

ANDRÉ BINA POSSATTO

**DIMENSIONAMENTO E SEQUENCIAMENTO DE LOTES COM CÉLULAS
PRODUTIVAS EM PARALELO**

São Paulo

2015

ANDRÉ BINA POSSATTO

**DIMENSIONAMENTO E SEQUENCIAMENTO DE LOTES COM CÉLULAS
PRODUTIVAS EM PARALELO**

Trabalho de Formatura apresentado à Escola
Politécnica da Universidade de São Paulo para
obtenção do diploma de Engenheiro de
Produção.

**São Paulo
2015**

ANDRÉ BINA POSSATTO

**DIMENSIONAMENTO E SEQUENCIAMENTO DE LOTES COM CÉLULAS
PRODUTIVAS EM PARALELO**

Trabalho de Formatura apresentado à Escola
Politécnica da Universidade de São Paulo para
obtenção do diploma de Engenheiro de
Produção.

Orientadora: Profa. Débora Pretti Ronconi

São Paulo
2015

Catálogo-na-publicação

Possatto, André Bina

DIMENSIONAMENTO E SEQUENCIAMENTO DE LOTES COM
CÉLULAS PRODUTIVAS EM PARALELO / A. B. Possatto -- São Paulo, 2015.
112 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São
Paulo. Departamento de Engenharia de Produção.

1.Dimensionamento de lotes 2.Modelos multiestágio 3.Pesquisa
operacional 4.Programação linear inteira mista 5.Sequenciamento de lotes
I.Universidade de São Paulo. Escola Politécnica. Departamento de
Engenharia de Produção II.t.

A todos aqueles que tornaram isso possível.

AGRADECIMENTOS

Agradeço à professora Débora Pretti Ronconi, por ter me orientado durante esse projeto, função que executou com maestria. Todo seu apoio e dedicação foram fundamentais para minha formação e para a execução desse trabalho, e por eles ofereço meus sinceros agradecimentos. Estendo esse agradecimento a todos os professores que tive até hoje, sem os quais não teria alcançado meus sonhos.

Agradeço também ao Márcio Kawamura pelo apoio em encontrar um problema relevante para a empresa e no fornecimento dos dados para o projeto.

Agradeço à toda minha família, tias, tios, primos, avós e outros, que mesmo sem serem família realmente, hoje considero como tal. Seu apoio incondicional permitiu que eu chegasse aqui hoje e por isso sou grato.

Agradeço aos meus amigos e parceiros da equipe Pingus, Alexys Pereira, Luiz Durão e Luiz Laurindo. Vocês me ensinaram o valor e o que é realmente trabalhar em equipe e ter confiança que, enquanto estivermos juntos, não faltarão esforços para alcançar o resultado através de apoio mútuo e ensinamentos contínuos. Obrigado por serem meus amigos.

Agradeço também à minha namorada, Heloisa, que me apoiou sempre durante esse projeto, tendo compreendido todas as vezes que abri mão de fazer algo com ela para redigir esse trabalho. Mais do que isso, agradeço por você ser você mesma, deixar eu ser eu mesmo e por estar comigo enquanto nos tornamos algo mais juntos.

Agradeço ainda aos meus amigos de diversas etapas da vida, sejam eles de São Caetano, do Colégio Etapa ou do Barquinho da Saudade. Vocês estiveram lá em todos os momentos e sem vocês a vida não seria tão interessante.

Por último agradeço aos meus pais Renata e Marcos, por se sacrificarem para prover-me a educação e criação necessária para que pudesse ter condições de lutar e conquistar meus sonhos. Junto com minha irmã Mariana vocês foram a base de tudo que sou hoje, e de tudo que conquistei. Obrigado por tudo.

*“I am older than I once was
And younger than I'll be; that's not unusual.*

Nor is it strange

After changes upon changes

We are more or less the same”

Simon & Garfunkel

RESUMO

O objetivo do presente trabalho é melhorar a eficácia da programação da produção de fraldas de uma multinacional de produtos de higiene, diminuindo para isso o tempo gasto com *setups* das máquinas. Esse processo produtivo caracteriza-se por apresentar duas etapas: produção das fraldas e empacotamento de certas quantidades dessas. Nessa empresa, isso é feito por células em paralelo em que cada célula tem uma máquina produtora e duas embaladoras, caracterizando um problema de máquinas multiestágio em paralelo. Através da revisão bibliográfica verificou-se que esse problema em específico é, no melhor conhecimento do autor, inédito na literatura, apesar de existirem alguns problemas semelhantes com dois estágios e uma máquina por estágio. Nesse trabalho, foi desenvolvido um modelo capaz de representar esse problema, sendo inclusive adequado para lidar com casos onde a taxa de produção das embaladoras é dependente da combinação de pacotes sendo empacotados no momento. Através da resolução do modelo de programação linear inteira mista no *software* CPLEX, foi possível obter em apenas dez minutos uma solução superior à da empresa em termos de tempo total de produção, que inclui o tempo efetivamente produzindo e tempo de *setup*. Foi verificado ainda um *gap* de otimalidade de 3,9%, considerado baixo, indicando que esse método atinge uma solução de boa qualidade de forma rápida. Foi feito ainda um estudo do impacto potencial causado pela redução de cada um dos três tipos de *setup* no tempo total de produção. Este estudo fornece, portanto, insumos para uma decisão sobre qual deles deve receber maiores esforços de melhoria.

Palavras-Chave: Dimensionamento de lotes, Modelos multiestágio, Pesquisa Operacional, Programação linear inteira mista, Sequenciamento de lotes.

ABSTRACT

The main goal of this work is to improve the efficacy of the production programming of diapers production batches for an international hygiene company, which is achieved by lowering the time spent on setups. This production process has two stages: the production of the diapers themselves and the packaging of different quantities of them for selling. In this company, this is accomplished by multiple cells in parallel, in which each cell has one machine to produce and two machines to pack, characterizing a multistage parallel machine problem. The bibliographic revision has shown that this specific problem was, to the best knowledge of the author, new to the literature, even though there are references to problems with two stages and one machine in each stage. A model to represent this situation was developed by the author, capable even of coping with production rates that depend on the combination of packages being packed at each cell. Solving the mixed linear integer programming model in the software CPLEX, a solution with a total time, including setup and production, lower than the company's current one was found in only ten minutes. It was verified that the optimality gap of this solution is 3,9%, considered very low, indicating that the solution obtained is a high-quality one. The model was further utilized to study the impact of reducing each of the three different setups in the total time of production, thus providing input to a better decision concerning where a process improvement initiative should be focused on.

Keywords: Lot-sizing. Multistage models. Operations research. Mixed integer linear programming. Scheduling.

ÍNDICE DE FIGURAS

Figura 1: Evolução do setor (ABIHPEC 2015)	25
Figura 2: Esquema de produção do setor estudado	27
Figura 3: Programação de A e depois de B	31
Figura 4: Programação de A e B	32
Figura 5: Representação esquemática do modelo de Toledo <i>et al.</i> (2007).....	44
Figura 6: Representação esquemática do modelo de Ferreira, Morabito e Rangel (2008)	45
Figura 7: Representação esquemática do modelo de Ferreira, Morabito e Rangel (2009)	46
Figura 8: Representação esquemática do modelo de Ferreira, Almada-Lobo e Morabito (2013)	47
Figura 9: Comparação do problema-tema com modelos da literatura: a) Modelo de Toledo (2007); b) Modelo de Ferreira, Morabito e Rangel (2008); c) Modelo de Ferreira, Morabito e Rangel (2009) e Ferreira, Almada-Lobo e Morabito (2013); d) Problema-tema desse trabalho	48
Figura 10: Abordagem de modelagem proposta.....	50
Figura 11: Exemplo de funcionamento dos subperíodos.	53
Figura 12: Duração de cada fração de tempo (exemplo).....	61
Figura 13: Produção em cada fração de tempo.....	62
Figura 14: Comparação do resultado obtido em 10 minutos com o atual	80
Figura 15: Evolução do valor da solução do modelo em 10 minutos.....	80
Figura 16: Evolução da solução obtida em 6 horas	81
Figura 17: Programação da produção para células 1 a 3	83
Figura 18: Programação da produção para as células 4 e 5	84
Figura 19: Ótimo e soluções factíveis. a) Soluções factíveis dispersas. b) Soluções factíveis próximas.....	91

ÍNDICE DE TABELAS

Tabela 1: Comparação das alternativas	32
Tabela 2: Classificação do problema tema	42
Tabela 3: Tipos, tamanhos e especificações de fralda	52
Tabela 4: Demanda (exemplo)	59
Tabela 5: Tempos disponíveis (exemplo)	60
Tabela 6: Capacidade de produção por unidade de tempo (exemplo)	60
Tabela 7: Tempo de <i>setup</i> (exemplo)	61
Tabela 8: <i>Setup</i> inicial (exemplo)	61
Tabela 9: Especificações possíveis por máquina	66
Tabela 10: Demanda por especificação e tamanho de pacote	67
Tabela 11: <i>Setup</i> inicial de cada célula	68
Tabela 12: Tempos disponíveis	69
Tabela 13: Comparação do tempo de resolução exata em função das entradas	72
Tabela 14: Número de variáveis do modelo	73
Tabela 15: Valores das variáveis s em função das variáveis d	77
Tabela 16: Nova configuração de variáveis	77
Tabela 17: Comparação do número de variáveis	78
Tabela 18: Comparação entre 25 e 30 subperíodos	86
Tabela 19: Impacto da redução do tempo de cada tipo de <i>setup</i>	87
Tabela 20: Tempos de <i>setup</i>	97
Tabela 21: Capacidade de produção por unidade de tempo	101

SUMÁRIO

1	INTRODUÇÃO.....	23
2	CONTEXTO DA EMPRESA E FOCO DESSE TRABALHO	25
2.1	O mercado de produtos de higiene pessoal.....	25
2.2	Contexto operacional da empresa	26
2.3	Problema a ser estudado	30
2.4	Objetivo do trabalho	33
2.5	Justificativa para o trabalho	34
3	REVISÃO BIBLIOGRÁFICA.....	35
3.1	O problema	35
3.2	Literatura de problemas de dimensionamento e sequenciamento de lotes	35
3.3	Classificação do problema-tema.....	40
3.4	O problema com máquinas em paralelo em dois estágios	43
3.5	Características específicas do problema-tema	47
4	MODELO PROPOSTO.....	51
4.1	Modelo	51
4.2	Descrição do modelo	57
4.3	Testes preliminares	59
4.4	Validação	63
5	COLETA DE DADOS	65
6	RESULTADOS OBTIDOS.....	71
6.1	Proposta inicial de resolução	71
6.2	Nova proposta de resolução do problema.....	72
6.3	Efeito do número máximo de subperíodos	86
6.4	Análise de sensibilidade: Tempo de <i>setup</i>	87
6.5	Análise de sensibilidade: Capacidade dependente da combinação de pacotes	89

7	CONCLUSÕES	93
8	BIBLIOGRAFIA	95
	ANEXO 1: TEMPOS DE <i>SETUP</i>	97
	ANEXO 2: TAXAS DE PRODUÇÃO	101
	ANEXO 3: PRIMEIRO MODELO IMPLEMENTADO NO IBM OPL IDE.....	105
	ANEXO 4: SEGUNDO MODELO IMPLEMENTADO NO IBM OPL IDE.....	109

1 INTRODUÇÃO

O Brasil vive hoje uma crise econômica que tem um impacto significativo no consumo das famílias. Nesse contexto, em um setor competitivo como o de bens de consumo, aumenta a pressão por melhorias de eficiência e cortes de custos que visem compensar a queda natural das vendas para garantir a sobrevivência do negócio.

Nesse contexto adverso, projetos de melhoria operacional e menor uso de recursos são requisitados pelas empresas, como é o caso desse trabalho. Através da relação de longa data do departamento com o gerente de Planejamento e Programação da Produção de uma multinacional do setor de higiene pessoal, identificou-se uma oportunidade de melhoria do processo de dimensionamento e sequenciamento de lotes de fraldas.

Em conversas com esse gerente e sua equipe, foi identificado que *setups* longos e repetitivos são um difícil problema para essa unidade fabril, e que tem impactos no uso de recursos da empresa, seja em termos de horas de trabalho ou de perda de produção até que a linha produza dentro do padrão.

Dessa forma, o presente trabalho aborda a programação da produção de lotes de fraldas de uma multinacional do setor de higiene pessoal. Mais especificamente, ele trata do dimensionamento e sequenciamento de lotes para um sistema de células produtivas multiestágio em paralelo, tendo como principal objetivo reduzir o tempo de *setup* despendido hoje pela fábrica. Para alcançar esse objetivo, esse trabalho usará técnicas de Pesquisa Operacional para desenvolver um modelo que tenha resultado melhor que o da empresa. Considerando então esse problema e a técnica proposta, o presente trabalho é dividido em seis capítulos, além dessa introdução (capítulo um).

O capítulo dois discute o contexto da empresa, além de apresentar em detalhe o problema que se deseja resolver e o objetivo do trabalho.

No capítulo três é feita uma revisão bibliográfica onde é apresentado o problema geral de dimensionamento e sequenciamento de lotes (*General Lot-sizing and Scheduling Problem*, ou *GLSP*) e suas variantes presentes na literatura. Em seguida posiciona-se este trabalho em relação a essas variantes e faz-se uma revisão de literatura direcionada aos problemas com mais

de um estágio, terminando por apresentar as características específicas do problema tema desse trabalho.

Uma proposta de modelo de programação linear inteira mista é apresentada no capítulo quatro, onde também se apresenta um exemplo de como ele representa o presente problema.

O quinto capítulo apresenta os dados obtidos da empresa para o problema real, já de acordo com a notação apresentada no capítulo anterior.

O capítulo seis trata da resolução do modelo, apresentando também algumas melhorias na formulação do problema e a análise de sensibilidade de parâmetros chave.

Finalmente, as conclusões são apresentadas no capítulo sete, seguidas da bibliografia e dos anexos.

2 CONTEXTO DA EMPRESA E FOCO DESSE TRABALHO

Nesse capítulo apresenta-se a empresa, seu contexto operacional e discute-se o problema que buscar-se-á resolver nesse trabalho.

2.1 O mercado de produtos de higiene pessoal

O presente trabalho está inserido em um contexto amplo de Planejamento, Programação e Controle da Produção (PPCP) numa multinacional de produtos de higiene pessoal e limpeza doméstica. Essa multinacional, que solicitou não ser identificada, está no Brasil há algumas décadas e tem diversas fábricas e centros de distribuição localizados no país.

O setor de higiene pessoal no país é representado pela Associação Brasileira da Indústria de Higiene Pessoal, Perfumaria e Cosméticos (ABIHPEC). Segundo a ABIHPEC (2015) o setor de Higiene Pessoal, Perfumaria e Cosméticos (HPPC) teve um crescimento deflacionado composto de aproximadamente 10% (em R\$) nos últimos 19 anos; no entanto, o faturamento estabilizou em dólares nos últimos quatro anos devido à alta da cotação dessa moeda, conforme pode ser visto na Figura 1.

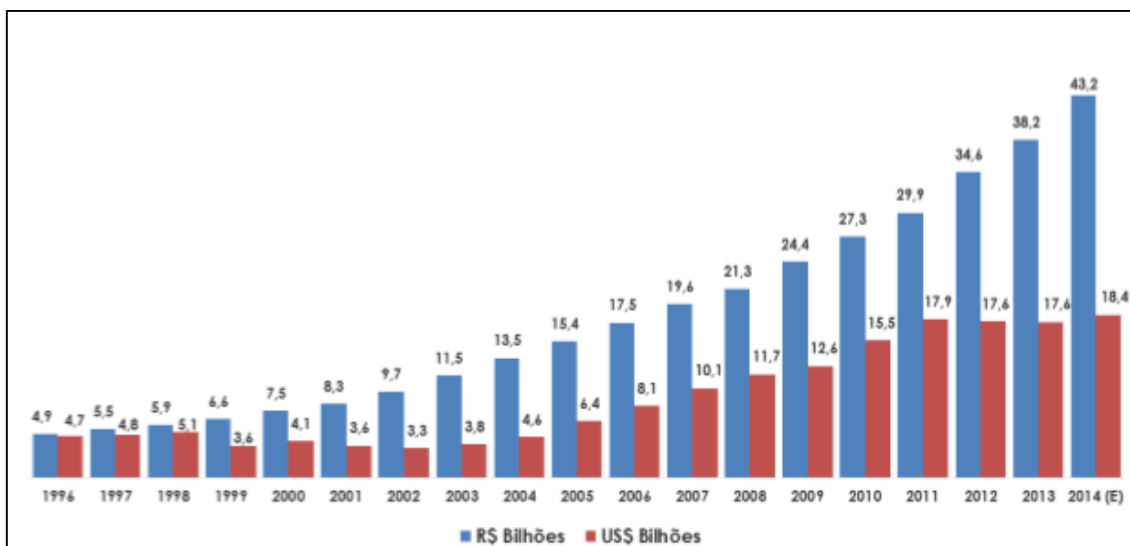


Figura 1: Evolução do setor (ABIHPEC 2015)

Esse crescimento se deu devido a seis fatores, segundo ABIHPEC (2015): melhoria de renda e consequente acesso das classes D e E a esses produtos; consumidores da classe C consumindo produtos de maior valor agregado; participação crescente da mulher no mercado de trabalho; aumento da produtividade e consequente redução dos preços (comparativamente à inflação);

lançamentos constantes; e aumento da expectativa de vida com consequente desejo de aparentar juventude.

Segundo também a ABIHPEC (2015), no Brasil, das 2540 empresas do setor, 20 tem faturamento líquido acima de R\$ 100 milhões, sendo por isso consideradas de grande porte. A empresa estudada faz parte desse grupo, que detém 73% do faturamento do setor, segundo ABIHPEC (2015).

Nesse trabalho, será estudada a produção de fraldas, dentro do segmento de produtos descartáveis de higiene pessoal. Segundo Kawamura (2011), esse segmento é caracterizado por consumo alto e frequente, relativo baixo valor agregado e qualidade similar entre produtos concorrentes. Ainda segundo esse autor, como os produtos são de consumo diário, e levando em conta a semelhança entre concorrentes, indisponibilidade no ponto de venda em geral significa perda de venda, o que tem como consequência adoção do modelo *make-to-stock*¹ de produção. Nesse modelo, produtos são feitos para atingir uma meta de estoque (produção empurrada), sendo que os clientes consomem então desse estoque. Está implícito portanto, nesse modelo, um alto custo de falta de estoque (“*stock out*”), que se deseja minimizar ou eliminar.

O mercado de fraldas, em específico, tem demanda bastante estável dada a natureza de uso contínuo do produto e sua imprescindibilidade para seus usuários. É também por causa disso que a fácil substituição entre marcas, característica desse setor, é ainda mais acentuada em fraldas.

2.2 Contexto operacional da empresa

Como esse trabalho irá focar na produção de fraldas, precisa-se primeiramente definir suas características fundamentais. Para o contexto dessa discussão, uma fralda (um *SKU* de fralda ou *Stock Keeping Unit*) é caracterizada por um tamanho (como por exemplo P, M, G, etc), um tamanho de pacote (que indica quantas fraldas têm dentro de um pacote, por exemplo 24 fraldas por pacote) e um tipo, como por exemplo básica ou especial, que na prática são traduzidas em outras características da fralda que incluem, mas não são limitadas a: material, cor, estampa, acabamento, número de folhas, nível de absorção, etc.

¹ São usados os termos em inglês, pois eles são mais comuns mesmo na literatura nacional.

O ambiente de produção dessa empresa inclui 6 conjuntos de 3 máquinas trabalhando em paralelo em três turnos (24 horas por dia e 7 dias por semana), sendo uma produtora e duas embaladoras por conjunto, conforme esquema na Figura 2.

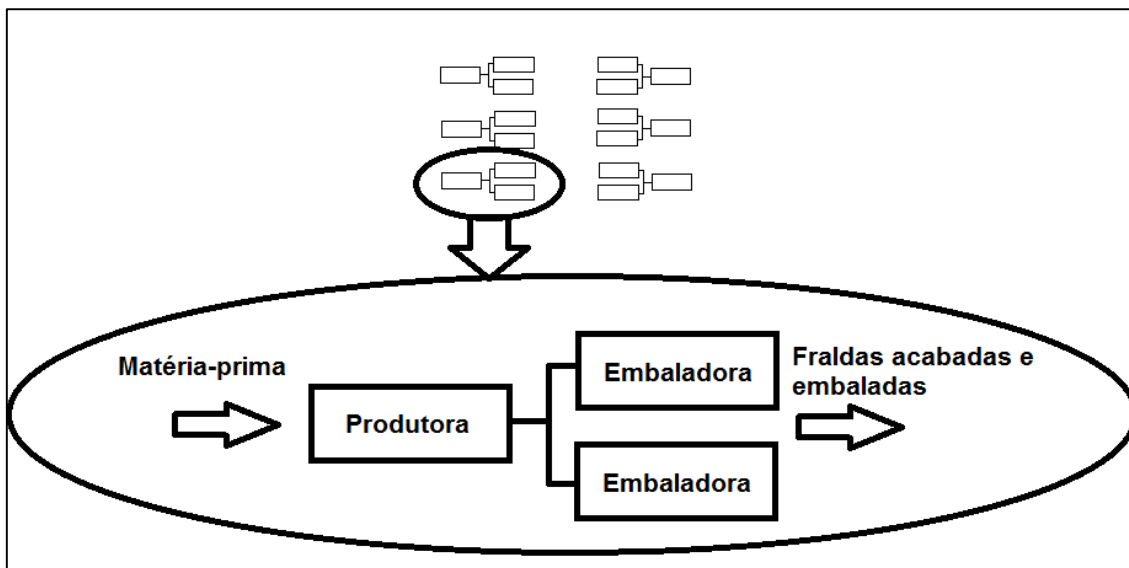


Figura 2: Esquema de produção do setor estudado

As fraldas podem ser de tipos e tamanhos diferentes, de modo que várias combinações diferentes podem ser processadas pela máquina produtora. As produtoras são diferentes entre si, tendo diferentes capacidades de produção além de estarem limitadas a alguns tipos de produtos.

