



**FACULDADE DE ENGENHARIA DE GUARATINGUETÁ**

JOSÉ LUIZ ELISEI

SEQUENCIAMENTO DE LOTES EM PRENSAS  
DE ALTA CAPACIDADE COM TEMPO DE  
*SETUP* DEPENDENTE DA SEQUÊNCIA

Guaratinguetá  
2012

JOSÉ LUIZ ELISEI

SEQUENCIAMENTO DE LOTES EM PRENSAS  
DE ALTA CAPACIDADE COM TEMPO DE  
*SETUP* DEPENDENTE DA SEQUÊNCIA

Dissertação apresentada à Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, para a obtenção do título de Mestre em Engenharia Mecânica na área de Projetos e Materiais.

Orientador: Prof. Dr. Edson Luiz França Senne

Guaratinguetá  
2012

E43s	<p data-bbox="349 1462 564 1496">Elisei, José Luiz</p> <p data-bbox="349 1496 1173 1608">Sequenciamento de lotes em prensas de alta capacidade com tempo de setup dependente da sequência / José Luiz Elisei. – Guaratinguetá : [s.n.], 2012</p> <p data-bbox="397 1608 523 1641">111 f. : il.</p> <p data-bbox="397 1641 683 1675">Bibliografia: f. 82-87</p> <p data-bbox="349 1720 1173 1792">Dissertação (mestrado) – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2012</p> <p data-bbox="397 1792 1005 1825">Orientador: Prof. Dr. Edson Luiz França Senne</p> <p data-bbox="397 1870 903 1904">1. Otimização combinatória I. Título</p> <p data-bbox="986 1937 1173 1971">CDU 519.863</p>
------	--

*JOSÉ LUIZ ELISEI*

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE  
"MESTRE EM CIÊNCIAS"

PROGRAMA: ENGENHARIA MECÂNICA  
ÁREA: GESTÃO E OTIMIZAÇÃO

APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO

  
Prof. Dr. Marcelo dos Santos Pereira  
Coordenador

*BANCA EXAMINADORA:*

  
Prof. Dr. EDSON LUIZ FRANÇA SENNE  
Orientador / Unesp-Feg

  
Prof. Dr. MARCOS ANTONIO PEREIRA  
Unesp-Feg

  
Prof. Dr. ANTONIO AUGUSTO CHAVES  
UNIFESP-SJC

*Fevereiro de 2012*

## DEDICATÓRIA

Dedico a Cristina, André, Raquel, Ângelo, Luiz e Hilda, que deram sentido à minha vida, fazendo de mim algo melhor.

## **AGRADECIMENTOS**

No mundo dependemos de todos. Nossos esforços são somados à energia divina, viva em cada ser com que nos deparamos e essa energia se materializa, por isso agradecer deve ser um exercício constante, a cada momento de nossa existência.

Agradeço ao meu amigo e orientador prof. Senne, que tem em seu comportamento a lição maior a nos ensinar. Aos professores Marins, Delamaro, Jorge, Messias, Cassilda, Salomon, Marcos Pereira e Galeno por seus esforços, que tanto fizeram por esse programa. Aos funcionários da FEG que sempre me atenderam com boa vontade. A meu chefe e amigo Rodolfo que compreendeu e me deu o espaço necessário para continuar. Ao aluno Viktor Otto Schwarzmeier, pela implementação na linguagem C do algoritmo heurístico proposto neste trabalho. Agradeço a meus pais, que deram sua vida por nós. E finalmente à minha família, que pagaram alto custo nessa jornada, mas me entenderam e apoiaram em todos os momentos.

ELISEI, J.L. Sequenciamento de lotes em prensas de alta capacidade com tempo de setup dependente da sequência. 2012. 111 f. Dissertação (Mestrado em Engenharia Mecânica) – Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2012

## RESUMO

O presente trabalho é fruto da observação de um problema real encontrado em uma indústria de autopeças, que produz peças estampadas em aço para caminhões, automóveis e tratores, utilizando prensas de alta capacidade. As prensas utilizadas necessitam de um ferramental, que precisa ser montado na prensa antes de começar a produção. Devido a esse fato, o *setup* de uma prensa pode variar de acordo com a sequência de produção que for realizada. Além disso, como o ferramental é único, quando uma peça está sendo produzida em uma prensa, outra peça que utilize o mesmo ferramental não poderá ser produzida em qualquer outra prensa. Neste trabalho procurou-se resolver o problema de programação da produção para esta indústria, que caracteriza-se como um problema de sequenciamento com máquinas paralelas e com tempo de *setup* dependente da sequência de produção. Para resolver tal problema, foram formulados alguns modelos matemáticos para obtenção de soluções exatas. No entanto, como trata-se de um problema de Otimização Combinatória NP-difícil, foi desenvolvido também um método heurístico híbrido utilizando as técnicas VND (*Variable Neighborhood Descent*) e ILS (*Iterated Local Search*) para a obtenção de soluções para grandes exemplares do problema em um tempo computacional razoável.

**Palavras-chave:** Otimização Combinatória, Problema de sequenciamento, *Setup* dependente da sequência, Formulação matemática, Método heurístico híbrido.

ELISEI, J.L. Scheduling batches in high capacity presses with setup time sequence dependent. 2012. 111 f. Dissertation (Master in Mechanical Engineering) – Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2012

## ABSTRACT

The present work is based on a real world problem found in the auto parts industry, which produces steel stamped parts for trucks, cars and tractors, using high-capacity presses. The presses used need a tooling that has to be mounted on the press before production begins. Because of this, the setup of a press can vary according to the sequence of production that is performed. Moreover, as the tooling is unique for each type of auto part, when an auto part is being produced on a press, another auto part of the same type can not be produced in any other press. In this work one tried to solve the problem of production scheduling for a specific plant, which is characterized as a scheduling problem with parallel machines and sequence-dependent setup times. To solve this problem, some mathematical models were formulated to obtain exact solutions. However, as the problem is a NP-hard Combinatorial Optimization problem, a hybrid heuristic method was also developed, using the techniques VND (Variable Neighborhood Descent) and ILS (Iterated Local Search) to obtain approximate solutions for large problem instances in a reasonable execution time.

**Keywords:** Combinatorial Optimization, Scheduling problem, Sequence-dependent setup, Mathematical formulation, Hybrid heuristic method.



## LISTA DE FIGURAS

Figura 1 - Ambientes de máquina.....	24
Figura 2 - Gestão da produção da empresa em estudo.....	40
Figura 3 - Sequenciamentos obtidos com o Modelo III.....	67
Figura 4 - Comportamento de heurística de refinamento (Fonte: Souza, 2008)....	69
Figura 5 - Heurística VND.....	70
Figura 6 - Algoritmo ILS.....	71
Figura 7 - Processo de perturbação (Fonte: Souza, 2008).....	71
Figura 8 - Algoritmo proposto.....	72
Figura 9 – Representação de solução .....	72
Figura 10 – Representação de solução viável .....	73
Figura 11 - Criação da solução inicial.....	73
Figura 12 - Procedimento VND.....	74
Figura 13 - Metaheurística ILS.....	75
Figura 14 - Sequenciamentos obtidos com o algoritmo heurístico.....	77

## LISTA DE TABELAS

Tabela 1 - Base de dados para o Teste 1.....	46
Tabela 2 - Tempos de <i>setup</i> de uma peça para outra para o Teste 1.....	46
Tabela 3 - Sequenciamento obtido no Teste 1.....	46
Tabela 4 - Resultados obtidos no Teste 1.....	47
Tabela 5 - Sequenciamento obtido no Teste 2.....	49
Tabela 6 - Resultados obtidos no Teste 2.....	49
Tabela 7 - Sequenciamento obtido no Teste 3.....	50
Tabela 8 - Resultados obtidos no Teste 3.....	51
Tabela 9 - Sequenciamento obtido no Teste 4.....	51
Tabela 10 - Resultados obtidos no Teste 4.....	52
Tabela 11 - Resultados obtidos no Teste 5.....	53
Tabela 12 - Parâmetros das peças.....	54
Tabela 13 - Tempo de <i>setup</i> .....	54
Tabela 14 - Incompatibilidade entre peças.....	54
Tabela 15 - Sequenciamento obtido no Teste 5.....	55
Tabela 16 - Resultados obtidos no Teste 5.....	55
Tabela 17 - Comparativo Xpress <i>versus</i> CPLEX.....	56
Tabela 18 - Comparação das características dos modelos.....	57
Tabela 19 - Conversão de variáveis.....	58
Tabela 20 - <i>Setup</i> de uma peça para outra.....	62
Tabela 21 - Demanda por peça.....	62
Tabela 22 - Incompatibilidade entre peças.....	62

Tabela 23 - <i>Setup</i> inicial por peça.....	62
Tabela 24 - Resultados obtidos com os Modelos I e II.....	62
Tabela 25 - Sequenciamentos obtidos com os Modelos I e II.....	63
Tabela 26 – Resultados dos novos testes computacionais.....	63
Tabela 27 - Resultados obtidos com o Modelo III - Solver Xpress .....	66
Tabela 28 - Resultados preliminares obtidos com o algoritmo heurístico.....	76
Tabela 29 - Resultados obtidos com o algoritmo heurístico.....	78

## SUMÁRIO

1.	INTRODUÇÃO.....	13
1.1	Justificativa.....	14
1.2	Determinação dos objetivos.....	15
1.2.1	Objetivo geral.....	15
1.2.2	Objetivos específicos.....	15
1.2.3	Classificação da pesquisa.....	15
1.2.4	Estruturação do trabalho.....	16
2.	REVISÃO BIBLIOGRÁFICA.....	18
2.1	Gestão de produção.....	18
2.1.1	EDI – <i>Electronic Data Interchange</i> .....	18
2.1.2	MRP – <i>Material Requeriment Planning</i> .....	19
2.1.3	JIT – <i>Just in Time</i> .....	20
2.2	Problemas de produção.....	21
2.2.1	Sequenciamento de produção.....	21
2.2.2	Modelos de sequenciamento de produção.....	23
2.2.3	Ambiente de máquina.....	23
2.2.4	Tempo de preparação de máquina.....	25
2.2.5	Horizonte de planejamento.....	25
2.2.6	Número de produtos.....	26
2.2.7	Restrição por capacidade.....	26
2.2.8	Deterioração de um produto.....	26
2.2.9	Demanda.....	26
2.2.10	Falta de estoque.....	27
2.3	Problemas de dimensionamento de lotes .....	27
2.4	Estudos sobre dimensionamento de lote e sequenciamento.....	28
2.5	Problemas de sequenciamento de lotes.....	30
3.	MODELOS MATEMÁTICOS PROPOSTOS.....	38
3.1	Descrição do problema.....	38

3.2	O processo produtivo.....	38
3.3	A Gestão da produção na empresa em estudo.....	39
3.4	Os modelos matemáticos .....	42
3.5	Modelo I: minimização do tempo total de <i>setup</i> .....	43
3.5.1	Testes computacionais com o Modelo I.....	46
3.6	Modelo II: minimização do tempo total de <i>setup</i> .....	57
3.7	Modelo III: minimização do <i>makespan</i> .....	65
4.	HEURÍSTICA PROPOSTA.....	68
4.1	O algoritmo heurístico híbrido proposto.....	72
4.2	Testes computacionais.....	75
5.	CONCLUSÃO.....	79
	Referências Bibliográficas.....	82
	Apêndice A: Modelo I formulado na linguagem Mosel.....	87
	Apêndice B: Alterações implementadas no Modelo I.....	91
	Apêndice C: Modelo II formulado na linguagem Mosel.....	93
	Apêndice D: Modelo III formulado na linguagem Mosel.....	97
	Apêndice E: Dados utilizados nos testes computacionais do Modelo III.....	102
	Apêndice F: Resultados para um exemplar real do problema.....	106

## 1. INTRODUÇÃO

O crescimento que vem ocorrendo nos últimos anos nas indústrias montadoras de automóveis, caminhões e tratores tem seu reflexo nas indústrias de autopeças. Esta, conforme *site* do Sindicato das Indústrias de Autopeças-Sindipeças, teve em 2009 um crescimento em seu faturamento de 11,8% sobre o de igual período de 2007. O principal segmento de mercado responsável por isso foram as montadoras, com participação de 70% nas vendas e o volume de faturamento em 2009 na casa dos 6 bilhões de reais.

Dentro do cenário de globalização, a demanda de insumos exige tanto qualidade quanto prazo de entrega, atendendo às mais modernas práticas internacionais. No campo da gestão da produção, a metodologia JIT (*Just-in-Time*) vem evoluindo para o chamado JIS (*Just-in-Sequence*), sistema de abastecimento, capaz de atender não só os itens necessários, na quantidade necessária e no momento determinado, mas também na sequência certa (TROQUE e PIRES, 2003). Este tipo de prática faz com que o planejamento de produção tenha que ser ágil na geração e na revisão da sequência de produção.

Neste trabalho considera-se um problema prático observado em uma fornecedora de peças estampadas para a indústria automobilística. A base da estrutura de um caminhão é composta de chassi e travessas, em que todo o caminhão é montado. Estas peças são feitas de chapas de aço estampadas ou dobradas, dependendo do formato da peça. Se a peça tiver maior complexidade em seu formato com curvas, esta deve ser feita em prensas; se ela não tiver grande complexidade poderá ser feita em máquinas que contorcem o metal. Aqui considera-se somente os processos em estamparia nos quais as prensas que realizam esse processo são de alta capacidade, variando de 3000 a 5200 toneladas. Neste trabalho são propostos modelos de Programação Matemática que exprimem a realidade deste processo de planejamento de produção. Apesar de estar analisando uma empresa em particular, atende-se a problemas vividos por inúmeras empresas, mesmo de outros setores.

## 1.1 - Justificativa

O ambiente aqui descrito, máquinas paralelas, sequenciamento de produção e *setup* de máquina, ocorre em uma quantidade muito grande de empresas. A solução de problemas com essa característica tem sido estudada ao longo dos anos e diversos avanços vem sendo observados, tanto no campo da programação heurística quanto na melhora dos *softwares* de mercado como Xpress, Gams, CPLEX etc.

Existe uma grande influência do processo em estudo no restante da empresa. Com este processo pode-se relacionar:

- Como realizado atualmente, o trabalho de programar a produção leva muito tempo e não existe a possibilidade de reprogramação, caso haja necessidade. Hoje em dia as solicitações dos clientes por alterações emergenciais ocorrem em grande quantidade. Na própria empresa a necessidade ocorre quando falta matéria prima ou quebra de algum equipamento que recai sobre esta célula de produção.
- Necessidade de se ter um aproveitamento maior dos recursos. Investir em novos equipamentos e ampliação de uma linha de produção é muito caro. Havendo a possibilidade de um melhor aproveitamento dos recursos atuais, isso deve ser feito.
- O cálculo atualmente é feito em planilhas, o que não assegura que a solução obtida é a melhor possível.

## 1.2 - Determinação dos objetivos

### 1.2.1 - Objetivo geral

Desenvolver um algoritmo que proporcione ganhos reais na programação de prensas, buscando assim, a aplicação de metodologias de Pesquisa Operacional a problemas combinatórios de interesse prático.

### 1.2.2 - Objetivos específicos

Os objetivos específicos deste trabalho são os seguintes:

- Evidenciar o potencial da modelagem matemática para resolver problemas comumente encontrados em processos industriais.
- Desenvolver e validar modelos matemáticos, comparando-os a outros obtidos em trabalhos já publicados.
- Desenvolver um método heurístico híbrido para obter uma boa solução do problema em estudo.
- Analisar comparativamente as soluções exatas obtidas pelas formulações matemáticas e as boas soluções obtidas pelo método heurístico proposto.

### 1.2.3 - Classificação da pesquisa

Segundo Gil (1999), a pesquisa tem um caráter pragmático, é um processo sistemático de desenvolvimento do método científico e seu objetivo é descobrir respostas para problemas fazendo uso de procedimentos científicos. Existem várias classificações para as pesquisas. A classificação dessa pesquisa é a seguinte:

- **Pesquisa Aplicada**, pois objetiva gerar conhecimentos para aplicação prática dirigida a solução de problemas específicos. Envolve verdades e interesses locais;



- **Quantitativa**, pois considera-se que tudo pode ser quantificado, o que significa traduzir, em números, opiniões e informações para classificá-las e analisá-las;
- **Exploratória**, pois visa proporcionar maior familiaridade com o problema com vistas a torná-lo explícito ou a construir hipóteses. Envolve levantamento bibliográfico; entrevista com pessoas que tiveram experiências práticas com o problema pesquisado; análise de exemplos que estimulem a compreensão.

#### 1.2.4 - Estruturação do trabalho

Este trabalho está organizado em 5 capítulos. O Capítulo 1 apresentou o contexto em que o trabalho se insere, a justificativa de porque abordar o problema em estudo, os objetivos do trabalho e o método de pesquisa utilizado.

O Capítulo 2 apresenta a revisão bibliográfica, tendo como objetivo reunir as informações fundamentais para o entendimento dos conceitos aqui observados. Estes conceitos são, basicamente, a otimização no sequenciamento de produção e a gestão de produção, e visam caracterizar a relevância do problema no contexto da administração de produção.

No Capítulo 3 é apresentado o desenvolvimento de um modelo matemático cujo objetivo é a minimização de todos os tempos de produção. Este modelo é testado em dois *solvers*: Xpress e CPLEX. Seus resultados são comparados para efeito de análise de desempenho e resultado. Após este teste o modelo criado é comparado a um outro, adaptado de uma solução já publicada. Na segunda parte deste capítulo, o modelo original é alterado para controlar o *makespan* e depois comparado a uma solução desenvolvida com base nas metaheurísticas ILS e VND.

No Capítulo 4 apresenta-se um algoritmo heurístico híbrido para o problema estudado. Nesta etapa contou-se com a colaboração de um bolsista de Iniciação Científica que implementou o algoritmo proposto com base nos dados apresentados neste trabalho.

No Capítulo 5 são apresentadas a conclusão final do trabalho e algumas propostas para estudos futuros.

## **2. REVISÃO BIBLIOGRÁFICA**

### **2.1 - Gestão de produção**

A gestão de produção é a principal função da empresa. Segundo Slack *et al.* (2002), ela consiste em buscar o objetivo principal da empresa, a razão pela qual ela existe. Ela é responsável por toda a cadeia produtiva, do fornecedor até a entrega ao cliente. Uma das etapas dessa gestão é o Planejamento e Controle da Produção (PCP). O PCP é o responsável pelo atendimento às demandas de longo prazo, que denota compras de matéria prima, e curto prazo, que é relativo à preparação de máquinas e alocações de mão de obra.

Em apoio ao PCP duas ferramentas se destacam, o MRP (*Material Requirement Planning*) e o EDI (*Electronic Data Interchange*) para planejamento de compras e programação de produção, respectivamente. A origem dos dados ocorre no cliente que envia, através de processos EDI, sua necessidade de consumo e sua programação de entrega. Esta por sua vez vai alimentar o MRP que gera ordens de compra de matéria prima e ordens de produção.

#### **2.1.1 - EDI – *Electronic Data Interchange***

O EDI é uma modalidade de troca de informações entre empresas atendendo a formatos padrões previamente acordados entre as partes (GOMES e RIBEIRO, 2004). O EDI foi adotado primeiramente nos Estados Unidos dentro das lojas de varejo e de empresas de transporte e, mais tarde, passou para as indústrias de automóveis e autopeças. Ao invés de emitir documentos em papel, a empresa pode transmiti-los eletronicamente direto de seu computador para o computador do fornecedor. Este tipo de processo aumenta em muito a produtividade devido a fatores como:

- Não há necessidade de manuseio por nenhum usuário, conseqüentemente a informação que sai de um sistema alimenta o outro sem risco de erro de digitação;
- Rapidez no processo. Uma vez automatizado o processo, não existe mais demora, pois todo ele transcorre via redes e sistemas integrados;
- Registro confiável de toda a informação. Não há a possibilidade de que a informação seja perdida.

Os processos de EDI mais usados pelas montadoras de automóveis e seus fornecedores são:

- Programação de entrega – Dá uma visão completa de longo prazo das necessidades do cliente;
- Programação diária – Visão de curto prazo das necessidades do cliente. Nesta modalidade se encaixam operações como JIT (*Just in Time*) e JIS (*Just in Sequence*).
- Aviso de embarque – Documento detalhado sobre a mercadoria expedida. Semelhante a uma nota fiscal eletrônica. Serve para agilizar a entrada da mercadoria na empresa receptora.

### **2.1.2 - MRP – *Material Requeriment Planning***

Conforme Slack *et al.* (2002), o MRP (*Material Requeriment Planning*) é uma ferramenta para cálculo das necessidades de matéria prima. Sua base de informação são as solicitações dos clientes e as previsões de vendas. O sistema verifica a lista de materiais necessários à produção e leva em consideração o saldo dos estoques para gerar a lista de faltas. Em paralelo a todo este processo são considerados os tempos de reabastecimento de matéria prima, perdas em produção e compras já realizadas.

Sua lógica é baseada no programa mestre de produção, na lista de materiais e na quantidade em estoque. O programa mestre de produção (MPS – *Master*

*Production Scheduling*) consiste na definição das quantidades de cada produto final que se deseja produzir em cada período (“*time bucket*”) dentro de um horizonte de planejamento EDI. A lista de materiais está baseada na estrutura dos componentes do produto final.

No princípio da concepção, nos anos 1970, a utilização do MRP encontrava bastante dificuldade com relação ao custo de *software* e *hardware*, mas com o tempo, a superação desses problemas levou à necessidade de integrar a ele o controle sobre a capacidade de produção, o que implica em considerar roteiros de produção (sequência e tempos das diferentes tarefas das ordens de produção) e um cadastro dos centros de produção com as respectivas capacidades. Desta forma tornou-se possível verificar a capacidade dos centros ao longo do tempo e conseqüentemente a viabilidade dos programas de produção. Este novo modelo foi chamado de MRP II – *Manufacturing Resources Planning*.

Uma das dificuldades do MRP está em trabalhar com “janelas de tempo” (*time bucket*) diferentes dos modelos de simulação por evento. Isto faz com que ele não consiga representar, em detalhes, as sequências das operações da fábrica. Os modelos matemáticos usando técnicas de sequenciamento e programação de produção obtêm bons resultados nesta área, mas não se encontra no mercado a integração destes com o MRP.

### **2.1.3 - JIT – *Just in Time***

O JIT (*Just in Time*) surgiu logo após a segunda Guerra Mundial nas indústrias japonesas (CERRA e BONADIO, 2000). O objetivo dessa ferramenta é gerenciar o fluxo de peças (a peça necessária, na quantidade necessária, no instante e local necessário). Algumas características naturais do JIT fazem com que a pressão aumente no fornecedor de matéria prima. Uma delas é a produção puxada, o cliente diz quando quer uma mercadoria; outra é uma consequência da produção enxuta, que é o baixo estoque.

## 2.2 - Problemas de produção

Dentro de uma empresa, a tomada de decisão sobre problemas de produção se dá em três possíveis níveis: operacional, tático e estratégico (ARAÚJO e ARENALES, 2000; LAUDON e LAUDON, 2004). O nível estratégico é o mais alto na hierarquia e está ligado às decisões de longo prazo, metas, escolhas de produtos a produzir etc.; o nível tático, o segundo na hierarquia, tem como foco a visão de poucas semanas e a tomada de decisão de nível médio; e o nível operacional trabalha diretamente na linha de produção e seu período é de poucos dias. Um processo como o sequenciamento de produção e dimensionamento de lotes pode ser considerado entre os níveis operacional e tático.

