveitamento da capacidade de cada período. Isso pode ser comprovado em uma das instâncias reais testadas, a R3, em que a empresa apresenta uma solução com 12.250 itens atrasados enquanto que a solução obtida pela heurística apresenta somente 4.875 itens em atraso, ou seja, uma redução de 60,20% no número de itens atrasados. Isso mostra a efetividade do método de solução proposto.

Conforme discutido no Capítulo 1, os objetivos propostos nesta tese foram alcan-

çados, ou seja, foram propostos modelos matemáticos e métodos de solução heurísticos para o problema de programação da produção de bebidas à base de frutas. Os resultados mostraram que os métodos são efetivos e têm potencial para resolver problemas em situações reais, gerando programas de produção factíveis e com vantagens sobre as soluções empregadas na prática das empresas.

O desenvolvimento dessa pesquisa gerou até o momento os seguintes trabalhos:

- TOSCANO, A.; FERREIRA, D.; MORABITO, R. Heurística baseada em modelo para resolução do problema de programação da produção de bebidas de frutas. In: Anais do XLVII Simpósio Brasileiro de Pesquisa Operacional, 2015.
- TOSCANO, A.; FERREIRA, D.; MORABITO, R. Modelo de dimensionamento e sequenciamento de lotes dois estágios com limpezas temporais: aplicação na produção de bebidas à base de frutas. In: Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional, 2016.
- TOSCANO, A.; FERREIRA, D.; MORABITO, R. A decomposition heuristic to solve the two-stage lot sizing and scheduling problem with temporal cleaning. Relatório técnico (submetido para publicação), 2017.

Pretende-se ainda escrever mais um artigo científico com a apresentação do modelo MDSL-2E-LT apresentado no Capítulo 4 e dos métodos de solução para resolvê-lo FXOV1-T_M_J, FXOV2 e FXO-VNDS apresentados no Capítulo 5.

Como perspectiva de trabalhos futuros, pretende-se investigar passos de melhoria nas heurísticas de decomposição (HE1 e HE2) e nas heurísticas baseadas em modelos relaxados (H-SUP e H-SUB). O critério de parada para essas heurísticas é um limite de tempo ou a obtenção de uma solução factível. Se uma solução factível inicial é encontrada de maneira rápida, o tempo restante disponível poderia ser utilizado para tentar melhorar a solução inicial. Propostas de melhoria poderiam partir da ideia de ir aumentando a capacidade aos poucos, até um determinado limite.

Dada a dificuldade de resolução do modelo MDSL-2E-LT, uma outra pesquisa futura interessante é propor heurísticas de horizonte rolante para o problema (MOHAMMADI et al., 2010a; RAMEZANIAN; SANAMI; NIKABADI, 2017). A ideia é resolver o problema considerando o modelo MDSL-2E-LT completo, com todas as variáveis e restrições, em somente um período, e para os outros períodos relaxar a exigência de limpezas temporais, ou seja, desconsiderar as restrições de limpezas temporais. Conforme discutido na Seção 6.5.3, o modelo MDSL-2E, que é uma simplificação do modelo MDSL-2E-LT desconsiderando as limpezas temporais, é mais fácil de ser resolvido e pode ser útil nessa linha de pesquisa.

Tendo em vista o sucesso da aplicação das heurísticas *fix-and-optimize* com busca em vizinhança no modelo MDSL-2E-LT, e a tendência na literatura da aplicar princípios de

outras metaheurísticas nas heurísticas *fix-and-optimize* (SEEANNER; ALMADA-LOBO; MEYR, 2013; MOHAMMADI; GHOMI, 2011; CHEN, 2015), outra abordagem de pesquisa que parece promissora é investir na hibridização das estratégias *fix-and-optimize* com outras metaheurísticas, como, por exemplo, com algoritmos genéticos (TOLEDO et al., 2016).

Como os limitantes inferiores obtidos com a resolução da relaxação linear no modelo MDSL-2E-LT não são tão apertados e como o CPLEX encontra dificuldade até mesmo para encontrar boas soluções factíveis para esse problema, propor reformulações para o problema também parece ser uma linha de pesquisa promissora. Pode-se investigar uma reformulação em que os lotes $o \in O_{mt}$ não sejam distribuídos para cada período $m \in M$. Isso poderia diminuir bastante o tamanho do modelo, tendo em vista a influência da cardinalidade do conjunto O_{mt} no número de variáveis e restrições, conforme discutido no Capítulo 4.

Por fim, uma importante pesquisa futura é a validação na prática das abordagens propostas nessa tese. Seria interessante uma abordagem mais iterativa com gerentes de produção de empresas de bebidas à base de frutas para analisar melhor as soluções obtidas pelos métodos aqui propostos, e uma comparação mais efetiva entre essas soluções e as soluções utilizadas pelas empresas. Seria interessante também investigar a aplicação desses métodos em outras indústrias com processos de produção parecidos com o de bebidas à base de frutas, como por exemplo, na indústria de laticínios.

Referências

- ABIR. Associação brasileira das indústrias de refrigerantes e bebidas não alcoólicas - dados de mercado. *Disponível em www.abir.org.br (última visita 05/07/2015).*, 2015. Citado na página 29.
- ALMADA-LOBO, B. et al. Industrial insights into lot sizing and scheduling modeling. *Pesquisa Operacional*, v. 35, n. 3, p. 439–464, 2015. Citado na página 43.
- ALMADA-LOBO, B.; JAMES, R. Neighbourhood search metaheuristics for capacitated lotsizing with sequence-dependent setups. *International Journal of Production Research*, v. 48, p. 861–878, 2010. Citado 2 vezes nas páginas 41 e 210.
- ALMADA-LOBO, B. et al. Single machine multi-product capacitated lot sizing with sequence-dependent setups. *International Journal of Production Research*, v. 45, n. 20, p. 4873–4894, 2007. Citado 3 vezes nas páginas 41, 210 e 211.
- ALMADA-LOBO, B. et al. Multiple machine continuous setup lotsizing with sequence-dependent setups. *Computational Optimization and Applications*, 2010. Citado 3 vezes nas páginas 42, 45 e 213.
- ALMADA-LOBO, B.; OLIVEIRA, J. F.; CARRAVILLA, M. A. A note on “the capacitated lot-sizing and scheduling problem with sequence dependent setup costs and setup times”. *Computers & Operations Research*, v. 35, p. 1374–1376, 2008. Citado na página 39.
- ALMADA-LOBO, B.; OLIVEIRA, J. F.; CARRAVILLA, M. A. Production planning and scheduling in the glass container industry: A VNS approach. *International Journal of Production Economics*, v. 114, p. 363–375, 2008. Citado 4 vezes nas páginas 39, 41, 45 e 211.
- ALMEDER, C.; ALMADA-LOBO, B. Synchronisation of scarce resources for a parallel machine lotsizing problem. *International Journal of Production Research*, v. 49, p. 7315–7335, 2011. Citado 7 vezes nas páginas 39, 41, 44, 45, 47, 71 e 207.
- ALMEDER, C. et al. Lead time considerations for the multi-level capacitated lot-sizing problem. *European Journal of Operational Research*, v. 241, n. 3, p. 727–738, 2015. Citado 6 vezes nas páginas 39, 41, 44, 47, 72 e 210.
- AMORIM, P. et al. Comparing models for lot-sizing and scheduling of single-stage continuous processes: Operations research and process systems engineering approaches. *Computers & Chemical Engineering*, v. 52, n. 177-192, 2013. Citado na página 39.
- ANGEL-BELLO, F. et al. A single machine scheduling problem with availability constraints and sequence-dependent setup costs. *Applied Mathematical Modelling*, v. 35, p. 2041–2050, 2011. Citado na página 46.
- ARAUJO, S. A.; ARENALES, M. N.; CLARK, A. R. Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. *Journal of Heuristics*, v. 13, p. 337–358, 2007. Citado 2 vezes nas páginas 42 e 213.

- ARAÚJO, S. A.; ARENALES, M. N.; CLARK, A. R. Lot-sizing and furnace scheduling in small foundries. *Computers & Operations Research*, v. 35, p. 916–932, 2008. Citado 2 vezes nas páginas 42 e 213.
- ARAÚJO, S. A.; CLARK, A. A priori reformulations for joint rolling-horizon scheduling of materials processing and lot-sizing problem. *Computers & Industrial Engineering*, v. 65, n. 4, p. 577–585, 2013. Citado 2 vezes nas páginas 42 e 213.
- BABAEI, M.; MOHAMMADI, M.; GHOMI, S. M. T. F. A genetic algorithm for the simultaneous lot sizing and scheduling problem in capacitated flow shop with complex setups and backlogging. *International Journal of Advanced Manufacturing Technology*, v. 70, n. 1-4, p. 125–134, 2014. Citado 2 vezes nas páginas 42 e 216.
- BALDO, T. A. *O problema integrado de dimensionamento e sequenciamento de lotes no processo de fabricação da cerveja: modelos e métodos de solução*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo (ICMC-USP), 2014. Citado na página 221.
- BALDO, T. A. et al. An optimization approach for the lot sizing and scheduling problem in the brewery industry. *Computers & Industrial Engineering*, v. 72, p. 58–71, 2014. Citado 12 vezes nas páginas 30, 43, 47, 48, 93, 116, 117, 120, 143, 181, 224 e 227.
- BELO-FILHO, M. A. F.; SANTOS, M. O.; MENESES, C. N. Asynchronous teams for joint lot-sizing and scheduling problem in flow shops. *International Journal of Production Research*, 2012. Citado 2 vezes nas páginas 42 e 216.
- BERTRAND, J.; FRANSOO, J. Modelling and simulation - operations management research methodologies using quantitative modeling. *International Journal of Operations & Productions Management*, v. 22, n. 3, p. 241–264, 2002. Citado na página 34.
- BITRAN, G.; YANASSE, H. Computational complexity of the capacitated lot size problem. *Management Science*, v. 28, p. 1174–1186, 1992. Citado 3 vezes nas páginas 33, 37 e 95.
- BOONMEE, A.; SETHANAN, K. A glnpso for multi-level capacitated lot-sizing and scheduling problem in the poultry industry. *European Journal of Operational Research*, v. 250, n. 2, p. 652–665, 2016. Citado 3 vezes nas páginas 42, 43 e 217.
- BRAHIMI, N. et al. Single item lot sizing problems. *European Journal of Operational Research*, v. 168, p. 1–16, 2006. Citado 2 vezes nas páginas 36 e 38.
- CAMARGO, V. C.; TOLEDO, F. M.; ALMADA-LOBO, B. Three time-based scale formulations for the two-stage lot sizing and scheduling in process industries. *Journal of the Operational Research Society*, v. 63, p. 1613–1630, 2012. Citado 7 vezes nas páginas 39, 41, 43, 44, 47, 71 e 208.
- CAMARGO, V. C.; TOLEDO, F. M.; ALMADA-LOBO, B. Hops - hamming-oriented partition search for production planning in the spinning industry. *European Journal of Operational Research*, v. 234, n. 1, p. 266–277, 2014. Citado 4 vezes nas páginas 41, 46, 120 e 208.
- CHASE, R. B.; JACOB, F. R.; AQUILANO, N. T. *Administração da produção para a vantagem competitiva*. [S.l.: s.n.], 2006. Citado na página 61.

- CHEN, H. Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. *Omega*, v. 56, p. 25–36, 2015. Citado 3 vezes nas páginas [45](#), [120](#) e [186](#).
- CHRISTOU, I.; LAGODIMOS, A.; LYCOPOULOU, D. Hierarchical production planning for multi-product lines in the beverage industry. *Production Planning & Control*, v. 18, n. 5, p. 367–376, 2007. Citado 3 vezes nas páginas [36](#), [47](#) e [222](#).
- CLARK, A.; MAHDIEH, M.; RANGEL, S. Production lot sizing and scheduling with non-triangular sequence-dependent setup times. *International Journal of Production Research*, v. 52, n. 8, p. 2490–2503, 2014. Citado 3 vezes nas páginas [41](#), [211](#) e [212](#).
- CLARK, A. R. Hybrid heuristics for planning lot setups and sizes. *Computers & Industrial Engineering*, v. 45, n. 4, p. 545–562, 2003. Citado 4 vezes nas páginas [48](#), [210](#), [222](#) e [227](#).
- CLARK, A. R. Optimization approximations for capacity constrained material requirements planning. *International Journal of Production Economics*, v. 84, p. 115–131, 2003. Citado na página [41](#).
- CLARK, A. R.; CLARK, S. J. Rolling-horizon lot-sizing when set-up times are sequence-dependent. *International Journal of Production Research*, v. 38, n. 10, p. 2287–2307, 2000. Citado 3 vezes nas páginas [41](#), [210](#) e [215](#).
- COFIL, K. et al. Simultaneous lotsizing and scheduling problems: a classification and review of models. *OR Spectrum*, p. 1–64, 2017. Citado 9 vezes nas páginas [33](#), [39](#), [40](#), [44](#), [45](#), [46](#), [181](#), [213](#) e [214](#).
- CUNHA, C. F. A.; FERREIRA, D. Uma estratégia relax-and-fix na resolução de um problema de dimensionamento e sequenciamento de lotes. In: *46rd Symposium of the Brazilian Operational Research Society (SOBRAPO)*. [S.l.: s.n.], 2014. Citado 2 vezes nas páginas [224](#) e [227](#).
- DANNA, E.; ROTHBERG, E.; PAPE, C. L. Exploring relaxation induced neighborhoods to improve MIP solution. *Mathematical Programming*, v. 102, p. 71–90, 2005. Citado na página [235](#).
- DAUZÈRE-PÉRÈS, S.; LASSERRE, J. B. Integration of lotsizing and scheduling decisions in a job-shop. *European Journal of Operational Research*, v. 75, n. 2, p. 413–426, 1994. Citado 3 vezes nas páginas [42](#), [44](#) e [214](#).
- DEFALQUE, C.; RANGEL, S.; FERREIRA, D. Usando o atsp na modelagem do problema integrado de produção de bebidas. *TEMA - Tendências em Matemática Aplicada e Computacional*, v. 12, n. 3, p. 195–209, 2011. Citado 8 vezes nas páginas [30](#), [39](#), [47](#), [103](#), [212](#), [220](#), [223](#) e [227](#).
- DÍAZ-MADROÑERO, M.; MULA, J.; PEIDRO, D. A review of discrete-time optimization models for tactical production planning. *International Journal of Production Research*, v. 52, n. 17, p. 5171–5205, 2014. Citado na página [38](#).
- DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. *Mathematical Programming*, v. 91, n. 2, p. 201–213, 2002. Citado na página [144](#).

- DREXL, A.; KIMMS, A. Lot sizing and scheduling - survey and extensions. *European Journal of Operational Research*, v. 99, p. 221–235, 1997. Citado 3 vezes nas páginas 38, 39 e 223.
- ELMAGHRABY, S. The economic lot scheduling problem (elsp): Review and extensions. *Management Science*, v. 24, n. 6, p. 587–598, 1978. Citado na página 36.
- EUROMONITOR. *Market Research on Fruit and Vegetable Juice*. [S.l.], 2015. Euromonitor International. Citado na página 29.
- FACHINI, R. F.; ESPOSTO, K. F.; CAMARGO, V. C. B. Glass container production planning with warm-ups and furnace extraction variation losses. 2016. Citado 2 vezes nas páginas 41 e 211.
- FERNANDES, F. C. F.; GODINHO, M. *Planejamento e controle da produção: dos fundamentos ao essencial*. [S.l.]: São Paulo: Editora Atlas S.A., 2010. Citado na página 35.
- FERREIRA, D. et al. Single-stage formulations for synchronised two-stage lot sizing and scheduling in soft drink production. *International Journal of Production Economics*, v. 136, n. 2, p. 255–265, abr. 2012. Citado 10 vezes nas páginas 30, 39, 47, 103, 181, 210, 212, 223, 224 e 227.
- FERREIRA, D.; MORABITO, R.; RANGEL, S. Um modelo de otimização inteira mista e heurísticas relax and fix para a programação da produção de fábricas de refrigerantes de pequeno porte. *Produção*, v. 18, n. 1, p. 76–88, 2008. Citado 5 vezes nas páginas 48, 210, 222, 225 e 227.
- FERREIRA, D.; MORABITO, R.; RANGEL, S. Solution approaches for the soft drink integrated production lot sizing and scheduling problem. *European Journal of Operational Research*, v. 196, p. 697–706, 2009. Citado 14 vezes nas páginas 16, 30, 43, 47, 48, 51, 76, 93, 210, 220, 221, 223, 224 e 227.
- FERREIRA, D.; MORABITO, R.; RANGEL, S. Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants. *Computers and Operation Research*, Elsevier, v. 37, n. 4, p. 684–691, 2010. Citado 11 vezes nas páginas 30, 47, 116, 118, 210, 219, 222, 223, 227, 233 e 234.
- FIGUEIRA, G.; SANTOS, M. O.; ALMADA-LOBO, B. A hybrid vns approach for the short-term production planning and scheduling: A case study in the pulp and paper industry. *Computers & Operations Research*, v. 7, n. 40, p. 1804–1818, 2013. Citado 3 vezes nas páginas 41, 43 e 208.
- FISCHETTI, M.; GLOVER, F.; LODI, A. The feasibility pump. *Mathematical Programming*, v. 104, p. 91–104, 2005. Citado na página 235.
- FLEISCHMANN, B. The discrete lot sizing and scheduling problem. *Management Science*, v. 44, p. 337–348, 1990. Citado na página 38.
- FLEISCHMANN, B. The discrete lot-sizing and scheduling problem with sequence-dependent set-up costs. *European Journal of Operational Research*, v. 75, n. 2, p. 395–404, 1994. Citado na página 39.

- FLEISCHMANN, B.; MEYR, H. The general lotsizing and scheduling problem. *OR Spectrum*, v. 19, p. 11–21, 1997. Citado 3 vezes nas páginas 38, 207 e 222.
- FLORIAN, M.; LENSTRA, J. K.; KAN, A. H. G. R. Deterministic production planning algorithms and complexity. *Management Science*, v. 26, p. 669–679, 1980. Citado na página 37.
- FOURER, R.; GAY, D. M.; KERNIGHAN, B. W. *AMPL - A Modeling Language for Mathematical Programming*. [S.l.: s.n.], 2003. Citado na página 138.
- FURLAN, M. et al. Unequal individual genetic algorithm with intelligent diversification for the lot-scheduling problem in integrated mills using multiple-paper machines. *Computers & Operations Research*, v. 59, p. 33–50, 2015. Citado 4 vezes nas páginas 41, 43, 45 e 209.
- GICQUEL, C.; MINOUX, M. Multi-product valid inequalities for the discrete lot-sizing and scheduling problem. *Computers and Operations Research*, v. 54, p. 12–20, 2015. Citado 2 vezes nas páginas 42 e 214.
- GLOCK, C. H.; GROSSE, E. H.; RIES, J. M. The lot sizing problem: A tertiary study. *International Journal of Production Economics*, v. 155, p. 39–51, 2013. Citado 3 vezes nas páginas 33, 37 e 38.
- GÓMEZ-URRUTIA, E. D.; AGGOUNE, R.; DAUZÈRE-PÉRÈS, S. Solving the integrated lot-sizing and job-shop scheduling problem. *International Journal of Production Research*, v. 52, n. 17, p. 5236–5254, 2014. Citado 3 vezes nas páginas 42, 44 e 215.
- GÖREN, H.; TUNALI, S. Solving the capacitated lot sizing problem with setup carryover using a new sequential hybrid approach. *Applied Intelligence*, v. 42, p. 805–816, 2015. Citado 2 vezes nas páginas 119 e 120.
- GÖTHE-LUNDGREN, M.; LUNDGREN, J. T.; PERSSON, J. A. An optimization model for refinery production scheduling. *International Journal of Production Economics*, v. 78, n. 3, p. 255–270, 2002. Citado 3 vezes nas páginas 42, 43 e 214.
- GUIMARÃES, L.; KLABJAN, D.; ALMADA-LOBO, B. Annual production budget in the beverage industry. *Engineering Applications of Artificial Intelligence*, v. 25, n. 2, p. 229 – 241, 2012. Citado 3 vezes nas páginas 30, 47 e 222.
- GUIMARÃES, L.; KLABJAN, D.; ALMADA-LOBO, B. Pricing, relaxing and fixing under lot sizing and scheduling. *European Journal of Operational Research*, v. 230, n. 2, p. 399–411, 2013. Citado 4 vezes nas páginas 41, 46, 95 e 211.
- GUIMARÃES, L.; KLABJAN, D.; ALMADA-LOBO, B. Modeling lotsizing and scheduling problems with sequence dependent setups. *European Journal of Operational Research*, v. 239, n. 3, p. 644–662, 2014. Citado 3 vezes nas páginas 39, 41 e 212.
- GUNTHER, H. The block planning approach: A case study application from the beverage industry. *2008 IEEE International Conference on Industrial Engineering and Engineering Management*, p. 359–363, 2008. Citado 2 vezes nas páginas 47 e 222.
- GUPTA, D.; MAGNUSSON, T. The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times. *Computers & Operations Research*, v. 32, p. 727–747, 2005. Citado 3 vezes nas páginas 39, 41 e 210.

- HAASE, K. *Lot sizing and scheduling for production planning*. [S.l.]: Berlim, 1994. Citado na página 38.
- HAASE, K. Capacitated lot-sizing with sequence dependent setup costs. *OR Spectrum*, v. 18, p. 51–59, 1996. Citado 4 vezes nas páginas 39, 72, 103 e 210.
- HAASE, K.; KIMMS, A. Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities. *International Journal of Production Economics*, v. 66, n. 2, p. 159–169, 2000. Citado 2 vezes nas páginas 42 e 215.
- HANSEN, P.; MLADENOVIC, N. Variable neighborhood search: principles and applications. 2001. Citado 2 vezes nas páginas 129 e 131.
- HANSEN, P.; MLADENOVIC, N. Variable neighborhood search. In: GLOVER, F.; KOCHENBERGER, G. A. (Ed.). *Handbook of Metaheuristics*. [S.l.: s.n.], 2003. Citado 2 vezes nas páginas 129 e 131.
- HANSEN, P.; MLADENOVIC, N.; PEREZ-BRITOS, D. Variable neighborhood decomposition search. v. 7, p. 335–350, 2001. Citado na página 131.
- HARRIS, F. W. How many parts to make at once. *Factory, The Magazine of Management*, v. 10, n. 2, p. 135–136, 1913. Citado 2 vezes nas páginas 36 e 37.
- HELBER, S.; SAHLING, F. A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, v. 123, p. 247–256, 2010. Citado 2 vezes nas páginas 119 e 125.
- HUTTER, F.; HOOS, H.; LEYTON-BROWN, K. Paramils: An automatic algorithm configuration network. *Journal of Artificial Intelligence Research*, v. 36, p. 267–306, 2009. Citado 2 vezes nas páginas 233 e 236.
- IBGE. Instituto brasileiro de geografia e estatística. *Disponível em www.ibge.gov.br/home (última visita 05/07/2015)*., 2015. Citado na página 29.
- ILOG. *IBM ILOG CPLEX 12.6 reference manual*. 2014. Disponível em: <<http://www.ilog.com/products/cplex/>>. Citado 5 vezes nas páginas 163, 222, 233, 234 e 235.
- JAMES, R. J.; ALMADA-LOBO, B. Single and parallel machine capacitated lotsizing and scheduling: New iterative MIP-based neighborhood search heuristics. *Computers & Operations Research*, v. 38, p. 1816–1825, 2011. Citado 5 vezes nas páginas 41, 120, 126, 208 e 210.
- JANS, R.; DEGRAEVE, Z. An industrial extension of the discrete lot-sizing and scheduling problem. *IIE Transactions (Institute of Industrial Engineers)*, v. 36, n. 1, p. 47–58, 2004. Citado 4 vezes nas páginas 42, 44, 45 e 214.
- JANS, R.; DEGRAEVE, Z. Modeling industrial lot sizing problems: a review. *International Journal of Production Research*, v. 46, n. 6, p. 1619–1643, March 2008. Citado 6 vezes nas páginas 33, 36, 38, 43, 44 e 46.
- JOHNSON, L. A.; MONTGOMERY, D. C. *Operations Research in Production Planning, Scheduling, and Inventory Control*. [S.l.]: John Wiley & Sons, Inc., 1974. Citado na página 35.

- KARIMI, B.; GHOMI, S. F.; WILSON, J. The capacitated lot sizing problem: a review of models and algorithms. *Omega*, v. 31, p. 365–378(14), October 2003. Citado 3 vezes nas páginas 37, 38 e 222.
- KARIMI-NASAB, M.; MODARRES, M. Lot sizing and job shop scheduling with compressible process times: A cut and branch approach. *Computers & Industrial Engineering*, v. 85, p. 196–205, 2015. Citado 3 vezes nas páginas 42, 44 e 217.
- KARIMI-NASAB, M.; MODARRES, M.; SEYEDHOSEINI, S. M. A self-adaptive pso for joint lot sizing and job shop scheduling with compressible process times. *Applied Soft Computing Journal*, v. 27, p. 137–147, 2015. Citado 2 vezes nas páginas 42 e 217.
- KARIMI-NASAB, M.; SEYEDHOSEINI, S. Multi-level lot sizing and job shop scheduling with compressible process times: A cutting plane approach. *European Journal of Operational Research*, 2013. Citado 3 vezes nas páginas 42, 44 e 216.
- KARIMI-NASAB, M. et al. Multi-period lot sizing and job shop scheduling with compressible process times for multilevel product structures. *International Journal of Production Research*, v. 51, n. 20, p. 6229–6246, 2013. Citado 4 vezes nas páginas 42, 44, 45 e 217.
- KARMARKAR, U. S.; SCHRAGE, L. The deterministic dynamic product cycling problem. *Operations Research*, v. 33, p. 326–345, 1985. Citado na página 38.
- KOPANOS, G. M.; PUIGJANER, L.; GEORGIADIS, M. C. Resource-constrained production planning in semicontinuous food industries. *Computers and Chemical Engineering*, v. 35, n. 12, p. 2929–2944, 2011. Citado 2 vezes nas páginas 42 e 216.
- KOPANOS, G. M.; PUIGJANER, L.; MARAVELIAS, C. T. Production planning and scheduling of parallel continuous processes with product families. *Industrial & Engineering Chemistry Research*, v. 50, n. 3, p. 1369–1378, 2011. Citado 2 vezes nas páginas 42 e 216.
- KOVÁCS, A.; BROWN, K. N.; TARIM, S. A. An efficient mip model for the capacitated lot-sizing and scheduling problem with sequence-dependent setups. *International Journal of Production Economics*, v. 118, n. 1, p. 282–291, 2009. Citado 2 vezes nas páginas 42 e 215.
- LANG, J. C.; SHEN, Z. J. M. Fix-and-optimize heuristics for capacitated lot-sizing with sequence-dependent setups and substitutions. *European Journal of Operational Research*, v. 214, n. 3, p. 595–605, 2011. Citado 3 vezes nas páginas 41, 45 e 211.
- LAWLER, E. et al. *The Traveling Salesman Problem*. [S.l.]: John Wiley e Sons Ltd, 1986. Citado na página 39.
- LEITE, R. P. M. *Um Estudo sobre o Problema de Dimensionamento e Sequenciamento da Produção no Setor de Bebidas*. Dissertação (Mestrado) — Departamento de Engenharia de Produção - Universidade Federal de São Carlos (UFSCar), 2008. Citado 5 vezes nas páginas 30, 31, 47, 225 e 227.
- MAHDIEH, M.; CLARK, A.; BIJARI, M. A novel flexible model for lot sizing and scheduling with non-triangular, period overlapping and carryover setups in different machine configuration. 2017. Citado 2 vezes nas páginas 41 e 212.

