

**CENTRO UNIVERSITÁRIO DA FEI**

**DÉCIO NORIO IMAMURA**

**UTILIZAÇÃO DE REGRAS DE DESPACHO PARA PROGRAMAÇÃO DE  
PRODUÇÃO EM AMBIENTE *JOB SHOP***

São Bernardo do Campo  
2013

DÉCIO NORIO IMAMURA

**UTILIZAÇÃO DE REGRAS DE DESPACHO PARA PROGRAMAÇÃO DE  
PRODUÇÃO EM AMBIENTE *JOB SHOP***

Dissertação de Mestrado apresentada ao Centro  
Universitário da FEI para obtenção do título de  
Mestre em Engenharia Mecânica, orientado pelo  
Prof. Dr. João Chang Jr.

São Bernardo do Campo  
2013

Imamura, Décio Norio

Utilização de regras de despacho para programação de produção em ambiente jobshop/ Décio Norio Imamura. São Bernardo do Campo, 2013.

79 f. : il.

Dissertação - Centro Universitário da FEI.

Orientador: Prof. Dr. João Chang Jr.

1. Scheduling. 2. Jobshop. 3. Sequenciamento. I. Chang Jr., João, orient. II. Título.

CDU 658.5



Centro Universitário da **FEI**

## APRESENTAÇÃO DE DISSERTAÇÃO

### ATA DA BANCA JULGADORA

**Aluno:** Décio Norio Imamura

**Matrícula:** 210113-7

**Título do Trabalho:** Utilização de regras de despacho para programação de produção em ambiente JOBSHOP.

**Área de Concentração:** Produção

**ORIGINAL ASSINADA**

**Orientador:** Prof. Dr. João Chang Junior

**Data da realização da defesa:** 15/05/2013

A Banca Julgadora abaixo-assinada atribuiu ao aluno o seguinte:

APROVADO

REPROVADO

São Bernardo do Campo, 15 de Maio de 2013.

#### MEMBROS DA BANCA JULGADORA

Prof. Dr. João Chang Junior

Ass.: \_\_\_\_\_

Prof. Dr. Alexandre Augusto Massote

Ass.: \_\_\_\_\_

Prof. Dr. Marcius Fabius Henriques de Carvalho

Ass.: \_\_\_\_\_

#### VERSÃO FINAL DA DISSERTAÇÃO

ENDOSSO DO ORIENTADOR APÓS A INCLUSÃO DAS  
RECOMENDAÇÕES DA BANCA EXAMINADORA

Aprovação do Coordenador do Programa de Pós-graduação

## RESUMO

A forte competição entre as organizações industriais faz com que se busque olhar em detalhe as atividades que impactam seus custos operacionais. Nesse contexto, observa-se cada vez mais a importância do planejamento detalhado da produção nos resultados do chão de fábrica, vitais para o bom resultado da empresa. Em função disso, esta atividade, conhecida como *scheduling*, responsável pelo sequenciamento final do plano de produção tem recebido grande atenção de pesquisadores nos últimos anos.

Dentre os ambientes industriais aquele conhecido como *jobshop* é um dos mais importantes e complexos. Uma das áreas de maior desenvolvimento no suporte aos métodos de *scheduling* do *jobshop* é a de softwares específicos para essa finalidade, disponíveis, porém de certa complexidade e dificuldade de implantação por organizações de menor porte e recursos. Este trabalho procurou conhecer um desses softwares, o IBM ILOG CPLEX®, ao mesmo tempo em que buscou alternativas mais acessíveis para essas últimas organizações e comparando-as, para avaliar sua viabilidade. Neste caso estudou-se a aplicação de regras simplificadoras conhecidas como regras de despacho.

Numa simulação prática observou-se que a solução mais simples continua a ser uma opção válida e acessível para organizações que não puderam ainda ter acesso a recursos mais sofisticados, para problemas de menor complexidade.

Palavras-chave: - *Jobshop – Scheduling* - Sequenciamento

## **ABSTRACT**

The hard competition among industrial organizations drives them to look forward, more and more, the details that have impact in their operational costs. In this context the detailed production planning importance in the shop floor results is being more and more recognized and this is a key to the successful final result of any organization. As a consequence, this activity, known as scheduling, responsible for the final production plan sequencing has received great attention from researchers in the last years.

Among the industrial environments the one known as jobshop is one of the most important and complexes. One of the areas of greatest development in the support to the jobshop scheduling is the area of software and there are some available in the market, specific to this purpose, but with a certain level of complexity and limited access for smaller organizations with limited resources. This work aimed at learning about one of these softwares, the IBM ILOG CPLEX ®, at the same time as aiming at the evaluation of easier alternatives, more attainable for these last organizations and comparing them to understand their feasibility. In this case it was studied the application of simplification rules, known as dispatch rules.

In a practical simulation it was observed that the easier solution can be a valid alternative, attainable for smaller organizations with limited access to more sophisticated resources, for problems of lower level of complexity.

**Key words:** - Jobshop – Scheduling - Sequencing

## LISTA DE FIGURAS

Figura 1 - Classificação de Processos Produtivos .....	11
Figura 2 - Fluxo de Informação no Processo de Planejamento de um Sistema de Manufatura	13
Figura 3 - Grafo disjuntivo para um Jobshop - Fonte: Bartak <i>et al.</i> , (2010:266).....	22
Figura 4 - Grafo Disjuntivo para 2 <i>Jobs</i> e 2 Máquinas .....	23
Figura 5 - Quebra do Primeiro Nó da Árvore B&B .....	16
Figura 6 - Dividindo $S_1$ em $S_{11}$ e $S_{12}$ .....	27
Figura 7 - Análise do Nó $S_{12}$ - Interrupção Por Otimalidade .....	25
Figura 8 - Algoritmo Genérico para o <i>CSP</i> .....	33
Figura 9 - Redução de domínio por meio da verificação lógica da consistência do arco .....	35
Figura 10 - Ilustração da Propagação Lógica da Restrição .....	36

## LISTA DE GRÁFICOS

Gráfico 1 - Gráfico de Gantt por máquina da regra <i>LPT</i> – parte 1 .....	46
Gráfico 2 - Gráfico de Gantt por Máquina da regra <i>LPT</i> – parte 2 .....	47
Gráfico 3 - Gráfico de Gantt por Máquina da regra <i>LPT</i> – parte 3 .....	47
Gráfico 4 - Gráfico de Gantt por <i>Job</i> da regra <i>LPT</i> – parte 1 .....	49
Gráfico 5 - Gráfico de Gantt por <i>Job</i> da regra <i>LPT</i> – parte 2.....	50
Gráfico 6 - Gráfico de Gantt por <i>Job</i> da regra <i>LPT</i> – parte 3.....	51
Gráfico 7 - Gráfico de Gantt por Máquina da Regra <i>MWKR</i> – parte 1 .....	54
Gráfico 8 - Gráfico de Gantt por Máquina da Regra <i>MWKR</i> – parte 2 .....	55
Gráfico 9 - Gráfico de Gantt por Máquina da Regra <i>MWKR</i> – parte 3 .....	56
Gráfico 10 - Gráfico de Gantt por <i>Job</i> da Regra <i>MWKR</i> – parte 1.....	57
Gráfico 11 - Gráfico de Gantt por <i>Job</i> da Regra <i>MWKR</i> – parte 2.....	58
Gráfico 12 - Gráfico de Gantt por <i>Job</i> da Regra <i>MWKR</i> – parte 3.....	59
Gráfico 13 - Gráfico de Gantt Solução por Máquina - Programação de Restrições– parte 1 ..	64
Gráfico 14 - Gráfico de Gantt Solução por Máquina - Programação de Restrições– parte 2 ..	65
Gráfico 15 - Gráfico de Gantt Solução por Máquina - Programação de Restrições– parte 3 ..	66
Gráfico 16 - Gráfico de Gantt Solução por <i>Job</i> - Programação de Restrições– parte 1 .....	67
Gráfico 17 - Gráfico de Gantt Solução por <i>Job</i> - Programação de Restrições– parte 2.....	68
Gráfico 18 - Gráfico de Gantt Solução por <i>Job</i> - Programação de Restrições– parte 3.....	69

## LISTA DE TABELAS

Tabela 1 - Dados Iniciais do problema com 20 <i>jobs</i> , 5 máquinas e 5 operações .....	41
Tabela 2 - Dados Sintetizados do problema com 20 <i>jobs</i> , 5 máquinas e 5 operações .....	42
Tabela 3 - Solução <i>LPT</i> no formato original – parte 1 .....	43
Tabela 4 - Solução <i>LPT</i> no formato original – parte 2 .....	44
Tabela 5 - Solução <i>LPT</i> – sequenciamento de cada máquina – <i>job</i> – tempo início – tempo operação.....	45
Tabela 6 - Solução <i>LPT</i> – sequenciamento de cada máquina – <i>job</i> – tempo início – tempo operação.....	53
Tabela 7 - Solução Programação de Restrições – formato original .....	60
Tabela 8 - Solução Programação de Restrições – <i>MaqJob</i> e tempos .....	63
Tabela 9 - Comparação dos Resultados.....	71

## LISTA DE ABREVIATURAS

*MTO – Make to Order*

*MTS – Make to Stock*

*ATO – Assemble to Order*

*ETO – Engineer to Order*

*CP – Constraint Programming (Programação de Restrições)*

*ERP – Enterprise Resource Planning*

*FIESP – Federação das Indústrias do Estado de São Paulo*

*LPT – Longest Processing Time – Maior Tempo de Processamento*

*MWKR – Most Working Remaining – Maior Tempo Remanescente de Processamento*

*CSP – Constraint Solution Problem – Problema de Solução de Restrições*

*EP – Expert System – Software Especializado de Planejamento*

*APS – Advanced Planning System – Software Avançado de Planejamento*

*B&B – Branch and Bound*

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>10</b>
<b>1.1. Motivação.....</b>	<b>11</b>
<b>1.2. Objetivos.....</b>	<b>14</b>
<b>1.3. Metodologia.....</b>	<b>15</b>
<b>1.4. Estruturação do Trabalho.....</b>	<b>15</b>
<b>2. REVISÃO BIBLIOGRÁFICA.....</b>	<b>17</b>
<b>2.1. Scheduling.....</b>	<b>17</b>
<b>2.2. Jobshop.....</b>	<b>21</b>
<b>3. SCHEDULING JOBSHOP NA PRÁTICA.....</b>	<b>36</b>
<b>3.1. Contexto do Trabalho.....</b>	<b>36</b>
<b>3.2. Modelos Estudados.....</b>	<b>37</b>
<b>3.3. Simulação.....</b>	<b>40</b>
<b>4. ANÁLISE.....</b>	<b>69</b>
<b>4.1. Observações Funcionais das Soluções Testadas.....</b>	<b>69</b>
<b>4.2. Comparação dos Resultados Encontrados.....</b>	<b>69</b>
<b>4.3. Fatores Para Utilização e Limitações das Soluções Testadas.....</b>	<b>71</b>
<b>5. CONCLUSÕES.....</b>	<b>73</b>
<b>6. CONSIDERAÇÕES FINAIS.....</b>	<b>74</b>
<b>REFERÊNCIAS.....</b>	<b>75</b>
<b>APÊNDICE.....</b>	<b>80</b>

## 1. INTRODUÇÃO

No ambiente concorrencial em que estão inseridas as organizações, a gestão de operações exerce um papel fundamental e, dentro dela, o planejamento e o controle dessas operações têm sido os fatores de diferencial competitivo, fundamentados na melhor utilização dos recursos e na entrega de valor ao consumidor final.

A gestão de operações ocupa-se da atividade de gerenciamento estratégico de recursos escassos, de sua interação e dos processos que produzem e entregam bens e serviços, visando atender as necessidades dos clientes, ao mesmo tempo em que compatibiliza esse objetivo com a necessidade de eficiência na utilização desses recursos (CORREA, H. L.; CORREA, C.A., 2006).

Dentro da gestão de operações as funções planejamento, programação e controle da produção exercem um papel fundamental na gestão da capacidade produtiva, ou da prestação de serviços, na gestão dos estoques, na utilização dos recursos da organização e na entrega do produto ou serviço ao cliente.

Planejar é entender como a consideração conjunta da situação presente e da visão de futuro influencia as decisões tomadas no presente para que atinjam determinados objetivos no futuro.

Planejar é projetar o futuro diferentemente do passado, por causas sob nosso controle (CORREA et al, apud CORREA et al, 2001).

No horizonte de tempo do planejamento, a programação detalhada da produção, conhecida como *scheduling* detalha o sequenciamento e as atividades no curto prazo, definindo as prioridades nas quais as ordens de fabricação devem ocorrer e alocando o tempo e os recursos para isso. Tem, portanto, influência decisiva nos resultados operacionais.

Dentre os ambientes de planejamento de uma produção, um dos mais complexos, é o que se conhece como *Jobshop*. Trata-se de ambiente de produção composto por  $n$  jobs e  $m$  máquinas em que cada *job* é processado nas  $m$  máquinas de acordo com roteiro pré-estabelecido de operações. A programação do sequenciamento da produção (*scheduling*) neste ambiente, mesmo para atividades simples, é complexa e tem desafiado os pesquisadores. Por ser um ambiente que encontrado nos mais diversos tipos de atividade, tem sido amplamente estudado, pelo menos nos últimos 50 anos.

## 1.1 Motivação

### 1.1.1 Contexto – Ambientes Produtivos

Tratando-se de ambientes produtivos, existem na literatura diversos critérios de classificação dos sistemas de produção. Um dos critérios leva em conta o volume e a variedade produzida, conforme figura 1:

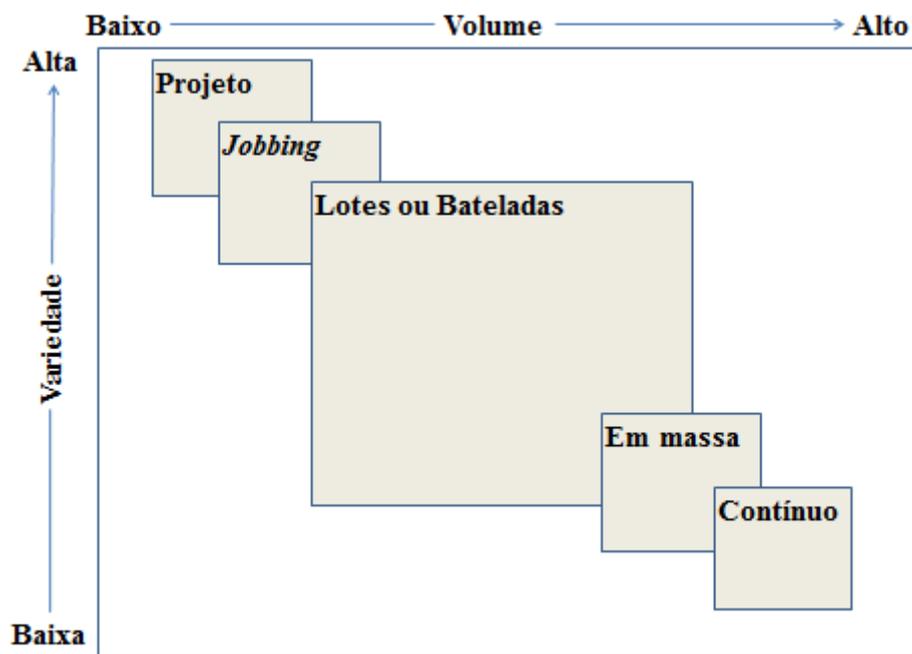


Figura 1 - Classificação de Processos Produtivos

Fonte: Slack *et al.*, 2002, p. 129

Nesta classificação o *jobshop* se insere no ambiente de *jobbing*, onde se encontra uma variedade relativamente alta associada a volumes relativamente baixos, não chegando a ser um ambiente de projeto.

Mais associada à gestão dos estoques, a classificação proposta por Lustosa (2008) apresenta as seguintes estratégias de produção:

- a) produzir para Pedido (*Make to Order – MTO*) – a produção só é iniciada após o recebimento formal do pedido do cliente. Normalmente os prazos de atendimento são maiores e existe uma necessidade de concentração de estoques no início da cadeia produtiva;

- b) produzir para Estoque (*Make to Stock – MTS*) – produtos com maior nível de padronização, possibilitando rápido atendimento. A produção é disparada sem necessidade do pedido formal, para manter um determinado nível de estoque;
- c) montar para Pedido (*Assemble to Order – ASO*) – Subconjuntos são mantidos em estoque e montados conforme a necessidade do cliente;
- d) projetar para Pedido (*Engineer to Order -ETO*) – Aplica-se a projetos únicos, totalmente customizados às necessidades dos clientes.

O *jobshop* pode ocorrer em qualquer das estratégias acima com diferentes impactos na utilização dos recursos. Particularmente nos sistemas *MTO*, *ASO* e *ETO*, devido à maior customização do produto fabricado, a utilização de boas técnicas de programação podem trazer maior impacto no resultado final (PEREIRA, 2009).

### 1.1.2 Contexto – Horizonte de Planejamento

Correa *et al.* (2006) apresentam as atividades de planejamento dentro de uma ordenação hierárquica que tem a escala de tempo como fator principal. Nesse contexto propõe-se decompor os problemas a partir do horizonte de tempo mais longo em subproblemas menores e mais detalhados quanto mais se aproxima do evento. O *scheduling* trata dos problemas de curto prazo, no coração do que se denomina “sistema de execução da manufatura”, que agrega os recursos e sistemas que dão suporte às decisões de curto prazo. Para Pinedo (2008) o processo de planejamento segue um fluxo de informação conforme a figura 2, estando a atividade de *scheduling* identificada pelo retângulo pontilhado. Ela vem depois de um planejamento macro dos pedidos num primeiro momento e do planejamento de materiais em seguida.

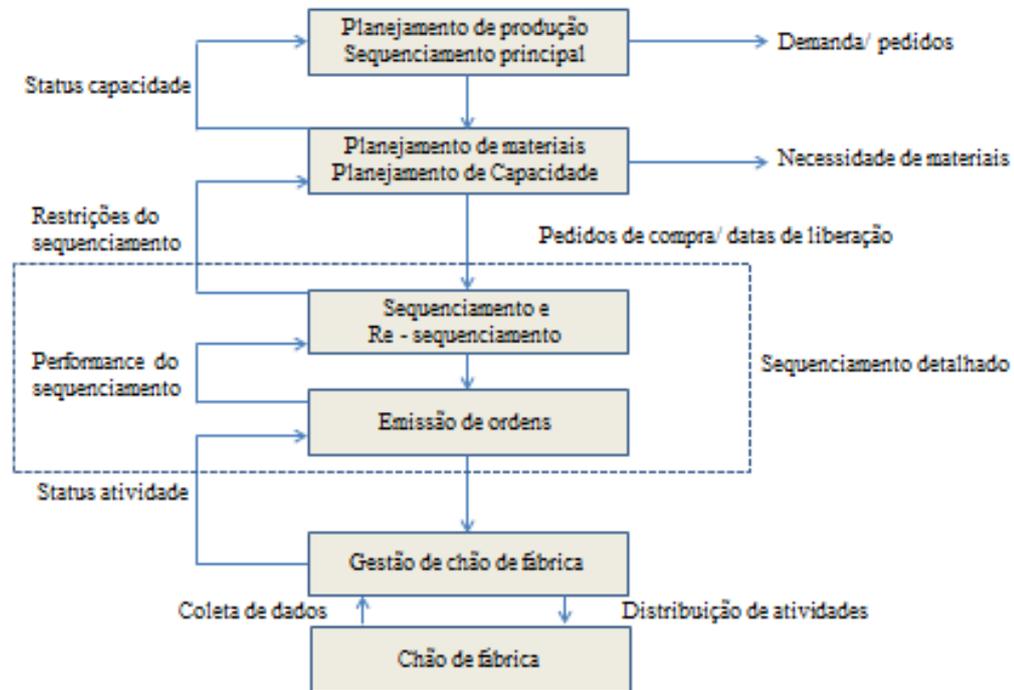


Figura 2 - Fluxo de Informação no Processo de Planejamento de um Sistema de Manufatura

Fonte: Pinedo (2008)

### 1.1.3. Dificuldades para Evolução do *Scheduling*

Apesar do grande volume de pesquisas sobre o assunto, na prática, pelo menos no Brasil, ainda é incipiente a utilização, no dia-a-dia, de métodos e técnicas mais elaboradas de *scheduling*, ou seja, a utilização de técnicas de programação finita, dentro do planejamento e controle das operações. Em pesquisa realizada com empresas filiadas à FIESP (GIACON, 2010) estudou-se a necessidade e as dificuldades de se implantar os APS's (*Advanced Planning System*), onde se insere a programação em ambiente *Jobshop*. A pesquisa concluiu que os fatores financeiros e de capacitação são os principais obstáculos ao avanço na implantação desse recurso e que a planilha eletrônica ainda é o principal meio utilizado para a programação detalhada da produção.

Em paralelo a esse cenário, efetivamente muitas pesquisas têm se realizado no sentido de desenvolver métodos para solucionar situações específicas em ambientes de programação da produção. Tais pesquisas têm se pautado por identificar um determinado ambiente, desenvolver uma heurística nova ou aprimorada e avaliá-la por meio de simulação computacional, ou seja, a pesquisa por meio do experimento. No apêndice 1 estão

relacionadas diversas dissertações de mestrado e tese de doutorado recentes que ilustram essa linha de pesquisa.

Há um contraste, portanto, entre o desenvolvimento de pesquisas com certo grau de sofisticação para solucionar problemas de programação finita e a sua utilização prática mais ampla no dia-a-dia das empresas, pelo menos no Brasil.

O presente trabalho foca um ambiente *jobshop* de uma indústria de pequeno / médio porte, que conforme pesquisa desenvolvida por de Giacon (2010), na palavra de seus gestores da atividade de planejamento, há necessidade de sistemas complementares aos seus *ERP's* para a programação detalhada da produção. Paralelamente, avaliaram-se os fatores restritivos mencionados por Giacon (2010) a uma utilização mais ampla de sistemas disponíveis para esta atividade.

## 1.2 Objetivos

O principal objetivo do trabalho é avaliar, com base em pesquisa quantitativa experimental apoiada em simulação, a viabilidade de utilização de métodos baseados em regras de despacho simples, desenvolvidos em planilhas eletrônicas, acessíveis a organizações industriais de pequeno ou médio porte com ambientes do tipo *jobshop*. As principais perguntas a serem respondidas são:

a) Uma metodologia simples, desenvolvida em planilha eletrônica, aplicada ao planejamento em um ambiente *jobshop* representa uma perda significativa no resultado operacional que a inviabilize, ou seja, o resultado esperado da atividade de programação da produção realizada com este método apresenta um resultado significativamente pior do que um método que permita chegar a um resultado ótimo?

b) Que fatores podem justificar a utilização de uma metodologia simples, baseada em planilha eletrônica quando estão disponíveis métodos e recursos que permitam que se chegue a resultados ótimos?

Neste trabalho avaliou-se, como referência, um software desenvolvido especificamente para o *scheduling*, utilizando a metodologia *Constraint Programming* (CP – Programação de Restrições), que se apresenta indicado para problemas de otimização combinatória em geral. Em contrapartida avaliou-se um método mais simples, baseado em heurística conhecida como regra de despacho, desenvolvida sobre planilha eletrônica.

### 1.3 Metodologia

Segundo Silva et al. (2000), a pesquisa científica deve, de forma simplificada “...procurar respostas para indagações propostas”.

Então, as principais questões a serem respondidas neste trabalho são:

- a) Como é um ambiente *jobshop*?
- b) Quais são os métodos conhecidos de programação do sequenciamento da produção num ambiente como este?
- c) Como funciona um determinado método escolhido na prática?

Em relação às fases da pesquisa, foram seguidas aquelas propostas por Silva et al (2000):

- a) fase decisória: o tema sequenciamento em um ambiente *jobshop* foi o escolhido, pela relevância e complexidade, limitado ao detalhamento e aplicação de um método específico;
- b) fase construtiva: plano de pesquisa envolvendo a pesquisa bibliográfica, análise e seleção do método adequado ao ambiente de estudo;
- c) fase redacional: compilação e ordenação histórica da pesquisa bibliográfica realizada em relação aos métodos de solução desenvolvidos ao longo do tempo. Análise dos resultados das observações práticas realizadas.

Ainda conforme Silva et al (2000), a pesquisa, conforme a natureza do problema abordado pode ser classificada como pesquisa aplicada, cujo objetivo é gerar conhecimentos para aplicação prática visando a solução de problemas específicos de interesse local.

Quanto à abordagem do problema o trabalho se situa no campo da pesquisa quantitativa, buscando-se traduzir em números as observações realizadas.

Quanto aos procedimentos técnicos, partiu-se de ampla pesquisa bibliográfica para uma pesquisa experimental, com a aplicação prática de um método selecionado.

Quanto ao método científico utilizado, conforme Lakatos et al. (1983), seguiu-se o método indutivo, fundamentado na experiência.

### 1.4 Estruturação do Trabalho

Este trabalho está dividido em quatro partes.