É importante para um melhor entendimento do planejamento e da programação de produção, o esclarecimento de uma série de termos e conceitos conforme faz-se a seguir.

### 2.2.1 Sequenciamento de produção

Conforme Davis (1990) e Absi e Kedad-Sidhoum (2008), o processo de sequenciamento vem sendo estudado desde os anos 50. Sua aplicabilidade abrange desde transporte até produção de bens. Seu objetivo é a alocação de recursos ou atividades dentro de um período de tempo.

É importante esclarecer que a palavra "sequenciamento" está sendo usada neste trabalho em referência à determinação da sequência e tempo inicial quando um processo é realizado por uma máquina. Em inglês este termo é denominado *scheduling*, diferente de *sequence*, que é somente a sequência de processamento.

Os primeiros trabalhos de sequenciamento de produção, em sua maioria, eram voltados a sistemas baseados em máquinas simples. Alguns desses trabalhos

derivaram da regra da data de vencimento mais antiga (*EDD - Early Due Data*), na qual as tarefas (na literatura inglesa referenciadas como *jobs*) são sequenciadas na ordem cronológica da data de vencimento mais antiga para a mais nova. Outros derivaram da regra do tempo de processamento de menor peso (*SWPT - Short Weighted Processing Time*), na qual as tarefas são sequenciadas na ordem crescente de tempo de processamento (POTTS e KOVALYOV, 2000).

Conforme Stoop e Wiers (1996), o planejamento de chão de fábrica costuma desviar-se do esperado devido a fatores imprevisíveis como:

- Quebra de máquina;
- Falta de material;
- Valores inexatos do tempo de produção.

Quanto à quebra de máquina, o problema pode ter sua incidência reduzida por meio da manutenção preventiva. Outros fatores, como os relacionados à gerência de pessoas como férias, licença maternidade, podem ser controladas, mas acidentes e doenças não há como. O segundo tipo de problema refere-se a ordens emergenciais, atrasos na entrega pelo fornecedor, problemas externos impossíveis de se prever. O terceiro problema ocorre, entre outras causas, devido à especialização de um operador, mudança de produto, falta de avaliação de novo modelo.

As pesquisas sobre problemas de sequenciamento reconhecem que o problema é complexo e classificam o problema como NP-difícil (*NP-hard*). A classificação de um problema como NP-difícil sugere que sua solução exige algoritmos, cujos tempos de execução crescem exponencialmente com o tamanho do problema, como por exemplo, os algoritmos *branch-and-bound*. Assim, em alguns casos, como em exemplares pequenos, esses algoritmos conseguem resolver os problemas de forma ótima. No entanto, na prática, os exemplares do problema são de grandes dimensões e deve-se aceitar o fato de se ter apenas soluções aproximadas para o problema.

### 2.2.2 - Modelos de sequenciamento de produção

Um problema de sequenciamento pode ser definido como a forma de processar  $J$  tarefas em  $M$  máquinas. Uma solução para um problema de sequenciamento deve definir para cada máquina  $m$  e para cada tarefa  $j$ , os instantes de tempo em que a máquina  $m$  processa a tarefa  $j$ . Admite-se em problemas de sequenciamento que uma tarefa não pode ser executada em duas máquinas ao mesmo tempo, e que uma máquina não pode executar duas tarefas ao mesmo tempo. Além destas restrições gerais, para cada problema específico a solução deve satisfazer a uma série de restrições relativas ao problema. O tipo de problema é especificado pelo ambiente da máquina, as características da tarefa e o critério de otimalidade.

### 2.2.3 - Ambiente de máquina

Com relação ao ambiente de máquina pode-se considerar que os ambientes são de estágio simples ou de multi-nível. A produção em estágio simples requer uma operação para cada tarefa, ao passo que na produção em ambiente multi-nível há tarefas que requerem operações em diferentes máquinas. O termo máquina pode ser substituído por centro de trabalho, unidade produtiva etc.

A produção em estágio simples envolve ambientes com uma ou mais máquinas operando paralelamente. Para máquinas paralelas podem ser consideradas as seguintes características (MULLER *et al.*, 2002):

- **máquinas paralelas idênticas**, quando existe um conjunto único contendo os tempos de execução das tarefas e esses tempos de execução permanecem constantes não importando para qual máquina a tarefa é atribuída;
- **máquinas paralelas uniformes**, quando existe um conjunto único contendo os tempos de execução das tarefas, mas os tempos de execução são alterados por um único fator uniforme dependendo da tarefa ser executada em uma máquina ou em outra;



- **máquinas paralelas não relacionadas**, quando tem-se uma série de máquinas que diferem entre si na capacidade e no tempo, mas que, com alguma alteração podem realizar o mesmo trabalho.

Considera-se como estágio, ou níveis dentro de uma linha de produção, a etapa pela qual um produto tem que passar. Diz-se que um produto é de estágio simples (*single-level*) quando este produto passa somente por uma etapa para ficar pronto. Quando um produto passa por vários níveis para ficar pronto, diz-se tratar de um produto multi-estágio (*multi-level*). Problemas multi-estágios são bem mais complexos do que os de nível simples.

Segundo Arenales *et al.* (2007), um tipo de ambiente multi-estágio é conhecido como *job shop*, em que as máquinas estão dispostas em vários níveis e existe uma sequência pré-determinada de estágio a ser obedecida para a produção de um produto. Outra modalidade de ambiente multi-estágio é o *flow shop*, em que uma peça deve passar por várias máquinas, mas sem a necessidade de obedecer a uma sequência pré-estabelecida.

A Figura 1 apresenta uma classificação de ambientes de máquina.

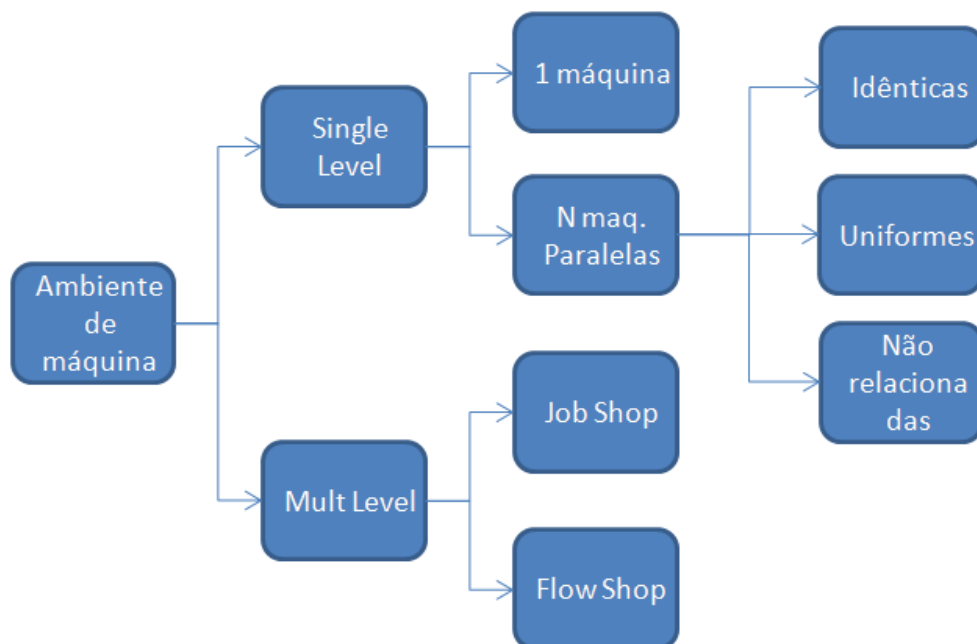


Figura 1 – Ambientes de máquina

#### 2.2.4 - Tempo de preparação de máquina

Conforme Dauzère-Peres e Lasserre (2002), *setup* é o termo utilizado para designar o trabalho de preparar uma máquina ou processo para a produção. Isto inclui obter ferramentas, posicionar o material para o processamento, limpeza, inspeção etc. O *setup* foi considerado por algum tempo e para alguns autores como insignificante ou parte do tempo de processo. Apesar disso ser justificado para alguns problemas, em outros ele precisa ser considerado.

O *setup* pode ser para um processamento em lote (*batch*) ou não (*non-batch*). O tempo de *setup* por lote ocorre quando as tarefas são produzidas em lote e um *setup* precede cada um desses lotes. Conforme notação de Dauzère-Peres e Lasserre (2002), neste caso, as tarefas são particionadas em  $F$  famílias, com  $F > 1$ .

Segundo Allahverdi *et al.* (1999), um *setup* é dependente da sequência se sua duração (ou custo) depende tanto da tarefa corrente como da tarefa que a precede imediatamente. O *setup* é independente da sequência se sua duração (custo) depende somente da tarefa em execução. Allahverdi *et al.* (1999) destacam a importância da análise do tempo de *setup* dependente de sequência quando a ocupação da máquina está perto de sua capacidade total e a importância de se considerar o *setup* para um efetivo gerenciamento dos estoques e da capacidade produtiva, pois o controle do *setup* tem forte ação sobre a redução dos níveis de estoques.

#### 2.2.5 - Horizonte de planejamento

Para problemas de planejamento de produção o horizonte de planejamento é sempre finito e a demanda associada a ele pode ser dinâmica (pode se alterar no horizonte de planejamento) ou estática. Além disso, o horizonte de planejamento pode ser do tipo *big-bucket*, onde o período de tempo é grande o bastante para produzir vários itens, ou do tipo *small-bucket*, onde o período de tempo é tão pequeno, que somente um item pode vir a ser produzido. Também para o horizonte de planejamento,

um outro dado importante é a certeza ou não da data de entrega de um produto. Quando esta data não é certa, tem-se um horizonte rolante (*rolling horizon*).

### **2.2.6 - Número de produtos**

Uma linha de produção pode ser caracterizada como de produção de um único produto (*single-item*) ou de vários produtos (*multi-item*). A possibilidade de produção de vários produtos aumenta em muito a complexidade do problema, dada a possibilidade de variação dos tempos de produção, *setups* etc.

### **2.2.7 - Restrição por capacidade**

Um recurso necessário à produção pode ter uma capacidade limitada. Esse limite de capacidade aumenta em muito a complexidade do problema. Na literatura um recurso sem limite de capacidade é conhecido como não-capacitado (*uncapacited*) e um recurso com limite de capacidade é conhecido como capacitado (*capacitated*). O limite de capacidade pode ser o tempo limite de produção, uma restrição de mercado, o espaço de armazenagem, o tempo disponível de uma máquina, etc.

### **2.2.8 - Deterioração de um produto**

O tempo em que um produto pode ser mantido nos estoques é outro fator que deve ser levado em consideração no planejamento da produção. Alguns produtos têm validade de uso, ou alguma particularidade que pode vir a afetar sua qualidade final (ferrugem, deformação etc.). Este problema é pouco estudado pela literatura em geral e torna o problema bastante complexo.

### **2.2.9 - Demanda**

A demanda de um produto é uma importante entrada de dados para a solução de um problema de sequenciamento de produção (e de dimensionamento de lotes).

Atualmente, processos como o JIT (*Just In Time*), o JIS (*Just In Sequence*) e a venda sob uma modalidade em que o próprio cliente monta seu produto, fazem com que a demanda sofra alterações abruptas e tanto a sequência de produção quanto as quantidades a serem produzidas podem ser alteradas. Quando uma demanda não se altera com o passar do tempo é denominada estática; a demanda é dinâmica se esta é alterada o tempo todo. Se o valor da demanda é conhecido antecipadamente esta é denominada determinística; se a demanda é baseada em cálculos estatísticos ela é dita probabilística. Um produto pode depender de outro para sua produção e, neste caso, este produto é chamado de demanda dependente. Se um produto não depende de nenhum outro para sua produção, sua demanda é independente.

#### **2.2.10 - Falta de estoque**

A falta ou não de estoque pode ser uma estratégia dentro do negócio. Na literatura a falta de estoques (*shortage*) é encontrada em vários casos, pois aceita-se a entrega futura ou pode-se perder a venda. O atraso de produção (*backlogging*) combinado com a perda de venda é possível conforme Wee (1999) *apud* Karimi *et al.* (2003). Problemas com falta de estoques são mais difíceis de resolver.

### **2.3 - Problemas de dimensionamento de lotes**

O planejamento de produção envolve o dimensionamento de lotes de peças a ser produzido. Para este problema, as principais variáveis são: horizonte de produção (que pode ser finito ou infinito) e a demanda de peças a produzir.

Em Campbel (1992), os processos de sequenciamento e dimensionamento de lotes são considerados como níveis de decisão distintos devido às suas características tão diferentes. Além disso, o sequenciamento e dimensionamento de lotes são considerados dois dos mais importantes processos dentro da cadeia de suprimentos.

O dimensionamento de lotes, conforme Araújo e Arenales (2000), consiste em planejar a quantidade de itens a serem produzidos em várias ou em uma única máquina ao longo de um tempo finito, de modo a atender a uma certa demanda, sujeito a restrições de limitação de capacidade, tendo como objetivo a minimização do custo de produção ou a maximização da quantidade produzida.

Conforme Feng e Cheng (1998), com o aumento da complexidade das técnicas de produção industrial, o dimensionamento de lotes de produção torna-se cada vez mais difícil. Para estes autores, os seguintes fatores são importantes:

- Disponibilidade limitada de recursos;
- A existência de vários produtos usando o mesmo recurso;
- Demanda variável período a período no horizonte de planejamento;
- Tempos de preparação;
- Custos de preparação.

#### **2.4 - Estudos sobre dimensionamento de lote e sequenciamento**

O dimensionamento de lotes pode ser dividido em mono-estágio ou multi-estágio (ARAÚJO e ARENALES, 2000). Mono-estágio é a designação dada quando um item não depende da produção ou processamento de nenhum outro anterior a ele para começar sua produção. Multi-estágio ocorre quando um item depende da produção de outro para sua fabricação.

Estudos sobre dimensionamento de lotes são antigos. Keachie e Fontana (1966) descrevem a importância de analisar o dimensionamento de lotes. Wortham e Mayyasi (1972) destacam o modelo de horizonte finito, falta de estoques (*shortage*) e ordens não atendidas (*backorder*).

Segundo Meyer e Fleischmann (1997), um modelo realista de dimensionamento de lote deve ter capacidade de produção finita, vários produtos usando essa capacidade

e uma situação de planejamento dinâmica, que parte do estoque existente em um intervalo de tempo finito com demanda flutuante.

Araújo e Arenales (2000), baseado em um modelo de Trigueiro *et al.* (1989), considerou o problema de dimensionamento de lotes mono-estágio com itens, restrição de capacidade, custos de produção, preparação e estoque. Para este modelo foi incluído o tempo de preparação para considerar o consumo dos recursos.

Buschkühl e Tempelmeir (2008), desenvolveu para uma indústria de plásticos fornecedora da indústria automobilística, um modelo envolvendo preparação e dimensionamento proporcional de lote (PLSP – *Proportional Lot Sizing and Scheduling Problem*), proposto por Haase (1994).

Em Sicilia *et al.* (2008), há uma consideração sobre um limite para o nível de estoques, que foca o problema através do dimensionamento dinâmico de lote. Neste modelo a demanda é conhecida e a falta de estoques não é permitida.

A programação de produção deve designar qual tarefa deve ser executada em qual máquina (designação de tarefas) e quando cada uma das tarefas deve ser executada (*scheduling*). A variedade de ambientes existentes faz com que este assunto seja amplamente discutido. A quantidade de trabalhos a respeito é muito grande.

As principais medidas de desempenho usadas nos problemas de programação da produção de acordo com (ARENALES *et al.*, 2007) e (BARROS e MOCELIN, 2004) são as seguintes:

- *Makespan* – É o instante de término de todas as tarefas a serem processadas. Ele mede a eficiência operacional, informando o tempo necessário para se executar um conjunto de tarefas;
- Tempo total de fluxo - É a soma dos instantes de término de processamento das tarefas, que mede o estoque em processamento.

- Atraso máximo – É a medida de atendimento ao cliente, que consiste na diferença entre o momento em que a tarefa é terminada e aquele em que ela supostamente seria concluída no prazo.
- Atraso total - Soma dos atrasos; está relacionada com uma data de entrega.
- *Lateness* – É a diferença entre o momento que uma tarefa deveria ter sido concluída e o momento em que ela foi concluída com atraso.

Em Behnamian *et al.* (2009), é desenvolvido um algoritmo heurístico para um problema de *setup* dependente da sequência tendo como objetivo a minimização do *makespan*. Este algoritmo faz uso das meta-heurísticas ACO (*Ant Colony Optimization*), VNS (*Variable Neighborhood Search*) e SA (*Simulated Annealing*).

Chen e Chen (2009), desenvolveram um algoritmo para máquinas paralelas não relacionadas e com *setup* dependente da sequência, visando minimizar o número de trabalhos atrasados. Neste algoritmo foram usadas as heurísticas de busca local de descida e busca tabu.

Tahar *et al.* (2006), discutem uma solução para um problema de sequenciamento de um conjunto de *jobs* independentes com *setup* dependente da sequência de produção, objetivando minimizar o *makespan*. A característica mais interessante desse processo é a possibilidade de haver *split* de lotes, isto é, um lote pode ser dividido em tempos e máquinas diferentes. Como exemplo prático desse problema é apontado uma confecção, onde um lote de roupas pode começar a ser produzido em uma máquina e, se necessário, sua produção pode ser interrompida e recomeçar posteriormente.

## 2.5 - Problemas de sequenciamento de lotes

Problemas com *setup* dependente da sequência e máquinas paralelas são analisados a seguir.

Omar *et al.* (2007), realizaram um estudo baseado em uma linha de produção de uma indústria química. O processo químico se caracteriza pela grande quantidade de *setups* dependentes da sequência, pois a contaminação entre produtos dividindo as mesmas máquinas é um fator crítico. O processo caracteriza-se por ter *setup* dependente da sequência, em lotes. Ainda neste trabalho foram construídos modelos tanto para atender à uma máquina simples quanto a máquinas paralelas. Abaixo segue o exemplo da formulação para máquinas paralelas.

Parâmetros:

$m$  = número de famílias;

$r$  = número de linhas de produção;

$n$  = total número de *jobs*;

$f_j$  = família do *job*  $j$ ,  $j = 1, 2, \dots, m$ ;

$d_j$  = data de entrega do *job*  $j$ ;

$p_{jl}$  = tempo de processamento do *job*  $j$  na linha  $l$ ;

$e_j$  = penalidade por adiantar a produção;

$t_j$  = penalidade por atrasar a produção;

$s_{jk}$  = tempo de *setup* da família  $j$  para a família do *job*  $k$ .

$$G_{jk} = \begin{cases} s_{jk} & \text{se } f_j \neq f_k \\ 0 & \text{se } f_j = f_k \end{cases}$$

$M$  = Número positivo muito grande.

Variáveis de decisão:

$E_j$  = Atraso para o *job*  $j$

$T_j$  = Adiantamento para o *job*  $j$

$C_j$  = Tempo total da *job*  $j$



$$\alpha_{jl} = \begin{cases} 1 & \text{se o job } j \text{ é o primeiro a ser processado na linha } l \\ 0 & \text{senão} \end{cases}$$

$$\theta_{jk} = \begin{cases} 1 & \text{se o job } k \text{ é o próximo na sequência após o job } j \\ 0 & \text{senão} \end{cases}$$

$\beta_{jl}$  = Variável contínua, restrita ao intervalo  $[0,1]$ , denota que o *job*  $j$  foi sequenciado na linha  $l$  mas não na primeira posição.

Formulação:

$$\text{Min } \sum_{j=1}^n (e_j E_j + t_j T_j) \quad 2.1$$

Sujeito a:

$$\sum_{l=1}^r (\alpha_{jl} + \beta_{jl}) = 1 \quad j = 1, 2, \dots, n \quad 2.2$$

$$\alpha_{jl} + \beta_{jl} \leq \beta_{kl} + 1 - \theta_{jk} \quad j = 1, 2, \dots, n; k = 1, 2, \dots, n; l = 1, 2, \dots, r \quad 2.3$$

$$\sum_{j=1}^n \alpha_{jl} \leq 1 \quad l = 1, 2, \dots, r \quad 2.4$$

$$\sum_{l=1}^r \alpha_{jl} + \sum_{k=1}^n \theta_{kj} = 1 \quad j = 1, 2, \dots, n \quad 2.5$$

$$\sum_{k=1}^n \theta_{jk} \leq 1 \quad j = 1, 2, \dots, n \quad 2.6$$

$$C_k \geq C_j + G_{jk} + \sum_{l=1}^r p_{kl} \beta_{kl} + M \theta_{jk} - M \quad j = 1, 2, \dots, n; k = 1, 2, \dots, n \quad 2.7$$

$$C_j \geq \sum_{l=1}^r p_{jl} (\alpha_{jl} + \beta_{jl}) \quad j = 1, 2, \dots, n \quad 2.8$$

$$C_j + E_j - T_j = d_j \quad j = 1, 2, \dots, n \quad 2.9$$

$$d_j \theta_{jk} \leq d_k \theta_{jk} \quad j = 1, 2, \dots, n \quad 2.10$$

$$C_j, E_j, T \geq 0 \quad j = 1, 2, \dots, n \quad 2.11$$

A equação 2.1 determina o objetivo da formulação: minimizar a soma ponderada de atrasos e adiantamentos. Para isso, o número de atrasos e adiantamentos é multiplicado por uma penalização. A equação 2.2 assegura que um *job* deve ser atribuído a somente uma linha de produção de cada vez. A equação 2.3 assegura que um *job* e seu sucessor devem seguir na mesma linha. A equação 2.4 afirma que cada *job*, se for o primeiro a ser processado, só pode ser processado em uma linha de produção. A equação 2.5 garante que um *job* ou é o primeiro a ser processado ou sucede outro *job* na sequência de processamento. A equação 2.6 garante que um *job* sempre deve ser sucedido por outro, a menos que seja o último da sequência. A equação 2.7 garante que o instante de início de processamento de um *job* não pode ser menor que o instante de término de seu predecessor direto na sequência de processamento. A equação 2.8 garante que o tempo necessário para realizar um *job* deve ser maior ou igual que seu tempo de processamento. A equação 2.9 mede o grau de atraso ou antecipação de um *job*, ou seja, garante que o tempo total de uma tarefa somado aos atrasos e subtraído do que foi adiantado deve ser igual a data de entrega da tarefa. Na equação 2.10 certifica-se que a data de entrega de um *job* deve ser igual ou anterior à data de entrega de seu sucessor direto. A equação 2.11 garante a não negatividade das variáveis. Este modelo está preparado para controlar a data de entrega, não de uma maneira rígida, como um objetivo, mas como uma restrição, o que diminui a complexidade do modelo. A penalidade de atrasos e adiantamentos obriga o atendimento mais rígido a uma data.