A produtora provê as fraldas que alimentam as embaladoras. As embaladoras acumulam a quantidade adequada de fraldas de um certo tamanho e então as empacotam. Há vários tamanhos de pacote, sendo que um mesmo tamanho de pacote terá quantidades variando de acordo com o tamanho da fralda que contém². As embaladoras podem embalar pacotes de tamanhos diferentes em paralelo, contanto que a fralda em si seja a mesma, devido à existência de apenas uma produtora em cada conjunto. Não há estoque entre as etapas, o que não gera interrupções pois a capacidade das embaladoras é maior que a da produtora, no entanto isso implica que para balancear as embaladoras a proporção de fraldas que vão para cada uma

² Ou seja, um pacote de um certo tamanho contendo fraldas grandes terá menos fraldas que o mesmo pacote contendo fraldas pequenas

depende da combinação entre tamanhos de pacote. No decorrer desse trabalho o conjunto de uma produtora e suas duas respectivas embaladoras será considerado uma célula produtiva.

A cada mês é feita uma previsão de vendas para o mês seguinte, que então alimenta um modelo de planejamento da produção baseado no cálculo de necessidades, dado pelo balanço de estoques apresentado na equação a seguir.

$$P_i = E_i - E_{i-1} + D_i$$

Nesta equação, P_i representa a produção no mês i , E_i o estoque desejado no final do mês i , E_{i-1} o estoque disponível no final do mês $i - 1$ e D_i a demanda prevista para o mês i . O estoque desejado para o fim do mês i é definido como uma porcentagem fixa sobre a previsão de vendas do mês $i + 1$ que o setor comercial envia para a produção. Essa porcentagem já é pré-definida para balancear o custo de estocagem.

O sequenciamento de produção é realizado com base no plano de produção anterior, porém é limitado por vários fatores, sendo o tempo de *setup* o mais importante desses fatores pela sua duração de várias horas. Os *setups* dependem da mudança de características do produto que está sendo feita, por exemplo uma troca de tamanho da fralda demora 12 horas, enquanto uma mudança de tipo de fralda dura 6 horas. Essas duas mudanças ocorrem na máquina produtora, porém pode ocorrer uma mudança do tamanho do pacote na embaladora, que dura 4 horas.

O *setup* de mudança de tamanho de fralda é demorado pois as folhas que compõe a fralda devem ter tamanho diferente também, de modo que o processo exige então a calibragem de todas as etapas de produção para esse novo tamanho. Nota-se ainda que as primeiras fraldas não saem de acordo com os padrões de qualidade, de modo que no processo de *setup* há gastos com insumos que se transformam em produtos que acabam sendo descartados.

Uma mudança de tipo de fralda é mais simples e, portanto, mais rápida, pois há apenas a troca de algumas matérias-primas, ou até mesmo da mesma matéria-prima com uma estampa diferente, de modo que apenas pequenos ajustes são necessários até se obter produtos de qualidade.

Uma mudança de tamanho de pacote é ainda mais simples pois envolve apenas as embaladoras, que têm menos etapas internas que as produtoras. Além disso, os requisitos de precisão dimensional são muito menores para a embalagem do que para o produto em si, tornando os ajustes mais fáceis.

Os tempos de *setup* da produtora e das embaladoras não se somam se mais de uma mudança é feita, sendo o tempo mais longo o que prevalece. Ou seja, uma mudança de tamanho e tipo de fralda dura também 12 horas, uma mudança de tipo de fralda e tamanho de pacote dura apenas 6 horas e uma mudança do tamanho de pacote nas duas embaladoras dura tanto quanto a mudança de só uma delas (ou seja, 4 horas e não 8 horas). Para o primeiro caso isso ocorre porque os ajustes para uma mudança de tipo estão de certa forma contidos nos ajustes para uma mudança de tamanho de fralda. Nos outros casos isso ocorre pois os tempos de *setup* são considerados altos e por isso a empresa tem uma política de usar mais operadores quando necessário para não aumentar esse tempo.

Outro ponto importante é que dentro da célula produtiva há uma dependência total e síncrona entre a produtora e suas duas embaladoras para ocorrência do fluxo. Isso advém da não existência de estoque intermediário no interior da célula, de modo que uma parada em uma das embaladoras, por exemplo, faz com que a produção da célula seja parada. Além disso, caso ocorra *setup* na produtora, *setups* adicionais na embaladora podem ser feitos concomitantemente, sem que o processo se atrase.

Os tempos de *setup* apresentados são padrões tabelados para a fábrica, tendo sido definidos de forma conservadora através de um acordo com o sindicato dos trabalhadores para que não haja pressão para realizá-los de forma mais rápida e potencialmente perigosa para os trabalhadores. Desse modo o *setup*, para o contexto desse trabalho, é determinístico e relativo a célula inteira e como já foi mencionado, pode durar 4 horas (mudança de tamanho do pacote em pelo menos uma das embaladoras), 6 horas (mudança de tipo de fralda incluindo ou não mudança de tamanho de pacote) ou 12 horas (mudança de tamanho de fralda, podendo mudar também o tipo de fralda e tamanhos do pacote).

Para realizar o sequenciamento e dimensionamento de lotes, o planejador primeiramente faz o planejamento da produção conforme o método do cálculo das necessidades descrito na equação de balanço de estoques, usando para isso uma planilha eletrônica de cálculo. Depois, com a produção planejada para o mês (P_i) ele faz a programação, seguindo sua própria experiência, em outra planilha.

Deve-se notar que certos aspectos que influenciam nas decisões de sequenciamento e definição dos tamanhos dos lotes de produção estão apenas parcialmente disponíveis para os programadores. Eles têm uma noção de quando certos tipos de matéria-prima vão chegar,

pedidos urgentes e limites de estoque. No entanto a informação não está disponível de maneira numérica na hora de tomada de decisão, de modo que o operador trabalha com as informações coletadas previamente e apenas checa posteriormente se há coerência com um sistema MRP.

2.3 Problema a ser estudado

Observando a ordem de grandeza dos tempos de *setup*, com células produtivas paradas por até meio dia, buscou-se avaliar as consequências desse tempo no objetivo da empresa para o setor. Esse objetivo, segundo o gerente de planejamento da produção da empresa, é aumentar a produção, diminuindo o uso de recursos produtivos e seu custo. De fato, ao avaliar o plano de produção fornecido para aproximadamente um mês, notou-se que do tempo disponível por célula produtiva (em média, descontadas as horas comprometidas com manutenção preventiva e teste do setor de inovação), 92% foi usado para produção e 8% para *setups*.

Além do consumo de tempo, *setups* geram custos de pessoal, pois operadores precisam ser destacados para realiza-los. Desse modo, reduzir o tempo total gasto com *setups* (seja diminuindo sua quantidade ou aumentando a predominância de *setups* mais curtos) implica em diminuir o número de homens-hora usados para essa função e o custo associado a elas.

Adicionalmente, há o custo do material desperdiçado dentro do período de *setup* até que a máquina esteja produzindo de acordo com as especificações. Esse custo, desse modo, também está relacionado com os *setups*. Por fim, deve-se considerar que conforme foi explicado na seção anterior, os consumidores trocam de marca facilmente, de forma que diminuir o tempo consumido com *setup* permite disponibilizar recursos para cumprir pedidos extraordinários, diminuindo a chance de perda de venda.

Desse modo, ineficiências na programação da produção que gerem maior tempo despendido em *setups*, implicam em resultados piores para essa linha de produto. Assim, um método eficaz de sequenciamento e dimensionamento de lotes, que seja capaz de reduzir o tempo empregado em *setups*, é desejado pelos responsáveis pelo planejamento, programação e controle da produção dessa empresa.

Além da possibilidade de produzir o mesmo produto em diferentes células com taxas de produção diferentes, um aspecto desafiador desse problema é que mesmo dentro da mesma célula, pode-se produzir simultaneamente dois tamanhos de pacote diferentes (um em cada embaladora). Considerando por exemplo dois pacotes de tamanho A e B que contém a mesma fralda (ou seja, mesmo tipo e tamanho de fralda). Uma célula pode produzir A e depois B, B e

depois A ou A e B simultaneamente. Além disso, a taxa de produção quando as embaladoras empacotam tamanhos diferentes dependem da combinação em questão, sendo feitos ajustes para manter uma proporção adequada entre fraldas que permita um fluxo contínuo de fraldas sem estoques intermediários, que não são possíveis nessas células³.

Suponha uma taxa de produção de A de 20 pacotes por hora (pctes/h), de B 10 pctes/h e que quando A e B são produzidas juntas, produz-se 10 pctes/h de A e 4 pctes/h de B. Suponha ainda que como trata-se de uma mudança de tamanho de pacote, o *setup* é de 4 horas. Se a demanda para um dia é de 220 pctes de A e 80 pctes de B, produzindo A e depois B, temos o plano apresentado na Figura 3.

Hora	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Embaladora 1	10	10	10	10	10	10	10	10	10	10	10					5	5	5	5	5	5	5	5	
Embaladora 2	10	10	10	10	10	10	10	10	10	10	10					5	5	5	5	5	5	5	5	
Legenda																								
K	Produção de K unidades de A																							
K	Produção de K unidades de B																							
	Setup																							

Figura 3: Programação de A e depois de B

Nesse caso a célula produz primeiramente o pacote A nas duas embaladoras por 11 horas, a uma taxa de 20 pctes/h, totalizando a demanda de 220 pctes, faz um *setup* de 4 horas e depois atinge a demanda de 80 pctes de B produzindo 10 pctes/h por 8 horas, sobrando ao final 1 hora. No entanto, há uma segunda alternativa, que é produzir os dois pacotes simultaneamente, conforme mostrado na Figura 4.

³ As embaladoras tem espaço para acumular o número de fraldas até completar o pacote, mas isso não é um estoque intermediário e sim parte do processo. Após completar a quantidade de fraldas elas finalizam o processo (fecham o pacote) e fraldas novas não são admitidas. É nesse sentido que se afirma que essa célula não tem estoque intermediário.

Hora	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Embaladora 1	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10			
Embaladora 2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4		

Legenda	
K	Produção de K unidades de A
K	Produção de K unidades de B
	Setup

Figura 4: Programação de A e B

Essa programação difere em vários aspectos da anterior. Primeiramente, deve-se notar que ao fazer os dois produtos ao mesmo tempo, para atingir a demanda de A seriam necessárias 22 horas, porém para atingir a de B, apenas 20 horas, de modo que essa alternativa gera superprodução de B (são produzidos 8 ptes a mais que o necessário). A superprodução, apesar de parecer um desperdício, hoje não é considerada como problemática pela empresa. Isso acontece pois a própria política de produção da empresa é *make-to-stock* e eles mesmos já pressupõe que pode haver variação no estoque final e não vêm problema nisso, principalmente pois a armazenagem hoje não é um gargalo.

Além disso, apesar da produção por embaladora por máquina ser menor (para o produto B), o tempo ocioso ao final das 24 horas é maior, justamente pois fazer os dois pacotes ao mesmo tempo evita o *setup* de 4 horas. Sendo assim, enquanto o tempo produzindo aumentou de 19 horas para 22 horas, o tempo de *setup* caiu de 4 horas para 0, diminuindo o tempo total de 23 horas para 22 horas. Esses dados, resumidos na Tabela 1, indicam que em alguns casos há um *trade-off* parcial entre o tempo produzindo e o tempo em *setup*, fazendo com que a solução ótima não seja trivial.

Tabela 1: Comparação das alternativas

Tempo (horas)	A e depois B	A e B juntos
Produzindo	19	22
Setup	4	0
Total	23	22

Por fim, a informação disponível apenas parcialmente para o programador e não de forma sistemática, numérica e completa, imprime um desafio para o uso de métodos quantitativos, que naturalmente precisam de dados. Custos de perdas de venda, por exemplo, não estão

disponíveis, assim como informações consolidadas sobre pedidos pendentes. Parte dessas informações o programador tem de forma aproximada, de forma que certa flexibilidade de entradas para o modelo é desejada. Sendo assim, a ausência de dados e a flexibilidade para o operador são desafios adicionais para a programação da produção. Considerando tudo isso, define-se na próxima seção o objetivo do trabalho.

2.4 Objetivo do trabalho

Considerando o problema descrito na seção anterior, o objetivo do presente trabalho é desenvolver um modelo matemático que represente o sistema produtivo apresentado e seja capaz de dimensionar e sequenciar os lotes de produção nas células produtivas. Para que o trabalho atinja totalmente seu objetivo, esse modelo deve não só gerar esse programa da produção, mas também fazê-lo atingindo resultados melhores do que os atuais. Desse modo, o modelo será aplicado em um conjunto de dados fornecido pela empresa e seus resultados serão comparados com os reais para verificar se, de fato, o modelo cumpre seu objetivo.

Além disso, o modelo deverá ser desenvolvido considerando a limitação de informações disponíveis de forma objetiva. O modelo deve permitir que o programador tenha flexibilidade de separar parte do tempo produtivo para pedidos urgentes ou deixar para o fim do mês a produção de lotes que dependem de matéria-prima não disponível. Lembrando que isso acontece no dia-a-dia da empresa pois certas informações disponíveis ao programador (estimativa de estoque, pedidos urgentes, etc.) não estão tabeladas, de modo que não seria possível usá-las no modelo diretamente.

Para fazer a comparação, além de considerar-se as limitações de informação, deve-se definir uma métrica pela qual o modelo poderá ser comparado com o método atual usados pelos programadores. Conforme apresentado anteriormente, os *setups* são o grande problema apresentado hoje por esse setor da empresa. Ao diminuir os *setups* em quantidade e tempo, aumenta-se o tempo disponível para produção e a flexibilidade para lidar com pedidos imprevistos e, além disso, diminui-se os custos de mão-de-obra e de falta de estoque. Desse modo, foi definida como métrica principal do trabalho o tempo total gasto em produção e *setups* por período de tempo.

Essa medida, ao incluir tanto o tempo de produção efetiva quanto o tempo de *setup*, faz com que o modelo seja capaz de avaliar o *trade-off* entre produzir dois pacotes diferentes na mesma célula (um em cada embaladora) ou produzir apenas um tipo de pacote nas duas embaladoras e

depois o outro tipo. Esse *trade-off*, discutido em mais detalhes na seção anterior, é uma das chaves desse problema, e por isso a métrica foi definida para contemplá-lo, sendo que uma minimização apenas do tempo de *setup* poderia vir a custa de um aumento do tempo médio gasto produzindo cada unidade.

É importante notar que essa métrica não é a única possível, já que ela não inclui, por exemplo, as consequências da superprodução. No entanto, considerando o quadro apresentado em que superprodução não é um grande problema, e levando em conta as limitações de informações disponíveis para levar em conta outros fatores, essa métrica foi proposta. Para garantir que isso corresponde às necessidades reais, ela foi apresentada para o responsável pelo planejamento e programação da produção da empresa, que a julgou válida e relevante.

2.5 Justificativa para o trabalho

Em um mercado competitivo como o de higiene pessoal, ganhos de eficiência operacional podem resultar em menores custos ou maior disponibilidade, que se traduzem em maiores margens e mais vendas. Nesse sentido, o presente trabalho busca aumentar a eficiência através de algoritmos de sequenciamento e dimensionamento de lotes de produção que diminuam o tempo gasto atualmente em *setups*, que chegam a durar até meio dia inteiro de trabalho apenas para uma mudança e consomem quase 10% do tempo produtivo da empresa, com diversos custos associados.

Melhorias nesse sentido teriam, portanto, impacto direto no lucro da unidade de produção além de gerarem maior uniformidade e potenciais externalidades positivas no uso de uma solução similar em outras unidades.

3 REVISÃO BIBLIOGRÁFICA

Neste capítulo é realizada uma revisão da literatura relacionada ao trabalho. Partindo do problema apresentado no capítulo anterior e suas características, apresenta-se então uma breve revisão dos modelos matemáticos relacionados e suas principais características. A partir dela, classifica-se o presente problema e aprofunda-se em problemas com máquinas em dois estágios, por fim diferenciando o problema atual dos da literatura.

3.1 O problema

Esse trabalho trata de um problema no qual, para satisfazer uma certa demanda para um período, deve-se alocar a produção de certas quantidades de produtos diferentes para cada máquina ao longo do tempo. Procura-se então decidir quanto de cada produto, em cada máquina, em cada intervalo de tempo irá ser produzido. Como deve-se decidir conjuntamente o quanto e o quando produzir, esse problema é conhecido como problema integrado de dimensionamento e sequenciamento de lotes.

No caso do presente estudo, busca-se diminuir o uso de recursos produtivos atendendo-se a demanda com o menor tempo somado de produção e *setup*. Para isso, deve-se avaliar não só que tipo e tamanho de fraldas cada produtora irá realizar, mas também para qual tamanho de pacotes cada uma das embaladoras vai empacotar. Desse modo, cada célula, composta por uma produtora e duas embaladoras, pode produzir dois produtos diferentes contanto que a única diferença seja o tamanho do pacote. Esse é um grande diferencial desse problema em relação à literatura, como veremos nas seções a seguir.

3.2 Literatura de problemas de dimensionamento e sequenciamento de lotes

Diversos trabalhos já foram publicados revisando a literatura sobre dimensionamento e sequenciamento de lotes, muitas vezes com enfoques específicos. Drexler e Kimms (1997) fazem uma revisão abrangente de dimensionamento e sequenciamento, enquanto Quadrt e Kuhn (2009) focam-se nos problemas com limitação de capacidade. Meyr (2002) por sua vez aborda problemas de máquinas em paralelo, expandindo o trabalho de Meyr (2000) em que esse mesmo autor havia proposto o GLSP (*General Lot-sizing and Scheduling Problem*), que modela uma única máquina.