- MALDONADO, M.; RANGEL, S.; FERREIRA, D. A study of different subsequence elimination strategies for the soft drink production planning. *Journal of Applied Research and Technology*, v. 12, n. 4, p. 1–10, 2014. Citado 7 vezes nas páginas [30](#), [39](#), [45](#), [47](#), [212](#), [223](#) e [227](#).
- MANIEZZO, V.; STÜTZLE, T.; VOSS, S. *Matheuristics - Hybridizing Metaheuristics and Mathematical Programming*. [S.l.]: Springer, 2010. ISBN 978-1-4419-1305-0. Citado na página [46](#).
- MARINELLI, F.; NENNI, M. E.; SFORZA, A. Capacitated lot sizing and scheduling with parallel machines and shared buffers: A case study in a packaging company. *Annals of Operations Research*, v. 150, n. 1, p. 177–192, 2007. Citado 4 vezes nas páginas [42](#), [43](#), [46](#) e [213](#).
- MARTÍNEZ, K. Y.; TOSO, E. A.; MORABITO, R. Production planning in the molded pulp packaging industry. *Computers & Industrial Engineering*, v. 98, p. 554–566, 2016. Citado 2 vezes nas páginas [41](#) e [209](#).
- MATEUS, G. R. et al. Capacitated lot sizing and sequence dependent setup scheduling: an iterative approach for integration. *Journal of Scheduling*, v. 13, n. 3, p. 245–259, 2010. Citado 3 vezes nas páginas [42](#), [46](#) e [215](#).
- MENEZES, G. C.; MATEUS, G. R.; RAVETTI, M. G. A hierarchical approach to solve a production planning and scheduling problem in bulk cargo terminal. *Computers & Industrial Engineering*, v. 97, p. 1–14, 2016. Citado na página [46](#).
- MEYR, H. Simultaneous lotsizing and scheduling on parallel machines. *European Journal of Operational Research*, v. 139, n. 2, p. 277 – 292, 2002. Citado na página [209](#).
- MEYR, H.; MANN, M. A decomposition approach for the general lotsizing and scheduling problem for parallel production lines. *European Journal of Operational Research*, v. 229, n. 3, p. 718–731, 2013. Citado 3 vezes nas páginas [41](#), [46](#) e [209](#).
- MOHAMMADI, M. Integrating lotsizing, loading and scheduling decisions in flexible flow shops. *The International Journal of Advanced Manufacturing Technology*, v. 50, p. 1165–1174, 2010. Citado 2 vezes nas páginas [41](#) e [45](#).
- MOHAMMADI, M.; GHOMI, S. F.; JAFARI, N. A genetic algorithm for simultaneous lotsizing and sequencing of the permutation flow shops with sequence-dependent setups. *International Journal of Computer Integrated Manufacturing*, v. 24, n. 1, p. 87–93, 2011. Citado 2 vezes nas páginas [42](#) e [216](#).
- MOHAMMADI, M.; GHOMI, S. M. T. F. Genetic algorithm-based heuristic for capacitated lotsizing problem in flow shops with sequence-dependent setups. *Expert Systems With Applications*, v. 38, n. 6, p. 7201–7207, 2011. Citado 4 vezes nas páginas [42](#), [45](#), [186](#) e [216](#).
- MOHAMMADI, M. et al. MIP-based heuristics for lotsizing in capacitated pure flow shop with sequence-dependent setup. *International Journal of Production Research*, v. 48, p. 2957–2973, 2010. Citado 4 vezes nas páginas [42](#), [185](#), [215](#) e [216](#).

- MOHAMMADI, M. et al. Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lotsizing problem with sequence-dependent setups. *Journal of Intelligent Manufacturing*, v. 21, n. 4, p. 501–510, 2010. Citado 3 vezes nas páginas 42, 215 e 216.
- MOHAMMADI, M.; POURSAZBI, O. A rolling horizon-based heuristic to solve a multi-level general lot sizing and scheduling problem with multiple machines (MLGLSP_MM) in job shop manufacturing system. *Uncertain Supply Chain Management*, v. 2, n. 3, p. 167–178, 2014. Citado 4 vezes nas páginas 41, 44, 45 e 209.
- MOHAMMADI, M. et al. A new algorithmic approach for capacitated lot-sizing problem in flow shops with sequence-dependent setups. *International Journal of Advanced Manufacturing Technology*, v. 49, n. 1-4, p. 201–211, 2010. Citado 2 vezes nas páginas 42 e 215.
- MORABITO, R.; PUREZA, V. *Modelagem e Simulação (In Metodologia de pesquisa em engenharia de produção e gestão de operações. Miguel, A. M.)*. [S.l.]: Rio de Janeiro, Elsevier: ABEPRO, 2012. v. 2. Citado na página 34.
- MORENO, A.; ALEM, D.; FERREIRA, D. Heuristic approaches for the multiperiod location-transportation problem with reuse vehicles in emergency logistics. *Computers and Operations Research*, v. 69, p. 79–96, 2016. Citado na página 116.
- ÖNCAN, T.; ALTINEL, I. K.; LAPORTE, G. A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, v. 36, n. 3, p. 637–654, 2009. Citado 3 vezes nas páginas 39, 74 e 223.
- PACHECO, J.; BELLO, F. Ángel; ÁLVAREZ, A. A multi-start tabu search method for a single-machine scheduling problem with periodic maintenance and sequence-dependent set-up times. *Journal of Scheduling*, v. 16, n. 6, p. 661–673, 2013. Citado na página 46.
- PAGLIARUSSI, M. *Contribuições para a Otimização da Programação da Produção de Bebidas a Base de Frutas*. Dissertação (Mestrado) — Programa de Pós-graduação em Engenharia de Produção - Universidade Federal de São Carlos (UFSCar), 2013. Citado na página 225.
- PAGLIARUSSI, M.; MORABITO, R.; SANTOS, M. Optimizing the production scheduling of fruit juice beverages using mixed integer programming models. *Gestão & Produção*, v. 24, n. 1, p. 64–77, 2017. Citado 7 vezes nas páginas 30, 31, 47, 109, 181, 225 e 227.
- PINEDO, M. *Scheduling: Theory, Algorithms and Systems*. 3rd edition. ed. [S.l.]: Springer, 2008. Citado na página 31.
- POCHET, Y.; WOLSEY, L. A. *Production Planning by Mixed Integer Programming (Springer Series in Operations Research and Financial Engineering)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387299599. Citado 2 vezes nas páginas 116 e 118.
- QUADT, D.; KUHN, H. Capacitated lot-sizing with extensions: a review. *4OR: A Quarterly Journal of Operations Research*, v. 6, n. 1, p. 61–83, 2008. Citado na página 38.

- RAMEZANIAN, R.; SAIDI-MEHRABAD, M.; FATTAHI, P. Mip formulation and heuristics for multi-stage capacitated lot-sizing and scheduling problem with availability constraints. *Journal of Manufacturing Systems*, v. 32, n. 2, p. 392–401, 2013. Citado 5 vezes nas páginas [42](#), [44](#), [46](#), [47](#) e [217](#).
- RAMEZANIAN, R.; SAIDI-MEHRABAD, M.; TEIMOURY, E. A mathematical model for integrating lot-sizing and scheduling problem in capacitated flow shop environments. *The International Journal of Advanced Manufacturing Technology*, v. 66, n. 1-4, p. 347–361, 2013. Citado 3 vezes nas páginas [42](#), [47](#) e [216](#).
- RAMEZANIAN, R.; SANAMI, S. F.; NIKABADI, M. S. A simultaneous planning of production and scheduling operations in flexible flow shops: case study of tile industry. *The International Journal of Advanced Manufacturing Technology*, v. 88, p. 2389–2403, 2017. Citado 6 vezes nas páginas [39](#), [42](#), [43](#), [72](#), [185](#) e [217](#).
- RANGEL, S.; FERREIRA, D. Um modelo de dimensionamento de lotes para uma fábrica de refrigerantes. *TEMA - Tendências em Matemática Aplicada e Computacional*, v. 2, p. 237–246, 2003. Citado 3 vezes nas páginas [48](#), [222](#) e [227](#).
- RAVETTI, M. G. et al. A scheduling problem with unrelated parallel machines and sequence dependent setups. *International Journal of Operational Research*, v. 2, p. 380–399, 2007. Citado na página [215](#).
- ROBINSON, P.; NARAYANAN, A.; SAHIN, F. Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms. *Omega*, v. 37, n. 1, p. 3–15, 2009. Citado na página [38](#).
- ROHANINEJAD, M.; KHEIRKHAH, A.; FATTAHI, P. Simultaneous lot-sizing and scheduling in flexible job shop problems. *The International Journal of Advanced Manufacturing Technology*, v. 78, n. 1-4, p. 1–18, 2015. Citado 4 vezes nas páginas [41](#), [44](#), [46](#) e [209](#).
- SALOMON, M. et al. Some extensions of the discrete lotsizing and scheduling problem. *Management Science*, v. 37, p. 801–812, 1991. Citado na página [38](#).
- SALOMON, M.; WASSENHOVE L.N.V.; DUMAS, Y.; DAUZÈRE-PÉRÈS, S. Solving the discrete lotsizing and scheduling problem with sequence dependent setup costs and setup times using the traveling salesman problem with time windows. *European Journal of Operational Research*, v. 100, n. 494–513, 1997. Citado 4 vezes nas páginas [39](#), [42](#), [45](#) e [214](#).
- SANTOS, M. O.; ALMADA-LOBO, B. Integrated pulp and paper mill planning and scheduling. *Computers & Industrial Engineering*, v. 63, p. 1–12, 2012. Citado 6 vezes nas páginas [41](#), [43](#), [45](#), [116](#), [208](#) e [209](#).
- SEEANNER, F.; ALMADA-LOBO, B.; MEYR, H. Combining the principles of variable neighborhood decomposition search and the fix&optimize heuristic to solve multi-level lot-sizing and scheduling problems. *Computers & Operations Research*, v. 40, n. 1, p. 303–317, 2013. Citado 11 vezes nas páginas [41](#), [44](#), [45](#), [46](#), [47](#), [120](#), [125](#), [128](#), [133](#), [186](#) e [209](#).

- SEEANNER, F.; MEYR, H. Multi-stage simultaneous lot-sizing and scheduling for flow line production. *OR Spectrum*, v. 35, n. 1, p. 33–73, 2013. Citado 5 vezes nas páginas [41](#), [44](#), [72](#), [95](#) e [209](#).
- SEL, C.; BILGEN, B. Hybrid simulation and mip based heuristic algorithm for the production and distribution planning in the soft drink industry. *Journal of Manufacturing Systems*, v. 33, n. 3, p. 385–399, 2014. Citado 3 vezes nas páginas [30](#), [47](#) e [222](#).
- SLACK, N.; CHAMBERS, S.; JOHNSTON, R. *Administração da Produção*. 3. ed. [S.l.]: São Paulo: Editora Atlas S.A., 2009. Tradução: Henrique Luiz Corrêa. Citado na página [35](#).
- SOUZA, P. S.; TALUKDAR, S. N. Asynchronous organizations for multi-algorithm problem. In: *SAC '93 Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing: states of the art and practice*. [S.l.: s.n.], 1993. p. 286–293. Citado na página [216](#).
- STADTLER, H. Multi-level single machine lot-sizing and scheduling with zero lead times. *European Journal of Operational Research*, v. 209, n. 3, p. 241–252, 2011. Citado 2 vezes nas páginas [42](#) e [212](#).
- STADTLER, H.; SAHLING, F. A lot-sizing and scheduling model for multi-stage flow lines with zero lead times. *European Journal of Operational Research*, v. 225, n. 3, p. 404–419, 2013. Citado 4 vezes nas páginas [42](#), [44](#), [45](#) e [213](#).
- STEINRÜCKEA, M. Integrated production, distribution and scheduling in the aluminium industry: a continuous-time milp model and decomposition model. *International Journal of Production Research*, v. 53, n. 19, p. 5912–5930, 2015. Citado na página [116](#).
- SUERIE, C. *Basic models in lot-sizing - time continuity in discrete time models*. [S.l.]: Springer Berlin Heidelberg (Lecture Notes in Economics and Mathematical Systems), 2005. Citado 2 vezes nas páginas [37](#) e [39](#).
- TEMPELMEIER, H.; BUSCHKÜHL L., L. Dynamic multi-machine lotsizing and sequencing with simultaneous scheduling of a common setup resource. *Or Spectrum*, v. 113, n. 1, p. 401–412, 2008. Citado 3 vezes nas páginas [42](#), [44](#) e [212](#).
- TEMPELMEIER, H.; COPIL, K. Capacitated lot sizing with parallel machines, sequence-dependent setups, and a common setup operator. *OR Spectrum*, 2015. Doi: 10.1007/s00291-015-0410-2. Citado 4 vezes nas páginas [41](#), [44](#), [72](#) e [212](#).
- TETRAPAK. 2015. Última visita em 07 de julho de 2015. Disponível em: www.tetrapak.com. Citado na página [201](#).
- TOLEDO, C. F. M. et al. A relax-and-fix with fix-and-optimize heuristic applied to multi-level lot-sizing problems. *Journal of Heuristics*, v. 21, p. 687–717, 2015. Citado 3 vezes nas páginas [116](#), [119](#) e [120](#).
- TOLEDO, C. F. M.; ARANTES, M. D. O. D. S.; ALMADA-LOBO, B. R. R. R. Glass container production scheduling through hybrid multi-population based evolutionary algorithm. *Applied Soft Computing Journal*, v. 13, n. 3, p. 1352–1364, 2013. Citado 3 vezes nas páginas [42](#), [45](#) e [213](#).

- TOLEDO, C. F. M. et al. Mathematical programming-based approaches for multi-facility glass container production planning. *Computers & Operations Research*, v. 74, p. 92–107, 2016. Citado 4 vezes nas páginas 41, 120, 186 e 211.
- TOLEDO, C. F. M. et al. Um modelo de otimização para o problema integrado de dimensionamento de lotes e programação da produção em fábricas de refrigerantes. *Pesquisa Operacional*, v. 21, n. 1, p. 155–186, 2007. Citado 5 vezes nas páginas 30, 48, 210, 223 e 227.
- TOLEDO, C. F. M. et al. Multi-population genetic algorithm to solve the synchronized and integrated two-level lot sizing and scheduling problem. *International Journal of Production Research*, v. 47, n. 11, p. 3097–3119, 2009. Citado 5 vezes nas páginas 30, 48, 210, 223 e 227.
- TOLEDO, C. F. M. et al. The synchronized and integrated two-level lot sizing and scheduling problem: Evaluating the generalized mathematical model. *Mathematical Problems in Engineering*, v. 2015, p. 1–18, 2015. Citado 6 vezes nas páginas 30, 43, 47, 210, 223 e 227.
- TOLEDO, C. F. M. et al. A genetic algorithm/mathematical programming approach to solve a two-level soft drink production problem. *Computers & Operations Research*, v. 48, p. 40–52, 2014. Citado 5 vezes nas páginas 30, 48, 210, 224 e 227.
- TOLEDO, F. M. B.; ARMENTANO, V. A. A lagrangean-based heuristic for the capacitated lot-sizing problem in parallel machines. *European Journal of Operational Research*, v. 175, p. 1070–1083, 2006. Citado na página 143.
- TOSCANO, A.; FERREIRA, D.; MORABITO, R. Heurística baseada em modelo para resolução do problema de programação da produção de bebidas de frutas. In: *Anais do XLVII Simpósio Brasileiro de Pesquisa Operacional*. [S.l.: s.n.], 2015. Citado 3 vezes nas páginas 102, 137 e 237.
- TOSCANO, A.; FERREIRA, D.; MORABITO, R. Modelo de dimensionamento e sequenciamento de lotes dois estágios com limpezas temporais: aplicação na produção de bebidas à base de frutas. In: *Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional*. [S.l.: s.n.], 2016. Citado na página 84.
- TOSCANO, A.; FERREIRA, D.; MORABITO, R. A decomposition heuristic to solve the two-stage lot sizing and scheduling problem with temporal cleaning. *Flexible Service and Manufacturing Journal*, 2017. Citado 3 vezes nas páginas 102, 137 e 237.
- TOSO, E.; MORABITO, R.; CLARK, A. R. Lotsizing and sequencing optimisation at an animal-feed plant. *Computer & Industrial Engineering*, v. 57, p. 813–821, 2009. Citado 3 vezes nas páginas 39, 41 e 207.
- TRANSCHEL, S. et al. A hybrid general lot-sizing and scheduling formulation for a production process with a two-stage product structure. *International Journal of Production Research*, v. 49, n. 9, p. 2463–2480, 2011. Citado 2 vezes nas páginas 41 e 207.
- TRIGEIRO, W. W.; THOMAS, L. J.; MCCLAIN, J. O. Capacitated lot sizing with setup times. *Management Science*, v. 35, p. 353–366, 1989. Citado 2 vezes nas páginas 37 e 100.

- VANDERBECK, F. Lot-sizing with start-up times. *Management Science*, v. 44, n. 10, p. 1409–1425, 1998. Citado 3 vezes nas páginas 42, 45 e 213.
- VENTURINI, W. *Bebidas - Bebidas Não Alcoólicas - Ciência e Tecnologia*. [S.l.]: Blucher, 2010. Citado 6 vezes nas páginas 15, 29, 51, 52, 53 e 54.
- WAGNER, H.; WHITIN, T. Dynamic version of the economic lot size model. *Management Science*, v. 5, p. 89–96, 1958. Citado na página 37.
- WOLOSEWICZ, C.; DAUZÈRE-PÉRÈS, S.; AGGOUNE, R. A lagrangian heuristic for an integrated lot-sizing and fixed scheduling problem. *European Journal of Operational Research*, v. 244, p. 3–12, 2015. Citado 3 vezes nas páginas 42, 44 e 215.
- WOLSEY, L. A. *Integer Programming*. [S.l.]: John Wiley & Sons, Inc., 1998. Citado na página 222.
- XIAO, J. et al. A hybrid lagrangian-simulated annealing-based heuristic for the parallel-machine capacitated lot-sizing and scheduling problem with sequence-dependent setup times. *Computers & Operations Research*, v. 63, p. 72–82, 2015. Citado 4 vezes nas páginas 41, 46, 95 e 211.
- XIAO, J. et al. MIP-based fix-and-optimize algorithms for the parallel machine capacitated lot-sizing and scheduling problem. *International Journal of Production Research*, v. 51, p. 5011–5028, 2013. Citado 3 vezes nas páginas 41, 45 e 211.
- YU, X.; ZHANG, Y.; STEINER, G. Single-machine scheduling with periodic maintenance to minimize makespan revisited. *Journal of Scheduling*, v. 17, p. 263–270, 2014. Citado na página 46.
- ZANAKIS, S. H.; EVANS, J. R.; VAZACOPOULOS, A. A. Theory and methodology heuristic methods and applications : A categorized survey. v. 43, p. 88–110, 1989. Citado na página 95.
- ZIPKIN, P. Computing optimal lot sizes in the economic lot scheduling problem. *Operations Research*, v. 39, n. 1, p. 56–63, 1991. Citado na página 36.

APÊNDICE A – Fábrica A

A fábrica de bebidas Fábrica A é uma franquia de uma companhia de grande porte que possui outras fábricas no Brasil e no mundo. As bebidas fabricadas nessa indústria são sucos, bebidas mistas de frutas, energéticos, isotônicos, chás e refrigerantes, sendo que ela é responsável por cerca **70%** da produção nacional de uma bebida mista de frutas destinada à crianças. A fábrica A localiza-se no interior do estado de São Paulo e é referência nacional em tecnologia e qualidade. A fábrica possui 9 tanques e 8 linhas de produção para o atendimento de toda sua demanda.

A produção de bebidas mistas de frutas (refrescos) acontece em uma das linhas de produção desta fábrica, denotada por Linha 1. O processo de envase é asséptico e a bebida é envasada em embalagens cartonadas *TetraPak* de 200ml (TETRAPAK, 2015), podendo ser encontrada em cinco sabores: laranja, maracujá, abacaxi, uva e morango. Um produto final é considerado um fardo com 12 embalagens de 200ml.

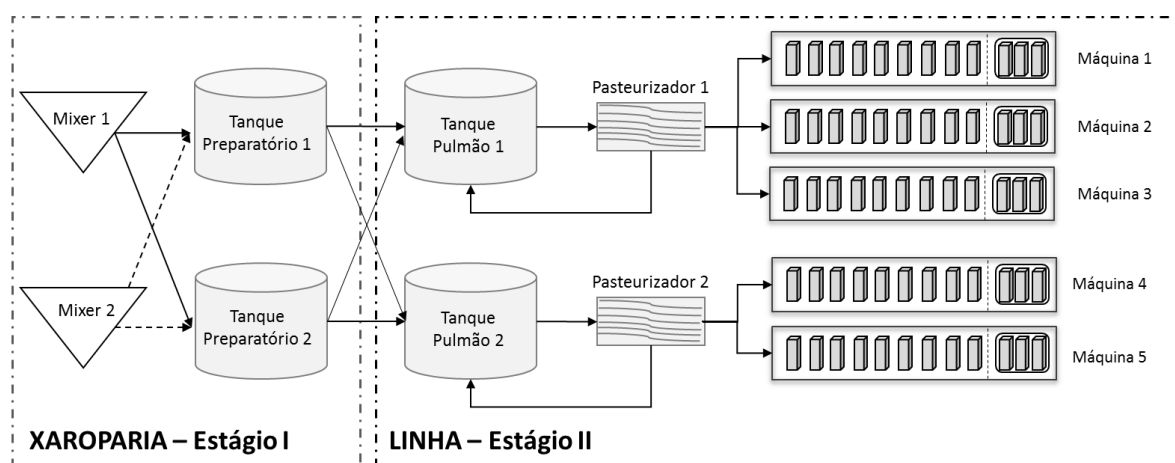


Figura 44 – Processo de produção de bebida mista de fruta. Fonte: a autora.

A Figura 44 é uma representação do processo produtivo de refrescos dessa fábrica. Na Xaroparia existe um mixer e dois tanques preparatórios dedicados somente ao preparo das bebidas à base de frutas. Algumas vezes um segundo mixer é utilizado para preparo do suco. Na Linha observada há dois tanques pulmões, dois pasteurizadores e cinco máquinas de envase. Assim como foi descrito na Seção 3.1, as máquinas de 1 a 5 podem ser ligadas a qualquer pasteurizador desde que a capacidade do pasteurizador não seja excedida. Uma prática adotada pela Fábrica A é deixar sempre ligadas as máquinas 5, 3 e 2 ao pasteurizador 1 e as máquinas 1 e 4 ao pasteurizador 2.

Os *kits* para produção da bebida são comprados da matriz da franquia. Os lotes de bebidas são produzidos em frações múltiplas de 25% do *kit*. Como a capacidade física dos

tanques preparatórios e pulmões é de 12.000 litros, a Tabela 45 apresenta a quantidade em litros que é possível produzir com 1 *kit* para cada sabor da bebida, e os lotes máximo e mínimo requeridos de cada sabor, respeitando as proporções dos *kits* e a capacidade do tanque preparatório. Por exemplo, para fazer um lote máximo de suco sabor laranja basta utilizar 100% de um kit, que já equivale a um tanque cheio de 12.000 litros. Já para fazer um lote máximo de sabor uva é preciso utilizar 50% de um kit, o que equivale a 10.000 litros. Não é possível preparar em uma única batelada mais do que 10.000 litros, pois para isso seria necessário utilizar 75% de um kit, que corresponderia a 15.000 litros de bebida, ultrapassando a capacidade do tanque. Da mesma forma, os lotes mínimos também variam de acordo com o sabor, pois os lotes mínimos de todos os sabores são sempre correspondentes à 25% de um *kit*.

Tabela 45 – Descrição dos valores em litros dos *kits* e dos lotes máximo e mínimo.

Sabor	<i>Kit</i> (Litros)	Lote Máximo (% de <i>Kit</i>)	Lote Máximo (Litros)	Lote Mínimo (Litros - 25% do <i>Kit</i>)
Laranja	12.000	100%	12.000	3.000
Maracujá	24.000	50%	12.000	6.000
Abacaxi	24.000	50%	12.000	6.000
Uva	20.000	50%	10.000	5.000
Morango	16.000	75%	12.000	4.000

O tempo de preparo dos lotes independente do sabor e quantidade é de 1h e 30 minutos. Considerando que o tempo de transferência para o tanque pulmão é de aproximadamente 10 minutos, considera-se como tempo de preparo 100 minutos. As capacidades de produção dos pasteurizadores nas unidades de medidas: litros por hora, embalagens de 200ml por hora e produto final (caixas com 12 embalagens de 200ml) por hora, estão descritas na Tabela 46.

Tabela 46 – Velocidade de produção dos pasteurizadores.

Pasteurizadores	Litros/h	Emb. 200ml/h	Fardos/h
Pasteurizador A	4.000	20.000	1.666,67
Pasteurizador B	7.500	37.500	3.215

No período em que as primeiras visitas foram realizadas à fábrica, haviam cinco máquinas de envase, sendo três máquinas mais novas e sofisticadas e duas mais antigas. Atualmente a fábrica opera com quatro máquinas, as duas máquinas antigas foram substituídas por uma máquina mais nova. As capacidades nominais das cinco máquinas encontradas nas primeiras visitas e das máquinas utilizadas atualmente estão apresentadas nas Tabelas 47 e 48, respectivamente. As velocidades estão apresentadas em litros por

hora, embalagens de 200ml por hora e produto final por hora (caixas com 12 embalagens de 200ml).

Tabela 47 – Velocidades nominais das máquinas de envase na primeira visita realizada a fábrica.

Máquinas	Litros/h	Emb. 200ml/h	Fardos/h
Máquina 1	1.200	6.000	500
Máquina 2	2.260	11.300	942
Máquina 3	2.260	11.300	942
Máquina 4	1.500	7.500	625
Máquina 5	2.260	11.300	942

Tabela 48 – Velocidades nominais das máquinas de envase na última visita realizada a fábrica.