Outro modelo que contém importantes elementos para uma solução do problema de sequenciamento é o de Gupta e Magnusson (2005). Este modelo se caracteriza por ter sido feito para um ambiente de máquina simples e com tempo de *setup* dependente da sequência. O problema para o qual o modelo foi desenvolvido é conhecido na literatura como dimensionamento de lote capacitado e sequenciamento (*CLSP – Capacitated Lot-sizing and Scheduling Problem*). O modelo foi formulado

para um problema de uma indústria de abrasivos e o objetivo é determinar a sequência e tamanho do lote que minimize o custo de *setup* e os estoques. Uma característica desse tipo de produção é que dependendo do tipo de lixa (abrasivo) que se pretende produzir, é necessário um tipo de cola e um grão específico. Por isso, após a produção de um tipo de abrasivo, a limpeza de máquina é imprescindível para dar início à produção de outro tipo, devido à diferença no material necessário para a produção e aos resíduos restantes nas máquinas. Este tempo de preparação (limpeza) é muito grande e compromete a capacidade produtiva da empresa. O modelo criado é o seguinte:

Parâmetros:

$P$  Número de períodos; o índice de períodos é  $t = \{1, \dots, P\}$ ;

$N$  Número de produtos; os índices de produtos são  $i, j = \{1, \dots, N\}$ ;

Variáveis de decisão:

$X_{it}$  Quantidade de produto  $i$  no período  $t$ ;

$Y_{it}$  Variável binária associada a  $X_{it}$ ;  $Y_{it} = 1$ , se o produto  $i$  for produzido no período  $t$ ; Caso contrário,  $Y_{it} = 0$ ;

$T_{ijt}$  Variável binária;  $T_{ijt} = 1$ , se houve mudança do produto  $i$  para o produto  $j$  no período  $t$ ; Caso contrário,  $T_{ijt} = 0$ ;

$D_{it}$  Demanda do produto  $i$  no período  $t$ ;

$I_{it}$  Saldo do produto  $i$  em estoque no período  $t$ ;

$h_i$  Custo do estoque do produto  $i$ ;

$C_{ij}$  Custo do *setup* do produto  $i$  para o produto  $j$ ; Assume-se que  $C_{ij} \leq C_{jk} + C_{kj}$  para todos  $i, j$  e  $k$ ; Para impedir subturnos considera-se  $C_{ii} = \infty$ ;

$s_{ij}$  Tempo de *setup* do produto  $i$  para o produto  $j$ ;

$\alpha_{it}$  Variável binária;  $\alpha_{it} = 1$ , se o produto  $i$  for produzido no início do período  $t$ ; Caso contrário,  $\alpha_{it} = 0$ ;

$\beta_{it}$  Variável binária;  $\beta_{it} = 1$ , se o produto  $i$  for produzido no final do período  $t$ ; Caso contrário,  $\beta_{it} = 0$ ;

$\gamma_{it}$  Variável binária;  $\gamma_{it} = 1$  se a máquina está preparada (*setup*) para o produto  $i$  ao final do período  $t$ ; Caso contrário,  $\gamma_{it} = 0$ ;

$\omega_{it}$  Variável contínua entre 0 e 1;  $\omega_{it} > 0$  se o produto  $i$  for produzido no período  $t$ ;

Caso contrário,  $\omega_{it} = 0$ ;

$\delta_{it}$  Variável contínua;  $\delta_{it} = 0$  se exatamente um produto  $i$  é produzido no período

$t$ ; Caso contrário,  $\delta_{it} > 0$ .

Modelo:

$$\text{Min} \sum_{i=1}^P \sum_{t=1}^N \sum_{j=1}^N C_{ij} T_{ijt} + \sum_{i=1}^P \sum_{t=1}^N h_i I_{it} \quad 2.12$$

Sujeito a:

$$I_{it-1} + X_{it} - I_{it} = D_{it} \quad 2.13$$

$$X_{it} - Y_{it} \leq 0 \quad 2.14$$

$$\sum_{i=1}^N X_{it} + \sum_{i=1}^N \sum_{j=1}^N T_{ijt} s_{ij} \leq 1 \quad \forall t \quad 2.15$$

$$Y_{it} \leq \omega_t \quad \forall i, t \quad 2.16$$

$$\sum_{i=1}^N Y_{it} - 1 \leq (N-1)\delta_t \quad \forall t \quad 2.17$$

$$\omega_t \leq \sum_{i=1}^N \alpha_{it} \leq 1 \quad \forall t \quad T \quad 2.18$$

$$\omega_t \leq \sum_{i=1}^N \beta_{it} \leq 1 \quad \forall t \quad 2.19$$

$$\alpha_{it} \leq Y_{it} \quad \forall i, t \quad 2.20$$

$$\beta_{it} \leq Y_{it} \quad \forall i, t \quad 2.21$$

$$\alpha_{it} + \beta_{it} \leq 2 - \delta_t \quad \forall i, t \quad 2.22$$

$$\sum_{i=1}^N Y_{it} = 1 \quad \forall t \quad T \quad 2.23$$

$$\sum_{j=1}^N T_{ijt} \geq Y_{it} - \alpha_{it} \quad \forall i, t \quad 2.24$$

$$\sum_{j=1}^N T_{ijt} \geq Y_{it} - \beta_{it} \quad \forall i, t \quad 2.25$$

$$T_{ijt} \geq Y_{it-1} + Y_{jt} - \omega_t - 1 \quad \forall i \neq j, t \quad 2.26$$

$$T_{ji} \geq \alpha_i + \gamma_{j-1} - 1 \quad \forall i \neq j, t \quad 2.27$$

$$T_{ij} \geq \beta_i + \gamma_{jt} - 1 \quad \forall i \neq j, t \quad 2.28$$

$$0 \leq \omega_t \leq 1 \quad \forall t \quad 2.29$$

$$X_{it}, I_{it}, \delta_t \geq 0 \quad \forall i, t \quad 2.30$$

$$T_{ji}, Y_{it}, \alpha_i, \beta_i, \gamma_{it} \in \{0,1\} \quad \forall i, j, t \quad 2.31$$

Características da formulação (2.12 – 2.31):

- O *setup* é preservado durante os possíveis períodos vazios;
- As restrições 2.13, 2.14 e 2.15 são típicas de controle de estoque;
- Devido às suas características, para considerar um *setup* seguido de produção que pode agrupar vários períodos, fez-se necessário o uso de variáveis de controle, como  $\gamma$ ,  $\omega$  e  $\delta$ ;
- Restrições 2.16 a 2.23 controlam quando um *setup* ocorre, o início da produção de um produto e seu fim. Essas informações são necessárias porque precisa-se garantir que o *setup* ocorra antes do início de cada produção;
- As restrições 2.24 e 2.25 são sempre aplicadas quando mais do que um produto é produzido em um período simples. Eles forçam que pelo menos um  $T_{ijt}$  seja igual a 1 para cada produto  $i$ , exceto quando este produto é o primeiro ou o último na sequência;
- Restrições 2.26 a 2.28 obrigam a existência de um *setup* sempre que houver mudança na produção de uma peça para outra;
- Restrições 2.29 a 2.31 determinam o escopo das variáveis (não negatividade);

Este modelo mostrou-se eficiente para encontrar a solução de pequenos exemplares do problema, mas é impraticável para resolver problemas práticos de grandes dimensões.

Outro trabalho que merece destaque é o de Meyr (2002), que também associa o sequenciamento de produção e dimensionamento de lotes de vários produtos em máquinas paralelas não uniformes, com *setup* dependente da sequência. O objetivo nesse trabalho foi a minimização do *setup* e o estoque em mãos. A solução foi obtida pelo uso de duas heurísticas simultâneas *Threshold Accepting* (TA) e a *Simulated Annealing* (SA).

Lee e Pinedo (1997) desenvolveram um trabalho em um ambiente de máquinas paralelas com data de entrega, *setup* dependente da sequência. Seu objetivo era a minimização do *makespan*. Para este cenário foi usada uma heurística de três fases, sendo a primeira, como um pré-processamento, estimando o *makespan* baseado na data de entrega; a segunda fase busca gerar uma sequência de entrega e na terceira fase é aplicada a metaheurística *Simulated Annealing*.

### **3. MODELOS MATEMÁTICOS PROPOSTOS**

#### **3.1 - Descrição do problema**

A fabricação de peças usando prensas de alta capacidade tem uma série de peculiaridades que afetam diretamente o planejamento da sequência de produção. A variação da demanda por parte do cliente é constante e pode ser feita dentro de um período curto (dias ou horas), exigindo revisão rápida de toda a sequência de ordens de produção. O tempo de *setup* é um fator fundamental quando este tem componentes variáveis. No caso em estudo o tempo de *setup* é um dos fatores mais importantes. Atrasos na entrega de matéria prima é outro fator que exige atenção, pois o tempo necessário para compra é grande e o pedido deve ser colocado com 90 dias de antecedência junto ao fornecedor. Assim, qualquer mudança na data de entrega compromete o planejamento pré-estabelecido e outro plano de produção precisa ser elaborado. Considera-se, neste trabalho, que o tamanho do lote econômico está pré-determinado. Neste estudo não se está considerando problemas de estocagem ou suprimentos de matéria prima, pois estes processos são independentes.

#### **3.2 O processo produtivo**

A empresa em estudo produz peças estampadas específicas para caminhões, camionetes, automóveis e tratores, atendendo aos clientes somente sob contrato. Seu produto não atende à venda direta ao consumidor final nem à empresas distribuidoras de peças. O processo de produção tem início após o fechamento de um contrato que normalmente significa fornecer peça para uma linha de produção enquanto o veículo for produzido. A quantidade de diferentes modelos de peças que essa empresa pode produzir é muito grande e a linha onde este estudo se realiza, chamada internamente de prensaria pesada, é onde prensas grandes fazem a conformação das peças a partir de chapas de aço. Prensas desse tipo precisam de um dispositivo acoplado que confere a elas capacidade de dar à chapa formato, corte e furação necessário. Este dispositivo é conhecido como ferramental e sua montagem na prensa necessita de muito tempo e

extrema precisão. O *setup* de máquina aqui considerado começa no momento que esse ferramental é colocado na prensa até seu teste, quando uma peça é feita e medida com calibres específicos para verificar se atende ao desenho. O ferramental é dividido em estampo inferior, estampo superior e blocos. Um ferramental pode servir para a produção de mais do que um tipo de peça, pois, com algumas modificações na sua montagem, agregando peças ou tirando punções, pode servir para produzir um modelo diferente de peça. Este tipo de operação é bastante comum porque de um ano para o outro um modelo de caminhão pode sofrer alterações que, por menores que sejam, podem significar a alteração de uma furação, dimensão etc. Outro fator que faz com que haja alteração em uma peça é uma mudança na estrutura global do veículo para atender a alguma funcionalidade particular, como aumento da capacidade do tanque de combustível, aumento do comprimento da carroceria etc. Devido ao alto custo para confecção de um ferramental, ele é único, impedindo assim que duas prensas utilizem o mesmo ferramental ao mesmo tempo.

### **3.3 – A Gestão da produção na empresa em estudo**

Na empresa em estudo o processo de gestão da produção ocorre da seguinte maneira: as peças estampadas são produzidas com base em um cronograma de entrega de peças fornecido pelo cliente. Esta solicitação chega através de transmissão eletrônica de dados (EDI – *Electronic Data Interchange*) sendo que uma programação para o curto prazo, de uma semana a vinte dias, dá subsídios para a programação da produção. A programação de peças para um longo prazo também chega via EDI.

O ERP (*Enterprise Resource Planning*) da empresa executa as rotinas de programação de compra e emissão de ordens de produção através de rotinas do MRP (*Material Requirement Planning*). As ordens de produção criadas são agrupadas de acordo com o lote econômico de produção. Todo este cálculo feito pelo MRP não leva em consideração a capacidade de produção das máquinas disponíveis.



Os dados recebidos via EDI alimentam a carteira de pedidos da empresa. Outra base de dados, denominada previsão de produção é alimentada pela área comercial com base nas tendências de mercado. A intersecção dos dados dessas duas informações vai fornecer o Plano Mestre de Produção. As rotinas do MRP irão se basear nestes dados como demanda de produção. As rotinas do MRP precisam ainda da lista de materiais e saldo dos estoques para criar as ordens de produção necessárias e as ordens de compra de matéria prima necessária. O fluxo da gestão de produção pode ser visto na Figura 2.

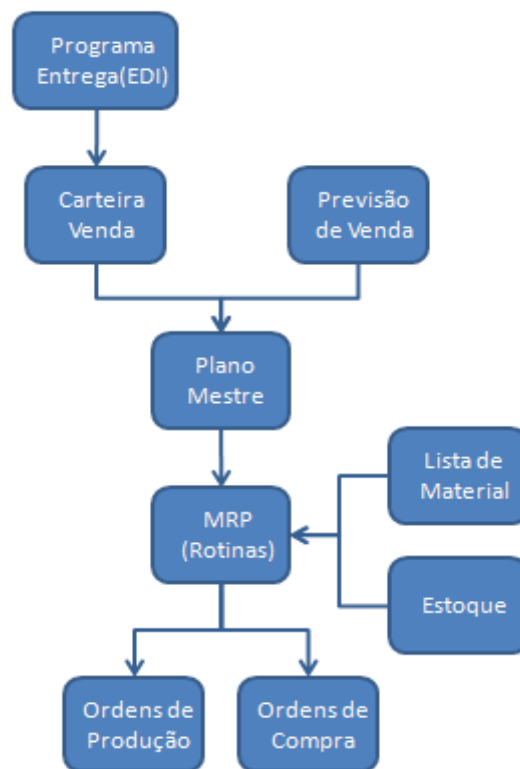


Figura 2 – Gestão da produção da empresa em estudo

Com base no cronograma de entrega de peças de curto prazo fornecido pelo cliente, os planejadores elaboram o plano de produção para uma semana. Para isso, fazem uso de planilhas eletrônicas, ou seja, não fazem uso de nenhuma ferramenta de Programação Matemática. Este plano é passado para o setor de ferramentaria, que por sua vez prepara os ferramentais na ordem em que devem ser usados. O plano é também passado para o setor de fornecimento de matéria prima, que deve preparar a

chapa de aço para a produção. Esta preparação consiste no corte da chapa em um formato adequado para a produção, conhecido por *blank*. Em alguns casos, esse processo de corte da chapa no tamanho desejado é realizado por terceiros, pois a compra de aço é feita em forma de bobinas e para o desbobinamento dessa peça é necessário o uso de maquinário adequado.

A vivência nesse ambiente de produção tem mostrado que a ocorrência de reprogramações nos clientes é um fator constante. Uma das causas disto é o uso de conceitos como o JIS e o JIT, que levam as montadoras a ter um estoque enxuto e suscetível a uma série de imprevistos. Isto exige dos fornecedores agilidade em atender às alterações na demanda. Com isso, todo o processo de planejamento precisa ser revisto rapidamente.

A planta dessa linha de produção consiste de três prensas de alta capacidade de 3000 a 5200 toneladas. Uma relação de 2118 tipos diferentes de peças é possível ser produzida e 75 ferramentais são usados para a fabricação destas peças. O ferramental tem por característica servir, com algumas modificações, para produzir várias peças, conforme dito anteriormente, criando assim uma forte restrição na programação da produção, pois quando um ferramental estiver em uso para produção de uma peça, outras peças que utilizam o mesmo ferramental não poderão ser produzidas. A quantidade de modelos de peças produzidas por semana pode chegar a 150, sendo a produção unitária semanal de cerca de 3500 peças.

O tempo de *setup* é muito longo, tornando-se um fator crítico na programação da produção. Além disso, a sequência do *setup* pode alterar seu tempo, pois, algumas partes de um ferramental podem ser compartilhadas para a produção de outras peças. Assim, a partir da produção de uma peça, pode-se agilizar a preparação para a produção da próxima peça. Portanto, neste estudo, o *setup* tem tempo variável, dependente da sequência de produção.

O planejamento feito com base em planilhas não garante que o caminho escolhido para produção das peças é o melhor. As variações acarretam grandes transtornos, uma vez que esta planilha é de difícil manutenção.

### 3.4 - Os modelos matemáticos

O estudo do ambiente do problema mostra duas possíveis soluções para a programação da produção. A primeira delas tem como objetivo a redução do custo total de produção dentro de um período, levando-se em consideração principalmente o tempo de *setup*. A ideia é que, com o período total estabelecido, a redução de tempo gasto com o *setup* se converta em redução de custo para a empresa. A segunda alternativa para a programação da produção tem como objetivo minimizar o *makespan*.

Para a minimização do tempo de *setup* são apresentados dois modelos. O Modelo I, criado especificamente para este trabalho, baseia-se no DLSP (*Discrete Lotsizing Scheduling Problem*). O Modelo II baseia-se no trabalho de Gupta e Magnusson (2005) para o CLSP (*Capacitated Lotsizing Scheduling Problem*).

Para a minimização do *makespan* criou-se o Modelo III, também estabelecido especificamente para este trabalho. Os resultados obtidos a partir deste modelo serão comparados com os resultados obtidos por um método heurístico híbrido, também desenvolvido para o mesmo objetivo e descrito no Capítulo 4.

Os modelos propostos objetivam atender a um cenário de máquinas paralelas com tempo de *setup* dependente da sequência de produção em que estoques devem ser evitados, mas são aceitáveis tendo em vista a prática de lotes mínimos de produção. Atrasos devem ser evitados e a demanda deve ser atendida plenamente. O custo de produção não varia de acordo com a prensa e neste trabalho não é necessário controlar o dimensionamento de lotes pois este é feito através das políticas implementadas nas rotinas do MRP. Considera-se também que o tamanho do lote não altera o tempo de *setup*.

### 3.5 - Modelo I: minimização do tempo total de *setup*

O Modelo I prevê a produção de um item por unidade de tempo (*small bucket*). Este tipo de modelo permite um maior controle sobre a produção das peças, pois sabe-se em que período cada peça será produzida. Este detalhe é de alta importância nesse modelo, devido à uma forte restrição que impede que algumas peças sejam produzidas em duas prensas ao mesmo tempo. Como explicado anteriormente, esta restrição ocorre porque um ferramental não pode ser usado ao mesmo tempo em duas prensas diferentes. Outras restrições consideradas no modelo são: o tempo máximo de produção é fixo, ou seja, a produção não pode exceder a este tempo máximo pré-estabelecido (programação finita).

O modelo proposto nesta seção tem por objetivo resolver o problema descrito anteriormente de sequenciar as peças na linha de produção de modo a se obter uma otimização do tempo de produção a partir da redução dos tempos de *setup* das máquinas. Conforme comentado anteriormente, este tempo é muito alto devido à complexidade do ferramental necessário.

Para validar o modelo desenvolvido foram conduzidos alguns testes computacionais. Apesar de ser baseado em um problema real, os valores dos parâmetros e os dados usados nestes testes foram gerados aleatoriamente, pois os dados da empresa referem-se a grandes quantidades, o que dificulta um teste preliminar. Os testes foram executados em um microcomputador com processador Pentium Dual Core, com 3 Gigabytes de memória, 120 Gigabytes de disco, sob o sistema operacional Windows Vista. Para a solução do modelo foram utilizados os *solvers* Xpress (versão 1.22.04) e CPLEX (versão 12).

Como será apresentado a seguir, o desenvolvimento do Modelo I foi realizado paulatinamente. Alguns fatores que não foram considerados inicialmente, foram sendo incorporados ao modelo de modo a torná-lo cada vez mais realista com o problema encontrado na prática. Essa metodologia utilizada no desenvolvimento deste modelo

visa enriquecer a pesquisa e mostrar como novos estudos podem vir a ser feitos, uma vez que qualquer mudança na política de produção ou em algum fator como matéria prima, disponibilidade de máquina, ferramental etc., terá que ser incorporada ao modelo como se fez, neste caso, com a metodologia adotada.

Os elementos que compõem o Modelo I são apresentados a seguir.

Parâmetros:

- $I$  Número de produtos; os índices  $i$  e  $j$  são usados para representar uma peça em produção ou uma peça predecessora;
- $K$  Número de prensas; o índice  $k \in \{1, \dots, K\}$  é usado para representar uma prensa; O número de prensas pode variar, caso algum centro de trabalho seja dedicado a esta linha, ou alguma prensa ser paralisada para manutenção.
- $P$  Número de períodos; o índice  $p \in \{1, \dots, P\}$  é usado para representar um período.

Dados:

- $D_i$  Demanda da peça  $i$ ; Conforme descrição anterior deve ser atendida em sua totalidade.
- $Pt_i$  O tempo de processamento da peça  $i$ ;
- $S_{ij}$  Tempo de *setup* da peça  $i$  após a produção da peça  $j$ .

Variáveis de decisão:

- $t_{kp}$  Tempo de início da produção de uma peça no período  $p$  na prensa  $k$ ;
- $z_{ikp}$  Variável binária tal que  $z_{ikp} = 1$  se a peça  $i$  foi alocada no período  $p$  da sequência de produção da prensa  $k$ ;  $z_{ikp} = 0$ , caso contrário.
- $w_{ijkp}$  Variável binária tal que  $w_{ijkp} = 1$  se a peça  $i$  foi alocada no período  $p$  e a peça  $j$  foi alocada no período  $(p+1)$  da sequência de produção da prensa  $k$ .

Formulação:

$$\text{Min} \sum_{\substack{i \in I \\ k \in K \\ p \in P}} P t_i z_{ikp} + \sum_{\substack{i, j \in I \\ i \neq j \\ k \in K \\ p \in P}} S_{ij} w_{ijkp} \quad 3.1$$

Sujeito a

$$t_{k(p+1)} - t_{kp} - \sum_{\substack{i \in I \\ k \in K}} P t_i z_{ikp} - \sum_{\substack{i, j \in I \\ i \neq j \\ k \in K}} S_{ij} w_{ijkp} \geq 0 \quad \forall p \in P \quad 3.2$$

$$\sum_{i \in I} z_{ikp} = 1 \quad \forall p \in P, k \in K \quad 3.3$$

$$\sum_{\substack{k \in K \\ p \in P}} z_{ikp} \geq D_i \quad \forall i \in I \quad 3.4$$

$$w_{ijkp} - z_{ikp} - z_{jk(p+1)} \geq -1 \quad \forall i, j \in I; i \neq j; k \in K; p \in P \quad 3.5$$

$$t_{kp} \geq 0 \quad \forall k \in K; p \in P \quad 3.6$$

$$w_{ijkp} \in \{0,1\} \quad \forall i, j \in I; k \in K; p \in P \quad 3.7$$

$$z_{ikp} \in \{0,1\} \quad \forall i \in I; k \in K; p \in P \quad 3.8$$

A função-objetivo 3.1 busca a minimização dos tempos de processamento e tempos de preparação das peças a serem produzidas. As restrições 3.2 indicam que o processamento no período  $(p+1)$  da sequência de produção da prensa  $k$  só pode ser iniciado depois que o processamento (incluindo preparação) no período anterior desta prensa tenha sido terminado. As restrições 3.3 indicam que em cada período  $p$  da sequência de produção da prensa  $k$  pode existir apenas uma peça. As restrições 3.4 indicam que a demanda de cada peça deve ser atendida plenamente. Estas restrições serão de igualdade, caso não se deseje estoque de peças produzidas. As restrições 3.5 indicam que peça  $i$  é alocada no período  $p$  e a peça  $j$  é alocada no período imediatamente seguinte  $(p+1)$  da sequência de produção da prensa  $k$ . As restrições 3.6, 3.7 e 3.8 determinam o domínio das variáveis.