Assim como Saboia (2010) e Kawamura (2011) que também trataram de problemas de dimensionamento e sequenciamento de lotes na indústria de bens de consumo de higiene pessoal, esse trabalho usará a caracterização proposta por Karimi, Fatemi Ghomi e Wilson (2003) para melhor posicionar o problema a ser tratado à luz da literatura. Essa caracterização é apresentada a seguir, sendo que na seção seguinte far-se-á o posicionamento do problema em relação ao apresentado.

a) Demanda

A demanda representa a quantidade de cada produto que os clientes (internos ou externos) desejam em cada período.

i) Variação ao longo do tempo

A demanda pode ser constante ao longo do tempo, caso em que é considerada estática, ou variar, sendo portanto considerada dinâmica. Casos em que a demanda varia de forma constante (por exemplo sazonalidade) podem levar a simplificações na modelagem, porém demanda dinâmica normalmente se refere a uma demanda que varia ao longo do tempo devido a fatores que também variam (como promoções, mudanças de gosto, etc.). Segundo Karimi, Fatemi Ghomi e Wilson, problemas com demanda dinâmica são mais complexos que os de demanda estática.

ii) Previsibilidade

A demanda pode ser tanto determinística, caso no qual ela é prevista com antecedência e esse valor é tomado como verdade na hora da modelagem, ou probabilística, quando na hora da modelagem considera-se uma distribuição de probabilidades para essa demanda. É importante notar que uma demanda pode ser determinística e dinâmica, bastando para isso que ela varie ao longo do tempo, mas que no momento de modelagem ela seja conhecida para os períodos dentro do horizonte de planejamento. Segundo Karimi, Fatemi Ghomi e Wilson, problemas com demanda probabilística são mais complexos que os de demanda determinística.

iii) Dependência da demanda

Quando há múltiplos níveis, a demanda normalmente é dependente, ou seja, a demanda de um nível depende da demanda do próximo. Um exemplo disso são fábricas de automóveis, onde a linha de montagem final depende de linhas auxiliares, cuja demanda é dependente da demanda principal. A

maioria dos problemas tem apenas um nível e, portanto, demanda independente. Segundo Karimi, Fatemi Ghomi e Wilson, problemas com demanda dependente são mais complexos que os de demanda independente.

b) Horizonte de planejamento

Horizonte de planejamento refere-se, em planejamento e programação da produção, ao momento no futuro até qual se deseja planejar a produção. Ou seja, o objetivo de modelo é obter um plano para todos os períodos até esse horizonte.

i) Finito ou infinito

Um horizonte de planejamento pode ser finito ou infinito. Drexler e Kimms (1997) afirmam que pesquisas em dimensionamento de lotes começam com o modelo clássico de lote econômico, um exemplo de horizonte de planejamento infinito. Em situações de demanda estática esses modelos funcionam, porém em situações em que a demanda é dinâmica, e portanto deve ser revisada constantemente, adota-se modelos com horizonte finito e ligado a capacidade de previsão dos parâmetros associados a ele.

ii) Monitoramento contínuo ou discreto

O modelo pode ser monitorado a todo instante (se há equações que descrevem seu comportamento de forma contínua), ou em intervalos discretos. Esse último é mais comum pela própria característica dos sistemas de produção, ligados a pedidos discretos e metas associadas a intervalos constantes de tempo (por exemplo atender a demanda de um determinado mês).

iii) Subperíodos de tempo

Os modelos são usualmente divididos em períodos, para o qual estão associadas métricas (estoque ao final da semana, por exemplo), que por sua vez são muitas vezes associados a subperíodos, que permitem efetivamente o sequenciamento da produção. Modelos *small bucket* são aqueles no qual apenas um produto é produzido por subperíodo, enquanto *big bucket* refere-se a mais de um produto. Nesse caso, ganha-se em performance computacional porém perde-se em qualidade e detalhamento da solução, como destacado por Quadt e Kuhn (2009). Além disso, os subperíodos podem ter duração fixa ou dependente da quantidade.

c) Máquinas

Máquinas referem-se genericamente aos recursos produtivos fixos pelos quais os produtos devem passar. Vale notar que uma máquina, para efeitos de modelagem, pode muitas vezes representar uma linha inteira de produção automatizada, já que uma vez que o produto entra nela ele sofre uma sequência de transformações sempre na mesma sequência e quantidade. Ao mesmo tempo, se há possibilidade de rotas dentro de uma máquina real, pode ser conveniente quebrá-la em diversas máquinas virtuais para efeito de modelagem.

i) Capacidade

Refere-se a existência ou não de limites para a máquina (ou para qualquer outro recurso produtivo, como por exemplo mão-de-obra). Quando existem limites, por exemplo uma taxa de produção, o problema é chamado de *capacitated*, caso contrário, ele é conhecido como *uncapacitated*. Enquanto um exemplo de problema *uncapacitated* é o modelo clássico de lote econômico, a maioria dos problemas reais e da literatura, tem restrição de capacidade. Deve-se ressaltar que, segundo Karimi, Fatemi Ghomi e Wilson, problemas com restrição de capacidade são mais complexos.

ii) Quantidade

Os modelos são usualmente divididos em de uma só máquina e de várias máquinas. Zhu e Wilhelm (2006) pontuam que boa parte dos trabalhos publicados referem-se a uma só máquina, como por exemplo o trabalho de Toso e Morabito (2005). Já Meyr (2002) esclarece que problemas multi-máquinas não são frequentemente tratados na literatura. Deve-se notar ainda que a maioria dos modelos de múltiplas máquinas baseia-se em máquinas em paralelo. Casos diferentes desse são tratados dentro de “Níveis de produção”, de modo que se deve enfatizar que nesse item foca-se em múltiplas máquinas por nível de produção.

iii) Diferenciação entre máquinas

Se há múltiplas máquinas é importante verificar se essas máquinas são idênticas ou não, sendo que as máquinas podem ser ou não capazes de produzir certos produtos, terem taxas de produção diferentes, *setup* diferentes, etc. Quadt e Kuhn (2009) exemplificam um modelo com máquinas iguais, enquanto Meyr (2002), máquinas diferentes.

d) Possibilidade de não atender pedidos

Devido a restrições de capacidade (ou mesmo buscando soluções de menor custo total), alguns problemas tratam da possibilidade de não atender ou atender com atraso a demanda (*backorder*). Para isso são introduzidas punições referentes a cada caso que representam, usualmente, o custo da venda não realizada ou atrasada, incluindo a perda de boa vontade do cliente (*goodwill*). Problemas que permitem essa possibilidade são mais complexos segundo Karimi, Fatemi Ghomi e Wilson. Ferreira, Morabito e Rangel (2008) e Quadt e Kuhn (2009) trabalham com essa possibilidade.

e) Produtos

Os modelos podem ter apenas um produto ou múltiplos. Além disso estes podem ser perecíveis ou não.

i) Quantidade de tipos de produtos

Os modelos podem ter um ou mais tipos de produtos, sendo que adicioná-los torna o modelo mais complexo segundo Karimi, Fatemi Ghomi e Wilson (2003). Os dois casos são bastante comuns.

ii) Perecibilidade

Produtos perecíveis introduzem uma complexidade para os modelos pois estocagens longas não são permitidas, sob o risco de perder os produtos. Nesse tipo de modelo, a coordenação entre produção e demanda é ainda mais crítica do que em casos onde o simples custo de estocagem é considerado.

f) Estrutura de *setup*

A complexidade da estrutura de *setup* é um fator bastante discutido na literatura, pois leva a introdução de variáveis binárias no modelo. Essas, por sua vez, causam grande impacto na performance computacional do mesmo, conforme discutido por Karimi, Fatemi Ghomi e Wilson.

i) Estrutura de *setup* simples ou complexa

Uma estrutura simples de *setup* ocorre quando o tempo (e o custo) dessa mudança são independentes da sequência, como em Quadt e Kuhn (2009). Já uma estrutura complexa corresponde a um caso de dependência do tempo/custo de *setup* da sequência de produção, caso discutido em Kawamura (2011).

ii) Existência de *setup carry-over*

Setup carry-over representa a manutenção do estado de *setup* entre períodos ou subperíodos. Ou seja, se a máquina terminou o período $n - 1$ preparada para produzir um produto específico, ela precisará de um *setup* no início do próximo período para produzir qualquer outro item.

iii) Obediência à desigualdade triangular

O *setup* ainda pode respeitar ou não a desigualdade triangular, o que significa que o tempo de *setup* entre quaisquer dois estados A e B é menor do que o de A para um outro estado qualquer C e depois para B. Casos de *setup* que não respeitam essa condição são reportados por exemplo em Toso, Morabito e Clark (2008) para uma fábrica de ração, onde ocorrem processos que limpam o equipamento.

g) Níveis de produção

Modelos de um só nível são aqueles em que, na perspectiva do modelo, ocorre a transformação da matéria-prima diretamente em produto final após uma única etapa. Ressalta-se que isso depende das escolhas de modelagem, ou seja, uma linha automatizada é composta de diversos estágios, possivelmente inclusive com várias entradas, mas dependendo do tipo de problema ela pode ser encarada como uma única máquina e esse sistema ser modelado como de um único nível. Pochet e Wolsey (2006) classificam esses problemas em sequencial (uma entrada formando uma saída, passando por diversos estágios), montagem (várias entradas formando uma única saída) ou geral (sem estrutura definida). Karimi, Fatemi Ghomi e Wilson (2003) lembram que problemas multi-níveis são muito mais complexos do que os que tem apenas um nível.

Considerando a miríade de características diferentes entre os problemas da literatura, na próxima seção classifica-se o problema tema desse trabalho para que se faça então uma revisão de outros textos direcionados ao problema em questão.

3.3 Classificação do problema-tema

Tendo em vista o sistema de classificação apresentado na seção anterior, adaptado e ampliado de Karimi, Fathemi Ghomi e Wilson (2003), Saboia (2010) e Kawamura (2011), pode-se assim classificar o presente trabalho.

a) Demanda

No presente trabalho busca-se fazer o sequenciamento e dimensionamento de lotes para apenas um mês (ou seja, a cada novo mês o modelo deve ser rodado novamente). Sendo assim, para o mês em questão a demanda é estática. Além disso, a demanda é assumida como determinística para o planejamento da produção. Por fim, como a demanda da produtora é dependente da das embaladoras, a demanda nesse caso é considerada dependente.

b) Horizonte de Planejamento

O planejamento é feito para apenas um mês, fazendo com que o modelo seja de horizonte de planejamento finito. Além disso, não há equações que o descrevem a todo instante, sendo, portanto, considerado de monitoramento discreto. Por fim, são adotados subperíodos nos quais apenas um produto pode ser produzido, ou seja, um modelo *small bucket* em relação aos subperíodos. No período em si, mais de um produto pode ser produzido, então em relação ao período o modelo é *big bucket*. Os subperíodos têm tamanho variável.

c) Máquinas

Neste trabalho têm-se diversas máquinas de capacidade limitada, sendo as produtoras diferentes entre si e as embaladoras associadas a cada produtora idênticas entre si. Cada conjunto de uma produtora e duas embaladoras, chamado de célula produtiva, está em paralelo com outras células.

d) Possibilidade de não atender pedidos

Pedidos atrasados não são permitidos pelo modelo, mas poderão ser levados em conta na resolução do problema ao diminuir-se o tempo disponível de programação, deixando o resto do tempo separado para pedidos atrasados.

e) Produtos

Trabalha-se com vários produtos, sendo classificados em três níveis: tipo da fralda, tamanho da fralda (ambos decididos na produtora) e tamanho do pacote (decidido na embaladora). Os produtos não são perecíveis.

f) Estrutura de *setup*

Como o *setup* é dependente da sequência (sendo diferente entre mudanças de tipo de fralda, tamanho de fralda ou tamanho de pacote) ele é considerado complexo ou dependente. Como modela-se apenas um período, não há *carry-over*, mas considera-se que há um *setup* inicial na máquina que deve ser respeitado. Por fim, um único

setup direto tem sempre tempo inferior a dois outros *setups* que levassem a mesma transição, de modo que a desigualdade triangular é respeitada.

g) Níveis de produção

O problema do presente trabalho refere-se a células produtivas (conjuntos de uma produtora e duas embaladoras) que são paralelos entre si. Ao mesmo tempo, dentro de cada célula o fluxo começa na produtora e se divide nas duas embaladoras. Dessa forma, esse problema pode ser considerado de diversos níveis. No entanto, esse arranjo não é nenhum dos tipos específicos apresentados por Pochet e Wolsey (2006), ou seja, não é sequencial nem montagem, encaixando portanto na última categoria (geral).

A classificação do problema tema é resumida na Tabela 2.

Tabela 2: Classificação do problema tema

Item	Sub-item	Classificação
Demanda	Variação ao longo do tempo	Estática
	Previsibilidade	Determinística
	Dependência da demanda	Dependente
Horizonte de planejamento	Finito ou infinito	Finito
	Monitoramento contínuo ou discreto	Discreto
	Subperíodos de tempo	<i>Small-bucket</i> de tamanho variável
Máquinas	Capacidade	Limitada
	Quantidade	Várias em série e paralelo
	Diferenciação entre máquinas	Máquinas diferentes entre si
Possibilidade de não atender pedidos	-	Inexistente
Produtos	Quantidade de tipos de produtos	Vários produtos
	Perecibilidade	Imperecíveis

Item	Sub-item	Classificação
Estrutura de <i>setup</i>	Estrutura simples ou complexa	Complexa
	<i>Setup carry-over</i>	Apenas <i>setup</i> inicial
	Obediência à desigualdade triangular	Obedece
Níveis de produção	-	Vários níveis (caso geral)

Dentre as características levantadas, a que traz maior complexidade ao problema, e ao mesmo tempo uma das menos exploradas na literatura, é a existência de diversos níveis de produção, com máquinas em série e em paralelo. Sendo assim, faz-se a seguir uma revisão da literatura direcionada especificamente a esse tipo de problema.

3.4 O problema com máquinas em paralelo em dois estágios

Como visto na seção anterior, o problema desse trabalho pode ser classificado como de máquinas em paralelo em dois estágios. Na literatura são encontrados alguns exemplos de trabalhos com essas características. Toledo (2005), por exemplo, apresenta um modelo de programação linear inteira mista com essas características, baseando-se no GLSP.

Toledo, França, Morabito e Kimms (2007) aprofundam essa abordagem. O modelo desenvolvido por esses autores, que contém 65 conjuntos de restrições, inova ao juntar características de vários modelos em um só. A representação matemática em questão foi desenvolvida para fábricas de refrigerante onde existem tanques (estágio 1) e linhas de produção (estágio 2). Nesse modelo, todos os tanques podem, a princípio, atender todas as linhas de produção. Esse arranjo é representado esquematicamente na Figura 5.

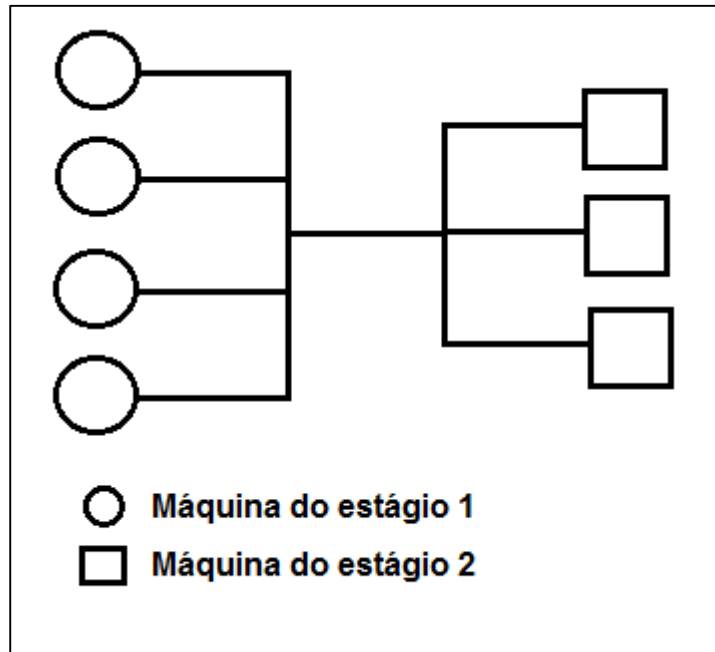


Figura 5: Representação esquemática do modelo de Toledo *et al.* (2007)

No entanto os próprios autores concluem que devido a alta complexidade do modelo ele não atinge o ótimo em tempo computacional de 1 hora mesmo para pequenas instâncias geradas por eles. Esses autores propõem então alternativas heurísticas de relaxamento baseadas em Gupta e Magnusson (2005), onde variáveis binárias são relaxadas para poderem assumir valores reais. Consegue-se assim resolver o problema parcialmente, obtendo-se boas soluções para problemas pequenos, porém havendo dificuldades até para encontrar soluções factíveis para instâncias maiores.

Também no contexto de fábricas de refrigerantes e observando a complexidade do modelo de Toledo (2007), Ferreira Morabito e Rangel propuseram algumas simplificações capazes de resolver satisfatoriamente problemas reais. Ferreira, Morabito e Rangel (2008) executam um relaxamento da solução de Toledo (2007) e uma significativa simplificação da complexidade do modelo. Para uma fábrica pequena, eles consideraram apenas uma máquina no segundo estágio, de modo que temos a representação esquemática da Figura 6.

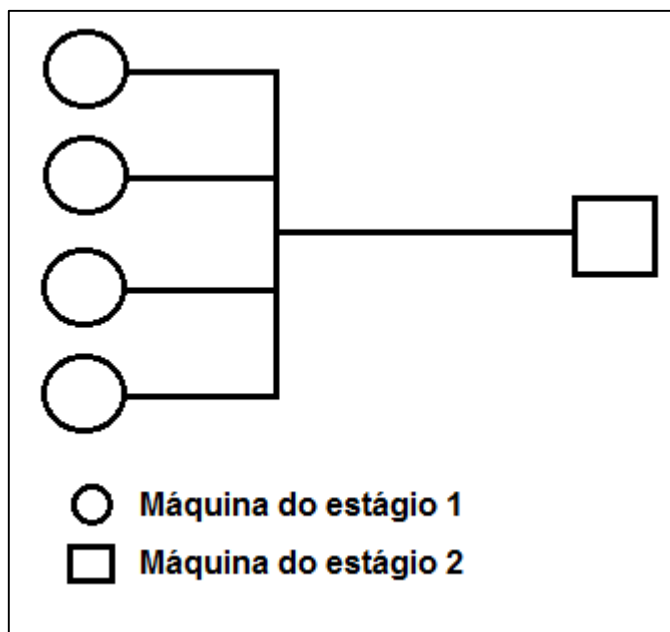


Figura 6: Representação esquemática do modelo de Ferreira, Morabito e Rangel (2008)

Através dessa simplificação, esses autores conseguem resultados satisfatórios para seu modelo em tempo computacional hábil. Para isso, Ferreira, Morabito e Rangel (2008) usam a heurística *relax-and-fix*, além de ajustar parâmetros do *software* de resolução (CPLEX) relacionados aos possíveis cortes (*Gomory's cutting*, por exemplo)⁴ para obter soluções melhores, indicando a importância desses parâmetros para os experimentos computacionais.

Ferreira, Morabito e Rangel (2009) também fazem uma simplificação do modelo de Toledo (2007), porém menos restritiva que a do artigo desses mesmo autores de 2008. Eles consideram várias máquinas no segundo estágio, porém cada uma com uma máquina de primeiro estágio exclusiva. Uma representação esquemática é apresentada na Figura 7.

⁴ Termo adotado pelo *software* CPLEX

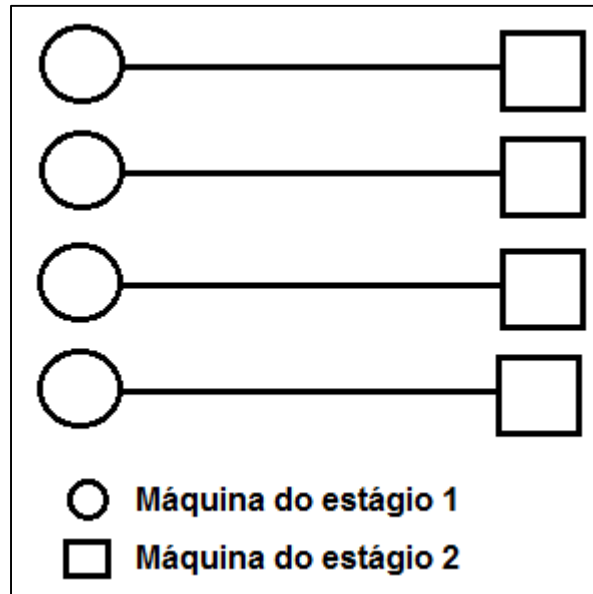


Figura 7: Representação esquemática do modelo de Ferreira, Morabito e Rangel (2009)

Ferreira, Morabito e Rangel (2009) também indicam que uma solução exata não é possível em tempo computacional hábil, propondo então duas heurísticas. Primeiramente eles usam uma simplificação de que o gargalo está na segunda etapa e a resolvem de forma independente, resolvendo a primeira etapa depois baseando-se na segunda etapa já resolvida. Além disso, assim como Ferreira, Morabito e Rangel (2008) eles também se utilizam de uma heurística *relax-and-fix*, ressaltando que a obtenção de boas soluções depende do agrupamento que se faz das variáveis binárias que serão resolvidas, relaxadas e fixadas a cada passagem do mecanismo de solução.

Partindo de modelo de Ferreira, Morabito e Rangel (2009), Ferreira, Almada-Lobo e Morabito (2013) propõe duas formulações monoestágio para o problema de sequenciamento e dimensionamento da produção em dois estágios. Uma representação esquemática desse modelo é apresentada na Figura 8.

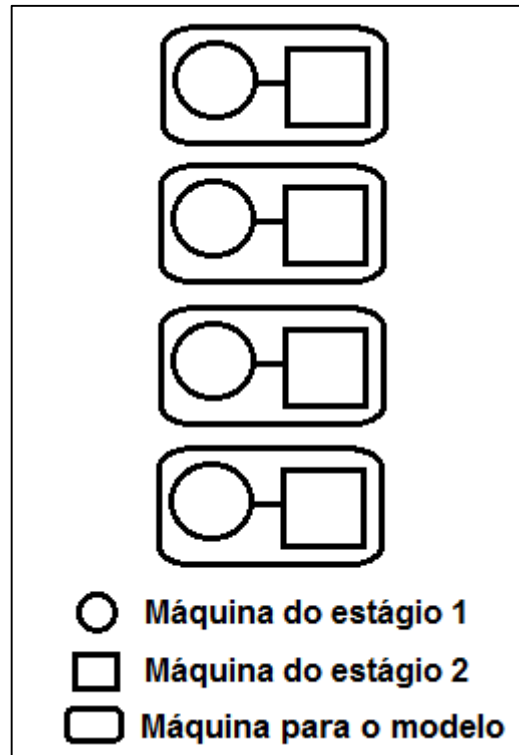


Figura 8: Representação esquemática do modelo de Ferreira, Almada-Lobo e Morabito (2013)

Essas formulações são resolvidas através do mecanismo *relax-and-fix* e obtém soluções comparáveis as de Ferreira, Morabito e Rangel (2009), porém, assim como esses autores obtém-se um *gap* de otimalidade alto (96%), indicando que o modelo não consegue achar um bom limitante inferior em tempo hábil, mesmo conseguindo uma solução factível melhor que a da empresa. Vale notar que tanto Ferreira, Morabito e Rangel (2009) e Ferreira, Almada-Lobo e Morabito (2013) tratam do mesmo problema, apenas modelado de formas diferentes.

3.5 Características específicas do problema-tema

Observando as representações esquemáticas da literatura apresentadas na seção anterior e, comparando com a representação esquemática do problema da empresa, percebe-se que ele não se encaixa diretamente em nenhum deles, conforme mostrado na Figura 9.