Máquinas	Litros/h	Emb. 200ml/h	Fardos/h
Máquina 1	2.260	11.300	942
Máquina 2	2.260	11.300	942
Máquina 3	2.260	11.300	942
Máquina 4	2.260	11.300	942

Para programar a produção, são consideradas as velocidades operacionais das máquinas que variam em torno de 75% a 90% da capacidade original. Na configuração antiga das máquinas (Tabela 47), a estratégia adotada pela empresa era de ligar as máquinas 2 e 4 ao pasteurizador 1 e as outras máquinas (1, 3 e 5) ao pasteurizador 2. Atualmente (Tabela 48) a máquina 1 é ligada ao pasteurizador 1 e as outras máquinas, 2, 3 e 4 são ligadas ao pasteurizador 2.

É possível notar que, em ambas as situações (Tabelas 47 e 48), existe capacidade ociosa nos pasteurizadores, pois somando as capacidades de todas as máquinas temos 9.480 litros/hora para as máquinas da Tabela 47 e 9.040 litros/hora na Tabela 48, ao invés de 11.500 litros/hora dos dois pasteurizadores. Ou seja, o ritmo de produção é puxado pela velocidade das máquinas. Nesse caso, ainda poderiam ser adicionadas mais máquinas à essa linha de produção devido à capacidade ociosa dos pasteurizadores.

O CIP da xaroparia demora em torno de 50 minutos e o CIP da linha 5 horas. As limpezas temporais acontecem a cada $TP_{max} = 24$ horas na xaroparia e a cada $TL_{max} = 48$ horas na linha. Uma característica interessante da limpeza temporal, tanto da xaroparia quanto da linha, é que a validade de 24hs e 48hs, respectivamente, começa a ser contada a partir do término da limpeza. Isto é, independentemente de logo após terminar o CIP houver ou não produção, o tempo de validade do CIP já começa a ser contado. Por exemplo, suponha que no início do período a linha já esteja pronta para receber o suco, mas a

xaroparia ainda não tenha terminado de prepará-lo. Nesse caso a linha de produção fica parada aguardando o recebimento do suco e esse tempo de espera já começa a ser contado no tempo de validade do CIP. Cabe lembrar que as máquinas de envase podem esperar no máximo 40 minutos para para iniciar o envase, caso contrário elas devem ser novamente limpas (CIP).

Programação da Produção

A programação da produção da Fábrica A é feita por um funcionário do setor de PCP (Planejamento e Controle da Produção). Essa programação é realizada todas as segundas-feiras com base nos dados da demanda da semana e do estoque. As franquias do mesmo tipo da Fábrica A possuem um *joint venture* (denotamos por Empresa L) responsável por comercializar todas as bebidas não carbonatadas dessa rede de franquias. Sendo assim, a Fábrica A não comercializa os seus produtos não carbonatados diretamente com os clientes, mas produz para atender parte da demanda da Empresa L, que negocia e distribui essas bebidas para clientes de todo o Brasil. O escoamento da produção para os centros de distribuição da Empresa L acontece todos os dias, sendo o estoque da bebida de frutas muito pequeno dentro da Fábrica A. Logo, a programação da produção da bebida mista de fruta é feita *make to order* e o programador não tem como principal objetivo controlar o estoque, mas sim atender a demanda existente.

Existe um acordo contratual entre a Fábrica A e a Empresa L, em que a Fábrica A tem que sempre atender determinada porcentagem de demanda da Empresa L. Em contrapartida a Empresa L sempre garante um mínimo de demanda mensal para a Fábrica A. Caso alguma das partes descumpra o contrato, esta tem que pagar altas multas. Em todo início de mês o programador recebe da Empresa L a previsão de demanda mensal das bebidas de frutas. Com base nessa previsão, o programador realiza a programação da produção mensal, porém todas as segundas-feiras essa programação acaba recebendo ajustes para se adequar a realidade dos dados da Empresa L.

As bebidas carbonatadas (por exemplo, refrigerantes) são negociadas diretamente com os clientes e a produção dessas bebidas é realizada *make to stock*. Para essas bebidas, existe uma previsão de demanda de 6 meses. Com base nessa previsão, o funcionário sempre faz a programação para quatro semanas, mas em cada semana é necessário fazer o ajuste da programação da produção com base na demanda real para a semana. Toda a programação da produção dessa fábrica é realizada em planilhas eletrônicas. Para fazer a programação, o programador se baseia nos históricos de dados da empresa, nas demandas da semana e na sua experiência. Para a programação das linhas de bebidas carbonatadas, o programador demora, em média, de 4 a 5 horas para realizar a programação da semana. Já para as bebidas não carbonatadas, mais especificamente para a linha de bebidas de frutas, o operador leva 2 horas para fazer a programação mensal. Caso aconteça algum imprevisto durante a semana e a programação precise ser refeita, o mesmo tempo é gasto

para reprogramar a produção da semana.

Depois que a programação está pronta, esta é enviada ao MRP da empresa que se liga ao SAP para gerar os pedidos de compras de matéria-prima. Todas as linhas da Fábrica A operam 7 dias por semana, 24 horas por dia, exceto a Linha 1, aqui estudada, que também funciona 24 horas por dia, mas apenas 6 dias por semana. Assim, o número de horas de capacidade de produção que o programador utiliza para programação dessa linha é de 142 horas ($5 \text{ dias} \times 24\text{hs} + 1 \text{ dia} \times 22\text{hs}$, pois nos sábados os funcionários vão embora as 22h).

APÊNDICE B – Trabalhos sobre dimensionamento e sequenciamento de lotes

Nesta seção são discutidos em mais detalhes os trabalhos sobre dimensionamento e sequenciamento de lotes abordados na revisão de literatura apresentada no Capítulo 2. Assim como na Tabela 1 em que esses trabalhos estão sumarizados, estes serão discutidos de acordo com o tipo de modelagem do qual foram derivados na seguinte ordem: GLSP, CLSD, CSLP, DSLP e OUTROS.

Trabalhos baseados no GLSP

Desde a apresentação do GLSP por [Fleischmann e Meyr \(1997\)](#) muitos trabalhos surgiram considerando diversas variações baseadas em aplicações reais, por exemplo, à partir desse modelo. [Toso, Morabito e Clark \(2009\)](#) apresentam dois modelos para o problema de planejamento da produção de ração animal. Os modelos diferem-se pela maneira de considerar o *setup*: dependente ou independente da sequência. Nesse problema a desigualdade triangular não precisa ser satisfeita pois existem produtos considerados como limpantes (a desigualdade triangular é satisfeita quando realizar uma troca de um produto i para um produto j é menos oneroso do que trocar de i para k e depois de k para j). Os estágios de produção são serial, mas somente um estágio (gargalo) é considerado na modelagem. Para resolver o problema os autores utilizam o CPLEX e diferentes estratégias *relax-and-fix*.

[Almeder e Almada-Lobo \(2011\)](#) consideram uma situação em que existem máquinas em paralelo não relacionadas que precisam de uma ferramenta (ou um tipo de recurso) para produzir. Essa ferramenta deve ser compartilhada, porém não pode ser utilizada por mais de duas máquinas ao mesmo tempo, logo sua utilização deve sincronizada entre as máquinas. Assim, trocas de ferramentas são modeladas ao invés de trocas de produtos. Os autores propõem dois modelos baseados no GLSP e no CLSD. Ambos utilizam variáveis contínuas de controle de tempo. Os autores mostram que o modelo baseado no CLSD é um caso particular do modelo baseado no GLSP e que o primeiro tem um desempenho computacional melhor. Esse tipo de característica de produção é encontrada na indústria de semicondutores.

Inspirados no caso real de uma indústria de processo, [Transchel et al. \(2011\)](#) formulam um modelo considerando uma máquina produzindo diversos itens finais vindo de múltiplos pré-produtos. Trata-se de uma produção em dois estágios, porém o modelo leva em conta somente o primeiro estágio considerando que o primeiro estágio fornece pré-produtos em quantidade limitadas. Dado os limitantes fracos obtidos com o modelo

baseado no GLSP, os autores propõem duas reformulações baseadas no problema do transporte e compara os três modelos.

O problema tratado por [Camargo, Toledo e Almada-Lobo \(2012\)](#) tem dois estágios de produção. Em um primeiro estágio uma única máquina produz um pré-produto que alimentará o segundo estágio. Os produtos finais são produzidos no segundo estágio em máquinas paralelas e não idênticas. Cada produto final requer um pré-produto. Embora um pré-produto possa ser usado para diferentes produtos finais, em cada micro-período um único pré-produto pode ser produzido e todas as máquinas do segundo nível são alimentadas por esse pré-produto. Os autores apresentam três modelos para esse problema. Um modelo é baseado na estrutura de subperíodos do GLSP, o segundo modelo é baseado no CLSD e um terceiro modelo baseado em uma formulação alternativa que desconsidera a divisão do horizonte de planejamento. Os três modelos utilizam variáveis contínuas de controle do tempo que indicam os inícios e finais dos lotes e do recurso comum. Essas variáveis são utilizadas para sincronizar os dois estágios. Testes computacionais são realizados com instâncias geradas a partir de dados reais de uma empresa de fiação. O modelo baseado no CLSD obtém as melhores soluções e o modelo alternativo é mais flexível para incorporar restrições operacionais adicionais. Em [Camargo, Toledo e Almada-Lobo \(2014\)](#) os autores fazem uma formulação adaptada do modelo baseado no GLSP de [Camargo, Toledo e Almada-Lobo \(2012\)](#) para torná-lo mais forte. Para resolver o problema os autores propõem um novo método híbrido denominado HOPS (*Hamming-Oriented Partition Search*). Trata-se de um procedimento que utiliza *branch-and-bound* em combinação com heurísticas de melhoria do tipo *fix-and-optimize*. A partição em que as variáveis serão fixadas são escolhidas baseadas nas distâncias de Hamming.

[Santos e Almada-Lobo \(2012\)](#) propõem um modelo para o planejamento e programação da produção da indústria de celulose e papel encontrado em uma situação real. Trata-se de um processo de produção em dois estágios em que pedaços de madeira são processados em um digestor resultando na própria celulose virgem e em um fluxo de resíduos conhecido como licor negro. Esse licor negro é utilizado para produzir energia elétrica e a celulose virgem. Esta última é combinada com celulose reciclada para produzir diversos tipos de papeis. O gargalo de produção não é definido e pode variar entre esses dois processos. Diversas características específicas do processo produtivo são levadas em consideração na modelagem. Para resolver o modelo os autores utilizam uma estratégia *fix-and-optimize* estocástica, apresentada por [James e Almada-Lobo \(2011\)](#). [Figueira, Santos e Almada-Lobo \(2013\)](#) propõem um novo método de solução para o problema descrito por [Santos e Almada-Lobo \(2012\)](#). O método determina padrões de *setup* via uma heurística do tipo VNS (*variable neighborhood search*). As variáveis que definem esses padrões são fixadas no modelo, e as variáveis contínuas remanescentes são calculadas através de um método exato. Testes computacionais são realizados com dados reais e os resultados mostram que o método tem desempenho melhor do que métodos exatos e

MIP-heurísticas para resolver instâncias reais. [Furlan et al. \(2015\)](#) estende o modelo de [Santos e Almada-Lobo \(2012\)](#) para o caso de máquinas paralelas não idênticas e utiliza algoritmo genético como método de solução.

[Martínez, Toso e Morabito \(2016\)](#) estudam o problema de planejamento da produção em indústrias de embalagens à base de polpa moldada. As embalagens são processadas através de uma máquina de moldagem das embalagens. Vários moldes diferentes e várias combinações de moldes podem ser utilizados pela máquina de moldagem. A produção de um *mix* de produtos depende da combinação de moldes que estão conectados na máquina de molde. Os autores propõem um modelo, baseado no GLSP, em que, além das decisões de dimensionamento e sequenciamento dos lotes, são tomadas decisões de quais moldes serão utilizados. Instâncias reais são resolvidas via CPLEX.

[Meyr e Mann \(2013\)](#) apresentam uma nova solução heurística para o GLSP considerando máquinas paralelas *General Lotsizing and Scheduling Problem for Parallel production Lines* (GLSPPL) publicado por [Meyr \(2002\)](#). O método heurístico decompõe, de maneira iterativa, o problema multimáquinas em um conjunto de problemas com uma única máquina, que são mais fáceis de serem resolvidos. Se alguma solução é infactível com relação à capacidade original do problema, as capacidades de produção do problema multimáquinas são alteradas para a próxima iteração. Diferentes abordagens são propostas para decomposição e para iteração entre o problema geral multimáquinas e os problemas monomáquinas. Dessa forma, diferentes tipos de problemas reais são resolvidos. Um modelo para o GLSP considerando multiníveis de produção em um sistema de produção *flow-shop* e a sincronia entre as máquinas é apresentado em [Seeanner e Meyr \(2013\)](#). Diferentes reformulações e MIP-heurísticas *relax-and-fix* e *fix-and-optimize* são comparadas. A estrutura de controle de tempo no GLSP multiestágio é a formulação monoestágio, entretanto o início dos microperíodos é idêntico em todos os estágios. Essa estrutura de modelagem facilita o controle da sincronia e da relação entre os itens sucessores e predecessores. [Seeanner, Almada-Lobo e Meyr \(2013\)](#) fazem pequenas modificações do modelo de [Seeanner e Meyr \(2013\)](#) para permitir trocas no primeiro microperíodo e propõe uma formulação com menos variáveis. A sincronia entre os estágios é formulada de maneira mais geral e compacta. Os autores propõem uma combinação de uma heurística do tipo *Variable Neighborhood Decomposition Search* (VNDS) com a *fix-and-optimize*.

Uma formulação do tipo GLSP multiestágio para um sistema de produção do tipo *job-shop* é proposta por [Rohaninejad, Kheirkhah e Fattahi \(2015\)](#). Diferentes estágios de produção são sincronizados usando variáveis que controlam o início e final dos microperíodos. Um algoritmo que combina algoritmos genéticos e otimização de enxame de partículas (*particle swarm optimization*) é usado para resolver o problema. [Mohammadi e Poursabzi \(2014\)](#) também apresentam um modelo do tipo GLSP multiestágio para sistemas *job-shop*. Heurísticas de horizonte rolante são utilizadas para resolver o modelo.

Trabalhos com aplicação na indústria de bebidas que utilizam na modelagem a estrutura do tipo GLSP são: Toledo et al. (2007), Ferreira, Morabito e Rangel (2008), Toledo et al. (2009), Ferreira, Morabito e Rangel (2009), Ferreira, Morabito e Rangel (2010), Ferreira et al. (2012), Toledo et al. (2014) e Toledo et al. (2015). Esses trabalhos são melhores discutidos no Apêndice C.

Trabalhos baseados no CLSD

Diversos trabalhos considerando abordagens de modelagem do tipo CLSD são encontradas desde o trabalho de Haase (1996) até os dias de hoje. Clark e Clark (2000) estudam o problema de dimensionamento e sequenciamento único nível e com máquinas em paralelo. Para lidar com o sequenciamento os autores admitem que existe um número pré-definido de *setups* em cada período, e uma variável binária controla a ocorrência de cada *setup*. Itens fantasmas, cujo custo de *setup* nulo, são utilizados quando ocorrem menos *setups* do que o número pré-definido. MIP-heurísticas do tipo horizonte rolante e *relax-and-fix* são utilizadas. Clark (2003a) estendem o problema para múltiplos estágios e para produtos com estrutura de montagem multinível. Almeder et al. (2015) abordam um problema similar ao de Clark (2003a), considerando o *setup* dependente da sequência. O diferencial do trabalho é a consideração de necessidades de recursos e precedência entre os itens. Se essas restrições não são levadas em consideração, podem ser gerados planos de produção ineficazes ou até mesmo estoques desnecessários. Para lidar com essas questões os autores consideram a sincronia entre os estágios. São utilizadas variáveis de controle de tempo e duas formulações para o problema são propostas: a primeira garante que a produção de um item não pode ser iniciada enquanto o lote de produção de seu antecessor não for finalizado. A segunda abordagem é do tipo *lot streaming*, em que o item pode ser iniciado antes da finalização do lotes do seu antecessor.

Gupta e Magnusson (2005) apresentam um modelo monomáquina para uma fábrica de lixa. Os autores consideram que os *setups* podem ser carregados de um período para o outro e que esses são mantidos em períodos ociosos. Uma heurística composta por fases de inicialização de uma solução factível, sequenciamento e melhoria da solução encontrada é proposta.

Almada-Lobo et al. (2007) apresenta uma formulação alternativa para o CLSD baseado em um problema real de produção de recipientes de vidro. Subrotas desconexas são permitidas nesse problema, ou seja, um item pode ser produzido mais de uma vez por período. Eles mostram que o modelo apresenta menos variáveis binárias que os modelos de Clark e Clark (2000) e Gupta e Magnusson (2005). Além disso, apresentam uma formulação mais compacta para o problema e desenvolvem inequações válidas. Uma heurística composta de passos de construção dos lotes, sequenciamento e melhoria é apresentada. Almada-Lobo e James (2010) apresentam duas metaheurísticas para o problema: VNS e Busca tabu. James e Almada-Lobo (2011) estendem o modelo para máquinas paralelas e propõem um

método que combina metaheurísticas de busca local, INS (*Iterative Variable Neighborhood Search*) com MIP-heurísticas do tipo *relax-and-fix*. O modelo apresentado por [Almada-Lobo et al. \(2007\)](#) é estendido em [Almada-Lobo, Oliveira e Carravilla \(2008b\)](#) para considerar que diversos fornos podem transferir vidro derretido para diversas máquinas em paralelo, que fazem os recipientes de vidro. Além do modelo, heurísticas do tipo VNS com buscas em vizinhança realizadas de forma estocástica e determinística são utilizadas. [Toledo et al. \(2016\)](#) tratam o planejamento da produção da indústria de vidros considerando multi-instalações. Os autores propõem um modelo, heurísticas *fix-and-optimize* e algoritmo genético para resolver o problema. Uma variação do problema de programação da produção da indústria de recipientes de vidro é proposta por [Fachini, Esposito e Camargo \(2016\)](#). Os autores consideram no modelo o reaquecimento dos fornos entre a troca de produtos e a influência dos *setups* na quantidade produzida pelos fornos. O intuito é analisar a influência dessas restrições tecnológicas na solução. Para resolver o problema os autores propõem uma VNS e fazem um estudo de caso em uma empresa do setor.

[Lang e Shen \(2011\)](#) consideram o caso real de uma fábrica de folhas de plástico que compõem para-brisas de carros. Os autores propõem um modelo do tipo CLSD monoestágio e utilizam MIP-heurísticas *relax-and-fix* e *fix-and-optimize* para resolvê-lo.

[Guimarães, Klabjan e Almada-Lobo \(2013\)](#) também tratam o problema monoestágio. A formulação proposta admite que um conjunto pré-definido de sequências está disponível. A partir desse conjunto o modelo determina a sequência de produção que será utilizada e calcula o tamanho dos lotes com base nessa escolha. Sequências que não respeitam a desigualdade triangular são possíveis. Os autores propõem uma heurística para construir as sequências. Heurísticas do tipo horizonte rolante e *relax-and-fix* para gerar soluções factíveis. Para melhoria dessas soluções são utilizadas heurísticas do tipo *fix-and-optimize*.

Motivados pelo problema de dimensionamento e sequenciamento de lotes em empresas de manufatura de semicondutores, [Xiao et al. \(2013\)](#) propõem um modelo com um único estágio de produção, considerado gargalo, nesse estágio existem máquinas paralelas heterogêneas. Dada as especificidades da produção são consideradas janelas de tempo para realização de determinados *setups*, ou seja, para a produção de determinados itens. Heurísticas *relax-and-fix* e *fix-and-optimize* são utilizadas para resolver problemas reais em tempos computacionais aceitáveis. [Xiao et al. \(2015\)](#) apresentam uma heurística híbrida que combina um algoritmo de relaxação Lagrangeana com a metaheurística *simulated annealing*. O problema de dimensionamento e sequenciamento é decomposto por máquinas através da relaxação Lagrangeana. A *simulated annealing* é utilizada para melhorar o sequenciamento e fornecer limitantes superiores para a otimização subgradiente da relaxação Lagrangeana.

[Clark, Mahdiah e Rangel \(2014\)](#) apresentam uma formulação que permite sequências que não obedecem a desigualdade triangular. Os autores estudam as vantagens de permitir

a produção de múltiplos lotes de um mesmo item em cada período. [Guimarães, Klabjan e Almada-Lobo \(2014\)](#) também permitem múltiplos lotes por período e utilizam restrições de eliminação de subrotas do tipo *multi-commodity*. Além disso, os autores propõem uma classificação para os problemas de dimensionamento e sequenciamento de lotes com *setups* dependentes da sequência.

Com base no modelo de [Clark, Mahdiah e Rangel \(2014\)](#), [Mahdiah, Clark e Bijari \(2017\)](#) propõem o primeiro MIP para o problema de dimensionamento e sequenciamento de lotes com tempos de *setup* não triangulares e dependentes da sequência que permite, além de múltiplos lotes de um produto em um período usando um número de restrições polinomiais e incorporando todas as características de *setup carryover*, como em [Clark, Mahdiah e Rangel \(2014\)](#), mas também a sobreposição de *setups* sobre os limites dos períodos. A inclusão desses *setups* atravessando os períodos permite a modelagem de sistemas mais realistas por relaxar todas as limitações de separação física entre os períodos.

[Tempelmeier e Copil \(2015\)](#) consideram o problema sob o contexto da indústria alimentícia, em um ambiente monoestágio com múltiplas máquinas. No modelo apresentado pelos autores é levado em consideração que existe um recurso escasso para realização do *setup* que tem que ser compartilhado entre todas as máquinas. Os autores estendem o modelo para considerar situações práticas desse tipo de processo produtivo, tais como, perecibilidade dos itens e processos de limpeza. Heurísticas *relax-and-fix* e *fix-and-optimize* são utilizadas para resolver instâncias reais.

Modelagens do tipo CLSP também são encontradas em trabalhos aplicados à indústria de bebidas: [Defalque, Rangel e Ferreira \(2011\)](#), [Ferreira et al. \(2012\)](#) e [Maldonado, Rangel e Ferreira \(2014\)](#). Para mais detalhes sobre esses trabalhos veja Apêndice C.

Trabalhos baseados no PLSP

[Tempelmeier e Buschkühl L. \(2008\)](#) apresenta uma formulação baseada no PLSP para um problema encontrado na indústria automobilística. Múltiplas máquinas estão disponíveis e cada produto é designado unicamente para uma máquina. Os *setups* são realizados por um único operador, assim como em [Tempelmeier e Copil \(2015\)](#). Para lidar com esse recurso escasso, os autores utilizam uma variável contínua que registra o início de um *setup* em cada máquina em cada microperíodo. Uma outra variável armazena a sequência em que o operador de *setup* percorre as máquinas. Uma reformulação baseada em localização de facilidades é apresentada. Um outro modelo também é proposto diferenciando maiores e menores *setups*.

Também baseado em uma situação real, [Stadtler \(2011\)](#) apresentam um modelo monomáquina, com produtos com estrutura de montagem para um problema encontrado na indústria farmacêutica. Os autores consideram a transferência imediata dos produtos de um estágio para o outro (ou seja, consideram o *lead time* como sendo zero). [Stadtler e](#)

Sahling (2013) apresentam duas formulações para o PLSP considerando múltiplos estágios, produtos multinível e múltiplas máquinas. Variáveis de controle de tempo são utilizadas para permitir que um *setup* inicie em um período, mas não termine no mesmo período. A consideração de que os produtos podem ser armazenados de um estágio para o outro por apenas um período é considerada em um dos modelos. Heurísticas *relax-and-fix* geram soluções iniciais que são melhoradas por heurísticas *fix-and-optimize*.

Trabalhos baseados no CSLP

Vanderbeck (1998) apresentam uma formulação CSLP com tempos de *setup* independentes da sequência. Para resolver o problema os autores utilizam o método de geração de colunas combinado com um algoritmo de planos de corte. Os subproblemas utilizados na geração de colunas são resolvidos via programação dinâmica.

Um caso real de uma empresa de fundição é considerado em Araujo, Arenales e Clark (2007). A matéria-prima é processada em uma única máquina e serve como base para fazer múltiplos produtos. A troca do tipo de matéria-prima têm custos e tempos dependentes da sequência. Apesar dos autores afirmarem que o problema é baseado no GLSP, os tamanhos dos microperíodos são fixos, e por isso este trabalho está classificado como CSLP. Segundo a classificação de Copil et al. (2017), o GLSP tem microperíodos de tamanhos variáveis. Para resolver o problema os autores utilizam uma abordagem de horizonte rolante combinado com heurísticas do tipo *relax-and-fix*. Araujo, Arenales e Clark (2008) adaptam a formulação para o caso independente da sequência. Araujo e Clark (2013) apresentam uma reformulação do modelo proposto em Araujo, Arenales e Clark (2007) através do problema de localização de facilidades. Os mesmos métodos de solução são utilizados.

Marinelli, Nenni e Sforza (2007) consideram um sistema de produção de iogurte em dois estágios. No primeiro estágio os iogurtes são preparados em tanques e embalados em potes no segundo estágio. Os tempos são dependentes da sequência tanto para mudança de sabor quanto para mudança de embalagem. Os autores propõem um modelo e o classificam como um híbrido entre CSLP e CLSP. Heurísticas que decompõem o problema em dimensionamento e sequenciamento de lotes são utilizadas.

Almada-Lobo et al. (2010) consideram o processo de produção de recipiente de vidro, que consiste de um forno que alimenta máquinas de moldagem em paralelo com vidro derretido. Para cada máquina de moldagem e período, é possível decidir o número de cavidades no molde que serão utilizadas, e portanto, a taxa de produção. O número de cavidades é inteiro, porém os autores relaxam o modelo permitindo lotes com valores contínuos. O modelo é decomposto em subproblemas através da relaxação Lagrangeana. Toledo, Arantes e Almada-Lobo (2013) consideram o mesmo problema, mas permitem apenas lotes discretos. Os autores resolvem o problema através de algoritmos genéticos combinados com heurísticas para determinar o melhor número de cavidades do molde.

Trabalhos baseados no DLSP

Salomon, Wassenhove L.N.V.; Dumas e Dauzère-pérès (1997) consideram um modelo do tipo DLSP com tempos e custos de *setup* dependentes da sequência. Os autores transformam o problema em um problema do caixeiro viajante com janela de tempo, que é resolvido. O modelo é transformado em um método de solução exato é resolvido via programação dinâmica. O processo consiste em transformar óleo cru em óleos especiais.

Um problema real de uma refinaria de óleo é abordado em Göthe-Lundgren, Lundgren e Persson (2002) através de um modelo multiestágio. O processo de produção é formado por uma unidade de destilação e duas unidades de hidro-tratamento. Essas unidades de produção tem modos de operação diferentes. O objetivo é decidir qual modo de cada unidade usar. Inequações válidas são propostas e uma heurística baseada em busca tabu é utilizada para resolver o problema.