Neste ponto é interessante destacar a variável  $P$ , conforme descrito anteriormente, que representa a quantidade de períodos. Por isso ela deve ter um valor grande o bastante para conter toda a produção. Esta variável pode aumentar o processamento consideravelmente uma vez que ela é uma das dimensões das variáveis  $z$  e  $w$ .

Tabela 4 – Resultados obtidos no Teste 1

Tempo total de produção	72
Tempo total de preparação	1

Como pode ser observado, a solução obtida considera uma distribuição de peças por todos os períodos. Entretanto, alguns problemas foram identificados neste modelo, os quais são comentados a seguir.

### Problema 1: Preparação inicial

O modelo não considera o tempo de *setup* para a preparação inicial, ou seja, para a produção da primeira peça da sequência para cada prensa. O *setup* inicial deve existir, pois a preparação consiste em colocar um ferramental na máquina, específico para uma peça, por isso a prensa nunca fica previamente montada.

Para resolver este problema, verificou-se que as restrições (3.5) não atenderam à necessidade de existir um *setup* para cada início de produção. A alternativa para isso foi a imposição de que se há produção no primeiro período o *setup* deve existir. Para isso foram criadas as restrições (3.9):

$$prep\_ini = \sum_{\substack{i \in N \\ k \in K}} St_i * z_{ik1} \quad 3.9$$

em que  $St_i$  corresponde ao tempo de *setup* inicial da peça  $i$ , que será considerado sempre que esta peça for produzida no período 1. A implementação na linguagem Mosel de todos os modelos pode ser vista no Apêndice B.



## Problema 2: Atendimento da demanda

O Modelo I também não considera um tratamento adequado para a demanda. Uma quantidade muito maior que o previsto foi programado. A produção excede à necessidade porque a restrição 3.4 impõe um limite mínimo e nenhuma outra restrição limita a produção. Além disso, a solução adotada inicialmente foi a inclusão do custo de produção na função objetivo.

Deve-se observar que, neste trabalho, o custo de preparação ( $cp$ ) de uma peça não está relacionado com a máquina onde esta peça será produzida. A função objetivo acrescida do custo de produção para resolver esse problema fica da seguinte maneira:

$$\text{Min} \sum_{\substack{i \in I \\ k \in K \\ p \in P}} Pt_i z_{ikp} + \sum_{\substack{i, j \in I \\ i \neq j \\ k \in K \\ p \in P}} S_{ij} w_{ijkp} + \sum_{\substack{i \in I \\ k \in K \\ p \in P}} cp_i z_{ikp} \quad 3.10$$

Conforme comentado anteriormente, a demanda deve ser atendida plenamente e a produção em excesso deve ser evitada, pois os produtos aqui abordados são específicos para alguns veículos e, caso a fabricação deste veículo seja descontinuada, qualquer estoque vai significar perda de produção e material. A restrição usada primeiramente como uma desigualdade do tipo "maior ou igual", condicionada pela minimização dos custos deveria bastar. Entretanto, ao incluir esta restrição no modelo o problema tornou-se inviável. Verificou-se então que as restrições 3.3 obrigam a existência de produção em todos os períodos. Para resolver este conflito optou-se pela eliminação das restrições 3.3 e a troca das restrições 3.4 pela 3.11 conforme abaixo:

$$\sum_{\substack{k \in K \\ p \in P}} z_{ikp} = D_i \quad \forall i \in I \quad 3.11$$

### Problema 3: Produção simultânea de peças

O modelo, da forma como se encontra, apresenta uma falha no tratamento de uma restrição muito forte do problema: uma peça não pode ser produzida em duas prensas ao mesmo tempo. Isso não é possível, pois o ferramental associado a uma peça é único, como já explicado anteriormente. Portanto quando uma peça está sendo produzida em uma prensa, ela não pode ser produzida em outra prensa no mesmo período. Para assegurar essa restrição foram incluídas as restrições 3.12 e 3.13:

$$\sum_{i \in I} z_{ikp} \leq 1 \quad \forall p \in P, k \in K \quad 3.12$$

$$\sum_{k \in K} z_{ikp} \leq 1 \quad \forall p \in P, i \in I \quad 3.13$$

Este novo modelo, com o acréscimo das restrições 3.9, 3.10, 3.11, 3.12 e 3.13, menos as restrições 3.3 e 3.4 foi executado para os mesmos dados e os resultados obtidos estão mostrados nas Tabelas 5 e 6.

Tabela 5 – Sequenciamento obtido no Teste 2

		Período											
		1	2	3	4	5	6	7	8	9	10	11	12
Prensa	1		3		1	1		3					1
	2		2	2	2			2	2		1		3

Tabela 6 – Resultados obtidos no Teste 2

Tempo total de produção	12
Tempo total de preparação	0
Tempo de <i>setup</i> inicial	0
Custo de produção das peças	35

Embora o modelo modificado representa melhor a realidade da empresa, alguns outros problemas ainda foram identificados.

#### Problema 4: O *setup* é obrigatório

O modelo atual não considera a obrigatoriedade dos *setups*. A análise da solução apresentada mostra que a falta de *setup* foi criada pelo programa como uma maneira de evitar a agregação de custo ao resultado final. Na prática, os espaços entre a produção de peças não existem, pois toda a produção é tratada de maneira ininterrupta. As paradas resumem-se a eventos programados como hora para o almoço, manutenção etc. Para resolver essa deficiência da formulação foram acrescentadas mais algumas restrições ao modelo. Um primeiro conjunto de restrições força a realização de *setup* sempre que houver a mudança na produção de uma peça para outra. Outras restrições forçam a existência de um *setup* apesar de haver intervalos entre as peças a serem produzidas.

A obrigatoriedade de *setup* é identificada sempre que uma peça ocupa uma posição do período de produção e na posição seguinte tem-se uma outra peça diferente. Quando isso ocorre, necessariamente neste intervalo deve ocorrer um *setup*. Para isso, foram acrescentadas ao modelo as restrições 3.14 e 3.15. Estas restrições obrigam a variável  $w$  assumir o valor 1 (restrição 3.14) se houver produção de peças diferentes em posições consecutivas, ou a assumir o valor 0 (restrição 3.15) se não houver. Dessa maneira as restrições 3.5 e 3.6 podem ser eliminadas, pois sua função já foi atendida por estas duas novas restrições.

$$z_{i,k,p} + z_{j,k,p+1} - 2 * w_{i,j,k,p} \geq 0 \quad \forall i, j \in I, i \neq j; k \in K, p \in P \quad 3.14$$

$$z_{i,k,p} + z_{j,k,p+1} - 2 * w_{i,j,k,p} \leq 1 \quad \forall i, j \in I, i \neq j; k \in K, p \in P \quad 3.15$$

Com a inclusão das restrições 3.14 e 3.15 e a eliminação das restrições 3.5 e 3.6, obtém-se a solução mostrada nas Tabelas 7 e 8.

Tabela 7 – Sequenciamento obtido no Teste 3

		Período											
		1	2	3	4	5	6	7	8	9	10	11	12
Prensa	1		3	3		1		3					1
	2		2	2				1	1		2	2	2

Tabela 8 – Resultados obtidos no Teste 3

Tempo total de produção	12
Tempo total de preparação	0
Tempo de <i>setup</i> inicial	0
Custo de produção das peças	35

Mesmo com essas novas restrições, observa-se que o modelo deixa espaços (posições vazias no período de produção) entre a produção de duas peças diferentes, como forma de evitar a inclusão de *setup* entre elas. Para impedir que isso ocorra, criou-se a variável  $\alpha_{kp}$  para controlar o conteúdo de cada período  $p$  em cada prensa  $k$ . Se no período  $p$ , a prensa  $k$  produz alguma peça então  $\alpha_{kp} = 1$ ; senão  $\alpha_{kp} = 0$ . Assim, foram acrescentadas as seguintes restrições:

$$\alpha_{kp} = \sum_i^I z_{i,k,p} \quad 3.16$$

$$\alpha_{k,p} \geq \alpha_{k,p+1} \quad 3.17$$

As restrições 3.16 garantem que o valor da variável  $\alpha$  será 1 sempre que houver produção de alguma peça. As restrições 3.17 fazem com que não seja possível a existência de espaços vazios entre a produção de duas peças diferentes. Com isso, tem-se uma nova formulação para o Modelo I.

Para essa nova formulação, com a inclusão das restrições 3.16 e 3.17 o teste realizado obteve a solução mostrada nas Tabelas 9 e 10.

Tabela 9 – Sequenciamento obtido no Teste 4

		Período											
		1	2	3	4	5	6	7	8	9	10	11	12
Prensa	1	1	1	1	1	3	3	3					
	2	2	2	2	2	2							

Tabela 10 – Resultados obtidos no Teste 4

Tempo total de produção	12
Tempo total de preparação	1
Tempo de <i>setup</i> inicial	2
Custo de produção das peças	35

Observa-se que, finalmente, a solução obtida atende aos requisitos impostos ao problema: as peças são produzidas continuamente, não havendo intervalos no período de produção; tem-se, obrigatoriamente, um *setup* inicial e um *setup* sempre que ocorre mudança de produção de peças.

Deve-se observar que nesta formulação, o parâmetro  $P$  (número de períodos) assume um papel muito importante. Considerando a restrição de que uma peça não pode ocupar a mesma posição em duas prensas ao mesmo tempo, o valor de  $P$  pode ser estimado da seguinte forma:  $P$  não pode ser menor do que o tempo total necessário para fazer a peça que demora mais períodos e nem pode ser menor do que a soma de todos os períodos necessários para a produção de todas as peças.

Novos testes computacionais foram realizados com esta formulação do Modelo I, utilizando o *solver* Xpress. A Tabela 11 mostra os resultados obtidos nestes novos testes computacionais. As legendas utilizadas nas colunas desta tabela são as seguintes:

NM - Número de máquinas (prensas);

NP - Número de peças;

NL - Número de linhas (restrições) do modelo;

NC - Número de colunas (variáveis) do modelo;

CPU – Tempo necessário para resolver o modelo (em segundos).

Produção - Tempo gasto para produzir o item;

Custo - Custo resultante da produção das peças;

Setup - Tempo de *setup* utilizado durante a produção;

Inicial - Tempo gasto com o *setup* no início da produção (momento 0);

Tabela 11 – Resultados obtidos no Teste 5

NM	NP	NL	NC	CPU	Produção	Custo	Setup	Inicial
2	3	141	132	0,3	8,76	51	210	260
3	3	306	306	1,0	17,04	88	0	390
4	3	306	408	1,1	18,00	89	0	390
2	4	458	456	7,8	16,44	84	90	390
3	4	333	348	8,0	11,88	62	190	390
4	4	432	464	7,3	12,24	63	0	520
2	5	453	460	87,2	10,44	52	280	260
3	5	581	615	203,0	13,32	71	120	390
4	5	572	620	310,5	14,88	79	250	520

Os testes foram feitos com uma sequência de 3, 4, 5, 10, 15 e 20 peças. A partir de 10 peças, a execução do modelo torna-se demasiadamente lenta e após 1 hora de processamento o sistema encerrou a execução por falta de recurso (memória). Pelos resultados da Tabela 11, pode-se observar que o tempo de execução é alto, mesmo para um número pequeno de peças. Isto inviabiliza a utilização desta formulação para exemplares do problemas mais próximos do real, com 3 prensas e 180 peças.

Esta formulação, no entanto, ainda não inclui uma restrição importante do problema real: duas ou mais peças que dividem o mesmo ferramental, ou parte dele, não podem ocupar a mesma posição no período de produção, em prensas diferentes. Tais peças são denominadas, neste trabalho, como peças incompatíveis e uma nova variável  $R_{ij}$  foi criada para indicar se as peças  $i$  e  $j$  são incompatíveis ( $R_{ij} = 1$ ) ou não ( $R_{ij} = 0$ ). Para isso, a seguinte restrição foi inserida no Modelo I:

$$z_{ikp} + z_{jmp} + R_{i,j} \leq 2 \quad \forall i, j \in I; \quad i \neq j; \quad k, m \in K; k \neq m; \quad p \in P \quad 3.18$$

Com isso, as restrições 3.2 podem ser retiradas deste modelo porque ele está ocupando todas as posições na linha do tempo e caracterizando esta posição de maneira que a variável  $t$  não precisa mais ser controlada. Além disso, a função de custo se mostrou ineficiente na função-objetivo, além de comprometer seu resultado,

uma vez que este custo não está relacionado com o tempo como as duas outras parcelas. Para isso introduz-se a função objetivo 3.19 no lugar da 3.1.

$$\text{Min} \sum_{\substack{i \in I \\ k \in K \\ p \in P}} P t_i z_{ikp} + \sum_{\substack{i, j \in I \\ i \neq j \\ k \in K \\ p \in P}} S_{ij} w_{ijkp} \quad 3.19$$

Para esta nova formulação, acrescentando à anterior as restrições 3.18 e 3.19, e retirando as restrições 3.1 e a 3.2, foram realizados testes computacionais considerando os dados mostrados nas Tabelas 12, 13 e 14. Os resultados obtidos são mostrados nas Tabelas 15 e 16.

Tabela 12 – Parâmetros das peças

	Peça		
	1	2	3
Demanda	4	5	3
Custo	2	3	4
Tempo de produção	3	3	3

Tabela 13 – Tempo de *setup*

		Peça		
		1	2	3
Peça	1	0	3	1
	2	3	0	2
	3	1	2	0

Tabela 14 – Incompatibilidade entre peças

		Peça		
		1	2	3
Peça	1	1	1	0
	2	1	1	0
	3	0	0	1

Tabela 15 – Sequenciamento obtido no Teste 5

		Período											
		1	2	3	4	5	6	7	8	9	10	11	12
Prensa	1												
	2	2	2	2	2	2	3	3	3	1	1	1	1

Tabela 16 – Resultados obtidos no Teste 5

Tempo total de produção	12
Tempo total de preparação	3
Tempo de <i>setup</i> inicial	1
Custo de produção das peças	35

O modelo obtido final fica da seguinte maneira:

$$\text{Min} \sum_{\substack{i \in I \\ k \in K \\ p \in P}} Pt_i z_{ikp} + \sum_{\substack{i,j \in I \\ i \neq j \\ k \in K \\ p \in P}} S_{ij} w_{ijkp} \quad (3.19)$$

Sujeito a

$$w_{ijkp} \in \{0,1\} \quad \forall i, j \in I; k \in K; p \in P \quad (3.7)$$

$$z_{ikp} \in \{0,1\} \quad \forall i \in I; k \in K; p \in P \quad (3.8)$$

$$\text{prep\_ini} = \sum_{\substack{i \in I \\ k \in K}} St_i * z_{ik1} \quad (3.9)$$

$$\sum_{\substack{k \in K \\ p \in P}} z_{ikp} = D_i \quad \forall i \in I \quad (3.11)$$

$$\sum_{i \in I} z_{ikp} \leq 1 \quad \forall p \in P, k \in K \quad (3.12)$$

$$\sum_{k \in K} z_{ikp} \leq 1 \quad \forall p \in P, i \in I \quad (3.13)$$

$$z_{i,k,p} + z_{j,k,p+1} - 2 * w_{i,j,k,p} \geq 0 \quad \forall i, j \in I, i \neq j; k \in K, p \in P \quad (3.14)$$

$$z_{i,k,p} + z_{j,k,p+1} - 2 * w_{i,j,k,p} \leq 1 \quad \forall i, j \in I, i \neq j; k \in K, p \in P \quad (3.15)$$

$$\alpha_{kp} = \sum_i^I z_{i,k,p} \quad (3.16)$$



$$\alpha_{k,p} \geq \alpha_{k,p+1} \quad (3.17)$$

$$z_{ikp} + z_{jmp} + R_{i,j} \leq 2 \quad \forall i, j \in I; \quad i \neq j; \quad k, m \in K; k \neq m; \quad p \in P \quad (3.18)$$

O tempo excessivo gasto pelo *solver* Xpress versão 1.22.04 motivou a utilização do *solver* CPLEX versão 12. A Tabela 17 compara os resultados obtidos pelos dois *solvers* para alguns exemplares do problema. Nesta tabela as legendas das colunas são as seguintes:

NM - Número de máquinas (prensas);

NP - Número de peças;

FO - Valor da função-objetivo;

NL - Número de linhas (restrições) do modelo;

NC - Número de colunas (variáveis) do modelo;

CPU - Tempo de execução (em segundos);

XP – *Solver* Xpress;

CP – *Solver* CPLEX.

Tabela 17 – Comparativo Xpress versus CPLEX

		FO		NL		NC		CPU	
NM	NP	XP	CP	XP	CP	XP	CP	XP	CP
3	3	18	18	844	867	526	531	2,1	0,34
3	4	26	26	1549	1572	930	935	7,9	49,4
3	5	32	32	2582	2625	1449	1454	1670	440,6
3	6	49	41	3744	3786	2582	2086	7200	7200
3	7	**	54	5133	5175	2827	2833	7200	7200

O tempo máximo de execução foi estabelecido em 7200 segundos. A partir desse limite os resultados não são ótimos. O modelo submetido ao CPLEX foi obtido a partir do modelo submetido ao Xpress. O Xpress faz, por padrão, uma otimização dos dados de entrada, eliminando equações que não terão função na execução do modelo. Esta otimização é conhecida por *presolve* e tem influência direta nos dados gerados para o CPLEX. Os resultados mostraram que o CPLEX é mais rápido para a maioria

dos exemplares; a exceção ocorreu para o exemplar com  $NP = 4$ . Para 7 peças o Xpress não obteve sequer uma solução viável no limite de tempo estabelecido.

### 3.6 - Modelo II: minimização do tempo total de *setup*

O alto tempo de execução do Modelo I, tanto pelo Xpress como pelo CPLEX, motivou a busca por outros modelos para a solução do problema. O modelo desenvolvido por Gupta e Magnusson (2005), detalhado na seção 2.5, para controle de máquinas paralelas com tempos de *setup* dependentes da sequência de produção, foi adaptado para o problema em estudo. Isto foi possível devido ao fato do modelo de Gupta e Magnusson controlar os momentos inicial e final de cada lote produzido, o que permite controlar, a cada instante, qual produto está sendo produzido, permitindo, portanto, implementar a restrição que impede que duas peças que utilizem o mesmo ferramental sejam produzidas no mesmo momento em prensas diferentes.

A Tabela 18 compara as características do modelo original de Gupta e Magnusson (Modelo GM) em relação ao Modelo II proposto nesta seção. A Tabela 19 mostra as conversões realizadas nas variáveis e parâmetros do modelo de Gupta e Magnusson para atender ao problema em estudo.

Tabela 18 – Comparação das características dos modelos

	<b>Modelo GM</b>	<b>Modelo II</b>
Quantidade de estágios	Estágio simples	Estágio simples
Ambiente de máquinas	Paralelas / idênticas	Simple
<i>Setup</i>	Dependente da sequência	Dependente da sequência
Parâmetros de <i>setup</i>	Somente tempos	Tempos e custos
Preserva <i>setup</i> por períodos	Não	Sim
Restrição de capacidade	Capacitado	Capacitado
Considera os estoques	Não	Sim
Período de produção	<i>Small bucket</i> : um item é	<i>Big bucket</i> : vários itens são

	produzido por período	produzidos em um período
Número de produtos	Vários itens	Vários itens
Deterioração de produto	Não considera	Não considera
Demanda	Estática	Estática
Falta de estoque	Não permitida	Não permitida
Dimensionamento de lote	Não	Sim
Divisão de lote	Não é permitido	Permitido

Tabela 19 – Conversão de variáveis

Variável	Modelo GM	Variável	Modelo II
$T$	Número de períodos, tendo como índice $t = \{1, \dots, T\}$	$P$	Número de períodos, tendo como índice $t = \{1, \dots, P\}$
$N$	Quantidade de produtos	$N$	Sem mudança
$x_{it}$	Quantidade de produção do produto $i$ no período $t$	$x_{ikt}$	Quantidade de produção do produto $i$ na máquina $k$ no período $t$
$Y_{it}$	Igual a 1, se o produto $i$ for produzido no período $t$ ; igual a 0, caso contrário	$Y_{ikt}$	Igual a 1, se o produto $i$ for produzido na máquina $k$ no período $t$ ; igual a 0, caso contrário
$T_{ijt}$	Igual a 1, se o <i>setup</i> ocorreu da peça $i$ para $j$ no período $t$ ; igual a 0, caso contrário	$T_{ijkt}$	Igual a 1, se o <i>setup</i> ocorreu da peça $i$ para $j$ na máquina $k$ no período $t$ ; igual a 0, caso contrário
$D_{it}$	Demanda do produto $i$ ao final do período $t$	$D_{it}$	Sem mudança
$I_{it}$	Estoque do produto $i$ ao final do período $t$		Não utilizada
$H_i$	Custo do estoque do produto $i$		Não utilizada
$C_{ij}$	Custo do <i>setup</i> do produto $i$ para o produto $j$	$C_{ij}$	Sem mudança
$S_{ij}$	Tempo de <i>setup</i> do produto $i$ para o produto $j$	$S_{ij}$	Sem mudança
$\alpha_{it}$	Igual a 1, se o produto $i$ for o primeiro a ser produzido no período $t$ ; igual a 0, caso contrário	$\alpha_{ikt}$	Igual a 1, se o produto $i$ for o primeiro a ser produzido no período $t$ na máquina $k$ ; igual a 0, caso contrário

$\beta_{it}$	Igual a 1, se o produto $i$ for produzido por último no período $t$ ; igual a 0, caso contrário	$\beta_{ikt}$	Igual a 1, se o produto $i$ for produzido na máquina $k$ por último no período $t$ ; igual a 0, caso contrário
$\gamma_{it}$	Igual a 1, se a máquina está preparada para o produto $i$ no período $t$ ; igual a 0, caso contrário	$\gamma_{ikt}$	Igual a 1, se a máquina $k$ está preparada para o produto $i$ no período $t$ ; igual a 0, caso contrário
$\omega_t$	Positivo quando pelo menos um produto é produzido no período $t$	$\omega_{kt}$	Positivo quando pelo menos um produto é produzido no período $t$ na máquina $k$
$\delta_t$	Igual a zero, se exatamente um produto é produzido no período $t$ ; qualquer valor positivo, caso contrário	$\delta_{kt}$	Igual a zero, se exatamente um produto é produzido no período $t$ na máquina $k$ ; qualquer valor positivo, caso contrário

Apresenta-se a seguir a formulação do Modelo II. O Apêndice D apresenta esta formulação escrita na linguagem de modelagem Mosel do *solver* Xpress.

$$\text{Min} \sum_{\substack{i,j \in N \\ k \in K \\ t \in T}} S_{ij} * T_{ijk} + \sum_{\substack{i \in N \\ k \in K \\ t \in T}} x_{ikt} \quad 3.20$$

Sujeito a:

$$\sum_{k \in K} x_{ikt} = D_i \quad \forall i, t \quad 3.21$$

$$x_{ikt} - y_{ikt} \leq 0 \quad \forall i, k, t \quad 3.22$$

$$\sum_{i \in N} y_{it} \leq 1 \quad \forall t, k \quad 3.23$$

$$y_{ikt} \leq \omega_{kt} \quad \forall i, k, t \quad 3.24$$

$$\sum_{i \in N} y_{ikt} - 1 \leq (N - 1)\delta_{kt} \quad \forall k, t \quad 3.25$$

$$\omega_{kt} \leq \sum_{i \in N} \alpha_{ikt} \leq 1 \quad \forall k, t \quad 3.26$$

$$\omega_{kt} \leq \sum_{i \in N} \beta_{ikt} \leq 1 \quad \forall k, t \quad 3.27$$

$$\alpha_{ikt} \leq y_{ikt} \quad \forall i, k, t \quad 3.28$$

$$\beta_{ikt} \leq y_{ikt} \quad \forall i, k, t \quad 3.29$$

$$\alpha_{ikt} + \beta_{ikt} \leq 2 - \delta_{kt} \quad \forall i, k, t \quad 3.30$$

$$\sum_{i \in N} \gamma_{ikt} = 1 \quad \forall t, k \quad 3.31$$

$$T_{ijkt} \geq \gamma_{ikt-1} + \gamma_{jkt} - \omega_{kt} - 1 \quad \forall i \neq j, kt \quad 3.32$$

$$T_{ijkt} \geq \alpha_{ikt} + \beta_{jkt-1} - 1 \quad \forall i \neq j, k, t \quad 3.33$$

$$0 \leq \omega_{kt} \leq 1 \quad \forall t, k \quad 3.34$$

$$0 \leq x_{ikt} \leq 1 \quad \forall i, t, k \quad 3.35$$

$$0 \leq \delta_{kt} \leq 1 \quad \forall t, k \quad 3.36$$

$$T_{jii}, Y_{ii}, \alpha_{ii}, \beta_{ii}, \gamma_{ii} \in \{0,1\} \quad \forall i, j, k, t \quad 3.37$$

$$T_{iikt} = 0 \quad \forall i, k, t \quad 3.38$$

$$y_{ikt} + y_{ik't} - R_{ij} \leq 2 \quad \forall i \neq j, k \neq k', t \quad 3.39$$

$$\sum_{i \in N} \alpha_{ikt} \leq 1 \quad \forall k, t \quad 3.40$$

$$\sum_{i \in N} \beta_{ikt} \leq 1 \quad \forall k, t \quad 3.41$$

$$\omega_{kt} \geq \omega_{k,t+1} \quad \forall k, t \quad 3.42$$

$$prep\_ini = \sum_{\substack{i \in N \\ k \in K}} St_i * y_{ik1} \quad \forall k, t \quad 3.43$$

A função-objetivo 3.20 busca minimizar o tempo de *setup* realizado e a quantidade de peças produzidas. As restrições 3.21 obrigam o atendimento à demanda. Nelas foram desconsideradas o controle de estoque, pois o modelo em estudo não necessita do dimensionamento de lote conforme comentado anteriormente, mas é necessário considerar a quantidade de peças a produzir, uma vez que isto está relacionado diretamente com o planejamento do *setup*.