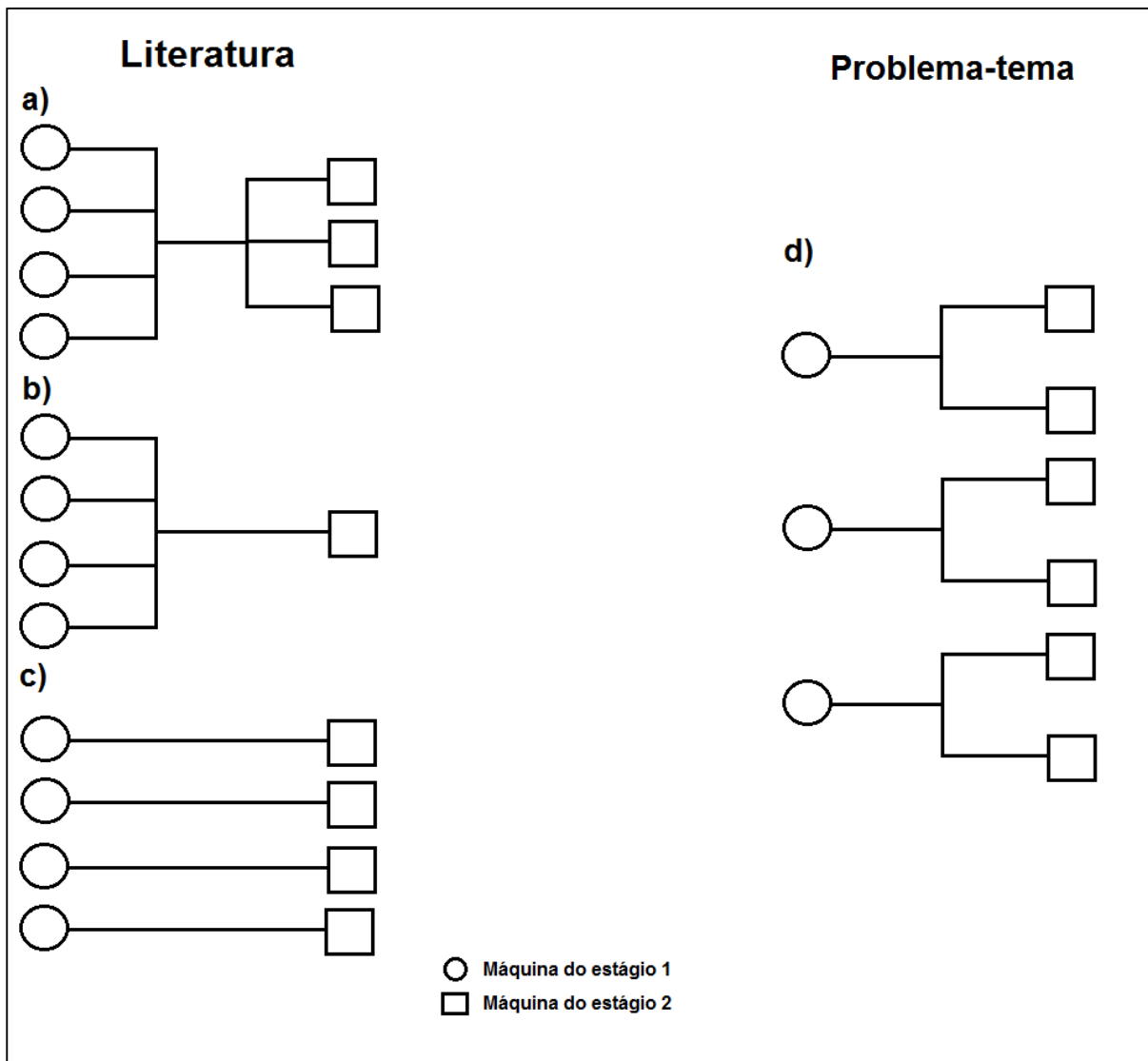


Figura 9: Comparação do problema-tema com modelos da literatura: a) Modelo de Toledo (2007); b) Modelo de Ferreira, Morabito e Rangel (2008); c) Modelo de Ferreira, Morabito e Rangel (2009) e Ferreira, Almada-Lobo e Morabito (2013); d) Problema-tema desse trabalho

O modelo da Figura 9-a é mais genérico do que os outros dois, porém não trata da exclusividade entre duas embaladoras para cada produtora. Além disso, esse modelo de Toledo (2007) se provou inaplicável na prática, tendo inclusive servido de motivação para as simplificações dos modelos Figura 9-b e Figura 9-c.

O modelo da Figura 9-b de Ferreira, Morabito e Rangel (2008), por outro lado, é capaz de ser resolvido e representa bem a relação exclusiva entre máquinas dos dois estágios, porém no sentido contrário do nosso problema, onde há mais de uma máquina do estágio 2 para cada máquina do estágio 1, e não o oposto. Vale notar também que nesse modelo várias máquinas

do estágio 1 podem servir uma máquina do estágio 2, mas apenas uma por vez, enquanto no problema desse trabalho as duas embaladoras são servidas ao mesmo tempo.

O modelo da Figura 9-c, que representa conceitualmente tanto Ferreira, Morabito e Rangel (2009) quanto Ferreira, Almada-Lobo e Morabito (2013), é uma simplificação ainda maior, havendo uma relação de um para um entre máquinas do estágio 1 com máquinas do estágio 2. Esse modelo tem resolução mais fácil, mas se afasta ainda mais do problema-tema desse trabalho.

Conclui-se então que, até onde se sabe, o problema específico desse trabalho é inédito na literatura. Tendo em vista as dificuldades práticas de modelos muito complexos no contexto de um trabalho que visa resolver um problema prático, buscar-se-á a aplicação de uma estratégia de complexidade crescente, partindo do modelo mais simples dos que foram apresentados, ou seja, a formulação monoestágio de Ferreira, Almada-Lobo e Morabito (2013) para que esta consiga lidar com duas máquinas no segundo estágio, produzindo portanto dois produtos ao mesmo tempo que compartilham algumas características (tamanho e tipo da fralda) mas diferem em outra (tamanho do pacote).

Desse modo, a abordagem de modelagem desse trabalho é mostrada na Figura 10.

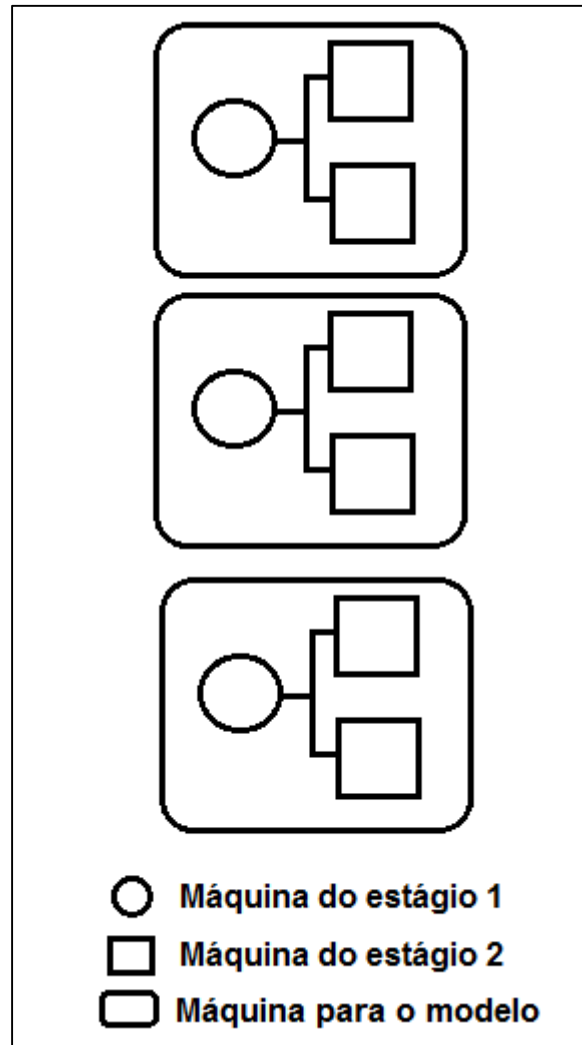


Figura 10: Abordagem de modelagem proposta

4 MODELO PROPOSTO

Nesse capítulo é apresentado o modelo matemático de programação linear inteira mista desenvolvido para representar o problema da empresa. Primeiramente é introduzido e apresentado o modelo completo, incluindo índices, variáveis, constantes, parâmetros, função objetivo e restrições. Em seguida são fornecidas explicações para cada uma das restrições do modelo, auxiliando assim sua compreensão. É apresentado então um problema piloto que permite entender como através do modelo se toma decisões de programação da produção e o comportamento geral das variáveis nesses casos. Por fim, é discutida a validade do modelo à luz do problema piloto, para que então ele seja aplicado ao problema real nas etapas seguintes.

4.1 Modelo

Para a modelagem foi adotada como unidade de produção a perspectiva de uma célula produtiva, que contém uma produtora e duas embaladoras. Essa perspectiva foi adotada pois facilita a comparação com o modelo de máquinas em paralelo, sendo que a relação entre uma produtora e suas embaladoras é internalizada. Isso é possível já que duas embaladoras servem exclusivamente uma produtora, que por sua vez fornece produtos apenas a essas duas embaladoras. Vale ressaltar que a existência de relações internas entre máquinas numa célula é o que diferencia esse trabalho de outros na literatura, conforme mostrado na revisão bibliográfica. Sendo assim, os exemplos serão direcionados para os aspectos do modelo relacionados a isso.

Além disso, decidiu-se por trocar as dimensões tamanho e tipo da fralda, por especificações da fralda, que contém as combinações entre as duas, diminuindo assim a quantidade de dimensões das variáveis e parâmetros. Em situações que nem todas as combinações de tipos e tamanhos são possíveis, isso gera também menor número de variáveis ao eliminar variáveis desnecessárias (associadas a produto de demanda zero). Uma fralda especial grande, por exemplo, pode não existir, apesar de existirem fraldas especiais de outros tamanhos ou grandes de outros tipos⁵. Essa escolha de modelagem, dessa forma, elimina uma complexidade

⁵ Foram usados nomes genéricos para evitar identificação da empresa.

desnecessária ao modelo ao mesmo tempo em que o torna mais facilmente entendível e replicável. A relação entre tipos, tamanhos e especificações é mostrada na Tabela 3.

Tabela 3: Tipos, tamanhos e especificações de fralda

Tipo de fralda	Tamanho de fralda	Código da especificação	Índice da especificação
1	M	1_M	1
	L	1_L	2
	XL	1_XL	3
2	M	2_M	4
	L	2_L	5
	XL	2_XL	6
3	M	3_M	7
	L	3_L	8
	XL	3_XL	9
	XX	3_XX	10
4	M	4_M	11
	L	4_L	12
	XL	4_XL	13
	XX	4_XX	14

Deve-se apenas tomar cuidado ao inserir os dados para saber quais transições de especificações são de tipo (mais rápidas) ou de tamanho de fralda (mais lentas). Uma transição entre a especificação um para a dois dura por exemplo 12 horas já que trata-se de uma mudança de uma fralda normal média para grande (mudança de tamanho), mas ser de apenas 6 horas se for uma mudança de uma fralda média normal para especial (mudança de tipo, como entre as especificações 9 e 13).

O modelo utiliza-se de uma divisão do tempo do tipo *small bucket* para efetivamente definir a programação. Isso significa que o período de programação é dividido em intervalos (subperíodos) menores de tempo. Para esse modelo são utilizados subperíodos de tempo de tamanho variável. Em cada intervalo, apenas uma ação pode ser tomada, podendo produzir ou realizar *setup*. Em caso de produção, pode-se fazer apenas uma especificação de fralda, com um tamanho de pacote em cada embaladora. Em caso de *setup*, pode-se mudar de uma combinação entre uma especificação para a produtora e um tamanho de pacote para cada embaladora, para outra combinação de uma especificação e dois tamanhos de pacote.

Uma restrição que naturalmente emerge dessa escolha de modelagem, é que só é possível produzir se no intervalo anterior foi feito um *setup* para essa combinação. Dois *setups* consecutivos não fazem sentido já que os tempos de *setup* obedecem a desigualdade triangular, ou seja, para qualquer mudança de uma combinação entre especificação e dois tamanhos de pacote para outra combinação desse tipo, o tempo de *setup* direto é sempre menor do que fazê-lo em dois estágios. Como consequência disso, alterna-se entre um subperíodo de produção e um de *setup* para cada célula produtiva. Vale notar que as células começam já preparadas para produzir uma combinação, porém permite-se mudar esse *setup* logo no primeiro subperíodo, seguindo então o padrão de alternância discutido.

Para entender como isso leva ao dimensionamento e sequenciamento dos lotes, mostra-se na Figura 11 um exemplo. Nesse exemplo têm-se um tempo total disponível de 24 horas e três subperíodos. Nele nota-se a característica variável do tamanho dos subperíodos. Isso válido não só para cada célula, mas entre células diferentes, já que o mesmo subperíodo pode ter duração diferente, como o subperíodo 1 do exemplo que dura 10 horas para a célula 1 e 12 horas para a célula 2. Outra característica interessante é que subperíodos podem ter duração zero, como o subperíodo 3 da célula 2. Não é necessário também que o tempo total seja utilizado.

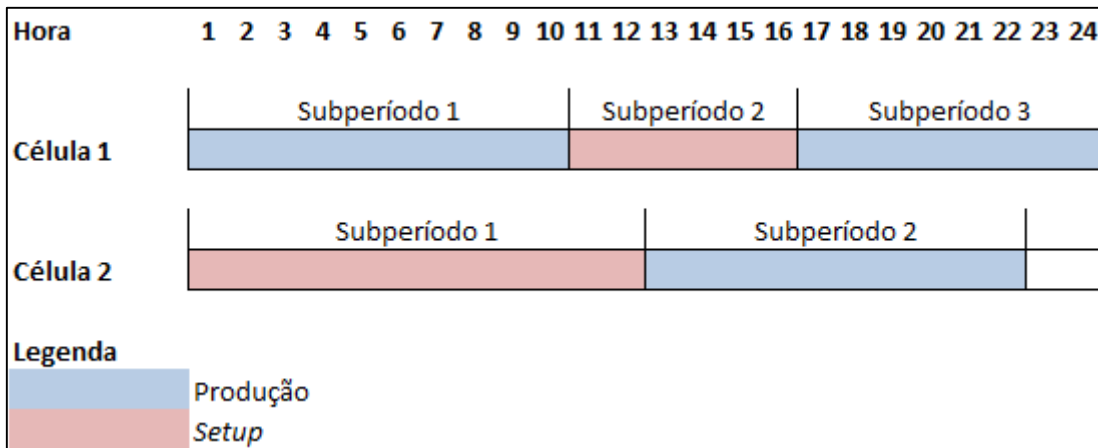


Figura 11: Exemplo de funcionamento dos subperíodos.

No modelo, cada subperíodo é associado a variáveis binárias que indicam se há determinada ação (produção ou *setup*) naquele subperíodo. É atribuída uma variável binária para cada possível combinação entre especificação e dois tamanhos de pacote para a produção. O mesmo é feito para o caso de *setup*, onde há a mudança da combinação atual de especificação e pacotes

para a outra combinação para a qual está sendo feita a preparação. Com essas variáveis já é possível saber a sequência na qual os lotes serão produzidos meramente observando quais binárias são verdadeiras para cada subperíodo.

Já o dimensionamento dos lotes é definido na medida em que se define a duração de cada subperíodo. Essa duração é então multiplicada pela taxa de produção para aquela determinada combinação para a célula produtiva em questão. Desse modo, o modelo consegue atingir o resultado desejado de dimensionar e sequenciar a produção de fraldas.

Vale notar que as simplificações apresentadas (perspectiva da célula produtiva, uso da especificação e não tipo e tamanho da fralda e divisão do tempo) auxiliam a modelagem sem alterar de forma alguma o problema que se deseja resolver, ou seja, sem que haja perda da qualidade da solução.

Por fim, para a função objetivo foi decidido minimizar o tempo total gasto em produção em todas as células. Isso se justifica pois assim permite-se minimizar o tempo gasto com *setups* ao mesmo tempo que é possível um uso melhor dos recursos produtivos. Pode-se assim diminuir a necessidade de mão-de-obra e liberar recursos para outras áreas produtivas, já que algumas máquinas poderiam ser aproveitadas para outros produtos ou para pedidos inesperados. Busca-se então com essa escolha uma alternativa de produção que use da melhor forma possível a capacidade instalada para minimizar o tempo gasto produzindo cada produto (incluindo *setups*).

Sendo assim, apresenta-se a seguir o modelo proposto:

Índices

$p \in [1; C]$	Célula produtiva
$f \in [1; F]$	Fração de tempo
$e, e' \in [1; E]$	Especificação da fralda
$c, c', c_1, c_2, c_1', c_2' \in [1; P]$	Tamanho do pacote

Convencionar-se-á que os índices c, c' serão usados quando a constante independe da embaladora (como no caso da demanda) enquanto c_1, c_1' serão usados para a embaladora 1 e

c_2, c_2' para a 2. Isso é feito para facilitar a compreensão do modelo, sendo que todos os índices representam tamanhos de pacote.

Variáveis

$t_{p,f,e,c_1,c_2} \in R$	Tempo gasto na célula produtiva p na fração de tempo f produzindo a fralda com especificação e nos tamanhos de pacote c_1 e c_2
$s_{p,f,e,c_1,c_2,e',c_1',c_2'} \in B$	Indica se a célula produtiva p na fração de tempo f realizou <i>setup</i> de e, c_1, c_2 para e', c_1', c_2'
$d_{p,f,e,c_1,c_2} \in B$	Indica se a célula produtiva p está preparada em f para produzir a fralda com especificações e nos tamanhos de pacote c_1 e c_2

Constantes

$C_{p,e,c,c'}$	Quantidade que a célula produtiva p produz do pacote de especificação e e tamanho de pacote c por unidade de tempo em qualquer das embaladoras sendo c' o outro pacote sendo embalado simultaneamente na outra embaladora
$D_{e,c}$	Demanda do mês para o pacote e, c (lembrando que a demanda para o modelo é a demanda real ajustada pelos estoques, ou a necessidade de produção daquele mês)
$Z_{e,e'}$	Tempo de <i>setup</i> para mudar da especificação e para e' . Note que se $e = e'$ (mesmo tipo e tamanho de fralda) trata-se de uma transição de tamanho de pacotes

T_p	Tempo total disponível naquele mês para a célula produtiva p , considerando programações feitas pelo programador antes de usar o modelo
I_{p,e,c_1,c_2}	Setup inicial da célula no início do período. 1 se a célula p estiver preparada para e, c_1, c_2 e 0 caso contrário

Parâmetros

F	Número máximo de períodos no mês
C	Número de células produtivas
E	Número de especificações
P	Número de tamanhos de pacote
M	Número suficientemente grande

Função objetivo

$$\text{Min} \sum_{p=1}^C \sum_{f=1}^F \sum_{e=1}^E \sum_{c_1=1}^P \sum_{c_2=1}^P \left(t_{p,f,e,c_1,c_2} + \sum_{e'=1}^E \sum_{c_1'=1}^P \sum_{c_2'=1}^P (s_{p,f,e,c_1,c_2,e',c_1',c_2'} \cdot Z_{e,e'}) \right) \quad (1)$$

Sujeito a:

$$\sum_{p=1}^C \sum_{f=1}^F \sum_{c'=1}^P ((t_{p,f,e,c,c'} + t_{p,f,e,c',c}) \cdot C_{p,e,c,c'}) \geq D_{e,c} \quad \forall e, c \quad (2)$$

$$t_{p,f,e,c_1,c_2} \leq d_{p,f,e,c_1,c_2} \cdot M \quad \forall p, f, e, c_1, c_2 \quad (3)$$

$$\sum_{e=1}^E \sum_{c_1=1}^P \sum_{c_2=1}^P \left(d_{p,f,e,c_1,c_2} + \sum_{e'=1}^E \sum_{c_1'=1}^P \sum_{c_2'=1}^P s_{p,f,e,c_1,c_2,e',c_1',c_2'} \right) \leq 1 \quad \forall p, f \quad (4)$$

$$d_{p,f,e,c_1,c_2} \leq \sum_{e'=1}^E \sum_{c_1'=1}^P \sum_{c_2'=1}^P s_{p,f-1,e',c_1',c_2',e,c_1,c_2} \quad \forall p, f > 1, \quad e, c_1, c_2 \quad (5)$$

$$d_{p,1,e,c_1,c_2} + \sum_{e'=1}^E \sum_{c_1'=1}^P \sum_{c_2'=1}^P s_{p,1,e,c_1,c_2,e',c_1',c_2'} \leq l_{p,e,c_1,c_2}, \quad \forall p, e, c_1, c_2 \quad (6)$$

$$\sum_{e'=1}^E \sum_{c_1'=1}^P \sum_{c_2'=1}^P s_{p,f,e,c_1,c_2,e',c_1',c_2'} \leq d_{p,f-1,e,c_1,c_2} \quad \forall p, f > 1, \quad e, c_1, c_2 \quad (7)$$

$$\sum_{f=1}^F \sum_{e=1}^E \sum_{c_1=1}^P \sum_{c_2=1}^P \left(t_{p,f,e,c_1,c_2} + \sum_{e'=1}^E \sum_{c_1'=1}^P \sum_{c_2'=1}^P s_{p,f,e,c_1,c_2,e',c_1',c_2'} \cdot Z_{e,e'} \right) \leq T_p \quad \forall p \quad (8)$$

$$t_{p,f,e,b,c_1,c_2} \in R_+ \quad \forall p, f, e, b, c_1, c_2 \quad (9)$$

$$s_{p,f,e',c_1',c_2',e,c_1,c_2} \in B \quad \forall p, f, e, c_1', c_2', e', c_1, c_2 \quad (10)$$

$$d_{p,f,e,c_1,c_2} \in B \quad \forall p, f, e, c_1, c_2 \quad (11)$$

4.2 Descrição do modelo

Nessa seção detalha-se cada uma das equações apresentadas na seção anterior.

A função objetivo (1) minimiza a somatória do tempo total utilizado por todas as células, ou seja, o tempo despendido produzindo e em *setups* naquele mês. Utiliza-se assim uma lógica de melhor uso da capacidade instalada conforme explicado na seção anterior.