Na Introdução foi feita inicialmente uma contextualização dos problemas de *jobshop*, posicionando-os, como problemas de *scheduling*, na hierarquia dos problemas de planejamento de produção. Em seguida foram apresentadas a justificativa e a importância do

trabalho, relacionadas ao grande impacto que pode ter nos resultados das organizações e a pouca utilização prática das soluções disponíveis.

O capítulo 2 é dedicado ao levantamento histórico dos modelos de solução para os problemas do tipo *jobshop*, identificando seus fundamentos básicos.

No capítulo 3 são escolhidos e desenvolvidos três dos métodos pesquisados, de forma a aplica-los num ambiente fabril de uma média empresa. Um problema de teste é aplicado, de forma a se gerar resultados a serem analisados.

No capítulo 4 é feita uma análise dos resultados encontrados, juntamente com as conclusões gerais do trabalho.

No capítulo 5 são apresentadas as conclusões e no capítulo 6 são feitas as considerações finais do trabalho.

## 2. REVISÃO BIBLIOGRÁFICA

### 2.1. *Scheduling*

#### 2.1.1 Evolução da Metodologia

Conforme apresentado em 1.1.2 o *scheduling* é a atividade de planejamento da produção mais próxima do chão da fábrica, focando as próximas sequências de ordens de fabricação, ou seja o curtíssimo prazo, sempre buscando a alocação de recursos escassos para determinadas atividades, com o objetivo de otimizar uma ou mais medidas de *performance* (LEUNG, 2004).

A crescente preocupação com esta atividade se justifica, pelo impacto que pode ter nos resultados operacionais e, portanto, na competitividade das organizações, uma vez que bons algoritmos de sequenciamento podem reduzir significativamente os custos de produção numa organização industrial.

Coube a Gantt (1916) o pioneirismo na utilização de métodos gráficos para visualizar a programação e o status das atividades num posto de trabalho. Mais adiante, falando em organização das atividades, ele apresenta um primeiro princípio para medir as atividades no gráfico pela quantidade de tempo necessário para completá-las e um segundo princípio associando o espaço no gráfico à quantidade de atividade que deveria ter sido completada até aquele momento (GANTT, H.L., 1919). Os gráficos de Gantt serviram de base para muitos trabalhos posteriores, até os dias de hoje.

A evolução dos métodos de programação se deu com o início de utilização dos algoritmos computacionais. Pode-se mencionar o *Critical Path Method (CPM)* desenvolvido pela Du Pont em 1956 e o *Performance Evaluation and Review Technique (PERT)*, desenvolvido pela Marinha dos Estados Unidos em 1958 como precursores desta evolução, ao possibilitar, pela utilização dos recursos computacionais, a utilização da simulação.

Estudos mais sistemáticos de *scheduling* podem ser encontrados a partir da década de 50, por exemplo, com a solução desenvolvida por Johnson (1954) para o problema de *flowshop* (condição particular do *jobshop* em que os *Jobs* têm a mesma sequência) com 2 máquinas e no máximo 2 operações por *job* e com a extensão deste modelo para o *jobshop* que foi desenvolvido por Jackson (1956) apud Potts et al. (2009).

Outras contribuições importantes foram dadas também por Jackson (1955) no desenvolvimento de modelos para *scheduling* de uma única máquina, por Mc Naughton (1959) no caso de máquinas paralelas idênticas e por Akers Jr. e Friedman (1955) apud Potts et al. (2009) nos problemas envolvendo dois *jobs*. Ao final da década de 1950 os problemas de *scheduling* de máquinas havia formado um ramo independente da pesquisa operacional. Um sistema de classificação tendo por base os sistemas de processamento subdividia os grupos entre aqueles com um único-estágio, com uma máquina ou diversas máquinas em paralelo, ou multe estágio, com máquinas operando em série (POTTS et al., 2009).

Ao longo da década de 1950 Os modelos inicialmente simples foram ganhando complexidade, a ponto de determinados problemas, como o do *jobshop* 10 x 10 x 10 (10 *jobs*, 10 máquinas e 10 operações por *job*) permanecer sem solução por mais de uma década.

A partir da década de 1960 ganham relevância os algoritmos enumerativos como os de programação dinâmica e, mais particularmente, o *branch and bound*.

O entendimento da complexidade dos problemas e a sua relação com o tempo computacional para se chegar às soluções para os mesmos foi um elemento impulsionador para o desenvolvimento das teorias de *scheduling* na década de 1970.

Entre os anos de 1970 e 1980, outro fator consolidador e impulsionador foi o desenvolvimento de uma classificação para os tipos de problema tratados, os ambientes em que estão inseridos, os tipos de objetivo a alcançar e uma notação específica para descrever esses problemas. Esses fatores são descritos em mais detalhes a seguir.

Muitas pesquisas têm se desenvolvido, com variados algoritmos, heurísticas e até softwares bastante sofisticados propostos, como se verá adiante. Estas soluções, no entanto, encontram-se ainda distantes de uma utilização prática mais intensa, particularmente no Brasil, como se pode constatar da pesquisa conduzida por Giacon (2010) com empresas filiadas à FIESP.

### 2.1.2 *Scheduling* – Modelo Geral e Terminologia

Foi importante para a evolução da pesquisa e organização dos problemas de *scheduling* a metodologia de classificação, proposta por Graham et al. (1979), baseada em três campos: o ambiente do problema, a condição específica de processamento e a função objetivo a ser minimizada.

Como notação universal, um problema de *scheduling* é descrito por um tripé com aspecto  $\alpha | \beta | \gamma$  onde  $\alpha$  representa o ambiente de máquina,  $\beta$  a característica de processamento e  $\gamma$  a função objetivo a ser minimizada (POTTS et al, 2009).

Conforme Pinedo (2008) os modelos determinísticos de *scheduling* têm sido estudados ao longo dos últimos 50 anos, ao longo dos quais uma terminologia se consolidou:

- a) todos os problemas de *scheduling* tratam de um número de *jobs* ( $j$ ) e máquinas ( $m$ ) finito;
- b) par  $(i, j)$  se refere a uma etapa de processo de um *job*  $j$  na máquina  $i$ ;
- c) tempo de processamento ( $p_{ij}$ ): representa o tempo de processamento do *job*  $j$  na máquina  $i$ ;
- d) data de liberação ( $r_j$ ): data em que o *job*  $j$  pode iniciar o processamento;
- e) prazo objetivo ( $d_j$ ): data em que o *job*  $j$  foi prometido ao cliente;
- f) peso ( $w_j$ ): Fator que designa uma prioridade do *job*  $j$ ;
- g) tempo de processamento  $C_j$ : tempo total de processamento do *job*  $j$ ;
- h) tempo total de processamento ( $C_{max}$ ) ou *makespan*: tempo no qual todas as operações de todos os *jobs* foram concluídos.

#### 2.1.2.1 *Scheduling* – Ambientes de Máquina ( $\alpha$ )

Importante na classificação, os ambientes de máquina estudados subdividem-se em:

- a) máquina única (1);
- b) máquinas Paralelas Idênticas ( $P_j$ );
- c) máquinas Paralelas com Diferentes Velocidades ( $Q_m$ );
- d) máquinas Paralelas Diferentes ( $R_m$ );
- e) *flowshop* ( $F_m$ ) –  $m$  máquinas em série, cada *job* processado em cada uma das  $m$  máquinas na mesma sequência;
- f) *flowshop* Flexível ( $FF_c$ ) - associação do *flow shop* com máquinas paralelas;
- g) *jobshop* ( $J_m$ ): ambiente objeto do presente estudo; diferencia-se do *flow shop* em relação à sequência diferenciada de cada *job* pelas máquinas;
- h) *jobshop* Flexível ( $FJ_c$ ) - associação do *job shop* com o ambiente de máquinas paralelas;
- i) *openshop* ( $O_m$ ) – existem máquinas designadas para cada operação de cada *job*, porém não existe uma sequência pré-definida de operações em cada *job*.

### 2.1.2.2 *Scheduling* – Característica de Processamento ( $\beta$ )

Quanto à característica de processamento, os principais processos estudados são:

- a) data de liberação ( $r_j$ ): se  $r_j$  aparecer no campo  $\beta$  o *job*  $j$  não pode ser iniciado antes da data de liberação;
- b) interrupção (*preemption* -  $p_{rmp}$ ): significa que uma operação de um *job* em execução em uma máquina pode ser interrompido e ser completado posteriormente;
- c) precedência (*precedence constraints* – *prec*): uma ou mais operações de um ou mais *jobs* precisam ser completados antes que outros sejam iniciados;
- d) sequência dependente de *set up* ( $s_{jk}$ ): a sequência depende do tempo de *set up* entre as atividades  $j$  e  $k$ ;
- e) recirculação ( $r_c$ ): ocorre quando um *job shop* pode visitar um centro de trabalho mais de uma vez.

### 2.1.2.3 *Scheduling* – Principais funções objetivo estudadas ( $\gamma$ )

Ainda conforme Pinedo (2008), as principais funções objetivo estudadas em problemas de *scheduling* são:

- a) minimizar  $C_{max}$  (*makespan*): normalmente implica em boa utilização das máquinas;
- b) máximo atraso (*Lateness*  $L_{max}$ ): mede a máxima violação do prazo objetivo  $d_j$ ;
- c) mínimo tempo total ponderado ( $\sum w_j C_j$ ), associa os custos de retenção ou estoque ao tempo total de processamento.

### 2.1.3 Teoria da Complexidade

Um divisor de águas na classificação dos problemas de *scheduling* foi a teoria da complexidade, que possibilita ao pesquisador, diante de um problema, saber se é possível desenvolver algum algoritmo mais eficiente do que uma técnica de busca enumerativa para o mesmo (LEUNG, 2004).

O tempo de processamento de um algoritmo é medido basicamente pelo número de etapas de que ele se utiliza e ele é avaliado como uma função dos dados de entrada (*inputs*).

Supondo-se  $A$  um algoritmo com tempo de processamento  $T(n) = O(g(n))$ , onde  $n$  é a quantidade de variáveis do problema, diz-se que  $A$  é um algoritmo de tempo polinomial se  $g(n)$  é uma função polinomial de  $n$ . Diz-se que um problema é tratável se um algoritmo de tempo polinomial pode ser associado a ele. Há uma grande classe de problemas classificados como NP-Difíceis e/ou NP-Completo que são provavelmente intratáveis (LEUNG, 2004).

Conforme Potts et al (2009), os fundamentos desta teoria foram lançados por Edmonds (1965) apud Potts et al. (2009) e consolidados por Garey et al. (1979).

## 2.2 Jobshop

Um problema típico de *scheduling* em um *jobshop* é caracterizado por um conjunto de *Jobs*,  $J_1, J_2, \dots, J_n$ , que tem que ser processado num conjunto  $M$  de diferentes máquinas,  $M_1, M_2, \dots, M_m$ . Cada *Job*  $J_j$  é constituído por  $m_j$  operações  $O_{j1}, O_{j2}, \dots, O_{jm}$ , que precisa ser programado nesta ordem. Além disso cada operação precisa ser processada numa máquina específica, dentre todas as  $m$  existentes. Cada operação  $O_{jk}$  tem um tempo de processamento  $p_{jk}$ . O objetivo pode ser encontrar uma sequência de operações em cada máquina que minimize o *makespan*,  $C_{\max} = \max_{j=1, n} C_j$ , onde  $C_j$  representa o tempo de conclusão da última operação do *Job*  $J_j$  ( $j=1, \dots, n$ ) (LEUNG, 2004).

Este é um problema de alta complexidade, onde apenas poucos casos especiais têm solução conhecida em tempo polinomial. (LEUNG, 2004).

### 2.2.1 O Grafo Disjuntivo

Uma contribuição importante para o desenvolvimento do estudo do *jobshop* foi a representação do problema através de um grafo disjuntivo, formulação proposta por Roy e Sussman (1964) apud Potts et al.(2009).

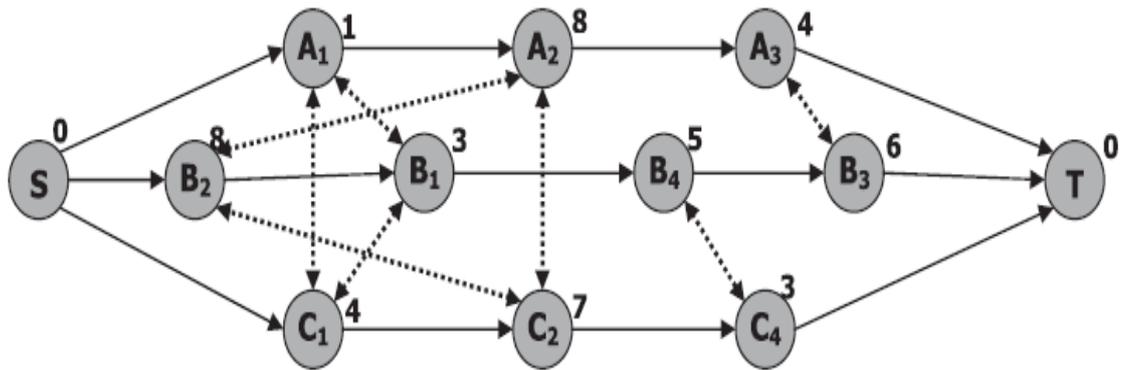


Figura 3 - Grafo disjuntivo para um Jobshop

Fonte: Bartak *et al.*, (2010:266)

Considerando-se o grafo  $G$  da figura 3, onde  $A$ ,  $B$  e  $C$  são *jobs* e os números associados às máquinas, o conjunto de  $N$  nós representa todas operações a serem executadas nos  $n$  *jobs*. Os arcos sólidos, chamados de conjuntivos, representam sequências definidas de *jobs*. Os arcos pontilhados representam 2 *jobs* que precisam ser processados na mesma máquina, ou seja ilustra o compartilhamento. Apontam nas 2 direções e são chamados de disjuntivos. Os números no lado externo da circunferência representa o tempo da operação. Os nós  $S$  e  $T$  são fantasmas adicionados ao modelo.  $J$  representa o conjunto de arcos conjuntivos e  $K$  os disjuntivos. A notação para o grafo da figura é  $G = (N, J, K)$ .

Uma solução possível corresponde à seleção de um arco disjuntivo do par, de forma que o grafo dirigido resultante seja acíclico (não há recirculação). Se  $D$  denotar o subconjunto de arcos disjuntivos e  $G(D)$  o conjunto de arcos conjuntivos mais o subconjunto  $D$ , então o *makespan* de uma solução possível em  $G(D)$  é determinado pelo caminho mais longo entre  $S$  e  $T$ , considerando o tempo de  $T$  igual a zero.

A representação disjuntiva está diretamente relacionada à formulação da programação disjuntiva que é base para a solução de problemas do tipo minimizar  $C_{max}$  num ambiente *jobshop*. Às restrições de precedência são acrescentadas as restrições disjuntivas (PINEDO, 2008).

## 2.2.2 Principais Metodologias para *Scheduling* em um *Jobshop*

### 2.2.2.1 Modelos de Programação Dinâmica e Programação Inteira

Na programação dinâmica o problema original é quebrado em uma sequência de decisões que são tomadas em estágios:

- a) remove-se um elemento do problema;
- b) resolve-se o problema menor;
- c) usa-se a solução do problema menor para adicionar o elemento removido de maneira adequada, produzindo-se uma solução para o problema maior.

Trata-se de uma técnica enumerativa que define critérios para tomada de decisão à medida que se avança na árvore de soluções possíveis. É utilizada principalmente para problemas de pequeno porte, devido ao relativo longo tempo de processamento requerido quando se trata de problemas com maior número de *jobs* e/ou máquinas (MASSOTE, 1991) .

Na mesma linha, modelos de programação matemática, principalmente os de programação inteira têm se constituído em base para modelagem e estudo porém também enfrentam a restrição pelo alto tempo de processamento computacional requerido, quando se trata de problemas com grande quantidade de *jobs* e/ou máquinas, apesar do avanço no desenvolvimento de recursos de inteligência artificial.

#### 2.2.2.2 Modelo de Programação Inteira Binária

Uma representação básica para o problema de minimização de  $C_{max}$  em um *jobshop* é formulado conforme um modelo de programação inteira binária. Partindo-se da representação de um grafo disjuntivo  $G=(G,D)$ , onde  $G=(X,U)$  é o grafo conjuntivo associado com a sequência de *jobs* e  $D$  a sequência de disjunções.

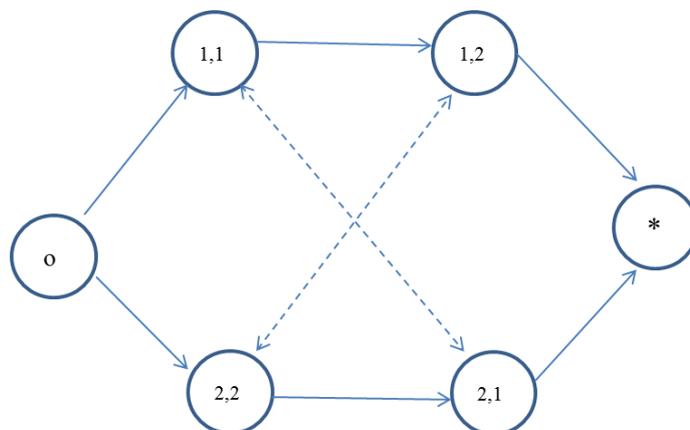


Figura 4 - Grafo Disjuntivo para 2 Jobs e 2 Máquinas

Uma disjunção  $[i,j] = \{(i, j), (j, i)\}$  é associada a cada par de operações processada na mesma máquina. o e \* são operações fantasmas representando o início e final do sequenciamento. O sequenciamento em  $G = (G, D)$  é um conjunto de tempos de início  $T = \{t_i \mid i \in X\}$  tal que:

-as restrições conjuntivas são satisfeitas por:

$$t_j - t_i \geq p_i, \quad \forall (i,j) \in U$$

-as restrições disjuntivas são satisfeitas por:

$$t_j - t_i \geq p_i \vee t_i - t_j \geq p_j, \quad \forall (i, j) \in D$$

-uma formulação inteira mista é definida pela introdução de uma variável binária  $x_{ij}$  para cada disjunção, que especifica se a operação  $i$  precede a operação  $j$  ou não ( $M$  é um inteiro grande):

$$\text{Min } C_{max}$$

$$\text{s.a.} \quad \forall i \in X, t_i \geq 0 \tag{1}$$

$$\forall i \in X, t_i + p_i \leq C_{max} \tag{2}$$

$$\forall (i, j) \in U, t_i + p_i \leq t_j \tag{3}$$

$$\forall [i, j] \in D, t_i + p_i - M(1 - x_{ij}) \leq t_j \tag{4}$$

$$\forall [i, j] \in D, t_j + p_j - Mx_{ij} \leq t_i \tag{5}$$

$$\forall [i, j] \in D, x_{ij} \in [0, 1] \tag{6}$$

Este é um problema NP – difícil, com poucos casos especiais com solução em tempo polinomial, como a desenvolvida por Jackson (1956) a partir da solução de Johnson, ou as soluções gráficas de Akers Jr.(1956) apud Potts et al. (2009) e de Brucker (1988) apud Leung (2004), para o *jobshop* de dois *jobs*.

### 2.2.2.3 O Modelo *Branch and Bound*

Problemas de natureza combinatória envolvendo relações de precedência com determinadas condições a serem satisfeitas, característica do *jobshop*, ensejaram o desenvolvimento de algoritmos enumerativos, sendo particularmente importante o *branch and bound* (B&B), desenvolvido por Land e Doig (1960) apud Potts et al. (2009) para problemas de programação inteira e por Eastman (1958) para o problema do Caixeiro Viajante.

A combinação de heurísticas de busca local com técnicas de enumeração como o *branch and bound* é uma das formas de programação dinâmica mais usada.

O algoritmo *branch & bound* (B&B) para problemas de minimização é caracterizado por:

- a) regra de ramificação (*branching*) que define como o problema é dividido em um subconjunto de problemas;
- b) regra para *limite inferior* que computa o limite inferior para os custos de qualquer solução dentro do subconjunto de soluções em consideração;
- c) configurações opcionais, tais como *regras para limite superior* que constroem soluções viáveis que podem ser avaliados em relação à produção de um limite superior
- d) *regras de dominância* que eliminam alguns subconjuntos de solução das análises posteriores.

Analogamente pode-se aplicar o mesmo raciocínio multiplicado por (-1) para problemas de maximização.

O *B&B* é comumente representado por uma árvore de busca, onde cada nó representa um subgrupo de soluções, e os ramos (*branches*) correspondem à divisão da solução de acordo com a regra de ramificação.

Exemplificando a ideia deste algoritmo, considere-se o seguinte problema, extraído de Wolsey (1998) apud Camponogara (2006):

$$S: z = \text{Max } 4x_1 - x_2 \quad (7)$$

$$\text{s.a.: } 7x_1 - 2x_2 \leq 14 \quad (8)$$

$$x_2 \leq 3 \quad (9)$$

$$2x_1 - 2x_2 \leq 3 \quad (10)$$

$$x \in \mathbf{Z}_+^2$$

Relaxação é a técnica em que algumas restrições do problema são desconsideradas, trabalhando-se com subproblemas mais simples, porém dentro dos limites de solução do problema original.

Para um problema de maximização qualquer solução viável impõe um limite inferior, chamado limite *primal*. Utiliza-se a relaxação para se obter limites superiores, chamados limites *duais*.

No *B&B* utiliza-se a relaxação linear.

No problema acima:

Limitação (*bounding*): obtém-se o primeiro limite superior ao se resolver a relaxação linear,

$$R(S), \text{ com } \bar{z} = \frac{59}{7} \text{ com } (\bar{x}_1, \bar{x}_2) = \left(\frac{20}{7}, 3\right), \text{ assume-se } \underline{z} = -\infty \quad (11)$$

Ramificação (*branching*): uma vez que  $\underline{z} \leq \bar{z}$ ,  $S$  é quebrado em subproblemas:

$$S_1 = S \cap \{x : x_j \leq \bar{x}_j\} \quad (12)$$

$$S_2 = S \cap \{x : x_j \geq \bar{x}_j\} \quad (\text{vide fig.1}) \quad (13)$$

Escolhendo-se um nó arbitrariamente,  $S_1$ :

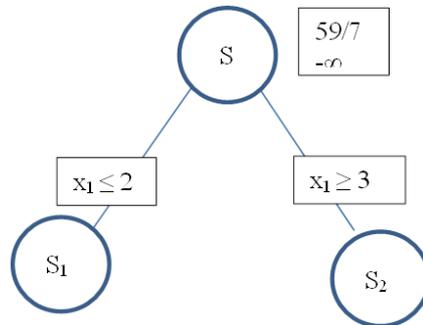


Figura 5 - Quebra do primeiro nó da árvore B&B

Limitação: resolvendo-se a relaxação linear  $R(S_1)$  associada a  $S_1$ :

$$S_1: \bar{z}_1 = \text{Max } 4x_1 - x_2 \quad (13)$$

$$\text{s. a. } 7x_1 - 2x_2 \leq 14 \quad (14)$$

$$x_2 \leq 3 \quad (15)$$

$$2x_1 - 2x_2 \leq 3 \quad (16)$$

$$x_1 \leq 2 \quad (17)$$

cuja solução ótima é  $(2, \frac{1}{2})$ , que por sua vez induz o limite superior  $\bar{z} = \frac{15}{2}$  (18)

Ramificação: quebra-se o subconjunto  $S_1$  em dois subconjuntos:

$$S_{11} = S_1 \cap \{x : x_2 \leq 0\} \text{ e} \quad (19)$$

$$S_{12} = S_1 \cap \{x : x_2 \geq 1\}, \text{ o que faz com que a lista de nós ativos se torne} \quad (20)$$

$$L = \{S_2, S_{11}, S_{12}\}, \text{ conforme figura 2.} \quad (21)$$

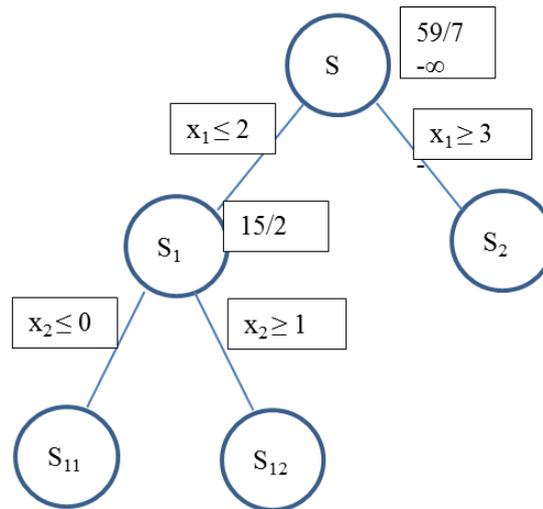


Figura 6 - Dividindo  $S_1$  em  $S_{11}$  e  $S_{12}$

Escolhendo um nó: arbitrariamente escolhe-se o nó  $S_2$

Limitação: analogamente ao processo anterior resolve-se a relaxação linear

$R(S_2)$ :

$$S_2: z_2 = \text{Max } 4x_1 - x_2 \quad (22)$$

$$\text{s.a. } 7x_1 - 2x_2 \leq 14 \quad (23)$$

$$x_2 \leq 3 \quad (24)$$

$$2x_1 - 2x_2 \leq 3 \quad (25)$$

$$x_1 \geq 3 \quad (26)$$

$S_2$  é infactível, uma vez que  $\bar{z}_2 = -\infty$ , pode-se cortar  $S_2$  em função de inviabilidade.