As restrições 3.22 garantem que sempre que houver uma produção  $x_{ikt} > 0$ , o valor de  $y$  será automaticamente igual a 1. A variável  $y$  é usada para identificar a

necessidade de *setup*. As restrições 3.23 impõem o limite de uma peça por período. Esta mudança caracteriza a alteração do modelo original de várias peças por período para uma peça por período. O controle de uma peça por período cria a possibilidade de saber quando uma peça está sendo feita, impedindo assim que a mesma ocupe outra máquina durante o mesmo período, que é uma restrição forte do problema. Note-se que no modelo proposto muda-se o controle da variável  $x$ , quantidade de peças, para a variável  $y$ , *status* se uma peça é produzida no período  $t$ .

Nas restrições 3.24 o  $\omega$  é forçado a ser 1 se algum item for produzido no período  $t$ . As restrições 3.25 garantem que se mais do que um produto for produzido no período, então  $\delta$  será positivo. As restrições 3.26 e 3.27 garantem qual produto foi produzido primeiro e em último em cada período. As restrições 3.28 e 3.29 fazem com que  $\alpha$  e  $\beta$  sejam exatamente iguais a 1 se um produto for produzido em um período. As restrições 3.30 garantem que  $\delta$  seja igual a zero quando somente um produto é produzido no período  $t$ . As restrições 3.31 fazem com que somente uma máquina esteja preparada para uma peça em um período  $t$ . As restrições 3.32 e 3.33 são aplicadas quando mais do que um produto for produzido em um período. Estas restrições forçam que ao menos um  $T_{ijk}$  seja 1, para cada  $i$ , exceto quando este produto for o primeiro ou o último na sequência. As restrições 3.34 a 3.37 definem o domínio das variáveis. As restrições 3.38 garantem que não haverá *setup* quando uma máquina continuar preparada para uma mesma peça.

As restrições 3.39 atendem à restrição de que existem peças que não podem ser produzidas ao mesmo tempo (peças incompatíveis), mesmo em máquinas diferentes. Esta restrição é representada pela variável  $R_{ij}$  tal que  $R_{ij} = 1$ , se as peças  $i$  e  $j$  não podem ser produzidas ao mesmo tempo, e  $R_{ij} = 0$ , caso contrário. As restrições 3.40 e 3.41 garantem que não haverá mais do que uma peça produzida em um mesmo período. As restrições 3.42 evitam que exista espaços vazios no programa de entrega. As restrições 3.43 obrigam a ocorrência do *setup* inicial.

Para os testes computacionais com o Modelo II foi utilizado o solver Xpress com os dados mostrados nas Tabelas 20, 21, 22 e 23. Para estes testes foram consideradas 3 prensas.

Tabela 20 – *Setup* de uma peça para outra

		Peça		
		1	2	3
Peça	1	0	3	5
	2	3	0	4
	3	5	4	0

Tabela 21 – Demanda por peça

Peça		
1	2	3
2	2	2

Tabela 22 – Incompatibilidade entre peças

		Peça		
		1	2	3
Peça	1	1	1	0
	2	1	1	0
	3	0	0	1

Tabela 23 – *Setup* inicial por peça

Peças		
1	2	3
7	8	9

Os resultados obtidos podem ser verificados nas Tabelas 24 e 25.

Tabela 24 – Resultados obtidos com os Modelos I e II

	Modelo	
	I	II
<i>Setup</i> inicial	16	16
<i>Setup</i> intermediário	3	3
Tempo de execução(s)	0,1	0,8

Tabela 25 – Sequenciamentos obtidos com os Modelos I e II

		Modelo I				Modelo II			
		Período				Período			
		1	2	3	4	1	2	3	4
Prensa	1	1	1	2	2	1	1	2	2
	2	3	3			3	3		
	3								

Os resultados deste teste mostram que o tempo de execução do Modelo II é 8 vezes maior do que o do Modelo I. O sequenciamento mostrado na Tabela 25 foi idêntico para os dois modelos e respeita a restrição que impede que as peças 1 e 2 sejam produzidas ao mesmo tempo. A peça 1 foi colocada em primeiro lugar na sequência pois seu *setup* inicial é menor do que o da peça 2.

Após este teste inicial uma nova bateria de testes foi realizada. Nestes testes fixou-se o tempo limite de execução em 7200 segundos. Isso se deve ao fato de que o tempo para solução de instâncias maiores é muito alto. A Tabela 26 apresenta os resultados dos testes realizados.

NM - Número de máquinas;

NP - Número de peças;

NL - Número de linhas do modelo;

NC - Número de colunas do modelo;

Gap - Valor percentual do *gap* de dualidade (retornado pelo *solver*); Setup - Tempo total de *setup* necessário;

FO - Valor da função-objetivo;

CPU – Tempo de execução (em segundos).

Tabela 26 – Resultados dos novos testes computacionais

		Modelo I						Modelo II					
NM	NP	NL	NC	Gap	Setup	FO	CPU	NL	NC	Gap	Setup	FO	CPU
3	3	804	318	0	9	22	0,7	1614	690	0	7	20	1,0
3	4	1499	546	0	9	27	1,6	2509	1020	0	9	27	7,7
3	5	2422	840	0	11	33	2,8	3632	1410	0	11	33	40,8
3	6	3573	1200	0	14	41	58,6	4983	1860	0	14	41	321,4



3	7	5992	1926	7,16	19	50	7200	7924	3075	17	16	47	7200
3	8	7937	2508	9,68	24	60	7200	10109	3792	32,65	28	64	7200
3	9	11922	3660	6,7	19	78	7200	15819	5733	20,69	18	58	7200
3	10	19249	5724	10,81	27	72	7200	19267	6810	34,49	33	78	7200
3	11	23648	6918	12,98	33	82	7200	27806	9579	24,66	24	73	7200
3	12	28107	8226	8,25	26	80	7200	32805	11088	39,98	46	100	7200
3	13	36954	10668	13,33	42	100	7200	42574	14121	33,33	38	96	7200
3	14	54094	15321	15,85	53	116	7200	59111	19254	37,15	45	106	7200
3	15	62352	17580	15,63	56	123	7200	70377	22605	30,93	30	97	7200
3	20	135407	37230	12,77	59	149	7200	148037	45120	32,39	52	142	7200
3	25	285782	76920	16,11	92	204	7200	306622	90165	41,71	89	201	7200

Pela análise dos resultados obtidos, verifica-se que com 3 peças e 3 máquinas o Modelo II obtém um resultado melhor que o Modelo I. Isto se deve ao fato de que o Modelo II tem a capacidade de preservar a preparação de máquina mesmo sem obrigatoriamente ter uma produção, enquanto que no Modelo I esta característica não existe. Tal fato é inerente aos problemas que originaram estes dois modelos: o Modelo I teve origem em um DLSP enquanto que o Modelo II teve origem em um CLSP. O CLSP preserva o *setup* da máquina, enquanto que o DLSP obrigatoriamente precisa produzir em todos os períodos.

A partir de 7 peças, ambos os modelos não conseguem obter a solução ótima dentro do tempo máximo programado para execução, por isso o *gap* de dualidade é não nulo. Segundo Rosa e Souza (2011), o *gap* mostrado pelo *solver* é a diferença percentual entre o valor da melhor solução viável encontrada (limite superior) e o limite inferior encontrado, é ou seja:

$$\text{Gap} = 100\% \times (\text{Limite superior} - \text{Limite inferior}) / (\text{Limite inferior})$$

Deve-se observar, no entanto, que para a comparação dos Modelos I e II, o Gap não é uma boa medida, visto que as funções-objetivo não são as mesmas.

### 3.7 - Modelo III: minimização do *makespan*

Lenstra *et al.* (1977) demonstram que o problema de minimização do *makespan* em máquinas paralelas é NP-difícil. Este problema pode ser considerado como o problema do caixeiro viajante e sua solução exata só é possível às custas de um grande esforço computacional.

Alguns trabalhos serviram de base para o desenvolvimento do modelo proposto nesta seção. Tahar *et al.* (2006) apresentam um modelo matemático para a solução de um problema com máquinas paralelas idênticas com *setup* dependente da sequência e *split* de tarefas. Dastidar e Nagi (2005) desenvolveram um modelo para problema com máquinas não relacionadas e com *setup* dependente da sequência. Este modelo serviu como base para métodos de relaxação, visando encontrar soluções aproximadas para exemplares de médio e grande porte. Rocha *et al.* (2008) também apresentam um modelo para máquinas paralelas não relacionadas com *setup* dependente da sequência. Nenhum destes trabalhos, no entanto, trata da restrição de produção de peças que dividem o mesmo recurso.

Com base nas idéias destes trabalhos, procurou-se modificar o Modelo I visando a minimização do *makespan*. Duas alterações foram necessárias. A primeira é na função objetivo, que deve ser mudada de uma minimização dos tempos de *setup* e produção para a minimização do *makespan*. O *makespan* é representado no modelo por meio de uma nova variável,  $C_{max}$ .

$$\text{Minimizar } C_{max} \tag{3.44}$$

$$C_{max} \geq \sum_{\substack{i \in I \\ k \in K \\ p \in P}} P t_i z_{ikp} + \sum_{\substack{i, j \in I \\ i \neq j \\ k \in K \\ p \in P}} S_{ij} w_{ijkp} + \sum_{\substack{i \in I \\ k \in K}} S t_i z_{ik1} \tag{3.45}$$

A função objetivo 3.44 passa então a ser o máximo da soma do tempo de produção mais os tempos de *setup* entre as peças e o *setup* inicial. Mas apenas esta mudança não basta, a condição de minimizar o *makespan*, que é previsto na restrição

3.45, pode levar a um *split* (divisão da produção de uma peça entre várias máquinas). Para que isso seja evitado foram incluídas as restrições 3.46, que impedem que uma peça seja produzida em mais de uma máquina.

$$z_{i,k,p} + z_{j,k',p} \leq 1 \quad \forall i, j \in I, k, k' \in K, p \in P ; i \neq j; k \neq k' \quad 3.46$$

Para o Modelo III, alguns testes computacionais foram realizados. Os dados utilizados nestes testes podem ser vistos no Apêndice E. Os resultados destes testes são mostrados na Tabela 27. Nesta tabela, as legendas das colunas são as seguintes:

NM - Número de máquinas (prensas);

NP - Número de peças;

CPU - Tempo de execução (em segundos);

NL - Número de linhas (restrições) do modelo;

NC - Número de colunas (variáveis) do modelo;

Gap - Valor percentual do *gap* de dualidade (retornado pelo *solver*);

FO - Valor da função-objetivo.

Tabela 27 - Resultados obtidos com o Modelo III – Solver Xpress

NM	NP	CPU	NL	NC	Após pré-processamento		Gap	FO
					NL	NC		
3	3	4,7	9753	661	2210	581	0	13
	4	14,4	13718	1123	3153	1027	0	13
	5	40,9	18073	1717	4324	1600	0	14
	6	680,9	22818	2443	5564	2297	0	14
	7	7200,0	27953	3301	6907	3122	27,7	18
	8	7200,0	37006	4291	11845	4074	28,6	21
	9	7200,0	54027	6151	17362	5878	40,0	25

Os testes computacionais foram conduzidos mantendo-se 3 prensas (máquinas), por ser este o ambiente de produção mais provável para o problema em estudo. A quantidade de peças variou de 3 a 10. O tempo de execução foi limitado a 7200 segundos. Nota-se que a partir de 7 peças o *solver* já não consegue resolver o modelo até a otimalidade dentro desta limitação de tempo. Com 10 peças o *solver* não

consegue chegar a nenhuma solução viável dentro do limite de tempo estabelecido. A Figura 3 apresenta os gráficos de Gantt com o sequenciamento das peças correspondentes às soluções obtidas para estes exemplares do problema. Nesta figura, a coluna "m" indica o índice da máquina, o símbolo "i" representa *setup* inicial para a respectiva peça e o símbolo "s" representa o tempo de *setup* entre as peças em questão.

NP	m	Períodos																								
		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
3	1	i	i	2	2	2	2	2																		
	2	i	i	i	i	3	3	3	3	s	1	1	1	1												
	3																									
4	1	i	i	i	i	3	3	3	3	s	1	1	1	1												
	2	i	i	i	4	4	4	4	4																	
	3	i	i	2	2	2	2	2																		
5	1	i	i	i	4	4	4	4	4	s	s	1	1	1	1											
	2	i	i	5	5	5	5																			
	3	i	i	2	2	2	2	2	s	s	3	3	3	3	3											
6	1	i	i	i	i	6	6	6	6	6	s	1	1	1	1											
	2	i	i	2	2	2	2	2	s	s	3	3	3	3	3											
	3	i	i	5	5	5	5	s	s	s	6	6	6	6	6											
7	1	i	i	2	2	2	2	2	2	2	s	s	s	1	1	1	1									
	2	i	i	i	i	6	6	6	6	6	6	s	s	s	4	4	4	4	4							
	3	i	i	5	5	5	5	s	s	7	7	7	7	s	s	3	3	3	3							
8	1	i	i	5	5	5	5	s	s	7	7	7	7	7	7	s	s	6	6	6	6	6				
	2	i	i	2	4	4	4	s	s	2	2	2	2	s	s	4	4	4	4							
	3	i	i	8	8	8	8	8	s	s	3	3	3	3	3	3	3	s	1	1	1	1				
9	1	5	5	5	5	5	5	7	7	2	7	7	2	2	2	2	2	2	2	2	2	2	7	7	7	7
	2	6	9	9	9	9	9	9	1	6	1	1	6	6	6	6	6	6	6	6	6	6	1	1	1	1
	3	8	8	8	8	8	8	8	4	4	4	4	4	4	4	4	8	8	4	4	3	3	3	3	3	3

Figura 3 – Sequenciamentos obtidos com o Modelo III

A partir de 9 peças o modelo já não consegue obter soluções viáveis coerentes, descartando o *setup* inicial e os *setups* intermediário.

#### 4. HEURÍSTICA PROPOSTA

Problemas de sequenciamento, como qualquer problema de Otimização Combinatória, podem ser resolvidos por métodos enumerativos, métodos de aproximação e métodos heurísticos. Os métodos enumerativos, como os usados pelos *solvers* Xpress e CPLEX usados no capítulo anterior, são métodos capazes de obter a solução ótima do problema, porque enumeram (explícita ou implicitamente) todas as soluções viáveis possíveis e, portanto, escolhe a melhor. Estes métodos exigem um esforço computacional muito grande e, em geral, são aplicados apenas a pequenos exemplares dos problemas. Assim, os métodos enumerativos, em geral, não são aplicáveis a problemas práticos reais, que costumam ser de grandes dimensões.

Os métodos heurísticos, por outro lado, procuram explorar o espaço de soluções de um problema de otimização, buscando encontrar as melhores soluções. Para estes métodos, não existe a garantia de se atingir a solução ótima, mas estes métodos podem ser aplicados a problemas de grandes dimensões.

Para o problema de sequenciamento de tarefas em máquinas com tempo de preparação dependente da sequência de produção, vários métodos de solução já foram desenvolvidos, sejam para solucionar o problema de forma ótima como, por exemplo, nos trabalhos de Coleman (1992), Balakrishnan *et al.* (1999), Zhu e Heady (2000), Tseng e Stafford (2001), Eren (2007), ou para solucionar o problema de forma heurística como, por exemplo, nos trabalhos de Kate *et al.* (1991), Reeves (1993), Nowicki e Smutnicki (1996), Sourd (2005), Low (2005), Ravetti *et al.* (2006), Wang *et al.* (2008), Kampke *et al.* (2011).

Neste capítulo propõe-se um algoritmo heurístico para o problema em estudo. Este algoritmo combina a utilização da heurística de refinamento VND (*Variable Neighborhood Descent*) com a metaheurística ILS (*Iterated Local Search*). Para a implementação deste algoritmo utilizando a linguagem C, contou-se com a colaboração do bolsista de Iniciação Científica Viktor Otto Schwarzmeier.

Segundo Souza (2008), as heurísticas de refinamento, também conhecidas como métodos de busca local, constituem uma família de técnicas baseadas na noção de vizinhança. Para um problema cujo espaço de soluções é denominado por  $S$ , uma estrutura de vizinhança  $V$  associa a cada solução  $s \in S$ , um conjunto de soluções vizinhas  $V(s) \subseteq S$ . Para uma heurística de refinamento, denomina-se movimento a transformação de uma solução  $s$  em outra solução  $s' \in V(s)$ . A ideia de uma heurística de refinamento é partir de uma solução inicial qualquer e, a cada iteração, visitar as soluções vizinhas obtidas de acordo com uma dada estrutura de vizinhança. O método pára quando um ótimo local é encontrado. Diferentes soluções iniciais podem conduzir a diferentes soluções finais, conforme mostra a Figura 4, em que  $s$  indica um ponto de partida e  $s^*$  um ótimo local encontrado. Entenda-se por um ótimo local a inexistência de uma solução melhor na vizinhança, o que não significa a melhor solução global para o problema.

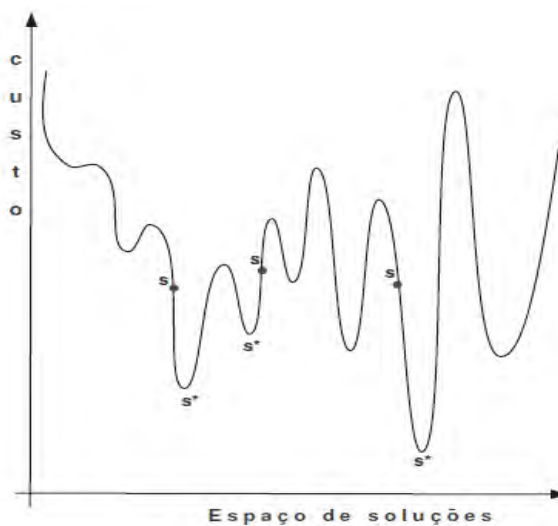


Figura 4 - Comportamento de heurística de refinamento (Fonte: Souza, 2008)

O método VND (*Variable Neighborhood Descent*), ou Descida em Vizinhança Variável, proposto por Mladenovic e Hansen (1997), é uma heurística de refinamento que consiste em explorar o espaço de soluções por meio de trocas sistemáticas de heurísticas de busca local, estabelecidas por diferentes estruturas de vizinhança, aceitando somente soluções de melhora da solução corrente e retornando à primeira

heurística de busca local quando uma solução melhor é encontrada. A Figura 5 apresenta os passos gerais do algoritmo VND.

#### **Algoritmo VND**

1. Seja  $s$  a solução corrente;
2. Seja  $\{ V_1, V_2, \dots, V_N \}$  o conjunto de estruturas de vizinhança;
3.  $k = 1$ ; // Estrutura de vizinhança corrente
4. Enquanto ( $k \leq N$ ) fazer:
  5. Encontrar o melhor vizinho  $s' \in V_k(s)$ ;
  6. Se ( $f(s') < f(s)$ ) então:
    7.  $s = s'$ ;
    8.  $k = 1$ ;
  9. senão:
    10.  $k = k + 1$ ;
  11. fim-se;
12. fim-enquanto;
13. Retorne  $s$ ;
14. fim-algoritmo.

Figura 5 - Heurística VND

As metaheurísticas são procedimentos para encontrar uma boa solução de um problema de otimização, em que, a cada passo, aplica-se uma heurística subordinada, desenvolvida especificamente para o problema a ser resolvido. As metaheurísticas são procedimentos de caráter geral e dispõem de diferentes mecanismos para tentar escapar de ótimos locais (Sucupira, 2004).

O método ILS (*Iterated Local Search*), ou Busca Local Iterativa, proposto por Stützle *et al.* (2006), baseia-se na ideia de que um algoritmo de busca local pode obter melhores soluções gerando-se novas soluções de partida, as quais são obtidas por meio de perturbações na solução ótima local. O algoritmo ILS compreende, basicamente, quatro componentes: (a) um procedimento que gera uma solução inicial  $s_0$  para o

problema; (b) um procedimento de busca local; (c) um procedimento de perturbação, que modifica a solução corrente  $s$ , gerando uma nova (talvez pior) solução  $s'$ ; (d) um critério de seleção, que decide qual será a próxima solução a ser perturbada. A Figura 6 apresenta os passos gerais do algoritmo ILS.

### Algoritmo ILS

1.  $s_0 = \text{GeraSolucaoInicial}();$
2.  $s = \text{BuscaLocal}(s_0);$
3. Enquanto (os critérios de parada não estão satisfeitos) fazer:
  4.  $s' = \text{Perturbacao}(s);$
  5.  $s'' = \text{BuscaLocal}(s');$
  6.  $s = \text{CriterioSelecao}();$
7. fim-enquanto;
8. fim-algoritmo.