O conjunto de restrições (2) indica se a demanda é atendida, considerando todas as células produtivas p e frações de tempo f . Note que a mesma capacidade é utilizada para ambos os tempos, já que essa independe de qual embaladora está fazendo cada tamanho de pacote. Para

a capacidade, o primeiro índice de pacote c dita a qual pacote essa capacidade se refere, enquanto o segundo índice indica o outro pacote c' que, produzido na outra embaladora, leva a essa capacidade para o pacote c . Já para o tempo de produção, a ordem dos índices refere-se a cada uma das embaladoras individualmente, isto é, c_1 refere-se à embaladora 1 e c_2 à embaladora 2. Por isso, deve-se levar em conta o tempo produzindo um determinado pacote c_1 em ambas as embaladoras (por isso os dois tempos) com a capacidade referente àquele pacote quando o outro pacote é c_2 (ou seja, a capacidade é a mesma independente de qual embaladora produz qual pacote).

Se uma célula produz os pacotes grande e médio simultaneamente, de especificação 7 na célula produtiva 2, temos por exemplo $e = 7$, $p = 2$, $c_1 = 4$ e $c_2 = 1$ (especificação, célula, pacotes grande e médio respectivamente). Se $C_{2,7,1,4} = 5$ e $C_{2,7,4,1} = 3$ e a produção ocorre por 10 unidades de tempo, ou seja, $t_{2,f,7,4,1} = 10$, produz-se 30 pacotes grandes na embaladora 1 ($c_1 = 4$) e 50 pacotes médios na embaladora 2 ($c_2 = 1$). Desse modo, enquanto os índices de t indicam qual produto é feito em qual embaladora, os índices de C indicam a quem se refere a capacidade (primeiro c) e quem é o outro produto sendo produzido (segundo c). Por isso, utiliza-se $C_{2,7,1,4}$ mesmo que o pacote médio (1) esteja na embaladora 2.

O conjunto de restrições (3) indica que só pode haver produção no intervalo de tempo f se a máquina está preparada para os produtos desejados.

O conjunto de restrições (4) indica que só pode ser feito no intervalo f ou um produto com especificação e e no máximo dois tamanhos c_1 e c_2 ou um *setup*.

O conjunto de restrições (5) obriga que a máquina só possa estar preparada para produzir se no intervalo anterior já foi feito o *setup* necessário.

O conjunto de restrições (6) indica que a célula p está preparada inicialmente para e, c_1, c_2 . Essas restrições também permitem para a célula p um *setup* de e, c_1, c_2 para e', c_1', c_2' apenas no primeiro período.

O conjunto de restrições (7) permite para a célula p um *setup* de e, c_1, c_2 para e', c_1', c_2' apenas se no período anterior foi produzido e, c_1, c_2 . Para $f = 1$ o conjunto de restrições (6) é usado.

O conjunto de restrições (8) afirma que o tempo de produção em cada célula produtiva p deve ser inferior ao tempo disponível para essa célula produtiva.

Os conjuntos de restrições (9), (10) e (11) indicam o domínio das variáveis.

4.3 Testes preliminares

Para validar o modelo, após uma primeira versão, foram realizados diversos testes com versões simplificadas do modelo e um conjunto reduzido de dados. Em cada iteração foi verificado se o modelo apresentava uma solução aderente à realidade e, caso contrário, aprimorou-se o modelo. Todos esses testes foram realizados com o *software* CPLEX 12.6.

Na primeira iteração foi resolvido um problema com uma célula, um subperíodo e demanda para só um produto em um tipo de pacote. Em seguida testou-se um tipo de produto e dois tipos de pacote (exigindo que a célula estivesse com embaladoras preparadas para pacotes diferentes). Testou-se então três subperíodos, teste esse que ajudou a amadurecer a modelagem em relação à preparação inicial das máquinas antes do primeiro subperíodo. Por fim, foi realizado o teste com duas células produtivas e cinco subperíodos, usando portanto a modelagem completa.

Para fins didáticos de explicação do modelo, será apresentado em detalhe nessa seção o último desses testes, já com o modelo completo, porém com um tamanho menor que o real. Foram utilizadas duas células produtivas, com duas especificações de produto, dois tamanhos de pacote e cinco subperíodos. Para efeitos desse teste, foi considerada uma demanda de 100 pacotes para todos os produtos e todos os tamanhos de pacote conforme Tabela 4.

Tabela 4: Demanda (exemplo)

$D_{e,c}$	e	c	D
	1	1	100
		2	100
	2	1	100
		2	100

Além disso, foi considerada uma célula “lenta” com tempo disponível de 100 unidades de tempo e uma célula “rápida”, porém com tempo disponível de apenas 50 unidades de tempo. Esses dados são apresentados na Tabela 5.

Tabela 5: Tempos disponíveis (exemplo)

T_p	p	T
	1	100
	2	50

A célula “lenta” ($p = 1$) tem metade da capacidade da célula rápida ($p = 2$) em todos os casos. Para a especificação 1 a célula performa melhor com as duas embaladoras usando o mesmo pacote (C é maior quando $c = c'$), já para a especificação 2 a situação se inverte, conforme mostra-se na Tabela 6.

Tabela 6: Capacidade de produção por unidade de tempo (exemplo)

$C_{p,e,c,c'}$	p	e	c	c'	C
	1	1	1	1	2
				2	1
			2	1	1
				2	2
		2	1	1	1
				2	2
			2	1	2
				2	1
	2	1	1	1	4
				2	2
			2	1	2
				2	4
		2	1	1	2
				2	4
			2	1	4
				2	2

Foi considerado um tempo de *setup* de 12 para mudanças de especificações de produto e de 2 para as mudanças de tamanho de pacote ($e = e'$), conforme a Tabela 7.

Tabela 7: Tempo de *setup* (exemplo)

$Z_{e,e'}$	e	e'	Z
	1	1	2
		2	12
	2	1	12
		2	2

Por fim, foi considerado que a célula 1 estava preparada inicialmente para a especificação 2 e pacote 2 nas duas embaladoras, enquanto a célula estava preparada para especificação 1 e pacote 1 na embaladora 1 e pacote 2 na embaladora 2. Esses dados são apresentados na Tabela 8. As combinações omitidas têm valor zero.

Tabela 8: *Setup* inicial (exemplo)

I_{p,e,c_1,c_2}	p	e	c_1	c_2	I
	1	2	2	2	1
	2	1	1	2	1

Ao executar o modelo com esses dados, obteve-se um tempo total (função objetivo) de 81 sendo este dividido por célula conforme a Figura 12.

Frações de tempo					
Célula	1	2	3	4	5
1	2	50	0	0	0
2	2	12,5	2	12,5	0
Legenda					
t	Produção por t unidades de tempo				
i	Setup de duração i				

Figura 12: Duração de cada fração de tempo (exemplo)

Pode-se verificar em detalhe agora na Figura 13, o que cada célula produziu em cada fração de tempo para efetivamente entender o sequenciamento. Uma coluna extra com o *setup* inicial será adicionada nesta figura (em relação à Figura 12) para auxiliar o entendimento. Onde houve *setup* será colocado 1 nas linhas que poderão produzir no próxima fração de tempo, de modo

que só pode haver produção em uma linha da tabela se na coluna anterior está indicado que houve *setup* (1).

Célula	Embaladora	Especificação	Pacote	Frações de tempo					
				0	1	2	3	4	5
1	1	1	1				0	0	0
			2				0	0	0
		2	1				0	0	0
			2	1	1	100	0	0	0
	2	1	1				0	0	0
			2				0	0	0
		2	1		1	100	0	0	0
			2	1			0	0	0
2	1	1	1	1			1	50	0
			2		1	50			0
		2	1						0
			2						0
	2	1	1				1	50	0
			2	1	1	50			0
		2	1						0
			2						0

Legenda

K	Produção de K unidades
1	Setup ocorreu
1	Setup inicial da máquina

Figura 13: Produção em cada fração de tempo

Nesse exemplo, fica claro que o modelo consegue avaliar a melhor opção entre produzir o mesmo pacote nas duas embaladoras ou produzir pacotes diferentes. Para a especificação 1, onde produzir um pacote por vez usando as duas embaladoras é melhor, o modelo mudou o *setup* para essa condição (célula 2). Já para a especificação 2, onde dividir é mais produtivo, a célula 1 também fez o *setup* para atingir essa situação. Isso demonstra inclusive que o modelo permite um *setup* na primeira fração de tempo caso o *setup* inicial da célula não seja satisfatório. Isso acontece para as duas células nesse exemplo, mas deveria ser menos frequente em um caso real, no entanto, foi usado esse exemplo para testar o modelo nessa situação mais complexa.

Desse modo, percebe-se que o modelo é capaz de escolher adequadamente as situações de *setup* que levam ao menor tempo total. Esse exemplo ilustra que ele consegue lidar bem com as escolhas relacionadas a existirem duas embaladoras com capacidades dependentes da combinação de produtos entre elas, que é a principal característica desse modelo, validando assim sua aplicação.

4.4 Validação

Tendo realizado testes preliminares com complexidade crescente até chegar-se no modelo completo, foi possível testar diversos casos simples de aplicação até que o modelo obtivesse a solução considerada adequada (facilmente identificável para casos pequenos). Sendo assim, o modelo evoluiu até que, nessa versão final, ele foi considerado válido por sempre oferecer a solução ótima esperada. O último desses testes é o apresentado na seção anterior e foi validado com a empresa, que considerou que o problema real estava adequadamente representado.

5 COLETA DE DADOS

Para o desenvolvimento do trabalho foi fornecido pela empresa o conjunto de dados referentes a aproximadamente um mês de produção. Esse conjunto de dados foi obtido na forma de uma planilha, acompanhada por diversas explicações dos responsáveis pelo planejamento da produção, dentre eles o gerente de planejamento da produção. Essa planilha continha também o plano de produção realizado pela própria empresa para que ele pudesse ser comparado com os resultados desse trabalho.

A pedido da empresa, a planilha real não será mostrada, porém apresentaremos a seguir os dados obtidos a partir dela que serão usados como entradas do modelo. A própria empresa alterou os dados para que ficassem coerentes, porém diferentes dos reais, também como medida de proteção de suas informações.

Vale notar que das seis máquinas utilizadas na produção, uma delas é específica para alguns produtos (i.e. nenhuma outra máquina produz os mesmos produtos que ela e vice versa). Sendo assim, e considerando que o planejamento dessa máquina é trivial (para os planejadores) e independente, foi decidido, em conjunto com a empresa, excluí-la do escopo desse trabalho.

A primeira informação relevante que pôde-se extrair da planilha são as combinações possíveis entre tipos e tamanhos de fraldas, que gerarão as especificações para o modelo. Vale lembrar que essas duas informações foram agrupadas no modelo para facilitar sua expressão em equações e sem que houvesse nenhum prejuízo para a representação da realidade. O único aspecto do modelo afetado pela diferenciação entre tipo e tamanho de fralda é o tempo de *setup* ($Z_{e,e'}$), que é de 6 horas para mudanças de tipo de fralda e 12 horas para mudanças de tamanho de fralda. Sendo assim, a partir de agora só usaremos as especificações (e), que já foram apresentadas anteriormente na Tabela 3.

Para uma mudança da especificação 1 para a 2, por exemplo, temos uma mudança de tamanho de fralda, que dura 12 horas. Já da especificação 7 para a 11, muda-se apenas o tipo de fralda, ou seja, um *setup* de 6 horas. Em casos onde há as duas mudanças, vale o tempo maior (i.e. 12 horas), como é o caso da 8 para a 14. Por fim, *setups* de tamanho de pacote (onde não muda-se a especificação), o tempo é de 4 horas, como por exemplo uma mudança da especificação 9 para ela mesma. A tabela completa está no Anexo 1.

Em seguida, foram relacionadas quais especificações de produtos (e) podem ser feitos em quais produtoras (p). Essas combinações são aquelas que, no modelo, tem taxa de produção maior que zero. Dessa forma, são apresentadas na Tabela 9 abaixo as máquinas e as especificações possíveis. Vale notar que nem todas as máquinas fazem um determinado produto, porém mais de uma máquina pode fazer o mesmo produto.

Tabela 9: Especificações possíveis por máquina

p	e	p	e	p	e
1	2	3	2	4	1
	3		3		1
	2		2		4
	3		3		7
	2		2		7
	3		6		7
	2		9		11
	3		9		11
	2		9		11
	3		10		2
2	2	4	10	5	2
	1		10		2
	2		13		2
	1		13		2
	2		13		2
	1		14		5
	2		14		8
	1		14		8
	2		1		8
	1		1		12
3	2	4	2		12
	3		1		12
	2		1		

No modelo em si, essa informação está embutida na taxa de produção, que é zero para as combinações não possíveis. Apesar de, segundo o engenheiro responsável, taxas diferentes deveriam ser aplicadas para combinações diferentes de pacotes sendo embalados, na prática hoje simplifica-se para que ou uma combinação é possível, ou ela é impossível. Se ela for possível, a capacidade independe do outro produto sendo embalado. Como esse é o modelo que

a própria empresa usa, para manter as propostas comparáveis, optou-se por simplificar o modelo apresentado de modo que a taxa de produção dependa apenas do produto em questão. No entanto, deixa-se como sugestão de trabalhos futuros uma coleta mais precisa dos dados, o que não foi permitido nesse caso pela empresa. Dessa forma, para as combinações possíveis, apresenta-se as taxas de produção (independentes do segundo produto sendo embalado) no Anexo 2.

Coletou-se também a demanda (D) de cada pacote (c) de cada especificação (e) para as combinações possíveis. Vale notar que a demanda da produção é dada pela demanda da empresa corrigida pela política de estoques. Os dados obtidos correspondem a demanda do setor de produção, ou seja, já corrigidos. Os dados são apresentados na Tabela 10 a seguir. Os valores são fracionários pois são obtidos do cálculo descrito acima. Vale notar que essa abordagem de não arredondar para um número inteiro também é usada pela empresa na sua programação, sendo que os operadores da produção na prática arredondam para cima para um número inteiro de unidades.

Tabela 10: Demanda por especificação e tamanho de pacote

e	c	D
1	1	6380,02
	2	2596,78
	3	11531,47
	4	6320,59
	6	2795,51
2	1	8014,18
	2	1981,38
	3	11382,98
	4	9831,56
	5	2771,87
	6	5324,85
3	1	1795,21
	2	1316,49
	3	6645,59
	4	5150,29
	6	2955,08
4	6	0,00
5	6	653,50

e	c	D
6	6	1606,94
7	3	310,28
	4	1303,19
	5	1099,29
8	3	2964,54
	4	2584,22
	5	3510,64
9	3	0,00
	4	1031,10
	5	1031,10
10	3	502,36
	4	2086,74
	5	2550,46
11	3	310,28
	4	961,88
	5	1418,44
12	3	1092,20
	4	1072,70
	5	2287,23
13	3	883,80
	4	491,00
	5	1374,80
14	3	452,13
	4	2364,98
	5	1947,63

O *setup* inicial foi obtido diretamente da planilha, e é apresentado na Tabela 11. Os valores não apresentados são zero (0). Cada célula produtiva pode estar preparada para uma especificação (e) e um tamanho de pacote para cada embaladora (c_1 e c_2), podendo estes ser iguais. A última coluna apresenta a variável I correspondente.

Tabela 11: Setup inicial de cada célula

p	e	c_1	c_2	I
1	3	2	4	1
2	1	3	4	1

p	e	c_1	c_2	I
3	2	1	3	1
4	1	3	4	1
5	2	2	6	1

Finalmente, foram oferecidos dados de 29 dias para o trabalho (o plano foi feito no dia dois do mês, de forma que os dados dos dois primeiros dias não foram planejados a priori e por isso excluídos da comparação). No entanto, o tempo efetivo disponível deve ser descontado de planejamentos forçados (por pedidos atrasados ou matéria-prima não existente), pausas para manutenção planejada e dias em que as máquinas foram usadas para testes de inovação ou engenharia. Sendo assim, o número de dias efetivamente disponível (T_p) por célula produtiva é mostrado na Tabela 12.

Tabela 12: Tempos disponíveis

p	T
1	29
2	28
3	27
4	27
5	29

Usando então esses dados, pôde-se usar o modelo para obter uma nova solução para ser comparada com a solução atual, conforme será mostrado nos capítulos subsequentes.

6 RESULTADOS OBTIDOS

Nesse capítulo apresentar-se-á como o problema descrito foi resolvido, considerando-se para isso métodos de resolução exata e heurísticas.

6.1 Proposta inicial de resolução

Inicialmente procurou-se resolver o modelo proposto de programação linear inteira mista de forma exata. Uma solução exata tem como principal vantagem a garantia de que o ótimo global foi atingido e para isso aplica-se métodos para os quais já foi demonstrado matematicamente que se obtém sempre o ótimo, como por exemplo, o método *branch and bound* descrito em Nemhauser e Wolsey (1988).

Para resolver esse problema de forma exata, utilizou-se o *software* CPLEX. Este *software* foi escolhido por estar disponível na empresa e o responsável pela área de planejamento da produção saber usá-lo, de forma que o modelo de resolução poderia ser facilmente replicado pela empresa posteriormente. Além disso, é um *software* usado comumente na literatura, vide Kawamura (2011), Toso e Morabito (2005) e Quadri e Kuhn (2009) por exemplo.

Para usar o CPLEX o modelo foi resolvido na interface IBM OPL IDE versão 12.6. O código do modelo implementado pode ser visto no Anexo 3. Esse *software* tem como uma de suas principais vantagens que se pode separar o modelo em si dos dados, sendo fácil a mudança do conjunto de dados utilizados.

O modelo foi colocado para rodar em um computador com 16 GB de memória RAM, rodando em um Windows 64-bit. Após aproximadamente 45 minutos de pré-processamento, o *software* indicou que toda a memória RAM do computador havia sido consumida, e que dessa forma a execução foi finalizada sem resultados.

Levando em conta que todas as especificações dos computadores da empresa são iguais ou inferiores a essas, percebeu-se que alguma simplificação seria necessária. Para melhor entender as limitações, testes adicionais foram realizados variando-se a dimensão do problema analisado. Os resultados obtidos são apresentados na Tabela 13.

Tabela 13: Comparação do tempo de resolução exata em função das entradas

Dimensões	Problema 1	Problema 2	Problema 3	Problema Real
Células produtivas	2	2	4	5
Especificações	2	4	4	14
Tamanhos de pacote	2	4	4	6
Subperíodos	5	10	30	30
Tempo	0,37 s	55,71 s	Excede capacidade de processamento	Excede capacidade de processamento

Através dessa tabela observa-se que o *software* encontra a solução ótima do Problema 1, com apenas duas especificações de fralda, dois tamanhos de pacote, duas células produtivas e 5 subperíodos em 0,37s. Note que quando dobramos algumas dimensões, ou seja Problema 2, e executamos o modelo com 10 subperíodos, 4 tamanhos de pacote e 4 especificações, o tempo fica em 55,71 s, mostrando que a relação entre o tempo de execução e o tamanho do problema não é linear.

Aumentando-se também o número de células produtivas para 4 e ajustando o número de subperíodos para o real (30) já não se obtém resposta, sendo que essa instância ainda é muito menor que o problema real.

Considerando os tempos apresentados, fica claro que uma resolução exata desse modelo não é viável. Conforme foi discutido na revisão bibliográfica, é bastante comum nesse tipo de problema de programação linear inteira mista que o número de variáveis binárias force a resolução de problemas reais através de métodos heurísticos.

Isso ocorre pois os problemas de *GLSP* são considerados NP-difíceis (*NP-Hard*) o que significa que seu tempo de resolução cresce exponencialmente com o aumento da dimensão dos dados de entrada. Isso é comum em problemas de programação linear inteira mista, já que elas normalmente fazem com que o problema tenha natureza combinatória e, portanto, o tempo de resolução não cresce linearmente.

6.2 Nova proposta de resolução do problema

Conforme discutiu-se na revisão bibliográfica, uma heurística bastante comum para problemas de dimensionamento e sequenciamento de lotes de produção é a *relax-and-fix*. Essa heurística

resolve o problema original em diversos estágios obtidos através de cortes do modelo. Em cada estágio resolve-se uma versão *relaxada* do modelo⁶, fixando-se parte da solução para o próximo estágio até percorrer todos os estágios e obter uma solução factível. Para mais detalhes ver, por exemplo, Kawamura (2011).

Para avaliar a possibilidade de se usar esse método tentou-se relaxar todas as variáveis do modelo que eram binárias para reais. Uma solução para o modelo que fosse obtida dessa forma indicaria um limitante inferior para a função objetivo, já que um modelo com menos restrições sempre tem uma solução melhor ou igual a um modelo cujo espaço de soluções é um subconjunto do anterior. Além disso, isso indicaria a viabilidade de uma heurística do tipo *relax-and-fix*.

Ao se fazer isso ainda assim não foi possível obter uma solução devido à exaustão dos recursos computacionais. Levando-se em conta que o computador da empresa disponível para esse tipo de aplicação tem características semelhantes ao utilizado para os testes, usar uma máquina melhor não era uma solução viável.

Tendo em vista as barreiras encontradas, optou-se por buscar aprimorar o próprio modelo. Para compreender a melhor maneira de se fazer isso, fez-se um levantamento das variáveis do modelo, classificando-as em binárias e reais e observando sua quantidade, baseada na cardinalidade dos seus índices. Esse levantamento é apresentado na Tabela 14.

