

TESE

1032

ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

Uma contribuição para a resolução de problemas de programação de operações em sistemas de produção intermitentes flow-shop: a consideração de incertezas

ALEXANDRE FERREIRA DE PINHO

Itajubá — 1999

ALEXANDRE FERREIRA DE PINHO



**Uma contribuição para a resolução de
problemas de programação de operações
em sistemas de produção intermitentes
flow-shop: a consideração de incertezas**

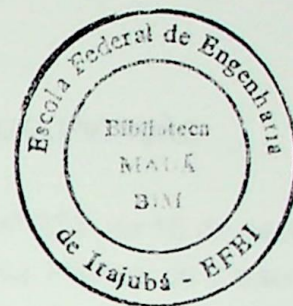
Esta dissertação apresentada é resultado dos cursos de pós-graduação em Engenharia de Produção da Escola Paulista de Engenharia de Itajubá para obtenção do título de Mestre em Ciências em Engenharia de Produção.

Professora: Prof.ª Dr.ª Alcides Basso Maciel, Eng.ª

Coordenadora do Curso de Engenharia de Produção

Itajubá - 1999

ALEXANDRE FERREIRA DE PINHO



Uma contribuição para a resolução de problemas de programação de operações, em sistemas de produção intermitentes flow-shop - a consideração de incertezas

Dissertação apresentada à coordenação dos cursos de pós-graduação em Engenharia de Produção da Escola Federal de Engenharia de Itajubá para obtenção do título de Mestre em Ciências em Engenharia de Produção.

Orientador: Prof. José Arnaldo Barra Montevechi, Dr. Eng.

Área de concentração: Gerência da Produção

Itajubá - 1999



Ministério da Educação
ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ
Reconhecida Lei 3232 - 05/01/1917

ANEXO I

PRONUNCIAMENTO DA BANCA DE EXAME DE QUALIFICAÇÃO

A Banca de Examinadora, abaixo assinada, nomeada pela Portaria nº 333 de 16 de dezembro de 1999, considerando o resultado do Julgamento da Prova de Defesa Pública da Dissertação de Mestrado intitulada: **“Uma Contribuição para a Resolução de Problemas de Programação de Operações em Sistemas de Produção Intermitentes Flow-Shop: a Consideração de Incerteza”**, apresenta pronunciamento no sentido que o Coordenador dos Cursos de Pós-Graduação em Engenharia de Produção da Escola Federal de Engenharia de Itajubá solicite ao DRA (Departamento de Registro Acadêmico) a expedição do título de **Mestre em Ciências em Engenharia de Produção**, na **Área de Concentração de Gerência da Produção**, satisfeitas as demais exigências regimentais, a **Alexandre Ferreira de Pinho**.

Itajubá, 17 de dezembro de 1999.

Prof. Paulo Eigi Miyagi
1º Examinador USP

Prof. Dagoberto Alves de Almeida
2º Examinador - EFEI

Prof. José Arnaldo Barba Montevechi
3º Examinador EFEI



Ministério da Educação
ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ
Reconhecida Lei 3232 - 05/01/1917

A N E X O II

PRONUNCIAMENTO DA BANCA EXAMINADORA

Título da Dissertação: "Uma Contribuição para a Resolução de Problemas de Programação de Operações em Sistemas de Produção Intermitentes Flow-Shop: a Consideração de Incerteza"

Aluno: Alexandre Ferreira de Pinho

JULGAMENTO

Examinadores	Conceito	Rubrica
1º	A+	
2º	A+	
3º	A+	

Resultado Médio: A+ ou seja, aprovado com distinção
Conceito,

Observações: _____

Itajubá, 17 de dezembro de 1999.

Prof. Paulo Eigi Miyagi
1º Examinador USP

Prof. Dagoberto Alves de Almeida
2º Examinador - EFEI

Prof. José Arnaldo Barra Montevechi
3º Examinador - EFEI

*À minha esposa, Elenice, e a minha família,
incentivadores e sempre compreensivos com o pouco
tempo que esta dissertação permitiu-me dedicar-lhes
neste último ano.*

AGRADECIMENTOS

Gostaria de expressar meus sinceros agradecimentos a todas as pessoas que contribuíram de alguma forma, direta ou indiretamente, na realização deste trabalho.

Em particular gostaria de citar o professor José Arnaldo Barra Montevechi, pois, além de orientador, tornou-se um grande amigo.

Também gostaria de agradecer os professores Valério Antônio Plamplona Salomon (UNESP) e Eurycibiades Barra Rosa (EFEI), pelas sugestões e opiniões decisivas na conclusão desse trabalho.

Ao engenheiro Carlos A. Magalhães da IMBEL, pelo seu precioso tempo e informações que possibilitaram a realização do estudo de caso.

Finalmente, gostaria de agradecer a fundação CAPES do Ministério da Educação e do Desporto (MEC), pelo auxílio financeiro. A diretoria do colégio de Aplicação da Fundação de Ensino e Pesquisa de Itajubá (COLFEPI), por permitirem meu afastamento temporário, sem o qual não conseguiria concluir este trabalho. E a paciência e o incentivo constante de minha esposa e de meus pais.

SUMÁRIO

LISTA DE FIGURAS	VIII
LISTA DE SIGLAS	XIII
SIMBOLOGIA	XIV
RESUMO	XVI
ABSTRACT	XVII
Capítulo 1 INTRODUÇÃO	1
1.1. Considerações iniciais	1
1.2. Consideração de incertezas	3
1.3. Matriz com os tempos de operação	5
1.4. Objetivos	7
1.5. Estrutura do trabalho	7
Capítulo 2 REVISÃO BIBLIOGRÁFICA	9
2.1. Considerações iniciais	9
2.2. O problema de programação de operações	9
2.2.1. Notações em problemas de programação	10
2.2.2. Padrões de fluxo de produção	11
2.2.3. Medidas de desempenho	13
2.2.4. Classificação dos problemas de programação	14
2.3. Sistemas de produção intermitentes	15
2.4. Abordagens para o problema de programação	16
2.4.1. Alguns algoritmos heurísticos para a programação de operações	17
2.4.2. Alguns algoritmos ótimos para a programação de operações	18
2.4.3. Algoritmo heurístico NEH	20
2.4.4. Algoritmo ótimo de JOHNSON	23
2.5. Considerações sobre incertezas	25
2.5.1. Lógica fuzzy	25
2.5.2. Fuzzificação	30

2.5.3. Desfuzzificação	33
2.5.4. Operações algébricas com números fuzzy triangulares	34
2.5.5. Comparação entre números fuzzy triangulares	35
2.5.6. Incertezas em algoritmos de programação de operações	37
2.5.7. Diferenças entre TFNs e a distribuição Beta	39
2.6. Considerações sobre tempo padrão	41
2.7. Considerações finais	43
Capítulo 3 METODOLOGIA PROPOSTA	45
3.1. Considerações iniciais	45
3.2. Fuzzificação da matriz com os tempos de operações	45
3.3. Resolução de problemas de programação de operações com incertezas pela desfuzzificação	50
3.3.1. Resolução pelo algoritmo NEH	50
3.3.2. Resolução pelo algoritmo JOHNSON	55
3.4. Nova abordagem proposta para a resolução de problemas de programação de operações com incertezas	57
3.4.1. Resolução pelo algoritmo NEH	57
3.4.2. Resolução pelo algoritmo JOHNSON	63
3.5. Considerações finais	69
Capítulo 4 ESTUDO DE CASO	70
4.1. Considerações iniciais	70
4.2. Estudo de caso real	70
4.2.1. Resolução pelo algoritmo NEH	71
4.2.2. Resolução pelo algoritmo JOHNSON	75
4.3. Considerações finais	79
Capítulo 5 CONCLUSÕES	80
5.1. Considerações iniciais	80
5.2. Contribuição do trabalho	80
5.3. Propostas para trabalhos futuros	82

LISTA DE FIGURAS

Apêndice	MODELO COMPUTACIONAL	83
A.1.	Considerações iniciais	83
A.2.	O software e suas principais telas	83
A.3.	Considerações finais	89
	REFERÊNCIAS BIBLIOGRÁFICAS	90
Anexo A	TABELAS PARA O FATOR EFICIÊNCIA	93
Anexo B	RELAÇÃO DE ARTIGOS ORIGINADOS POR ESTE TRABALHO	95