Jans e Degraeve (2004) abordam o DLSP monoestágio para resolver o problema de dimensionamento e sequenciamento de lotes de uma indústria de manufatura de pneu. Os pneus são moldados (endurecidos) em aquecedores. Para cada processo desses, um tipo de molde é usado. Aquecedores e moldes são recursos escassos, e a principal parte do processo de *setup* é esquentar os moldes. O modelo é resolvido com o processo de geração de colunas.

Gicquel e Minoux (2015) consideram um DLSP monomáquina com conservação do estado de *setup* entre os períodos. Os autores também, propõem inequações válidas para o problema.

Existem muitos outros trabalhos de dimensionamento e sequenciamento de lotes com modelagens baseadas no PLSP, CSLP e DLSP que não foram abordados aqui por não estarem no foco das características investigadas. Alguns deles estão listados e discutidos em Copil et al. (2017).

Outras abordagens de modelagem

Outros modelos diferentes dos clássicos, GLSP, CLSD, PLSP, CSLP e DLSP, podem ser encontrados na literatura científica de dimensionamento e sequenciamento de lotes. Muitas dessas publicações estão mais próximas da área de sequenciamento (*scheduling*) do que de dimensionamento (*lot sizing*). O foco desses modelos muitas vezes está mais em coordenar o início e final de processamento de operações do que em dimensionar. Por exemplo, Dauzère-Pérès e Lasserre (1994) investigam a integração do dimensionamento de lotes com o sequenciamento em um sistema de produção *job-shop*. Os autores apresentam uma combinação entre um modelo para o dimensionamento e um modelo para o sequenciamento. Para resolver o problema, os autores fazem uma iteração entre resolver o dimensionamento de lotes para uma dada sequência de *jobs* em cada máquina, e computar uma sequência para tamanhos de lotes fixos. Diferentes

abordagens de solução são apresentadas para o problema de sequenciamento. [Gómez-Urrutia, Aggoune e Dauzère-Pérès \(2014\)](#) utiliza o mesmo procedimento para resolver o problema. O dimensionamento de lotes é resolvido via um modelo do tipo CLSP. Esse modelo tem as restrições de capacidade relaxadas, uma sequência de produção fixada e é resolvido via Relaxação Lagrangeana. Para melhorar o sequenciamento utiliza-se busca tabu, que leva em consideração informações da heurística lagrangeana. [Wolosewicz, Dauzère-Pérès e Aggoune \(2015\)](#) abordaram o mesmo problema descrito acima. Uma nova formulação é apresentada para gerar um plano de produção dada uma sequência de operações para cada máquina. O método de Relaxação Lagrangeana é utilizada para resolver o problema.

[Haase e Kimms \(2000\)](#) apresenta um modelo que considera sequências de produção pré-definida através do problema do caixeiro viajante. O problema de dimensionamento e sequenciamento de lotes é resolvido simultaneamente decidindo qual a sequência mais eficiente a ser utilizada e a quantidade que deve ser produzida. O modelo representa o problema de uma indústria de máquinas de pintura e é resolvido através do método *Branch-and-Bound*. [Kovács, Brown e Tarim \(2009\)](#) reformulam o modelo e melhoram a definição das sequências utilizando programação dinâmica.

O problema de dimensionamento e sequenciamento de lotes em um sistema de manufatura com um único estágio, composto por máquinas paralelas não relacionadas e considerando os tempos de *setup* dependentes da sequência é abordado por [Mateus et al. \(2010\)](#). Os autores propõem um modelo em que os lotes são interpretados como *jobs* e o modelo coordena os tempos de início dos lotes e variáveis que associam produtos a períodos e máquinas. Uma vez que o modelo é difícil de ser resolvido, eles utilizam decomposição. Primeiramente o dimensionamento de lotes é resolvido até otimalidade. Depois os *jobs* (lotes) são sequenciados através de uma metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) desenvolvida por [Ravetti et al. \(2007\)](#).

[Mohammadi et al. \(2010a\)](#) e [Mohammadi et al. \(2010b\)](#) tratam o problema de dimensionamento e sequenciamento de lotes com setups dependentes da sequência em um sistema de produção do tipo *flow-shop*. Um item pode ser transferido para uma máquina somente depois de ser completamente finalizado. Em cada período e em cada máquina devem ser realizados *setups* em número idêntico ao de produtos, assim como no trabalho de [Clark e Clark \(2000\)](#). *Setups* de um produto para ele mesmo são permitidos e não consomem tempo da capacidade. Ambos os trabalhos apresenta o mesmo modelo matemático. [Mohammadi et al. \(2010a\)](#) apresentam cinco heurísticas, três baseadas em horizonte rolante e outras duas do tipo *relax-and-fix* utilizando um modelo modificado para *flow-shop* permutacional. [Mohammadi et al. \(2010b\)](#) também propõem MIP-heurísticas do tipo horizonte rolante e *relax-and-fix*. [Mohammadi et al. \(2010\)](#) apresentam uma simplificação do modelo de [Mohammadi et al. \(2010a\)](#) e [Mohammadi et al. \(2010b\)](#),

considerando os tamanhos dos lotes idênticos em todos os estágios. O modelo é resolvido via heurística de horizonte rolante. [Mohammadi, Ghomi e Jafari \(2011\)](#) propõem um algoritmo genético para o modelo inicial. Um algoritmo genético combinado com uma heurística de horizonte rolante são utilizados para resolver o mesmo problema em [Mohammadi e Ghomi \(2011\)](#). A cada passo da heurística de horizonte rolante, as variáveis binárias são determinadas através do algoritmo genético. [Babaei, Mohammadi e Ghomi \(2014\)](#) incluem atraso no problema e apresentam um algoritmo genético para resolvê-lo. Usando o mesmo modelo proposto por [Mohammadi et al. \(2010a\)](#), [Belo-Filho, Santos e Meneses \(2012\)](#) propõem uma nova variante do método de solução “*Asynchronous Team*” (A-Team) de [Souza e Talukdar \(1993\)](#) para resolver o problema. Nesse método, os agentes são métodos heurísticos que constroem soluções, armazenam essas soluções para serem melhoradas ou remove uma solução não considerada boa. Esse método é capaz de encontrar melhores soluções do que as encontradas na literatura, principalmente para instâncias consideradas grandes. [Ramezani, Saidi-Mehrabad e Teimoury \(2013\)](#) abordam o mesmo problema e à partir do modelo proposto em [Mohammadi et al. \(2010a\)](#) e [Mohammadi et al. \(2010b\)](#), propõem uma formulação mais apertada, porém considerando apenas o caso do *flow-shop* permutacional e duas variações: com e sem *setup carry-over*. Para resolver o problema são propostas heurísticas do tipo horizonte rolante.

[Kopanos, Puigjaner e Maravelias \(2011\)](#) propõem um modelo inteiro misto para o problema de planejamento e programação da produção em indústrias de processo contínuo em um ambiente monoestágio com máquinas em paralelo. Os produtos são agregados por famílias. *Setups* entre famílias são dependentes da sequência e *setups* entre produtos da mesma família são independentes da sequência. Existe o aproveitamento do *setup* de um período para o outro e atrasos são permitidos. A ligação entre o dimensionamento e o sequenciamento é feito via variável de controle da produção. Manutenções podem ser consideradas no modelo. Estas podem ser do tamanho de um período (cada período é considerado como um dia nesse modelo), e nesse caso, a capacidade para o período em que haverá a manutenção é zerada. Quando a manutenção é menor do que o tamanho do período, está é abordada como um *setup* entre duas famílias. [Kopanos, Puigjaner e Georgiadis \(2011\)](#) estudam o processo de produção de iogurte em um sistema contínuo de produção, também. Um modelo parecido com o de [Kopanos, Puigjaner e Maravelias \(2011\)](#) é abordado, porém considerando especificidades da fabricação de iogurte, como por exemplo uma limpeza no final de cada período (dia), que é descontada da capacidade.

[Karimi-Nasab e Seyedhoseini \(2013\)](#) consideram o problema de dimensionamento e sequenciamento de lotes multinível em um sistema *job shop*, com a consideração realística de que as velocidades das máquinas podem ser alteradas no início de produção. Os modos em que as máquinas podem operar (velocidades) são pré-definidos e escolhidos pelo modelo. Várias decisões dependem das velocidades em que as máquinas operam, por exemplo, a quantidade que pode ser produzida e a iteração entre as máquinas por onde

um item tem que passar. Os *setups* podem ocorrer em duas situações: troca do tipo de item que está sendo produzido, nesse caso, o *setup* depende da sequência; e uma limpeza que acontece sempre que um lote de um item é finalizado, antes do início de um novo lote do mesmo item. Os autores apresentam diversas inequações válidas para o problema e aplicam os métodos *cut-and-branch* e *branch-and-cut* para resolvê-lo. Karimi-Nasab et al. (2013) abordam o mesmo problema, porém considerando que o tempo de duração do processamento de um item em uma máquina pode variar dentro de um intervalo. Considerando que os produtos finais são formados pela montagem de itens processados nas máquinas (a estrutura de produto nesse caso é montagem), uma quantidade suficiente de itens deve estar disponível antes do produto final ser montado. Os autores apresentam um algoritmo memético (algoritmo genético combinado com algum algoritmo de busca local) para resolver o problema. Karimi-Nasab, Modarres e Seyedhoseini (2015) continua abordando o mesmo problema, porém com a consideração que um produto final é um item que passou por várias máquinas, ou seja, a estrutura do produto é serial. Para resolver o problema os autores utilizam um algoritmo de enxame de partículas. Karimi-Nasab e Modarres (2015) também abordam um problema com as mesmas hipóteses, mas consideram que se um item é iniciado em um período, tem que ser finalizado no mesmo período, ou seja, não pode haver estoque de itens em processo de um período para o outro. Inequações válidas são propostas e adicionadas ao modelo através do método *cut-and-branch*.

Ramezani, Saidi-Mehrabad e Fattahi (2013) estudam o problema de dimensionamento e sequenciamento de lotes em um ambiente *flow-shop*, considerando a programação de manutenções preventivas. As decisões de programação dos lotes e das manutenções preventivas são tomadas de maneira integrada. As manutenções preventivas devem acontecer dentro de janelas de tempo pré-definidas. Para resolver o problema os autores propõem um modelo em que variáveis de controle de tempo são utilizadas para garantir a programação dos lotes juntamente com a programação das manutenções. Heurística de horizonte rolante são utilizadas para resolver o problema. Ramezani, Sanami e Nikabadi (2017) abordam a programação da produção de uma fábrica de telhas de barro, também em um sistema do tipo em um ambiente *flow-shop*. O objetivo é minimizar custos de estoque e produção. Atrasos não são permitidos, os autores admitem que a demanda não atendida deve ser adquirida de fornecedores externos por valores muito mais altos do que de produção. Para resolver o problema eles utilizam heurísticas de horizonte rolante e enxame de partículas. Recentemente, Boonmee e Sethanan (2016) apresentaram um modelo para tratar o planejamento e programação da produção de uma indústria de ovos de galinha. Trata-se de um problema multiestágio, pois para obtenção do ovo considera-se os estágios desde o nascimento das galinhas. Os autores utilizam uma variante do método de enxame de partículas.

APÊNDICE C – Problemas de Programação da Produção na Indústria de Bebidas

Existem na literatura alguns trabalhos que tratam da programação da produção de bebidas, sendo a grande maioria deles relacionados a problemas de dimensionamento e sequenciamento de lotes de produção de refrigerantes e cerveja. Antes de detalhar esses trabalhos, apresenta-se brevemente o processo de produção de bebidas.

C.1 Processo de Produção de Bebidas

O processo de produção de bebidas em geral é composto por tratamento de água, preparo da bebida, envase e empacotamento. Todos os tipos de refrigerantes, e outras bebidas como chás, sucos, água saborizadas e cerveja passam por estas etapas. No refrigerante o preparo da bebida trata-se do preparo de xaropes que são depois misturados à água e à gás carbônico. Como representado na Figura 45, os dois principais estágios desse processo produtivo são a preparação do sabor do líquido (estágio 1) em tanques de diferentes capacidades e o engarrafamento, ou melhor, envase (estágio 2) desse líquido (um concentrado de xarope mais um pouco de água). Dois sabores diferentes não podem ser preparados ao mesmo tempo em um mesmo tanque. Por razões técnicas o tanque deve estar vazio antes de um lote de um novo líquido ser preparado neste tanque, mesmo que o novo lote seja do mesmo sabor do anterior. Os tempos de preparação (limpeza) dos tanques e os custos são dependentes da sequência em que os sabores são preparados. Uma quantidade mínima de líquido de determinado sabor deve ser preparada para garantir a homogeneidade do líquido, dado que a hélice do tanque tem de ser completamente coberta, a fim de misturar adequadamente os ingredientes necessários, [Ferreira, Morabito e Rangel \(2010\)](#).

No segundo estágio os líquidos são envasados. Uma linha de envase é composta por uma esteira rolante e máquinas que lavam os vasilhames, enche-os com uma combinação do xarope com água (carbonatada ou não carbonatada) e então selam, rotulam e empacotam os vasilhames cheios. Conceitualmente, a linha inteira pode ser considerada como uma única máquina também referida como linha de produção ou simplesmente linha que processa itens caracterizados por diferentes combinações de sabor/vasilhame. Os tanques do estágio I são liberados para preparação de outro lote de xarope apenas após a linha terminar o enchimento de todos os vasilhames. Obviamente, uma linha só pode iniciar o envase se o xarope estiver pronto ([FERREIRA; MORABITO; RANGEL, 2010](#)).

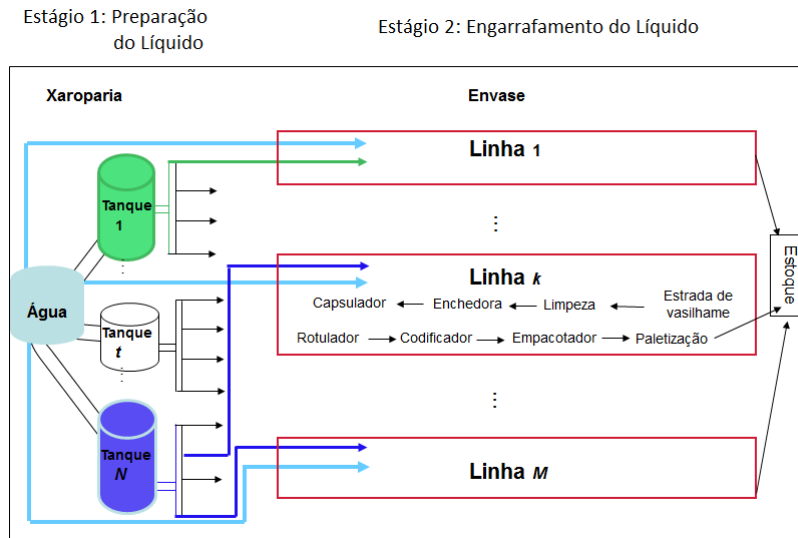


Figura 45 – Processo produtivo de refrigerantes. Fonte: Adaptado de Ferreira *et al.* (2011).

Uma linha só pode receber xarope de apenas um tanque por vez, mesmo que outros tanques estejam disponíveis. Entretanto, um tanque pode enviar líquido para mais de uma linha simultaneamente, se estas estiverem envasando bebidas do mesmo sabor, conforme a representação da Figura 45, onde o tanque N abastece as linhas K e M . Para produzir uma bebida de um outro sabor e/ou tamanho-vasilhame, é necessário parar a produção e fazer toda a limpeza e/ou ajustes necessários para a produção de um outro tamanho de vasilhame e/ou xarope. Estes ajustes envolvem custos e tempos de preparo dependentes da sequência de produção (DEFALQUE; RANGEL; FERREIRA, 2011).

Dado que a produção de bebidas é feita em dois estágios, um outro fator importante na programação dessa produção é a consideração da sincronia entre estes estágios (FERREIRA; MORABITO; RANGEL, 2009). Se o tanque não estiver com a bebida pronta para ser enviado para a linha de envase, esta deve aguardar até que a bebida esteja pronta. Do mesmo modo, o tanque só pode iniciar o envio da bebida para a linha de envase se ela estiver preparada para o envase. Assim, podem ocorrer esperas da linha pelo tanque e do tanque pela linha de envase. A Figura 46 representa a situação em que os tanques estão dedicados a uma determinada linha de produção. Porém, na Figura 46a não há sincronia entre os estágios, enquanto, que na Figura 46b há sincronia e os retângulos de cor preta representam a diferença entre o tempo de troca de produtos da linha e o tempo de troca de xaropes no tanque.

O processo de produção descrito acima é, de maneira geral, encontrado em diversas indústrias de refrigerantes (FERREIRA; MORABITO; RANGEL, 2009). Nessas empresas existem a demanda (semanal, mensal ou até diária) dos clientes que deve ser atendida procurando reduzir custos, tais como de estoque e de atraso. Sendo assim, o responsável pelo planejamento da produção deve decidir o que produzir (sequenciar) e em que quantidade

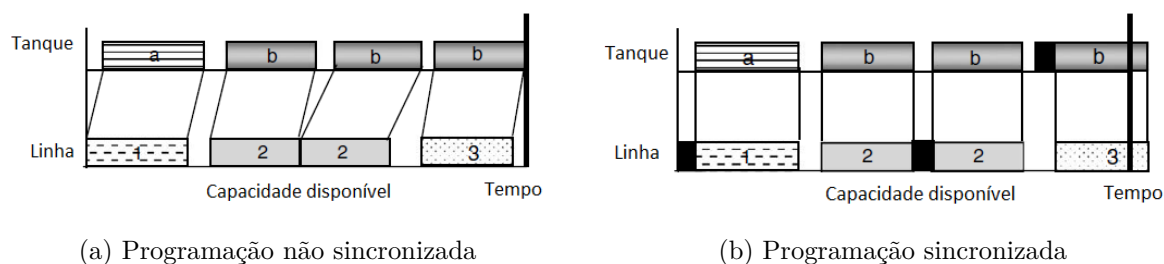


Figura 46 – Exemplo de programação sincronizada e não sincronizada. Fonte: Adaptado de (FERREIRA; MORABITO; RANGEL, 2009).

produzir (dimensionar), considerando diversas restrições tais como: muitos itens, diversas máquinas com tempos de processamento diferentes, capacidade restrita, sincronia entre os estágios e tempos de troca dependentes da sequência. Por exemplo, produzir um refrigerante *diet* e, logo em seguida, um refrigerante normal gasta menos tempo do que o contrário, pois do *diet* para o normal o tempo de troca é menor.

A produção de cerveja segue basicamente os mesmos passos da produção de bebidas em geral (preparo e envase), sendo que o preparo da bebida é a maior diferença com relação à produção de refrigerantes, por exemplo. Para preparar a cerveja são necessários os seguintes passos: maltagem (transformação de cevada em malte), brassagem (obtenção do mosto cervejeiro), fermentação/maturação (são adicionadas leveduras ao mosto que transformam os açúcares em dióxido de carbono e álcool). Para mais detalhes sobre essas etapas veja, por exemplo, (BALDO, 2014). Diferentemente do refrigerante em que o preparo da bebida tem duração de algumas horas, o processo de maturação da cerveja que acontece nos tanques preparatórios pode durar de 3 a 41 dias. Além disso, o líquido pode esperar dentro dos tanques de fermentação e maturação até o envase começar. Outra diferença com a produção de refrigerantes é que o tempo e custo de troca de líquido nos tanques independe da sequência. Na linha de envase, os tempos e custos de troca dependem da sequência. Assim como na produção de refrigerantes, é necessário considerar a sincronia entre os estágios de preparo e envase.

Na produção de bebidas à base de frutas, também estão presentes os estágios de preparo e envase da bebida. Diferente do refrigerante, em que o xarope é misturado à água e gás carbônico na linha, as bebidas a base de frutas saem prontas da xaroparia e vão para um tanque pulmão, de onde saem para o processo de pasteurização, antes de serem envasadas. O envase dessas bebidas também é um pouco diferente, exige alguns cuidados de assepsia e por isso é necessário a realização de limpezas obrigatórias a cada determinado tempo de produção contínua. Assim como nos refrigerantes e na cerveja, é preciso considerar a sincronia entre os dois estágios de produção. Mais detalhes da produção de bebidas à base de frutas são apresentados no Capítulo 3.

C.2 Trabalhos encontrados da literatura

Nesta seção serão descritos e discutidos os trabalhos encontrados na literatura científica que tratam o planejamento e programação da produção de bebidas no nível tático/operacional (decisões de curto prazo). Até onde se pesquisou existem 16 trabalhos que tratam do dimensionamento e sequenciamento de lotes, sendo que destes apenas 2 não são relacionados a produção de refrigerantes e cerveja. Esses trabalhos estão sumarizados na Tabela 49. A primeira coluna indica o tipo de bebida; a segunda a referência do trabalho; a terceira apresenta a formulação clássica na qual os modelos são baseados; as demais colunas referem-se as principais características do problema que foram consideradas na formulação como o número de estágios (único ou múltiplo), o número de máquinas e as características do *setup* (dependente ou independente da sequência); a penúltima coluna diz respeito ao tipo de instância utilizada para os testes computacionais e a última coluna descreve o método de solução utilizado.

No nível tático/operacional, não foram encontrados mais trabalhos na indústria de bebidas além dos citados na Tabela 49. Porém no nível estratégico são encontrados os seguintes trabalhos: Guimarães, Klabjan e Almada-Lobo (2012), Christou, Lagodimos e Lycopoulou (2007) e Gunther (2008). Um outro trabalho que também explora a programação da produção de bebidas, porém sob o ponto de vista estocástico foi encontrado em Sel e Bilgen (2014). Esses trabalhos não serão descritos em detalhes nesta seção e nem classificados na Tabela 49, por não tratarem do problema alvo desse trabalho.

Os primeiros trabalhos sobre bebidas que surgiram na literatura e que tratam do dimensionamento de lotes de uma indústria de bebidas foram propostos por Clark (2003a) e Rangel e Ferreira (2003). Ambos trabalhos fazem uma adaptação do modelo *Capacitated Lot Sizing Problem* (CLSP) (KARIMI; GHOMI; WILSON, 2003) para apresentar o planejamento da produção de uma fábrica de bebidas. Rangel e Ferreira (2003) resolvem o problema através do método *Branch-and-Cut* do *solver* CPLEX (ILOG, 2014) com algumas instâncias geradas aleatoriamente com base em dados reais. Clark (2003a) busca resolver um exemplar específico de uma fábrica. Para resolver o problema, o autor propõe heurísticas híbridas que combinam busca local com programação matemática.

Em Ferreira, Morabito e Rangel (2008), Ferreira, Morabito e Rangel (2010) foi estudado um problema de dimensionamento e sequenciamento da produção de refrigerantes em uma fábrica de pequeno porte por meio de um modelo de otimização baseado no modelo GLSP (*General Lotsizing and Scheduling Problem*) (FLEISCHMANN; MEYR, 1997). O modelo considera apenas o estágio envase da bebida, tratado como gargalo da produção, com uma única linha de envase. Heurísticas do tipo *relax-and-fix* (WOLSEY, 1998) foram propostas para resolver o modelo, sendo que em Ferreira, Morabito e Rangel (2010) essas heurísticas foram combinadas com a calibração das estratégias de solução utilizadas pelo *solver* CPLEX. Ambos trabalhos utilizam instâncias baseadas em dados

reais e encontraram soluções melhores que a da empresa.

Defalque, Rangel e Ferreira (2011) propõem um modelo para o problema integrado de dimensionamento e sequenciamento da produção de refrigerantes com as decisões de dimensionamento baseadas no modelo de Ferreira, Morabito e Rangel (2010) e as decisões de sequenciamento modeladas utilizando restrições do problema do caixeiro viajante assimétrico (*ATSP - asymmetric traveling salesman problem*), com restrições de eliminação de subrotas (ÖNCAN; ALTINEL; LAPORTE, 2009). Os exemplares foram resolvidos pelo método *Branch-and-Cut* incluído no pacote computacional CPLEX. Na resolução de uma instância real, o modelo proposto por Defalque, Rangel e Ferreira (2011) teve um desempenho computacional do ponto de vista de solução melhor que o proposto por Ferreira, Morabito e Rangel (2010).

Ainda com relação à modelagem do problema via ATSP, Maldonado, Rangel e Ferreira (2014) comparam o uso de três conjuntos de restrições diferentes para eliminação de subrotas no modelo proposto por Defalque, Rangel e Ferreira (2011), dois conjuntos já utilizados na literatura de bebidas (MTZ e DFJ) e um conjunto pouco explorado das restrições de multi-comodite (ÖNCAN; ALTINEL; LAPORTE, 2009). O objetivo do trabalho é comparar o desempenho do modelo com relação à essas restrições. Testes computacionais foram realizados com instâncias reais e baseadas em dados reais. Os resultados mostraram que as restrições do tipo multi-comodite proveem boas soluções, porém em exige altos tempos computacionais.

Um caso mais geral do problema considerando os dois estágios de produção com sincronia e várias máquinas foi estudado em Toledo et al. (2007), Toledo et al. (2009) e Toledo et al. (2015). Toledo et al. (2007), Toledo et al. (2015) propõem um modelo de otimização inteira mista baseado nos modelos GLSP e CSLP (*Continuous Setup Lotsizing Scheduling Problem*) (DREXL; KIMMS, 1997). Para realizar testes computacionais são geradas instâncias com base em alguns parâmetros reais. Devido à dificuldade de solução do modelo foram propostas abordagens de solução por meio de algoritmos genéticos em (TOLEDO et al., 2009). O algoritmo consegue encontrar soluções factíveis melhores que as soluções factíveis encontradas pelo modelo.

Ferreira, Morabito e Rangel (2009) pesquisou o mesmo problema que Toledo et al. (2009). Formulações mais fáceis de serem resolvidas foram propostas onde os tanques do primeiro estágio são dedicados à linhas do segundo estágio. Assim como (TOLEDO et al., 2007), o modelo é baseado no GLSP e a aplicação é na programação da produção de uma fábrica de refrigerantes. Testes computacionais são realizados com dados reais e com instâncias baseadas em dados reais de uma fábrica de refrigerantes. Devido à dificuldade de resolução do modelo, são propostas heurísticas do tipo *relax-and-fix* combinadas com a calibração das estratégias de resolução do CPLEX. Tendo em vista que ainda não foram obtidas as soluções ótimas do problema e os *gaps* eram altos, Ferreira et al. (2012)

apresentam formulações alternativas nas quais o problema sincronizado de dois estágios é formulado como um problema mono-estágio. São apresentadas duas reformulações baseadas no GLSP (chamadas de R1 e R2) e duas baseadas no ATSP (F1 e F2). Os resultados mostram que as formulações monoestágio são muito mais eficientes que a formulação dois estágios proposta por (FERREIRA; MORABITO; RANGEL, 2009), e além disso, as estratégias baseadas no ATSP são ainda mais rápidas.