Tabela 14: Número de variáveis do modelo

Conjunto de variáveis	Cálculo do N° variáveis	Total	Tipo
t_{p,f,e,c_1,c_2}	$5 \cdot 30 \cdot 14 \cdot 6 \cdot 6$	75.600	Real
$s_{p,f,e,c_1,c_2,e',c_1',c_2'}$	$5 \cdot 30 \cdot 14 \cdot 6 \cdot 6 \cdot 14 \cdot 6 \cdot 6$	38.102.400	Binária
d_{p,f,e,c_1,c_2}	$5 \cdot 30 \cdot 14 \cdot 6 \cdot 6$	75.600	Binária

⁶ Com as variáveis binárias transformadas em reais.

Analisando a tabela fica claro que as variáveis do tipo s representam mais de 99% das variáveis, além de serem do tipo binária. Esse conjunto de variáveis é extenso pois ele indica a mudança exata de configuração que está ocorrendo em uma célula p no intervalo de tempo f entre uma especificação e dois tamanhos de pacote (e, c_1, c_2) para outra configuração desse tipo (e', c_1', c_2') .

Entretanto, levando em conta que o tempo de *setup* é dependente da sequência apenas em relação às especificações, ou seja, que mudanças de especificação tem tempos diferentes (12 horas para mudanças de tamanho de fralda e 6 horas para mudanças de tipo de fralda), mas mudanças de tamanho de pacote não (são sempre 4 horas), avaliou-se a possibilidade de fazer $s_{p,f,e,c_1,c_2,e',c_1',c_2'}$ não mais depender dos tamanhos de pacote, ou seja, usar apenas e e e' para representar todas as *setups*, tornando essa variável $s_{p,f,e,e'}$.

Ao fazer isso, incorre-se em algumas dificuldades. Os resultados da resolução do modelo não mais indicarão diretamente a sequência exata de *setups* que devem ser feitos. Ou seja, no modelo original a variável $s_{p,f,e,c_1,c_2,e',c_1',c_2'}$ sozinha indicava exatamente o *setup* que estava sendo feito, porém agora será necessário observar as variáveis d_{p,f,e,c_1,c_2} antes e depois daquele *setup* para inferir qual *setup* foi feito, fazendo com que seja necessário um pós-processamento para gerar essa informação.

Ou seja, uma mudança de uma certa especificação e na célula p e intervalo f dos tamanhos de pacote 1 e 2 para os tamanhos 3 e 4 antes era representada por $s_{p,f,e,1,2,e,3,4}$ e agora ficará apenas $s_{p,f,e,e}$. Para saber exatamente qual mudança está sendo feita, é necessário observar que $d_{p,f,e,1,2} > 0$ e $d_{p,f,e,3,4} > 0$.

Além disso, o modelo deve ser adaptado para que ele considere que mudanças de tamanhos de pacote incorrem ainda que a variável de *setup* tenha $e = e'$. Isso pode ser atendido através da alternância entre *setup* e produção, utilizando-se para isso três condições importantes do problema já discutidas no primeiro modelo:

- A todo instante ou se produz ou se faz *setup*, nunca os dois;
- Todo *setup* é precedido de produção;
- Toda produção é precedida de *setup*.

Deve-se notar que essas condições, não pioram a solução do modelo, já que na solução ótima espera-se justamente que só se façam *setups* necessários, não havendo necessidade de dois

setups subsequentes (a não ser no primeiro intervalo de tempo, caso tratado separadamente). Cabe deixar claro, no entanto, que nesse novo modelo essas condições são necessárias enquanto o modelo anterior poderia ser adaptado, por exemplo, para ter-se dois subperíodos subsequentes com produção ou *setup*, o que pode ser útil em casos onde o tempo de *setup* não obedece a desigualdade triangular.

Para verificar se essa nova configuração é válida executou-se o modelo nos exemplos anteriormente discutidos (Problemas 1 e 2 da Tabela 13), tendo-se obtido exatamente os mesmos resultados. Sendo assim apresenta-se a seguir o modelo reformulado. Os parâmetros e constantes permaneceram iguais. No entanto, o conjunto de restrições (6) teve de ser quebrado em duas partes (17 e 18) para lidar com o primeiro intervalo de tempo enquanto outros tiveram que ser adaptadas, muitas vezes trocando somatórias por índices que devem ser repetidos ou o inverso.

Função objetivo

$$\text{Min} \sum_{p=1}^C \sum_{f=1}^F \sum_{e=1}^E \sum_{c_1=1}^P \sum_{c_2=1}^P \left(t_{p,f,e,c_1,c_2} + \sum_{e'=1}^E \sum_{c_1'=1}^P \sum_{c_2'=1}^P (s_{p,f,e,e'} \cdot Z_{e,e'}) \right) \quad (12)$$

Sujeito a:

$$\sum_{p=1}^C \sum_{f=1}^F \sum_{c'=1}^P ((t_{p,f,e,c,c'} + t_{p,f,e,c',c}) \cdot C_{p,e,c,c'}) \geq D_{e,c} \quad \forall e, c \quad (13)$$

$$t_{p,f,e,c_1,c_2} \leq d_{p,f,e,c_1,c_2} \cdot M \quad \forall p, f, e, c_1, c_2 \quad (14)$$

$$\sum_{e=1}^E \left(\sum_{c_1=1}^P \sum_{c_2=1}^P d_{p,f,e,c_1,c_2} + \sum_{e'=1}^E s_{p,f,e,e'} \right) \leq 1 \quad \forall p, f \quad (15)$$

$$\sum_{c_1=1}^P \sum_{c_2=1}^P d_{p,f,e,c_1,c_2} \leq \sum_{e'=1}^E s_{p,f-1,e',e} \quad \forall p, f > 1, e \quad (16)$$

$$\sum_{e'=1}^E s_{p,1,e,e'} \leq \sum_{c_1=1}^P \sum_{c_2=1}^P I_{p,e,c_1,c_2} \quad \forall p, e \quad (17)$$

$$d_{p,1,e,c_1,c_2} \leq I_{p,e,c_1,c_2}, \quad \forall p, e, c_1, c_2 \quad (18)$$

$$\sum_{e'=1}^E s_{p,f,e,e'} \leq \sum_{c_1=1}^P \sum_{c_2=1}^P d_{p,f-1,e,c_1,c_2} \quad \forall p, f > 1, e \quad (19)$$

$$\sum_{f=1}^F \sum_{e=1}^E \left(\sum_{c_1=1}^P \sum_{c_2=1}^P t_{p,f,e,c_1,c_2} + \sum_{e'=1}^E s_{p,f,e,e'} \cdot Z_{e,e'} \right) \leq T_p \quad \forall p \quad (20)$$

$$t_{p,f,e,b,c_1,c_2} \in R^+ \quad \forall p, f, e, b, c_1, c_2 \quad (21)$$

$$s_{p,f,e,e'} \in R^+ \quad \forall p, f, e, e' \quad (22)$$

$$d_{p,f,e,c_1,c_2} \in B \quad \forall p, f, e, c_1, c_2 \quad (23)$$

Para entender-se melhor o novo modelo, comenta-se as mudanças em cada expressão.

A função objetivo (1) foi modificada levemente, apenas para que o tempo seja calculado considerando as novas variáveis s (12).

O conjunto de restrições (2) não foi alterado sendo rerepresentado como (13).

O conjunto de restrições (3) não foi alterado sendo rerepresentado como (14).

O conjunto de restrições (4) foi apenas adaptado a mudança na variável s (15).

O conjunto de restrições (5) teve de internalizar os tamanhos de pacote, que agora são contabilizados através de somatórias e não mais para cada índice, refletindo a inexistência desses índices nas variáveis s (16).

O conjunto de restrições (6) foi quebrado em dois, sendo (17) para inicializar as variáveis s e (18) para as variáveis d . Vale notar que apenas uma variável s ou d assumirá valor 1, e não uma de cada, já que o conjunto de equações (15) garante isso.

O conjunto de restrições (7), analogamente ao (5) teve de transformar mais equações (identificadas pelos índices) por somatórias, tornando-se (19).

O conjunto de restrições (8) foi apenas adaptado a nova variável sem grandes mudanças (20).

Os conjuntos de restrições (9), (10) e (11) indicam o domínio das variáveis, e são rerepresentados como (21), (22) e (23). Ao observar-se as equações (10) e (22) nota-se não só a diminuição no número de parâmetros nas variáveis s , mas que elas agora são reais.

Essa mudança foi feita pois apenas uma variável do tipo s (a máquina faz *setup*) ou d (a máquina está preparada e produz) pode ser 1 a qualquer instante e todas as outras devem ser 0 (15). Além disso, é necessário para que uma variável d seja 1 que uma variável s seja 1 no instante anterior, já que só é possível haver produção em um intervalo se foi feito *setup* para ele no intervalo anterior (16) e só pode haver *setup* se houve produção no instante anterior (19).

Nessa configuração, como d_{p,f,e,c_1,c_2} é binária, têm-se dois casos. Caso $d_{p,f,e,c_1,c_2} = 1$ pelo menos uma $s_{p,f-1,e',e}$ deve ser um e todas as outras, como penalizam a função objetivo, serão 0. Caso $d_{p,f,e,c_1,c_2} = 0$ todas as $s_{p,f-1,e',e}$ serão zero, também por penalizar a função objetivo. Desse modo, todas as variáveis s são forçadas pelo modelo a também serem 0 ou 1 para possibilitar a produção no intervalo subsequente. Ou seja, basta as variáveis d serem binárias para que as próprias restrições se encarregam de fazer com que todas as s também sejam binárias, conforme Tabela 15.

Tabela 15: Valores das variáveis s em função das variáveis d

Valor de d_{p,f,e,c_1,c_2}	Valor de $s_{p,f-1,e',e}$
0	0
1	0 ou 1

Isso é positivo pois pode-se fazer $s \in R$ e diminuir a quantidade de variáveis binárias do modelo, aumentando sua performance. Considerando essas mudanças obtém-se o novo perfil de variáveis mostrado na Tabela 16.

Tabela 16: Nova configuração de variáveis

Conjunto de variáveis	Cálculo do N° variáveis	Total	Tipo
t_{p,f,e,c_1,c_2}	$5 \cdot 30 \cdot 14 \cdot 6 \cdot 6$	75.600	Real

Conjunto de variáveis	Cálculo do N° variáveis	Total	Tipo
$s_{p,f,e,e'}$	$5 \cdot 30 \cdot 14 \cdot 14$	29.400	Real
d_{p,f,e,c_1,c_2}	$5 \cdot 30 \cdot 14 \cdot 6 \cdot 6$	75.600	Binária

Comparando-se as duas situações na Tabela 17, percebe-se um perfil provavelmente mais fácil de resolver computacionalmente, com menos variáveis em geral e com transformação de parte das binárias em reais.

Tabela 17: Comparação do número de variáveis

Tipo	Modelo antigo	Modelo novo
Real	75.600	105.000
Binária	38.178.000	75.600

Após aproximadamente 14 horas de execução, o *software* não conseguiu obter a solução ótima do modelo, também excedendo a capacidade do computador. No entanto, diferentemente das vezes anteriores, soluções factíveis foram encontradas. Mais do que isso, ele atingiu essas soluções rapidamente. De fato, após 14 horas obteve-se um *gap* de otimalidade de aproximadamente 3,2%, sendo que em apenas 10 minutos, atinge-se um *gap* de aproximadamente 3,9%. Para se entender melhor o que isso significa, vejamos a definição abaixo.

O *gap* de otimalidade é definido (para problemas de minimização), segundo Kawamura (2011), como a diferença entre o valor da função objetivo da melhor solução factível encontrada e o limitante inferior do valor ótimo encontrado até então pelo mecanismo de resolução do problema. Vamos supor que para determinado modelo de programação linear inteira mista, a uma certa altura da resolução com objetivo de minimização, obteve-se uma solução factível cujo valor da função objetivo é de 10. No entanto, esse mesmo mecanismo, ao avaliar as

restrições, já concluiu que o menor valor possível para a função objetivo é 8. Desse modo, nesse momento, o *gap* de otimalidade é de 20%⁷.

Voltando ao problema real, um *gap* de 3,9% em 10 minutos indica que o mecanismo achou uma solução factível boa bastante rápido, enquanto um *gap* de 3.2% em 14 horas mostra que provar a otimalidade da solução é bastante difícil, sendo esse fenômeno bastante comum em problemas fortemente combinatórios.

Uma razão para isso é que o próprio paradigma de modelagem do *GLSP*, com subdivisão em um número arbitrário de subperíodos deixa naturalmente espaço para muitos subperíodos vazios, fazendo com que o método de resolução analise muitos subproblemas que não melhoram a solução.

Considerando que no ambiente real da empresa deseja-se ter respostas rápidas aos problemas, optou-se inicialmente por comparar a solução atual com aquela obtida pelo modelo em apenas 10 minutos, conforme Figura 14.

⁷ Cabe ressaltar que o limitante inferior não é comprovadamente o ótimo, e sim o menor valor possível para o ótimo. O mínimo do problema pode ser 10 e a solução ótima já foi encontrada, porém a otimalidade ainda não foi demonstrada, e por isso o *gap* existe. De fato, o ótimo é obtido justamente quando o *gap* é zero, ou seja, a solução atual é comprovadamente a melhor possível.

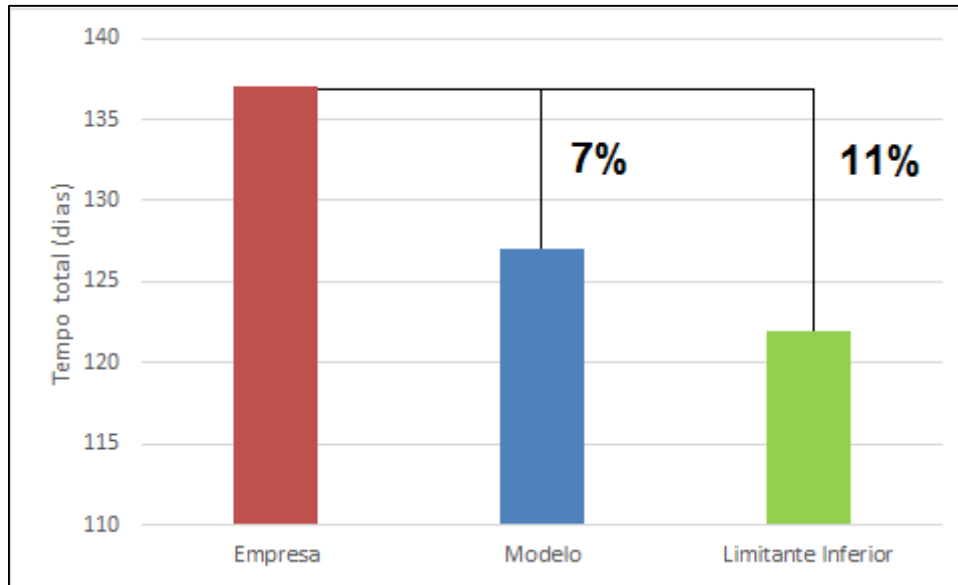


Figura 14: Comparação do resultado obtido em 10 minutos com o atual

A evolução da solução ao longo do tempo de processamento é mostrada na Figura 15.

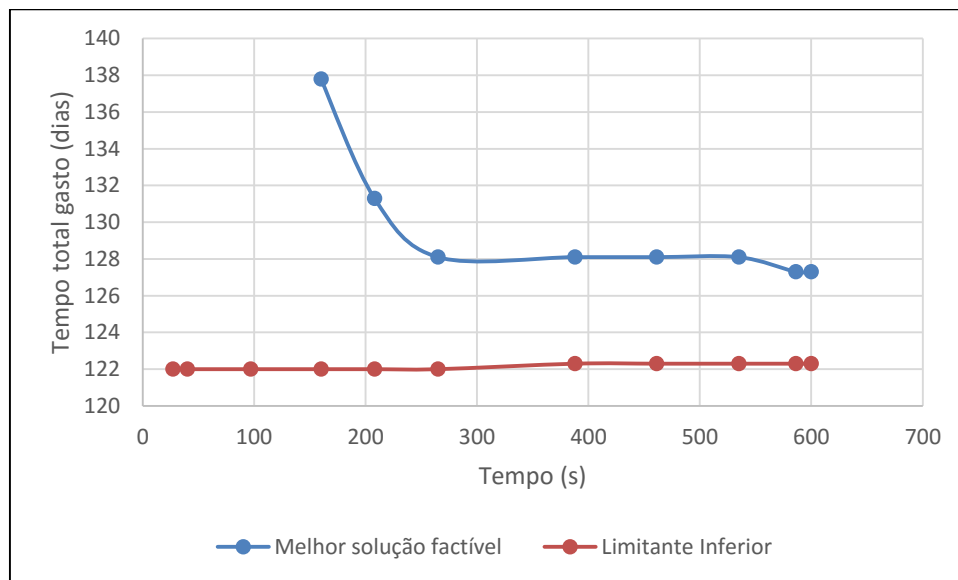


Figura 15: Evolução do valor da solução do modelo em 10 minutos

Considerando que em 10 minutos obtém-se uma melhoria de 7 % em relação a empresa e que o *gap* é muito baixo, decidiu-se manter esse tempo de resolução. De fato, observando a evolução do valor da solução ao permitir a execução do *software* por mais tempo, fica claro que há pouca vantagem nesse aumento, conforme mostrado na Figura 16.

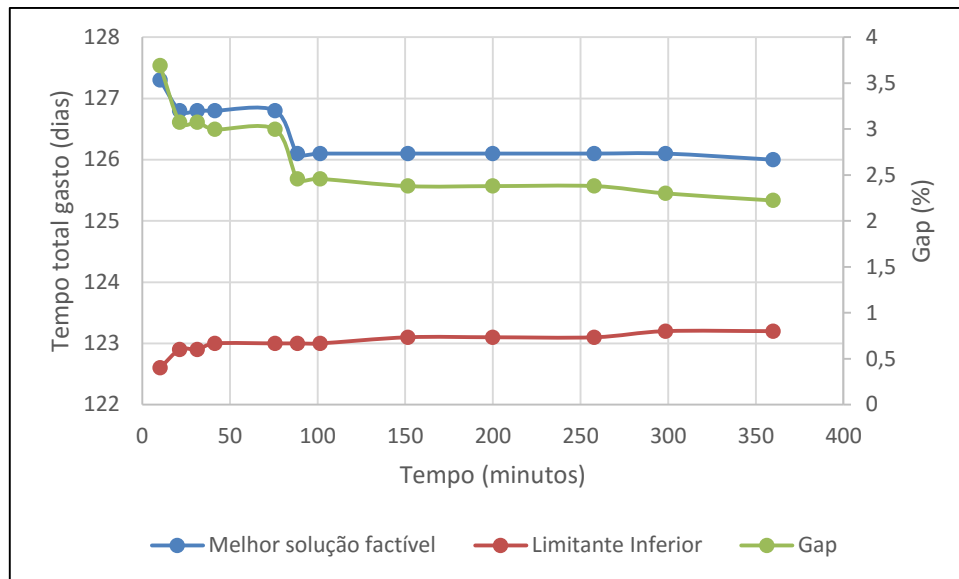


Figura 16: Evolução da solução obtida em 6 horas

Esse gráfico indica que uma solução é encontrada em menos de 10 minutos e que, apesar de dois momentos em que o mecanismo acha melhores soluções, a diferença é muito pequena entre a solução depois de 10 minutos, 60 minutos (1 hora) e 360 minutos (6 horas). O valor da função objetivo diminui de 127,3 para 126,8 entre 10 minutos e uma hora, uma melhoria de 0,4%. Multiplicando novamente o tempo por 6 e executando o modelo por 6 horas (360 minutos), o valor da função objetivo cai para 126,0 porém a melhoria em relação a solução de 10 minutos é de apenas 1%. Vale notar que a solução de 10 minutos é 7,0% melhor que a da empresa enquanto a solução de 6 horas é 8,0% melhor em com um tempo 36 vezes maior.

Vale destacar nesse gráfico que o *gap* vai diminuindo com o tempo, porém em muitos casos mais pelo aumento do limitante inferior do que pela diminuição da função objetivo. Isso mostra que, de fato, o modelo acha rapidamente uma boa solução porém demora a provar sua otimalidade.

Além disso, optou-se por não usar a heurística *relax-and-fix*, já que há pouco espaço da melhoria da qualidade ou tempo da solução em relação à obtida pelo método otimizador com tempo limitado (que não deixa de ser uma heurística). Sendo assim, foi decidido em conjunto com o a equipe de programação da produção responsável por essa linha manter-se o método otimizador com tempo de 10 minutos como mecanismo de solução.

Sendo assim apresenta-se na Figura 17 a programação da produção das células produtivas 1 a 3, onde os valores são o tempo em dias que cada subperíodo durará para aquela célula, seja produzindo ou fazendo *setup*. Na Figura 18 apresenta-se a programação das células 4 e 5.