LISTA DE FIGURAS

Figura 1.1	Matriz com os tempos de processamento de cada operação	2
Figura 1.2	Gráfico de <i>Gantt</i> para a programação de operações J2-J1-J3	2
Figura 1.3	Gráfico de <i>Gantt</i> para a programação de operações J1-J3-J2	2
Figura 1.4	Matriz com os tempos de processamento <i>fuzzy</i> para cada operação	6
Figura 2.1	Relação entre as classes de problemas de programação	11
Figura 2.2	a) Flow-shop b) Flow-shop permutacional	13
Figura 2.3	Algoritmo NEH	20
Figura 2.4	Matriz com os tempos de processamento das operações	21
Figura 2.5	Tempo do fluxo de operação para a ordenação parcial J1-J3	21
Figura 2.6	Tempo do fluxo de operação para a ordenação parcial J3-J1	21
Figura 2.7	Tempo do fluxo de operação para a ordenação parcial J3-J1-J2	22
Figura 2.8	Tempo do fluxo de operação para a ordenação parcial J3-J2-J1	22
Figura 2.9	Tempo do fluxo de operação para a ordenação parcial J2-J3-J1	22
Figura 2.10	Algoritmo de JOHNSON	23
Figura 2.11	Matriz com os tempos de processamento das operações em 2 máquinas	23
Figura 2.12	Solução do problema pelo algoritmo de JOHNSON	24
Figura 2.13	Gráfico de <i>Gantt</i> para o problema	25
Figura 2.14	Cesto com bolinhas azuis	26
Figura 2.15	Cesto com bolinhas vermelhas	26
Figura 2.16	Cestos com bolinhas azuis e vermelhas misturadas	26
Figura 2.17	Representação <i>fuzzy</i> da pergunta: “Este é um cesto de bolinhas azuis?”	27
Figura 2.18	Idade adulta pela lógica binária	28
Figura 2.19	Idade adulta pela lógica <i>fuzzy</i>	29
Figura 2.20	Número <i>fuzzy</i> senoidal para “em torno de cinco”	30
Figura 2.21	Número <i>fuzzy</i> triangular	31
Figura 2.22	Número <i>fuzzy</i> trapezoidal	31

Figura 2.23	Desfuzzificação do TFN (1, 2, 6)	33
Figura 2.24	Matriz com o resultado da desfuzzificação do TFN (2, 1, 6)	34
Figura 2.25	Representação gráfica da operação de adição entre dois TFNs	34
Figura 2.26	Distribuição <i>Beta</i> de probabilidades extraída de CONTADOR (1997)	40
Figura 3.1	Matriz com os tempos de processamento das operações determinísticos	49
Figura 3.2	Matriz com os tempos de processamento após processo de fuzzificação	49
Figura 3.3	Tempo de fluxo de operação para a ordenação parcial J1-J3	51
Figura 3.4	Tempo de fluxo de operação para a ordenação parcial J3-J1	51
Figura 3.5	Tempo de fluxo de operação para a ordenação parcial J3-J1-J2	51
Figura 3.6	Tempo de fluxo de operação para a ordenação parcial J3-J2-J1	52
Figura 3.7	Tempo de fluxo de operação para a ordenação parcial J2-J3-J1	52
Figura 3.8	Matriz com os tempos de processamento após o processo de desfuzzificação	52
Figura 3.9	Tempo de fluxo de operação para a ordenação parcial J1-J3	53
Figura 3.10	Tempo de fluxo de operação para a ordenação parcial J3-J1	53
Figura 3.11	Tempo de fluxo de operação para a ordenação parcial J3-J1-J2	53
Figura 3.12	Tempo de fluxo de operação para a ordenação parcial J3-J2-J1	54
Figura 3.13	Tempo de fluxo de operação para a ordenação parcial J2-J3-J1	54
Figura 3.14	Gráfico de <i>Gantt</i> obtido pela desfuzzificação dos tempos de processamento	54
Figura 3.15	Matriz com os tempos de processamento das operações em duas máquinas	55
Figura 3.16	Matriz com os tempos de processamento com incertezas	55
Figura 3.17	Matriz com os tempos de processamento após a desfuzzificação	56
Figura 3.18	Solução do problema pelo algoritmo de JOHNSON através da desfuzzificação	56

Figura 3.19	Gráfico de <i>Gantt</i> obtido para o problema	57
Figura 3.20	Matriz com os tempos de processamento	57
Figura 3.21	Matriz com os tempos de processamento <i>fuzzy</i>	58
Figura 3.22	Matriz com os tempos de processamento otimistas	58
Figura 3.23	Matriz com os tempos de processamento de maior possibilidade	59
Figura 3.24	Matriz com os tempos de processamento pessimistas	59
Figura 3.25	Matriz com os tempos de processamento representados pela sua desfuzzificação	59
Figura 3.26	Resultado da programação das operações	60
Figura 3.27	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação otimistas com os valores de tempos de maior possibilidade	61
Figura 3.28	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação otimistas com os valores de tempos pessimistas	61
Figura 3.29	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação otimistas com os valores de tempos da desfuzzificação	62
Figura 3.30	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempo otimistas	62
Figura 3.31	Análise dos resultados pelo algoritmo NEH	63
Figura 3.32	Matriz com os tempos de processamento	63
Figura 3.33	Matriz com os tempos de operação <i>fuzzy</i>	64
Figura 3.34	Matriz com os tempos de processamento otimistas	64
Figura 3.35	Matriz com os tempos de processamento de maior possibilidade	65
Figura 3.36	Matriz com os tempos de processamento pessimistas	65
Figura 3.37	Matriz com os tempos de processamento representados pela sua desfuzzificação	65
Figura 3.38	Resultado da programação das operações	66
Figura 3.39	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos de otimistas	67
Figura 3.40	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos de maior possibilidade	67

Figura 3.41	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos pessimistas	67
Figura 3.42	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação otimistas com os valores de tempo da desfuzzificação	68
Figura 3.43	Análise dos resultados pelo algoritmo JOHNSON	68
Figura 4.1	Matriz com os tempos de processamento fornecida pela empresa	70
Figura 4.2	Entrada da matriz com os tempos de operação	71
Figura 4.3	Matriz com os tempos de operação <i>fuzzy</i>	71
Figura 4.4	Resultado da programação das operações	72
Figura 4.5	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação otimistas com os valores de tempo da desfuzzificação	72
Figura 4.6	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos otimistas	73
Figura 4.7	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos de maior possibilidade	73
Figura 4.8	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos pessimistas	74
Figura 4.9	Análise dos resultados pelo algoritmo NEH	74
Figura 4.10	Programação de operações com a visualização dos gráficos de <i>Gantt</i>	75
Figura 4.11	Entrada da matriz com tempos de operação	75
Figura 4.12	Matriz com os tempos de operação <i>fuzzy</i>	76
Figura 4.13	Resultado da programação das operações	76
Figura 4.14	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação otimistas com os valores de tempo da desfuzzificação	77
Figura 4.15	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos otimistas	77
Figura 4.16	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos de maior possibilidade	78

Figura 4.17	Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos pessimistas	78
Figura 4.18	Análise dos resultados pelo algoritmo JOHNSON	78
Figura A.1	Fluxograma do modelo computacional proposto	84
Figura A.2	Tela inicial do sistema	85
Figura A.3	Janela para iniciar um novo estudo	86
Figura A.4	Janela com os tempos de processamento <i>fuzzy</i>	86
Figura A.5	Janela de resultados sem os gráficos de <i>Gantt</i>	87
Figura A.6	Janela de resultados com os gráficos de <i>Gantt</i>	87
Figura A.7	Janela para análise dos resultados sem os gráficos de <i>Gantt</i>	88
Figura A.8	Janela para análise dos resultados com os gráficos de <i>Gantt</i>	88

LISTA DE SIGLAS

CDS	Algoritmo desenvolvido por Campbell, Dudek e Smith
CPM	Critical Path Method (Método do caminho crítico)
F	Flow-shop
IMBEL	Indústria de Material Bélico do Brasil
J	Job-shop
NEH	Algoritmo desenvolvido por Nawaz, Enscore e Ham
O	Open-shop
PERT	Program Evaluation and Review Technique (Técnicas de Avaliação e Revisão de Programas)
TFN	Triangular fuzzy number
TrFN	Trapezoidal fuzzy number
SVGA	Super Video Graphics Array
RAM	Random Access Memory
MHz	Mega Hertz
Mb	Mega Bytes

SIMBOLOGIA

\bar{C}	Data média de término de um conjunto de n tarefas
\bar{F}	Tempo médio de fluxo de um conjunto de n tarefas
\hat{A}	Se A um número <i>fuzzy</i> , \hat{A} será seu representativo ordinário
\exists	Quantificador existencial
\forall	Quantificador universal
\parallel	Máquina única
a_1	Valor otimista do número triangular <i>fuzzy</i>
a_2	Valor de maior possibilidade do número triangular <i>fuzzy</i>
a_3	Valor pessimista do número triangular <i>fuzzy</i>
a_i	Período permitido de permanência da tarefa J_i na oficina
C_i	Data de término da tarefa J_i
C_{\max}	Duração total da programação (<i>makespan</i>)
d_i	<i>Due-date</i> – data de entrega da tarefa i
F_{perm}	<i>Flow-shop</i> permutacional
f_1	Fator de ritmo de trabalho
f_2	Fator de recuperação de fadiga
F_i	Tempo de fluxo total da tarefa J_i
F_{\max}	Tempo máximo de permanência de um conjunto de n tarefas
g_i	Conjunto de operações para a tarefa J_i
$J, k\text{-parallel}$	<i>Job-shop</i> com k máquinas duplicadas em cada estágio
J_i	Tarefa i de um total de n tarefas
$k\text{-parallel}$	k máquinas em paralelo
L_i	Atraso da tarefa J_i
m	Quantidade total de máquinas
M_j	Máquina j de um total de m máquinas
n	Quantidade total de tarefas
$NP\text{-hard}$	Problemas de natureza combinatorial complexos
op_{ik}	Conjunto de operações k da tarefa i
p_i	Tempo total de processamento da tarefa J_i
p_{ik}	Tempo de processamento da tarefa i na máquina k
r_i	<i>Release-time</i> – data de liberação da tarefa i