Figura 6 - Algoritmo ILS

O desempenho do método ILS depende do mecanismo de perturbação, que deve ser forte o suficiente para permitir a diversificação (e assim escapar de um ótimo local) e fraca o suficiente para permitir a intensificação nas vizinhanças da melhor solução encontrada ou do ótimo local corrente. A Figura 7 mostra a ação de uma perturbação no processo de busca.

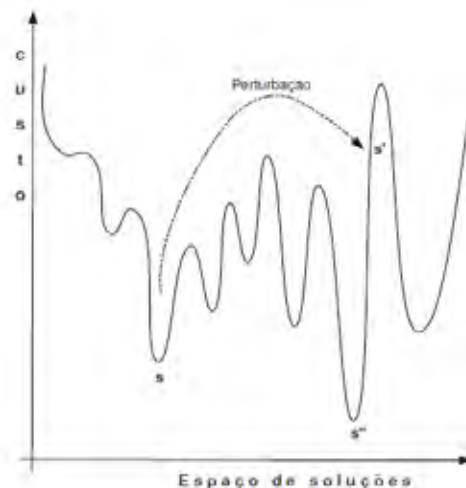


Figura 7 - Processo de perturbação (Fonte: Souza, 2008)



#### 4.1 - O algoritmo heurístico híbrido proposto

O algoritmo heurístico proposto compreende uma adaptação da metaheurística ILS, que faz uso da heurística VND. A Figura 8 apresenta os passos gerais do algoritmo proposto

##### Algoritmo heurístico híbrido

1.  $s = \text{CriarSolucaoInicial}();$
2.  $s = \text{ILS}(s);$
3. Retornar  $s;$
4. fim-algoritmo.

Figura 8 - Algoritmo proposto

A representação da solução do problema utiliza uma matriz, em que as linhas representam as máquinas (prensas) e as colunas, os possíveis de sequenciamento de peças. A Figura 9 ilustra uma possível solução de um exemplar do problema com 3 máquinas e 8 peças.

1	4	5
2	3	
6	7	8

Figura 9 – Representação de solução

No exemplo da Figura 9 tem-se que as peças 1, 4 e 5 devem ser processadas, nesta ordem, na máquina 1; as peças 2 e 3 devem ser processadas, nesta ordem, na máquina 2; e as peças 6, 7 e 8 devem ser processadas, nesta ordem, na máquina 3.

Para o cálculo da função-objetivo, deve-se considerar, além dos tempos de processamento e de *setup*, a compatibilidade entre as peças. Se houver incompatibilidade entre duas peças que ocupam o mesmo período, uma delas deve ser

postergada de forma a não ser produzida ao mesmo tempo. Esta situação pode ser verificada na Figura 10, em que considera-se que as peças 1 e 2 não podem ser produzidas ao mesmo tempo, e as peças 3 e 7 também não podem ser produzidas ao mesmo tempo.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	4	4	4	5	5	5	5						
		2	2	2	3	3								
6	6	6					7	7	7	7	8	8	8	8

Figura 10 - Representação de solução viável

A representação da solução viável obtida pelo cálculo da função-objetivo fornece o valor do *makespan*. No exemplo mostrado na Figura 10, o *makespan* é igual a 15.

As Figuras 11, 12 e 13 mostram os passos gerais de cada um dos algoritmos que compõem o algoritmo proposto.

#### **CriarSolucaoInicial**

1.  $j = 1$ ;
2. Enquanto ( $j \leq NP$ ) fazer: //  $NP = \text{Número de peças}$ 
  3.  $i = \text{Selecionar uma máquina, aleatoriamente}$ ;
  4. Seja  $k$  a primeira posição disponível na máquina  $i$ ;
  5.  $s[i][k] = j$ ;
  6.  $j = j + 1$ ;
7. fim-enquanto;
8. retornar  $s$ ;
8. fim-algoritmo.

Figura 11 – Criação da solução inicial

**VND(s)**

1.  $k = 1$ ;
2. Enquanto ( $k \leq 3$ ) fazer:
  3.  $s' = s$ ;
  4. Se ( $k = 1$ ) então:
    5.  $s = \text{TrocaSimples}(s)$ ;
  6. senão:
  7. Se ( $k = 2$ ) então:
    8.  $s = \text{Intercambio}(s)$ ;
  9. senão:
    10.  $s = \text{Realocacao}(s)$ ;
  11. fim-se;
  12. Se ( $\text{CustoSolucao}(s) < \text{CustoSolucao}(s')$ ) então:
    13.  $k = 1$ ;
    14.  $s = s'$
  15. senão:
    16.  $k = k + 1$ ;
  17. fim-se;
18. fim-enquanto;
19. Retornar  $s$ ;
20. fim-algoritmo.

Figura 12 – Procedimento VND

Da Figura 12 nota-se que a heurística VND utiliza as seguintes heurísticas:

- **Troca Simples:** a partir de uma solução inicial, considera-se todas as trocas possíveis de posições de duas tarefas quaisquer alocadas em uma mesma máquina;
- **Intercâmbio:** a partir de uma solução inicial, considera-se todas as possíveis trocas de duas tarefas alocadas em máquinas diferentes;
- **Realocação:** a partir de uma solução inicial, considera-se todas as possíveis trocas de uma tarefa da máquina onde estava alocada para outra máquina capaz de atendê-la.

**ILS(s)**

1.  $s = \text{VND}(s)$
2.  $s^* = s$
3. Enquanto ( $\text{iter} < \text{MAXITER}$ )
  4.  $\text{iter} = \text{iter} + 1$ ;
  5.  $c_{\text{Inicial}} = \text{CustoSolucao}(s)$ ;
  6.  $s_{\text{Temp}} = s$ ;
  7.  $\alpha = \text{Random}(0.2, 0.6)$
  8.  $\beta = \text{Max}(1, \text{INT}(\alpha * \text{NP}))$  // NP= Número de peças
  9. Para  $i = 1$  até  $\beta$ 
    10.  $s = \text{Perturbacao}(s)$ ;
    11.  $s = \text{VND}(s)$ ;
    12.  $c_{\text{Final}} = \text{CustoSolucao}(s)$ ;
    13. If ( $c_{\text{Final}} < c_{\text{Inicial}}$ )
      14.  $s^* = s$ ;
    15. senão
      16.  $s = s_{\text{Temp}}$ ;
17. Fim-enquanto

Figura 13 – Metaheurística ILS

A perturbação considerada na metaheurística ILS corresponde a uma única realocação, em que considera-se as possíveis trocas de posição de uma tarefa de uma máquina de origem para uma outra máquina de destino. Nesta função de perturbação, a tarefa e a máquina de destino são escolhidas aleatoriamente.

#### 4.2 - Testes computacionais

Inicialmente foram realizados com o algoritmo heurístico os mesmos testes realizados com o Modelo III, visando mostrar a qualidade da solução heurística. Os dados utilizados nestes testes podem ser vistos no Apêndice E. Os resultados destes testes são mostrados nas Tabelas 28 e 30.

Na tabela 28 são mostrados os resultados obtidos pelo algoritmo heurístico em comparação com os obtidos pelo Modelo III. Nesta tabela, tem-se:

NM – Número de máquinas

NP – Número de peças

CPU – Tempo de execução (em segundos)

FO – Valor da função objetivo (*Makespan*)

NL – Número de linhas

NC – Número de colunas

Gap - Valor percentual do *gap* de dualidade (retornado pelo *solver*)

Tabela 28 - Resultados preliminares obtidos com o algoritmo heurístico

		Heurística		Modelo III				
NM	NP	CPU	FO	NL	NC	Gap	CPU	FO
3	3	0,05	13	2210	581	0	4,7	13
	4	0,1	13	3153	1027	0	14,4	13
	5	0,15	14	4324	1600	0	40,9	14
	6	0,16	14	5564	2297	0	680,9	14
	7	0,34	18	6907	3122	27,7	7200	18
	8	0,13	19	11845	4074	28,6	7200	21
	9	0,12	19	17362	5878	40	7200	25
	10	0,13	23	*	*	*	*	*

Pelos resultados comparativos mostrados na Tabela 28, nota-se que o algoritmo heurístico obteve as mesmas soluções ótimas obtidas com o Modelo III, para os exemplares com 3 a 7 peças, mas em um tempo computacional muito menor. Para os demais exemplares, o *solver* não conseguiu obter as soluções ótimas usando o Modelo III, mas o algoritmo heurístico obteve soluções melhores. Estes resultados podem ser entendidos como uma indicação de que o algoritmo heurístico proposto é correto e eficaz.

A Figura 14 apresenta os gráficos de Gantt com o sequenciamento das peças correspondente às soluções obtidas pelo algoritmo heurístico para estes exemplares do

problema. Nesta tabela o símbolo "i" representa *setup* inicial para a respectiva peça e o símbolo "s" representa o tempo de *setup* entre as peças em questão.

		Períodos																								
Peças	Prensa	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
3	1	i	i	2	2	2	2	2																		
	2	i	i	i	i	3	3	3	3	s	1	1	1	1												
	3																									
4	1	i	i	i	i	3	3	3	3	s	1	1	1	1												
	2	i	i	2	2	2	2	2																		
	3	i	i	i	4	4	4	4	4																	
5	1	i	i	5	5	5	5																			
	2	i	i	i	4	4	4	4	4	s	s	1	1	1	1											
	3	i	i	2	2	2	2	2	2	s	s	3	3	3	3											
6	1	i	i	5	5	5	5	s	s	s	6	6	6	6	6											
	2	i	i	2	2	2	2	2	s	s	s	1	1	1	1											
	3	i	i	i	4	4	4	4	4	s	s	3	3	3	3											
7	1	i	i	i	i	6	6	6	6	6	s	s	7	7	7	7										
	2	i	i	5	5	5	5	s	s	s	3	3	3	3	s	1	1	1	1							
	3	i	i	2	2	2	2	2	s	s	4	4	4	4	4											
8	1	i	i	8	8	8	8	8	s	s	6	6	6	6	6	1	1	1	1	1						
	2	i	i	5	5	5	5	s	s	s	3	3	3	3	7	7	7	7	7	7						
	3	i	i	2	2	2	2	2	s	s	4	4	4	4	4											
9	1	i	i	5	5	5	5	s	s	7	7	7	7	s	s	4	4	4	4	4						
	2	i	i	2	2	2	2	2	s	s	9	9	9	9	s	s	3	3	3	3						
	3	i	i	8	8	8	8	8	s	s	6	6	6	6	6	s	1	1	1	1						
10	1	i	i	2	2	2	2	2	s	s	4	4	4	4	4	s	s	s	6	6	6	6	6			
	2	i	i	8	8	8	8	8	s	s	9	9	9	9	9	s	s	10	10	10	10	10				
	3	i	i	5	5	5	5	s	s	7	7	7	7	s	s	1	1	1	1	s	3	3	3	3		

Figura 14 – Sequenciamentos obtidos com o algoritmo heurístico

A fim de demonstrar o desempenho do algoritmo heurístico para a solução de exemplares mais realistas do problema, foram executados testes computacionais para 16 exemplares construídos com base nos dados obtidos pela empresa para a qual este estudo foi realizado. Estes exemplares consideram de 25 a 200 peças e 3 máquinas. Como o algoritmo heurístico possui elementos aleatórios, os testes computacionais consideraram 20 execuções para cada exemplar. O objetivo deste teste é demonstrar

que apesar da aleatoriedade, o algoritmo proposto apresenta um bom comportamento. A Tabela 30 mostra os resultados obtidos. Esta tabela mostra, para cada exemplar do problema, o custo (*makespan*) da melhor solução encontrada, a solução média e o tempo de execução (CPU) em segundos. Para estes testes computacionais, impôs-se um limite de 1800 segundos para o tempo de execução.

Tabela 30 - Resultados obtidos com o algoritmo heurístico

Exemplar	Melhor solução	Solução média	CPU
p025	2019	2021,5	534,5
p030	1697	1699,5	566,2
p035	2590	2592,8	1438,9
p040	3742	3745,9	1800,0
p045	4973	4976,5	1800,1
p050	4059	4063,7	1800,0
p055	4515	4523,2	1800,0
p060	6955	6961,0	1800,0
p070	6334	6343,8	1800,0
p080	7893	7903,2	1800,0
p090	8753	8763,0	1800,0
p100	9159	9168,6	1800,0
p120	12737	12746,5	1800,0
p150	14409	14419,4	1800,0
p170	16604	16616,8	1800,1
p200	18679	18697,2	1800,1

Para uma validação final, executou-se o algoritmo heurístico para uma base de dados real, visando comparar a solução obtida pelo algoritmo heurístico com a solução empregada hoje (manual). A base de dados foi de 194 produtos com uma demanda total de 36323 peças para 3 prensas. O resultado obtido pelo algoritmo heurístico corresponde a um *makespan* de 10710,40, em média, com 10585 sendo o melhor resultado encontrado, contra um resultado de 12446 obtido atualmente, ou seja, o algoritmo heurístico obtêm uma solução 16% melhor em média contra a solução utilizada atualmente na empresa. Este resultado mostra a eficiência do algoritmo heurístico desenvolvido, o que viabiliza sua utilização em um ambiente de produção que é muito dinâmico. Os resultados desta execução podem ser encontrados no Apêndice F.

## 5. CONCLUSÃO

O problema aqui analisado, sequenciamento de produção em máquinas paralelas com tempo de *setup* dependente da sequência, com restrição de recursos, no caso, uma peça não poder ser produzida ao mesmo tempo que algumas outras, é comum a uma grande quantidade de empresas. Existe no mercado grande número de *softwares* aplicativos que prometem a solução desse problema, mas devido à falta de conhecimento dos gestores da dimensão do problema gerado, muitas implantações vêm a falhar.

Neste trabalho explora-se as possibilidades de solução exata, através da modelagem matemática, e um método heurístico híbrido utilizando as técnicas VND (*Variable Neighborhood Descent*) e ILS (*Iterated Local Search*). Para testar a modelagem matemática fez-se uso do *software* Xpress utilizando a linguagem Mosel e também do *software* CPLEX.

O modelo matemático desenvolvido primeiramente, denominado Modelo I, é baseado em um DLSP (*Discret Lotsizing Scheduling Problem*), o qual tem por característica a ocupação total de cada período de tempo disponível, usando o conceito de *small bucket*. Este modelo mostrou-se eficiente para minimização geral de todos os tempos envolvidos na produção do produto nos testes realizados, gerando soluções exatas para o problema. Este modelo foi comparado a outro já desenvolvido por Gupta e Magnussom (2005), adaptado aqui como Modelo II, que tem por característica ser baseado em um CLSP (*Capacitated Lotsizing Scheduling Problem*), o qual trabalha usando uma característica de *big bucket* e permite que um determinado período seja compartilhado pela produção de mais do que um produto. Sua maleabilidade implica em uma maior complexidade na modelagem. O Modelo I utiliza um número menor de variáveis do que o Modelo II, sendo, conseqüentemente, mais rápido. E tem como particularidade o uso completo do período disponível. Isso faz com que a produção seja maior que o necessário, aumentando assim os estoques. O Modelo II é mais flexível no trato das quantidades produzidas, uma vez que a produção é representada



por uma variável contínua. Outra característica importante é o fato de aceitar períodos vazios com preservação de *setup*. Esta característica de todos os CLSP pode significar um ganho na qualidade do resultado final

A minimização de todos os tempos envolvidos em produção não garante que o melhor *makespan* possível seja atingido quando se trata de máquinas paralelas. Isto pode ser verificado quando se leva em consideração mesmo que os tempos de produção não variem de uma máquina para outra, mas o *setup* depende da sequência de produção. Por isso, em uma segunda fase desenvolve-se um modelo visando a minimização do *makespan*, denominado aqui Modelo III. Este modelo é uma adaptação do Modelo I com a entrada de uma nova variável e o controle do *split* de produção nas peças. Este modelo é comparado com uma heurística desenvolvida em linguagem C, baseado nas metaheurísticas VND e ILS, conforme citado anteriormente. Pela análise do resultado de ambos sob a mesma base de dados pode-se validar estas soluções para o problema proposto.

Devido ao grande número de variáveis e suas combinações, a modelagem para este problema não pode ser validada para instâncias médias e grandes. As restrições de concorrência, onde uma peça não pode ser produzida ao mesmo tempo que outra que compartilha o mesmo ferramental, faz com que o controle da produção seja feito em cada período, pois o conteúdo deste deve ser comparado com o mesmo período em outra prensa o que aumenta o número de variáveis e combinações a serem verificadas pelo programa. Sendo assim, seu uso efetivo fica inviabilizado, o que não ocorre com o método heurístico, capaz de resolver uma instância real do problema em cerca de 30 minutos de execução e obter resultado melhor do que o processo padrão utilizado na empresa atualmente.

Para desenvolvimento futuro, sugere-se considerar na formulação do problema a data de entrega do produto e a dimensão do lote de produção, tendo em vista que a tendência do mercado é a criação de lotes cada vez menores de produção tendo como objetivo diminuir os estoques e atender às demandas criadas a partir da metodologia

JIS (*Just in Sequence*). Neste contexto deve-se levar em consideração o fato que o processo de produção utilizando prensas de alta capacidade tem alto custo de *setup*, o que influencia o tamanho do lote a ser produzido.

Outra possibilidade para futuras pesquisas parte da ideia de se usar a metaheurística para criação de uma solução inicial para o CPLEX ou o Xpress. Isto poderá melhorar a performance do *solver*, pois ele partirá de uma boa solução, poupando-lhe grande esforço, possibilitando encontrar a solução ótima dentro de limitações de tempo razoáveis.

## Referências Bibliográficas

- Absi, N.; Kedad-Sidhoum, S. The multi-item capacitated lot-sizing problem with setup times and shortage cost. *European Journal of Operational Research*, v. 185, p. 1351-1374, 2008.
- Allahverdi, A.; Gupta, J.N.D.; Aldowaisan, T. A review of scheduling research involving setup considerations. *Omega - The International Journal of Management System*, v. 27, p. 219-239, 1999.
- Araújo, S.A.; Arenales, M.N. Problema de Dimensionamento de Lotes Monoestágio com Restrição de Capacidade: Modelagem, Método de Resolução e Resultados Computacionais. *Pesquisa Operacional*, v. 20, n. 2, p. 287-306, 2000.
- Arenales, M.; Armentano, V.; Morabito, R.; Yanasse, H. *Pesquisa Operacional para Cursos de Engenharia*, Rio de Janeiro: Elsevier, 2007.
- Balakrishnan, N.; Kanet, J.J.; Sridharan, S.V. Early/tardy scheduling with sequence dependent setups on uniform parallel machines. *Computers and Operations Research*, v. 26, p. 127-141, 1999.
- Barros, A.D.; Moccelin, J.V. Análise da Flutuação do Gargalo em Flow Shop Permutacional com Tempos de Setup Assimétricos e Dependentes da Sequência. *Gestão & Produção*, v. 11, n. 1, p. 101-108, 2004.
- Behnamian, J.; Zandieh, M.; Ghomi, S.M.T.F. Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA an VNS hybrid algorithm. *Expert System with Applications*, v. 36, p. 9637-9644, 2009.
- Buschkühl, L.; Tempelmeir, H. Dynamic multi-machine lotsizing and sequencing with simultaneous scheduling of a common setup resource. *International Journal of Production Economics*, v. 113, p. 401-412, 2008.
- Campbell, G.M. Using Short-term Dedication for Scheduling Products on Parallel Machines. *Productions and Operations Management*, v. 1, n. 3, 1992.
- Cerra, A. L.; Bonadio, P. V. G.. As relações entre estratégia de produção, TQM (Total Quality Management ou Gestão da Qualidade Total) e JIT (Just-In-Time): estudos de caso em uma empresa do setor automobilístico e em dois de seus fornecedores. *Gestão da Produção*, São Carlos, v. 7, n. 3, 2000.

- Chen, C.; Chen, C. Hybrid metaheuristics for unrelated parallel machine scheduling with sequence-dependent setup times. *The International Journal of Advanced Manufacturing Technology*, v. 46, p. 161-169, 2009.
- Coleman, B.J. A simple model for optimizing the single machine early/tardy problem with sequence-dependent setups. *Production and Operations Management*, v. 2, p. 225-228, 1992.
- Dastidar, S.G.; Nagi, R. Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and cost. *Computer and Operations Research*, v. 32, p. 2987-3005, 2005.
- Dauzère-Peres, S.; Lasserre, J.B. On the Importance of Sequencing Decisions in Production Planning and Scheduling. *International Transactions in Operational Research*, v. 9, p. 779-793, 2002.
- Davis, S.G. Scheduling Economic Lot Sizing Productions Runs. *Management Science*, v. 36, p. 985-998, 1990.
- Eren, T. A Multicriteria Scheduling with Sequence-Dependent Setup Times. *Applied Mathematical Sciences*, v. 1, n. 58, p. 2883-2894, 2007.
- Feng, H.; Cheng, H. Solving mix integer programming production planning problem with setups by shadow price information. *Computer and Operations Research*, v. 25, p. 1027-1042, 1998.
- Gil, A. C. *Métodos e técnicas de pesquisa social*, São Paulo: Atlas, 5ª ed., 1999.
- Gomes, C.F.S.; Ribeiro, P.C.C. *Gestão da cadeia de suprimentos integrada à tecnologia da informação*. São Paulo: Pioneira Thomson Learning, 2004.
- Gupta, D.; Magnusson, T. The Capacitated Lot-sizing and Scheduling Problem with Sequence-Dependent Setup Cost and Setup Times. *Computers and Operations Research*, v. 32, p. 727-747, 2005.
- Haase, K. *Lotsizing and Scheduling for Production Planning*. Berlin: Springer, 1994.
- Kampke, E.H.; Arroyo, J.E.C.; Santos, A.G. *Grasp Reativo e Iterated Local Search com Path Relinking aplicado ao problema de programação de tarefas com tempos de preparação dependentes da sequência e de recursos*. Simpósio de Pesquisa Operacional e Logística da Marinha, 2011. Disponível em

- <https://www.casnav.mar.mil.br/spolm/artigos/artigos/88363.pdf>. Acessado em 01/09/2011.
- Karimi, B.; Ghomi, S.M.T.F.; 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.
- Kate, H.A.; Wijngaard, J.; Zijm, W.H.M. *Minimizing weighted total earliness, total tardiness and setup costs*. Research Report 12, School of Management and Organization, University of Groningen, Holland, 1991.
- Keachie, E.C.; Fontana, R.J. Production lotsizing under a learning effect. *Management Science*, v. 13, p. 102-108, 1966.
- Laudon, K.C.; Laudon, J.P. *Sistemas de Informações Gerenciais, Administrando a Empresa Digital*. São Paulo: Pearson Prentice Hall, 5ª ed., p. 86-91, 2004.
- Lee, H.Y.; Pinedo, M. Scheduling jobs on parallel machines with sequence-dependent setup times. *European Journal of Operational Research*, v. 100, p. 464-474, 1997.
- Lenstra, J.K.; Johnson, D.S.; Kan, A.H.G.R. The complexity of the Network Design Problem. *Networks*, v. 8, p. 279-285, 1977.
- Lourenço, H.R.; Martin, O.; Stutzle, T. Iterated Local Search. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, p. 321–353, 2002.
- Low, C. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. *Computers and Operations Research*, v. 32, n. 8, p. 2013-2025, 2005.
- Meyer, H.; Fleischmann, B. The general lotsizing and scheduling problem. *Operations Research*, v. 19, p. 11-21, 1997.
- Meyr, H. Simultaneous Lotsizing and Scheduling on Parallel Machines, *European Journal of Operational Research*, v. 139, p. 277-292, 2002.
- Mladenovic, N.; Hansen, P. Variable Neighborhood Search. *Computers and Operations Research*, v. 24, p. 1097-1100, 1997.
- Muller, F.M.; Dias, O.B.; Araújo, O.C. Algoritmo para o problema de sequenciamento em máquinas paralelas não-relacionadas. *Revista Produção*, v. 12, n. 2, 2002.