Repete-se o processo para  $S_{12}$ :

$$R(S_{12}) \text{ tem um espaço de soluções factíveis } S_{12} = S \cap \{x: x \leq 2 \text{ e } x_2 \geq 1\}, \quad (27)$$

$$\text{obtendo-se a solução } \bar{x}_{12} = (2, 1), \text{ que gera o limite superior } \bar{z}_{12} = 7. \quad (28)$$

Sendo inteira, obtém-se como limite inferior  $\underline{z}_{12} = 7$ , ou seja, neste ramo chegou-se a um valor ótimo = 7, o que faz com que este nó seja interrompido neste ponto.

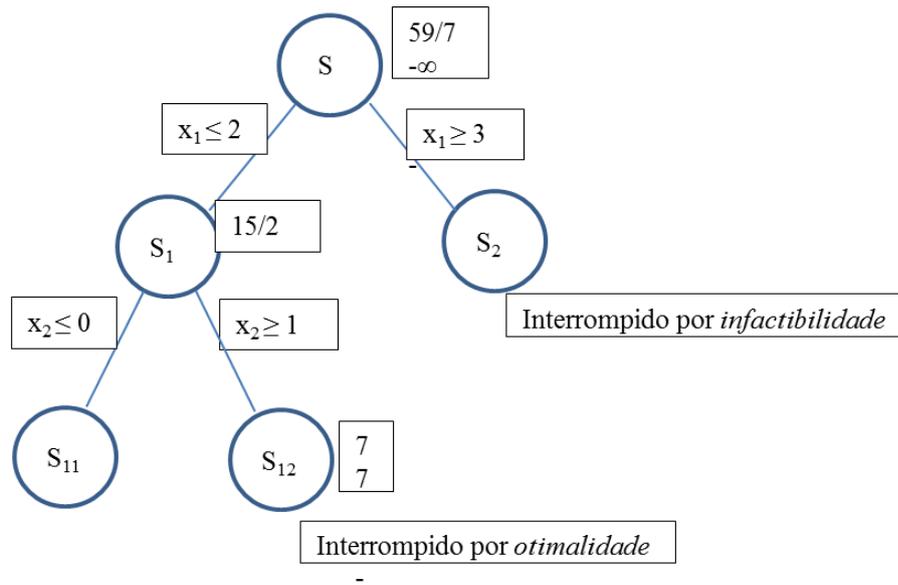


Figura 7 - Análise do nó  $S_{12}$ : interrupção por *otimalidade*

Restará o nó  $S_{11}$

Limitação:  $S_{11} = S \cap \{ x : x_1 \leq 2 \text{ e } x_2 = 0 \}$ , onde (29)

$R(S_{11})$  tem como solução  $\bar{x}_{11} = (\frac{3}{2}, 0)$ , cujo limite superior  $\bar{z}_{11} = 6$ . (30)

Sendo  $z_{11} < 7$ , solução do outro ramo  $S_{11}$  é descartada.

Portanto  $x^* = (2, 1)$  é a solução ótima, com  $z^* = 7$ . (31)

Algumas soluções exatas baseadas em *branch and bound* são:

- baseado em ramificação de arcos disjuntivos em um grafo disjuntivo (NÉMETI, 1964, apud POTTS et al, 2009);
- baseado em pesquisa de operações críticas em um grafo disjuntivo (BALAS, 1969);
- baseados em *job* crítico e *jobs* com data de entrega superiores a este (McMAHON et al, 1975);
- solução de problema de 10 máquinas, 10 *jobs*, que ficou insolúvel durante vinte anos (CARLIER et al, 1989);
- idem anterior (APPLEGATE et al, 1991);
- método de programação de restrições (BAPTISTE et al., 1996 apud LEUNG, 2004);
- Shifting Bottleneck Procedure* (ADAMS et al, 1988) –é um dos métodos mais bem sucedidos e aplicados: a cada etapa a entrada e a saída de uma máquina gargalo é otimizada, utilizando-se o algoritmo de Carlier. Uma vez concluído, o processo é

repetido para a próxima máquina gargalo, e assim sucessivamente. Trata-se de um algoritmo de otimização bastante efetivo para problemas relacionados ao *makespan*, mas que também tem sido utilizado como um procedimento heurístico, minimizando a utilização de tempo computacional, neste caso sem garantir a otimalidade (BAKER, 1995).

#### 2.2.2.4 Métodos Heurísticos

Muitos métodos heurísticos são propostos na literatura para o *scheduling* de *jobshops*, tais como as regras de prioridade de despacho, o *simulated annealing*, a busca tabu, ou os algoritmos genéticos.

Trata-se de métodos que, de alguma forma, reduzem o espaço de busca e, portanto reduzem o tempo computacional para a busca da solução e têm sido bastante utilizadas em combinação com a metodologia *branch and bound*. Como desvantagem, geralmente não levam à solução ótima.

Particularmente importantes têm sido as soluções baseadas em regras de eliminação, que permitem reduzir consistentemente o trabalho de busca nas árvores de possibilidades geradas pelo método *branch and bound*. Particularmente importantes são os métodos classificados como de passe único, ou seja, definem um único caminho na árvore de possibilidades de sequenciamento. Dentre os métodos de passe único as regras de despacho têm sido amplamente utilizadas isoladamente ou em associação com outras heurísticas ou algoritmos já conhecidos para as mais diversas situações de sequenciamento (LEUNG, 2004).

O despacho é uma atividade que permite a tomada de decisões que afetem uma determinada máquina concomitantemente à sequência em que as atividades acontecem, ou seja o sequenciamento não precisa ser definido a priori, de uma vez, mas apenas no momento em que se faz necessário, ou seja, quando uma máquina fica livre. São práticas que, por permitirem uma rápida adaptação à dinâmica de chegada das tarefas, às quebras de máquina, por exemplo, e a outros fatores que normalmente ocorrem no chão de fábrica, são bastante utilizadas. As regras definidas permitem varrer toda a lista de operações a serem executadas do início ao fim, construindo a sequência da esquerda para a direita no gráfico de Gantt, amarradas por alguma regra que relaciona a operação precedente com a próxima (BAKER, 1995).

As heurísticas geradas constroem um caminho a percorrer na árvore *branch and bound*.

Algumas das muitas regras de despacho utilizadas são:

- a) *FIFO* – *first in first out* – o primeiro a chegar ao centro de trabalho é o primeiro a ser atendido;
- b) *EDD* – *earliest due date* – a operação com a data prometida de entrega mais próxima é a primeira a ser atendida;
- c) *LPT* – *longest processing time* - seleciona-se a operação com maior tempo de processamento
- d) *SPT* – *shortest processing time* – seleciona-se a operação com menor tempo de processamento;
- e) *MWKR* – *most work remaining* – seleciona-se a operação associada ao *job* com maior quantidade de trabalho remanescente;
- f) *LWKR* – *least work remaining* - seleciona-se a operação associada ao *job* com menor quantidade de trabalho remanescente.

Com relação a outros métodos heurísticos, podem ser citados:

- a) métodos baseados em busca tabu: Barnes et al (1991) e Nowicki et al (1996),
- b) métodos baseados em algoritmo genético – Dorndorf et al (1995).

A escolha da melhor regra depende muito da atividade e dos objetivos a alcançar e a simulação é a melhor forma de avaliá-las, podendo mais de uma regra ser utilizada de forma conjugada.

No Brasil diversos trabalhos têm sido publicados apresentando a utilização das regras de despacho em situações diversas:

- a) Morabito et al (2002) estudaram a aplicação de regras de despacho à atividade de descarga de caminhões em usina de beneficiamento de cana, aplicando o *software* “ARENA”;
- b) Mainieri et al (2010) estudaram o objetivo de entrega em *flowshops* utilizando as regras de despacho *MDD* (máximo entre data de entrega e soma dos tempos de processamento de tarefas em estágios restantes) e *PRTT* (regra de prioridade para atraso total);
- c) França et al (1995) estudaram a programação em máquina simples com tempo de preparação dependente da sequência comparando o desempenho de uma heurística baseada em busca tabu a regras de despacho *EDD* (menor data de entrega) e *SPT*

- (menor tempo de processamento), mostrando resultado bastante inferior destes dois últimos;
- d) Domingos et al (2008) compararam estratégias lançadas em uma base de conhecimento *Fuzzy* com regras despacho *SPT*, *EDD* e *LPT* , propondo rápidas simulações para escolha da melhor política;
  - e) Rodrigues (2006) apresentou, em tese de mestrado, um estudo de complexas regras de despacho aplicadas à programação de caminhões de transporte em minas a céu aberto;
  - f) Ronconi et al (2000) estudaram as regras *EDD*, *SPT* em comparação à regra PSK (Panwalkar, Smith, Koulamos – regra de priorização da atividade de menor atraso) aplicadas a uma estamperia, com resultado superior desta última.

#### 2.2.2.5 Modelo de Programação de Restrições

Mais recentemente Barták et al. (2010) apresentaram pesquisa focando especificamente a evolução dos modelos de satisfação de restrições (*Constraint Satisfaction Models*), aplicados ao planejamento e *scheduling*. Para começar, uma definição formal do Problema de Satisfação de Restrições (*Constraint Satisfaction Problem – CSP*) é estabelecida:

- a) domínio de uma variável – conjunto de todos os valores que a variável pode assumir, normalmente domínios discretos;
- b) instanciação: é um conjunto de pares  $(x_i; a_i)$  tais que:
  - a.  $x_i$  é uma variável;
  - b.  $a_i$  é um valor do domínio de  $x_i$ ;
  - c. cada variável aparece no máximo uma vez na instanciação;
  - d. a instanciação é dita completa para um conjunto de variáveis  $X$  se cada variável de  $X$  aparece na instanciação e nenhuma outra variável aparece lá. Se a instanciação de  $X$  não contém todas as variáveis de  $X$ , então ela é incompleta (parcial) de  $X$ . Diz-se que a instanciação  $I$  estende a instanciação  $J$  se  $I \supseteq J$ ;
- c) restrição (*constraint*): é um par  $(t, R)$  onde  $t$  é um conjunto de variáveis (chamado escopo; o tamanho do escopo é chamado *arity*) e  $R$  é o conjunto de instanciações completas de  $t$  (domínio da restrição). A restrição limita os valores que as variáveis podem assumir simultaneamente. A instanciação  $I$  satisfaz a restrição  $(t, R)$  se existe  $J \in R$  tal que  $I$  estende  $J$ . Restrição pode ser especificada extensivamente como um conjunto de *tuples* (instanciações satisfatórias).

Resumindo, *CSP* consiste de:

- a) conjunto de variáveis  $X$  ( $x_1, x_2, \dots, x_n$ );
- b) conjunto de domínios  $D$  ( $D_1, D_2, \dots, D_n$ ) tal que para cada  $x_i \in X$  existe um domínio  $D_i$ ;
- c) conjunto de restrições  $C$  ( $c_1, c_2, \dots, c_k$ ) tal que o escopo de cada restrição é um subconjunto de  $X$ ;
- d) solução – completa instanciação das variáveis de  $X$  satisfazendo todas as restrições em  $C$ ;
- e) consistência – se um *CSP* tem pelo menos uma solução diz-se que ele é consistente;
- f) modelagem de um *CSP* – descrição do problema utilizando a sintaxe acima;
- g) a partir destas definições Barták et al. (2010) apresentam e classificam as diversas técnicas para abordagem do *CSP*:
  - a) técnicas de consistência – utilizam algoritmos específicos para redução do domínio das variáveis;
  - b) técnicas de busca – algoritmos que exploram sistematicamente o espaço amostral instanciando as variáveis uma após a outra de forma que cada instanciação particular é consistente. Se isso não for possível para uma variável na sua vez, ou seja, se os possíveis valores estiverem em conflito com alguma restrição em função de alguma premissa anterior, utiliza-se da técnica de *backtracking*, retornando-se à última instanciação consistente:
    - i. algoritmos de busca completa – percorrendo todo o espaço amostral, garantem chegar a uma solução, porém requerem um tempo muito longo para a busca. Destacam-se os métodos “gerar e testar”, que consistem em imaginar uma solução e verificar se atende todas as restrições, repetindo-se o processo até que uma solução efetiva seja encontrada;
    - ii. algoritmos de busca incompleta – técnicas que não exploram todo o espaço amostral e, portanto, não garantem chegar a uma solução ótima. Meta-heurísticas que vão desde os algoritmos evolucionários até as técnicas de busca local estão incluídos neste grupo.

Uma visão mais prática dos modelos de abordagem do *CSP* pode ser visto na abordagem de Kanet et al. (2004) apud Leung (2004), que trata a metodologia genericamente como *Constraint Programming (CP)*.

A abordagem *CP* utiliza um estilo declarativo de enunciação do problema, associado a algoritmos de resolução de restrições para solucionar o problema de otimização combinatória. Utilizando uma linguagem própria, o *CP* formula os problemas genericamente por meio de variáveis discretas, dentro de domínios finitos e um conjunto de restrições envolvendo essas variáveis, aplicando sistematicamente técnicas de busca para solucionar as restrições. Para superar a natureza combinatória de problemas *NP*-completos a *CP* utiliza algoritmos de filtragem com base lógica para cada restrição, o que sistematicamente reduz o domínio dessas variáveis. Como o domínio da variável é reduzido, cada restrição que utiliza a variável é ativada para utilização de seus algoritmos de filtragem associados. Esse processo sistemático é chamado de propagação de restrições e redução de domínio. Esse processo para solução de problemas de restrição (*Constraint Solution Problem, CSP*) pode ser tratado por meio de um algoritmo genérico, conforme ilustrado na figura 8:

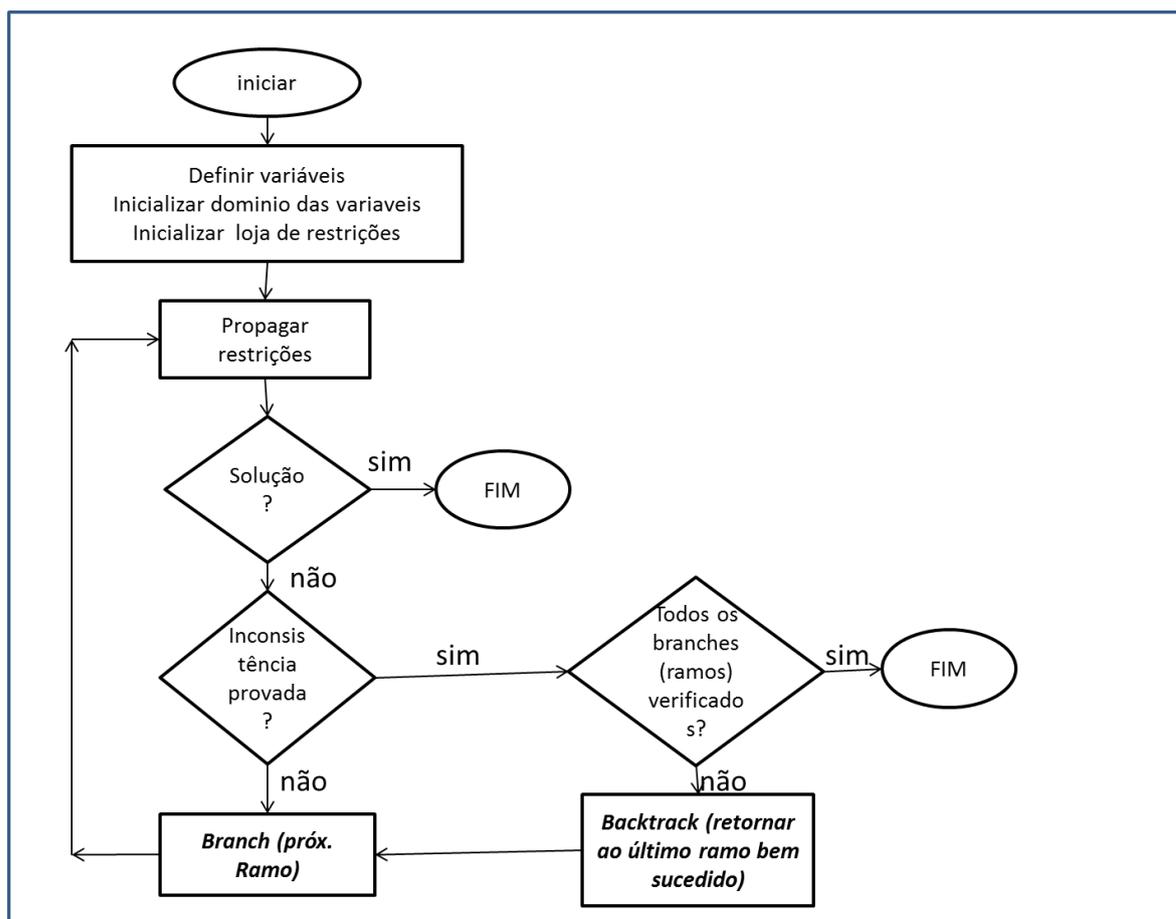


Figura 8 - Algoritmo Genérico para o *CSP*

Fonte: Kanet et al (2004), apud Leung (2004)

Uma das técnicas de propagação de restrições utiliza o conceito de verificação de consistência dos arcos de forma a comunicar a redução do domínio das variáveis entre e por meio das restrições envolvendo variáveis específicas.

O exemplo abaixo extraído de Kanet et al. (2004) apud Leung (2004) ilustra, nas figuras 9 e 10 este modelo de propagação de restrições. Sejam:

- a) ambiente: máquina simples;
- b) 2 *jobs*;
- c)  $C_1$ : tempo para completar o *job*<sub>1</sub>;
- d)  $s_1$ : tempo em que o *job*<sub>1</sub> estará liberado para iniciar;
- e) tempo de ciclo do *job*<sub>1</sub> = 3;
- f)  $C_1 = s_1 + 3$ ; (32)
- g) outras restrições: *job*<sub>1</sub> não pode ser interrompido, cada máquina processa 1 *job* de cada vez;
- h)  $C_2 = s_2 + 2$ ; (33)
- i)  $d_1$ : prazo objetivo para *job*<sub>1</sub> = 5, ou seja,  $C_1 < 5$ .

A redução do domínio da variável  $C_1$  é feita inicialmente aplicando-se a restrição ( $C_1 = s_1 + 3$ ). A verificação da consistência bi direcional do arco faz com que se reduza também o domínio de  $s_1$ . Esta análise da consistência do arco  $C_1 s_1$  é ilustrada na figura 9:

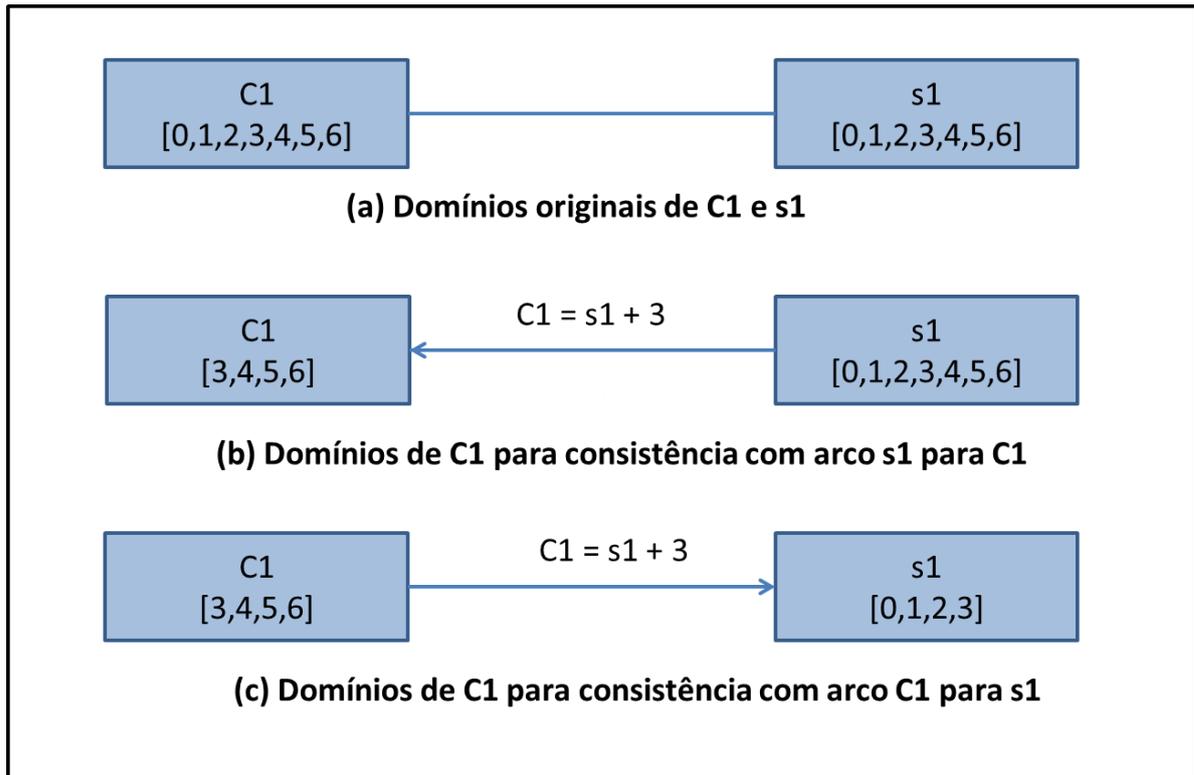


Figura 9 - Redução de domínio por meio da verificação lógica da consistência do arco

Fonte: Kanet et al. (2004), apud Leung (2004)

O mesmo conceito permite a comunicação entre restrições envolvendo variáveis comuns para reduzir seus domínios. Aplicando-se a propagação a  $C_1$  e  $C_2$  e incluindo-se a restrição  $d_1 = 5$ , chega-se ao domínio final do problema, conforme se pode ver na figura 10:

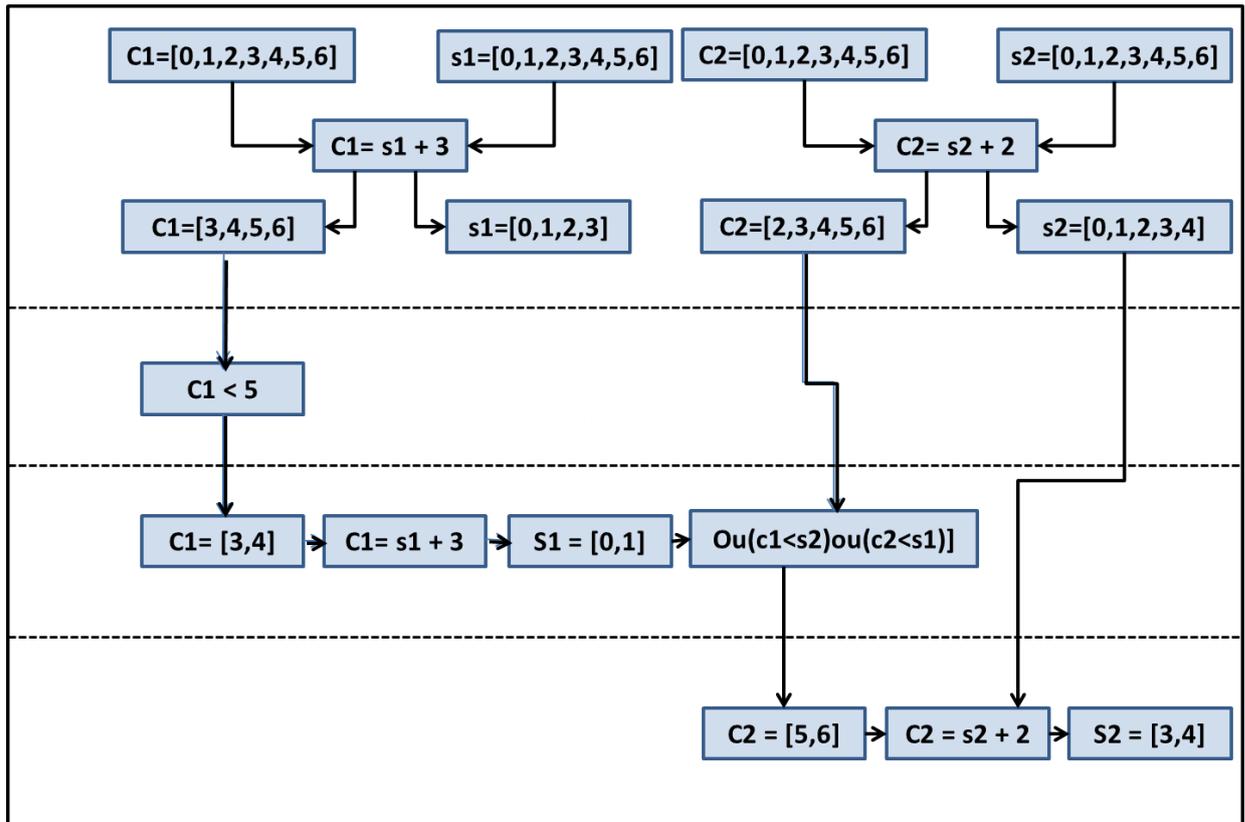


Figura 10 - Ilustração da Propagação Lógica da Restrição

Fonte: Kanet et al.(2004) , apud Leung (2004)

Alguns estudos, como o de Darby-Dowman et al (1997) apud Leung (2004) compararam a performance da metodologia *CP* para redução de domínio com técnicas de Programação Inteira e concluíram que o modelo *CP* se comportou de forma mais previsível por reduzir o espaço de busca mais rapidamente.