$t_0 = a$	Mínimo tempo estimado para execução de uma atividade
t_1	Tempo médio cronometrado
T_i	Atraso máximo da tarefa J_i
T_{\max}	Tempo máximo de atraso de um conjunto de n tarefas
$t_{mp} = m$	Tempo mais provável estimado para execução de uma atividade
T_p	Tempo padrão de uma operação
$t_p = b$	Máximo tempo estimado para execução de uma atividade
u.t.	Unidade de tempo
W_i	Tempo total de espera da tarefa J_i
z_1	Fadiga mental
z_2	Fadiga física
z_3	Abono por monotonia
z_4	Tolerância pessoal
z_5	Ajustes e trocas de ferramentas
μ	Função de pertinência

RESUMO

Dentre os principais fatores que compõem o controle e gerenciamento da produção industrial, encontra-se a programação de operações. Esta programação possui como principal variável os tempos de processamento de cada operação, sendo estes tempos sujeitos a incertezas. Deste modo, esta dissertação é direcionada à algoritmos de programação de operações, com considerações de incertezas, para sistemas de produção do tipo intermitente *flow-shop* permutacional.

Durante a década de 90 publicou-se vários trabalhos considerando-se as incertezas existentes, nos tempos de processamento, através da lógica *fuzzy*. Constatou-se que na resolução de problemas de programação de operações *flow-shop* com incertezas, onde o objetivo é obter o tempo total de produção (*makespan*), são utilizadas dois procedimentos. O primeiro, é a fuzzificação da matriz de tempos de operação de cada atividade, ou seja, transformar os tempos de processamento determinísticos em números *fuzzy*. O segundo, é a comparação entre diversos números *fuzzy*, na intenção de ordená-los crescente ou decrescentemente. Sendo esta comparação realizada através da desfuzzificação desses números *fuzzy*, ou seja, determinar um número real que represente o número *fuzzy* em questão.

Nestes trabalhos, verificou-se que o processo de fuzzificação é de total responsabilidade do especialista, ou seja, uma pessoa que conhece muito bem o problema em questão, e tem total capacidade de ponderar entre as incertezas que existem nos tempos de processamento das atividades. Entretanto, conforme a quantidade de máquinas e tarefas envolvidas no problema, o processo de fuzzificação torna-se exaustivo. Também verificou-se que o processo de comparação entre diversos números *fuzzy*, pela desfuzzificação, nem sempre gera o melhor resultado em problemas de programação de operações *flow-shop* com incertezas.

Deste modo, esta dissertação apresenta duas propostas. A primeira é propor uma nova abordagem ao processo de fuzzificação, baseando-se na teoria de obtenção de tempo padrão das operações, no intuito de auxiliar o especialista na inserção de incertezas nos tempos de processamento das atividades. A segunda, é propor uma nova abordagem na resolução de problemas de programação de operações, além do procedimento de comparação, através da desfuzzificação, utilizado pelos autores pesquisados.

ABSTRACT

One of the main issues in production planning and control is production scheduling. The scheduling process considers as one of its main parameters the time processing of each operation, which is inherently uncertainty. Accordingly, this dissertation is addressed to operations program algorithms, with considerations of uncertainties, for flow-shop scheduling problem.

During the 90's it has being published several works considering the existent uncertainties in the time processing, applying the fuzzy logic. It was verified that two procedures are used in the resolution of flow-shop problems under uncertainties, where the objective function is to obtain the total time of production (makespan). The first, is the fuzzyfication of operation time for each activity. The second, is the comparison among several fuzzy numbers, being this comparison accomplished through the defuzzyfication of those fuzzy numbers.

In this dissertation, it is verified that the fuzzyfication process is a matter of specialist's responsibility. Usually, a person that knows the addressed problem, and has total capacity to ponder among the uncertainties that exist in the activities processing times. However, according to the amount of machines and tasks involved in the problem, the fuzzyfication process is exhaustive. It is also verified that the comparison process among several fuzzy numbers, in the defuzzyfication, doesn't always generate the best result for problems of flow-shop scheduling under uncertainties.

This dissertation presents two purposes. The first is to use a new approach to the fuzzyfication process for obtaining operations standard time. The second, is to use a new approach in the resolution of dispatching problems through defuzzyfication, besides the comparison procedure presented in the researched literature.

Capítulo 1: Introdução

1.1 – Considerações iniciais

O controle da produção é dividido, segundo BURBIDGE (1975), em 3 níveis:

- Plano de produção (*Programming*);
- Emissão de ordens (*Ordering*);
- Liberação (*Dispatching*).

O plano de produção, denominado também plano mestre ou programa mestre de produção, é o documento que especifica a quantidade de cada um dos produtos que deverá ser produzida no próximo período de programação. Ou seja, especifica *o que e quanto* produzir.

A emissão de ordens define o volume de produção entre os setores produtivos e a solicitação de material proveniente de fornecedores. A finalidade de uma ordem é passar instruções sobre *o que, como, onde e quando* os diversos setores da fábrica devem executar determinadas tarefas.

A liberação define a programação das quantidades produzidas por máquinas necessárias para completar as ordens de fabricação na data pré estabelecida.

Segundo MOCCELLIN (1992), o terceiro nível do controle da produção, liberação, é o mais específico e o mais complexo devido a sua natureza combinatorial, tendo como atividade relevante a *Programação de Operações*.

Durante a pesquisa bibliográfica constatou-se que existem diferenças entre os termos *sequenciamento* e *programação de operações*. Segundo MOCCELLIN (1994), o termo sequenciamento refere-se à ordenação de tarefas (operações) em *uma* determinada máquina. Já o termo programação de operações refere-se à ordenação, ao longo do tempo, das tarefas em *uma* ou *diversas* máquinas. Nesta dissertação considerou-se o termo programação de operações, pois, os algoritmos utilizados, buscam uma ordenação em mais de uma máquina, ao longo do tempo.

BURBIDGE (1975), afirmava que uma troca adequada na programação de operações pode ocasionar em economias na produção. De fato, se considerarmos os tempos de

processamento de cada tarefa (J_i) em cada máquina (M_j) mostrado na matriz da figura 1.1, e considerarmos duas programações de operações diferentes, J2-J1-J3 e J1-J3-J2, pode-se notar que a simples troca de posição entre as tarefas ocasionou em ganho de tempo. De 16 u.t. (unidade de tempo), para a ordenação J2-J1-J3, mostrada na figura 1.2, para 13 u.t., na ordenação J1-J3-J2, conforme mostrado na figura 1.3.