Na tentativa de encontrar boas soluções em um tempo razoável para o problema apresentado em Ferreira, Morabito e Rangel (2009), Toledo et al. (2014) apresentam uma abordagem que combina um algoritmo genético com técnicas de programação matemática para resolver o problema. Para avaliar a performance dessa abordagem, os autores realizam testes computacionais com os dados reais de Ferreira, Morabito e Rangel (2009) e comparam os resultados com as estratégias F1 e F2 de Ferreira et al. (2012). O método proposto supera os resultados de F1 e F2 em termos de tempo computacional, dependendo apenas um quarto do tempo.

Cunha e Ferreira (2014) apresentam um modelo mais geral que o modelo F1 de Ferreira et al. (2012). Nele foram desconsideradas as restrições de estoque mínimo e limitação do número de lotes de produção. Por tratar-se de um problema ainda difícil, a autora apresenta heurísticas construtivas, de decomposição do problema em diversas máquinas, heurísticas *relax-and-fix* e heurísticas híbridas que combinam uma heurística construtiva com o método *Cut-and-Branch* do CPLEX. Testes computacionais realizados com instâncias reais da literatura mostraram que as heurísticas são promissoras para resolver o problema.

No setor de bebidas alcoólicas o problema se diferencia da produção de refrigerantes principalmente no primeiro estágio, em que os tanques podem ficar ocupados entre 3 e 41 dias para fazer a maturação da cerveja. No refrigerante, assim que é terminada a preparação do xarope, que dura em torno de horas, este é imediatamente enviado para o estágio de envase. Além disso, na produção de cerveja, nos tanques os tempos e custos de *setup* não são dependentes da sequência, somente nas linhas de envase. Para os refrigerantes os custos e tempos de *setups* são dependentes da sequência nos dois estágios. O gargalo de produção de cervejas pode alternar entre os estágios de preparo e envase da bebida e a sincronia entre estes estágios também é o grande desafio desse problema. Baldo et al. (2014) resolvem a programação da produção das indústrias de cerveja. Como o envase é parecido com a produção de refrigerantes, Baldo et al. (2014) apresentam um modelo dois estágios em que as decisões do segundo estágio são baseadas no modelo de Ferreira, Morabito e Rangel (2009). Conforme citado acima, a principal diferença entre os dois modelos são as restrições dos tanques, que consideram a maturação da cerveja e a maneira como é feita a sincronia entre os dois estágios. Como nos trabalhos sobre refrigerantes, para as instâncias reais de grande porte os modelos são difíceis de serem resolvidos. Logo

os autores propuseram métodos de solução construtivos baseados em heurísticas do tipo *relax-and-fix* e procedimentos de melhoria baseados nas estratégias *fix-and-optimize*. Testes computacionais foram realizados com instâncias geradas com base em dados reais. Os resultados encontrados são melhores que os fornecidos pelo CPLEX tanto em termos de valor de solução quanto de tempo computacional.

Ao contrário da produção de refrigerantes, para a produção de bebidas de frutas são encontrados apenas dois trabalhos na literatura, Leite (2008) e Pagliarussi (2013). Em ambos trabalhos foram observadas as características inerentes a esse processo produtivo descritas acima: a existência de um pequeno estoque intermediário entre os dois estágios; um processo de pasteurização; limpezas periódicas; e a perecibilidade dos itens.

Para tratar desse problema, Leite (2008) apresentou três modelos de dimensionamento de lotes baseados no modelo de Ferreira, Morabito e Rangel (2008). Os três modelos consideraram apenas uma máquina na linha de produção e um único estágio de produção (envase). As únicas restrições do estágio de preparo consideradas foram as de lote mínimo e máximo do tanque. A diferença entre os modelos foi a maneira de considerar estoque de segurança dos produtos finais, que era uma característica da fábrica que inspirou o trabalho de Leite (2008).

Pagliarussi, Morabito e Santos (2017) considera o problema tratando o caso de múltiplas máquinas e também propõe três modelos, um para maximização da produção, baseado no GLSP e outros dois para minimização de custos, sendo um deles baseado no GLSP e ou outro no CLSP. A autora não considera estoque de segurança como feito em Leite (2008) e modela apenas o estágio de envase.

Tanto Leite (2008) quanto Pagliarussi, Morabito e Santos (2017) realizaram testes computacionais com dados reais coletados de uma mesma fábrica no interior do estado de São Paulo. Ambos trabalhos resolvem os respectivos modelos através do método *Branch-and-Cut default* do CPLEX e obtêm resultados competitivos quando comparados à solução real.

Observando essa revisão de literatura, é possível notar que, apesar de haver uma gama de trabalhos que estudaram o planejamento e a programação da produção de bebidas, pouco foi feito especificamente para o setor de bebidas à base de frutas. No entanto, o problema de programação da produção de bebidas à base frutas, apesar de ser no setor de bebidas, não pode ser resolvido por uma simples adaptação dos problemas já encontrados na literatura. Trata-se de um novo problema de dimensionamento de lotes com características peculiares que necessitam de uma nova abordagem de modelagem e métodos de solução. Um exemplo disso é a existência das limpezas temporais, que exigem um controle explícito do tempo decorrido ao longo da produção. Uma outra característica inerente a esse problema, e que não foi encontrada na literatura, é a consideração de um lote de folga entre os dois estágios. Isso garante que enquanto o segundo estágio está

processando um lote, o primeiro estágio já está preparando o próximo lote. Além de todas essas características é preciso considerar a sincronia entre os estágios, que já foi abordada em outros trabalhos (veja Tabela 49), porém sem a consideração do tanque pulmão e das limpezas periódicas, o que muda totalmente o problema.

Tabela 49 – Resumo dos trabalhos relacionados ao dimensionamento e sequenciamento de lotes na indústria de bebidas.

Bebida	Referência	Modelo	Estágios (=1 ou >1) ^a	Máquinas (=1 ou >1) ^b	Setup (DS ou IS) ^c	Instâncias (R ou BR ou GBR) ^d	Métodos de Solução
Refrigerante	Clark (2003a)	CLSP	= 1	= 1	IS	R	Programação Matemática + heurísticas de busca local CPLEX (<i>default</i>)
	Rangel e Ferreira (2003) Toledo et al. (2007)	CLSP GLSP+CSLP	= 1 > 1	= 1 > 1	IS DS	GBR GBR	Relaxou algumas variáveis e utilizou CPLEX (<i>default</i>) Heurísticas <i>Relax-and-Fix</i> Algoritmo Genético
	Ferreira, Morabito e Rangel (2008) Toledo et al. (2009)	GLSP GLSP+CSLP	= 1 > 1	= 1 > 1	DS DS	R+BR GBR	Heurísticas <i>Relax-and-Fix</i> + CPLEX (variando estratégias)
	Ferreira, Morabito e Rangel (2009)	GLSP	> 1	> 1	DS	R+BR	Heurísticas <i>Relax-and-Fix</i> + CPLEX (variando estratégias)
	Ferreira, Morabito e Rangel (2010)	GLSP	= 1	= 1	DS	R+BR	Heurísticas <i>Relax-and-Fix</i> + CPLEX (variando estratégias)
	Defalque, Rangel e Ferreira (2011) Ferreira et al. (2012)	CLSP+ATSP CLSP+ATSP e GLSP	= 1 > 1	= 1 > 1	DS DS	R+BR BR	Heurísticas <i>Relax-and-Fix</i> + CPLEX (variando estratégias) CPLEX (<i>default</i>) CPLEX (<i>default</i>)
	Cunha e Ferreira (2014)	CLSP+ATSP	> 1	> 1	DS	BR	CPLEX (<i>default</i>) + Heurísticas construtivas, de decomposição e híbridas CPLEX (<i>default</i>)
	Maldonado, Rangel e Ferreira (2014) Toledo et al. (2014)	CLSP+ATSP GLSP	= 1 > 1	= 1 > 1	DS DS	R+BR BR	Algoritmo Genético com Programação Matemática Relaxou algumas variáveis e utilizou CPLEX (<i>default</i>)
	Toledo et al. (2015)	GLSP+CSLP	> 1	> 1	DS	GBR	
	Fruita (Néctar e Refresco)	Leite (2008) Pagliarussi, Morabito e Santos (2017)	GLSP GLSP e CLSP	= 1 = 1	= 1 > 1	IS IS	R GBR
Cerveja	Baldo et al. (2014)	GLSP	> 1	> 1	DS	BR	Heurísticas <i>Relax-and-Fix</i> e <i>Fix-and-Optimize</i>

^a = 1: Único estágio; > 1: Múltiplos estágios.^b = 1: Única máquina; > 1: Múltiplas máquinas.^c DS: Dependente da sequência; IS: Independente da sequência. Instância gerada com base em dados reais.^d R: Dados reais; BR: Baseados em dados reais; GBR: gerados aleatoriamente com base em parâmetros reais.

APÊNDICE D – Modelo MDSL-2E-LT completo

Apresenta-se a seguir o modelo MDSL-2E-LT completo apresentado no Capítulo 4.

$$\begin{aligned}
\text{Min } Z = & \sum_{t \in T} \sum_{j \in J} (h_j^+ I_{jt}^+ + h_j^- I_{jt}^-) + \sum_{m \in M} \sum_{t \in T} \sum_{j \in J} \sum_{\substack{i \in J \\ i \neq j}} C_{ij} Z_{mijt} \\
& + \sum_{m \in M} \sum_{j \in J} \sum_{t \in T} \sum_{o \in O_{mt}} \sum_{k \in Q_{mt}^I} (Cl_t W_{mitok}^I + \alpha \omega_{mitok}^I) \\
& + \sum_{m \in M} \sum_{j \in J} \sum_{t \in T} \sum_{o \in O_{mt}} \sum_{l \in Q_{mt}^{II}} (Cl_t W_{mitol}^{II} + \alpha \omega_{mitol}^{II}) \quad (4.4)
\end{aligned}$$

sujeito a:

$$I_{j(t-1)}^+ + I_{jt}^- + \sum_{m \in M} \sum_{o \in O_{mt}} X_{mjto} = d_{jt}^I + I_{jt}^+ + I_{j(t-1)}^-, \quad \forall j \in J, \forall t \in T. \quad (4.5)$$

$$\rho X_{mjto} \geq LB_j Y_{mjto}, \quad \begin{array}{l} \forall m \in M, \forall j \in J, \\ \forall t \in T, \forall o \in O_{mt}. \end{array} \quad (4.6)$$

$$\rho X_{mjto} \leq \min\{UB_j, S_m TL_{max}\} Y_{mjto}, \quad \begin{array}{l} \forall m \in M, \forall j \in J, \\ \forall t \in T, \forall o \in O_{mt}. \end{array} \quad (4.7)$$

$$Y_{mjto} \geq Y_{mjto-1}, \quad \begin{array}{l} \forall m \in M, \forall j \in J, \forall t \in T, \\ \forall o \in O_{mt}, o > 1. \end{array} \quad (4.8)$$

$$\sum_{o \in O_{mt}} Y_{mioto} \leq 1, \quad \forall m \in M, \forall t \in T. \quad (4.9)$$

$$\sum_{\substack{i \in J \\ i \neq j}} Z_{mijt} = \sum_{\substack{i \in J \\ i \neq j}} Z_{mjit}, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (4.10)$$

$$\sum_{\substack{j \in J \\ j \neq i_0}} Z_{mi_0jt} \geq \sum_{\substack{i \in J \\ i \neq k}} Z_{mikt}, \quad \forall m \in M, \forall k \in J, \forall t \in T. \quad (4.11)$$

$$\sum_{\substack{j \in J \\ j \neq i}} Z_{mijt} \leq 1, \quad \forall m \in M, \forall i \in J, \forall t \in T. \quad (4.12)$$

$$V_{mjt} \geq (V_{mit} + 1) - (|J| - 1)(1 - Z_{mijt}), \quad \begin{array}{l} \forall t \in T, \forall m \in M, \\ \forall i, j \in J \setminus i_0, i \neq j. \end{array} \quad (4.13)$$

$$\sum_{o \in O_{mt}} Y_{mjto} \leq |O_{mt}| \sum_{\substack{i \in J \\ i \neq j}} Z_{mijt}, \quad \begin{array}{l} \forall m \in M, \forall j \in J, j \neq i_0, \\ \forall t \in T. \end{array} \quad (4.14)$$

$$\mu_{mj1t}^{Is} \geq LT^I Z_{mi_0jt}, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (4.15)$$

$$\mu_{mj1t}^{II_s} \geq \max\{LT^{II} Z_{mi_0jt}, \mu_{mj1t}^{Ie}\}, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (4.16)$$

$$\mu_{mjto}^{Ie} = \mu_{mjto}^{Is} + Tp Y_{mjto}, \quad \begin{array}{l} \forall m \in M, \forall j \in J, \forall t \in T, \\ \forall o \in O_{mt}. \end{array} \quad (4.17)$$

$$\mu_{mjto}^{II_e} = \mu_{mjto}^{II_s} + (X_{mjto}/S_m), \quad \begin{array}{l} \forall m \in M, \forall j \in J, \forall t \in T, \\ \forall o \in O_{mt}. \end{array} \quad (4.18)$$

$$\mu_{mjto}^{Is} \geq \mu_{mj(o-1)t}^{Ie}, \quad \begin{array}{l} \forall m \in M, \forall j \in J, \forall t \in T, \\ \forall o \in O_{mt}, o > 1. \end{array} \quad (4.19)$$

$$\mu_{mjto}^{II_s} \geq \mu_{mj(o-1)t}^{II_e}, \quad \begin{array}{l} \forall m \in M, \forall j \in J, \forall t \in T, \\ \forall o \in O_{mt}, o > 1. \end{array} \quad (4.20)$$

$$\mu_{mj1t}^{Is} \geq \mu_{mi_0t}^{Ie} + TC_{ij}^I - Mgde(1 - Z_{mijt}), \quad \begin{array}{l} \forall m \in M, \forall i, j \in J, \forall t \in T, \\ i \neq j, j \neq i_0. \end{array} \quad (4.21)$$

$$\mu_{mj1t}^{II_s} \geq \mu_{mi_0t}^{II_e} + TC_{ij}^{II} - Mgde(1 - Z_{mijt}), \quad \begin{array}{l} \forall m \in M, \forall i, j \in J, \forall t \in T, \\ i \neq j, j \neq i_0. \end{array} \quad (4.22)$$

$$\mu_{mjto}^{II_s} \geq \mu_{mjto}^{Ie}, \quad \begin{array}{l} \forall m \in M, \forall j \in J, \forall t \in T, \\ \forall o \in O_{mt}, o > 1. \end{array} \quad (4.23)$$

$$\begin{aligned} \mu_{mjt0}^{Is} &\geq \mu_{mj(o-1)t}^{IIs} + Mgde(1 - Y_{mjto}), & \forall m \in M, \forall j \in J, \forall t \in T, \\ & & \forall o \in O_{mt}, o > 1. \end{aligned} \quad (4.24)$$

$$\mu_{mjt0}^{IIe} \leq Cap_{mt}, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (4.25)$$

$$\mu_{mjt0}^{Ie} \leq Cap_{mt}, \quad \forall m \in M, \forall j \in J, \forall t \in T. \quad (4.26)$$

$$\begin{aligned} \mu_{mjt0}^{IIs} - \mu_{mjt1}^{Is} - Mgde(1 - Y_{mjto}) &\leq Mgde \omega_{mjt01}^I + TP_{max}, & \forall m \in M, \forall j \in J, \forall t \in T, \\ & & \forall o \in O_{mt}, o > 1. \end{aligned} \quad (4.27)$$

$$\begin{aligned} W_{mjt0k}^I &\geq \omega_{mjt0k}^I - \omega_{mjt(o-1)k}^I, & \forall m \in M, \forall j \in J, \forall t \in T, \\ & & \forall o \in O_{mt}, \forall k \in Q_{mt}^I, o > 1. \end{aligned} \quad (4.28)$$

$$\begin{aligned} \omega_{mjt1k}^I &= 0, & \forall m \in M, \forall j \in J, \\ & & \forall t \in T, \forall k \in Q_{mt}^I. \end{aligned} \quad (4.29)$$

$$\begin{aligned} \mu_{mjt0}^{IIe} - \mu_{mjt1}^{IIs} - Mgde(1 - Y_{mito}) &\leq Mgde \omega_{mjt01}^{II} + TL_{max}, & \forall m \in M, \forall j \in J, \forall t \in T, \\ & & \forall o \in O_{mt}, o > 1. \end{aligned} \quad (4.30)$$

$$\begin{aligned} W_{mjt0l}^{II} &\geq \omega_{mjt0l}^{II} - \omega_{mjt(o-1)l}^{II}, & \forall m \in M, \forall j \in J, \forall t \in T, \\ & & \forall o \in O_{mt}, \forall l \in Q_{mt}^{II}, o > 1. \end{aligned} \quad (4.31)$$

$$\begin{aligned} \omega_{mjt1k}^{II} &= 0, & \forall m \in M, \forall j \in J, \\ & & \forall t \in T, \forall k \in Q_{mt}^I. \end{aligned} \quad (4.32)$$

$$\begin{aligned} \mu_{mjt0}^{IIs} - (\psi_{mjt0'(k-1)}^I + LT^I) - Mgde(2 - W_{mjt0'(k-1)}^I - Y_{mjto}) &\leq Mgde \omega_{mjt0k}^I + TP_{max}, \\ \forall m \in M, \forall j \in J, \forall t \in T, \forall o, o' \in O_{mt}, \forall k \in Q_{mt}^I, o > o', k > 1. \end{aligned} \quad (4.33)$$

$$\begin{aligned} \psi_{mjt0k}^I &\leq Mgde W_{mjt0k}^I, & \forall m \in M, \forall j \in J, \forall t \in T, \\ & & \forall o \in O_{mt}, \forall k \in K_{mt}^I. \end{aligned} \quad (4.34)$$

$$\mu_{mjt0}^{IIe} - (\psi_{mjt0'(l-1)}^{II} + LT^{II}) - Mgde(2 - W_{mjt0'(l-1)}^{II} - Y_{mjto}) \leq Mgde \omega_{mjt0l}^{II} + TL_{max},$$

$$\forall m \in M, \forall j \in J, \forall t \in T, \forall o, o' \in O_{mt}, \forall l \in Q_{mt}^{II}, o > o', l > 1. \quad (4.35)$$

$$\psi_{mjtol}^{II} \leq Mgde W_{mjtol}^{II}, \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.36)$$

$$\forall o \in O_{mt}, \forall l \in Q_{mt}^{II}.$$

$$\mu_{mjto}^{Is} \geq \psi_{mjtok}^I + LT^I W_{mjtok}^I - Mgde (1 - W_{mjtok}^I), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.37)$$

$$\forall o \in O_{mt}, \forall k \in Q_{mt}^I.$$

$$\mu_{mjto}^{Is} \leq \psi_{mjtok}^I + LT^I W_{mjtok}^I - Mgde (1 - W_{mjtok}^I), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.38)$$

$$\forall o \in O_{mt}, \forall k \in Q_{mt}^I.$$

$$\psi_{mjtok}^I \geq \mu_{mjt(o-1)}^{Is} - Mgde (1 - W_{mjtok}^I), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.39)$$

$$\forall o \in O_{mt}, \forall k \in K_{mt}^I, o > 1.$$

$$\mu_{mjto}^{IIs} \geq \psi_{mjtol}^{II} + LT^{II} W_{mjtol}^{II} - Mgde (1 - W_{mjtol}^{II}), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.40)$$

$$\forall o \in O_{mt}, \forall l \in Q_{mt}^{II}.$$

$$\mu_{mjto}^{IIs} \leq \psi_{mjtol}^{II} + LT^{II} W_{mjtol}^{II} - Mgde (1 - W_{mjtol}^{II}), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.41)$$

$$\forall o \in O_{mt}, \forall l \in Q_{mt}^{II}.$$

$$\psi_{mjtok}^{II} \geq \mu_{mjt(o-1)}^{IIs} - Mgde (1 - W_{mjtok}^{II}), \quad \forall m \in M, \forall j \in J, \forall t \in T, \quad (4.42)$$

$$\forall o \in O_{mt}, \forall l \in Q_{mt}^{II}, o > 1.$$

$$X_{mjto}, I_{jt}^+, I_{jt}^-, V_{mjt}, \mu_{mjto}^{Is}, \mu_{mjto}^{Ie}, \mu_{mjto}^{IIs}, \mu_{mjto}^{IIs}, \psi_{mjtok}^I, \psi_{mjtol}^{II} \geq 0;$$

$$W_{mjtok}^I, W_{mjtol}^{II} \in \mathbb{Z}_+; \quad Z_{mijt}, Y_{mjto}, \omega_{mjtok}^I, \omega_{mjtol}^{II} \in \{0, 1\}$$

$$\forall m \in M, \forall i, j \in J, \forall t \in T, \forall o \in O_{mt}, \forall k \in Q_{mt}^I, \forall l \in Q_{mt}^{II}. \quad (4.43)$$

APÊNDICE E – Ajuste de parâmetros do CPLEX

Para resolução das instâncias dos modelos desse trabalho foi utilizado o *solver* CPLEX na versão 12.6.1, que é um pacote de otimização eficiente para resolução de problemas gerais de otimização e que tem sido um dos mais utilizados nas pesquisas dos últimos anos. Apesar do CPLEX em sua versão *default* ter um bom desempenho, alguns trabalhos tem mostrado que a adequação de parâmetros desse *solver* pode apresentar melhores resultados, dependendo do tipo de problema e das instâncias que estão sendo resolvidas (FERREIRA; MORABITO; RANGEL, 2010; ILOG, 2014). Hutter, Hoos e Leyton-Brown (2009) apresentou um método de calibração automática para o CPLEX, porém nem todos os parâmetros desejáveis são analisados durante esse processo de calibração. Além disso o tempo utilizado para fazer a calibração automática é muito alto, comparado a calibração manual. Sendo assim, nesse trabalho a calibração do CPLEX é feita manualmente.

O primeiro parâmetro modificado foi o processamento em paralelo (*parallel mode*). Todos os testes computacionais deste trabalho foram realizados com processamento em paralelo. Com essa estratégia ativada, o CPLEX utiliza diversos núcleos de processamento do computador para tentar melhorar seu desempenho. No método *branch-and-cut*, a otimização em paralelo é aplicada na ramificação e na solução dos subproblemas no nó raiz e nos nós da árvore, utilizando diferentes algoritmos nos diferentes núcleos (ILOG, 2014). A otimização em paralelo pode ainda ser testada em modo determinístico ou oportunista, dependendo do valor do parâmetro *threads* escolhido. No modo determinístico, várias execuções do mesmo problema com os mesmos parâmetros retornarão soluções idênticas. No modo oportunista, nem sempre serão encontradas as mesmas soluções e o mesmo tempo computacional em várias execuções. Essa característica advém das diferenças no tempo dedicado às tarefas (operações) de cada estratégia testada e/ou na ordem em que as mesmas são executadas nos núcleos da máquina. Para resolver os subproblemas, por exemplo, podem ser usados diversos tipos de métodos de solução em cada execução, por exemplo, os algoritmos primal e dual simplex, e o algoritmo de pontos interiores (*barrier*). Os testes computacionais foram realizados no modo oportunista.

A estratégia de geração de planos de corte (*mipcuts*), que implica na inclusão de outras equações e inequações no modelo, foi testada no modo *default* que é ativa, e também no modo desligada. Esses cortes são inseridos com o intuito de melhorar os limitantes do problema e a convergência dos métodos, mas em algumas situações podem deixar o problema intratável, devido ao grande número de restrições inseridas no

problema (FERREIRA; MORABITO; RANGEL, 2010). Para o modelo MDSL-2E-LT, testes mostraram que os planos de corte ligados melhoram as soluções. Assim, em todos os outros testes desse trabalho os planos de corte gerados pelo CPLEX estarão ativos.

O pré-solver ajuda a simplificar o problema eliminando redundâncias de variáveis e restrições. Porém, em alguns casos, o problema reduzido pode ser mais difícil de ser resolvido, mesmo sendo menor. Assim, testes também foram realizados com o pré-solver ligado e desligado. Para o modelo MDSL-2E-LT os testes apontaram que a melhor opção é manter o pré-solver ligado.

Logo, para todos os testes realizados neste trabalho utilizou-se o processamento em paralelo em modo oportunista, os planos de corte ativos e o pré-solver ligado. Para testar diversas estratégias (combinações de parâmetros), variações foram realizadas nos parâmetros que controlam a maneira de selecionar variáveis, nós e a ênfase do MIP. Também foram realizadas variações na frequência de aplicação das heurísticas RINS e *feasibility pump*.

A Tabela 50 resume as estratégias CPLEX testadas. Lembrando que já estão considerados o processamento em paralelo e planos de corte e pré-solver ligados.

O funcionamento dos parâmetros modificados são detalhados a seguir:

- Três estratégias de seleção de variáveis foram testadas (*varselect*). Com a regra da máxima infactibilidade, o CPLEX escolhe a variável com a maior parte fracionária para ramificar. Isso força mudanças maiores no início da árvore de busca. Para a regra da mínima infactibilidade, a variável com a menor parte fracionária é escolhida, o que pode levar a uma primeira solução inteira mais rapidamente, mas geralmente será mais lento para encontrar a solução inteira ótima. A última regra testada é a de ramificação forte (*strong branching*) em que inicialmente são escolhidos diferentes ramos pra resolver subproblemas por diferentes escolhas de variáveis de ramificação. A variável que produzir o melhor resultado é então escolhida. Essa estratégia requer mais tempo em cada nó, mas geralmente precisa de menos nós para resolver o problema (ILOG, 2014). No modo *default* a escolha é feita por uma heurística interna baseada no problema e no progresso do processo.
- Para seleção do nó de ramificação (*nodeselect*) duas estratégias foram testadas. A busca em profundidade, em que é escolhido o nó criado recentemente, e a ramificação pelo nó com melhor limitante (*bound*), em que é escolhido o nó com o melhor valor de função objetivo associado na relaxação linear.
- O método *branch-and-cut* foi testado com duas ênfases (parâmetro *mipemphasis*): factibilidade e otimalidade. A factibilidade enfatiza fortemente a busca por novas

Tabela 50 – Estratégias CPLEX testadas.

	Seleção de variáveis ^a	Seleção de nós ^b	Ênfase MIP ^c	Heurística RINS ^d	Heurística <i>feasibility pump</i> ^e
CPX1	1				
CPX2	-1				
CPX3	3				
CPX4		2			
CPX5		3			
CPX6			1		
CPX7			2		
CPX8				-1	
CPX9				10	
CPX10				100	
CPX11	-1			10	
CPX12		3		10	
CPX13	-1	3	2	10	
CPX14	-1	3	2	10	
CPX15	-1	3	2	10	-1
CPX16	-1	3	2	10	1
CPX17	-1	3	2	10	2

^a 1: máxima inactibilidade; -1: mínima inactibilidade; 3: ramificação forte.

^b 2: nó criado recentemente; 3: nó com melhor limitante.

^c 1: ênfase em factibilidade; 2: ênfase em otimalidade.