### 3. SCHEDULING *JOB*SHOP NA PRÁTICA

#### 3.1 Contexto do Trabalho

Foram descritos nas seções anteriores o escopo e a importância do *scheduling* no desenvolvimento e nos resultados das organizações industriais.

Na prática, no entanto, esta atividade de planejamento na qual ela se insere, não é contemplada plenamente pela maior parte dos sistemas *ERP*, que dão sustentação ao planejamento na maior parte das organizações industriais, particularmente em ambientes *jobshop* (METAXIOTIS et al., 2003).

Estes recursos têm sido disponibilizados em sistemas específicos conhecidos como “*EP - Expert Systems*”, ou “*APSS*” (*Advanced Planning Systems*). Já em 1994 uma pesquisa realizada nos Estados Unidos com empresas do *Fortune 500* industrial, em organizações com uma média de 19000 empregados e US\$ 6.2 bilhões de faturamento médio mostraram que 52% delas faziam uso de *EPs* a maioria das quais em *scheduling* (WONG et al, 1994).

Alguns dos sistemas mencionados são o ISIS, o OPIS, o Patriarch, e o OPT (METAXIOTIS et al., 2001; KHATAWALA et al., 1993), além do ILOG Solver®, o OPL Studio® e o CPLEX® (LEUNG, 2004).

Este último *software*, disponibilizado no mercado pela IBM Inc., é um dos que possuem um módulo específico desenvolvido com base na programação de restrições, que se distingue de *softwares* de programação matemática por permitir a introdução de restrições especializadas e lógicas e só trabalharem com variáveis de decisão discretas ou booleanas.

O desenvolvimento de *softwares* tem sido o maior elemento de suporte para o progresso da tecnologia de *scheduling*, graças, principalmente, ao avanço dos recursos de Inteligência Artificial.

Retornando, no entanto, à pesquisa realizada com empresas filiadas à FIESP, conforme mencionado na seção I.2, a utilização de *softwares* ou sistemas mais avançados de suporte ao *scheduling* estão ainda distantes da nossa realidade (GIACON, 2010).

Quando se foca particularmente empresas de menor porte esta realidade se torna ainda mais evidente e, neste contexto, sistemas mais simples, baseadas em regras de despacho simples desenvolvidos em planilha *Excel*, como o trabalho proposto por Girotti et al. (2011), podem ser uma alternativa. A questão, neste caso, é se verificar a relação custo – benefício entre a solução encontrada e a solução ótima.

Para esta análise uma comparação foi realizada entre um sequenciamento de um *jobshop* de 20 *jobs*, 5 máquinas, 5 operações estabelecido com a metodologia de programação

de restrições, através da utilização do software *IBM ILOG CPLEX 12.3*, em comparação a sequências estabelecidas utilizando-se as regras de despacho *LPT* (maior tempo de processamento) e *MWKR* (maior trabalho remanescente) , métodos desenvolvidos em uma planilha *Excel*, tendo como função objetivo minimizar o *makespan*.

### 3.2 Modelos estudados

#### 3.2.1 Regra de Despacho *LPT*

A heurística utilizada para aplicação da regra de despacho *LPT* foi desenvolvida em planilha *Excel*, conforme a seguinte lógica:

- a) a partir do programa de pedidos colocado, são inicialmente definidos os *jobs* com suas respectivas sequências de operação máquina por máquina, na condição de um *jobshop* clássico, ou seja, cada *job* tem uma sequência de operações nas máquinas pré determinada, não há interrupção de cada operação na máquina e cada *job* passando somente uma vez em uma máquina pré determinada;
- b) são inicialmente separadas as operações iniciais para processamento. Para isso foram consideradas todas as primeiras operações;
- c) dentre as operações disponíveis aplica-se a regra despacho *LPT* – operação com maior tempo de processamento – em cada máquina que tenha operação inicial a processar;
- d) aplicando-se a regra de despacho sucessivamente estabelece-se a sequência das diversas operações iniciais nas máquinas;
- e) para cada operação é definido seu tempo de início e de conclusão, considerando-se seu tempo de processamento;
- f) seleciona-se as diversas segundas operações a executar em cada máquina;
- g) verifica-se o tempo da liberação da máquina para a operação anterior;
- h) aplica-se a regra de despacho para definição da sequência dentre as segundas operações aguardando para processamento;
- i) verifica-se se a operação indicada para a sequência já foi liberada pela máquina que processou a operação anterior. Caso negativo, se faz a verificação com a operação seguinte e assim por diante;
- j) com a sequência definida, são estabelecidos os tempos de início e fim de cada operação;

- k) repete-se o processo de definição do sequenciamento para a segunda operação para a terceira e assim por diante.

### 3.2.2 Regra de Despacho *MWKR*

Uma segunda regra despacho segundo a heurística conhecida como *Most Work Remaining - MWKR*, ou seja priorizar os *Jobs* com maior tempo total remanescente de processo foi desenvolvida e testada, como evolução do modelo anterior.

As alterações introduzidas na solução em Excel anterior rotinas em *VB* de tal forma que cada operação tivesse associado o tempo total remanescente para conclusão do *job* e se separasse uma planilha com a atividade em execução e outra com atividades futuras liberadas para produção.

Assim, o fluxo da heurística fica:

- a) carregamento inicial das máquinas é feito da mesma forma que na planilha anterior;
- b) uma planilha de eventos futuros é criada, colocando-se em frente a cada máquina as operações liberadas para processamento, em sequência decrescente de tempos remanescentes de processamento do respectivo *job*;
- c) numa planilha geral é possível ver as atividades iniciadas em cada máquina, o momento que a atividade começou e o tempo de operação previsto. Avança-se até o momento de conclusão da primeira das atividades em execução. Registra-se o tempo de finalização dessa operação. Verifica-se na planilha de eventos futuros se há operações em espera para execução nesta máquina e, se positivo, carrega-se a mesma e se registra o tempo de início. Caso negativo a máquina ficará ociosa até que uma próxima operação fique disponível para execução. Verifica-se a próxima operação no mesmo *job* que acabou de ser concluído. A máquina designada para executá-la está livre? Se positivo carrega-se a mesma na máquina e se registra o tempo de início na mesma. Se negativo a operação é colocada em espera na planilha de eventos futuros na sua devida sequência de prioridade;
- d) na planilha geral avança-se com o relógio até a próxima operação a ser concluída e se repete a mesmo processo de análise anterior. Caso haja duas operações terminando ao mesmo tempo, prioriza-se pela regra de despacho. O processo é repetido até que a última operação seja concluída.

### 3.2.3 Programação de Restrições

Para a solução obtida através da programação de restrições foi utilizado com um código padrão específico para ambiente *jobshop*. O algoritmo utilizado pelo programa é aquele apresentado anteriormente na figura 8.

A Programação de Restrições é uma abordagem para formular e solucionar problemas de satisfação de restrições com variáveis discretas (inteiras, booleanas, simbólicas, conjunto de elementos e subconjunto de conjuntos) utilizando sistematicamente o raciocínio dedutivo, ou seja, técnicas de programação lógica associadas a técnicas de programação matemática. Softwares disponíveis dispõem de ampla gama de operadores lógicos e matemáticos, além de funções com sintaxe especificamente desenvolvida para modelar problemas detalhados de *scheduling*.

No software estudado, IBM ILOG®, há um módulo específico para a otimização *CP*. Dentro de um ambiente integrado de desenvolvimento, utiliza-se uma linguagem de modelação própria para desenvolvimento de modelos de otimização de problemas detalhados de *scheduling*.

Um típico problema de *scheduling* é definido por:

- a) um conjunto de intervalos de tempo – definição de atividades, operações ou tarefas a serem completadas, que podem ser mandatórias ou opcionais;
- b) um conjunto de restrições temporais – definição de uma possível relação entre início e fim dos intervalos;
- c) um conjunto de restrições especializadas – definição da complexa relação dentro do conjunto de intervalos devido à capacidade finita dos recursos;
- d) uma função custo – por exemplo, o tempo requerido para executar o conjunto de tarefas, o custo de não execução de um conjunto de tarefas opcionais, ou os custos devido a penalidade por não cumprimento de prazos de entrega.

As principais etapas para desenvolvimento do modelo são:

- a) descrever em linguagem de modelação o objetivo e o que é desconhecido;
- b) declarar variáveis de decisão para intervalos e precedência das tarefas;
- c) computar o término de cada tarefa e definir a função objetivo;
- d) definir as restrições de recurso;
- e) adicionar restrições alternativas para acelerar a busca.

### 3.3 Simulação

Para simulação dos modelos estudados foi criado um problema com 20 *jobs* em um ambiente constituído de 5 máquinas, com cada produto sendo processado em 5 operações, passando por cada uma das 5 máquinas numa sequência pré-estabelecida. Os pedidos são relacionados conforme suas entradas, nas quantidades solicitadas. A tabela 1, a seguir, resume os dados iniciais do problema:

-o pedido; o produto; a quantidade pedida; a sequência de operações com a indicação da máquina e o tempo operacional unitário.

Tabela 1 - Dados Iniciais do problema com 20 *jobs*, 5 máquinas e 5 operações

Pedido	Produto	Qtde.	OP O1	topu min	OP O2	topu min	OP O3	topu min	OP O4	topu min	OP O5	topu min
0	05	200	M1	2,4	M2	3	M3	3	M0	2,4	M4	3
1	02	250	M0	1,2	M3	1,2	M4	1,2	M2	1,2	M1	1,2
2	04	300	M1	0,6	M2	1,2	M3	1,2	M0	0,6	M4	1,2
3	01	200	M4	1,8	M2	1,8	M3	1,8	M0	1,8	M1	1,8
4	03	300	M1	1,2	M3	1,6	M4	1,6	M0	1,2	M2	1,6
5	01	100	M4	1,8	M2	1,8	M3	1,8	M0	1,8	M1	1,8
6	03	150	M1	1,2	M3	1,6	M4	1,6	M0	1,2	M2	1,6
7	01	100	M4	1,8	M2	1,8	M3	1,8	M0	1,8	M1	1,8
8	03	150	M1	1,2	M3	1,6	M4	1,6	M0	1,2	M2	1,6
9	02	300	M0	1,2	M3	1,2	M4	1,2	M2	1,2	M1	1,2
10	04	200	M1	0,6	M2	1,2	M3	1,2	M0	0,6	M4	1,2
11	03	150	M1	1,2	M3	1,6	M4	1,6	M0	1,2	M2	1,6
12	05	200	M1	2,4	M2	3	M3	3	M0	2,4	M4	3
13	01	200	M4	1,8	M2	1,8	M3	1,8	M0	1,8	M1	1,8
14	03	150	M1	1,2	M3	1,6	M4	1,6	M0	1,2	M2	1,6
15	04	100	M1	0,6	M2	1,2	M3	1,2	M0	0,6	M4	1,2
16	03	600	M1	1,2	M3	1,6	M4	1,6	M0	1,2	M2	1,6
17	05	100	M1	2,4	M2	3	M3	3	M0	2,4	M4	3
18	01	300	M4	1,8	M2	1,8	M3	1,8	M0	1,8	M1	1,8
19	02	400	M0	1,2	M3	1,2	M4	1,2	M2	1,2	M1	1,2

Nota: Dados Trabalhados pelo Autor

Considerando um *job* igual a uma ordem para execução de um pedido, tem-se a alocação total de tempos de operação de cada ordem ou *job* em cada máquina, conforme tabela 2 a seguir.

Tabela 2 - Dados Sintetizados do problema com 20 *jobs*, 5 máquinas e 5 operações

JOB	OP O1	top h	OP O2	top h	OP O3	top h	OP O4	top h	OP O5	top h
0	M1	8	M2	10	M3	10	M0	8	M4	10
1	M0	5	M3	5	M4	5	M2	5	M1	5
2	M1	3	M2	6	M3	6	M0	3	M4	6
3	M4	6	M2	6	M3	6	M0	6	M1	6
4	M1	6	M3	8	M4	8	M0	6	M2	8
5	M4	3	M2	3	M3	3	M0	3	M1	3
6	M1	3	M3	4	M4	4	M0	3	M2	4
7	M4	3	M2	3	M3	3	M0	3	M1	3
8	M1	3	M3	4	M4	4	M0	3	M2	4
9	M0	6	M3	6	M4	6	M2	6	M1	6
10	M1	2	M2	4	M3	4	M0	2	M4	4
11	M1	3	M3	4	M4	4	M0	3	M2	4
12	M1	8	M2	10	M3	10	M0	8	M4	10
13	M4	6	M2	6	M3	6	M0	6	M1	6
14	M1	3	M3	4	M4	4	M0	3	M2	4
15	M1	1	M2	2	M3	2	M0	1	M4	2
16	M1	12	M3	16	M4	16	M0	12	M2	16
17	M1	4	M2	5	M3	5	M0	4	M4	5
18	M4	9	M2	9	M3	9	M0	9	M1	9
19	M0	8	M3	8	M4	8	M2	8	M1	8

Nota: Dados Trabalhados pelo Autor

Este problema foi resolvido inicialmente com a utilização das regras de despacho *LPT* e *MWKR* e em seguida comparado à solução ótima desenvolvida com a utilização da programação de restrições, todas com o mesmo objetivo de minimizar o *makespan*.

Os resultados estão detalhados e comentados a seguir.

### 3.3.1 Solução com a Regra *LPT*

As tabelas 3 e 4 mostram todo o sequenciamento, com cada *job*, designada pela letra J com seu respectivo número, a sequência em cada uma das 5 máquinas, com seu respectivo número de operação, o tempo de início de operação ( $t_i$ ) e o tempo no qual a mesma será finalizada ( $t_f$ ).

O tempo total, ou seja o *makespan* resultante  $C_{max} = 148$  h.

Para melhor visualização, o resultado foi agrupado e sintetizado na tabela 5 e apresentado em gráficos de Gantt, inicialmente com a visão do sequenciamento de cada máquina (gráficos de 1 a 3) e, em seguida, com a visão de cada *job* (gráficos 4 a 6).

Tabela 3 - Solução *LPT* no formato original – parte 1

Job	Máquina	Pedido	Produto	Operação	top(h)	ti(h)	tf(h)	t total (h)
J0	M1	0	P05	O1	8,0	20,0	28,0	148,0
J0	M2	0	P05	O2	10,0	34,0	44,0	
J0	M3	0	P05	O3	10,0	86,0	96,0	
J0	M0	0	P05	O4	8,0	97,0	105,0	
J0	M4	0	P05	O5	10,0	105,0	115,0	
J1	M0	1	P02	O1	5,0	14,0	19,0	
J1	M3	1	P02	O2	5,0	46,0	51,0	
J1	M4	1	P02	O3	5,0	65,0	70,0	
J1	M2	1	P02	O4	5,0	87,0	92,0	
J1	M1	1	P02	O5	5,0	104,0	109,0	
J2	M1	2	P04	O1	3,0	50,0	53,0	
J2	M2	2	P04	O2	6,0	55,0	61,0	
J2	M3	2	P04	O3	6,0	108,0	114,0	
J2	M0	2	P04	O4	3,0	133,0	136,0	
J2	M4	2	P04	O5	6,0	136,0	142,0	
J3	M4	3	P01	O1	6,0	15,0	21,0	
J3	M2	3	P01	O2	6,0	44,0	50,0	
J3	M3	3	P01	O3	6,0	102,0	108,0	
J3	M0	3	P01	O4	6,0	111,0	117,0	
J3	M1	3	P01	O5	6,0	117,0	123,0	
J4	M1	4	P03	O1	6,0	28,0	34,0	
J4	M3	4	P03	O2	8,0	38,0	46,0	
J4	M4	4	P03	O3	8,0	51,0	59,0	
J4	M0	4	P03	O4	6,0	63,0	69,0	
J4	M2	4	P03	O5	8,0	108,0	116,0	
J5	M4	5	P01	O1	3,0	24,0	27,0	
J5	M2	5	P01	O2	3,0	68,0	71,0	
J5	M3	5	P01	O3	3,0	126,0	129,0	
J5	M0	5	P01	O4	3,0	130,0	133,0	
J5	M1	5	P01	O5	3,0	133,0	136,0	
J6	M1	6	P03	O1	3,0	47,0	50,0	
J6	M3	6	P03	O2	4,0	63,0	67,0	
J6	M4	6	P03	O3	4,0	82,0	86,0	
J6	M0	6	P03	O4	3,0	124,0	127,0	
J6	M2	6	P03	O5	4,0	128,0	132,0	
J7	M4	7	P01	O1	3,0	21,0	24,0	
J7	M2	7	P01	O2	3,0	65,0	68,0	
J7	M3	7	P01	O3	3,0	123,0	126,0	
J7	M0	7	P01	O4	3,0	127,0	130,0	
J7	M1	7	P01	O5	3,0	130,0	133,0	
J8	M1	8	P03	O1	3,0	44,0	47,0	
J8	M3	8	P03	O2	4,0	59,0	63,0	
J8	M4	8	P03	O3	4,0	78,0	82,0	
J8	M0	8	P03	O4	3,0	117,0	120,0	
J8	M2	8	P03	O5	4,0	124,0	128,0	

Nota: Dados Trabalhados pelo Autor

Tabela 4 - Solução *LPT* no formato original – parte 2

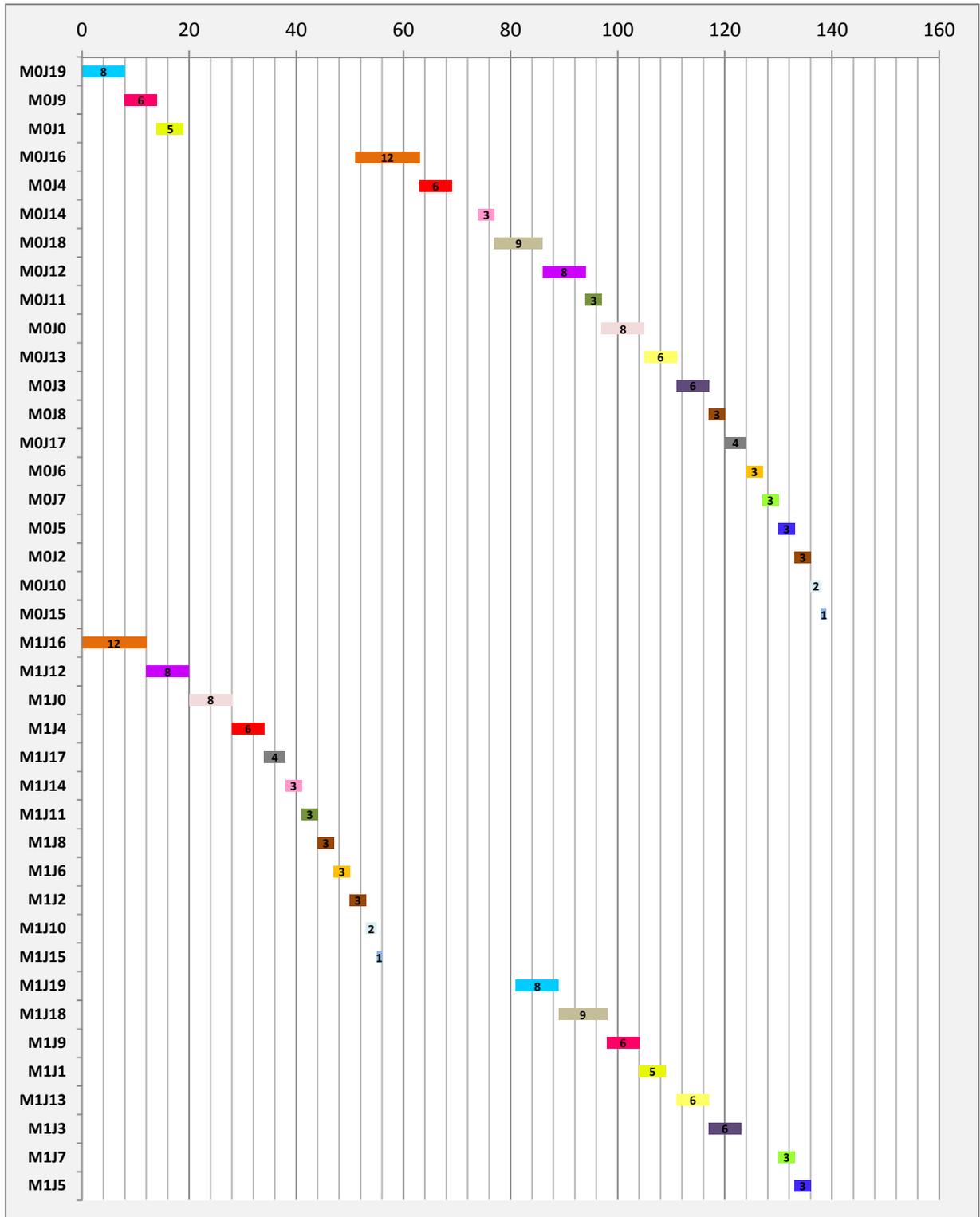
J9	M0	9	P02	O1	6,0	8,0	14,0	
J9	M3	9	P02	O2	6,0	32,0	38,0	
J9	M4	9	P02	O3	6,0	59,0	65,0	
J9	M2	9	P02	O4	6,0	81,0	87,0	
J9	M1	9	P02	O5	6,0	98,0	104,0	
J10	M1	10	P04	O1	2,0	53,0	55,0	
J10	M2	10	P04	O2	4,0	61,0	65,0	
J10	M3	10	P04	O3	4,0	119,0	123,0	
J10	M0	10	P04	O4	2,0	136,0	138,0	
J10	M4	10	P04	O5	4,0	142,0	146,0	
J11	M1	11	P03	O1	3,0	41,0	44,0	
J11	M2	11	P03	O2	4,0	55,0	59,0	
J11	M3	11	P03	O3	4,0	74,0	78,0	
J11	M0	11	P03	O4	3,0	94,0	97,0	
J11	M4	11	P03	O5	4,0	120,0	124,0	
J12	M1	12	P05	O1	8,0	12,0	20,0	
J12	M2	12	P05	O2	10,0	24,0	34,0	
J12	M3	12	P05	O3	10,0	76,0	86,0	
J12	M0	12	P05	O4	8,0	86,0	94,0	
J12	M4	12	P05	O5	10,0	94,0	104,0	
J13	M4	13	P01	O1	6,0	9,0	15,0	
J13	M2	13	P01	O2	6,0	18,0	24,0	
J13	M3	13	P01	O3	6,0	96,0	102,0	
J13	M0	13	P01	O4	6,0	105,0	111,0	
J13	M1	13	P01	O5	6,0	111,0	117,0	
J14	M1	14	P03	O1	3,0	38,0	41,0	
J14	M3	14	P03	O2	4,0	51,0	55,0	
J14	M4	14	P03	O3	4,0	70,0	74,0	
J14	M0	14	P03	O4	3,0	74,0	77,0	
J14	M2	14	P03	O5	4,0	116,0	120,0	
J15	M1	15	P04	O1	1,0	55,0	56,0	
J15	M2	15	P04	O2	2,0	71,0	73,0	
J15	M3	15	P04	O3	2,0	129,0	131,0	
J15	M0	15	P04	O4	1,0	138,0	139,0	
J15	M4	15	P04	O5	2,0	146,0	148,0	
J16	M1	16	P03	O1	12,0	0,0	12,0	
J16	M3	16	P03	O2	16,0	16,0	32,0	
J16	M4	16	P03	O3	16,0	35,0	51,0	
J16	M0	16	P03	O4	12,0	51,0	63,0	
J16	M2	16	P03	O5	16,0	92,0	108,0	
J17	M1	17	P05	O1	4,0	34,0	38,0	
J17	M2	17	P05	O2	5,0	50,0	55,0	
J17	M3	17	P05	O3	5,0	114,0	119,0	
J17	M0	17	P05	O4	4,0	120,0	124,0	
J17	M4	17	P05	O5	5,0	124,0	129,0	
J18	M4	18	P01	O1	9,0	0,0	9,0	
J18	M2	18	P01	O2	9,0	9,0	18,0	
J18	M3	18	P01	O3	9,0	67,0	76,0	
J18	M0	18	P01	O4	9,0	77,0	86,0	
J18	M1	18	P01	O5	9,0	89,0	98,0	
J19	M0	19	P02	O1	8,0	0,0	8,0	
J19	M3	19	P02	O2	8,0	8,0	16,0	
J19	M4	19	P02	O3	8,0	27,0	35,0	
J19	M2	19	P02	O4	8,0	73,0	81,0	
J19	M1	19	P02	O5	8,0	81,0	89,0	