Tarefas (J_i)	Máquinas (M_j)		
	M1	M2	M3
J1	1	3	4
J2	4	2	2
J3	3	1	3

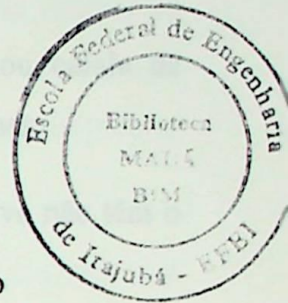


Figura 1.1 – Matriz com os tempos de processamento de cada operação

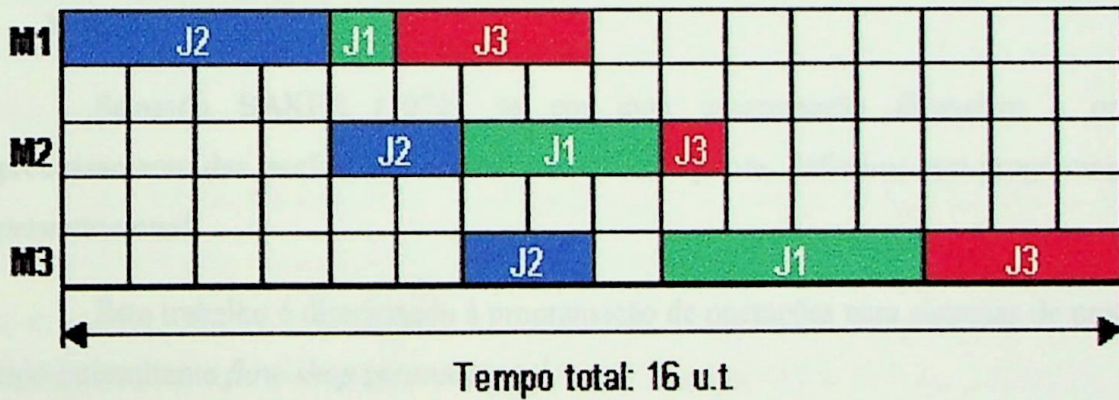


Figura 1.2 – Gráfico de *Gantt* para a programação de operações J2-J1-J3

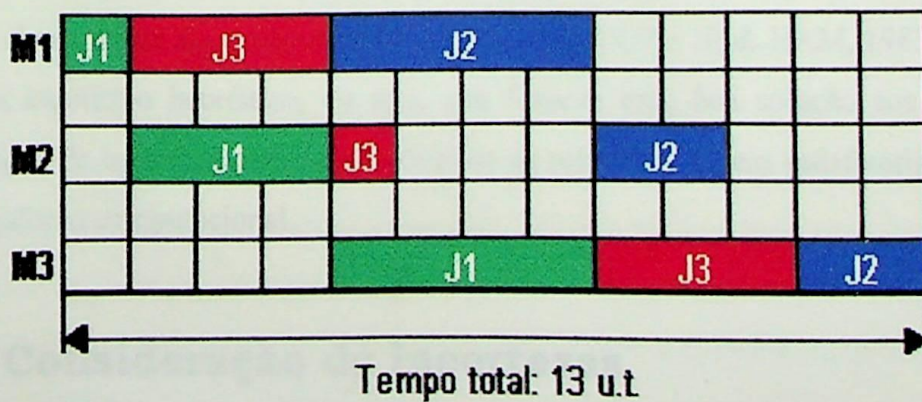


Figura 1.3 – Gráfico de *Gantt* para a programação de operações J1-J3-J2

A programação de operações é influenciada pelo tipo de sistema de produção utilizado pela empresa. JOHNSON & MONTGOMERY (1974) classificam o sistema de produção em:

- *Sistema contínuo*: onde são fabricados, em grande quantidade, poucos tipos de produtos similares;
- *Sistema Intermitente*: onde sempre ocorrem mudanças de um produto para outro nos estágios produtivos. Nesta classe, distinguem-se, basicamente, dois tipos de sistemas:
 - *Intermitente flow-shop*: onde os itens fabricados em uma linha ou célula de manufatura têm a mesma seqüência de operações nas diversas máquinas;
 - *Intermitente job-shop*: onde os itens fabricados em um setor produtivo não têm o mesmo roteiro de fabricação;
- *Sistema para grande projeto*: onde são produzidos itens complexos e especiais, muitas vezes únicos.

Segundo BAKER (1974), se em uma programação *flow-shop* a ordem de processamento das tarefas for a mesma entre as máquinas, define-se esta programação como *permutacional*.

Este trabalho é direcionado à programação de operações para sistemas de produção do tipo intermitente *flow-shop* permutacional.

Para o desenvolvimento desta dissertação, escolheu-se dois algoritmos. O primeiro, chamado de algoritmo de JOHNSON (JOHNSON, 1954), considerado como um algoritmo ótimo, ou seja, que fornece a solução exata ao problema de programação de operações. O segundo, chamado de algoritmo NEH (NAWAZ, ENSCORE JR & HAM, 1983), considerado como um algoritmo heurístico, ou seja, que fornece uma boa solução aos problemas de programação de operações, sendo aceitável se os resultados forem satisfatórios do ponto de vista do esforço computacional.

1.2 – Consideração de incertezas

Em algoritmos de programação de operações, a principal variável para os cálculos, é o tempo de processamento de cada operação. Vários autores assumem este tempo de processamento como exatos. Esta é uma suposição que segundo FORTEMPS (1997) é muito forte, pois pode causar dificuldades em aplicações práticas de problemas de programação.

Tais dificuldades ocorrem, pois este tempo de processamento estimado pode variar dependendo do operador que executará a atividade, conforme a máquina que será utilizada, do ferramental empregado, etc. Desta maneira, portanto, pode-se afirmar que o tempo de execução das atividades está sujeito a incertezas.

Para considerar as incertezas e ambigüidades existentes nos sistemas de manufatura, a lógica *fuzzy* tem provado ser uma ferramenta efetiva para levar em consideração o impreciso (MONTEVECHI, 1995). A maioria dos processos industriais são complexos devido ao número de variáveis envolvidas e a dificuldade de quantificação destas variáveis. Por vezes, diversas destas grandezas têm seus valores estabelecidos através de processos subjetivos, sendo classificados por meios de valores lingüísticos.

A lógica *fuzzy* é um meio de aproximar a precisão da matemática clássica e a inexatidão do mundo real. Esta teoria consegue manipular e operar quantidades exatas e inexatas (quantificadas através de valores lingüísticos). Existe uma grande variedade de tipos de números *fuzzy* (KAUFMANN e GUPTA, 1988), entretanto neste trabalho, tratou-se de um tipo específico: *números fuzzy triangulares (triangular fuzzy numbers - TFN)*.

As incertezas existentes nestes parâmetros de tempo de processamento receberam pouca atenção nas décadas de 70 e 80. Somente na década de 90 começaram a surgir trabalhos que levavam em consideração estas incertezas. Por exemplo, tem-se MCCAHERN & LEE (1992), ISHIBUCHI, YAMAMOTO, MURATA & TANAKA (1994), STÜTZLE (1998) e CURY (1999) que utilizaram números *fuzzy* em problemas de programação de operações para sistemas intermitentes do tipo *flow-shop*.

Esses autores, utilizaram na elaboração de seus trabalhos, duas técnicas aplicadas aos números *fuzzy*. A primeira, chamada de fuzzificação (*fuzzification*), consiste na geração do número *fuzzy* (WEBER, 1997), considerando a opinião de um especialista. A Segunda, consiste no processo de comparação entre diversos números *fuzzy*, na intenção de se estabelecer uma ordenação (*ranking*) crescente ou decrescente para os números avaliados. Para a realização dessa comparação é necessário utilizar um processo conhecido como desfuzzificação (*defuzzification*) do número *fuzzy*, que segundo (COX, 1995) é a conversão de um número *fuzzy* em um número real.

A comparação entre os diversos números *fuzzy* é necessária pois, os algoritmos de programação de operações *flow-shop*, onde a função objetivo é minimizar o tempo máximo de

duração de um conjunto de n tarefas, ou seja, obter o tempo total de produção (*makespan*), utilizam a comparação entre números *fuzzy* em sua resolução (FORTEMPS, 1997).

Porém, ao se utilizar a técnica de comparação pela desfuzzificação na resolução desses problemas, pode-se correr o risco de gerar resultados não satisfatórios do ponto de vista da programação de operações. Isto ocorre, pois o processo de desfuzzificação elimina a sensibilidade existente entre as possíveis variações de tempos de processamento, que podem ocorrer, ao se utilizar um algoritmo de programação de operações.

1.3 – Matriz com os tempos de operação

MOCCELLIN (1992) afirma que a programação da produção é fortemente dependente das particularidades do problema em questão e da distribuição dos dados, que neste caso são os tempos de processamento das tarefas nas diversas máquinas.

Desta forma, a matriz com os tempos de processamento de cada operação em cada máquina, semelhante a matriz da figura 1.1, deve fornecer um *retrato* muito próximo da realidade do que acontece na produção. Uma das maneiras de se construir essa matriz é através do uso de técnicas de obtenção de *tempo padrão*.

A obtenção do tempo padrão, segundo BARNES (1986), consiste em determinar o tempo necessário para uma pessoa qualificada e bem treinada, trabalhando em ritmo normal, executar uma tarefa. Ou seja, o resultado deste estudo é o tempo gasto por um operador adaptado ao trabalho e completamente treinado no método específico executar uma atividade trabalhando em um ritmo considerado normal.

Desse modo, ao considerarmos que para cada um dos tempos de processamento das diversas operações mostrados anteriormente na matriz da figura 1.1, foram realizados os procedimentos para obtenção do tempo padrão, teríamos, desta forma, uma boa aproximação do tempo real de cada operação.

Porém, em problemas de programação de operações com considerações de incertezas, a matriz com os tempos de processamento para cada atividade é na verdade uma matriz com os tempos *fuzzy*. Um exemplo desta matriz pode ser visto na figura 1.4, onde os tempos de processamento estão representados pelos *números fuzzy triangulares*.

Tarefas (J_i)	Máquinas (M_j)		
	M1	M2	M3
J1	(0.5, 1, 2)	(2, 3, 4)	(3.5, 4, 6)
J2	(1, 4, 4.5)	(1, 2, 3)	(1.5, 2, 2.5)
J3	(1, 3, 5)	(0, 1, 3)	(2, 3, 5)

Figura 1.4 – Matriz com os tempos de processamento *fuzzy* para cada operação

A geração de cada um dos tempos de operação *fuzzy* foi obtida através do processo de fuzziificação (*fuzzification*) do número exato.

A lógica *fuzzy* estabelece que a fuzziificação é sempre realizada baseada na opinião de um especialista, ou seja, por uma pessoa que domina completamente o problema em questão, e tem total capacidade de ponderar entre as incertezas que existem nos tempos de processamento das atividades. Deste forma, para problemas de programação de operações o especialista deverá avaliar as condições que envolvem as tarefas a serem analisadas, e ponderar entre as divergências que podem ocorrer no tempo de operação para cada uma dessas atividades.