^d -1: desligada; 10: aplicada a cada 10 nós; 100: aplicada a cada 100 nós.

^e -1: desligada; 1: qualquer solução factível; 2: boa solução factível.

soluções factíveis e pode ser apropriada para modelos difíceis, em que a prova de otimalidade é difícil de ser alcançada (ILOG, 2014). No padrão *default*, o CPLEX faz um balanço entre buscar soluções factíveis e provar otimalidade.

- A heurística RINS (*Relaxation Induced Neighborhood Search*) explora a vizinhança da solução incumbente com o objetivo de melhorá-la e diversificar a busca (DANNA; ROTHBERG; PAPE, 2005). A frequência de utilização (a cada quantos nós é aplicada) dessa heurística é definida automaticamente quando o CPLEX está no modo *default*. Entretanto, a aplicação dessa heurística com muita frequência pode resultar em resolver subproblemas MIP similares sem obtenção de melhorias. Por outro lado, aplicar a heurística RINS em intervalos de nós muito grandes pode reduzir o benefício de sua utilização. Cabe também testar se a RINS desligada pode produzir algum efeito no problema. Nesse estudo foram testadas as diretivas para a RINS (*rinsheur*) desligada, utilizada a cada 10 e a cada 100 nós.
- A heurística *feasibility pump* (FISCHETTI; GLOVER; LODI, 2005) busca por soluções factíveis para problemas NP-difícil. A diretiva *default* permite ao CPLEX decidir se vai ou não aplicar a heurística. Existem ainda as opções de desligar a

heurística, procurar por qualquer solução factível, ou procurar por uma solução factível com um bom valor de função objetivo. Essas três estratégias foram testadas para o modelo MDSL-2E-LT.

Ajustes na configuração dos parâmetros do CPLEX podem representar melhorias na resolução de determinados modelos matemáticos e determinadas instâncias (HUTTER; HOOS; LEYTON-BROWN, 2009). Entretanto, a calibragem do CPLEX não é uma tarefa fácil, devido ao grande número de opções de parâmetros que podem ser modificados e ao grande número de combinações dessas diretivas que podem ser feitas. Sendo assim, nesse trabalho o objetivo não é esgotar todas as possibilidades de configuração para encontrar a melhor delas, mas sim analisar algumas estratégias alternativas mais eficientes para o problema estudado.

APÊNDICE F – Experimentos computacionais para as heurísticas de decomposição para caso independente da sequência

Nesta seção realiza-se um experimento computacional para analisar o desempenho das heurísticas de decomposição HE1 e HE2, com suas variações do tipo A e do tipo B, entretanto com as técnicas de redução de capacidade feitas como nos trabalhos de [Toscano, Ferreira e Morabito \(2015\)](#) e [Toscano, Ferreira e Morabito \(2017\)](#), e considerando tempos e custos de troca independentes da sequência (IS): HE1A-IS, HE1B-IS, HE2A-IS e HE2B-IS.

Para analisar essas heurísticas, foram utilizados os mesmos dados das instâncias R1, R2, R3 e R4, descritas no Capítulo 6, porém com os custos de troca entre os itens nesse caso todos iguais ao custo de uma limpeza temporal, ou seja, 1 u.m. Os tempos de troca entre os itens também foi considerado como o tempo de realização das limpezas temporais (LT^I e LT^{II} , para tanque e linha, respectivamente). Além disso, para analisar essas instâncias o objetivo foi reduzir o número de CIPs nas linhas e nos tanques, sejam CIPs por limpezas temporais ou por troca de itens. Nessas instâncias também considera-se que toda a demanda deve ser atendida dentro do horizonte de planejamento, logo não é permitido atraso no último período, ou seja, considera-se $I_{jT}^- = 0$. Essas instâncias com esses valores alterados são nomeadas por I1, I2, I3 e I4. O computador, a linguagem de modelagem e o *solver* também são os mesmos descritos no Capítulo 6.

Com o objetivo de testar diversos cenários, outras instâncias foram geradas a partir de I1.01, I2.01 e I3.01, fazendo modificações em alguns parâmetros e criando 14 classes diferentes para essas instâncias. A descrição dessas modificações e o conjunto completo de instâncias são apresentados na Tabela 51.

A Tabela 52 apresenta os valores da função objetivo obtidos pelas heurísticas HE1A-IS, HE1B-IS, HE2A-IS e HE2B-IS para cada instância. Nesta tabela também são apresentados os valores da função objetivo e do *gap* encontrados pelo modelo MDSL-2E-LT para as mesmas instâncias. Para que o valor da função objetivo do modelo fosse comparável com as heurísticas, o termo que contabiliza as trocas do modelo MDSL-2E-LT ($\sum_{m \in M} \sum_{t \in T} \sum_{j \in J} \sum_{\substack{i \in J \\ i \neq j}} C_{ij} Z_{mijt}$) foi multiplicado por 2. Isto porque as heurísticas minimizam número de CIPs e uma troca de sabor acarreta em dois CIPs, um na linha e um no tanque preparatório.

Tabela 51 – Instâncias Testadas.

Classe	Instâncias	Modificações
01	I1.01 / I2.01 / I3.01	Dados da Empresa
02	I1.02 / I2.02 / I3.02	As velocidades das máquinas foram dobradas
03	I1.03 / I2.03 / I3.03	As velocidades das máquinas foram triplicadas
04	I1.04 / I2.04 / I3.04	As velocidades das máquinas foram dobradas e Cap_{mt} foi reduzida em 50%
05	I1.05 / I2.05 / I3.05	Cap_{mt} foi reduzida em 10%
06	I1.06 / I2.06 / I3.06	Cap_{mt} foi reduzida em 15%
07	I1.07 / I2.07 / I3.07	Cap_{mt} foi reduzida em 20%
08	I1.08 / I2.08 / I3.08	Período de menor demanda foi considerado com capacidade nula ($Cap_{mt} = 0$ para todo m , e para t com a menor demanda)
09	I1.09 / I2.09 / I3.09	Período de menor demanda (t) com $Cap_{1t} = 0$
10	I1.10 / I2.10 / I3.10	Período de menor demanda (t) com $Cap_{2t} = 0$
11	I1.11 / I2.11 / I3.11	Período de maior demanda (t), as capacidades Cap_{mt} foram reduzidas em 50%, para todo m
12	I1.12 / I2.12 / I3.12	B_{max_j} foi reduzido em 25%
13	I1.13 / I2.13 / I3.13	B_{max_j} foi dobrado
14	I1.14 / I2.14 / I3.14	B_{max_j} foi dobrado e foi considerada uma única máquina com velocidade o dobro da máquina mais veloz
15	I4	Dados da Empresa considerando 3 meses (13 períodos)

As instâncias marcadas com “–” na coluna do modelo MDSL-2E-LT indicam que o CPLEX não obteve nenhuma solução factível no limite de tempo de 3.600 segundos. Nas colunas das heurísticas, as instâncias sinalizadas com “*” significam que o modelo original é infactível, antes de qualquer redução de capacidade. Em outras palavras, para as instâncias I3.06, I2.07, I3.07, I1.08, I2.08 e I3.08 as heurísticas H2E-IS apresentaram solução infactível na primeira iteração de resolução dos modelos, antes da redução da capacidade. Isso indica, na realidade, que essas instâncias são infactíveis para o segundo estágio, ou seja, para esses cenários o problema não tem solução do ponto de vista das linhas. Cabe lembrar que essas instâncias foram criadas com o intuito de analisar cenários e a factibilidade das mesmas não foi avaliada antes dos testes. Os resultados apresentados com “**” indicam que em alguma iteração da heurística o modelo tornou-se infactível devido às reduções de capacidade realizadas ao longo das iterações. Logo, nesse caso não se pode afirmar que as instâncias são infactíveis, pois o modelo original foi resolvido.

As heurísticas podem terminar sem solução factível, pois nos testes realizados é proibido atraso no último período ($I_{jT}^- = 0$).

Para as instâncias da classe 01 (instâncias reais), as heurísticas H2E-IS encontraram melhores soluções do que H1E-IS. Para as instâncias I1.01 e I2.01, foram obtidas soluções sem estoque e atraso, sendo que o valor da função objetivo apresentado na Tabela 52 para essas instâncias referem-se somente ao número de CIPs (limpezas temporais e trocas de sabor) realizados. A instância I4.01 foi criada com a junção das instâncias I1.01, I2.01 e I3.01 (com 13 semanas) no intuito de aumentar o tamanho dos problemas. Neste caso as heurísticas H2E-IS apresentaram os melhores resultados.

Tabela 52 – Soluções das heurísticas HE1-IS e HE2-IS e do modelo MDL-2E-LT.

Inst.	HE1-IS		HE2-IS		MDL-2E-LT	
	HE1A-IS Função Obj (u.m.)	HE1B-IS Função Obj (u.m.)	HE2A-IS Função Obj (u.m.)	HE2B-IS Função Obj (u.m.)	Função Obj (u.m.)	Gap (%)
I1.01	68,00	70,00	65,00	65,00	61,00	79,24
I2.01	523.419,33	2.635.155,67	83,00	83,00	61.327,00	99,98
I3.01	2.050.074,00	2.050.073,00	1.404.236,67	1.404.236,67	-	-
I4.01	10.541.888,67	6.136.896,67	3.450.747,02	3.413.247,02	-	-
I1.02	46,00	46,00	43,00	43,00	41,00	51,32
I2.02	61,00	61,00	49,00	49,00	50,00	52,00
I3.02	46,00	46,00	39,00	39,00	36,00	48,78
I1.03	45,00	45,00	38,00	38,00	-	-
I2.03	61,00	61,00	45,00	45,00	51,00	56,83
I3.03	38,00	38,00	38,00	38,00	39,00	54,63
I1.04	1.744.583,33	**	1.198.832,41	3.554.183,02	41,00	39,44
I2.04	**	**	622.517,47	631.969,20	-	-
I3.04	**	**	12.678.858,73	14.578.322,54	-	-
I1.05	32.669,00	68,00	67,00	67,00	173.717,00	99,99
I2.05	**	**	1.154.694,45	1.129.694,45	-	-
I3.05	**	**	11.762.572,00	11.762.573,00	-	-
I1.06	1.694.670,00	**	1.031.238,95	1.613.255,24	1.053.632,67	100,00
I2.06	**	**	2.913.671,53	2.926.171,53	-	-
I3.06	**	**	*	*	-	-
I1.07	**	**	7.942.568,89	7.895.152,20	-	-
I2.07	**	**	*	*	-	-
I3.07	**	**	*	*	-	-
I1.08	**	**	*	*	-	-
I2.08	**	**	*	*	-	-
I3.08	**	**	*	*	-	-
I1.09	438.472,00	438.471,00	310.200,33	310.200,33	364.374,00	100,00
I2.09	**	**	2.958.328,24	2.958.328,24	-	-
I3.09	**	**	**	**	-	-
I1.10	65,00	66,00	65,00	65,00	30,00	36,70
I2.10	523.419,33	2.635.155,67	85,00	85,00	-	-
I3.10	7.066.736,67	**	6.810.365,41	6.519.742,41	-	-
I1.11	2.037.139,67	1.862.133,67	1.435.647,33	1.435.647,33	-	-
I2.11	**	**	4.429.855,17	4.429.203,62	-	-
I3.11	**	**	43.762.123,37	43.747.291,83	-	-
I1.12	65,00	68,00	66,00	66,00	-	-
I2.12	717.743,00	498.994,00	84,00	84,00	5.073,83	99,74
I3.12	3.975.077,00	3.975.077,00	2.988.462,14	2.986.608,20	-	-
I1.13	63,00	64,00	66,00	66,00	55,00	75,21
I2.13	3.681.820,33	3.215.152,67	86,00	86,00	73,00	81,72
I3.13	4.600.072,00	8.766.739,67	1.404.236,67	1.404.236,67	612.549,44	99,99
I1.14	34,00	34,00	36,00	36,00	19,00	22,38
I2.14	43,00	43,00	44,00	44,00	38,00	19,29
I3.14	32,00	32,00	34,00	34,00	21,00	35,56

* Não há solução factível. O modelo original é infactível.

** Não há solução factível. Em alguma iteração o modelo tornou-se infactível.

- Não foi encontrada solução factível no tempo de 3.600 segundos.

Nas instâncias das classes 02 e 03 o gargalo de produção está no estágio I. Neste caso, as heurísticas HE2-IS apresentaram as melhores soluções. Como as heurísticas HE1-IS resolvem o estágio I otimamente via MIP, poderia-se esperar que as heurísticas HE1-IS tivesse um melhor desempenho para essas instâncias. De fato, na primeira iteração, quando ainda não foram realizadas reduções de capacidade, o modelo MDL-P apresenta melhores soluções do que o MDL-L. Mas durante a realização da sincronia, à medida que as capacidades vão sendo reduzidas, os valores das reduções podem ser diferentes para

HE1-IS e HE2-IS, e portanto, não existe mais garantia que as soluções encontradas pelo MDL-P serão melhores do que as encontradas pelo MDL-L.

As heurísticas HE2-IS obtiveram mais soluções factíveis do que as heurísticas HE1-IS para as instâncias da classe 04. A heurística HE1-IS encontrou menos soluções factíveis, pois como a capacidade Cap_{mt} foi reduzida em 50%, o espaço de solução modelo MDL-P tornou-se mais restrito, e a medida que Cap_{mt} foi reduzida ao longo das iterações, o modelo tornou-se infactível. Por outro lado, nas heurísticas HE2-IS, apesar da redução da capacidade, o parâmetro S_m (velocidade de envase) foi dobrado, o que fez com que o modelo MDL-P se tornasse menos restrito do que o MDL-L.

Reduzindo a capacidade em 10% e 15%, para as instâncias das classes 05 e 06, respectivamente, as heurísticas HE2-IS apresentaram mais soluções factíveis do que HE1-IS. Para as instâncias da classe 07, em que a capacidade é reduzida em 20%, foi possível obter solução factível apenas para a instância I1.07. Na verdade os resultados indicam que, nesse caso, com a redução de 20% da capacidade o problema torna-se infactível do ponto de vista das linhas (instâncias das heurísticas HE2 sinalizadas com “ * ”). Ou seja, não é possível atender a demanda sem atraso no último período. Na classe 08, quando assume-se que o período com menor demanda tem capacidade nula, o problema também torna-se infactível do ponto de vista das linhas. Para a classe 09, nenhuma das heurísticas retornaram solução factível para a instância I3.09. E para as instâncias I1.09 e I2.09, as soluções factíveis que foram encontradas apresentam estoque e atraso. Em contraste, para a classe 10 todas as heurísticas apresentaram soluções factíveis. Para essa classe, o modelo obteve solução factível apenas para a instância I1.10, mas com custo quase metade do custo das heurísticas.

Para as instâncias da classe 11, com 50% de redução de capacidade, o gargalo está no estágio II. As heurísticas HE2-IS encontram soluções factíveis para todas as instâncias, enquanto que HE1-IS, apenas para a instância I1.11. Com o valor de Cap_{mt} muito restrito, as heurísticas HE2-IS tornam-se infactíveis ao longo das iterações. O modelo não encontrou nenhuma solução factível para as instâncias dessa classe.

As instâncias da classe 12 e 13 foram criadas para analisar o efeito da diminuição e aumento da capacidade dos tanques preparatórios (aumento e diminuição dos lotes no estágio I). Com a diminuição de 25% (classe 12), soluções factíveis foram obtidas por todas as heurísticas e os resultados não foram muito diferentes dos encontrados na classe 01. Entretanto, quando a capacidade física dos tanques preparatórios é dobrada (classe 13), para as instâncias I2.13 e I3.13, as heurísticas HE1-IS encontram soluções piores do que para as instâncias I2.12 e I3.1, onde a capacidade dos tanques foi reduzida. A primeira vista, pode parecer estranho que ao aumentar a capacidade dos tanques, a solução tenha piorado para essas duas instâncias. Porém, isso acontece porque na classe 13 o modelo do primeiro estágio, MDL-P, dimensiona lotes maiores. Quando esses lotes são

distribuídos nas linhas, no segundo passo da heurística, a capacidade utilizada ($CapUtil_{mt}$) ultrapassa muito a capacidade disponível ($CapDisp_{mt}$), fazendo com que sejam feitas grandes reduções ($\delta = CapUtil_{mt} - CapDisp_{mt}$) em Cap_{mt} . Com isso, o modelo MDL-P torna-se muito restritivo a cada iteração, forçando soluções com mais estoque e atraso.

A classe 14 simula uma situação na qual os dois tanques preparatórios são trocados por um único tanque maior. Essa instância foi criada com base em informações obtidas em uma das visitas a Fábrica A. Os tomadores de decisão estavam avaliando se essa mudança do sistema de produção poderia acarretar em melhorias para a produção. A ideia é que fosse transferida mais bebida ao mesmo tempo para as linhas, dando uma maior autonomia para estas e liberando o tanque maior mais rapidamente para o preparo de um novo lote, e com isso evitando esperas. Os resultados das heurísticas mostraram que trata-se de uma alternativa viável e interessante para os dados da Fábrica A, pois para as instâncias I1.10, I2.10 e I3.10, todas as heurísticas encontraram soluções factíveis sem nenhum estoque e/ou atraso.

As heurísticas, em geral, encontraram mais soluções factíveis quando comparadas com o modelo MDSL-2E-LT. O modelo tem dificuldades de encontrar soluções iniciais inteiras, devido à exigência de não haver atraso no último período. Os resultados da Tabela 52 mostram que as heurísticas HE2-IS apresentam mais soluções factíveis do que as heurísticas HE1-IS. Ambas HE2A-IS e HE2B-IS obtiveram soluções factíveis para 36 das 43 instâncias, 9 a mais do que HE1A-IS e 12 a mais do que HE1B-IS.

Na Tabela 52, as soluções destacadas em negrito são os melhores valores encontrados entre as quatro heurísticas. Quando o modelo apresenta uma solução melhor que as heurísticas, essas soluções aparecem sublinhadas. Entre as heurísticas, a HE2A-IS foi a que apresentou o melhor desempenho com relação a função objetivo, pois encontrou melhores resultados do que as outras heurísticas para a maioria das instâncias. A média de tempo despendido para cada instância foi de 127,17 segundos na heurística HE1A-IS, 251,07 na HE1B-IS, 8,80 na HE2A-IS e 8,82 na HE2B-IS. Para o modelo, foram utilizados o tempo total disponível de 3.600 segundos para todas as instâncias.

Logo, entre as quatro heurísticas testadas, a HE2A-IS é a que apresenta melhor desempenho em todos os quesitos: número de soluções factíveis, função objetivo e tempo. Quando comparada com o modelo, a HE2A-IS apresenta algumas soluções piores do que o modelo, como por exemplo, para a instância I1.04, em que o modelo exibe uma solução sem nenhum estoque e nem atraso. Por outro lado também, para a instância I1.05 a heurística apresenta solução sem estoque e atraso, enquanto o modelo exibe uma solução com 99% de *gap* e com estoque. A HE2A-IS é muito rápida quando comparada ao modelo e apresenta mais soluções factíveis e portanto é uma boa alternativa.

A Tabela 53 apresenta uma comparação das soluções da HE2A-IS com a solução da empresa adaptada. Na comparação com a heurística H2EA-IS as trocas foram contadas

como CIPs em ambos os estágios.

Tabela 53 – Resultados da heurística HE2A-IS comparados com a empresa.

Inst.	HE2A-IS				Empresa			
	Estoque (unid.)	Atraso (unid.)	CIP (unid.)	Função objetivo	Estoque (unid.)	Atraso (unid.)	CIP (unid.)	Função objetivo
I1.01	0	0	65	65,00	0	0	71	71,00
I2.01	0	0	83	83,00	0	0	90	90,00
I3.01	0	20.500	74	1.404.236,00	0	12.250	70	1.225.070,00

A heurística encontrou soluções melhores do que a empresa para duas (I1.01 e I2.01) das três instâncias, usando os parâmetros de custo aqui estimados. A solução para a instância I3.01 está próxima, com uma diferença de 8.250 unidades de estoque a mais para a heurística. De maneira geral, a heurística consegue produzir bons programas de produção para a empresa, em um tempo muito menor do que os programadores da empresa, uma vez que a heurística leva em torno de 8 segundos e os programadores da empresa levam duas horas para encontrar um plano de produção factível.

APÊNDICE G – Tabelas dos resultados computacionais

Nesta seção são detalhados alguns resultados computacionais dos testes apresentados e discutidos no Capítulo 6.

Tabela 54 – Soluções estratégias CPLEX - função objetivo (u.m.).

	S1	S1_1	M1	R1	R2	R3	R4
CPX1	14,01	14,01	25,02	49,00	307.436,33	1.862.547,00	884.021.818,33
CPX2	14,01	14,01	25,02	31.037,33	37.686,33	1.695.212,67	1.100.989.008,67
CPX3	14,01	14,01	25,02	55.602.828,67	58.415.763,33	46.614.665,00	2.717.200.000,00
CPX4	14,01	14,01	25,02	47,00	86.242,50	1.741.843,67	2.280.879.861,67
CPX5	14,01	14,01	25,02	46,00	53,00	1.987.549,00	2.295.995.330,00
CPX6	14,01	14,01	25,02	4.143.867,67	1.201.993,67	5.211.786,33	2.689.949.134,00
CPX7	14,01	14,01	26,02	49,00	494.712,50	2.402.963,67	2.717.200.000,00
CPX8	14,01	14,01	25,02	23.717.134,67	51.123.571,33	36.901.070,83	1.891.724.009,67
CPX9	14,01	14,01	25,02	43,00	87.984,33	1.987.562,00	234.167.253,33
CPX10	14,01	14,01	25,02	42,00	769.582,33	2.236.711,67	187.848.292,83
CPX11	14,01	14,01	25,02	123.523,51	305.134,56	5.489.156,68	384.407.679,74
CPX12	14,02	14,02	25,02	41,17	50,18	1.237.545,00	173.182.854,00
CPX13	14,02	14,02	25,02	46,00	883.957,17	2.494.631,33	582.048.875,67
CPX14	14,02	14,02	25,02	41,00	594.932,00	1.237.546,00	410.605.758,67
CPX15	14,02	14,02	25,02	43,00	559.938,33	1.237.547,00	445.785.847,00
CPX16	14,02	14,02	25,02	43,00	6.138.015,67	1.987.549,00	551.197.171,33
CPX17	14,02	14,02	25,02	45,00	95.357,00	1.254.215,67	575.314.619,67

Tabela 55 – Soluções estratégias CPLEX - *gap* (%).

	S1	S2	M1	R1	R2	R3	R4
CPX1	11,89	1,00	42,75	82,07	100,00	100,00	100,00
CPX2	27,71	1,00	47,48	99,97	99,97	100,00	100,00
CPX3	14,27	0,98	43,40	100,00	100,00	100,00	100,00
CPX4	10,48	0,90	39,31	76,74	99,99	100,00	100,00
CPX5	1,00	0,97	39,97	84,43	77,93	100,00	100,00
CPX6	10,76	1,00	42,07	100,00	100,00	100,00	100,00
CPX7	6,47	0,37	39,37	75,75	100,00	100,00	100,00
CPX8	7,62	0,95	37,30	100,00	100,00	100,00	100,00
CPX9	35,02	0,14	42,96	76,21	99,99	100,00	100,00
CPX10	17,86	0,63	39,93	73,62	100,00	100,00	100,00
CPX11	32,37	7,16	45,93	99,99	100,00	100,00	100,00
CPX12	21,45	0,04	41,78	75,82	79,41	100,00	100,00
CPX13	35,02	26,47	58,03	84,78	100,00	100,00	100,00
CPX14	31,92	14,23	43,60	74,29	100,00	100,00	100,00
CPX15	37,39	13,77	43,80	78,58	100,00	100,00	100,00
CPX16	28,99	0,97	44,40	83,87	100,00	100,00	100,00
CPX17	27,38	13,83	42,43	82,46	99,99	100,00	100,00

Tabela 56 – Soluções heurísticas *relax-and-fix* - função objetivo (u.m.).

	S1	S2	M1	R1	R2	R3	R4	Média
RFX1	14,01	14,01	25,02	41,00	49,00	1.237.543,00	67.716.383,50	9.850.581,36
RFX2	14,01	14,01	25,02	40,00	26.511,33	612.545,00	71.777.955,33	10.345.300,67
RFX3	14,01	14,01	25,02	41,00	49,00	1.862.552,00	38.612.304,67	5.782.142,82
RFX4	14,01	14,01	25,02	40,00	49,00	1.196.377,33	11.001.671,00	1.742.598,62
RFX5	14,01	14,01	25,02	40,00	49,00	1.987.547,00	49.829.598,83	7.402.469,70
RFX6	14,01	14,01	25,02	40,00	7.538,00	1.131.502,33	6.421.770,83	1.080.129,17
RFX7	14,01	14,01	25,02	39,00	49,00	1.237.546,00	24.622.251,00	3.694.276,86
RFX8	14,01	14,01	25,02	41,00	137.278,00	1.950.045,00	9.108.275,17	1.599.384,60
RFX9	14,01	14,01	25,02	39,00	49,00	1.237.550,00	37.263.189,14	5.500.125,74
RFX10	14,01	14,01	25,02	137.814,33	1.845.931,00	4.319.293,00	1.507.872.186,33	216.310.753,96
RFX11	14,01	14,01	25,02	40,00	49,00	1.403.839,67	309.759.672,33	44.451.950,58
RFX12	14,01	14,01	25,02	191.281,53	939.423,85	640.255,51	2.187.964.601,52	312.819.373,64
RFX13	14,01	14,01	25,02	40,00	49,00	1.862.548,00	9.120.082,00	1.568.967,43
RFX14	14,01	14,01	25,02	40,00	48,00	1.237.546,00	146.375.382,17	21.087.581,32
RFX15	14,01	14,01	25,02	40,00	49,00	1.862.547,00	17.733.585,28	2.799.467,76
RFX16	14,01	14,01	25,02	39,00	49,00	1.237.546,00	9.590.069,17	1.546.822,32

Tabela 57 – Soluções heurísticas *relax-and-fix* - gap (%).

	S1	S2	M1	R1	R2	R3	R4	Média
RFX1	1,00	1,00	41,77	75,71	78,92	100,00	100,00	56,91
RFX2	1,00	1,00	41,77	75,10	99,96	100,00	100,00	59,83
RFX3	1,00	1,00	41,77	75,71	78,92	100,00	100,00	56,91
RFX4	1,00	1,00	41,77	75,10	78,92	100,00	100,00	56,83
RFX5	1,00	1,00	41,77	75,10	78,92	100,00	100,00	56,83
RFX6	1,00	1,00	41,77	75,10	99,86	100,00	100,00	59,82
RFX7	1,00	1,00	41,77	74,46	78,92	100,00	100,00	56,73
RFX8	1,00	1,00	41,77	75,71	99,99	100,00	100,00	59,92
RFX9	1,00	1,00	41,77	74,46	78,92	100,00	100,00	56,73
RFX10	1,00	1,00	41,77	99,99	100,00	100,00	100,00	63,39
RFX11	1,00	1,00	41,77	75,10	78,92	100,00	100,00	56,83
RFX12	1,00	1,00	41,77	99,99	100,00	100,00	100,00	63,39
RFX13	1,00	1,00	41,77	75,10	78,48	100,00	100,00	56,76
RFX14	1,00	1,00	41,77	75,71	78,92	100,00	100,00	56,91
RFX15	1,00	1,00	41,77	75,71	78,92	100,00	100,00	56,91
RFX16	1,00	1,00	41,77	74,46	78,48	100,00	100,00	56,67

Tabela 58 – Tempos computacionais heurísticas *relax-and-fix* - tempo em segundos.