Nota: Dados Trabalhados pelo Autor

Tabela 5 - Solução *LPT* – sequenciamento de cada máquina – *job* – tempo início – tempo operação

MaqJob	Tinicio	Tciclo		MaqJob	Tinicio	Tciclo		MaqJob	Tinicio	Tciclo
M0J19	0	8		M1J1	104	5		M3J0	86	10
M0J9	8	6		M1J13	111	6		M3J13	96	6
M0J1	14	5		M1J3	117	6		M3J3	102	6
M0J16	51	12		M1J7	130	3		M3J2	108	6
M0J4	63	6		M1J5	133	3		M3J17	114	5
M0J14	74	3		M2J18	9	9		M3J10	119	4
M0J18	77	9		M2J13	18	6		M3J7	123	3
M0J12	86	8		M2J12	24	10		M3J5	126	3
M0J11	94	3		M2J0	34	10		M3J15	129	2
M0J0	97	8		M2J3	44	6		M4J18	0	9
M0J13	105	6		M2J17	50	5		M4J13	9	6
M0J3	111	6		M2J2	55	6		M4J3	15	6
M0J8	117	3		M2J10	61	4		M4J7	21	3
M0J17	120	4		M2J7	65	3		M4J5	24	3
M0J6	124	3		M2J5	68	3		M4J19	27	8
M0J7	127	3		M2J15	71	2		M4J16	35	16
M0J5	130	3		M2J19	73	8		M4J4	51	8
M0J2	133	3		M2J9	81	6		M4J9	59	6
M0J10	136	2		M2J1	87	5		M4J1	65	5
M0J15	138	1		M2J16	92	16		M4J14	70	4
M1J16	0	12		M2J4	108	8		M4J11	74	4
M1J12	12	8		M2J14	116	4		M4J8	78	4
M1J0	20	8		M2J11	120	4		M4J6	82	4
M1J4	28	6		M2J8	124	4		M4J12	94	10
M1J17	34	4		M2J6	128	4		M4J0	105	10
M1J14	38	3		M3J19	8	8		M4J17	124	5
M1J11	41	3		M3J16	16	16		M4J2	136	6
M1J8	44	3		M3J9	32	6		M4J10	142	4
M1J6	47	3		M3J4	38	8		M4J15	146	2
M1J2	50	3		M3J1	46	5				
M1J10	53	2		M3J14	51	4				
M1J15	55	1		M3J11	55	4				
M1J19	81	8		M3J8	59	4				
M1J18	89	9		M3J6	63	4				
M1J9	98	6		M3J18	67	9				
M1J1	104	5		M3J12	76	10				

Nota: Dados Trabalhados pelo Autor

Gráfico 1- Gráfico de Gantt por máquina da regra *LPT* – parte 1

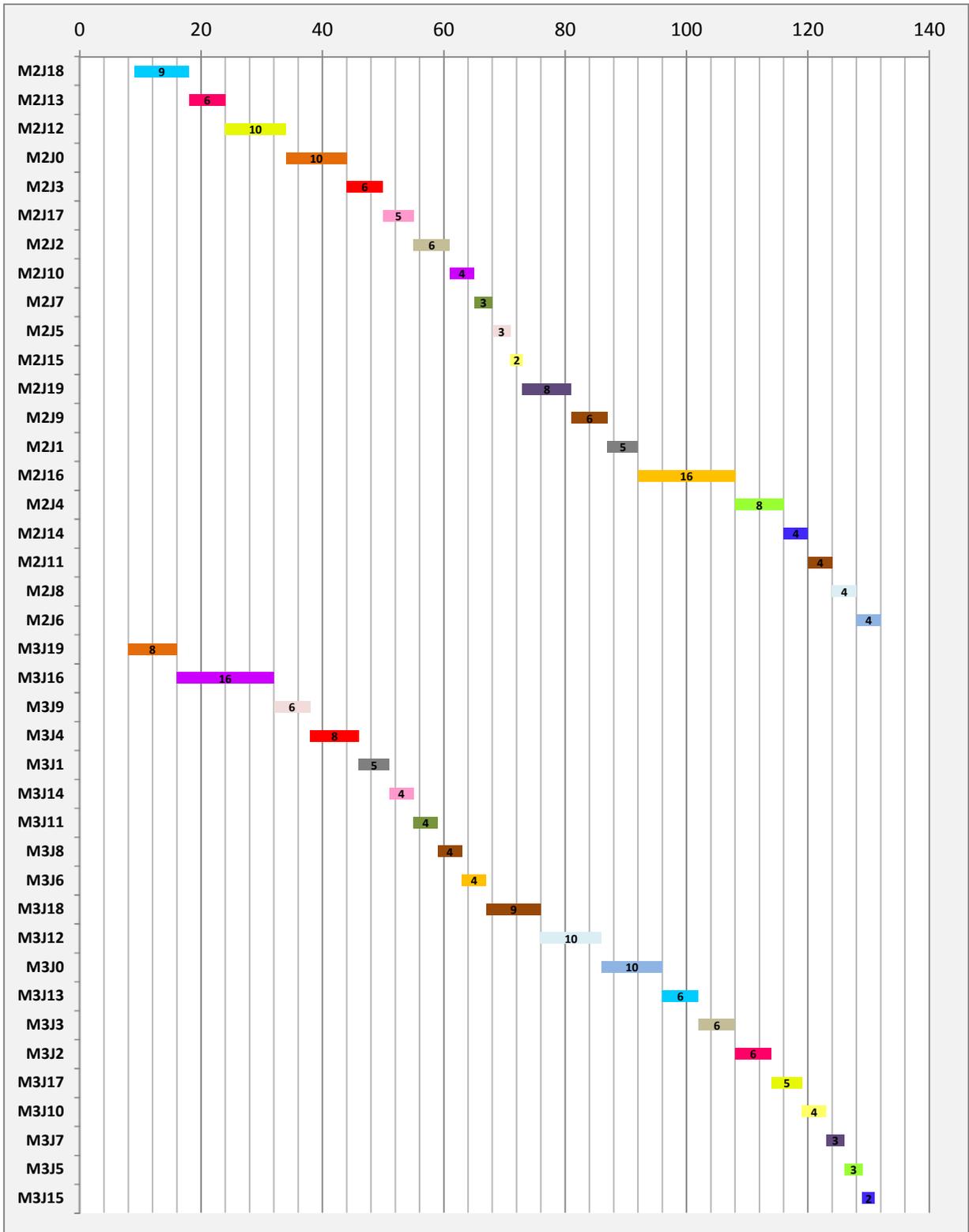


Gráfico 2 - Gráfico de Gantt por Máquina da regra LPT – parte 2

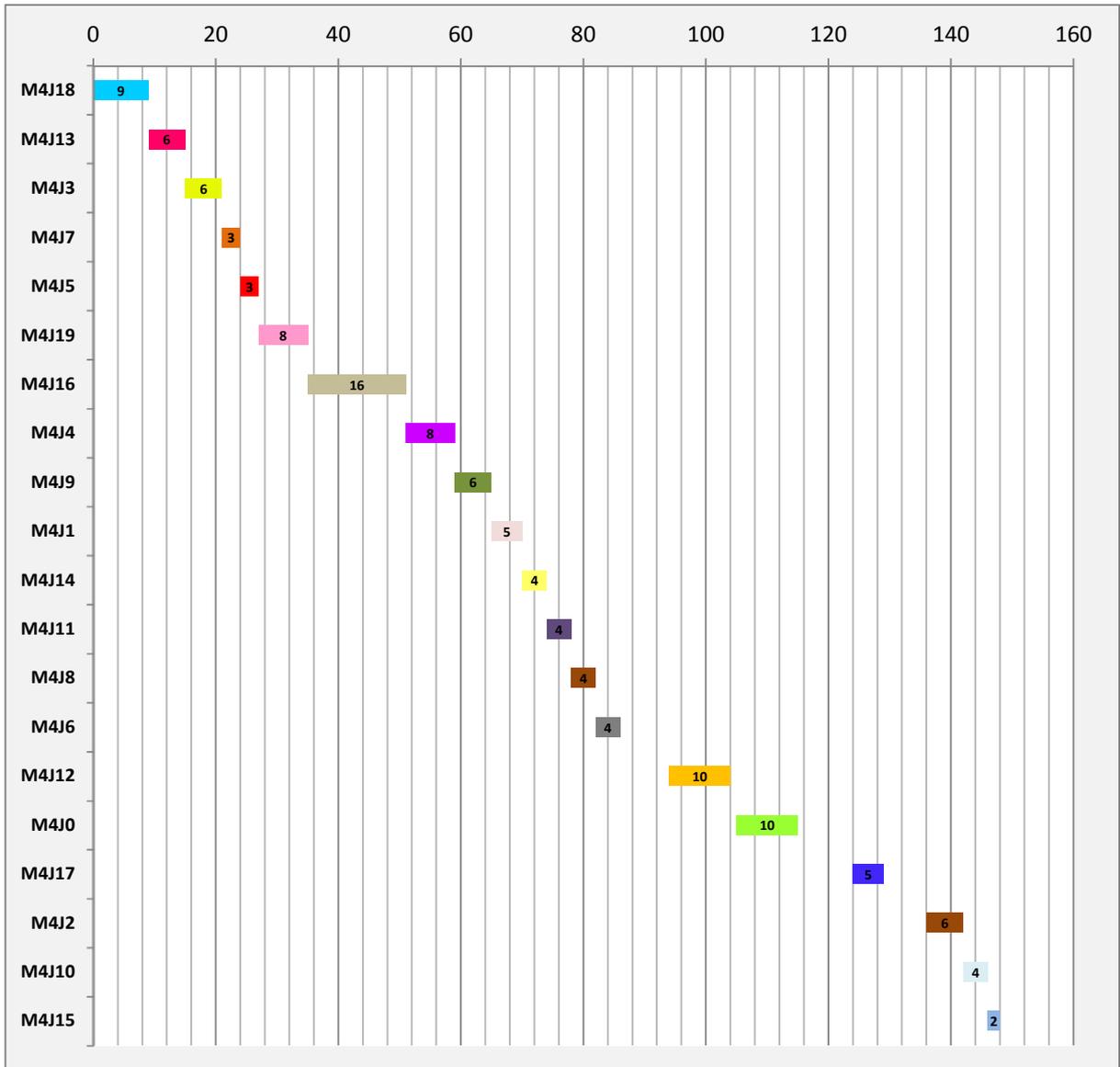


Gráfico 3 - Gráfico de Gantt por Máquina da regra *LPT* – parte 3

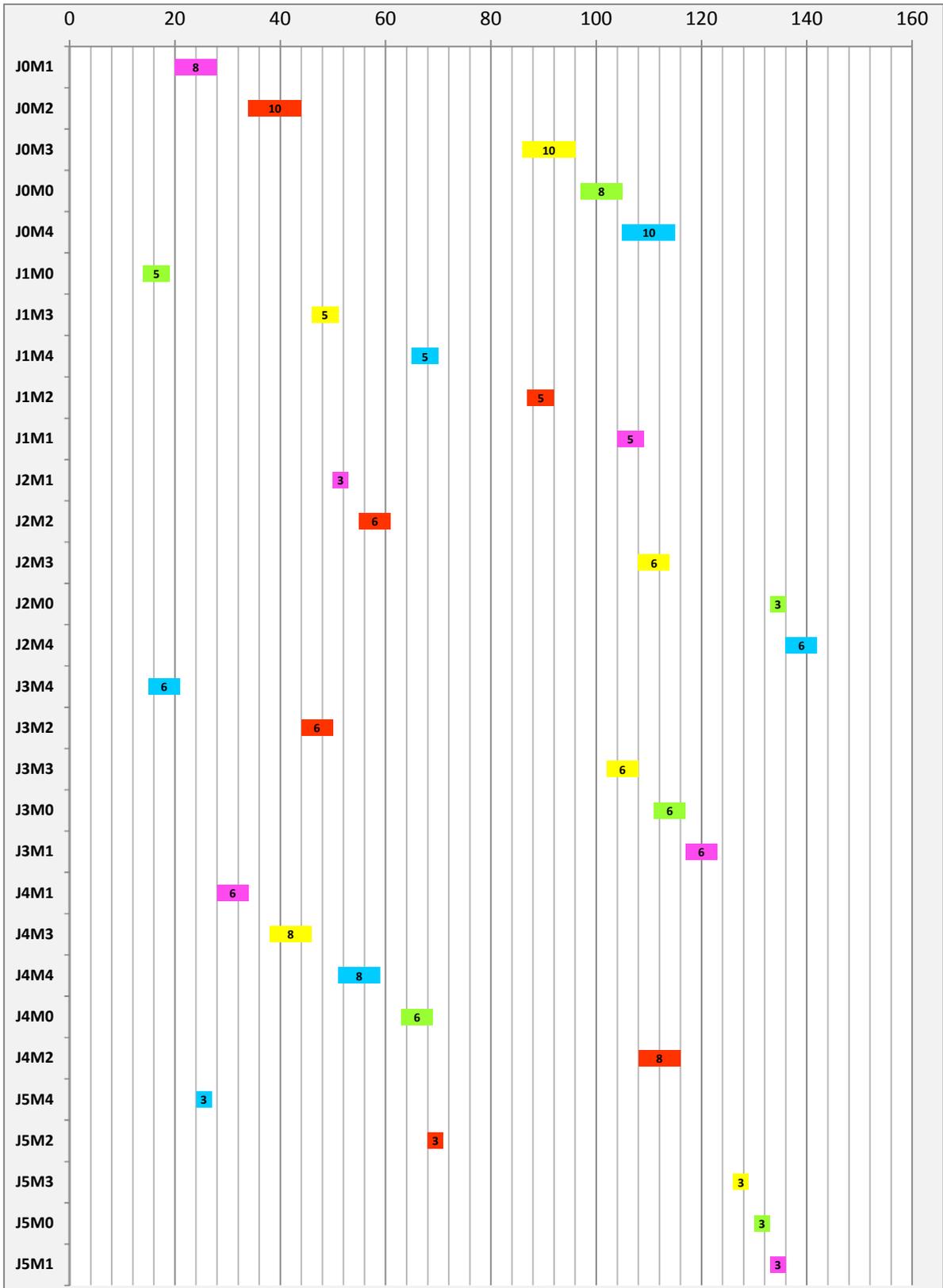


Gráfico 4 - Gráfico de Gantt por Job da regra LPT – parte 1

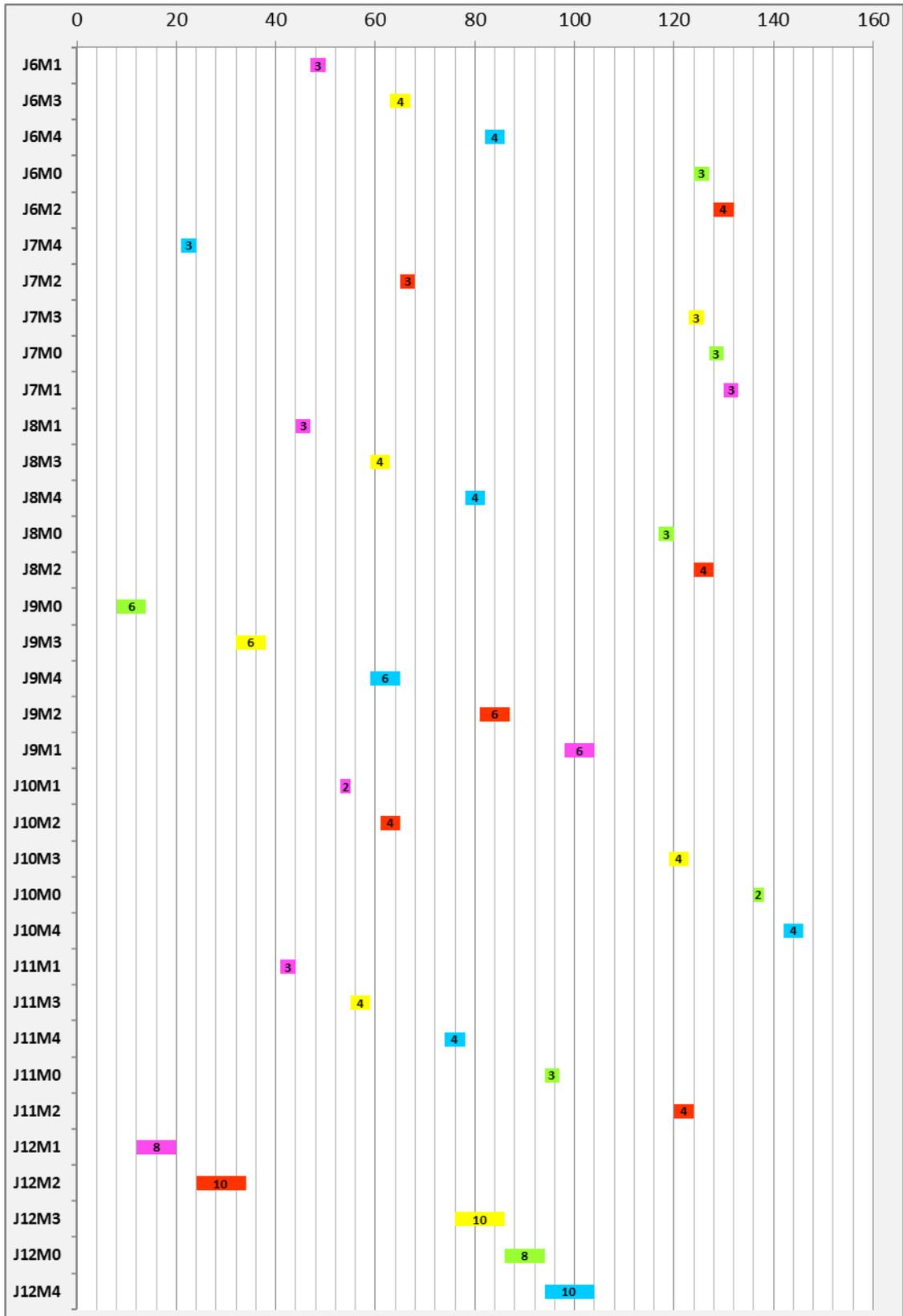


Gráfico 5 - Gráfico de Gantt por Job da regra LPT – parte 2

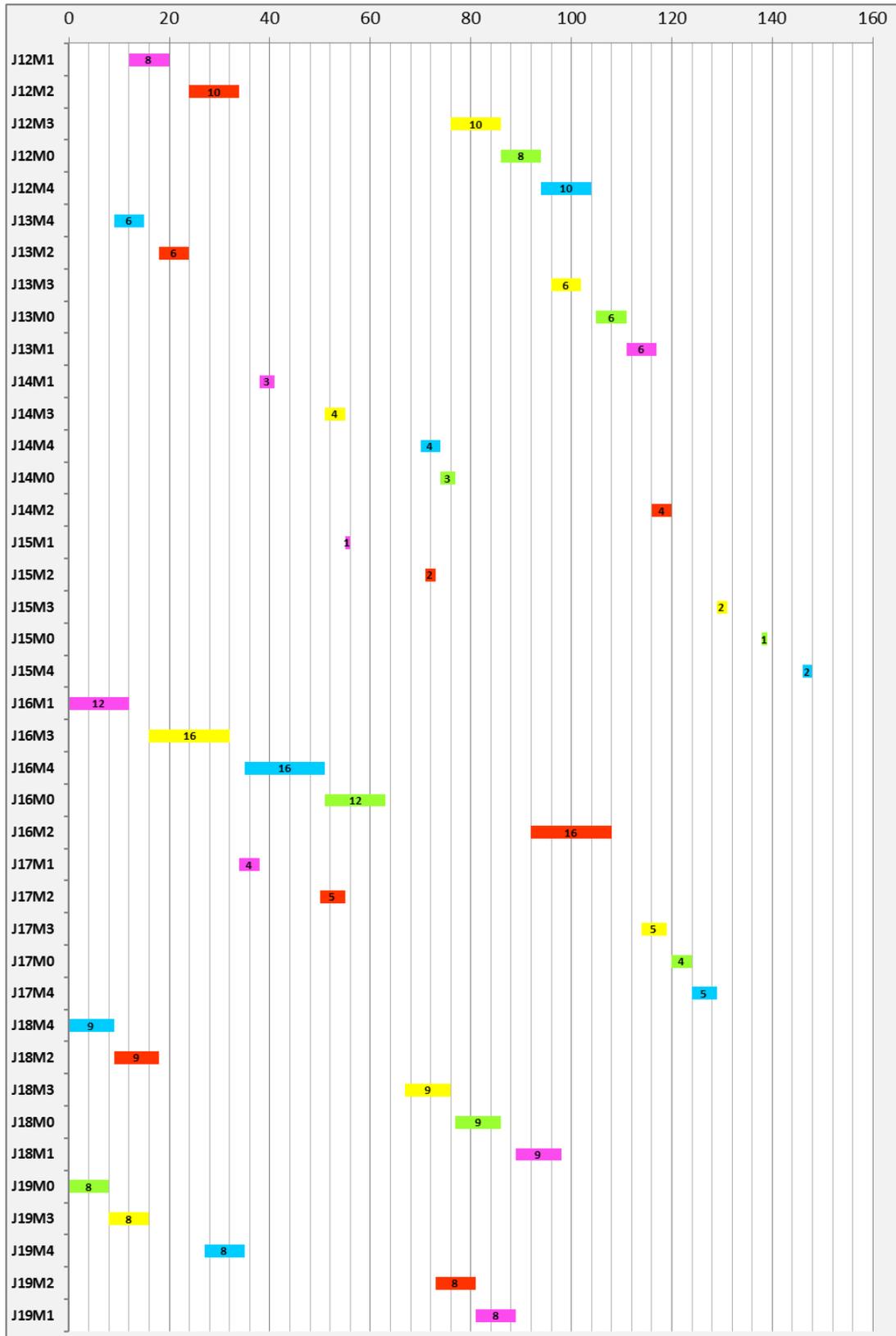


Gráfico 6 - Gráfico de Gantt por Job da regra LPT – parte 3

### 3.3.2 Solução com a Regra *MWKR*

A tabela 6 sintetiza o resultado final do sequenciamento, cujo *makespan* encontrado foi de  $C_{max} = 138$  h.

Nos gráficos de Gantt 7 a 9 podem ser visualizados os sequenciamentos das máquinas, suas ocupações e ociosidades.

Nos gráficos de Gantt de 10 a 12 podem ser visualizados os sequenciamentos dos jobs e o seu tempo de permanência em operação.