É fácil perceber, portanto, que esta é uma tarefa árdua, pois quanto maior a quantidade de máquinas e tarefas envolvidas na programação, maior será a quantidade de tempos de operações existentes no problema que se deseja avaliar.

Desse modo, foi desenvolvida uma metodologia com o intuito de auxiliar o especialista a ponderar entre os tempos de operações existentes em cada uma das atividades em questão. Esta metodologia gera automaticamente, com base nos tempos determinísticos de cada operação, uma matriz com os tempos de processamento *fuzzy*, semelhante a apresentada na figura 1.4, como sugestão inicial do problema, tendo o especialista total liberdade de alterar qualquer um dos números *fuzzy* gerados.

Esta geração da matriz com os tempos de processamento *fuzzy* é obtida com base em fundamentos científicos obtidos pelo princípio de obtenção de tempo padrão proposta por BARNES (1986) e será melhor discutida no *Capítulo 3 – Metodologia proposta*.

1.4 – Objetivos

O presente trabalho tem como objetivo principal fornecer uma contribuição para o tratamento de incertezas em algoritmos de programação de operações em sistemas intermitentes do tipo *flow-shop* permutacional. Esta contribuição estará voltada para os processos de fuzzificação e comparação através da desfuzzificação existentes no tratamento de incertezas pela lógica *fuzzy* nestes algoritmos.

Pretende-se atingir este objetivo com o cumprimento das seguintes metas:

- Utilizar as técnicas de obtenção de tempo padrão para a fuzzificação da matriz com os tempos de processamento de cada operação como sugestão inicial de entrada de dados;
- Desenvolver uma nova abordagem para obtenção da programação de operações, além da comparação pela desfuzzificação dos números *fuzzy*;
- Utilizar dois algoritmos de programação de operações *flow-shop* permutacional conhecidos como JOHNSON (JOHNSON, 1954) e como NEH (NAWAZ, ENSCORE JR & HAM, 1983) na intenção de testar os resultados pela nova abordagem proposta com os resultados obtidos através da comparação pela desfuzzificação, relatada na literatura;
- Implementar em um software a metodologia desenvolvida.

1.5 – Estrutura do trabalho

Além deste primeiro capítulo introdutório, o trabalho apresenta mais quatro capítulos. No capítulo 2, revisão bibliográfica, serão apresentados os conceitos e definições sobre algoritmos de programação de operações *flow-shop* permutacional, onde se escolherá dois algoritmos para o desenvolvimento da metodologia proposta. Também serão apresentados, neste capítulo, os principais conceitos sobre a lógica *fuzzy* e sobre a obtenção de tempo padrão de operações.

O capítulo 3 contém a metodologia desenvolvida, onde será apresentada uma metodologia para a fuzzificação da matriz de tempos de operações baseada nas técnicas de obtenção de tempo padrão. Também será realizada uma avaliação na resolução dos algoritmos de programação de operações pelo processo de comparação através da desfuzzificação com a nova abordagem proposta.

No capítulo 4, estudo de caso, será apresentado a implementação do modelo proposto em situação real.

O capítulo 5, conclusões, trará as contribuições e recomendações decorridas do presente trabalho.

A dissertação também possui um apêndice: *Modelo computacional*, onde será apresentado um software protótipo que facilitará a implementação da metodologia desenvolvida, com suas principais telas e funções. Este assunto foi apresentado em forma de apêndice para se evitar a descontinuidade da dissertação.

Capítulo 2: Revisão bibliográfica

2.1 – Considerações iniciais

Neste capítulo será realizado, inicialmente, um levantamento dos conceitos e definições referentes ao problema de programação de operações.

Conforme comentado na introdução desse trabalho, será dada especial atenção aos sistemas de produção intermitentes do tipo *flow-shop* permutacional, onde serão abordados em ordem cronológica, alguns dos algoritmos heurísticos e otimizantes existentes para estes tipos de sistemas.

Serão apresentados, neste capítulo, os principais conceitos sobre a lógica *fuzzy*, e em especial os *números fuzzy triangulares*. Na seqüência, serão comentadas algumas das abordagens utilizadas na inserção de incertezas, através da lógica *fuzzy*, em algoritmos de programação de operações que foram publicadas nos últimos tempos.

Também será realizada uma comparação entre os *números fuzzy triangulares* e a distribuição *beta*, com o intuito de demonstrar porque optou-se pela lógica *fuzzy* e não pela distribuição *beta* na consideração de incertezas.

Por fim, serão discutidos os principais fatores existentes na expressão de obtenção do tempo padrão de uma operação.

2.2 – O problema de programação de operações

Segundo BAKER (1974), a programação de operações é a definição da ordem de entrada das tarefas a serem executadas na produção, ou seja, determinar como as tarefas devem ser conduzidas de uma máquina para outra. Entende-se por tarefas qualquer tipo de operação (trabalhos, “jobs”, etc.) que se desenvolva em um produto (peça, item, etc.). A programação de operações tem como objetivo maximizar a taxa de produção e de minimizar o tempo total do fluxo de operação.

MACCARTHY e LIU (1993), afirmam que o problema de programação pode ser definido, de um modo geral, como a alocação de recursos no tempo de forma a executar um conjunto de tarefas.

Uma programação ruim, pode conduzir facilmente à utilização não otimizada de recursos e materiais, acarretando atrasos nas datas de entregas, baixa qualidade de produtos, baixa lucratividade, etc. (MAZZUCCO JR, 1999).

Desta forma, o resultado gerado pela programação é a seleção de uma ordenação adequada de operações, ou seja, o roteamento do processo de produção, ao qual resultará na conclusão de todas as tarefas do conjunto, no menor tempo possível.

2.2.1 – Notações em problemas de programação

Tem-se as seguintes notações para problemas de programação de operações, extraídas de MOCCELIN (1994):

- Um conjunto de n tarefas que devem ser processadas em m máquinas é conhecido como $J = \{J_1, J_2, \dots, J_i, \dots, J_n\}$, sendo o tempo de processamento da tarefa J_i na máquina k é p_{ik} ($i = 1, 2, \dots, n; k = 1, 2, \dots, m$).
- Se uma determinada tarefa não tiver operação em uma certa máquina, seu correspondente tempo de processamento é considerado igual a zero.
- A data de liberação da tarefa J_i , ou seja, a data a partir da qual ela pode ser executada é conhecida como r_i (*release-time*).
- A data de entrega da tarefa J_i , ou seja, a data limite estabelecida para término da tarefa, é conhecida como d_i (*due-date*).
- O período permitido de permanência da tarefa J_i na seção produtiva é chamado de a_i e é calculado pela expressão (2.1).

$$a_i = d_i - r_i \quad (2.1)$$

- Cada tarefa J_i é um conjunto g_i de operações ordenadas onde $J_i = \{op_{i1}, op_{i2}, \dots, op_{ig_i}\}$. Sendo que para cada operação op_{ij} , tem-se um número inteiro que identifica a máquina que deverá executar a j -ésima operação (op_{ij}) da tarefa J_i , conhecido como m_{ij} .
- O tempo de processamento da operação op_{ij} , na máquina m_{ij} é chamado de p_{ij} , sendo p_i o tempo total de processamento de J_i , calculado pela expressão (2.2).

$$p_i = \sum_{j=1}^{g_i} p_{ij} \quad (2.2)$$

- O tempo de espera da operação op_{ij} , após ter sido completada a operação $op_{i(j-1)}$ é conhecido como W_{ij} , sendo o tempo total de espera da tarefa J_i definido por (2.3).

$$W_i = \sum_{j=1}^{g_i} W_{ij} \tag{2.3}$$

2.2.2 – Padrões de fluxo de produção

O padrão de fluxo de produção consiste no direcionamento das tarefas com relação às máquinas, utilizadas por um certo período de tempo.

MACCARTHY & LIU (1993) afirmam que o número de máquinas m e a quantidade de máquinas de cada tipo k_j ($j = 1, 2, \dots, m$) disponíveis para desempenhar as tarefas são importantes na determinação do padrão de fluxo de produção. Com base nestas características, os autores, propuseram uma classificação entre as classes de problemas de programação, esquematizada na figura 2.1.