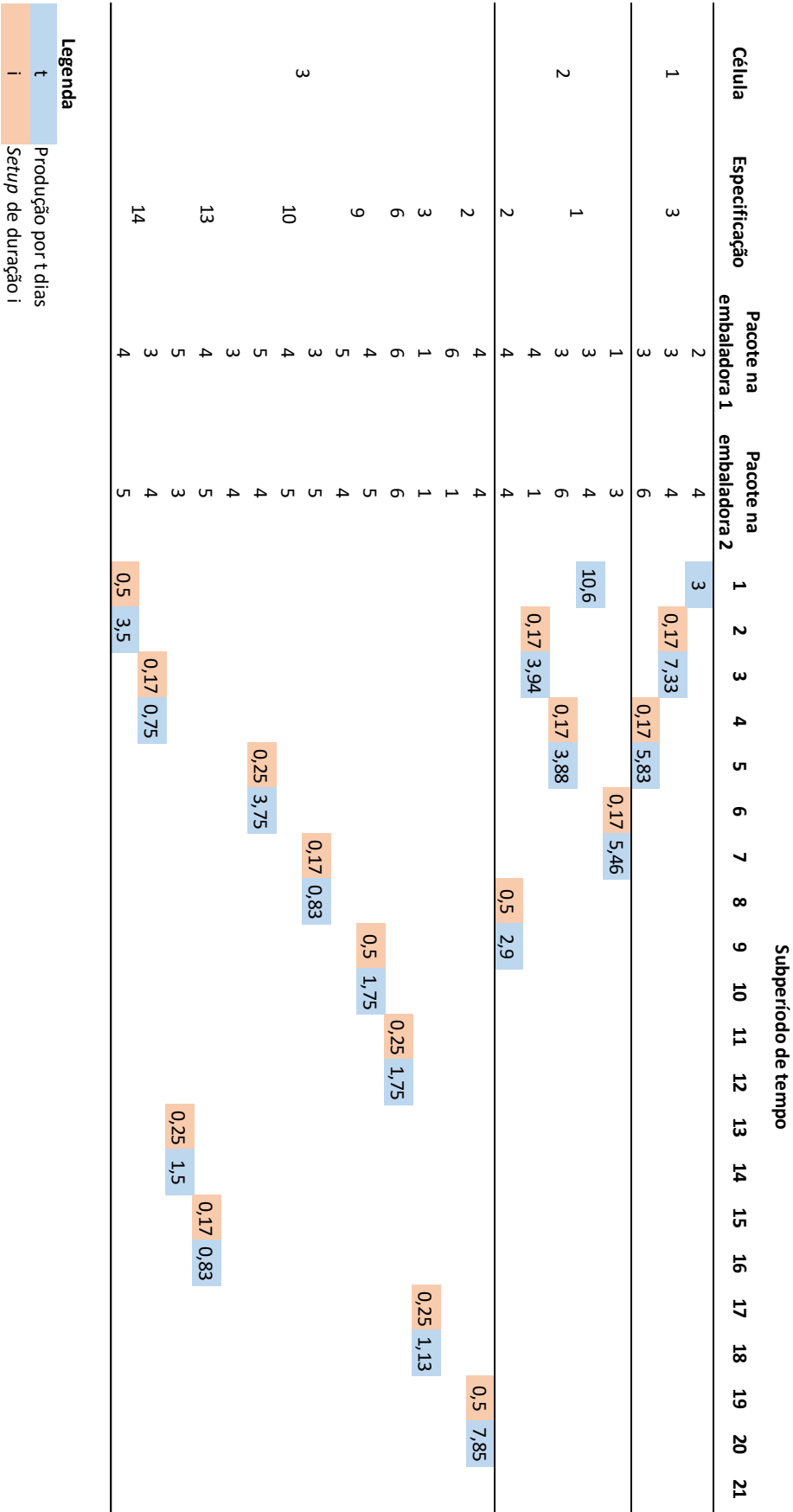


Figura 17: Programação da produção para células 1 a 3

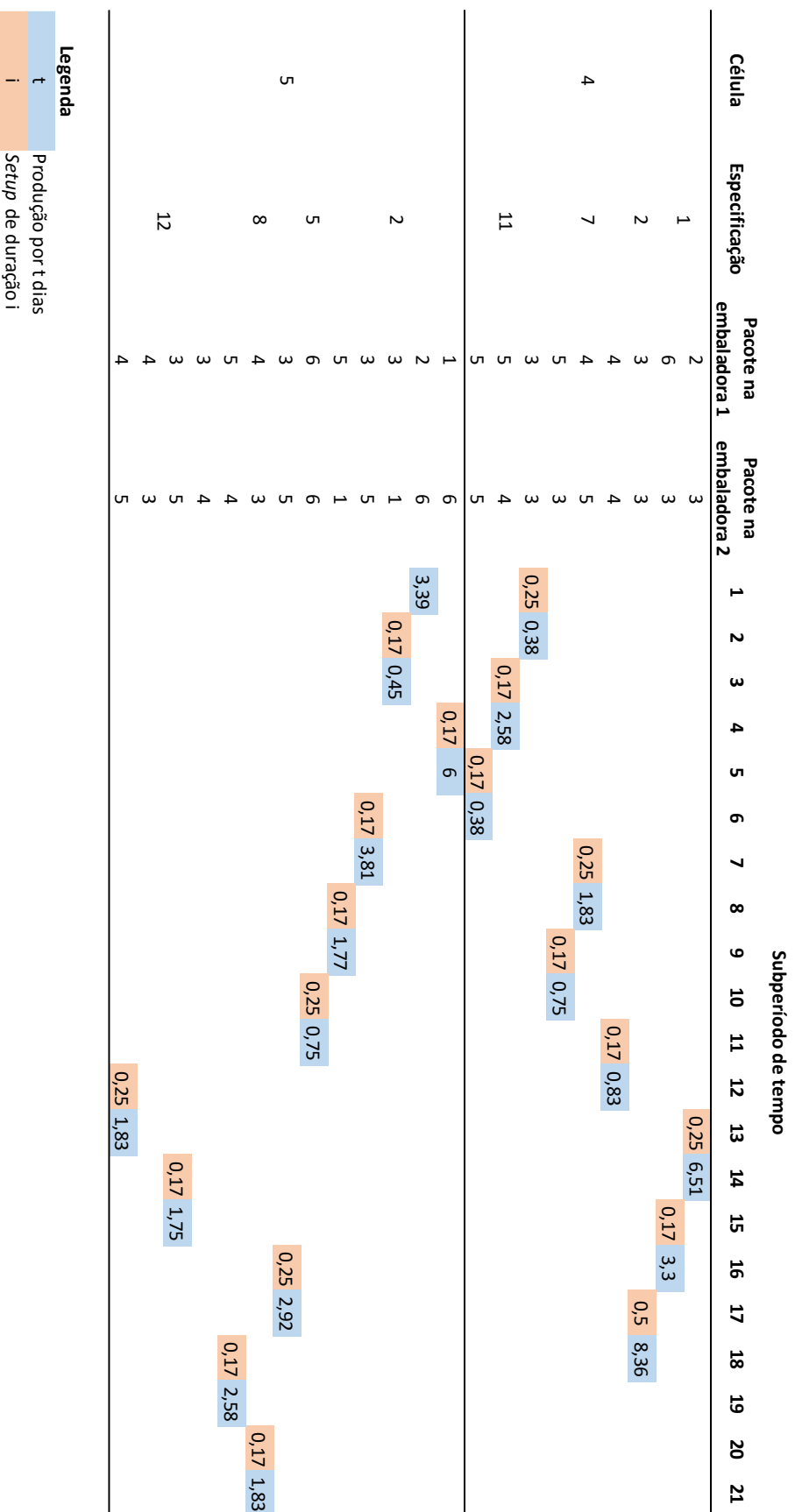


Figura 18: Programação da produção para as células 4 e 5

Pelas figuras (Figura 16 e Figura 17) fica fácil perceber a alternância de produção e *setup* (células azuis e vermelhas respectivamente), conforme era esperado. Além disso, é interessante notar que enquanto as células 1, 2 e 5 respeitam o *setup* inicial, o modelo decide para as células 3 e 4 fazer um *setup* no primeiro subperíodo, uma possibilidade que foi discutida nos testes preliminares e que de fato se mostrou útil no caso real.

Nesse tipo de situação onde começa-se com um *setup*, fica evidente a necessidade de se considerar não só o tempo de *setup*, mas também o tempo efetivamente produzindo de cada célula. Essa necessidade é uma consequência direta das máquinas serem diferentes entre si, aspecto que conforme discutido na revisão bibliográfica, traz mais complexidade ao modelo.

Outro aspecto interessante é que dos 30 subperíodos disponíveis, apenas 21 são efetivamente utilizados. A redução do número máximo de subperíodos, conforme já foi discutido, pode cortar soluções melhores ao torná-las não factíveis, porém é de se esperar que boas soluções tenham baixo número de subperíodos já que há forte correlação entre o número de subperíodos utilizados e o número de *setups* realizados. Uma solução com 25 subperíodos no máximo é discutida na seção 6.3.

Ainda em relação aos subperíodos, um aspecto interessante de notar é que estes variam sua duração para uma mesma célula em subperíodos diferentes e entre células para o mesmo subperíodo. Por exemplo na célula 5, o subperíodo cinco dura 6 dias e o três dura 0,45 dia. Já para a célula 4, ambos duram 0,17 dia (4 horas ou um *setup* de pacote), mostrando mais claramente como o *GLSP* funciona.

É interessante notar também que a célula 1, por exemplo, utiliza só 5 subperíodos. Fica transparente, dessa forma, que algumas células ficam especializadas para diminuir o número de *setups* (1 e 2) enquanto outras variam mais sua produção (3 a 5). Essa configuração com algumas células de pouca variedade e outras de muita já podia ser observado em algum nível na solução da empresa (que não é apresentada aqui a pedido da própria empresa), porém a resolução do modelo mostra que essa é realmente uma estratégia boa para esse sistema produtivo, conseguindo inclusive ampliá-la conforme o exemplo anterior. Cabe questionar, no entanto, se uma possível redução no número de SKUs não traria benefícios ao possibilitar que uma outra célula seja mais especializada.

Comparando com o plano da empresa têm-se uma redução do tempo de *setup* de aproximadamente 21% e ainda uma redução de 6% no tempo efetivamente produzindo. Essa melhoria no tempo produzindo se dá pois a capacidade das células varia, de modo que uma melhor alocação dos lotes fez com que mais produtos fossem feitos nas células que os fazem de forma mais eficiente.

Deve-se também considerar que uma redução de 21% no tempo de *setup* implica em redução de custos para essa linha, já que há menor utilização de mão-de-obra relacionada a essa atividade, menos insumos desperdiçados em produtos de qualidade baixa até que a máquina volte a produzir produtos conformes e mais disponibilidades para atender pedidos extraordinários, que em outros casos gerariam perda de venda. De fato, como *setup* é uma questão chave nessa empresa, pode-se usar o modelo também para avaliar o impacto de redução no tempo de cada *setup* no resultado final, o que é discutido na seção 6.4. Na seção 6.5, analisa-se o uso do modelo com capacidade de produção dependente da combinação, abrindo espaço para que a empresa implemente essa solução.

6.3 Efeito do número máximo de subperíodos

Para melhor compreender o efeito do número máximo de subperíodos, após verificar-se que possibilitando o uso de 30 subperíodos a solução do modelo só usa 21, executou-se o modelo com um limite de 25 subperíodos, também em 10 minutos. Na comparação, mostrada na Tabela 18, verifica-se que a diminuição do número de subperíodos melhora solução (reduz o valor da solução factível) e também encontra um limitante inferior maior, diminuindo *gap* de otimalidade de 3,9 para 3,2 % no mesmo tempo de execução.

Tabela 18: Comparação entre 25 e 30 subperíodos

Subperíodos	Solução factível (dias)	Limitante Inferior (dias)	Gap (%)
30	127,3	122,3	3,9
25	127,1	122,6	3,5

No entanto, não seria possível saber *a priori* quantos subperíodos são de fato necessários. Desse modo, será considerada para efeito desse trabalho a solução obtida em 10 minutos com um número máximo de 30 subperíodos, já que essa é uma alternativa mais segura em relação a não perder boas soluções e por isso foi acordada entre o autor e a empresa como a oficial.

6.4 Análise de sensibilidade: Tempo de *setup*

Pensando em auxiliar a empresa em reduzir ainda mais o tempo gasto com *setups*, e considerando que o modelo representou adequadamente a situação real, decidiu-se avaliar o impacto da redução de cada tempo de *setup* individualmente no resultado do modelo. Busca-se com isso verificar qual *setup* (entre tamanhos de fralda, tipos de fralda ou tamanhos de pacote) mais beneficiaria a empresa se reduzido. Essa redução, que não é escopo desse projeto, poderia ser alcançada por técnicas conhecidas da produção enxuta, como por exemplo a *SMED (Single Minute Exchange of Dies)* discutida por exemplo em Shingo (1985). No entanto, como esse tipo de técnica consome recursos é interessante saber em qual *setup* uma possível redução traria mais impacto para a empresa.

Dessa forma, mantendo-se um tempo de resolução de 10 minutos e 30 subperíodos, utilizou-se o modelo para prever o impacto na função objetivo do modelo de uma redução do tempo de *setup* em um terço para cada um dos tipos de *setup* por vez. Esses resultados são apresentados na Tabela 19. É mostrado o *gap* de otimalidade também para avaliar a qualidade da solução em relação ao ótimo daquela configuração. Além disso são dadas as melhorias em relação a solução atual da empresa.

Tabela 19: Impacto da redução do tempo de cada tipo de *setup*

Solução	Empresa	Modelo	Reduzindo <i>setup</i> entre tamanhos de fralda	Reduzindo <i>setup</i> entre tipos de fralda	Reduzindo <i>setup</i> entre tamanhos de pacote
Tempo total (dias)	137	127	128	126	126
Melhoria do tempo total em relação à empresa (%)	-	7	6	8	8
<i>Gap</i> de otimalidade (%)	-	3,9	4,5	2,0	3,5

Solução	Empresa	Modelo	Reduzindo <i>setup</i> entre tamanhos de fralda	Reduzindo <i>setup</i> entre tipos de fralda	Reduzindo <i>setup</i> entre tamanhos de pacote
Tempo de <i>setup</i>	10,6	8,3	9,3	6,8	7,7
Melhoria do tempo de <i>setup</i> em relação à empresa (%)	-	21	13	36	27
Melhoria do tempo de <i>setup</i> em relação ao modelo (%)	-	-	-12	18	7

A partir dessa tabela, verifica-se que o *setup* que teria maior impacto se tivesse seu tempo reduzido é entre tipos de fraldas. Através dessa melhoria, ter-se-ia uma redução de 36% no tempo total *setup* em relação à empresa e uma melhoria de 18% em relação ao modelo. É interessante que o efeito sobre o tempo total é menor mesmo em termos absolutos, o que se dá pois aceita-se capacidades produtivas menores nesses casos para diminuir o tempo de *setup*.

Esse resultado é interessante por esse se tratar do *setup* de duração intermediária, 6 horas. Apesar do *setup* de tamanho de fralda ser o maior, ele é pouco feito já que a programação já visa dividir certos tamanhos por célula. Por outro lado, o *setup* entre tamanhos de pacote é frequente, mas por ser de menor duração, sua redução proporcional tem um impacto menor.

Outro ponto interessante é que a melhor solução tem também um *gap* menor, ou seja, essa redução faz com que o mecanismo de resolução mais rapidamente chegue em um bom resultado. Uma possível explicação para isso é que se torna indiferente nesse caso fazer uma mudança de tipo de fralda ou de tamanho de pacote. Dessa forma, há provavelmente várias boas soluções e, apesar do mecanismo ainda demorar para chegar de fato na melhor, ele encontra rapidamente uma solução boa.

Esse mesmo raciocínio também explica o porquê de, ao se diminuir o tempo de *setup* de tamanhos de fralda, a solução piorar em relação ao modelo sem a diminuição. Nesse caso, soluções com muitos *setups* de tamanho de fralda, que antes eram rapidamente eliminadas, tornam-se mais atrativas, já que a duração de um *setup* de tamanho fica próxima a de uma mudança de tipo. Com isso, provavelmente há várias alternativas de solução não tão boas, mas que tornam mais difícil para o modelo encontrar as soluções realmente boas.

Percebe-se dessa maneira que os dados de entrada influenciam na qualidade da solução ao tornar mais frequentes soluções boas ou ruins (caso dos dois últimos parágrafos, respectivamente), fazendo com que a solução obtida seja melhor ou pior dependendo de qual ponto o mecanismo de resolução “engargala”. Essa discussão é retomada na próxima seção.

Vale notar que essa variação só acontece pois executa-se o modelo com tempo limitado, sendo que se permitíssemos que o modelo atingisse o ótimo, a solução necessariamente deveria ser melhor para os casos com *setup* menor do que para o modelo sem essa redução. De fato, se executarmos o modelo com *setup* de tamanho mais rápido por uma hora, e não dez minutos, o tempo total de *setup* cai para 6,5 dias com tempo de produção de 119 dias aproximadamente. Dessa forma, deve-se tomar cuidado com a redução do tempo de execução sem antes verificar o efeito dessa redução na qualidade da solução.

Conclui-se assim que uma redução no tempo de *setup* entre tipos de fralda é o que traria mais impacto para a empresa, diminuindo em 8% o tempo total e 36% o tempo de *setup*. Ressalta-se também que a melhoria no tempo de *setup* em relação a solução da empresa, dos 36% de redução, 21% já tinham sido atingidos apenas com a programação do modelo. Isso indica que, nesse caso, o impacto de uma boa programação é maior do que a da redução do *setup* em si, um resultado que mostra a importância da pesquisa operacional dentro do contexto de melhoria das operações industriais. Por fim, vale lembrar que os atuais tempos de *setup* foram definidos em conjunto com o sindicato, de modo que qualquer proposta de mudança deve ser feita com cuidado e buscando a participação e consentimento deles, buscando assim soluções que não prejudiquem os trabalhadores para evitar problemas em negociações futuras.

6.5 Análise de sensibilidade: Capacidade dependente da combinação de pacotes

Conforme foi apresentado no capítulo 5, apesar da empresa reconhecer que a capacidade de produção depende da combinação de tamanhos de pacote sendo empacotados simultaneamente

nas duas embaladoras, essa informação não é usada hoje na programação da produção e nem existem dados sobre as capacidades reais nesses casos.

Segundo o gerente de planejamento e programação da produção dessa empresa, ele gostaria de implementar uma solução que levasse em conta essa diferença, porém isso não seria possível com o método atual da empresa. Dessa forma, o modelo foi desenvolvido para esse caso, conforme já foi discutido nos capítulos anteriores, mas foi executado com capacidades independentes para ser comparável com a solução da empresa.

Visando compreender como o modelo se comporta nesse caso, mesmo sabendo que a mudança do parâmetro de capacidade não aumenta a dimensão do modelo, decidiu-se junto com a empresa executar um caso teste usando o modelo completo e dados adaptados.

Supondo que quando os tamanhos de pacote são iguais há uma redução da complexidade que poderia acarretar em um aumento da capacidade, buscou-se verificar como modelo se comportaria caso a capacidade, quando os dois tamanhos de pacote sendo embalados são iguais, fosse 10% maior. Nota-se que esse não é necessariamente o caso, já que um estudo mais profundo seria necessário, sendo esse apenas um pretexto para entender o efeito de se usar a capacidade dependente em um caso de mesma complexidade computacional que o real. Desse modo, o que se deseja verificar é o *gap* de otimalidade e, portanto, a qualidade da solução, e não o valor da função objetivo em si.

Ao resolver o modelo com 30 subperíodos e 10 minutos (mesma configuração usada anteriormente para comparação com a empresa), obteve-se um *gap* de otimalidade de apenas 1,2%. O fato do *gap* na verdade ser menor do que o caso original pode ser explicado pelo mesmo fenômeno mencionado na análise de sensibilidade anterior, ou seja, que variações nos dados podem tornar as soluções factíveis mais ou menos dispersas.

Em um caso onde as soluções factíveis são mais bem distribuídas ao longo do espectro de valores, torna-se mais fácil achar boas soluções rapidamente. Já em um caso onde há uma concentração de valores próximos do ótimo (muitos sub-ótimos), fica mais difícil encontrar os valores melhores dentre estes, apesar de ser mais rápido achar valores dentro dessa região de proximidade. Esses dois casos são exemplificados na Figura 19.

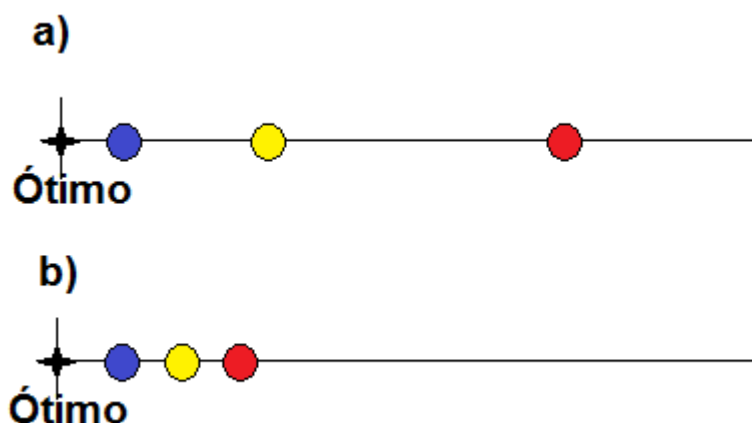


Figura 19: Ótimo e soluções factíveis. a) Soluções factíveis dispersas. b) Soluções factíveis próximas

Para melhor compreender a figura, é necessário entender como o *Branch and Bound* e o *Branch and Cut*, principais métodos de resolução de problemas lineares inteiros mistos funcionam. Esses métodos fazem uma árvore de soluções representando os possíveis valores das variáveis binárias e então vão resolvendo cada ramo. Até que se atinja uma solução factível, prove-se que naquele ramo não existe solução factível ou que a melhor solução factível possível desse ramo é pior que uma já encontrada.