Tabela 6 - Solução LPT – sequenciamento de cada máquina – job – tempo início – tempo operação

JobMaq	Tinicio	Tciclo	JobMaq	Tinicio	Tciclo	JobMaq	Tinicio	Tciclo
J20M0	0	8	J4M1	98	6	J15M3	92	4
J10M0	8	6	J10M1	104	6	J12M3	96	4
J2M0	14	5	J2M1	114	5	J9M3	100	4
J19M0	41	9	J6M1	132	3	J7M3	104	4
J17M0	51	12	J8M1	135	3	J3M3	108	6
J13M0	63	8	J19M2	9	9	J18M3	114	5
J1M0	71	8	J14M2	18	6	J11M3	119	4
J5M0	79	6	J13M2	24	10	J8M3	123	3
J14M0	86	6	J1M2	34	10	J6M3	126	3
J4M0	92	6	J4M2	44	6	J16M3	129	2
J15M0	106	3	J3M2	50	6	J19M4	0	9
J12M0	110	3	J18M2	56	5	J14M4	9	6
J3M0	114	3	J20M2	61	8	J4M4	15	6
J9M0	117	3	J17M2	69	16	J20M4	21	8
J18M0	120	4	J11M2	85	4	J8M4	29	3
J11M0	124	2	J8M2	89	3	J6M4	32	3
J7M0	126	3	J6M2	92	3	J17M4	35	16
J6M0	129	3	J10M2	95	6	J5M4	51	8
J8M0	132	3	J5M2	101	8	J13M4	71	10
J16M0	135	1	J2M2	109	5	J10M4	81	6
J17M1	0	12	J16M2	114	2	J1M4	87	10
J13M1	12	8	J15M2	116	4	J2M4	97	5
J1M1	20	8	J12M2	120	4	J15M4	102	4
J5M1	28	6	J7M2	129	4	J12M4	106	4
J3M1	34	3	J9M2	133	4	J9M4	110	4
J18M1	37	4	J20M3	8	8	J7M4	114	4
J15M1	41	3	J17M3	16	16	J3M4	118	6
J12M1	44	3	J19M3	32	9	J18M4	124	5
J9M1	47	3	J5M3	41	8	J11M4	129	4
J7M1	50	3	J13M3	49	10	J16M4	136	2
J11M1	53	2	J1M3	59	10			
J19M1	55	9	J10M3	69	6			
J16M1	64	1	J2M3	75	5			
J20M1	69	8	J14M3	80	6			
J14M1	92	6	J4M3	86	6			

Nota: Dados Trabalhados pelo Autor

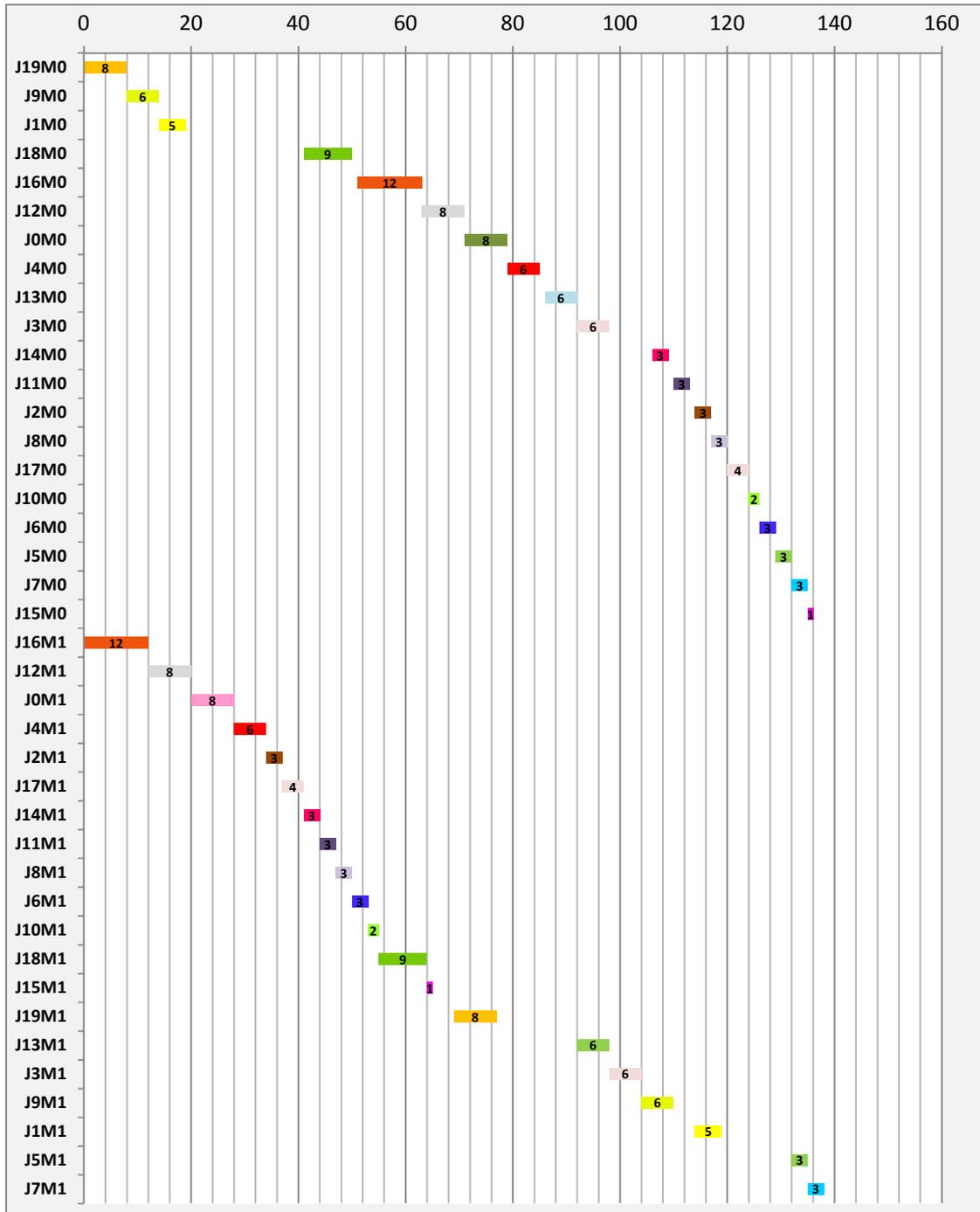
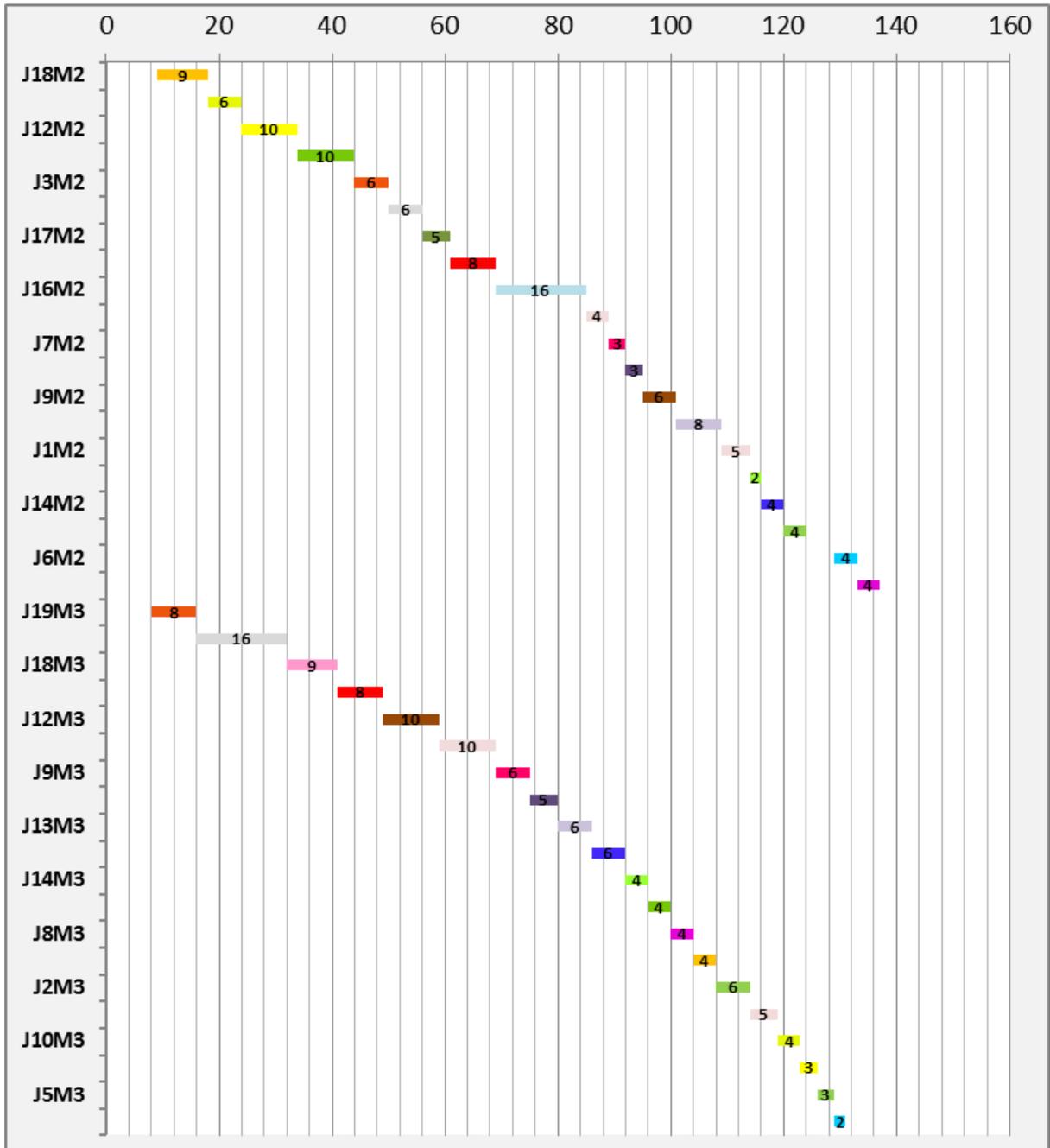


Gráfico 7 - Gráfico de Gantt por Máquina da Regra MWKR- parte 1

Gráfico 8 - Gráfico de Gantt por Máquina da Regra *MWKR* – parte 2

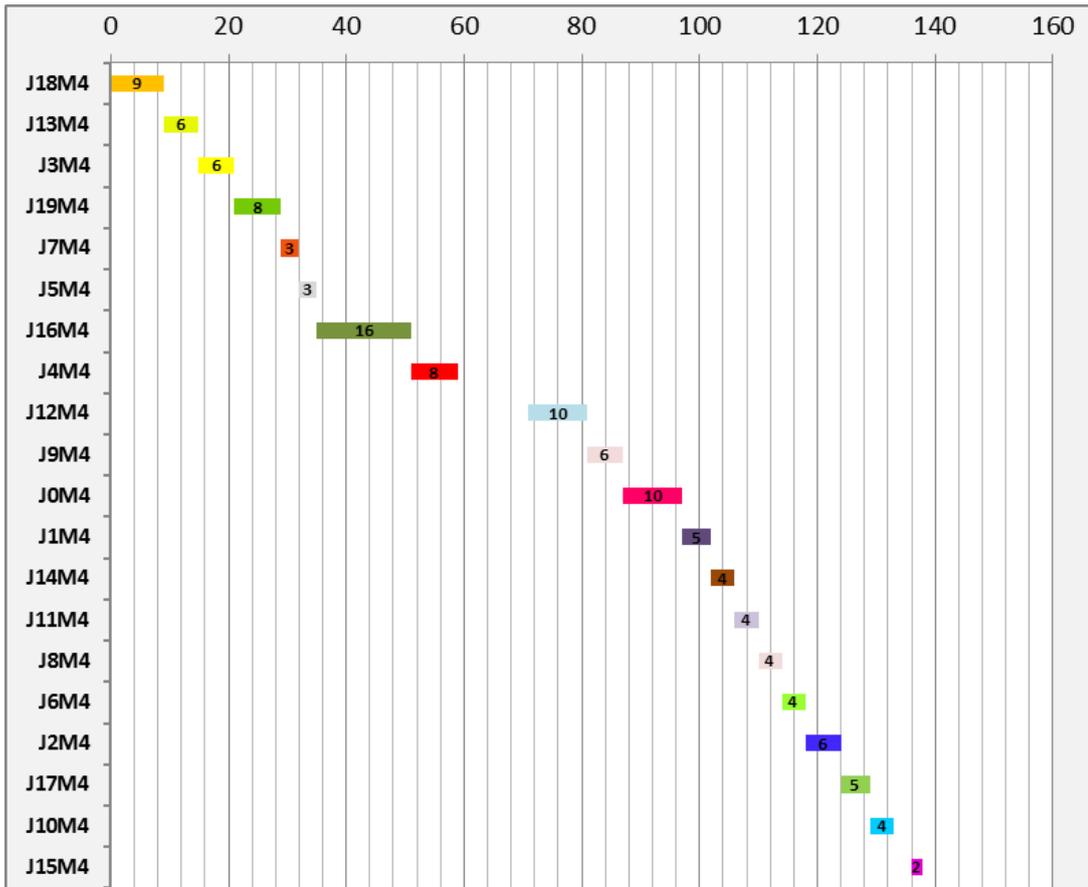
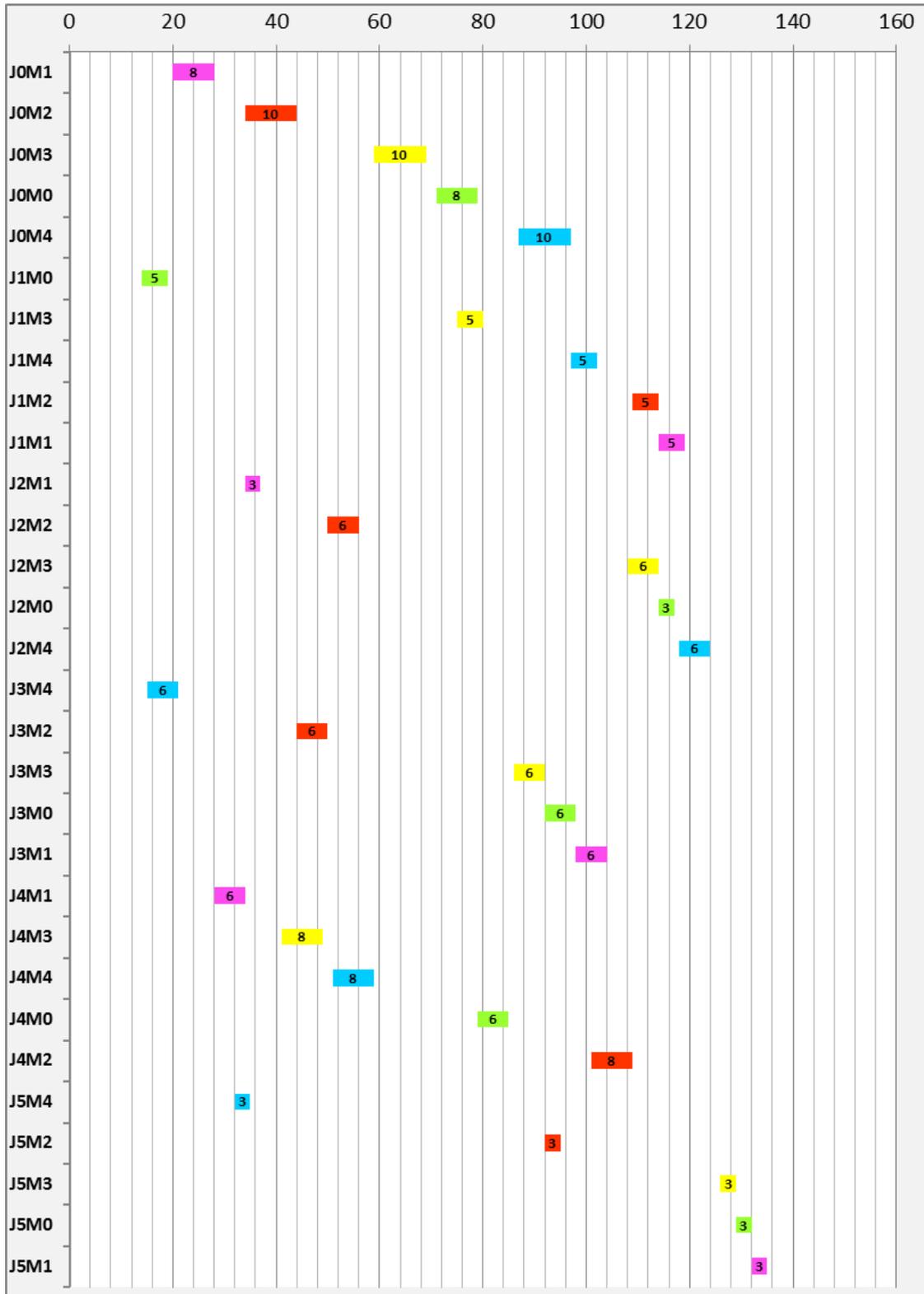


Gráfico 9 - Gráfico de Gantt por Máquina da Regra *MWKR* – parte 3

Gráfico 10 - Gráfico de Gantt por *Job* da Regra *MWKR* – parte 1

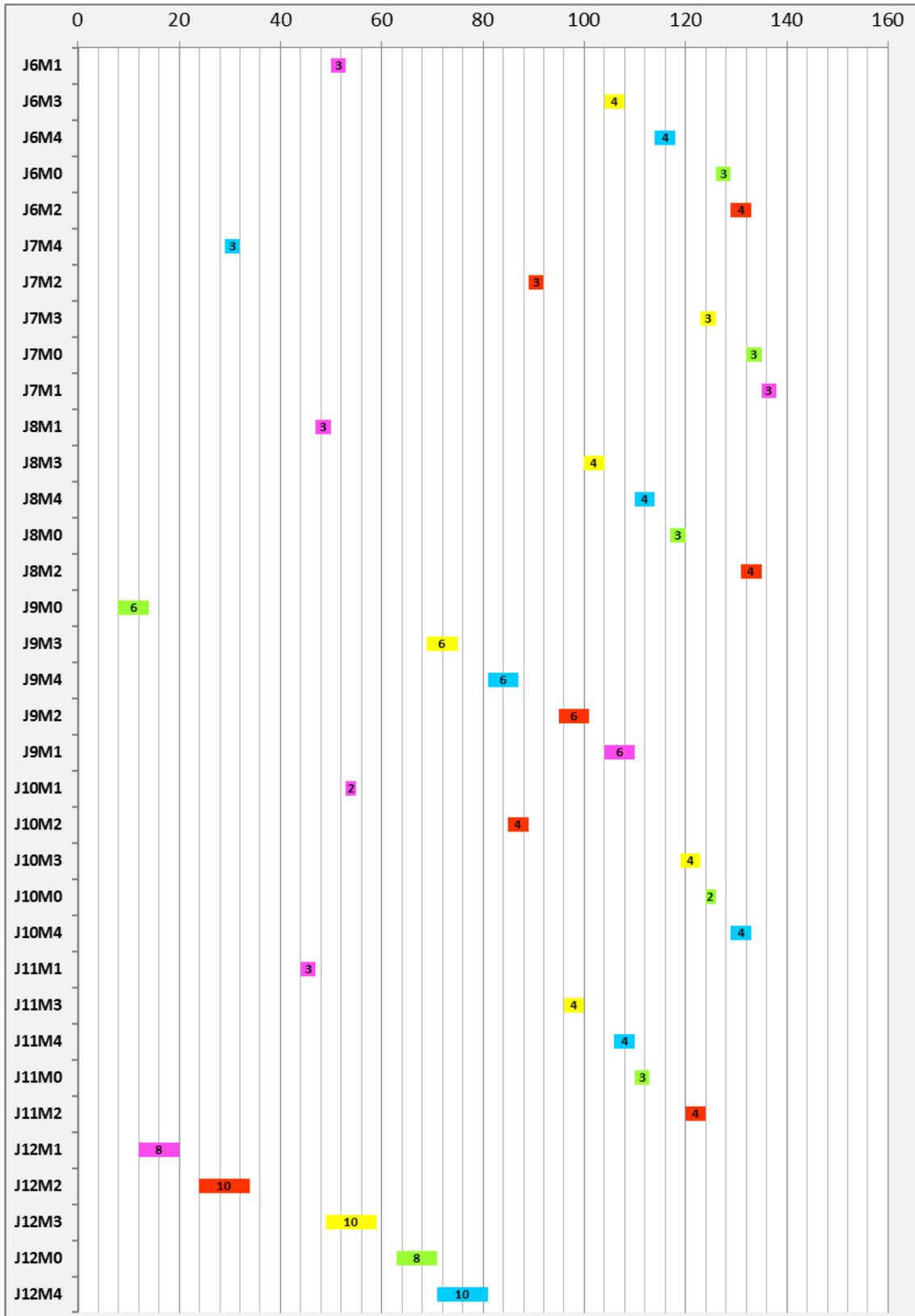


Gráfico 11 - Gráfico de Gantt por Job da Regra MWKR – parte 2

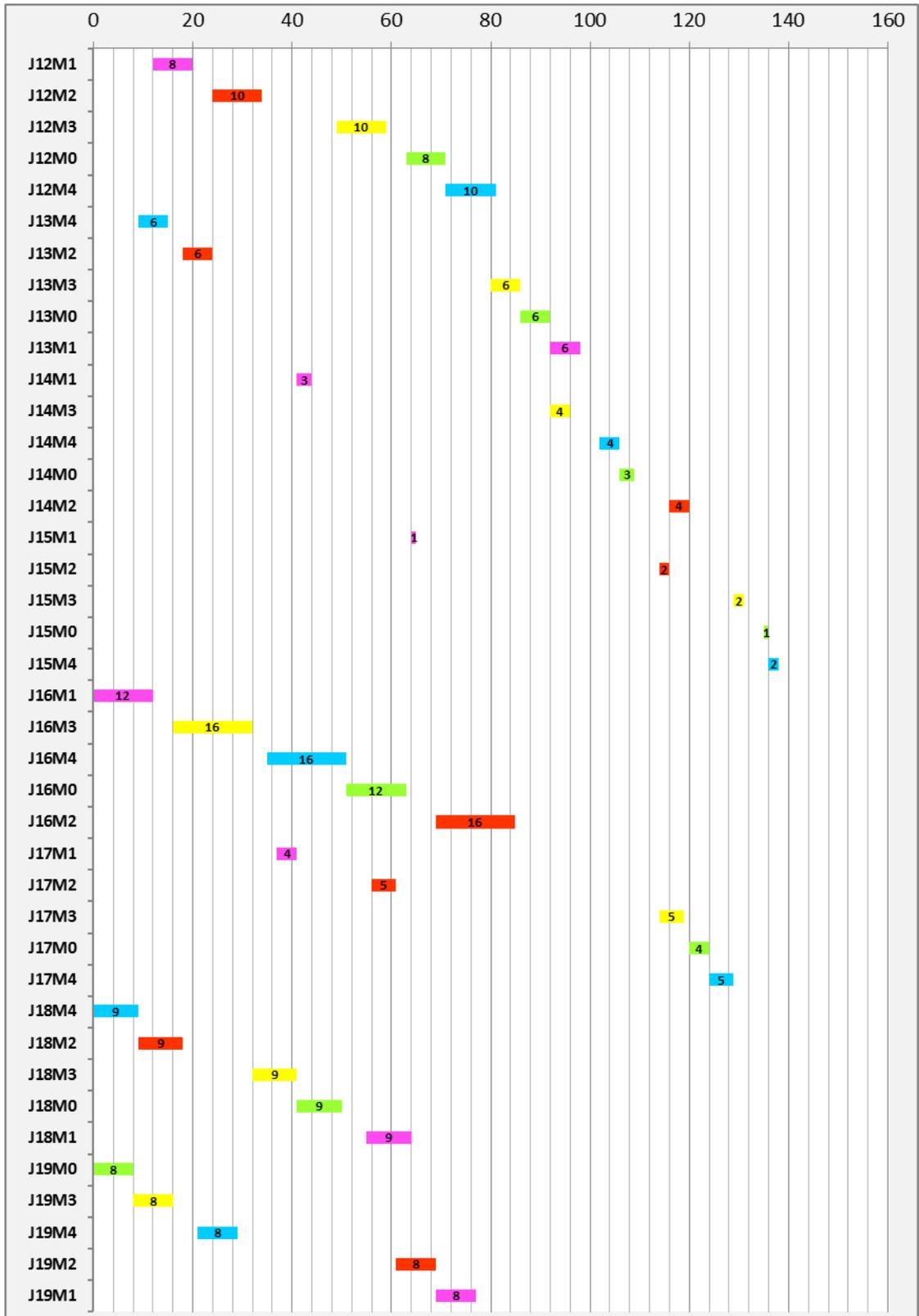


Gráfico 12 - Gráfico de Gantt por Job da Regra MWKR – parte 3

### 3.3.3 Solução pela Programação de Restrições

A tabela 7 mostra a solução ótima encontrada através do software CPLEX 12.3, no seu formato original com um  $C_{max} = 133h$  e a mesma informação sintetizada na tabela 8.

Os gráficos de Gantt 13 a 15 mostram o sequenciamento por máquina e os gráficos 16 a 18 mostram o sequenciamento por *job*.