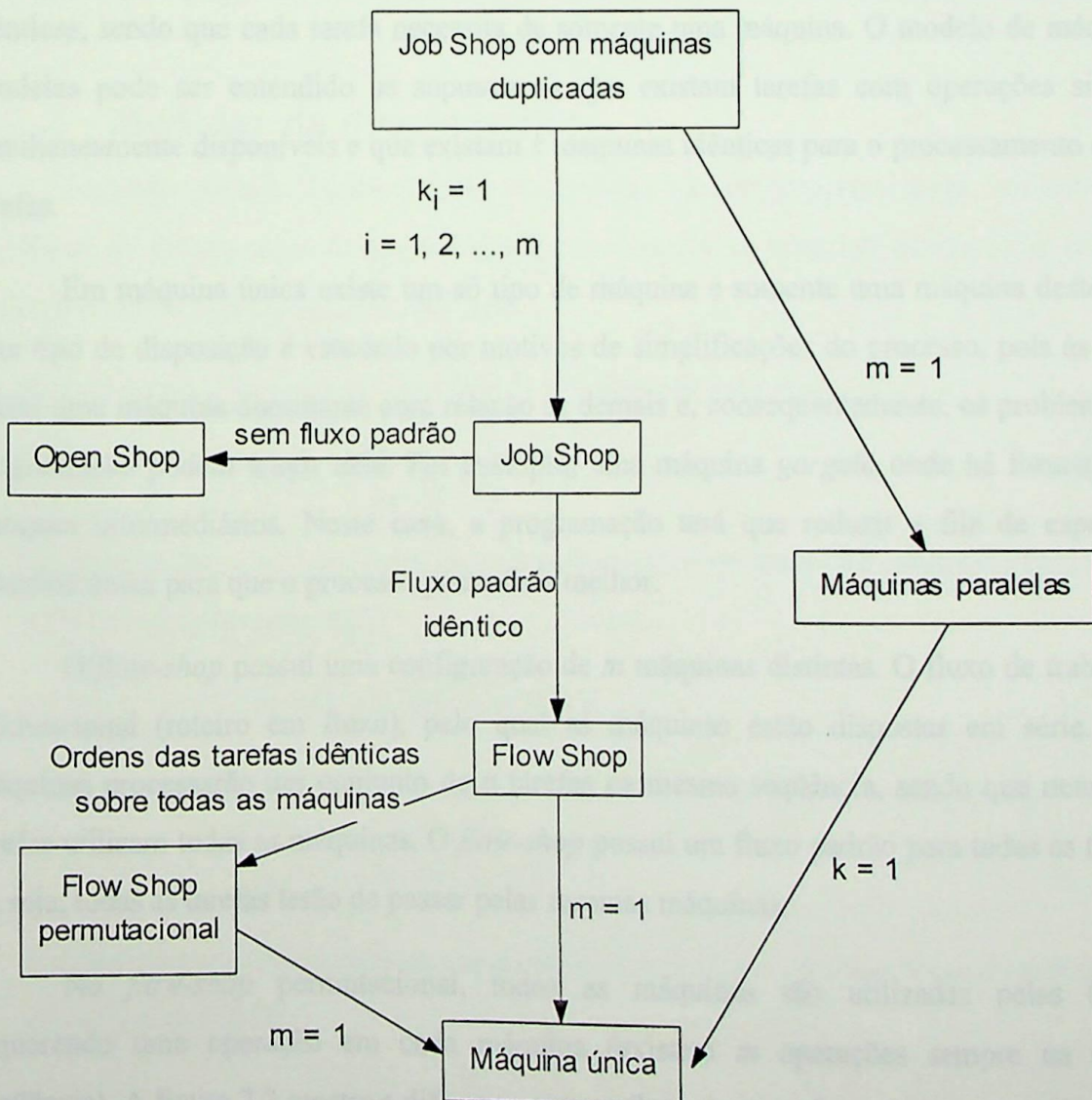


Figura 2.1 – Relação entre as classes de problemas de programação

O *job-shop* caracteriza-se por apresentar um conjunto de m máquinas distintas, onde o fluxo de trabalho não é unidirecional, permitindo a cada máquina a possibilidade de iniciar, terminar e prosseguir um processo (BAKER, 1974).

O *job-shop* com máquinas duplicadas trata-se de um tipo de *job-shop* onde a importância maior é dada aos estágios de processamento ao invés de máquinas. Existem k_i máquinas idênticas em cada estágio e qualquer tarefa que requerer um estágio necessita de uma e só uma dessas máquinas.

Segundo (BETIOLI & BELHOT, 1996), o *open shop* é um *job-shop* geral pelo qual não existe um fluxo padrão a ser seguido. A requisição das máquinas pelas tarefas ocorre de forma imprevisível, pois a produção não pode ser pré-estabelecida por tratar com produtos e mercados diversificados. No *open shop*, uma tarefa pode requerer processamentos pela mesma máquina mais do que uma vez em uma mesma seqüência de operação.

Em máquinas paralelas existe um só tipo de máquina ($m = 1$) com k máquinas idênticas, sendo que cada tarefa necessita de somente uma máquina. O modelo de máquinas paralelas pode ser entendido se supusermos que existam tarefas com operações simples simultaneamente disponíveis e que existam k máquinas idênticas para o processamento destas tarefas.

Em máquina única existe um só tipo de máquina e somente uma máquina deste tipo. Este tipo de disposição é estudado por motivos de simplificações do processo, pois às vezes existe uma máquina dominante com relação às demais e, conseqüentemente, os problemas de programação podem surgir nela. Por exemplo, uma máquina *gargalo* onde há formação de estoques intermediários. Neste caso, a programação terá que reduzir a fila de espera na máquina única para que o processo possa fluir melhor.

O *flow-shop* possui uma configuração de m máquinas distintas. O fluxo de trabalho é unidirecional (roteiro em fluxo), pelo qual as máquinas estão dispostas em série. Estas máquinas processarão um conjunto de n tarefas na mesma seqüência, sendo que nem todas as tarefas utilizam todas as máquinas. O *flow-shop* possui um fluxo padrão para todas as tarefas, ou seja, todas as tarefas terão de passar pelas mesmas máquinas.

No *flow-shop* permutacional, todas as máquinas são utilizadas pelas tarefas, requerendo uma operação em cada máquina (existem m operações sempre na mesma seqüência). A figura 2.2 mostra a diferença entre o *flow-shop* e o *flow-shop* permutacional.

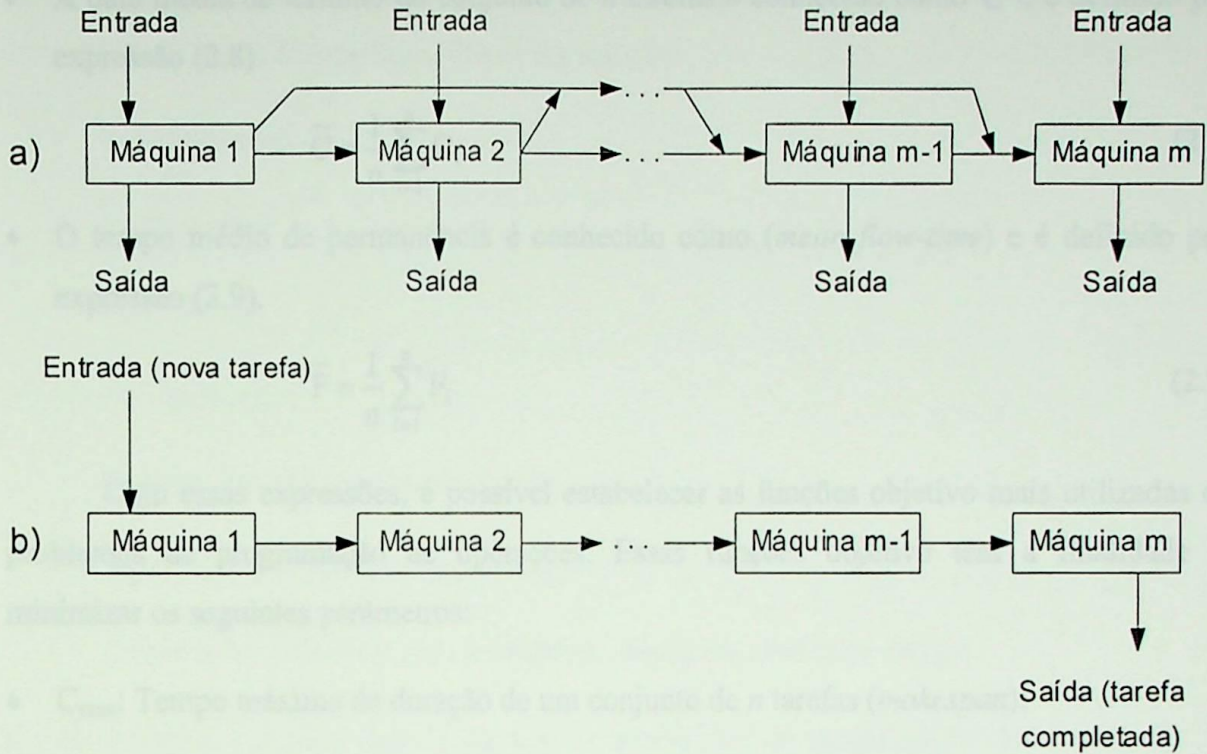


Figura 2.2 – Adaptada de BAKER (1974) - a) *Flow-shop* - b) *Flow-shop* permutacional

2.2.3 – Medidas de desempenho

Segundo MOCCELLIN (1994) as medidas de desempenho mais utilizadas em problemas de programação de operações são provenientes do tempo de espera (W_{ij}), definido anteriormente na expressão (2.3). São elas:

- Data de término da tarefa J_i , chamada de C_i e definida pela expressão (2.4)

$$C_i = r_i + \sum_{j=1}^{g_i} W_{ij} + \sum_{j=1}^{g_i} p_{ij} = r_i + W_i + p_i \quad (2.4)$$