- Nowicki, E.; Smutnicki, C. A Fast Tabu Search Algorithm for the Permutation Flow-Shop Problem. *European Journal of Operational Research*, v. 91, p. 160-175, 1996.
- Omar, M.K.; Teo, S.C.; Yasothai, S. Scheduling with Setup Considerations: An MIP Approach. *Multiprocessor Scheduling, Theory and Applications*, ISBN 978-3-902613-02-8, December 2007.
- Potts, C.N.; Kovalyov, M.Y. Scheduling with Batch: a Review. *European Journal of Operational Research*, v. 120, p. 228-249, 2000.
- Ravetti, M.G.; Nakamura, F.G.; Meneses, C.N.; Resende, M.G.C.; Mateus, G.R.; Pardalos, P.M. *Hybrid heuristics for the permutation flow shop problem*. AT&T Labs Research Technical Report TD-6V9MEV, Shannon Laboratory, Florham Park, NJ, USA, November 2006.
- Reeves, C. Improving the Efficiency of Tabu Search for Machine Sequencing Problems. *Journal of the Operational Research Society*, v. 44, n. 4, p. 375-382, 1993.
- Rocha, P.L.; Ravetti, M.G.; Mateus, G.R.; Pardalos, P.M. Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times. *Computers and Operations Research*, v. 35, p. 1250-1264, 2008.
- Rosa, B.F.; Souza, M.J.F. *Formulações de Programação Matemática para o Problema de Sequenciamento em uma Máquina com Janelas de Entrega Distintas e Tempo de Preparação Dependente da Sequência de Produção*. Disponível em: <http://www.iceb.ufop.br/decom/prof/marcone/Publicacoes/PSUMAA-CNMAC2009.pdf>. Acessado em: 20/10/2011.
- Sicilia, J.; Colebrook, M.; Sedeño-Noda, A.; Gutiérrez, J. An efficient approach for solving the lot-sizing problem with time-varying storage capacities. *European Journal of Operational Research*, v. 189, p. 682-693, 2008.
- Slack, N.; Johnston, R.; Chambers, S. *Administração da Produção*, 2ª.ed, São Paulo, Atlas, 2002.
- Sourd, F. Earliness-tardiness scheduling with setup considerations. *Computers and Operations Research*, v. 32, p. 1849-1865, 2005.

- Souza, M.J.F. *Inteligência Computacional para Otimização*, Departamento de Computação, Instituto de Ciências Exatas e Biológicas, Universidade Federal de Ouro Preto, Notas de aula, 2008.
- Stoop, P.P.M.; Wiers, V.C.S. The Complexity of scheduling in practice. *International Journal of Operations and Production Management*, v. 116, n. 10, p. 37-53, 1996.
- Stützle, T. Iterated local search for the quadratic assignment problem. *European Journal of Operational Research*, v. 174, n. 3, p. 1519\_1539, 2006.
- Sucupira, I.R. *Métodos Heurísticos Genéricos: Meta-heurísticas e hiper-heurística*; USP Universidade de São Paulo; disponível em <http://www.ime.usp.br/~igorrs/monografias/metahiper.pdf>, 2004.
- Yalaoui, F.; Chu, C.; Amodeo, L. A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times. *International Journal of Production Economics*, v. 99, p. 63-73, 2006.
- Trigueiro, W.W.; Thomas, L.J.; McClain, J.O. Capacitated Lot Sizing with Setup Times. *Management Science*, v. 35, n. 3, p. 353-36, 1989.
- Troque, W.A.; Pires, S.R.I. Influência das práticas da Gestão da Cadeia de Suprimentos na Gestão da Demanda. *XXIII Encontro Nacional de Engenharia de Produção - ENEGEP*, Ouro Preto, MG, 2003. Disponível em: [www.abepro.org.br/biblioteca/ENEGEP2003\\_TR0706\\_1501.pdf](http://www.abepro.org.br/biblioteca/ENEGEP2003_TR0706_1501.pdf).
- Tahar, D.N.; Yalaoui, F.; Chu, C.; Amodeo, L. A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times; *International Journal of Production Economics*, p.63-73, 2006.
- Tseng, F.T.; Stafford, E.F. Two MILP models for the N x M SDST flowshop sequencing problem. *International Journal of Production Research*, v. 39, p. 1777-1809, 2001.
- Wang, C.; Li, X.; Wang, Q. Iterative local search algorithm for no-wait flowshop scheduling problems to minimize makespan. In: *12th International Conference on Computer Supported Cooperative Work in Design*, p. 908-912, 2008.
- Wortham, A.W.; Mayyasi, A.M. Learning considerations with economic order quantity. *AIIE Transactions*, v. 4, n. 1, p. 69-71, 1972.

Zhu, Z.; Heady, R.B. Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach. *Computers and Industrial Engineering*, v. 38, p. 297-305, 2000.



## Apêndice A - Modelo I formulado na linguagem Mosel

```

model 'Programa de prensas - Modelo I'
uses "mmsxprs"; !gain access to the Xpress-Optimizer solver
setparam("XPRS_maxtime",7200)

declarations

I = 1..3           ! Quantidade de peças
K = 1..3           ! Quantidade de prensas
P = 15             ! Número de posições possíveis na sequencia
                  ! de produção, conforme o modelo
                  ! estudado, esta variável será o número de
                  ! peças a produzir.

dem:   array(I)   of integer      ! Demanda das peças
cp:    array(I)   of real         ! Custo de produção das peças
Pt:    array(I)   of real         ! Tpo proces da peça i naprensa k
S:     array(I,I) of real         ! Tpo proces. da peça j após a i
Sti:   array(I)   of real         ! Setup inicial
R:     array(I,I) of real         ! Um ferramental não pode ser
                  ! usado por duas prensas ao mesmo
                  ! tempo como o ferramental é
                  ! único para a peça, a restrição
                  ! por peça á possível
                  ! Assim: 1 = Não podem ser
                  ! produzidas ao mesmo tempo
                  ! 0 = Podem ser produzidas ao
                  ! mesmo tempo

t:     array(K,1..P)   of mpvar   ! Tempo de início da posição p na
                  ! prensa k
z:     array(I,K,1..P) of mpvar   ! Variável binária, onde = 1 se a peça
                  ! i for alocada na posição p da
                  ! sequencia de produção da prensa k ;
                  ! senão será = 0
w:     array(I,I,K,0..P) of mpvar ! Variável binária, onde = 1 se a peça
                  ! foi alocada na posição p e a peça j
                  ! foi alocada na posição (p+1) da

```

```

! sequencia de
produção da prensa k
  y: array(K,1..P)      of mpvar      ! Controle da ocupação de cada posição
impedir espaços vazios

end-declarations

initializations from 'e_0303.txt'
  cp dem Pt S Sti R
end-initializations

! ( 3.4 )
forall(i in I)
  DO
    sum(k in K, p in 1..P) z(i,k,p) = dem(i) +
                                     sum(j in I, k in K, p in 1..P | i
<> j) w(j,i,k,p) * S(i,j) +
                                     sum(k in K) Sti(i) * z(i,k,1)

  END-DO
! --- z/w são binários
! ( 3.7 )
forall(i,j in I,k in K, p in 1..P) w(i,j,k,p) is_binary

! ( 3.8 )
forall(i in I,k in K, p in 1..P) z(i,k,p) is_binary

! ( 3.9 )
!zzzzzz tot_prep_ini := sum(i in I, k in K ) Sti(i) * z(i,k,1)

! ( 3.10 )
forall (p in 1..P, k in K)
  sum(i in I) z(i,k,p) <= 1

! ( 3.11 )
forall (p in 1..P, i in I)
  sum(k in K) z(i,k,p) <= 1

! ( 3.12 )
forall (i,j in I, k in K, p in 1..P-1 | i <> j)
do

```

```

        z(i,k,p) + z(j,k,p+1) - 2 * w(i,j,k,p) >= 0
end-do

! ( 3.13 )
forall (i,j in I, k in K, p in 1..P-1 | i <> j )
do
    z(i,k,p) + z(j,k,p+1) - 2 * w(i,j,k,p) <= 1
end-do

! Restrição ( 3.14 )
! ----- y aponta produção no período na prensa
forall(k in K, p in 1..P)
    y(k,p) = sum(i in I) z(i,k,p)

! Restrição ( 3.15 )
! ----- Y Deve obrigar o setup existir se houver um vazio
forall (k in K, p in 1..P-1)
do
    y(k,p) >= y(k,p+1)
end-do

! Restrição ( 3.16 )
! ----- Um ferramental não pode ser usado ao mesmo tempo que outro
forall (i,j IN I, k , k2 in K, p IN 1..P | i <> j)
do
    if k <> k2 then
        z(i,k,p) + z(j,k2,p) + R(i,j) <= 2
    end-if
end-do

!
!
=====
tot_geral := sum(i in I, k in K, p in 1..P) z(i,k,p)

minimize( tot_geral )

writeln("SOLUÇÃO:")

writeln("-----")

```

```

forall (k in K,p in 1..P, i in I)
  if getsol(z(i,k,p)) > 0.1 and
    getsol(z(i,k,p)) < 1.1 then
    writeln ("z(",i,',',k,',',p,') = ',getsol(z(i,k,p)))
  end-if
writeln("-----")
forall (i,j in I, k in K, p in 1..P)
  if getsol(w(i,j,k,p)) > -1 and
    getsol(w(i,j,k,p)) < 2 and
    getsol(w(i,j,k,p)) <> 0 then
    writeln ("Val do w (",i,',',j,',',k,',',p,") =
",getsol(w(i,j,k,p) )," = ",S(3,1))
  end-if
writeln("-----")
forall (k in K, i in I)
  if getsol(z(i,k,1)) > 0.1 and
    getsol(z(i,k,1)) < 1.1 then
    writeln ("z(",i,',',k,',',1,') = ',getsol(z(i,k,1)), " - Setup inic
= ",Sti(i))
  end-if
writeln("-----")

end-model

```

## Apêndice B – Alterações implementadas no Modelo I

**Problema - Falta de *setup* inicial**

**Restrição ( 3.9 )**

**A variável *w* deve ser redefinida**

```
w: array(I,J,K,0..P) of mpvar      ! Variável binária, onde = 1 se a peça
                                    ! i foi alocada na
                                    ! posição p e a peça j foi alocada
                                    ! na posição (p+1) da
                                    ! sequencia de produção da prensa k
```

! Todo início de produção deve ter um *setup*

```
forall (i,j in I,k in K)
```

```
    w(i,j,k,0) = 1
```

! O *setup* inicial deve ser imposto pois ele é o *setup* da peça partindo  
! do nada.

```
prep_ini := sum(i in I, k in K ) S(i,i) * z(i,k,1)
```

**Restrições (3.12) e (3.13)**

! Se a peça *i* está na posição *p* e a peça *j*, na posição *p+1*,  
! então existe um *setup* (considerado na posição *p*).

```
forall (i,j in I, k in K, p in 1..P-1 | i <> j)
```

```
do
```

```
    z(i,k,p) + z(j,k,p+1) - 2 * w(i,j,k,p) >= 0
```

```
end-do
```

```
forall (i,j in I, k in K, p in 1..P-1 | i <> j)
```

```
do
```

```
    z(i,k,p) + z(j,k,p+1) - 2 * w(i,j,k,p) <= 1
```

```
end-do
```

**Problema: Produção maior que a demanda – Usado recurso de ter custo de produção na função objetivo.**

**Restrição (3.1)**

```
cp: array(I) of real      ! Custo de produção das peças
! ----- Custo total da produção
custo := sum(i in I, k in K, p in 1..P) cp(i) * z(i,k,p)
```

**Problema : Alguns ferramentais não podem ser usados ao mesmo tempo porque eles dividem os mesmos componentes.**

**Restrição (3.16)**

```
R: array(I,I) of real      ! Um ferramental não pode ser usado por
                           ! duas prensas ao mesmo tempo
                           ! como o ferramental é único para a
                           ! peça, a restrição por peça á possível
                           ! Assim: 1 = Não podem ser produzidas
                           ! ao mesmo tempo
                           ! 0 = Podem ser produzidas ao mesmo
                           ! tempo
```

```
! ----- Um ferramental não pode ser usado ao mesmo tempo que outro
```

```
forall (i,j IN I, k , k2 in K, p IN 1..P | i <> j)
do
    if k <> k2 then
        z(i,k,p) + z(j,k2,p) + R(i,j) <= 2
    end-if
end-do
```

## Apêndice C: Modelo II formulado na linguagem Mosel

```

model 'Modelo II'
uses "mmxprs"; !gain access to the Xpress-Optimizer solver
setparam("XPRS_maxtime",7200)
!
! declarations section
declarations

I = 1..9           ! Quantidade de peças
K = 1..3           ! Quantidade de prensas
P = 22             ! Número de posições possíveis na sequencia
                  ! de produção, conforme o modelo
                  ! estudado, esta variável será o número de
                  ! peças a produzir.
G = 999999        ! Número muito grande

dem:      array(I)    of integer  ! Demanda das peças
cp:       array(I) of real        ! Custo de produção das peças
Pt:       array(I)   of real      ! Tpo proces da peça i na prensa k
S:        array(I,I) of real      ! Tpo proces. da peça j após a i
Sti:      array(I)   of real      ! Setup inicial
R:        array(I,I) of real      ! Um ferramental não pode ser usado por
                  ! duas prensas ao mesmo tempo
                  ! como o ferramental é único para a
                  ! peça, a restrição por peça á possível
                  ! Assim: 1 = Não podem ser produzidas
                  ! ao mesmo tempo
                  ! 0 = Podem ser produzidas ao mesmo
                  ! tempo

t:        array(K,1..P)   of mpvar ! Tempo de início da posição p na
                  ! prensa k
z:        array(I,K,1..P) of mpvar ! Variável binária, onde = 1 se a peça
                  ! i for alocada na
                  ! posição p da sequencia de produção da
                  ! prensa k ;
                  ! senão será = 0
w:        array(I,I,K,0..P) of mpvar ! Variável binária, onde = 1 se a peça
                  ! i foi alocada na

```

```

! posição p e a peça j foi alocada na
! posição (p+1) da
! sequencia de produção da prensa k
y: array(K,1..P) of mpvar ! Controle da ocupação de cada posição
! impedir espaços vazios

tot_geral: mpvar
end-declarations

initializations from 'e_0309.txt'
    cp dem Pt S Sti R
end-initializations

! ( 3.4 )
forall(i in I)
    DO
        sum(k in K, p in 1..P) z(i,k,p) = dem(i) +
            sum(j in I, k in K, p in 1..P | i <> j) w(j,i,k,p) * S(i,j) +
            sum(k in K) Sti(i) * z(i,k,1)
    END-DO

! ( 3.7 )
!
forall(i,j in I,k in K, p in 1..P) w(i,j,k,p) is_binary

! ( 3.8 )
forall(i in I,k in K, p in 1..P) z(i,k,p) is_binary

! ( 3.10 )
forall (p in 1..P, k in K)
    sum(i in I) z(i,k,p) <= 1

! ( 3.11 )
forall (p in 1..P, i in I)
    sum(k in K) z(i,k,p) <= 1

! ( 3.12 )
forall (i,j in I, k in K, p in 1..P-1 | i <> j)
do
    z(i,k,p) + z(j,k,p+1) - 2 * w(i,j,k,p) >= 0
end-do

```



```

! ( 3.13 )
forall (i,j in I, k in K, p in 1..P-1 | i <> j )
do
    z(i,k,p) + z(j,k,p+1) - 2 * w(i,j,k,p) <= 1
end-do

! ( 3.14 )
! ----- y aponta produção no período na prensa
forall(k in K, p in 1..P)
    y(k,p) = sum(i in I) z(i,k,p)

! ( 3.15 )
! ----- Y Deve obrigar o setup existir se houver um vazio
forall (k in K, p in 1..P-1)
do
    y(k,p) >= y(k,p+1)
end-do

! ( 3.16 )
! ----- Um ferramental não pode ser usado ao mesmo tempo que outro
forall (i,j IN I, k , k2 in K, p IN 1..P | i <> j)
do
    if k <> k2 then
        z(i,k,p) + z(j,k2,p) + R(i,j) <= 2
    end-if
end-do

! ( 3.74 )
forall (i in I, k,k1 in K, p,p1 in 1..P)
    if k <> k1 AND p <> p1 then
        z(i,k,p) + z(i,k1,p1) <= 1
    end-if

! ( 3.41 )
forall ( k in K)
    tot_geral >= sum(i in I, p in 1..P) z(i,k,p)

! ( 3.40 )
minimize( tot_geral)

```

```

writeln("SOLUÇÃO:")
writeln("total geral = ",getsol(tot_geral))
writeln("-----")

forall (k in K,p in 1..P, i in I)
  if getsol(z(i,k,p)) > 0.1 and
    getsol(z(i,k,p)) < 1.1 then
    writeln ("z(",i,',',k,',',p,') = ',getsol(z(i,k,p)))
  end-if
writeln("-----")
forall (i,j in I, k in K, p in 1..P)
  if getsol(w(i,j,k,p)) > -1 and
    getsol(w(i,j,k,p)) < 2 and
    getsol(w(i,j,k,p)) <> 0 then
    writeln ("Val do w (",i,',',j,',',k,',',p,") = ",getsol(w(i,j,k,p) ))
  end-if
writeln("-----")

end-model

```



```

! período t 0 caso contrário
si:      array(K)      of real      ! setup inicial da peça
end-declarations

!      Gupta e Magnusson
initializations from 'g_0303.txt'
      D S NN R Si
end-initializations

! -----
! --- Subject to
! (3.18)
! Para teste estaremos considerando o estoque início e fim = 0, assim
! estaremos aproximando este modelo do problema em estudo.
forall ( i in N)
      sum(k in K, t in tau) x(i,k,t) = D(i)

! (3.19)
forall (i in N,k in K,t in tau)
      x(i,k,t) - y(i,k,t) <= 0

! (3.20)
      forall (t in tau, k in K)
            sum(i in N) y(i,k,t) <= 1

! (3.21)
forall (i in N,k in K, t in tau)
      y(i,k,t) <= omega(k,t)

! (3.22)
forall (k in K,t in tau)
      sum(i in N) y(i,k,t) - 1 <= ((NN - 1) * delta(k,t))

! (3.23)
forall (k in K,t in tau)
      do
            sum(i in N) alfa(i,k,t) >= omega(k,t)
            sum(i in N) alfa(i,k,t) <= 1
      end-do

! (3.24)

```

```

forall (k in K,t in tau)
  do
    sum(i in N) beta(i,k,t) >= omega(k,t)
    sum(i in N) beta(i,k,t) <= 1
  end-do

! (3.25)
forall (i in N,k in K, t in tau)
  alfa(i,k,t) <= y(i,k,t)

! (3.26)
forall (i in N,k in K, t in tau)
  beta(i,k,t) <= y(i,k,t)

! (3.27)
forall (i in N,k in K, t in tau)
  alfa(i,k,t) + beta(i,k,t) <= 2 - delta(k,t)

! (3.28)
forall (t in tau, k in K)
  sum(i in N) gama(i,k,t) = 1

! (3.29)
forall (i,j in N, t in 2..tau2, k in K | i<>j)
  T(i,j,k,t) >=gama(i,k,t-1) + gama(j,k,t) - omega(k,t) -1

! (3.30)
forall (i,j in N,k in K, t in 2..tau2 | i<>j)
  T(j,i,k,t) >= alfa(i,k,t) + beta(j,k,t-1) -1

! (3.31)
forall (t in tau,k in K)
  do
    omega(k,t) >= 0
    omega(k,t) <= 1
  end-do

! (3.32) (3.33)
forall (i in N,k in K,t in tau)
  do
    x(i,k,t) >= 0

```

```

        x(i,k,t) <= 1
        delta(k,t) >= 0
        delta(k,t) <= 1
    end-do

! (3.34)
forall(i,j in N,k in K,t in tau)
    do
        T(j,i,k,t)    is_binary
        y(i,k,t)       is_binary
        alfa(i,k,t)    is_binary
        beta(i,k,t)    is_binary
        gama(i,k,t)    is_binary
    end-do

! (3.35)
forall (i in N,k in K, t in 2..tau2)
    T(i,i,k,t) = 0

! (3.36)
forall(i,j in N,k, k2 in K, t in tau)
    do
        if k <> k2 then
            y(i,k,t) + y(j,k2,t) + R(i,j) <= 2
        end-if
    end-do

! (3.37) (3.38)
forall (t in tau, k in K)
    do
        sum(i in N) alfa(i,k,t) <= 1
        sum(i in N) beta(i,k,t) <= 1
    end-do

! (3.39)
forall (k in K, t in 1..tau2-1)
    do
        omega(k,t) >= omega(k,t+1)
    end-do

! (3.40)

```

```

prep_ini := sum(i in N, k in K ) Si(i) * y(i,k,1)

! -----
! Soma dos tempos de setup
soma_st := sum (i,j in N,k in K, t in tau) (S(i,j) * T(i,j,k,t))
soma_est := sum(i in N,k in K, t in tau) x(i,k,t)
soma_total := soma_st + soma_est + prep_ini

! (3.17) Função objetivo

minimize (soma_total)

! =====
! ===== PRINT =====
! =====

writeln("SOLUÇÃO:")
writeln("Valor setup tt = ",getsol(soma_st))
writeln("Valor estoq tt = ",getsol(soma_est))
writeln("Valor Setup ini = ",getsol(prepare_ini))
writeln("Valor total = ",getsol(soma_total))

writeln (' ===== X =====')
forall ( i in N,k in K, t in tau) do
    if getsol(x(i,k,t)) <> 0 then
        writeln ('x(' ,i , ',' ,k , ',' ,t ,') = ',getsol(x(i,k,t)))
    end-if
end-do
writeln("Fim -----")

end-model

```

### Apêndice E: Dados utilizados nos testes computacionais do Modelo III

Para 7 peças:

	Peça						
	1	2	3	4	5	6	7
Demanda	4	5	4	5	4	5	4
Setup inicial	3	2	4	3	2	4	3

	Peça							
	1	2	3	4	5	6	7	
Peça	1	0	3	1	2	3	1	2
	2	3	0	2	2	3	3	2
	3	1	2	0	2	3	3	2
	4	2	2	2	0	3	3	2
	5	3	3	3	3	0	3	2
	6	1	3	3	3	3	0	2
	7	2	2	2	2	2	2	0