	S1	S2	M1	R1	R2	R3	R4	Média
RFX1	609,58	735,13	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	2.770,11
RFX2	469,75	583,14	3.600,00	3.600,00	2.265,62	3.600,00	3.600,00	2.536,75
RFX3	332,67	401,67	3.074,09	3.600,00	3.600,00	1.109,28	3.600,00	2.491,21
RFX4	398,47	459,05	2.991,70	3.600,00	3.600,00	3.600,00	3.068,08	2.532,32
RFX5	440,94	566,08	2.857,97	3.600,00	3.600,00	1.156,78	3.600,00	2.346,36
RFX6	319,39	379,77	2.866,75	3.600,00	1.159,38	3.600,00	3.072,89	2.143,93
RFX7	647,70	777,45	2.851,81	3.600,00	3.600,00	1.104,00	3.600,00	2.797,55
RFX8	434,50	486,59	2.843,73	3.600,00	1.681,07	3.600,00	3.342,20	2.284,96
RFX9	492,44	559,48	2.981,53	3.600,00	3.600,00	1.182,22	3.600,00	2.326,01
RFX10	473,38	524,33	2.933,64	1.426,98	2.361,55	3.600,00	3.600,00	2.135,13
RFX11	575,58	697,59	2.931,33	3.600,00	3.600,00	1.414,88	3.600,00	2.352,56
RFX12	837,92	1.020,09	3.000,00	1.595,22	1.587,18	3.600,00	2.221,90	1.980,33
RFX13	659,92	728,80	3.600,00	3.600,00	3.600,00	1.222,67	3.600,00	2.430,20
RFX14	387,84	474,56	3.600,00	3.600,00	3.600,00	3.600,00	3.600,00	2.694,63
RFX15	593,08	669,98	3.600,00	3.600,00	3.600,00	1.157,92	3.600,00	2.403,00
RFX16	445,00	493,89	3.600,00	3.600,00	3.600,00	1.150,38	3.600,00	2.355,61

Tabela 59 – Soluções heurísticas *fix-and-optimize* com solução inicial fornecida por CPX12 - função objetivo (u.m.).

	S1	S2	M1	R1	R2	R3	R4	Média
CPX12+FXO1	14,02	14,01	25,07	40,62	49,66	1.237.544,66	152.553.872,17	21.970.222,88
CPX12+FXO2	14,02	14,01	25,07	40,62	86.144,32	1.237.544,67	284.660.883,59	40.854.952,33
CPX12+FXO3	14,02	14,01	25,07	38,67	202.948,62	1.237.548,79	7.460.307,28	1.271.556,64
CPX12+FXO4	14,02	14,01	25,07	41,60	48,66	1.237.546,75	24.619.918,78	3.693.944,13
CPX12+FXO5	14,02	14,01	25,07	39,65	9.998,44	1.265.253,03	30.009.119,88	4.469.209,16
CPX12+FXO6	14,02	14,01	25,07	40,59	51,74	1.265.252,02	222.639.723,08	31.986.445,79
CPX12+FXO7	14,02	14,01	25,07	57.610,38	1.615.385,15	1.265.253,03	49.110.128,22	7.435.489,98
CPX12+FXO8	14,02	14,01	25,07	39,62	177.553,84	737.545,81	50.321.676,49	7.319.552,69
CPX12+FXO9	14,02	14,01	25,07	39,62	50,70	3.003.799,81	10.366.287,31	1.910.032,93
CPX12+FXO10	14,02	14,01	25,07	20.877,97	51,73	612.546,80	24.768.321,67	3.628.835,89
CPX12+FXO11	14,02	14,01	25,07	42,65	231.735,57	1.862.547,76	12.432.007,60	2.075.198,10
CPX12+FXO12	14,02	14,01	25,07	44,68	57,86	1.237.545,72	19.474.140,83	2.958.834,60

Tabela 60 – Soluções heurísticas *fix-and-optimize* com solução inicial fornecida por CPX12 - *gap* (%).

	S1	S2	M1	R1	R2	R3	R4	Média
CPX12+FXO1	1,09	0,97	41,87	75,48	79,20	100,00	100,00	56,94
CPX12+FXO2	1,09	0,97	41,87	75,48	99,99	100,00	100,00	59,91
CPX12+FXO3	1,09	0,97	41,87	74,24	99,99	100,00	100,00	59,74
CPX12+FXO4	1,09	0,97	41,87	76,06	78,77	100,00	100,00	56,97
CPX12+FXO5	1,09	0,97	41,87	74,88	99,90	100,00	100,00	59,82
CPX12+FXO6	1,09	0,97	41,87	75,46	80,04	100,00	100,00	57,06
CPX12+FXO7	1,09	0,97	41,87	99,98	100,00	100,00	100,00	63,42
CPX12+FXO8	1,09	0,97	41,87	74,86	99,99	100,00	100,00	59,83
CPX12+FXO9	1,09	0,97	41,87	74,86	79,63	100,00	100,00	56,92
CPX12+FXO10	1,09	0,97	41,87	99,95	80,03	100,00	100,00	60,56
CPX12+FXO11	1,09	0,97	41,87	76,65	100,00	100,00	100,00	60,08
CPX12+FXO12	1,09	0,97	41,87	77,71	82,15	100,00	100,00	57,68

Tabela 61 – Tempos computacionais heurísticas *fix-and-optimize* com solução inicial fornecida por CPX12 - tempo em segundos.

	S1	S2	M1	R1	R2	R3	R4	Média
CPX+FXO1	1.681,95	762,47	2801,158	3.004,48	2.884,58	3.604,78	2.516,19	2.465,09
CPX+FXO2	1540,001	288,797	2.800,83	2.703,36	2.641,50	2.703,86	2.519,03	2.171,05
CPX+FXO3	1609,985	225,954	2800,799	2702,75	2220,874	2701,702	3.600,00	2.266,01
CPX+FXO4	1506,61	162,563	2800,812	2.702,91	2.644,42	2.405,34	2.364,94	2.083,94
CPX+FXO5	1560,735	226,015	2.772,77	2703,047	2315,593	2707,75	3538,372	2.260,61
CPX+FXO6	1.440,94	478,80	2.801,01	2703,015	2641,093	2.406,76	2.533,44	2.143,58
CPX+FXO7	1.690,80	484,34	2800,983	1.275,47	1.298,83	2.701,38	2.913,27	1.880,72
CPX+FXO8	1.526,84	576,08	2.800,75	2.702,77	1.356,69	2.407,97	2.522,33	1.984,77
CPX+FXO9	1.593,61	659,80	2.800,75	2.702,72	2.641,81	2.716,51	3.600,00	2.387,89
CPX+FXO10	1.507,98	330,37	2.800,77	1.274,05	2.641,14	2.405,53	2.408,41	1.909,75
CPX+FXO11	1.203,72	139,94	1.233,89	3.600,00	3.175,52	3.600,00	3.600,00	2.364,72
CPX+FXO12	1.203,31	1.200,78	1.230,94	3.600,00	3.600,00	3.600,00	3.456,97	2.556,00

Tabela 62 – Soluções heurísticas *fix-and-optimize* com solução inicial fornecida por RFX4 - função objetivo (u.m.).

	S1	S2	M1	R1	R2	R3	R4	Média
RFX4+FXO1	14,02	14,01	25,07	39,64	48,66	1.237.543,69	210.601.483,85	30.262.738,42
RFX4+FXO2	14,02	14,01	25,07	40,61	48,66	1.237.543,69	28.719.941,70	4.279.661,11
RFX4+FXO3	14,02	14,01	25,07	40,61	48,66	487.546,76	13.737.477,84	2.032.166,71
RFX4+FXO4	14,02	14,01	25,07	40,61	49,63	608.383,04	28.783.220,44	4.198.820,97
RFX4+FXO5	14,02	14,01	25,07	39,61	49,70	1.237.543,70	34.865.396,69	5.157.583,26
RFX4+FXO6	14,02	14,01	25,07	40,61	49,65	695.880,07	57.044.957,18	8.248.711,52
RFX4+FXO7	14,02	14,01	25,07	40,59	48,68	1.237.544,66	74.454.005,56	10.813.098,94
RFX4+FXO8	14,02	14,01	25,07	39,62	15.879,86	1.323.342,56	71.590.491,78	10.418.543,84
RFX4+FXO9	14,02	14,01	25,07	38,68	48,66	1.237.544,66	13.959.706,07	2.171.055,88
RFX4+FXO10	14,02	14,01	25,07	39,65	48,66	654.212,46	29.477.875,51	4.304.604,19
RFX4+FXO11	14,02	14,01	25,07	39,64	48,66	1.237.549,74	42.546.837,46	6.254.932,66
RFX4+FXO12	14,02	14,01	25,07	40,61	51,70	1.275.054,74	30.140.656,50	4.487.979,52

Tabela 63 – Soluções heurísticas *fix-and-optimize* com solução inicial fornecida por RFX14 - *gap* (%).

	S1	S2	M1	R1	R2	R3	R4	Média
RFX4+FXO1	1,09	0,97	41,87	74,88	78,77	100,00	100,00	56,80
RFX4+FXO2	1,09	0,97	41,87	75,48	78,77	100,00	100,00	56,88
RFX4+FXO3	1,09	0,97	41,87	75,48	78,77	100,00	100,00	56,88
RFX4+FXO4	1,09	0,97	41,87	75,48	79,18	100,00	100,00	56,94
RFX4+FXO5	1,09	0,97	41,87	74,86	79,22	100,00	100,00	56,86
RFX4+FXO6	1,09	0,97	41,87	75,48	79,20	100,00	100,00	56,94
RFX4+FXO7	1,09	0,97	41,87	75,46	78,78	100,00	100,00	56,88
RFX4+FXO8	1,09	0,97	41,87	74,86	99,93	100,00	100,00	59,82
RFX4+FXO9	1,09	0,97	41,87	74,25	78,77	100,00	100,00	56,71
RFX4+FXO10	1,09	0,97	41,87	74,88	78,77	100,00	100,00	56,80
RFX4+FXO11	1,09	0,97	41,87	74,88	78,77	100,00	100,00	56,80
RFX4+FXO12	1,09	0,97	41,87	75,48	80,02	100,00	100,00	57,06

Tabela 64 – Tempos computacionais heurísticas *fix-and-optimize* com solução inicial fornecida por RFX4 - tempo em segundos.

	S1	S2	M1	R1	R2	R3	R4	Média
RFX4+FXO1	901,42	206,45	3.334,89	3.268,00	2.698,30	3.600,00	3.121,09	2.447,17
RFX4+FXO2	715,89	158,30	3.334,69	3.600,00	2.749,64	3.600,00	3.124,15	2.468,95
RFX4+FXO3	732,61	121,36	2.800,95	2.703,66	3.300,78	2.706,22	3.600,00	2.280,80
RFX4+FXO4	572,49	70,02	2.800,92	2.703,10	2.401,92	2.406,16	2.467,78	1.917,48
RFX4+FXO5	665,95	107,56	2.800,70	2.702,98	2.383,34	2.702,22	3.600,00	2.137,54
RFX4+FXO6	543,00	73,98	2.800,91	2.703,16	2.390,36	2.407,70	2.518,45	1.919,65
RFX4+FXO7	760,52	121,64	2.802,24	2.703,42	2.645,20	2.702,25	2.356,64	2.013,13
RFX4+FXO8	493,88	74,95	2.745,72	2.703,74	597,41	2.421,17	2.581,94	1.659,83
RFX4+FXO9	645,20	98,39	2.800,94	2.703,39	2.644,49	2.701,99	3.600,00	2.170,63
RFX4+FXO10	607,60	69,14	2.801,00	2.703,30	2.644,83	2.406,42	2.814,08	2.006,62
RFX4+FXO11	293,63	35,83	1.227,86	3.600,00	3.600,00	3.600,00	3.600,00	2.279,62
RFX4+FXO12	377,05	51,66	1.243,78	3.600,00	3.600,00	3.600,00	3.600,00	2.296,07

Tabela 65 – Soluções heurísticas *fix-and-optimize* com solução inicial fornecida por HE1B-DS - função objetivo (u.m.).

	S1	S2	M1	R1	R2	R3	R4	Média
HE1B-DS+FXO1	14,02	14,01	25,07	40,59	57,01	487.544,77	395.532.542,76	56.574.319,75
HE1B-DS+FXO2	14,02	14,01	25,07	40,60	49,69	1.237.543,69	395.532.542,76	56.681.461,40
HE1B-DS+FXO3	14,02	14,01	25,07	39,61	49,63	1.237.548,89	12.684.500,04	1.988.884,47
HE1B-DS+FXO4	14,02	14,01	25,07	40,56	48,66	1.350.049,83	14.789.321,26	2.305.644,77
HE1B-DS+FXO5	14,02	14,01	25,07	40,59	48,66	612.547,91	10.722.480,39	1.619.310,09
HE1B-DS+FXO6	14,02	14,01	25,07	39,61	48,66	1.178.010,30	19.520.825,14	2.956.996,69
HE1B-DS+FXO7	14,02	14,01	25,07	40,59	195.972,61	1.801.843,67	395.532.542,76	56.790.064,68
HE1B-DS+FXO8	14,02	14,01	25,07	39,65	195.960,26	1.853.895,67	395.532.542,76	56.797.498,78
HE1B-DS+FXO9	14,02	14,01	25,07	40,59	48,66	612.552,01	15.755.625,25	2.338.331,37
HE1B-DS+FXO10	14,02	14,01	25,07	40,59	48,66	1.291.718,61	13.206.526,37	2.071.198,19
HE1B-DS+FXO11	18,08	18,02	25,10	42,72	63,22	487.564,99	12.256.831,19	1.820.651,90
HE1B-DS+FXO12	18,08	18,02	25,10	42,73	63,22	487.557,00	14.176.333,35	2.094.865,36

Tabela 66 – Soluções heurísticas *fix-and-optimize* com solução inicial fornecida por HE1B-DS - *gap* (%).

	S1	S2	M1	R1	R2	R3	R4	Média
HE1B-DS+FXO1	1,09	0,97	41,87	75,46	81,88	100,00	100,00	57,32
HE1B-DS+FXO2	1,09	0,97	41,87	75,46	79,21	100,00	100,00	56,94
HE1B-DS+FXO3	1,09	0,97	41,87	74,86	79,19	100,00	100,00	56,85
HE1B-DS+FXO4	1,09	0,97	41,87	75,44	78,77	100,00	100,00	56,88
HE1B-DS+FXO5	1,09	0,97	41,87	75,46	78,77	100,00	100,00	56,88
HE1B-DS+FXO6	1,09	0,97	41,87	74,86	78,77	100,00	100,00	56,79
HE1B-DS+FXO7	1,09	0,97	41,87	75,46	99,99	100,00	100,00	59,91
HE1B-DS+FXO8	1,09	0,97	41,87	74,88	99,99	100,00	100,00	59,83
HE1B-DS+FXO9	1,09	0,97	41,87	75,46	78,77	100,00	100,00	56,88
HE1B-DS+FXO10	1,09	0,97	41,87	75,46	78,77	100,00	100,00	56,88
HE1B-DS+FXO11	23,26	23,04	41,95	76,69	83,66	100,00	100,00	64,09
HE1B-DS+FXO12	23,26	23,03	41,95	76,69	83,66	100,00	100,00	64,08

Tabela 67 – Tempos computacionais heurísticas *fix-and-optimize* com solução inicial fornecida por HE1B-DS - tempo em segundos.

	S1	S2	M1	R1	R2	R3	R4	Média
HE1B-DS+FXO1	462,89	175,22	2.402,83	2.709,36	2.192,68	3.171,70	2.154,96	1.895,66
HE1B-DS+FXO2	475,13	140,39	2.402,42	2.709,38	2.903,20	3.611,52	2.154,58	2.056,66
HE1B-DS+FXO3	353,23	76,11	2.345,92	2.280,61	3.600,00	2.283,38	1.927,40	1.838,09
HE1B-DS+FXO4	267,91	43,45	2.402,20	2.259,84	2.193,62	1.851,66	2.395,91	1.630,66
HE1B-DS+FXO5	371,91	80,78	2.049,81	2.258,86	3.600,00	2.289,00	3.600,00	2.039,15
HE1B-DS+FXO6	256,70	32,58	2.091,23	2.260,05	2.193,67	1.854,14	2.075,79	1.537,74
HE1B-DS+FXO7	321,48	51,86	2.423,56	2.260,38	2.447,56	2.306,31	3.600,00	1.915,88
HE1B-DS+FXO8	387,94	43,11	2.214,58	2.259,42	386,48	1.851,70	2.156,18	1.328,49
HE1B-DS+FXO9	400,19	71,61	2.366,42	2.258,47	3.600,00	2.286,75	1.896,40	1.839,98
HE1B-DS+FXO10	301,52	45,06	2.402,24	2.258,48	2.543,62	1.847,73	2.484,53	1.697,60
HE1B-DS+FXO11	12,44	3,05	496,52	3.600,00	3.600,00	1.246,86	3.562,96	1.788,83
HE1B-DS+FXO12	12,11	2,91	605,22	3.600,00	3.600,00	1.712,48	3.548,29	1.868,72

Tabela 68 – Detalhes das soluções obtidas pela estratégia CPX12 na resolução do modelo MDSL-2E-LT para todas as instâncias.

Instância	Função objetivo	Função objetivo calculada	Limitante inferior	gap^1 Relaxação linear	gap^2	Atraso	Estoque	Tanq. Prep.	Limp. Temp. Linha	Temp. Trocas	Tempo
	(u.m.)	(u.m.)	(u.m.)	(%)	(%)	(u.m.)	(u.m.)	(umid.)	(umid.)	(u.m.)	(seg.)
S1	14,01	14,00	13,87	0,93	1,40	90,00	0,00	3	1	10,00	3.600,34
S2	14,01	14,00	13,87	0,93	2,79	80,07	0,00	3	1	10,00	531,89
M1	25,02	25,00	14,57	41,72	1,74	93,04	0,00	5	2	18,00	3.600,00
R1	41,18	41,00	9,96	75,71	4,78	88,34	0,00	18	4	19,00	3.600,00
R2	50,18	50,00	10,33	79,34	6,14	87,72	0,00	21	3	26,00	3.600,00
R3	1.237.545,21	1.237.545,00	8,62	100,00	5,25	100,00	1.237.500,00	23	4	18,00	3.600,00
R4	173.182.854,35	173.182.854,00	22,56	100,00	16,92	100,00	171.999.291,67	42	0	62,00	3.600,00
J3M2T2-1	4.570.081,36	4.570.080,40	11,25	100,00	3,26	100,00	4.570.009,60	15	5	50,80	3.600,00
J3M2T2-2	80,80	79,60	17,60	77,89	3,99	94,99	0,00	14	3	62,60	3.600,00
J3M2T2-3	64,52	63,80	34,24	46,33	3,25	94,91	0,00	13	5	45,80	3.600,00
J5M2T4-1	128,73	128,40	82,26	35,93	6,06	95,28	0,00	10	1	117,40	3.600,00
J5M2T4-2	4.668.067,93	4.668.067,29	27,19	100,00	7,18	100,00	2.392.285,46	17	5	56,30	3.600,00
J5M2T4-3	1.660.217,03	1.660.216,14	27,27	100,00	4,88	100,00	1.550.770,73	20	5	65,60	3.600,00
J10M2T4-1	44,26	43,80	22,46	48,72	4,92	88,77	0,00	11	4	28,80	3.600,00
J10M2T4-2	5.384.043,59	5.384.042,38	57,47	100,00	5,00	100,00	4.989.178,45	19	4	202,80	3.600,00
J10M2T4-3	47,45	46,80	16,08	65,64	5,55	88,14	0,00	19	7	20,80	3.600,00
J10M4T4-1	3.491.312,01	1.702.094.355,50	16,54	100,00	9,13	100,00	3.000.706,97	11	0	346,30	3.600,00
J10M4T4-2	1.702.094.355,50	1.331.390.668,00	11,48	100,00	12,05	100,00	1.702.094.355,50	0	0	0,00	3.600,00
J10M4T4-3	1.331.390.668,00	473.827,23	11,67	100,00	11,04	100,00	1.331.390.668,00	0	0	0,00	3.600,00
J15M6T4-1	3.722.444.221,00	3.722.444.221,00	*	*	14,75	100,00	3.625.455.192,60	0	0	0,00	3.600,00
J15M6T4-2	2.780.035.092,00	2.780.035.092,00	*	*	*	*	2.701.479.594,70	0	0	0,00	3.600,00
J15M6T4-3	2.321.584.616,00	2.321.584.616,00	*	*	14,11	100,00	2.280.360.636,00	0	0	0,00	3.600,00
J20M6T6-1	3.227.228.020,18	3.227.228.020,18	*	*	18,64	100,00	3.152.617.149,70	0	0	0,00	3.600,00
J20M6T6-2	4.657.845.415,15	4.657.845.415,15	*	*	*	*	4.548.040.612,70	0	0	0,00	3.600,00
J20M6T6-3	2.512.697.057,76	2.512.697.057,76	14,36	100,00	13,78	100,00	2.455.988.079,30	0	0	0,00	3.600,00

¹ gap obtido com o limitante inferior² gap obtido com a relaxação linear

Tabela 69 – Detalhes das soluções obtidas pela estratégia RFX4 para todas as instâncias.

Instância	Função objetivo (u.m.)	Função objetivo calculada (u.m.)	gap ¹ (%)	gap ² (%)	Atraso (u.m.)	Estoque (u.m.)	Limp. Tanq. Prep. (unid.)	Limp. Temp. Linha (unid.)	Trocas (u.m.)	Tempo (seg.)
S1	14,01	14,00	0,93	90,00	0,00	0,00	3	1	10,00	398,47
S2	14,01	14,00	0,93	80,07	0,00	0,00	3	1	10,00	459,05
M1	25,02	25,00	41,72	93,04	0,00	0,00	5	2	18,00	2.991,70
R1	40,20	40,00	75,10	88,05	0,00	0,00	18	4	18,00	3.600,00
R2	49,18	49,00	78,92	87,47	0,00	0,00	21	3	25,00	3.600,00
R3	1.196.377,70	1.196.377,33	100,00	100,00	1.196.333,33	0,00	21	6	17,00	3.600,00
R4	11.001.671,42	11.001.671,00	100,00	100,00	0,00	11.001.550,00	39	4	78,00	3.600,00
J3M2T2-1	71,68	70,80	84,11	95,40	0,00	0,00	15	5	50,80	3.600,00
J3M2T2-2	65,25	63,70	72,37	93,74	0,00	0,00	18	7	38,70	3.600,00
J3M2T2-3	63,51	62,80	45,48	94,82	0,00	0,00	13	4	45,80	3.600,00
J5M2T4-1	114,24	113,90	27,78	94,68	0,00	0,00	13	3	97,90	3.600,00
J5M2T4-2	1.027.427,42	1.027.426,64	100,00	100,00	0,00	1.027.349,94	21	6	49,70	3.600,00
J5M2T4-3	109.449,05	109.447,11	99,98	100,00	0,00	109.354,81	28	5	60,30	3.600,00
J10M2T4-1	37,92	37,40	39,95	86,84	0,00	0,00	16	6	15,40	3.600,00
J10M2T4-2	413.462,25	413.461,16	99,99	100,00	0,00	413.184,06	18	4	255,10	3.600,00
J10M2T4-3	49,47	48,80	67,05	88,63	0,00	0,00	21	7	20,80	3.600,00
J10M4T4-1	36.811.926,49	36.811.924,61	100,00	100,00	28.915.214,31	7.896.496,80	17	4	193,50	3.600,00
J10M4T4-2	1.693.417.888,18	1.693.417.888,18	100,00	100,00	1.693.417.862,38	0,00	0	0	25,80	3.600,00
J10M4T4-3	1.265.837.225,77	1.265.837.225,77	100,00	100,00	1.265.837.187,97	0,00	0	0	37,80	3.600,00
J15M6T4-1	2.587.771.219,62	2.587.771.272,00	*	100,00	2.587.742.403,80	28.763,02	0	0	52,80	3.600,00
J15M6T4-2	1.835.181.425,91	1.835.181.425,91	*	*	1.835.181.200,70	164,81	0	0	60,40	3.600,00
J15M6T4-3	1.204.192.427,82	1.204.192.427,82	*	100,00	1.202.113.406,41	2.078.964,52	0	0	56,90	3.600,00
J20M6T6-1	**	**	**	**	**	**	**	**	**	**
J20M6T6-2	**	**	**	**	**	**	**	**	**	**
J20M6T6-3	1.026.276.931,76	1.026.276.931,76	100,00	100,00	1.018.708.180,26	7.568.585,10	0	0	166,40	3.600,00

¹ gap obtido com o limitante inferior² gap obtido com a relaxação linear

** não foi encontrada solução factível no tempo

Tabela 70 – Detalhes das soluções obtidas pela estratégia HE1B-DS para todas as instâncias.