- O tempo de fluxo total ou tempo de permanência total da tarefa J_i é conhecido como F_i e é definido pela expressão (2.5).

$$F_i = C_i - r_i \quad \text{ou} \quad F_i = W_i - p_i \quad (2.5)$$

- O atraso (*Lateness*) da tarefa J_i chamado de L_i e é definido pela expressão (2.6).

$$L_i = C_i - d_i \quad \text{ou} \quad L_i = F_i - a_i \quad (2.6)$$

- O atraso máximo (*tardiness*) na execução da tarefa J_i é conhecido como T_i e é definido pela expressão (2.7).

$$T_i = \max(0, L_i) \quad (2.7)$$

- A data média de término do conjunto de n tarefas é conhecido como \bar{C} e é definido pela expressão (2.8).

$$\bar{C} = \frac{1}{n} \sum_{i=1}^n C_i \quad (2.8)$$

- O tempo médio de permanência é conhecido como (*mean flow-time*) e é definido pela expressão (2.9).

$$\bar{F} = \frac{1}{n} \sum_{i=1}^n F_i \quad (2.9)$$

Com essas expressões, é possível estabelecer as funções objetivo mais utilizadas em problemas de programação de operações. Essas funções objetivo têm a finalidade de minimizar os seguintes parâmetros:

- C_{\max} : Tempo máximo de duração de um conjunto de n tarefas (*makespan*);
- F_{\max} : Tempo máximo de permanência de um conjunto de n tarefas;
- T_{\max} : Tempo máximo de atraso de um conjunto de n tarefas.

2.2.4 - Classificação dos Problemas de Programação

LÓPES, BARCIA e EYADA (1995) propõem que a classificação dos problemas de programação deverá considerar os fatores internos que influenciam uma organização através das requisições tecnológicas, determinadas principalmente pelo padrão de fluxo das tarefas nas máquinas, pelo número e tipos de máquinas disponíveis e pelo critério a ser utilizado para avaliar as alternativas de programação.

CONWAY, MAXWELL & MILLER (1967) sugerem um esquema de classificação baseado em quatro descritores $A/B/C/D$, onde:

- **A**: descreve o processo de chegada das tarefas. Se for um problema estático, ou seja, o conjunto de tarefas a serem executadas é conhecido, esse espaço é utilizado para identificar o número de tarefas. Se for um problema dinâmico, ou seja, as tarefas chegam na oficina intermitentemente, em datas que podem ser previstas somente com bases estatísticas, esse espaço é utilizado para identificar a distribuição de probabilidades dos tempos entre chegadas sucessivas.
- **B**: descreve o número de máquinas na oficina.

- **C**: descreve o padrão de fluxo entre as máquinas na oficina. Se **B** for igual a 1 (uma máquina), então **C** é omitido. Possíveis valores:

||: Máquina única;

F: *flow-shop*

J: *Job-shop*;

O: *Open-shop*;

F,perm: *flow-shop* permutacional;

k-parallel: *k* máquinas em paralelo;

J,k-parallel: *Job-shop* com *k* máquinas duplicadas em cada estágio.

- **D**: descreve o critério através do qual as alternativas de programação serão avaliadas, ou seja, identifica qual será a medida de desempenho utilizada no problema.

Por exemplo, se encontrarmos a seguinte notação:

$n/2/F/C_{\max}$: trata-se de um problema de programação de $n \geq 2$ tarefas em 2 máquinas, em regime de *flow-shop*, de modo a minimizar a duração total da programação, em um ambiente estático.

$n/m/J/F_{\max}$: trata-se de um problema de programação de $n \geq 2$ tarefas em $m \geq 2$ máquinas, com fluxos diferenciados (*job-shop*), de modo a minimizar o tempo de permanência máximo das tarefas na produção, em um ambiente estático.

2.3 – Sistemas de produção intermitentes

Como mencionado na introdução, será abordado, neste trabalho, os sistemas de produção ditos intermitentes. Segundo CONTADOR & CONTADOR (1997), em indústrias com essas características, geralmente se fabricam uma grande variedade de produtos utilizando os mesmos equipamentos, o que acarreta em variação na carga de trabalho de cada recurso produtivo (máquina, equipamento e pessoal). Portanto o grande objetivo da programação da produção é otimizar a ocupação desses recursos procurando maximizar a produção.

Dentre os diversos padrões de fluxo de produção possíveis para sistemas intermitentes, optou-se pelo *flow-shop* permutacional.

Segundo MOCCELLIN (1992), um caso típico de programação *flow-shop* permutacional, são instalações de manufatura onde as tarefas são movidas de uma máquina para outra através de algum equipamento de movimentação de materiais.

MOCCELLIN (1992), cita as principais condições de contorno para a definição deste problema:

- Os tempos de processamento das tarefas nas diversas máquinas são determinados e fixos;
- As tarefas têm a mesma data de liberação, a partir da qual qualquer uma pode ser programada e executada;
- Os tempos de preparação (*setup times*) das operações nas diversas máquinas são incluídos nos tempos de processamento e independem da seqüência de operações em cada máquina;
- As operações nas diversas máquinas, uma vez iniciadas não devem ser interrompidas.

A solução do problema de programação de operações *flow-shop* permutacional consiste em determinar dentre as $n!$ seqüências possíveis das tarefas, aquela que minimiza o intervalo de tempo entre o início de execução da primeira tarefa, na primeira máquina, e o término de execução da última tarefa, na última máquina, ou seja determinar a duração total da programação (*makespan*).

2.4 - Abordagens para o problema de programação

Segundo LÓPES, BARCIA e EYADA (1995), a grande maioria de procedimentos de programação, podem ser categorizados em dois principais grupos, baseados nas abordagens empregadas. O primeiro, e de longe a maior categoria, denominado de *procedimentos heurísticos* objetiva produzir programações viáveis boas, sendo aceitável se os resultados forem satisfatórios quanto ao esforço computacional. O segundo grupo consiste dos procedimentos que procuram produzir a programação melhor ou ótima. Estes são denominados de *procedimentos ótimos*, também chamados *exatos* ou *analíticos* porque geralmente envolvem alguma forma de programação matemática ou outro procedimento analítico mais rigoroso.

Quando a abordagem do problema se caracteriza por uma certa quantidade de tarefas em um conjunto de máquinas, tem-se uma complexidade de natureza combinatorial, também chamado de *NP-hard*. A resolução desse problema através de uma abordagem ótima, apenas se faz eficiente em casos de pequeno porte. Por exemplo, em um problema envolvendo 10 tarefas, o número de diferentes programações possíveis é de 3.628.800. MOCCELLIN (1992) afirma que não existe um algoritmo eficiente que fornece uma solução ótima, a não ser para o caso de 2 ou 3 máquinas como o algoritmo de JOHNSON (JOHNSON, 1954). Assim muitos métodos heurísticos têm sido propostos para a solução de problemas de programação de operações *flow-shop*.

2.4.1 – Alguns algoritmos heurísticos para programação de operações

A seguir, são citados alguns dos métodos heurísticos, específicos para a programação *flow-shop* permutacional, mais conhecidos e referenciados na literatura.

PALMER (1965) definiu um índice denominado *slope index*, a partir do qual se estabelece uma ordem de processamento das tarefas nas máquinas. Tal índice é calculado de forma que as tarefas que possuem tempos de processamento que tendem a crescer na seqüência das máquinas, devem receber maior prioridade na programação, ou seja, devem ocupar as primeiras posições na ordem de execução das tarefas.

CAMPBELL, DUDEK & SMITH (1970) apresentaram o algoritmo conhecido como CDS, o qual é basicamente uma extensão do algoritmo de JOHNSON (BAKER, 1974) para solução exata do problema com 2 máquinas.

HOLLAND (1975), desenvolveu um método chamado de *Algoritmos Genéticos*, que faz uma analogia com a evolução das espécies, onde o mais adaptado sobrevive em detrimento do mais fraco. Esse algoritmo baseia-se na teoria da evolução, que considera como fatores evolutivos as mutações, a recombinação genética, as migrações, o tamanho da população, o isolamento reprodutivo e a seleção natural. Na aplicação do algoritmo genético para a solução de problemas de programação de operações *flow-shop* permutacional, um cromossomo representa uma possível ordenação das tarefas e os genes representam as tarefas.

DANNENBRING (1977) desenvolveu um procedimento denominado *rapid access* (RA), o qual procura combinar as vantagens do *slop index* de PALMER e do método CDS, obtendo uma boa solução de maneira simples e rápida.

NAWAZ, ENSCORE JR & HAM (1983) desenvolveram um algoritmo, conhecido como NEH, baseado na hipótese de que as tarefas devem ter prioridades de programação diretamente proporcionais às somas dos seus tempos de processamento nas m máquinas.

WIDMER & HERTZ (1989) desenvolveram um método denominado SPIRIT que a partir de uma solução inicial, busca obter uma melhor solução, examinando uma série de outras seqüências de tarefas, obtidas através do conceito de *vizinhança*. A solução inicial desse método é obtida utilizando-se de uma analogia com o problema do caixeiro-viajante. A segunda fase do algoritmo, de melhoria da solução inicial, utiliza uma técnica denominada *busca tabu* (GLOVER, 1989).