Tabela 7 - Solução Programação de Restrições – formato original

Mchs (size 5)	name	position	declaredPosition	type	start	end
0	itvs#9#0	0	9	0	0	6
	itvs#19#0	1	19	1	6	14
	itvs#1#0	2	1	2	14	19
	itvs#16#3	3	16	3	46	58
	itvs#12#3	4	12	4	58	66
	itvs#0#3	5	0	5	66	74
	itvs#18#3	6	18	6	74	83
	itvs#4#3	7	4	7	83	89
	itvs#3#3	8	3	8	89	95
	itvs#13#3	9	13	9	95	101
	itvs#14#3	10	14	10	101	104
	itvs#11#3	11	11	11	104	107
	itvs#17#3	12	17	12	107	111
	itvs#8#3	13	8	13	111	114
	itvs#6#3	14	6	14	115	118
	itvs#2#3	15	2	15	118	121
	itvs#10#3	16	10	16	121	123
	itvs#5#3	17	5	17	124	127
	itvs#7#3	18	7	18	127	130
	itvs#15#3	19	15	19	130	131
1	itvs#16#0	0	16	0	0	12
	itvs#12#0	1	12	1	12	20
	itvs#0#0	2	0	2	20	28
	itvs#4#0	3	4	3	28	34
	itvs#2#0	4	2	4	34	37
	itvs#17#0	5	17	5	37	41
	itvs#10#0	6	10	6	41	43
	itvs#11#0	7	11	7	43	46
	itvs#14#0	8	14	8	46	49
	itvs#8#0	9	8	9	49	52

	itvs#6#0	10	6	10	52	55
	itvs#15#0	11	15	11	55	56
	itvs#19#4	12	19	12	79	87
	itvs#18#4	13	18	13	87	96
	itvs#9#4	14	9	14	96	102
	itvs#13#4	15	13	15	102	108
	itvs#3#4	16	3	16	108	114
	itvs#1#4	17	1	17	114	119
	itvs#5#4	18	5	18	127	130
	itvs#7#4	19	7	19	130	133
2	itvs#18#1	0	18	0	9	18
	itvs#3#1	1	3	1	18	24
	itvs#12#1	2	12	2	24	34
	itvs#0#1	3	0	3	34	44
	itvs#13#1	4	13	4	44	50
	itvs#2#1	5	2	5	50	56
	itvs#17#1	6	17	6	56	61
	itvs#10#1	7	10	7	61	65
	itvs#5#1	8	5	8	65	68
	itvs#7#1	9	7	9	68	71
	itvs#19#3	10	19	10	71	79
	itvs#9#3	11	9	11	79	85
	itvs#15#1	12	15	12	85	87
	itvs#16#4	13	16	13	87	103
	itvs#1#3	14	1	14	103	108
	itvs#14#4	15	14	15	108	112
	itvs#4#4	16	4	16	112	120
	itvs#8#4	17	8	17	120	124
	itvs#6#4	18	6	18	124	128
	itvs#11#4	19	11	19	128	132
3	itvs#9#1	0	9	0	6	12
	itvs#16#1	1	16	1	12	28
	itvs#19#1	2	19	2	28	36
	itvs#4#1	3	4	3	36	44
	itvs#12#2	4	12	4	44	54
	itvs#0#2	5	0	5	54	64
	itvs#18#2	6	18	6	64	73
	itvs#1#1	7	1	7	73	78
	itvs#3#2	8	3	8	78	84
	itvs#13#2	9	13	9	84	90
	itvs#14#1	10	14	10	90	94
	itvs#11#1	11	11	11	94	98
	itvs#8#1	12	8	12	98	102

	itvs#17#2	13	17	13	102	107
	itvs#6#1	14	6	14	107	111
	itvs#2#2	15	2	15	111	117
	itvs#10#2	16	10	16	117	121
	itvs#5#2	17	5	17	121	124
	itvs#7#2	18	7	18	124	127
	itvs#15#2	19	15	19	127	129
4	itvs#18#0	0	18	0	0	9
	itvs#3#0	1	3	1	9	15
	itvs#13#0	2	13	2	15	21
	itvs#5#0	3	5	3	21	24
	itvs#9#2	4	9	4	24	30
	itvs#16#2	5	16	5	30	46
	itvs#19#2	6	19	6	46	54
	itvs#4#2	7	4	7	54	62
	itvs#7#0	8	7	8	62	65
	itvs#12#4	9	12	9	66	76
	itvs#0#4	10	0	10	76	86
	itvs#1#2	11	1	11	86	91
	itvs#14#2	12	14	12	94	98
	itvs#11#2	13	11	13	98	102
	itvs#8#2	14	8	14	102	106
	itvs#6#2	15	6	15	111	115
	itvs#17#4	16	17	16	115	120
	itvs#2#4	17	2	17	121	127
	itvs#10#4	18	10	18	127	131
	itvs#15#4	19	15	19	131	133

Nota: Dados Trabalhados pelo Autor

Tabela 8 - Solução Programação de Restrições – *MaqJob* e tempos

<b>MaqJob</b>	<b>Tinicio</b>	<b>Tciclo</b>	<b>MaqJob</b>	<b>Tinicio</b>	<b>Tciclo</b>	<b>MaqJob</b>	<b>Tinicio</b>	<b>Tciclo</b>
M0J9	0	6	M1J13	102	6	M3J13	84	6
M0J19	6	8	M1J3	108	6	M3J14	90	4
M0J1	14	5	M1J1	114	5	M3J11	94	4
M0J16	46	12	M1J5	127	3	M3J8	98	4
M0J12	58	8	M1J7	130	3	M3J17	102	5
M0J0	66	8	M2J18	9	9	M3J6	107	4
M0J18	74	9	M2J3	18	6	M3J2	111	6
M0J4	83	6	M2J12	24	10	M3J10	117	4
M0J3	89	6	M2J0	34	10	M3J5	121	3
M0J13	95	6	M2J13	44	6	M3J7	124	3
M0J14	101	3	M2J2	50	6	M3J15	127	2
M0J11	104	3	M2J17	56	5	M4J18	0	9
M0J17	107	4	M2J10	61	4	M4J3	9	6
M0J8	111	3	M2J5	65	3	M4J13	15	6
M0J6	115	3	M2J7	68	3	M4J5	21	3
M0J2	118	3	M2J19	71	8	M4J9	24	6
M0J10	121	2	M2J9	79	6	M4J16	30	16
M0J5	124	3	M2J15	85	2	M4J19	46	8
M0J7	127	3	M2J16	87	16	M4J4	54	8
M0J15	130	1	M2J1	103	5	M4J7	62	3
M1J16	0	12	M2J14	108	4	M4J12	66	10
M1J12	12	8	M2J4	112	8	M4J0	76	10
M1J0	20	8	M2J8	120	4	M4J1	86	5
M1J4	28	6	M2J6	124	4	M4J14	94	4
M1J2	34	3	M2J11	128	4	M4J11	98	4
M1J17	37	4	M3J9	6	6	M4J8	102	4
M1J10	41	2	M3J16	12	16	M4J6	111	4
M1J11	43	3	M3J19	28	8	M4J17	115	5
M1J14	46	3	M3J4	36	8	M4J2	121	6
M1J8	49	3	M3J12	44	10	M4J10	127	4
M1J6	52	3	M3J0	54	10	M4J15	131	2
M1J15	55	1	M3J18	64	9			
M1J19	79	8	M3J1	73	5			
M1J18	87	9	M3J3	78	6			

Nota: Dados Trabalhados pelo Autor

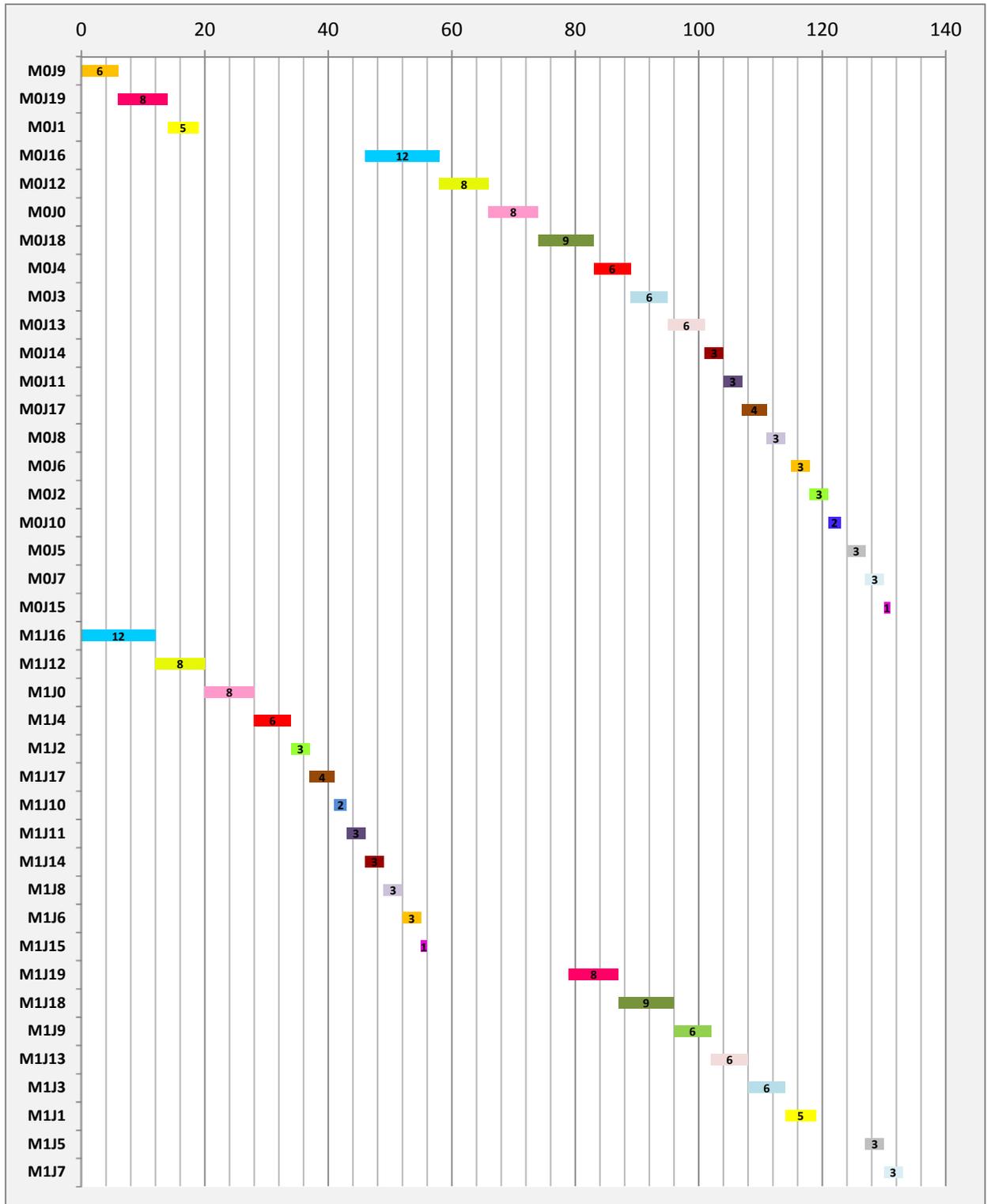


Gráfico 13 - Gráfico de Gantt Solução por Máquina - Programação de Restrições- parte 1

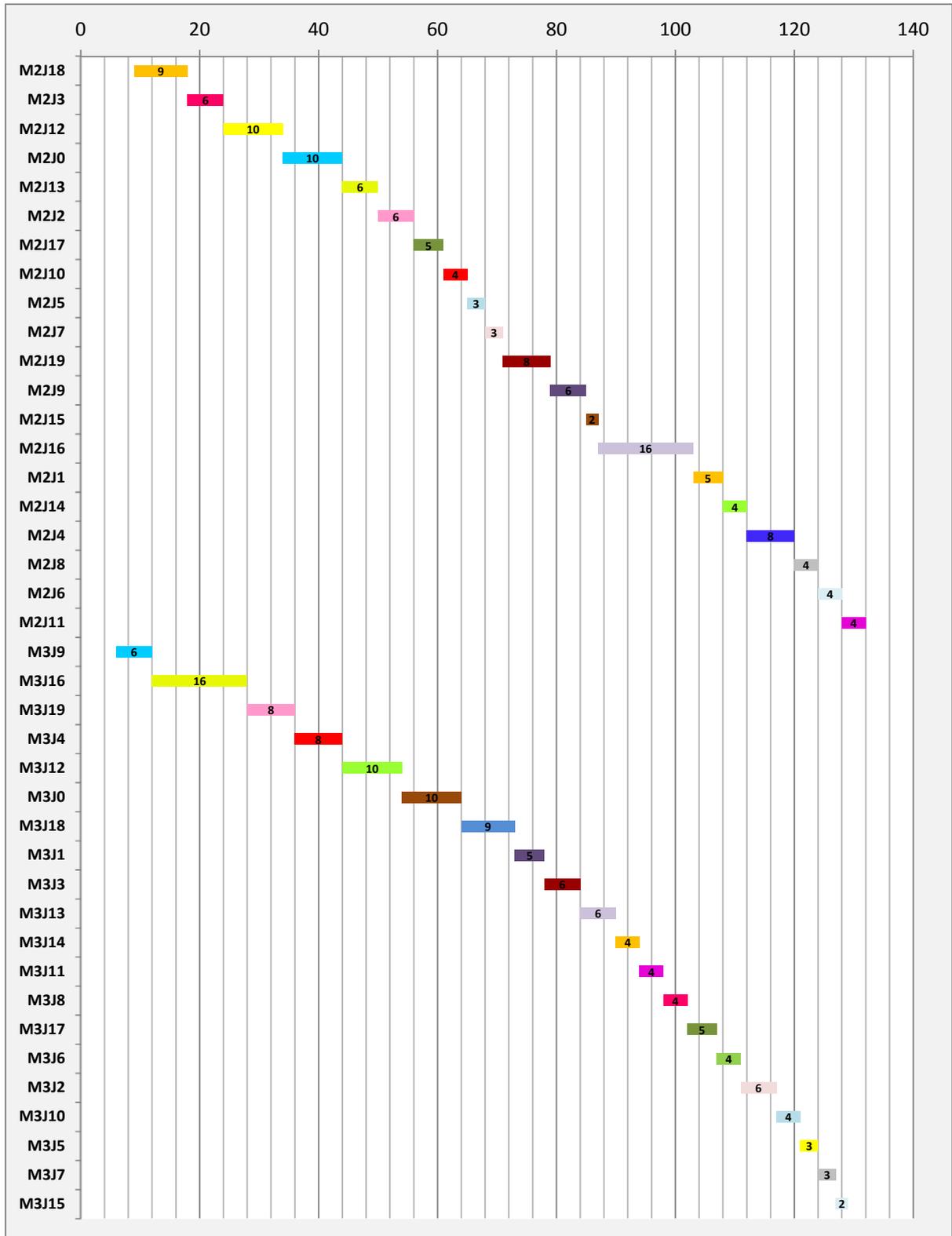


Gráfico 14 - Gráfico de Gantt Solução por Máquina - Programação de Restrições- parte 2

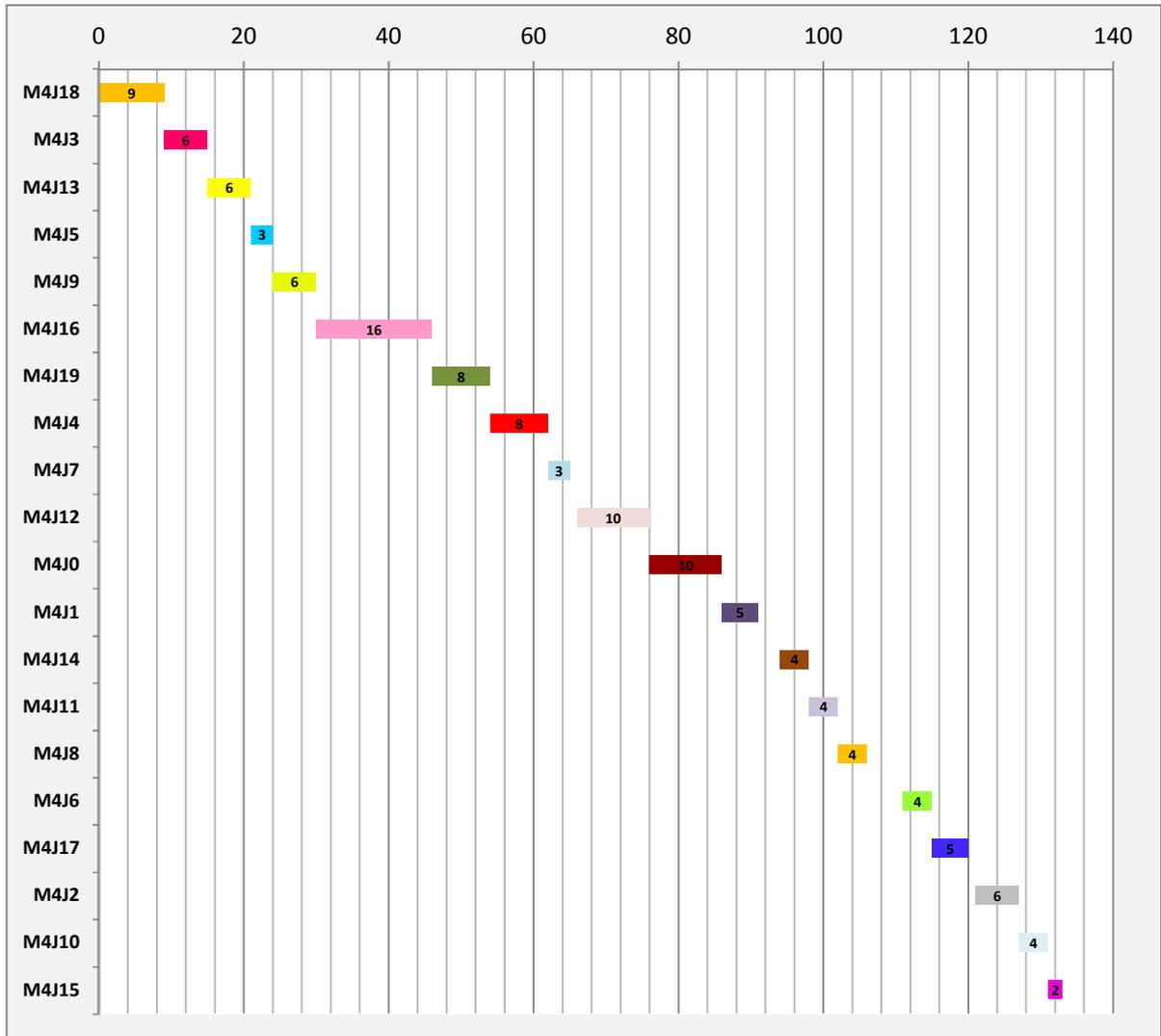


Gráfico 15 - Gráfico de Gantt Solução por Máquina - Programação de Restrições– parte 3

A seguir a mesma solução é apresentada através dos gráficos de Gantt sequenciados por *Job*.

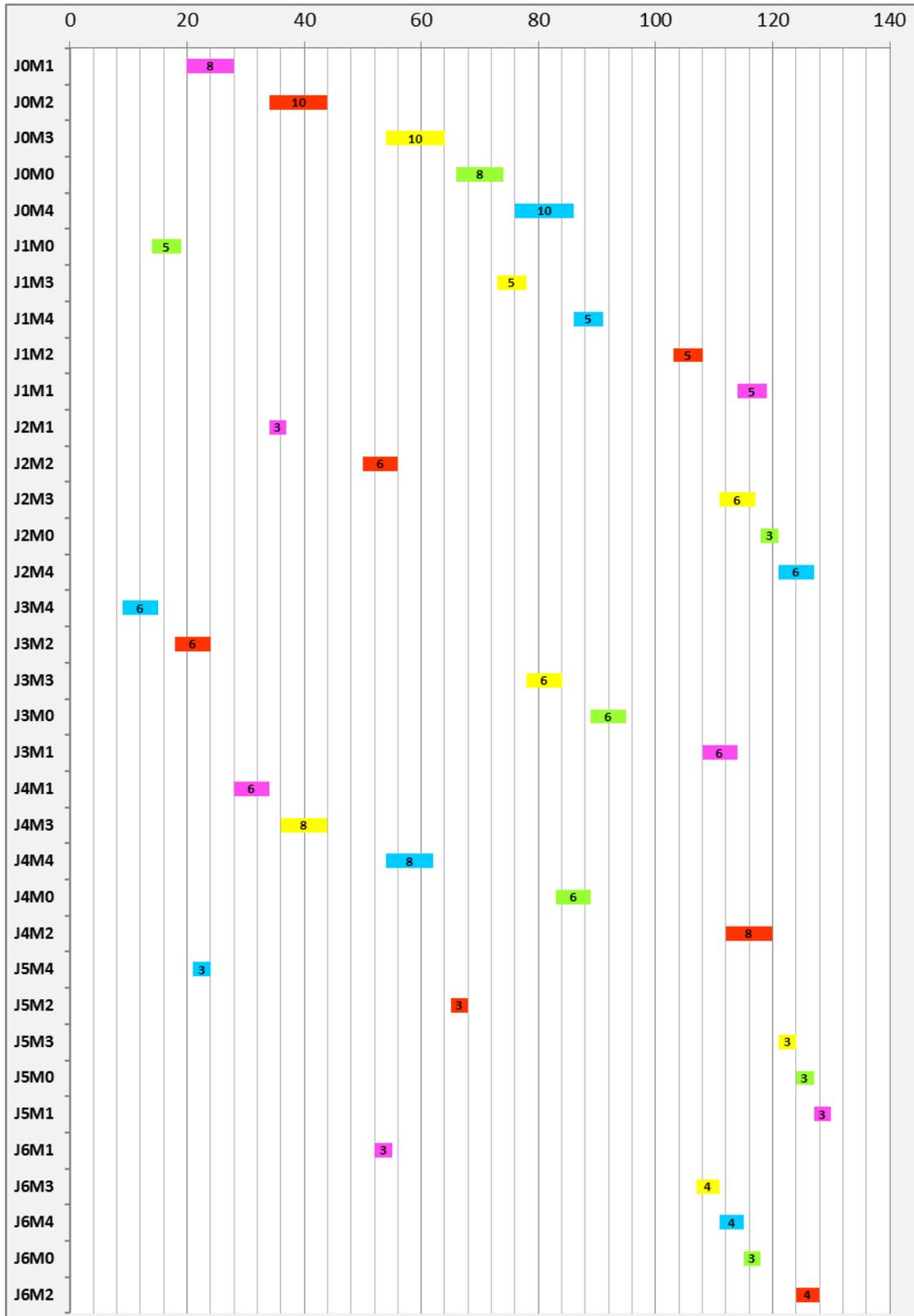


Gráfico 16 - Gráfico de Gantt Solução por Job - Programação de Restrições- parte 1

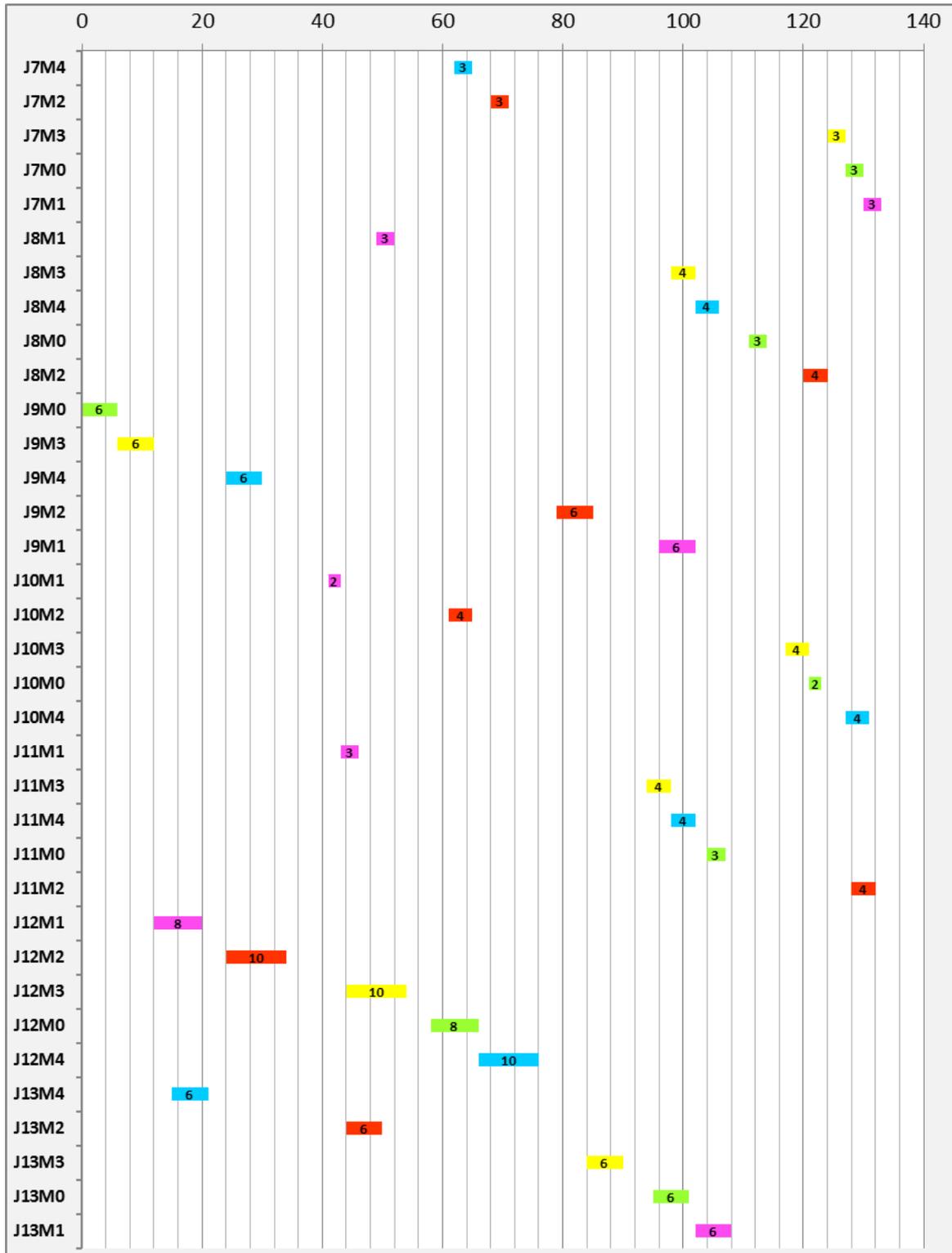


Gráfico 17 - Gráfico de Gantt Solução por Job - Programação de Restrições- parte 2