	Peça							
	1	2	3	4	5	6	7	
Peça	1	1	1	0	0	1	0	0
	2	1	1	0	0	0	0	0
	3	0	0	1	0	1	0	0
	4	0	0	0	1	0	0	0
	5	1	0	1	0	1	1	0
	6	0	0	0	0	0	1	0
	7	0	0	0	0	0	0	1

Para 8 peças:

Peça							
1	2	3	4	5	6	7	8





## Incompatibilidade

		Peça								
		1	2	3	4	5	6	7	8	9
Peça	1	1	1	0	0	1	0	0	0	0
	2	1	1	0	0	0	0	0	0	0
	3	0	0	1	0	1	0	0	0	1
	4	0	0	0	1	0	0	0	1	0
	5	1	0	1	0	1	1	0	0	0
	6	0	0	0	0	0	1	0	0	0
	7	0	0	0	0	0	0	1	0	0
	8	0	0	0	1	0	0	0	1	0
	9	0	0	1	0	0	0	0	0	1

## Para 10 peças

		Peça									
		1	2	3	4	5	6	7	8	9	10
Demanda		4	5	4	5	4	5	4	5	4	5
Setup inicial		3	2	4	3	2	4	3	2	4	3

## Setup

		Peça									
		1	2	3	4	5	6	7	8	9	10
Peça	1	0	3	1	2	3	1	2	3	1	1
	2	3	0	2	2	3	3	2	3	1	1
	3	1	2	0	2	3	3	2	3	1	1
	4	2	2	2	0	3	3	2	3	1	1
	5	3	3	3	3	0	3	2	3	1	1
	6	1	3	3	3	3	0	2	3	1	1
	7	2	2	2	2	2	2	0	3	1	1
	8	3	3	3	3	3	3	3	0	1	1
	9	1	1	1	1	1	1	1	1	0	1
	10	1	1	1	1	1	1	1	1	1	0

## Incompatibilidade

		Peça									
		1	2	3	4	5	6	7	8	9	10
Peça	1	1	1	0	0	1	0	0	0	0	0
	2	1	1	0	0	0	0	0	0	0	0
	3	0	0	1	0	1	0	0	0	1	0
	4	0	0	0	1	0	0	0	1	0	0
	5	1	0	1	0	1	1	0	0	0	0



## Apêndice F – Resultados para um exemplar real do problema

Resultado real da distribuição das peças

Prensa	Item	Qtd	Set-up	Produção	Produção	Tempos		
			Início	Início	Fim	Setup	Prod.	Final
1	1	50	29/8/11 5:30	29/8/11 6:00	29/8/11 6:39	30	39	69
1	2	1	29/8/11 6:39	29/8/11 7:02	29/8/11 7:03	23	0	92
1	3	88	29/8/11 7:03	29/8/11 8:00	29/8/11 9:02	56	62	210
1	4	83	29/8/11 9:02	29/8/11 9:02	29/8/11 10:00	0	58	268
1	5	88	29/8/11 13:07	29/8/11 14:04	29/8/11 15:06	56	62	386
1	6	83	29/8/11 15:06	29/8/11 15:44	29/8/11 16:42	37	58	481
1	7	50	29/8/11 16:42	29/8/11 17:39	29/8/11 18:14	56	35	572
1	8	80	29/8/11 18:14	29/8/11 19:10	29/8/11 20:07	56	56	684
1	9	150	29/8/11 20:07	29/8/11 21:03	29/8/11 22:49	56	105	845
1	10	150	29/8/11 22:49	29/8/11 22:49	30/8/11 0:35	0	105	950
1	11	8	30/8/11 0:35	30/8/11 1:32	30/8/11 1:37	56	5	1011
1	12	7	30/8/11 1:37	30/8/11 1:37	30/8/11 1:42	0	4	1015
1	13	150	30/8/11 1:42	30/8/11 2:39	30/8/11 4:25	56	105	1176
1	14	8	30/8/11 4:25	30/8/11 5:21	30/8/11 5:27	56	5	1237
1	15	41	30/8/11 5:27	30/8/11 6:23	30/8/11 6:59	56	36	1329
1	16	41	30/8/11 6:59	30/8/11 6:59	30/8/11 7:35	0	36	1365
1	17	41	30/8/11 7:35	30/8/11 9:00	30/8/11 9:36	84	36	1485
1	18	41	30/8/11 9:36	30/8/11 9:36	30/8/11 10:13	0	36	1521
1	19	41	30/8/11 10:13	30/8/11 12:37	30/8/11 13:13	144	36	1701
1	20	41	30/8/11 13:13	30/8/11 13:13	30/8/11 13:50	0	36	1737
1	21	150	30/8/11 13:50	30/8/11 14:46	30/8/11 16:32	56	105	1898
1	22	7	30/8/11 16:32	30/8/11 17:28	30/8/11 17:33	56	4	1958
1	23	296	30/8/11 17:33	30/8/11 18:30	30/8/11 21:24	56	174	2188
1	24	296	30/8/11 21:24	30/8/11 21:24	31/8/11 0:18	0	174	2362
1	25	100	31/8/11 0:18	31/8/11 1:15	31/8/11 2:09	56	54	2472
1	26	137	31/8/11 2:09	31/8/11 3:05	31/8/11 4:42	56	96	2624
1	27	133	31/8/11 4:42	31/8/11 4:42	31/8/11 6:16	0	93	2717
1	28	100	31/8/11 6:16	31/8/11 7:12	31/8/11 8:07	56	54	2827
1	29	284	31/8/11 8:07	31/8/11 9:03	31/8/11 14:14	56	310	3193
1	30	266	31/8/11 14:14	31/8/11 16:07	31/8/11 20:01	112	234	3539
1	31	137	31/8/11 20:01	31/8/11 20:58	31/8/11 22:35	56	96	3691
1	32	133	31/8/11 22:35	31/8/11 22:58	1/9/11 0:32	23	93	3807
1	33	41	1/9/11 0:32	1/9/11 1:28	1/9/11 1:57	56	28	3891
1	34	30	1/9/11 1:57	1/9/11 2:54	1/9/11 3:15	56	21	3968
1	35	300	1/9/11 3:15	1/9/11 3:53	1/9/11 7:24	37	211	4216
1	36	100	1/9/11 7:24	1/9/11 8:21	1/9/11 9:25	56	64	4336
1	37	100	1/9/11 9:25	1/9/11 9:53	1/9/11 10:57	28	64	4428
1	38	105	1/9/11 10:57	1/9/11 12:54	1/9/11 14:01	116	67	4611
1	39	105	1/9/11 14:01	1/9/11 14:25	1/9/11 15:32	23	67	4701
1	40	80	1/9/11 15:32	1/9/11 16:29	1/9/11 17:25	56	56	4813

1	41	80	1/9/11 17:25	1/9/11 17:25	1/9/11 18:22	0	56	4869
1	42	80	1/9/11 18:22	1/9/11 19:18	1/9/11 20:15	56	56	4981
1	43	80	1/9/11 20:15	1/9/11 20:38	1/9/11 21:35	23	56	5060
1	44	51	1/9/11 21:35	1/9/11 22:31	1/9/11 23:07	56	36	5152
1	45	51	1/9/11 23:07	1/9/11 23:07	1/9/11 23:43	0	36	5188
1	46	51	1/9/11 23:43	2/9/11 0:40	2/9/11 1:16	56	36	5280
1	47	51	2/9/11 1:16	2/9/11 1:16	2/9/11 1:52	0	36	5316
1	48	51	2/9/11 1:52	2/9/11 2:48	2/9/11 3:24	56	36	5408
1	49	51	2/9/11 3:24	2/9/11 3:24	2/9/11 4:00	0	36	5444
1	50	41	2/9/11 4:00	2/9/11 4:56	2/9/11 5:25	56	28	5528
1	51	51	2/9/11 5:25	2/9/11 6:22	2/9/11 6:58	56	36	5620
1	52	284	2/9/11 6:58	2/9/11 7:54	2/9/11 12:37	56	282	5958
1	53	266	2/9/11 12:37	2/9/11 13:52	2/9/11 17:21	75	208	6241
1	54	51	2/9/11 17:21	2/9/11 18:18	2/9/11 18:54	56	36	6333
1	55	54	2/9/11 18:54	2/9/11 18:54	2/9/11 19:32	0	38	6371
1	56	46	2/9/11 19:32	2/9/11 20:28	2/9/11 21:01	56	32	6459
1	57	47	2/9/11 21:01	2/9/11 21:01	2/9/11 21:34	0	33	6492
1	58	9	2/9/11 21:34	2/9/11 21:46	2/9/11 21:52	11	6	6509
1	59	9	2/9/11 21:52	2/9/11 22:04	2/9/11 22:10	11	6	6526
1	60	51	2/9/11 22:10	2/9/11 23:06	2/9/11 23:42	56	36	6618
1	61	102	2/9/11 23:42	3/9/11 0:39	3/9/11 1:37	56	57	6731
1	30	360	3/9/11 1:37	3/9/11 2:33	3/9/11 7:51	56	317	7104
1	63	360	3/9/11 7:51	3/9/11 8:47	3/9/11 13:01	56	254	7414
1	64	125	3/9/11 13:01	3/9/11 13:58	3/9/11 15:26	56	88	7558
1	65	125	3/9/11 15:26	3/9/11 15:26	3/9/11 16:54	0	88	7646
1	66	125	3/9/11 16:54	3/9/11 17:51	3/9/11 19:19	56	88	7790
1	67	125	3/9/11 19:19	3/9/11 19:19	3/9/11 20:47	0	88	7878
1	68	150	3/9/11 20:47	3/9/11 21:44	3/9/11 23:29	56	105	8039
1	69	150	3/9/11 23:29	3/9/11 23:48	4/9/11 1:34	18	105	8162
1	70	51	4/9/11 1:34	4/9/11 2:31	4/9/11 3:07	56	36	8254
1	71	54	4/9/11 3:07	4/9/11 3:40	4/9/11 4:18	32	38	8324
1	72	150	4/9/11 4:18	4/9/11 5:14	4/9/11 7:00	56	105	8485
1	73	150	4/9/11 7:00	4/9/11 7:38	4/9/11 9:24	37	105	8627
1	7	81	4/9/11 9:24	4/9/11 10:20	4/9/11 11:17	56	57	8740
1	9	150	4/9/11 11:17	4/9/11 13:14	4/9/11 15:00	116	105	8961
1	10	150	4/9/11 15:00	4/9/11 15:00	4/9/11 16:45	0	105	9066
1	77	24	4/9/11 16:45	4/9/11 17:42	4/9/11 17:59	56	16	9138
1	78	36	4/9/11 17:59	4/9/11 18:55	4/9/11 20:21	56	85	9279
1	79	199	4/9/11 20:21	4/9/11 20:56	4/9/11 23:17	35	140	9454

Prensa	Item	Qtd	Set-up	Produção	Produção	Tempos		
			Início	Início	Fim	Setup	Prod.	Final
2	81	23	29/8/11 5:30	29/8/11 7:00	29/8/11 7:11	90	11	101
2	82	23	29/8/11 7:11	29/8/11 7:11	29/8/11 7:22	0	11	112
2	83	296	29/8/11 7:22	29/8/11 9:00	29/8/11 12:53	97	233	442

2	84	296	29/8/11 12:53	29/8/11 13:32	29/8/11 16:25	39	173	654
2	85	238	29/8/11 16:25	29/8/11 18:03	29/8/11 20:57	97	174	925
2	86	497	29/8/11 20:57	29/8/11 22:34	30/8/11 4:38	97	363	1385
2	87	279	30/8/11 4:38	30/8/11 6:16	30/8/11 9:40	97	204	1686
2	88	279	30/8/11 9:40	30/8/11 9:40	30/8/11 14:04	0	264	1950
2	89	279	30/8/11 14:04	30/8/11 16:01	30/8/11 19:07	117	185	2252
2	90	279	30/8/11 19:07	30/8/11 20:05	30/8/11 23:11	58	185	2495
2	91	200	30/8/11 23:11	31/8/11 0:48	31/8/11 2:33	97	104	2696
2	92	200	31/8/11 2:33	31/8/11 2:33	31/8/11 4:17	0	104	2800
2	93	133	31/8/11 4:17	31/8/11 4:27	31/8/11 5:37	9	69	2878
2	94	139	31/8/11 5:37	31/8/11 5:46	31/8/11 6:59	9	72	2959
2	95	133	31/8/11 6:59	31/8/11 8:36	31/8/11 9:37	97	60	3116
2	96	200	31/8/11 9:37	31/8/11 9:56	31/8/11 11:27	18	91	3225
2	97	139	31/8/11 11:27	31/8/11 12:45	31/8/11 13:49	78	63	3366
2	98	200	31/8/11 13:49	31/8/11 14:07	31/8/11 15:39	18	91	3475
2	99	105	31/8/11 15:39	31/8/11 17:16	31/8/11 18:33	97	76	3648
2	100	105	31/8/11 18:33	31/8/11 18:33	31/8/11 19:50	0	76	3724
2	101	105	31/8/11 19:50	31/8/11 21:03	31/8/11 22:20	73	76	3873
2	102	105	31/8/11 22:20	31/8/11 22:20	31/8/11 23:37	0	76	3949
2	103	24	31/8/11 23:37	1/9/11 1:14	1/9/11 1:36	97	21	4067
2	104	24	1/9/11 1:36	1/9/11 3:04	1/9/11 3:26	87	21	4175
2	105	216	1/9/11 3:26	1/9/11 5:04	1/9/11 7:15	97	131	4403
2	106	216	1/9/11 7:15	1/9/11 8:53	1/9/11 12:31	97	218	4718
2	107	488	1/9/11 12:31	1/9/11 14:08	1/9/11 16:31	97	142	4957
2	108	488	1/9/11 16:31	1/9/11 16:31	1/9/11 18:54	0	142	5099
2	109	524	1/9/11 18:54	1/9/11 20:32	2/9/11 1:38	97	306	5502
2	110	524	2/9/11 1:38	2/9/11 1:38	2/9/11 6:45	0	306	5808
2	83	366	2/9/11 6:45	2/9/11 8:23	2/9/11 12:57	97	274	6179
2	84	366	2/9/11 12:57	2/9/11 13:36	2/9/11 17:10	39	214	6432
2	113	250	2/9/11 17:10	2/9/11 18:48	2/9/11 20:19	97	91	6620
2	114	130	2/9/11 20:19	2/9/11 21:57	2/9/11 22:56	97	59	6776
2	115	488	2/9/11 22:56	3/9/11 0:34	3/9/11 2:46	97	132	7005
2	116	488	3/9/11 2:46	3/9/11 4:24	3/9/11 6:12	97	108	7210
2	117	116	3/9/11 6:12	3/9/11 7:49	3/9/11 8:55	97	65	7372
2	118	116	3/9/11 8:55	3/9/11 8:55	3/9/11 10:00	0	65	7437
2	119	116	3/9/11 10:00	3/9/11 11:38	3/9/11 12:43	97	65	7599
2	120	250	3/9/11 12:43	3/9/11 14:20	3/9/11 15:21	97	60	7756
2	121	90	3/9/11 15:21	3/9/11 16:59	3/9/11 18:05	97	65	7918
2	122	90	3/9/11 18:05	3/9/11 19:03	3/9/11 20:09	58	65	8041
2	123	308	3/9/11 20:09	3/9/11 21:47	4/9/11 1:32	97	225	8363
2	124	314	4/9/11 1:32	4/9/11 1:32	4/9/11 7:22	0	349	8712
2	125	206	4/9/11 7:22	4/9/11 7:36	4/9/11 11:07	14	210	8936
2	126	206	4/9/11 11:07	4/9/11 12:07	4/9/11 14:38	60	150	9146
2	127	116	4/9/11 14:38	4/9/11 16:15	4/9/11 18:21	97	125	9368
2	128	350	4/9/11 18:21	4/9/11 20:58	5/9/11 0:15	157	197	9722
2	129	350	5/9/11 0:15	5/9/11 0:15	5/9/11 7:02	0	407	10129

2	130	168	5/9/11 7:02	5/9/11 8:40	5/9/11 10:22	97	102	10328
2	131	169	5/9/11 10:22	5/9/11 10:52	5/9/11 13:35	29	163	10520
2	85	300	5/9/11 13:35	5/9/11 15:12	5/9/11 18:52	97	219	10836
2	86	300	5/9/11 18:52	5/9/11 20:29	6/9/11 0:09	97	219	11152
2	134	90	6/9/11 0:09	6/9/11 1:46	6/9/11 2:30	97	43	11292
2	135	90	6/9/11 2:30	6/9/11 3:29	6/9/11 4:13	58	43	11393

Prensa	Item	Qtd	Set-up	Produção	Produção	Tempos		
			Início	Início	Fim	Setup	Prod.	Final
3	137	215	29/8/11 5:30	29/8/11 5:30	29/8/11 7:08	0	98	98
3	138	334	29/8/11 7:08	29/8/11 8:45	29/8/11 13:09	97	263	458
3	139	334	29/8/11 13:09	29/8/11 14:47	29/8/11 18:10	97	203	758
3	140	212	29/8/11 18:10	29/8/11 19:48	29/8/11 21:57	97	129	984
3	141	159	29/8/11 21:57	29/8/11 22:07	29/8/11 23:30	9	83	1076
3	142	67	29/8/11 23:30	30/8/11 1:08	30/8/11 1:47	97	39	1212
3	143	67	30/8/11 1:47	30/8/11 1:47	30/8/11 2:26	0	39	1251
3	144	67	30/8/11 2:26	30/8/11 4:04	30/8/11 4:43	97	39	1387
3	145	67	30/8/11 4:43	30/8/11 5:22	30/8/11 6:01	39	39	1465
3	146	397	30/8/11 6:01	30/8/11 7:39	30/8/11 10:40	97	181	1743
3	147	397	30/8/11 10:40	30/8/11 12:39	30/8/11 15:40	118	181	2042
3	148	193	30/8/11 15:40	30/8/11 17:18	30/8/11 18:52	97	94	2233
3	149	23	30/8/11 18:52	30/8/11 19:10	30/8/11 19:21	18	10	2261
3	150	23	30/8/11 19:21	30/8/11 19:39	30/8/11 19:50	18	10	2289
3	151	193	30/8/11 19:50	30/8/11 20:08	30/8/11 21:42	18	94	2401
3	152	1192	30/8/11 21:42	30/8/11 23:20	31/8/11 9:43	97	623	3121
3	153	1192	31/8/11 9:43	31/8/11 11:20	31/8/11 19:36	97	496	3714
3	154	1192	31/8/11 19:36	31/8/11 21:14	1/9/11 4:30	97	436	4247
3	155	1192	1/9/11 4:30	1/9/11 6:07	1/9/11 15:12	97	544	4888
3	156	1192	1/9/11 15:12	1/9/11 16:50	2/9/11 0:54	97	484	5469
3	157	1192	2/9/11 0:54	2/9/11 2:32	2/9/11 9:08	97	396	5962
3	91	422	2/9/11 9:08	2/9/11 10:46	2/9/11 15:26	97	280	6339
3	92	418	2/9/11 15:26	2/9/11 15:26	2/9/11 19:05	0	218	6557
3	160	65	2/9/11 19:05	2/9/11 20:42	2/9/11 21:20	97	38	6692
3	161	65	2/9/11 21:20	2/9/11 21:35	2/9/11 22:13	14	38	6744
3	162	214	2/9/11 22:13	2/9/11 23:51	3/9/11 2:01	97	130	6971
3	163	214	3/9/11 2:01	3/9/11 3:39	3/9/11 6:15	97	156	7224
3	164	81	3/9/11 6:15	3/9/11 7:53	3/9/11 8:40	97	47	7368
3	165	88	3/9/11 8:40	3/9/11 9:09	3/9/11 10:01	29	51	7448
3	166	81	3/9/11 10:01	3/9/11 11:38	3/9/11 12:26	97	47	7592
3	167	88	3/9/11 12:26	3/9/11 13:05	3/9/11 13:56	39	51	7682
3	168	65	3/9/11 13:56	3/9/11 15:34	3/9/11 16:12	97	38	7817
3	169	65	3/9/11 16:12	3/9/11 16:51	3/9/11 17:29	39	38	7894
3	170	94	3/9/11 17:29	3/9/11 19:07	3/9/11 20:02	97	55	8046
3	171	93	3/9/11 20:02	3/9/11 20:41	3/9/11 21:35	39	54	8139
3	172	89	3/9/11 21:35	3/9/11 23:13	4/9/11 0:07	97	54	8290

3	173	89	4/9/11 0:07	4/9/11 0:07	4/9/11 1:01	0	54	8344
3	174	89	4/9/11 1:01	4/9/11 2:39	4/9/11 3:22	97	43	8484
3	175	89	4/9/11 3:22	4/9/11 3:41	4/9/11 4:24	18	43	8545
3	176	282	4/9/11 4:24	4/9/11 6:01	4/9/11 8:53	97	171	8813
3	177	282	4/9/11 8:53	4/9/11 10:21	4/9/11 14:13	87	231	9131
3	178	150	4/9/11 14:13	4/9/11 15:51	4/9/11 17:41	97	109	9337
3	179	150	4/9/11 17:41	4/9/11 18:00	4/9/11 20:50	19	169	9525
3	180	156	4/9/11 20:50	4/9/11 22:27	4/9/11 23:43	97	76	9698
3	181	156	4/9/11 23:43	5/9/11 1:41	5/9/11 6:27	117	286	10101
3	182	156	5/9/11 6:27	5/9/11 8:04	5/9/11 9:16	97	71	10269
3	183	156	5/9/11 9:16	5/9/11 9:34	5/9/11 10:45	18	71	10358
3	184	102	5/9/11 10:45	5/9/11 13:23	5/9/11 14:22	157	59	10574
3	185	114	5/9/11 14:22	5/9/11 16:00	5/9/11 17:23	97	83	10754
3	186	114	5/9/11 17:23	5/9/11 18:51	5/9/11 20:15	87	83	10924
3	187	100	5/9/11 20:15	5/9/11 21:52	5/9/11 22:41	97	48	11069
3	188	100	5/9/11 22:41	5/9/11 22:41	5/9/11 23:30	0	48	11117
3	189	100	5/9/11 23:30	6/9/11 1:07	6/9/11 2:00	97	52	11266
3	190	100	6/9/11 2:00	6/9/11 2:24	6/9/11 3:16	24	52	11342
3	105	257	6/9/11 3:16	6/9/11 4:54	6/9/11 7:30	97	156	11595
3	106	257	6/9/11 7:30	6/9/11 9:08	6/9/11 13:16	97	248	11940
3	193	84	6/9/11 13:16	6/9/11 14:54	6/9/11 15:55	97	61	12098
3	120	130	6/9/11 15:55	6/9/11 16:34	6/9/11 18:09	39	95	12232
3	194	130	6/9/11 18:09	6/9/11 18:39	6/9/11 20:14	29	95	12356
3	112	84	6/9/11 20:14	6/9/11 20:43	6/9/11 21:44	29	61	12446

**Makespan = 12446**

### Distribuição encontrada pelo algoritmo heurístico

Exemplar: 03194.txt

m = 3 n = 194

Solução final:

Exemplar	Melhor Solução	Sol. Media	Tempo Medio (s)
-----			
03194.txt	10585	10710.40	1800.09

**Custo da solução = 10585 (Menor MAKESPAN)**

**Tempo = 1800 segundos**