Instância	Função objetivo (u.m.)	gap^1 (%)	gap^2 (%)	Atraso (u.m.)	Estoque (u.m.)	Limp. Temp. Tanq. Prep. (unid.)	Limp. Temp. Linha (unid.)	Trocas (u.m.)	Tempo (seg.)
S1	24,00	42,21	94,17	0,00	0,00	11	7	6,00	1,80
S2	24,00	42,21	88,38	0,00	0,00	11	7	6,00	1,80
M1	31,00	53,00	94,39	0,00	0,00	16	5	10,00	5,16
R1	47,00	78,81	89,83	0,00	0,00	25	6	16,00	17,19
R2	195.962,00	99,99	100,00	0,00	19.590,00	33	11	18,00	67,95
R3	5.094.801,00	100,00	100,00	50.947,50	0,00	26	6	19,00	85,72
R4	4.068.931,00	100,00	100,00	0,00	406.877,50	83	25	48,00	1.216,05
J3M2T2-1	58,90	80,90	94,47	0,00	0,00	22	8	28,90	2,23
J3M2T2-2	60,60	70,96	93,42	0,00	0,00	22	9	29,60	4,97
J3M2T2-3	4.470.994,22	100,00	100,00	29.190,46	1.448,18	16	5	45,80	21,33
J5M2T4-1	118,10	30,35	94,87	0,00	0,00	17	5	96,10	18,41
J5M2T4-2	47.420.730,87	100,00	100,00	309.979,42	81.500,87	16	4	72,10	1.236,39
J5M2T4-3	3.264.063,37	100,00	100,00	20.251,42	13.933,91	25	9	40,20	772,13
J10M2T4-1	32,00	29,81	84,63	0,00	0,00	18	6	8,00	1,31
J10M2T4-2	351.485,31	99,98	100,00	0,00	19.289,51	21	8	171,70	115,48
J10M2T4-3	45,70	64,81	87,86	0,00	0,00	23	8	14,70	15,70
J10M4T4-1	192,70	91,42	95,26	0,00	0,00	40	12	140,70	457,33
J10M4T4-2	291.490,11	100,00	100,00	0,00	16.204,97	32	4	502,90	1.406,50
J10M4T4-3	212,70	94,51	94,81	0,00	0,00	70	22	120,70	866,05
J15M6T4-1	460.742.743,66	*	100,00	3.212.346,00	30.360,24	49	10	536,00	1.704,27
J15M6T4-2	217.601.920,71	*	*	1.598.696,38	162.927,83	20	2	993,00	1.584,64
J15M6T4-3	232.987.274,03	*	100,00	1.491.951,81	54.184,39	53	10	399,20	1.877,61
J20M6T6-1	8.401.247,60	*	100,00	47.442,36	83.094,82	75	11	692,90	2.604,83
J20M6T6-2	1.774.291.352,15	*	*	12.288.975,75	0,00	43	5	586,00	2.081,95
J20M6T6-3	16.958.575,48	100,00	100,00	100.649,54	171.629,56	49	14	558,20	1.763,42

¹ gap obtido com o limitante inferior² gap obtido com a relaxação linear

Tabela 71 – Detalhes das soluções obtidas pela estratégia H-SUP para todas as instâncias.

Instância	Função objetivo (u.m.)	gap^1 (%)	gap^2 (%)	Atraso (u.m.)	Estoque (u.m.)	Limp. Temp. Tanq. Prep. (unid.)	Limp. Temp. Linha (unid.)	Trocas (u.m.)	Tempo (seg.)
S1	15,00	7,53	90,67	0,00	0,00	3	0	12,00	61,11
S2	15,00	7,53	81,40	0,00	0,00	3	0	12,00	22,06
M1	29,00	49,76	94,00	0,00	0,00	9	2	18,00	600,17
R1	47,00	78,81	89,83	0,00	0,00	24	4	19,00	600,42
R2	20.680,00	99,95	99,97	0,00	20.625,00	27	3	25,00	600,50
R3	2.154.214,67	100,00	100,00	2.154.166,67	0,00	26	5	17,00	600,50
R4	2.979.833,67	100,00	100,00	0,00	2.979.666,67	88	27	52,00	601,17
J3M2T2-1	57,90	80,57	94,37	0,00	0,00	21	8	28,90	600,27
J3M2T2-2	58,60	69,97	93,19	0,00	0,00	22	7	29,60	600,34
J3M2T2-3	56,60	39,51	94,26	0,00	0,00	15	5	36,60	601,50
J5M2T4-1	117,50	29,99	94,84	0,00	0,00	17	3	97,50	600,34
J5M2T4-2	1.805.573,62	100,00	100,00	0,00	1.805.496,92	20	7	49,70	601,08
J5M2T4-3	99.245,69	99,97	100,00	0,00	99.173,89	23	6	42,80	1.201,55
J10M2T4-1	39,40	42,99	87,51	0,00	0,00	18	6	15,40	2.533,22
J10M2T4-2	207.981,58	99,97	100,00	0,00	207.779,08	20	7	175,50	602,81
J10M2T4-3	44,70	64,03	87,58	0,00	0,00	22	8	14,70	603,39
J10M4T4-1	238,20	93,06	96,17	0,00	0,00	23	5	210,20	601,16
J10M4T4-2	574.870.779,97	100,00	100,00	569.379.753,40	5.490.869,47	41	14	102,10	607,52
J10M4T4-3	61.594.158,20	100,00	100,00	60.032.546,19	1.561.085,31	33	3	490,70	1.818,77
J15M6T4-1	2.655.564.912,90	*	100,00	2.655.564.912,90	0,00	0	0	0,00	3.600,00
J15M6T4-2	1.818.783.551,55	*	*	1.818.783.532,55	0,00	0	0	19,00	638,62
J15M6T4-3	906.508.710,24	*	100,00	905.779.271,96	729.321,39	14	1	101,90	1.208,08
J20M6T6-1	2.406.508.444,90	*	100,00	2.406.508.444,90	0,00	0	0	0,00	608,88
J20M6T6-2	3.449.992.588,20	*	*	3.449.992.588,20	0,00	0	0	0,00	649,53
J20M6T6-3	1.888.898.294,70	100,00	100,00	1.888.898.294,70	0,00	0	0	0,00	658,70

¹ gap obtido com o limitante inferior² gap obtido com a relaxação linear

Tabela 72 – Detalhes das soluções obtidas pela estratégia CPX12+FXO12 para todas as instâncias.

Instância	Função objetivo (u.m.)	Função objetivo calculada (u.m.)	gap^1 (%)	gap^2 (%)	Atraso (u.m.)	Estoque (u.m.)	Tanq. Prep. (umid.)	Limp. Temp. Linha (umid.)	Trocas (u.m.)	Tempo (seg.)
S1	14,02	14,00	0,93	90,00	0,00	0,00	3	1	10,00	1.203,31
S2	14,01	14,00	0,93	80,07	0,00	0,00	3	1	10,00	2.404,09
M1	25,07	25,00	41,72	93,04	0,00	0,00	6	2	17,00	1.230,94
R1	44,68	44,00	77,36	89,14	0,00	0,00	18	6	20,00	3.600,00
R2	57,86	57,00	81,88	89,23	0,00	0,00	27	4	26,00	3.600,00
R3	1.237.545,72	1.237.545,00	100,00	100,00	1.237.500,00	0,00	23	4	18,00	3.600,00
R4	19.474.140,83	19.474.138,17	100,00	100,00	0,00	19.473.979,17	70	10	79,00	3.456,97
J3M2T2-1	88,14	87,20	87,10	96,26	0,00	0,00	15	5	67,20	3.600,00
J3M2T2-2	100,16	98,00	82,04	95,93	0,00	0,00	23	5	70,00	3.600,00
J3M2T2-3	63,51	62,80	45,48	94,82	0,00	0,00	13	4	45,80	3.600,00
J5M2T4-1	128,73	128,40	35,93	95,28	0,00	0,00	10	1	117,40	1.228,33
J5M2T4-2	4.868.703,94	4.868.703,16	100,00	100,00	2.590.592,33	2.278.012,02	19	5	74,80	3.600,00
J5M2T4-3	104.407,86	104.406,50	99,97	100,00	0,00	104.316,20	26	5	59,30	2.028,89
J10M2T4-1	36,90	36,40	38,30	86,48	0,00	0,00	11	4	21,40	3.184,17
J10M2T4-2	5.668.630,28	5.668.628,59	100,00	100,00	2.095.058,61	3.573.332,28	25	4	208,70	3.600,00
J10M2T4-3	54,68	54,00	70,22	89,72	0,00	0,00	19	6	29,00	3.066,58
J10M4T4-1	6.350.285,47	6.350.284,54	100,00	100,00	0,00	6.349.797,14	32	4	451,40	3.600,00
J10M4T4-2	1.632.078.219,46	1.632.078.219,46	100,00	100,00	1.632.078.218,46	0,00	0	0	1,00	3.600,00
J10M4T4-3	959.318.014,00	959.318.013,61	100,00	100,00	959.317.842,61	0,00	10	9	152,00	3.600,00
J15M6T4-1	3.722.444.221,00	3.722.444.221,00	*	100,00	3.625.455.192,60	96.989.027,97	0	0	0,00	3.600,00
J15M6T4-2	2.780.035.092,00	2.780.035.092,00	*	*	2.701.479.594,70	78.555.496,90	0	0	0,00	3.600,00
J15M6T4-3	2.321.584.616,00	2.321.584.616,00	*	100,00	2.280.360.636,00	41.223.980,00	0	0	0,00	3.600,00
J20M6T6-1	3.227.228.020,18	3.227.228.020,18	*	100,00	3.152.617.149,70	74.610.870,48	0	0	0,00	3.600,00
J20M6T6-2	4.657.845.415,15	4.657.845.415,15	*	*	4.548.040.612,70	109.804.802,45	0	0	0,00	3.600,00
J20M6T6-3	2.512.697.057,76	2.512.697.057,76	100,00	100,00	2.455.988.079,30	56.708.978,46	0	0	0,00	3.600,00

¹ gap obtido com o limitante inferior² gap obtido com a relaxação linear

Tabela 73 – Detalhes das soluções obtidas pela estratégia RFX4+FXO3 para todas as instâncias.

Instância	Função objetivo (u.m.)	Função objetivo calculada (u.m.)	gap ¹ (%)	gap ² (%)	Atraso (u.m.)	Estoque (u.m.)	Limp. Tanq. Prep. (unid.)	Limp. Temp. Linha (unid.)	Trocas (u.m.)	Tempo (seg.)
S1	14,02	14,00	0,93	90,00	0,00	0,00	3	1	10,00	732,61
S2	14,01	14,00	0,93	80,07	0,00	0,00	3	1	10,00	853,97
M1	25,07	25,00	41,72	93,04	0,00	0,00	6	2	17,00	2.800,95
R1	40,61	40,00	75,10	88,05	0,00	0,00	18	4	18,00	2.703,66
R2	48,66	48,00	78,48	87,21	0,00	0,00	20	3	25,00	3.300,78
R3	487.546,76	487.546,00	100,00	100,00	487.500,00	0,00	23	6	17,00	2.706,22
R4	13.737.477,84	13.737.475,83	100,00	100,00	203.041,67	13.534.279,17	55	10	90,00	3.600,00
J3M2T2-1	65,70	64,80	82,64	94,97	0,00	0,00	16	6	42,80	3.600,00
J3M2T2-2	33.009.833,18	33.009.832,43	100,00	100,00	33.009.751,43	0,00	10	1	70,00	3.600,00
J3M2T2-3	63,51	62,80	45,48	94,82	0,00	0,00	13	4	45,80	3.600,00
J5M2T4-1	116,76	116,40	29,33	94,79	0,00	0,00	13	4	99,40	2.703,47
J5M2T4-2	1.169.295,63	1.169.294,83	100,00	100,00	0,00	1.169.217,23	21	7	49,60	2.417,34
J5M2T4-3	761.949,52	761.948,24	100,00	100,00	0,00	761.839,14	25	6	78,10	2.418,98
J10M2T4-1	5.840,90	5.840,39	99,62	99,92	0,00	5.774,39	12	6	48,00	853,50
J10M2T4-2	7.444.647,81	7.444.647,16	100,00	100,00	3.898.115,88	3.546.318,98	12	0	200,30	3.600,00
J10M2T4-3	168.581,24	168.579,60	99,99	100,00	0,00	168.490,20	28	7	54,40	936,00
J10M4T4-1	59.900.742,74	59.900.742,58	100,00	100,00	55.676.046,93	4.224.425,15	3	0	267,50	3.014,59
J10M4T4-2	1.693.417.888,18	1.693.417.888,18	100,00	100,00	1.693.417.862,38	0,00	0	0	25,80	2.766,14
J10M4T4-3	1.331.390.668,00	1.331.390.668,00	100,00	100,00	1.331.390.668,00	0,00	0	0	0,00	2.734,48
J15M6T4-1	2.587.771.219,62	2.587.771.219,62	*	100	2.587.742.403,80	28.763,02	0	0	52,80	3.600,00
J15M6T4-2	1.835.181.425,91	1.835.181.425,91	*	*	1.835.181.200,70	164,81	0	0	60,40	3.600,00
J15M6T4-3	1.204.192.427,82	1.204.192.427,82	*	100	1.202.113.406,41	2.078.964,52	0	0	56,90	3.600,00
J20M6T6-1	**	**	**	**	**	**	**	**	**	**
J20M6T6-2	**	**	**	**	**	**	**	**	**	**
J20M6T6-3	1.026.276.931,76	1.026.276.931,76	100	100	1.018.708.180,26	7.568.585,10	0	0	166,40	3.600,00

¹ gap obtido com o limitante inferior² gap obtido com a relaxação linear

* não foi possível encontrar um limitante inferior

** não foi encontrada solução factível no tempo

Tabela 74 – Detalhes das soluções obtidas pela estratégia HE1B-DS+FXO3 para todas as instâncias.

Instância	Função objetivo (u.m.)	Função objetivo calculada (u.m.)	gap ¹ (%)	gap ² (%)	Atraso (u.m.)	Estoque (u.m.)	Limp. Tanq, Prep. (unid.)	Limp. Temp. Linha (unid.)	Trocas (u.m.)	Tempo (seg.)
S1	14,02	14,00	0,93	90,00	0,00	0,00	3	1	10,00	353,08
S2	14,01	14,00	0,93	80,07	0,00	0,00	3	1	10,00	75,95
M1	25,07	25,00	41,72	93,04	0,00	0,00	6	2	17,00	2.345,74
R1	39,61	39,00	74,46	87,74	0,00	0,00	17	4	18,00	2.280,97
R2	49,63	49,00	78,92	87,47	0,00	0,00	21	3	25,00	3.584,91
R3	1.237.548,89	1.237.548,00	100,00	100,00	1.237.500,00	0,00	24	4	20,00	2.212,96
R4	2.227.575,79	2.227.572,67	100,00	100,00	0,00	2.227.416,67	79	24	53,00	1.641,05
J3M2T2-1	57,04	55,90	79,87	94,17	0,00	0,00	19	8	28,90	3.600,00
J3M2T2-2	63,28	60,60	70,96	93,42	0,00	0,00	22	9	29,60	2.704,56
J3M2T2-3	63,51	62,80	45,48	94,82	0,00	0,00	13	4	45,80	3.600,00
J5M2T4-1	114,46	114,10	27,91	94,69	0,00	0,00	13	4	97,10	3.600,00
J5M2T4-2	4.176.009,53	4.176.008,83	100,00	100,00	2.511.787,98	1.664.139,64	20	5	56,20	3.600,00
J5M2T4-3	270.542,79	270.541,28	99,99	100,00	0,00	270.439,68	24	8	69,60	3.600,00
J10M2T4-1	30,55	30,00	25,13	83,60	0,00	0,00	16	6	8,00	1.875,48
J10M2T4-2	138.821,84	138.819,90	99,96	100,00	0,00	138.608,60	25	11	175,30	1.166,49
J10M2T4-3	35,71	35,00	54,06	84,14	0,00	0,00	20	7	8,00	2.259,55
J10M4T4-1	197,55	196,20	91,57	95,35	0,00	0,00	29	10	157,20	3.600,00
J10M4T4-2	428.700,26	428.696,09	100,00	100,00	0,00	428.203,99	43	6	443,10	2.447,64
J10M4T4-3	1.767.056,09	1.767.050,07	100,00	100,00	1.332.555,01	434.165,36	56	13	260,70	2.499,09
J15M6T4-1	485.843.568,94	485.843.568,94	*	100,00	483.708.290,62	2.134.668,42	50	13	546,90	3.600,00
J15M6T4-2	303.871.618,31	303.871.618,31	*	*	301.328.460,27	2.541.715,84	25	1	1.416,20	3.600,00
J15M6T4-3	37.204.906,85	37.204.906,85	*	100,00	36.693.365,45	510.933,70	74	20	513,70	3.600,00
J20M6T6-1	30.504.781,92	30.504.781,92	*	100,00	28.531.604,18	1.972.419,34	85	16	657,40	3.600,00
J20M6T6-2	332.387.316,23	332.387.316,23	*	*	331.725.622,48	663.420,58	96	16	828,50	3.600,00
J20M6T6-3	6.237.371,10	6.237.371,10	100,00	100,00	4.066.798,55	2.169.890,05	48	16	618,50	3.600,00

¹ gap obtido com o limitante inferior² gap obtido com a relaxação linear

* não foi possível encontrar um limitante inferior

Tabela 75 – Detalhes das soluções obtidas pela estratégia FXOV1_T-M-J para todas as instâncias.

Instância	Função objetivo (u.m.)	Função objetivo calculada (u.m.)	gap ¹ (%)	gap ² (%)	Atraso (u.m.)	Estoque (u.m.)	Limp. Tanq. Prep. (unid.)	Limp. Temp. Linha (unid.)	Trocas (u.m.)	Tempo (seg.)
S1	14,02	14,00	0,93	90,00	0,00	0,00	3	1	10,00	2.907,13
S2	14,01	14,00	0,93	80,07	0,00	0,00	3	1	10,00	851,61
M1	25,07	25,00	41,72	93,04	0,00	0,00	6	2	17,00	3.600,00
R1	39,65	39,00	74,46	87,74	0,00	0,00	19	4	16,00	3.600,00
R2	48,66	48,00	78,48	87,21	0,00	0,00	20	3	25,00	3.600,00
R3	487.547,78	487.547,00	100,00	100,00	487.500,00	0,00	24	6	17,00	3.600,00
R4	2.576.476,70	2.576.474,33	100,00	100,00	0,00	2.576.333,33	63	15	63,00	3.600,00
J3M2T2-1	56,02	54,90	79,51	94,06	0,00	0,00	18	8	28,90	3.600,00
J3M2T2-2	63,28	60,60	70,96	93,42	0,00	0,00	22	9	29,60	3.479,99
J3M2T2-3	64,52	63,80	46,33	94,91	0,00	0,00	13	5	45,80	3.600,00
J5M2T4-1	114,46	114,10	27,91	94,69	0,00	0,00	13	4	97,10	3.600,00
J5M2T4-2	3.782.029,45	3.782.028,46	100,00	100,00	1.880.696,12	1.901.246,83	23	7	55,50	3.600,00
J5M2T4-3	132.738,70	132.737,84	99,98	100,00	0,00	132.658,34	19	4	56,50	3.600,00
J10M2T4-1	30,55	30,00	25,13	83,60	0,00	0,00	16	6	8,00	3.219,85
J10M2T4-2	204.232,72	204.231,49	99,97	100,00	0,00	204.028,89	19	8	175,60	3.600,00
J10M2T4-3	42,38	41,70	61,44	86,69	0,00	0,00	20	7	14,70	3.600,00
J10M4T4-1	203,69	202,60	91,84	95,49	0,00	0,00	31	10	161,60	3.600,00
J10M4T4-2	407.173,02	407.168,97	100,00	100,00	0,00	406.677,87	42	6	443,10	3.600,00
J10M4T4-3	64.927,25	64.923,96	99,98	99,98	0,00	64.593,26	54	11	265,7	3.510,69
J15M6T4-1	485.843.568,94	485.843.568,94	*	100,00	483.708.290,62	2.134.668,42	50	13	546,90	3.600,00
J15M6T4-2	303.871.618,31	303.871.618,31	*	*	301.328.460,27	2.541.715,84	25	1	1.416,20	3.600,00
J15M6T4-3	37.204.906,85	37.204.906,85	*	100,00	36.693.365,45	510.933,70	74	20	513,70	3.600,00
J20M6T6-1	30.504.781,92	30.504.781,92	*	100,00	28.531.604,18	1.972.419,34	85	16	657,40	3.600,00
J20M6T6-2	332.387.316,23	332.387.316,23	*	*	331.725.622,48	663.420,58	96	16	828,50	3.600,00
J20M6T6-3	6.237.371,10	6.237.371,10	100,00	100,00	4.066.798,55	2.169.890,05	48	16	618,50	3.600,00

¹ gap obtido com o limitante inferior² gap obtido com a relaxação linear

* não foi possível encontrar um limitante inferior

Tabela 76 – Detalhes das soluções obtidas pela estratégia FXOV2 para todas as instâncias.

Instância	Função objetivo (u.m.)	Função objetivo calculada (u.m.)	gap ¹ (%)	gap ² (%)	Atraso (u.m.)	Estoque (u.m.)	Limp. Tanq, Prep. (unid.)	Limp. Temp. Linha (unid.)	Trocas (u.m.)	Tempo (seg.)
S1	14,02	14,00	0,93	90,00	0,00	0,00	3	1	10,00	3.600,00
S2	14,01	14,00	0,93	80,07	0,00	0,00	3	1	10,00	412,28
M1	25,07	25,00	41,72	93,04	0,00	0,00	6	2	17,00	3.600,00
R1	40,59	40,00	75,10	88,05	0,00	0,00	18	4	18,00	3.600,00
R2	53,86	53,00	80,51	88,42	0,00	0,00	26	7	20,00	3.600,00
R3	487.544,75	487.544,00	100,00	100,00	487.500,00	0,00	21	6	17,00	3.600,00
R4	17.734.013,20	17.734.010,00	100,00	100,00	8.929.666,67	8.804.183,33	84	21	55,00	3.600,00
J3M2T2-1	56,98	55,90	79,87	94,17	0,00	0,00	19	8	28,90	3.600,00
J3M2T2-2	63,28	60,60	70,96	93,42	0,00	0,00	22	9	29,60	2.664,85
J3M2T2-3	63,51	62,80	45,48	94,82	0,00	0,00	13	4	45,80	3.600,00
J5M2T4-1	114,46	114,10	27,91	94,69	0,00	0,00	13	4	97,10	3.600,00
J5M2T4-2	7.835.851,69	7.835.850,49	100,00	100,00	6.126.123,64	1.709.647,85	27	6	46,00	3.600,00
J5M2T4-3	248.708,75	248.707,63	99,99	100,00	0,00	248.615,03	23	7	62,60	3.600,00
J10M2T4-1	30,55	30,00	25,13	83,60	0,00	0,00	16	6	8,00	3.529,65
J10M2T4-2	142.607,24	142.605,78	99,96	100,00	0,00	142.400,18	21	9	175,60	3.600,00
J10M2T4-3	42,38	41,70	61,44	86,69	0,00	0,00	20	7	14,70	3.600,00
J10M4T4-1	204,39	203,30	91,86	95,51	0,00	0,00	31	10	162,30	3.600,00
J10M4T4-2	485,42	482,10	97,62	97,50	0,00	0,00	33	6	443,10	3.600,00
J10M4T4-3	331,31	328,00	96,44	96,63	0,00	0,00	54	12	262,00	3.600,00
J15M6T4-1	485.843.568,94	485.843.568,94	*	100,00	483.708.290,62	2.134.668,42	50	13	546,90	3.600,00
J15M6T4-2	303.871.618,31	303.871.618,31	*	*	301.328.460,27	2.541.715,84	25	1	1.416,20	3.600,00
J15M6T4-3	37.204.906,85	37.204.906,85	*	100,00	36.693.365,45	510.933,70	74	20	513,70	3.600,00
J20M6T6-1	30.504.781,92	30.504.781,92	*	100,00	28.531.604,18	1.972.419,34	85	16	657,40	3.600,00
J20M6T6-2	332.387.316,23	332.387.316,23	*	*	331.725.622,48	663.420,58	96	16	828,50	3.600,00
J20M6T6-3	6.237.371,10	6.237.371,10	100,00	100,00	4.066.798,55	2.169.890,05	48	16	618,50	3.600,00

¹ gap obtido com o limitante inferior² gap obtido com a relaxação linear

* não foi possível encontrar um limitante inferior

Tabela 77 – Detalhes das soluções obtidas pela estratégia FXO+VNDS para todas as instâncias.

Instância	Função objetivo (u.m.)	Função objetivo calculada (u.m.)	gap^1 (%)	gap^2 (%)	Atraso (u.m.)	Estoque (u.m.)	Limp. Tanq. Prep. (umid.)	Limp. Temp. Linha (umid.)	Trocas (u.m.)	Tempo (seg.)
S1	14,02	14,00	0,93	90,00	0,00	0,00	3	1	10,00	2.735,14
S2	14,01	14,00	0,93	80,07	0,00	0,00	3	1	10,00	785,06
M1	25,07	25,00	41,72	93,04	0,00	0,00	7	3	15,00	3.600,00
R1	42,62	42,00	76,29	88,62	0,00	0,00	19	4	19,00	3.600,00
R2	56,91	56,00	81,55	89,04	0,00	0,00	27	9	20,00	3.600,00
R3	487.547,74	487.547,00	100,00	100,00	487.500,00	0,00	22	6	19,00	3.600,00
R4	14.690.612,56	14.690.610,50	100,00	100,00	0,00	14.690.462,50	55	7	86,00	3.600,00
J3M2T2-1	56,03	54,90	79,51	94,06	0,00	0,00	18	8	28,90	3.600,00
J3M2T2-2	63,28	60,60	70,96	93,42	0,00	0,00	22	9	29,60	2.756,78
J3M2T2-3	64,52	63,80	46,33	94,91	0,00	0,00	13	5	45,80	3.600,00
J5M2T4-1	118,98	118,10	30,35	94,87	0,00	0,00	17	5	96,10	3.600,00
J5M2T4-2	6.748.602,30	6.748.601,21	100,00	100,00	5.168.370,64	1.580.153,57	24	7	46,00	3.600,00
J5M2T4-3	358.837,85	358.836,45	99,99	100,00	0,00	358.753,55	25	9	48,90	3.600,00
J10M2T4-1	30,55	30,00	25,13	83,60	0,00	0,00	16	6	8,00	3.600,00
J10M2T4-2	223.822,75	223.821,48	99,97	100,00	0,00	223.621,78	20	8	171,70	3.600,00
J10M2T4-3	47,00	45,70	64,81	87,86	0,00	0,00	23	8	14,70	3.317,49
J10M4T4-1	204,41	203,30	91,86	95,51	0,00	0,00	31	10	162,30	3.600,00
J10M4T4-2	428.701,86	428.694,41	100,00	100,00	0,00	428.202,00	43	6	443,41	3.600,00
J10M4T4-3	1.767.056,09	1.767.050,07	100,00	100,00	1.332.575,00	434.215,00	56	13	260,07	3.600,00
J15M6T4-1	485.843.568,94	485.843.568,94	*	100,00	483.708.290,62	2.134.668,42	50	13	546,90	3.600,00
J15M6T4-2	303.871.618,31	303.871.618,31	*	*	301.328.460,27	2.541.715,84	25	1	1.416,20	3.600,00
J15M6T4-3	37.204.906,85	37.204.906,85	*	100,00	36.693.365,45	510.933,70	74	20	513,70	3.600,00
J20M6T6-1	30.504.781,92	30.504.781,92	*	100,00	28.531.604,18	1.972.419,34	85	16	657,40	3.600,00
J20M6T6-2	332.387.316,23	332.387.316,23	*	*	331.725.622,48	663.420,58	96	16	828,50	3.600,00
J20M6T6-3	6.237.371,10	6.237.371,10	100,00	100,00	4.066.798,55	2.169.890,05	48	16	618,50	3.600,00

¹ gap obtido com o limitante inferior² gap obtido com a relaxação linear

* não foi possível encontrar um limitante inferior