Observando o caso da Figura 19 a), assumindo que o modelo comece analisando o ramo que leve a solução factível amarela, ele vai ter uma solução não tão boa de início, porém se em seguida ele analisar o ramo que levará à solução vermelha, ele provavelmente vai rapidamente demonstrar que a melhor solução possível deste ramo é pior do que a que ele já tem, não sendo necessário aprofundar a busca nesse ramo. Descartando rapidamente soluções de baixa qualidade, aumenta a probabilidade de o método resolver ramos que levem a soluções melhores, como por exemplo a azul.

Observando agora a Figura 19 b), assumindo novamente que ele parta do ramo que leva a solução factível amarela, ele achará uma boa solução rapidamente, porém se ele então for para o ramo que leva a solução factível vermelha, poderá demorar muito até provar que de fato a solução vermelha é pior que a amarela. Nesse caso, diminui a probabilidade de no tempo dado ele explorar ramos que levem a soluções ainda melhores, como por exemplo a azul.

Levando em conta tudo isso, pode-se concluir que o modelo sempre apresenta *gaps* pequenos em um período curto de tempo (em todas as análises ele foi inferior a 5% em dez minutos),

porém pelos efeitos discutidos acima, os dados de entrada podem influir nele. Sendo assim, recomenda-se que se houver alguma mudança drástica no uso do modelo, como por exemplo seu uso em outras linhas de produção da empresa na qual o modelo se aplicaria, deve-se observar se o *gap* de otimalidade continua baixo e, caso contrário, encontrar um novo tempo de resolução do modelo que satisfaça os requisitos de qualidade para a solução.

7 CONCLUSÕES

Tendo em vista os resultados apresentados no capítulo anterior, verifica-se que o trabalho foi capaz de criar com sucesso um modelo válido para representar o problema da empresa, e através desse modelo, encontrar uma solução para esse problema que satisfaça as condições de contorno propostas. Esse modelo, diferente em suas características específicas dos encontrados na literatura, é capaz de em apenas 10 minutos, obter uma solução muito próxima do ótimo e superior em 7% a utilizada pela empresa em termos de tempo total e 21% em tempo de *setup*.

Além disso, o modelo é capaz de lidar com capacidades diferentes dependendo da combinação de tamanhos de pacote sendo produzidos simultaneamente. Essa característica do problema, até então não levada em conta de forma sistemática na programação da produção, é facilmente integrada no modelo, como mostraram exemplos desenvolvidos ao longo do trabalho e uma análise de sensibilidade com esse fim. Cabe notar que o modelo continua, nesse caso, obtendo uma solução boa rapidamente. De fato, para um conjunto de dados exemplo, obteve-se um *gap* de otimalidade de apenas 1,2% para esse caso.

Dessa forma, esse trabalho apresenta várias contribuições para a empresa. Primeiramente, ele apresenta hoje uma solução melhor para a programação de fraldas usando ainda a suposição da capacidade independente. Além disso, trata-se de uma alternativa já pronta para lidar com a complexidade real do problema, ou seja, com capacidades dependentes da combinação. Cabe notar que outros itens da empresa são produzidos de forma semelhante, de modo que essa solução também pode ser adaptada para outras unidades produtivas. Por fim, através de uma análise de sensibilidade indica-se também qual *setup* teria mais impacto se reduzido, por exemplo através de uma técnica como *SMED*.

Além dessas contribuições para a empresa, esse trabalho contribui para a literatura sobre dimensionamento e sequenciamento de lotes por apresentar um modelo de programação linear inteira mista monoestágio para problemas de dois estágios em que uma mesma máquina de estágio 1 atende duas máquinas de estágio 2, sendo esse modelo expansível para várias máquinas de estágio 2.

Para trabalhos futuros sugere-se, primeiramente, a verificação da efetividade do modelo e sua performance com dados reais em casos onde a capacidade é dependente da combinação de

tamanhos de pacote. Além disso, seria interessante saber como a performance computacional do modelo varia se adicionarmos mais embaladoras (mais máquinas de segundo estágio). Para a empresa, seria interessante também verificar a aplicabilidade do modelo a outras linhas produtivas, além de sua possível expansão para considerar outros fatores como pedidos atrasados, por exemplo.

8 BIBLIOGRAFIA

- ABIHPEC. Panorama do setor de HPPC. São Paulo: Associação Brasileira da Indústria de Higiene Pessoal, Perfumaria e Cosméticos, 2015.
- DREXL, A.; KIMMS, A. Lot Sizing and Scheduling - Survey and Extensions. **European Journal of Operations Research**, v. 99, p. 221-235, 1997.
- FERREIRA, D.; ALMADA-LOBO, B.; MORABITO, R. Formulações monoestágio para o problema de programação da produção de bebidas dois estágios com sincronia. **Produção**, v 23, p. 107-119, 2013.
- FERREIRA, D.; MORABITO, R., RANGEL, S. “Solution approaches for the soft drink integrated production lot sizing and scheduling problem.” **European Journal of Operations Research** doi:10.1016/j.ejor.2008.03.035, v. 196, n. 2, p. 697-706, 2009.
- 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 refrigerante de pequeno porte. **Produção** v 18, n 1, p. 76-88, 2008.
- GUPTA, D.; MAGNUSSON, T. The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times. **Computer & Operations Research**, v. 32, p. 727-747, 2005.
- KARIMI, B.; FATEMI GHOMI, S. M. T.; WILSON, J. M. The capacitated lot sizing problem: a review of models and algorithms. **Omega - The International journal of Management Science**, v. 31, p. 365-378, 2003.
- KAWAMURA, M. S. **Dimensionamento e sequenciamento de lotes de produção na indústria de bens de consumo de higiene pessoal**. 2011. Tese de doutorado apresentada à Universidade de São Paulo, São Paulo.
- MEYR, H. Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. **European Journal of Operations Research**, v. 120, p.311-326, 2000.
- MEYR, H. Simultaneous lotsizing and schedulling on parallel machines. **European Journal of Operations Research**, v. 139, p. 277-292, 2002.

- NEMHAUSER, G. L.; WOSLEY, L.A. **Integer and Combinatorial Optimization**. New York: Wiley, 1988.
- PINEDO, M. **Scheduling theory, algorithms and systems**. New Jersey: Prentice Hall, 2002.
- POCHET, Y.; WOLSEY, L. **Production planning by mixed integer programming**. New York: Springer, 2006.
- QUADT, D.; H. KUHN. Capacitated lot-sizing with extensions: a review. **4OR - Quarterly Journal of Operations Research**, v. 6, n. 1, p. 61-83, 2009.
- SABOIA, P. L. J. **Problema integrado de dimensionamento e sequenciamento de lotes de produção em uma fábrica de produtos de higiene pessoal**. 2010. Trabalho de formatura apresentado à Universidade de São Paulo, São Paulo.
- SHINGO, S. **A revolution in manufacturing: the SMED system** Cambridge: Productivity Press, 1985
- TOLEDO, C. F. M. **Problema conjunto de dimensionamento de lotes e programação da produção**. 2005. Tese de doutorado apresentada à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas, Campinas.
- 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. 27, n. 1, p. 155-186, jan/abr. 2007.
- TOSO, E.; MORABITO, R. Otimização no dimensionamento e sequenciamento de lotes de produção: estudo de caso numa fábrica de rações. **Gestão da Produção**, v. 12, n. 2, p. 203-217, 2005.
- TOSO, E.; MORABITO, R.; CLARK, A.. Combinação de abordagens glsp e atsp para o problema de dimensionamento e sequenciamento de lotes de produção de suplementos para nutrição animal. **Pesquisa Operacional**, v. 28, n. 3, p. 423-450, set./dez. 2008.
- ZHU, X., WILHELM, W. E. Scheduling and lot sizing with sequence-dependent setup: a literature review. **IEEE Transactions**, v. 38, p. 987-1007, 2006.

ANEXO 1: TEMPOS DE *SETUP*

Na Tabela 20 são apresentados os tempos de *setup*, sendo 12 horas para mudanças de tamanho de fralda, 6 horas para mudanças de tipo de fralda e 4 horas para mudanças de tamanho de pacote. Para o modelo convencionou-se o uso da unidade dia então os valores efetivamente utilizados são os números abaixo divididos por 24.

Tabela 20: Tempos de *setup*

e	e'	Z	e	e'	Z
1	1	4	8	1	12
	2	12		2	6
	3	12		3	12
	4	6		4	12
	5	12		5	6
	6	12		6	12
	7	6		7	12
	8	12		8	4
	9	12		9	12
	10	12		10	12
	11	6		11	12
	12	12		12	6
	13	12		13	12
	14	12		14	12
2	1	12	9	1	12
	2	4		2	12
	3	12		3	6
	4	12		4	12
	5	6		5	12
	6	12		6	6
	7	12		7	12
	8	6		8	12
	9	12		9	4
	10	12		10	12
	11	12		11	12
	12	6		12	12
	13	12		13	6

e	e'	Z	e	e'	Z
	14	12		14	12
3	1	12	10	1	12
	2	12		2	12
	3	4		3	12
	4	12		4	12
	5	12		5	12
	6	6		6	12
	7	12		7	12
	8	12		8	12
	9	6		9	12
	10	12		10	4
	11	12		11	12
	12	12		12	12
	13	6		13	12
	14	12		14	6
4	1	6	11	1	6
4	2	12	11	2	12
	3	12		3	12
	4	4		4	6
	5	12		5	12
	6	12		6	12
	7	6		7	6
	8	12		8	12
	9	12		9	12
	10	12		10	12
	11	6		11	4
	12	12		12	12
	13	12		13	12
	14	12		14	12
5	1	12	12	1	12
	2	6		2	6
	3	12		3	12
	4	12		4	12
	5	4		5	6
	6	12		6	12
	7	12		7	12
	8	6		8	6

e	e'	Z	e	e'	Z
	9	12		9	12
	10	12		10	12
	11	12		11	12
	12	6		12	4
	13	12		13	12
	14	12		14	12
6	1	12	13	1	12
	2	12		2	12
	3	6		3	6
	4	12		4	12
	5	12		5	12
	6	4		6	6
	7	12		7	12
	8	12		8	12
	9	6		9	6
	10	12		10	12
	11	12		11	12
	12	12		12	12
	13	6		13	4
	14	12		14	12
7	1	6	14	1	12
	2	12		2	12
	3	12		3	12
	4	6		4	12
	5	12		5	12
	6	12		6	12
	7	4		7	12
	8	12		8	12
	9	12		9	12
	10	12		10	6
7	11	6	14	11	12
	12	12		12	12
	13	12		13	12
	14	12		14	4

ANEXO 2: TAXAS DE PRODUÇÃO

Para cada combinação de máquina produtora, especificação, tamanho do pacote sendo embalado na embaladora em questão e tamanho do pacote sendo embalado na outra embaladora, há uma taxa de produção, conforme Tabela 21. Vale notar que nem todas as combinações de tamanhos de pacote são possíveis. Para as combinações não presentes nessa tabela, $C = 0$. Por fim, é importante perceber que conforme explicado no capítulo “Coleta de Dados”, a taxa de produção usada até o momento pela empresa para cada embaladora era simplificada para ser independente da combinação.

A capacidade apresentada abaixo refere-se apenas a uma embaladora, sendo, portanto, a taxa dobrada quando ambas as embaladoras são usadas. As taxas são dadas em pacotes por dia por embaladora.

Tabela 21: Capacidade de produção por unidade de tempo

p	e	c	C
1	2	1	620.57
	3	1	398.94
	2	2	372.34
	3	2	438.83
	2	3	354.61
	3	3	504.73
	2	4	310.28
	3	4	506.59
	2	6	330.97
	3	6	506.59
2	2	1	354.61
	1	1	678.19
	2	2	452.13
	1	2	376.77
	2	3	430.60
	1	3	381.21
	2	4	487.59
	1	4	433.41
	2	6	487.59

p	e	c	C
	1	6	381.21
3	2	1	726.95
	3	1	797.87
	2	2	452.13
	2	3	445.80
	3	3	520.09
	2	4	445.80
	3	4	445.80
	2	6	445.80
	6	6	459.13
	9	3	638.30
	9	4	589.20
	9	5	589.20
	10	3	602.84
	10	4	556.46
	10	5	556.46
3	13	3	589.20
	13	4	589.20
	13	5	589.20
	14	3	602.84
	14	4	556.46
	14	5	556.46
4	1	1	177.30
	1	2	398.94
	2	3	536.17
	1	3	398.94
	1	4	433.41
	1	4	433.41
	1	6	398.94
	4	6	340.43
	7	3	413.71
	7	4	372.34
	7	5	425.53
	11	3	413.71
	11	4	372.34
	11	5	425.53
5	2	1	975.18

<i>p</i>	<i>e</i>	<i>c</i>	<i>C</i>
	2	2	585.11
	2	3	567.38
	2	4	496.45
	2	5	496.45
	2	6	567.38
	5	6	435.66
	8	3	624.11
	8	4	585.11
	8	5	638.30
	12	3	624.11
	12	4	585.11
	12	5	638.30

ANEXO 3: PRIMEIRO MODELO IMPLEMENTADO NO IBM OPL IDE

O modelo deste trabalho foi implementado no *software* IBM OPL IDE, que por sua vez faz interface com o CPLEX. Nesse *software* pode-se separar dados de entrada, guardados em um arquivo *.dat* do modelo em si, guardado em um arquivo *.mod*. Como o arquivo de dados é extenso, e é uma mera reescrita dos dados já apresentados, ele não será apresentado. No entanto, o modelo, que pode ser facilmente reutilizado para outros contextos e conjuntos de dados, é apresentado a seguir.

//Índices

int NbTypes =...;

range Type = 1..NbTypes;

int NbPackages = ...;

range Package = 1..NbPackages;

int NbProductionCells = ...;

range ProductionCells = 1..NbProductionCells;

int NbTimeFrames = ...;

range TimeFrames = 1..NbTimeFrames;

range TimeFrames2 = 2..NbTimeFrames;

int M = ...;

//Parametros

float Capacity[ProductionCells][Type][Package][Package] = ...;

float Demand[Type][Package] = ...;

float SetupTime[Type][Type] = ...;

```
float SetupZero[ProductionCells][Type][Package][Package] = ...;
```

```
float TotalTime[ProductionCells] = ...;
```

```
//Variáveis
```

```
dvar float+ ProductionTime[ProductionCells][TimeFrames][Type][Package][Package];
```

```
dvar float+
```

```
Setup[ProductionCells][TimeFrames][Type][Package][Package][Type][Package][Package];
```

```
dvar float+ Doing[ProductionCells][TimeFrames][Type][Package][Package]; //d
```

```
//Modelo
```

```
minimize
```

```
sum( p in ProductionCells, f in TimeFrames, a in Type, c1 in Package, c2 in Package )
```

```
(ProductionTime[p][f][a][c1][c2]) +
```

```
sum( p in ProductionCells, f in TimeFrames, a1 in Type, c11 in Package, c12 in Package, a2
in Type, c21 in Package, c22 in Package )
```

```
(Setup[p][f][a1][c11][c12][a2][c21][c22] * SetupTime[a1][a2]);
```

```
subject to {
```

```
forall( a in Type, c1 in Package )
```

```
ctDemand:
```

```
sum( p in ProductionCells, f in TimeFrames, c2 in Package )
```

```
((ProductionTime[p][f][a][c1][c2] +
```

```
ProductionTime[p][f][a][c2][c1]) * Capacity[p][a][c1][c2]) >= Demand[a][c1];
```

forall(p in ProductionCells, f in TimeFrames, a in Type, c1 in Package, c2 in Package)

ctDoing:

ProductionTime[p][f][a][c1][c2] <= M * Doing[p][f][a][c1][c2];

forall(p in ProductionCells, f in TimeFrames)

ctDoingorSetup:

sum(a in Type, c1 in Package, c2 in Package)

(Doing[p][f][a][c1][c2]) +

sum(a1 in Type, c11 in Package, c12 in Package, a2 in Type, c21 in Package, c22 in Package)

(Setup[p][f][a1][c11][c12][a2][c21][c22])

<= 1;

forall(p in ProductionCells, f in TimeFrames2, a in Type, c1 in Package, c2 in Package)

ctDoingAfterSetup:

Doing[p][f][a][c1][c2] <=

sum(a2 in Type, c21 in Package, c22 in Package)

(Setup[p][f-1][a2][c21][c22][a][c1][c2]);

forall(p in ProductionCells, a in Type, c1 in Package, c2 in Package)

ctInitializeSetup:

Doing[p][1][a][c1][c2] +

sum(a2 in Type, c21 in Package, c22 in Package)

(Setup[p][1][a][c1][c2][a2][c21][c22])

<= SetupZero[p][a][c1][c2];

forall(p in ProductionCells, f in TimeFrames2, a in Type, c1 in Package, c2 in Package)

ctSetupAfterDoing:

sum(a2 in Type, c21 in Package, c22 in Package)

(Setup[p][f][a][c1][c2][a2][c21][c22]) <=

Doing[p][f-1][a][c1][c2];

forall(p in ProductionCells)

ctTime:

sum(f in TimeFrames, a in Type, c1 in Package, c2 in Package)

(ProductionTime[p][f][a][c1][c2]) +

sum(f in TimeFrames, a1 in Type, c11 in Package, c12 in Package, a2 in Type, c21 in
Package, c22 in Package)

(Setup[p][f][a1][c11][c12][a2][c21][c22] * SetupTime[a1][a2]) <= TotalTime[p];

};

ANEXO 4: SEGUNDO MODELO IMPLEMENTADO NO IBM OPL IDE

Os modelos deste trabalho foram implementados no *software* IBM OPL IDE, que por sua vez faz interface com o CPLEX. Nesse *software* pode-se separar dados de entrada, guardados em um arquivo *.dat* do modelo em si, guardado em um arquivo *.mod*. Como o arquivo de dados é extenso, e é uma mera reescrita dos dados já apresentados, ele não será apresentado. No entanto, o modelo, que pode ser facilmente reutilizado para outros contextos e conjuntos de dados, é apresentado a seguir.

//Índices

int NbTypes =...;

range Type = 1..NbTypes;

int NbPackages = ...;

range Package = 1..NbPackages;

int NbProductionCells = ...;

range ProductionCells = 1..NbProductionCells;

int NbTimeFrames = ...;

range TimeFrames = 1..NbTimeFrames;

range TimeFrames2 = 2..NbTimeFrames;

int M = ...;

//Parâmetros

float Capacity[ProductionCells][Type][Package][Package] = ...;

float Demand[Type][Package] = ...;

float SetupTime[Type][Type] = ...;

```
float SetupZero[ProductionCells][Type][Package][Package] = ...;
```

```
float TotalTime[ProductionCells] = ...;
```

```
//Variáveis
```

```
dvar float+ ProductionTime[ProductionCells][TimeFrames][Type][Package][Package];
```

```
dvar float+ SetupE[ProductionCells][TimeFrames][Type][Type];
```

```
dvar boolean Doing[ProductionCells][TimeFrames][Type][Package][Package];
```

```
//Modelo
```

```
minimize
```

```
sum( p in ProductionCells, f in TimeFrames, a in Type, c1 in Package, c2 in Package )
```

```
(ProductionTime[p][f][a][c1][c2]) +
```

```
sum( p in ProductionCells, f in TimeFrames, a1 in Type, a2 in Type )
```

```
(SetupE[p][f][a1][a2] * SetupTime[a1][a2]);
```

```
subject to {
```

```
forall( a in Type, c1 in Package )
```

```
ctDemand:
```

```
sum( p in ProductionCells, f in TimeFrames, c2 in Package )
```

```
((ProductionTime[p][f][a][c1][c2] +
```

```
ProductionTime[p][f][a][c2][c1]) * Capacity[p][a][c1][c2]) >= Demand[a][c1];
```

```
forall( p in ProductionCells, f in TimeFrames, a in Type, c1 in Package, c2 in Package )
```

```
ctDoing:
```

```

    ProductionTime[p][f][a][c1][c2] <= M * Doing[p][f][a][c1][c2];

forall( p in ProductionCells, f in TimeFrames )

ctDoingorSetup:

    sum( a in Type, c1 in Package, c2 in Package )

        (Doing[p][f][a][c1][c2]) +

        sum( a1 in Type, a2 in Type )

            (SetupE[p][f][a1][a2] )

    <= 1;

forall( p in ProductionCells, f in TimeFrames2, a in Type )

ctDoingAfterSetup:

    sum( c1 in Package, c2 in Package )

        (Doing[p][f][a][c1][c2]) <=

    sum(a2 in Type )

        (SetupE[p][f-1][a2][a] );

forall( p in ProductionCells, a in Type)

ctInitializeSetup:

    sum( a2 in Type )

        (SetupE[p][1][a][a2]) <=

    sum(c1 in Package, c2 in Package )

        (SetupZero[p][a][c1][c2]);

forall( p in ProductionCells, a in Type, c1 in Package, c2 in Package)

ctInitializeDoing:

```

```

    Doing[p][1][a][c1][c2] <=

    SetupZero[p][a][c1][c2];

forall( p in ProductionCells, f in TimeFrames2, a in Type )

ctSetupAfterDoing:

sum( a2 in Type )

    (SetupE[p][f][a][a2]) <=

sum(c1 in Package, c2 in Package )

    (Doing[p][f-1][a][c1][c2]);

forall( p in ProductionCells )

ctTime:

sum( f in TimeFrames, a in Type, c1 in Package, c2 in Package )

    (ProductionTime[p][f][a][c1][c2]) +

sum( f in TimeFrames, a1 in Type, a2 in Type )

    (SetupE[p][f][a1][a2] * SetupTime[a1][a2]) <= TotalTime[p];

};

```