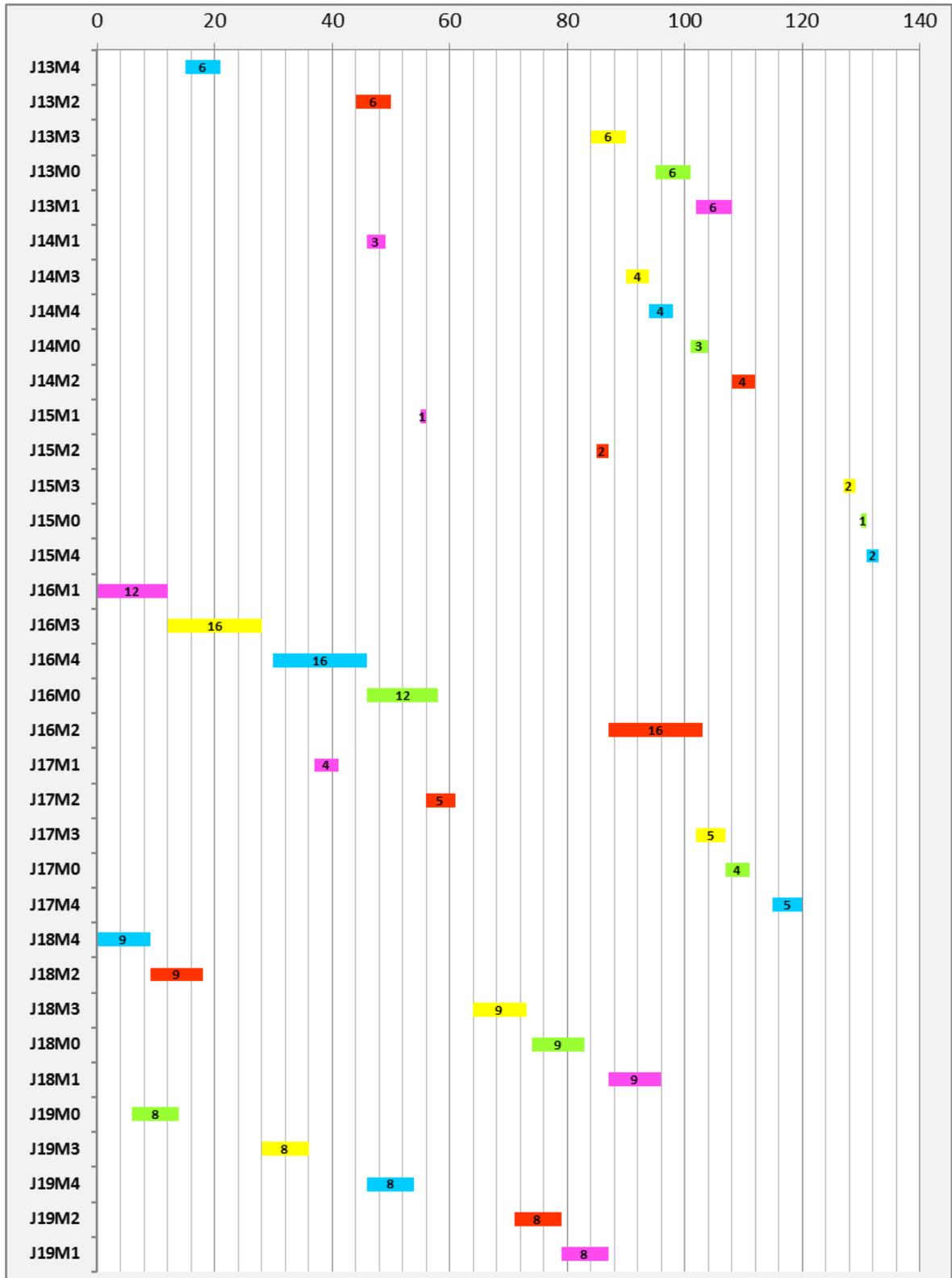


Gráfico 18 - Gráfico de Gantt Solução por *Job* - Programação de Restrições– parte 3

## 4. ANÁLISE

### 4.1. Observações Funcionais das Soluções Testadas

Partindo-se dos fatores restritivos a uma utilização mais intensiva de sistemas de programação restrita enumerados por Giacon (2010) são relacionadas a seguir as observações feitas durante o estudo e desenvolvimento das soluções testadas.

Os critérios relacionados por Giacon (2010), em consonância com outras pesquisas no mesmo sentido são:

- a) dificuldade de customização de procedimentos: não se observou essa dificuldade, a partir dos modelos desenvolvidos, pois são todos bastante flexíveis, operacionalmente independentes, não requerendo customização;
- b) cadastros desatualizados: pode ser um obstáculo, pois qualquer sistema precisa de dados corretos para fornecer bons resultados. A flexibilidade de programação em paralelo à evolução das atividades é um ponto a favor do método das regras de despacho, pela facilidade de correção de eventuais erros detectados;
- c) integração com sistemas: os dois sistemas permitem fácil interface com outros sistemas, principalmente com os *ERP's*;
- d) disciplina em cumprir o programado: mais uma vez o método das regras de despacho leva alguma vantagem pela facilidade de se refazer rapidamente a programação a partir do momento em que o programa deixou de ser cumprido;
- e) adaptação do usuário: efetivamente o desenvolvimento de programas através da programação de restrições vai exigir aprendizado específico, com razoável aprofundamento em fundamentos de programação, o que pode dificultar uma utilização mais intensiva. Já a planilha eletrônica é mais acessível, o que justifica ainda o uso mais intensivo das mesmas;
- f) convencimento da alta administração: com certeza o fator de convencimento será o retorno que a implementação de um recurso pode proporcionar, a princípio com larga vantagem inicial, em termos de investimento inicial necessário, a favor da planilha eletrônica.

### 4.2. Comparação dos Resultados Encontrados

Na solução através da regra de despacho conforme heurística *LPT* chegou-se a um  $C_{max}$  de 148 horas, em seguida com a heurística *MWKR* o  $C_{max}$  foi de 138 horas, enquanto que a solução através da programação de restrições resultou em  $C_{max}$  de 133 horas.

Comparando-se o tempo total de processamento em cada máquina, observa-se o seguinte quadro:

Tabela 9 – Comparação dos Resultados

<b>Máquina</b>	<b>Regra de Despacho <i>LPT</i> (horas)</b>	<b>Regra de Despacho <i>MWKR</i> (horas)</b>	<b>Programação de Restrições (horas)</b>
00	139	136	131
01	136	138	133
02	132	137	132
03	131	131	129
04	148	138	133

Fonte: Dados Trabalhados pelo Autor

Os tempos totais líquidos de processamento nas máquinas 00 e 01 são de 102 horas e nas máquinas 02, 03 e 04 é de 123 horas. As perdas por ociosidade foram maiores na máquina 04 que se pode observar no gráfico de Gantt, antes da operação dos *jobs* 12, 17 e 2 com a regra *LPT*.

A solução pela regra *MWKR* apresenta perdas menores e mais distribuídas do que a *LPT*, sendo que nas máquinas 1 e 4 observa-se um maior dispêndio de tempo, quando comparado à solução pela programação de restrições.

Sempre levando em consideração que as observações realizadas estão limitadas ao problema simulado, nos gráficos por máquina qualquer dos três métodos testados apresentam um comportamento bastante adequado, com ociosidades pequenas, indicando serem estas soluções apropriadas para problemas que buscam minimizar o  $C_{max}$  – *makespan*. Já os gráficos por *Job* mostram que esses métodos não buscam otimizar o tempo de conclusão dos *jobs*, o que pode implicar em aumento exagerado do estoque em processo, bem como atender a outros eventuais requisitos, caso o objetivo fosse minimizar os atrasos de entrega, por exemplo. Nestes casos as regras como o *EDD* (*earliest due date*), ou *FIFO* (*first in first out*) seriam mais adequadas.

### 4.3. Fatores para Utilização e Limitações dos Métodos Testados

O exemplo básico utilizado para simulação dos modelos testados foi extraído de uma estamperia de uma empresa de porte médio, que dispõe de um ERP para planejamento de suas atividades e uso intensivo de planilhas eletrônicas para o *scheduling*, de forma intuitiva, sem iniciativas estruturadas de busca de otimização. Em paralelo às planilhas, painéis visuais de sequenciamento procuram orientar planejadores e operadores ao melhor uso dos recursos e atendimento dos prazos de entrega. Dois indicadores utilizados são o de atendimento diário de programa (ordens atendidas sobre ordens programadas) e o de utilização dos equipamentos (horas utilizadas pelas ordens realizadas sobre horas disponíveis). O histórico apontava o índice de atendimento ao programa em 53% e a ocupação das máquinas em 61%, num cenário de constantes imprevistos e alterações de programa, com um esforço coletivo para manter as máquinas ocupadas, porém visivelmente frustradas pelas constantes alterações. O *scheduling* desta linha é feito por um planejador de linha, técnico de nível médio, estudante de terceiro ano de curso superior em logística, vindo do chão de fábrica, com oito anos de experiência na empresa, sem uma formação específica em planejamento, porém com bom conhecimento de Excel.

Nesse ambiente foi possível testar a planilha com a regra de despacho *LPT* durante uma semana, com um pequeno aumento no índice de atendimento ao programa para 61% e na ocupação das máquinas para 69%, ainda com muitas alterações de programa. O planejador absorveu rapidamente a ferramenta, sugerindo diversas oportunidades de melhoria. A aproximação utilizada ao se considerar tempos padrões de *set up* diluídos nos tempos de operação foi um forte limitador para o alcance de melhores resultados.

Desta observação prática feita, alguns fatores a considerar:

- a) a planilha utilizada para aplicação da regra de despacho *LPT* foi desenvolvida exclusivamente em Excel e requereu conhecimentos avançados do mesmo. Ainda assim, por ser a planilha estática, sem possibilidade de *loops* de decisão, ficou limitada à aplicação da regra operação por operação. Em análise com o planejador que participou do teste e mais dois planejadores de linha, observou-se que os mesmos compreenderam a planilha e seu uso, porém não conheciam os recursos avançados do Excel suficientemente para desenvolvê-la no momento, mas viam a possibilidade de absorvê-la rapidamente, desde que recebessem o treinamento;
- b) a planilha utilizada para aplicação da regra de despacho *MWKR* foi desenvolvida em Excel e Visual Basic, possibilitando a aplicação da regra independente da operação. A

planilha também foi mostrada aos mesmos planejadores, que compreenderam sua lógica e se sentiram em condições de utilizá-la, porém apenas um dos três possuía alguma noção de *Visual Basic* e nenhum dos três se sentiu em condições de desenvolvê-la no momento. Nenhum dos três identificou um obstáculo maior à absorção de um treinamento específico;

- c) a programação de restrições tem toda uma lógica própria. Compreendê-la requereu, além do estudo do problema de *scheduling* em si, o estudo do *software* CPLEX, sua linguagem, ambiente de trabalho e lógica próprios. Mesmo utilizando um programa padrão existente, ou seja, sem ter desenvolvido um programa específico, foram necessárias muitas horas de estudo que permitissem a utilização do mesmo. Apresentado o software aos planejadores, os mesmos ficaram impressionados com os recursos e biblioteca de problemas existentes, porém quando questionados quanto à realização de um treinamento, não se mostraram entusiasmados, demonstrando ter outras prioridades.

Estas observações todas ilustram bem as necessidades e oportunidades existentes no *scheduling* de atividades complexas, como o *jobshop*. O ambiente retratado é bastante típico de organizações industriais de pequeno e médio porte, onde os profissionais de planejamento acabam desenvolvendo suas atividades a partir de suas próprias experiências e iniciativas, utilizando principalmente planilhas eletrônicas bastante simples. O baixo desempenho do planejamento e da utilização dos recursos mostram uma grande oportunidade que pode ser explorada pelo *scheduling*, mas que, sem dúvida, tem etapas importantes a serem vencidas principalmente na preparação dos profissionais que trabalham na área.

## 5. CONCLUSÕES

A resposta à primeira questão do trabalho, que era avaliar se uma metodologia simples desenvolvida em planilha eletrônica poderia representar perda significativa no resultado que inviabilizasse sua utilização é que os resultados mostraram que a solução é viável.

Nos casos estudados observou-se que mesmo o método mais simples trouxe uma melhoria significativa na aplicação prática, com aumento no percentual de tempo de ocupação das máquinas de 53 para 61 e no percentual de atendimento ao programa de 61 para 69, quando comparado com metodologias atualmente em uso.

Comparados ao resultado ótimo, observa-se que existe espaço para melhoria, porém nada que inviabilize estes métodos, tendo em vista a resposta à segunda questão da pesquisa que era o de buscar razões que justificassem a utilização de métodos simples baseados em planilha eletrônica, quando existem métodos que proporcionam resultados melhores.

Fatores como a dinâmica do ambiente de programação, que requer flexibilidade para mudanças, como a pouca capacitação dos profissionais de planejamento e programação disponíveis e, principalmente, como o fator econômico, ainda justificam a utilização de métodos mais simples, tais quais os ora avaliados, tendo em vista as organizações com gestão operacional não plenamente desenvolvida, como o que se encontra em boa parte de pequenas e médias indústrias, focos deste trabalho.

## 6. CONSIDERAÇÕES FINAIS

Neste trabalho procurou-se avaliar, a partir de um problema real de *scheduling* de *jobshop*, típico de uma indústria de pequeno ou médio porte, alternativas de solução mais simples e mais complexa. Alternativas complexas incluem *softwares* especiais desenvolvidos para esta finalidade, com funções específicas, possibilidade de utilização de operadores lógicos e ampla utilização de recursos computacionais. A contrapartida é a simples planilha eletrônica, enriquecida com alguns recursos de programação em *Visual Basic*, por exemplo.

Os resultados encontrados mostraram que, no escopo do problema estudado, com a utilização de métodos adequados, é possível alcançar resultados bastante razoáveis, mesmo que se trabalhe com planilhas eletrônicas.

Soluções como aquelas baseadas na programação de restrições mostram-se bastante poderosas, dependendo ainda de difusão maior de conhecimento específico das metodologias de programação, o que parece ser ainda uma realidade não muito próxima para organizações de pequeno e médio porte.

Neste trabalho foram simulados três métodos, apenas para o problema de minimização do *makespan*. Cabe a avaliação de outros métodos e outros objetivos como sugestão para futuros trabalhos.

## REFERÊNCIAS

ADAMS, J.; BALAS, E.; ZAWACK, D. The Shifting Bottleneck Procedure for Job Shop Scheduling, **Management Science**, 34(3):391-401, 1988

APPLEGATE, D., COOK, W., A Computational Study Of The Jobshop Scheduling Problem. **Operational Research Society Journal**, Comp., 3, 149 – 156, 1991.

BAKER, K. R. **Elements of Sequencing & Scheduling**, New York: Wiley, 1995.

BALAS, E. Machine Scheduling Via Disjunctive Graphs: Implicit Enumeration Algorithm, **Operations Research**, 17, 941 -957, 1969.

BARNES J. W.; LAGUNA M. A Tabu Search Experience in Production Scheduling, **Operations Research**, 41, 141 -156, 1993.

BARTAK, R.; SALIDO, M.A. and ROSSI F., New trends in constraint satisfaction, planning, and scheduling: a survey, **The Knowledge Engineering Review**, v. 25:3, p. 249–279, 2010.

CAMPONOGARA, E. **Métodos de Otimização – Teoria e Prática**, Florianópolis: UFSC, 2006, p. 173 – 184

CARLIER, J.; PINSON, E. An Algorithm For Solving The Jobshop Problem, **Management Science**, 35(2), 165 – 176, 1989. .

CORREA, H. L.; CORREA, C. A. **Administração de Produção e Operações**, 2. ed., São Paulo: Atlas, 2006.

CORREA, H. L.; GIANESI, I. G. N.; CAON, M. **Planejamento, Programação e Controle da Produção: MRP II / ERP, Conceitos, Usos e Implantação**, São Paulo: Atlas, 1997.

DOMINGOS, J. C.; RODRIGUES, C. V.; PEREIRA, N. A.; POLITANO, P. R. Um Sistema de Apoio à Decisão Para Scheduling em Jobshop utilizando lógica Fuzzy. In: XXVIII Encontro Nacional de Engenharia de Produção, Rio de Janeiro, RJ, 2008, **Anais...**

DORNDORF, U.; PESCH, E. Evolution Based Learning In a Jobshop Scheduling Environment, **Operations Research**, 22 (1), 25 – 40, 1995.

EASTMAN, W.L. A Solution to The Traveling Salesman Problem., Presentation to the American Summer Meeting of Econometrics Society, Cambridge, MA, 1958. Automatic Method for Solving Discrete Programming Problems, **Econometrica**, 28, 497 – 520, 1960.

FRANÇA, P. M.; SANTOS, H. C. M. Meta heurística para programação da Produção com Tempos de Preparação Dependentes da Sequência, **Gestão e Produção**, v.2, n.3, p. 228-243, 1995.

GANTT, H. L.; **Work, Wages and Profits**, 2<sup>nd</sup> ed., New York: Engineering Magazine Co., 1916, Reprinted Easton Maryland: Hive Publishing Co., 1973.

GANTT, H. L.. **Organizing for Work**, **Harcourt, Brace and Howe**, New York: Engineering Magazine Co., 1919, Reprinted Easton Maryland: Hive Publishing Co., 1974.

GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability – A Guide to the Theory of NP-Completeness**, New York: Freeman, 1979.

GIACON, E. **Implantação de Sistemas de Programação Detalhada da Produção: Levantamento das Práticas de Programação da Produção na Indústria**, Dissertação de Mestrado apresentada à Escola Politécnica da Universidade de São Paulo, São Paulo, 2010, Disponível em <<http://www.teses.usp.br/teses/disponiveis/3/3136/tde-20082010-144158/pt-br.php>>. Acesso em 20/11/2011.

GIROTTI, L. J.; NISHIMURA, V. S.; MESQUITA, M. A. Simulação em Planilhas para Programação de Produção em Sistemas Jobshop, XXXI ENEGEP, 2011, Belo Horizonte, **Anais...**

GRAHAM, R. L.; LAWLER, E. L.; LENSTRA, J. K.; RINNOOY KAN, A. H. Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey, **Discrete Mathematics**, v.5, p. 287 – 326, 1979.

HAUPT, R., A Survey of Priority-Rule Based Scheduling, **OR Spektrum**, 11:3 – 16, 1989.

JACKSON, J.R. Scheduling a Production Line to Minimize Maximum Tardiness, **Technical Report 43**, Los Angeles: **University of California**, 1955.

JOHNSON, S. M. Optimal Two and Three Stages Production Schedules with Set Up Times Included, **Navy Research Logistics**, Quarter 1, p. 61 -68, 1954

KHATHAWALA, Y.; ALLEN, W. R. Expert Systems and Jobshop Scheduling, **International Journal Of Operations & Production Management**, 13.2, 1993.

LAKATOS, E.M. **Metodologia Científica**, São Paulo: Atlas, 1983.

LEUNG, J. Y .T. **Handbook of Scheduling: Algorithms, Models, and Performance Analysis**, Florida: CRC Press, 2004.

LUSTOSA L. **Planejamento e Controle da Produção**, Rio de Janeiro: Elsevier Campus, 2008

MAINIERI, G. B. **Heurísticas Para a Minimização do Atraso Total no Ambiente Flowshop Com Múltiplos Processadores**, Dissertação de Mestrado, Escola Politécnica – USP, 2009.

MASSOTE, A. A. **Um Estudo de Sequenciamento Aplicado na Linha de Chapas Grossas da Cosipa**, Dissertação de Mestrado Apresentado à Universidade Federal do Rio de Janeiro, Rio de Janeiro, 1991.

McMAHON, G. B.; FLORIAN, M. On Scheduling With Ready Times, And Due Dates To Minimize Maximum Lateness, **Operations Research**, 23(3), 475 – 482, 1975.

McNAUGHTON, R. Scheduling Deadlines and Loss Functions, **Management Science**, 6, 1 – 12, 1959.

METAXIOTIS, K. S.; PSARRAS, J. E.; ERGAZAKIS, K. A. Production Scheduling in ERP Systems: An AI Approach, **Business Management Journal**, 9.2, 221-247, 2003.

MORABITO, R.; IANNONI, A. P. Análise do Sistema Logístico de Recepção de Cana de Açúcar: Um Estudo de Caso Utilizando Simulação, **Gestão e Produção**, v9, n2, p 108-128, agosto 2002.

NOVICK, E.; SMUTNICKI, G. A Fast Taboo Search Algorithm For The Jobshop Scheduling Problem., **Management Science**, 42(6), 797 – 813, 1996.

PEREIRA, M.T. **Proposta de um Modelo de Simulação Computacional para a Programação em Sistemas Assembly Shop**, Dissertação de Mestrado apresentada à Escola Politécnica da Universidade de São Paulo, São Paulo, 2009, Disponível em <<http://www.teses.usp.br/teses/disponiveis/3/3148/tde-06052009-012159/en.php>>. Acesso em :19/11/2011.

PINEDO, M. **Scheduling: Theory, Algorithms, and Systems**, 3. ed., New York: Springer, 2008.

POTTS, C. N.; STRUSEVICH, V.A. Fifty Years of Scheduling: a Survey of Milestones, **Journal of the Operations Research Society**, 60, S41 – S68, 2009.

RODRIGUES, L.F. **Análise Comparativa de Metodologias Utilizadas no Despacho de Caminhões em Minas a Céu Aberto**, Dissertação de Mestrado, Universidade Federal de Minas Gerais, 2006.

RONCONI, D.P.; KIYUZATO, L., SALAMONI, R.; TSAI, C.K. O Problema de Programação de Tarefas Para Minimizar a Soma dos Atrasos: Um Estudo de Caso, XX Encontro Nacional de Engenharia de Produção, São Paulo, SP, 2000, **Anais...**

SADEH, N. **Look ahead Techniques for Micro Opportunistic Jobshop Scheduling**, Pittsburg: Carnegie Mellon University, 1991.

SILVA, E. L.; MENESES, E.M. **Metodologia da Pesquisa e Elaboração de Dissertação**, Florianópolis, Universidade Federal de Santa Catarina, 2000.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R.; CHAMBERS, S. **Administração da Produção**, 2. Ed., São Paulo: Atlas, 2002.

WONG, B. K.; CHONG, J. K. S.; PARK, J. Utilization and Benefits of Expert Systems in Manufacturing: a Study of Large American Industrial Corporations, **International Journal of Operations & Production Management**, v. 14, n.1, p. 38-49, 1994.

## **Apêndice - Pesquisas Recentes de Métodos de Programação Finita**

Estas pesquisas se baseiam em metodologia de pesquisa operacional aplicada para desenvolvimento de soluções específicas para problemas específicos de programação suportados por heurísticas desenvolvidas nas pesquisas e avaliação mediante experimentação computacional.

**BRANCO, F. J. C., Um Novo Método Heurístico Construtivo de Alto Desempenho Para o Problema de Idle Flowshop**, Dissertação de Mestrado, Escola de Engenharia de São Carlos – USP, 2011.

**FUCHIGAMI, H. Y., Flexible Flowline Com Tempos de Setup: Métodos Heurísticos**, Dissertação de Mestrado, Escola de Engenharia de São Carlos – USP, 2010

**BELO FILHO, M. A. F. B., Programação de Produção e Dimensionamento de Lotes Para Flowshop**, Dissertação de Mestrado, Instituto de Ciências Matemáticas e Computacionais, São Carlos, USP, 2010.

**GIGANTE, R. L., Heurística Construtiva Para a Programação de Operações Flowshop Permutacionais**, Dissertação de Mestrado, Escola de Engenharia de São Carlos - USP, 2010.

**CARNEIRO, M. C., Métodos Heurísticos Para Solução de Problema de Programação Flowshop Com Tempos de Setup assimétricos e Dependentes de Sequência**, Dissertação de Mestrado, Escola de Engenharia de São Carlos – USP, 2010.

**MORAIS, M. F., Métodos Heurísticos Construtivos Para Redução do Estoque em Processo em Ambientes de Produção Flowshop Híbridos Com Tempos de Setup Dependentes da Sequência**, Dissertação de Mestrado, Escola de Engenharia de São Carlos – USP, 2008.

**BOYCO, T. J. P., Métodos Heurísticos Para a Programação em Flowshop Permutacional Com Tempos de Setup Separados dos Tempos de Processamento e Independentes dos Tempos de Sequência de Tarefas**, Dissertação de Mestrado, Escola de Engenharia de São Carlos – USP, 2008

**SCARDOELLI, L. Y., Novos Métodos Heurísticos Para Programação de Operações No-Wait Flowshop Com Minimização do Tempo Total de Fluxo**, Dissertação de Mestrado, Escola de Engenharia de São Carlos – USP, 2006.

**ARAUJO JR, L. O. A., Método de Programação de sistemas de Manufatura do Tipo Jobshop Dinâmico Não Determinístico**, Tese de Doutorado, Escola Politécnica – USP, 2006