OSMAN & POTTS (1989) propuseram alguns métodos fundamentados em uma técnica denominada *Simulated Annealing*, que tem sua origem na física estatística e baseia-se na idéia de se aplicar o princípio de tratamento térmico (aquecimento e gradativo resfriamento) para a solução de problemas de natureza combinatorial. Os primeiros a utilizarem essa técnica foram KIRKPATRICK, GELATT & VECCHI (1983). Esses métodos são algoritmos onde, a partir de uma seqüência inicial das tarefas, são examinados uma série de outra seqüências vizinhas, à semelhança do algoritmo SPIRIT.

Para o desenvolvimento da metodologia proposta neste trabalho escolheu-se o algoritmo desenvolvido por NAWAZ, ENSCORE JR & HAM (1983), conhecido por NEH. Optou-se por este algoritmo pois, segundo MOCCELLIN & NAGANO (1999), o NEH é o melhor algoritmo heurístico reportado na literatura, onde é possível obter soluções de boa qualidade com pouco esforço computacional.

Por esta razão, o algoritmo NEH será ideal para o desenvolvimento da nova abordagem em problemas de programação de operações com considerações de incertezas, apresentado no *Capítulo 3 – Metodologia proposta*.

2.4.2 – Alguns algoritmos ótimos para programação de operações

Já foi mencionado anteriormente que os algoritmos ótimos para a programação de operações são viáveis até no máximo 3 máquinas. Acima de 3 máquinas é comum se optar pelos algoritmos ditos como heurísticos. A seguir, são citados alguns dos métodos ótimos, referenciados na literatura.

JOHNSON (1954) desenvolveu um algoritmo para problemas de programação em 2 máquinas que possui o objetivo de minimizar o tempo total de produção (*makespan*). Esta programação ótima é conhecida como programação permutacional. O autor apresenta, também, uma extensão do seu problema para o caso de 3 máquinas.

IGNALL & SCHRAGE (1965) apresentaram uma solução para o problema envolvendo 3 máquinas. A técnica utilizada pelos autores é conhecida como método *branch and bound*. *Branch* significa ramificação, ou seja, processo de partição de um problema em dois ou mais subproblemas. *Bound* significa limitação, ou seja, processo de cálculo de um valor limite para a medida de desempenho adotada, referente a solução ótima de um dado subproblema.

Para problemas de sequenciamento de operações em 1 máquina, temos:

- SMITH (1956) apresentou um algoritmo que buscava minimizar o tempo médio de permanência de cada tarefa na produção através da ordenação crescente dos tempo de processamento de cada atividade. Neste algoritmo considerou-se que o número de tarefas programadas com atraso era zero.
- MOORE (1968) desenvolveu um algoritmo que buscava minimizar o número de tarefas a serem programadas com atraso.
- HECK & ROBERTS (1972) apresentaram um algoritmo que pode ser considerado como uma extensão ao problema de SMITH (1956). Neste algoritmo, os autores consideram as tarefas programadas com atraso e buscam minimizá-las.
- SIDNEY (1973) apresentou um algoritmo que é a extensão do algoritmo de MOORE (1968), onde passou a considerar que algumas tarefas não devem ter atraso.
- EMMONS (1975) propõe a solução do problema de SIDNEY (1973) através de um algoritmo tipo *Branch and Bound*.

Para o desenvolvimento da metodologia proposta neste trabalho escolheu-se o algoritmo desenvolvido por JOHNSON (1954) para o caso de 2 máquinas. Optou-se por este método pois, para o problema real que será proposto no *Capítulo 4 – Estudo de caso*, este algoritmo se adequará perfeitamente.

2.4.3 – Algoritmo heurístico NEH

Neste algoritmo somente após o término do processamento de uma tarefa sobre um determinado equipamento, é que esta será liberada para ser processada em um equipamento posterior. Segundo NAWAS, ENSCORE e HAM (1983), este método baseia-se no tempo total gasto na execução de uma tarefa, onde tarefas que necessitam de maior tempo para serem processadas, recebem prioridade sobre as que requerem menores tempo.

Este algoritmo classifica-se como $n/m/F,perm/C_{max}$ onde a solução é obtida através de uma abordagem heurística, cujo objetivo é produzir um programa viável, com resultado próximo do ótimo. O algoritmo NEH é detalhado na figura 2.3 abaixo.

Passo 1:	Calcule, para cada tarefa, a soma dos tempos de processamento em todas as máquinas.
Passo 2:	Ordene as n tarefas de acordo com os valores decrescentes das somas dos tempos de processamento.
Passo 3:	Selecione as duas primeiras tarefas da ordenação, seqüenciando-as de maneira a minimizar a duração total da programação, considerando-se somente essas duas tarefas.
Passo 4:	Para k variando de 3 a n , faça: <ul style="list-style-type: none"> • Selecione a tarefa que ocupa a k-ésima posição na ordenação obtida no passo (2); • Examine as k possibilidades de acrescentar a tarefa na seqüência até então obtida, adotando aquela que leva a uma menor duração total da programação parcial.

Figura 2.3 – Algoritmo NEH

O número total de iterações desse algoritmo é obtido pela equação (2.10), onde n é o número de tarefas a serem processadas.

$$\frac{n(n+1)}{2} - 1 \quad (2.10)$$

Percebe-se desta maneira que o número total de iterações para atingir a solução final se reduz bastante, uma vez que se fosse testado todas as possíveis combinações teríamos $n!$ iterações.

Para exemplificar o algoritmo comentado, tem-se o seguinte problema envolvendo 3 tarefas (peças) e 4 equipamentos (máquinas). A matriz com os tempos de processamento das operações é mostrada na figura 2.4 a seguir.

	Máquinas (M _j)			
Peças (J _i)	M1	M2	M3	M4
J1	9	4	5	8
J2	9	3	10	1
J3	5	9	8	10

Figura 2.4 - Matriz com os tempos de processamento das operações

Conforme detalhado no algoritmo, como primeiro passo, calcula-se os tempo totais de processamento para cada peça:

$$p_1 = 9 + 4 + 5 + 8 = 26$$

$$p_2 = 9 + 3 + 10 + 1 = 23$$

$$p_3 = 5 + 9 + 8 + 10 = 32$$

A seguir, os tempos são colocados em ordem decrescente, resultando a ordem J3, J1, J2. Com isto, calcula-se a melhor ordenação parcial para as peças J1 e J3, conforme mostrado nas figura 2.5 e 2.6 a seguir.

	Máquinas (M _j)			
Peças (J _i)	M1	M2	M3	M4
J3	5/5	9/14	8/22	10/32
J1	9/14	4/18	5/27	8/40

Figura 2.5 - Tempo do fluxo de operação para a ordenação parcial J3-J1

	Máquinas (M _j)			
Peças (J _i)	M1	M2	M3	M4
J1	9/9	4/13	5/18	8/26
J3	5/14	9/23	8/31	10/41

Figura 2.6 - Tempo do fluxo de operação para a ordenação parcial J1-J3

Pode-se observar que a melhor ordenação parcial obtida anteriormente foi a das peças J3-J1. Esta ordenação possui o tempo de fluxo de operação é igual a 40 u.t. (unidades de tempo). Nos próximos passos a posição relativa das peças J1 e J3 deverá ser, então, primeiro a peça J3 e depois a peça J1.

Seguindo o algoritmo, seleciona-se a 3ª peça da lista de ordem decrescente de tempo, que é a peça número 2. Calcula-se, então, a melhor ordenação parcial colocando a peça J2 nas 3 posições possíveis. Estas ordenações são demonstradas nas figuras 2.7, 2.8 e 2.9 a seguir.

Peças (J _i)	Máquinas (M _j)			
	M1	M2	M3	M4
J3	5/5	9/14	8/22	10/32
J1	9/14	4/18	5/27	8/40
J2	9/23	3/26	10/37	1/41

Figura 2.7 – Tempo do fluxo de operação para a ordenação parcial J3-J1-J2

Peças (J _i)	Máquinas (M _j)			
	M1	M2	M3	M4
J3	5/5	9/14	8/22	10/32
J2	9/14	3/17	10/32	1/33
J1	9/23	4/27	5/37	8/45

Figura 2.8 – Tempo do fluxo de operação para a ordenação parcial J3-J2-J1

Peças (J _i)	Máquinas (M _j)			
	M1	M2	M3	M4
J2	9/9	3/12	10/22	1/23
J3	5/14	9/23	8/31	10/41
J1	9/23	4/27	5/36	8/49

Figura 2.9- Tempo do fluxo de operação para a ordenação parcial J2-J3-J1

Pelos cálculos obtidos no desenvolvimento do algoritmo acima, observa-se que a melhor programação de operações das peças é a J3-J1-J2, que determina o menor tempo de fluxo de operação igual a 41 u.t..

2.4.4 – Algoritmo ótimo de JOHNSON

Deve-se considerar na formulação desse problema a seguinte regra: se, na ordenação ótima a tarefa i precede a tarefa j então teremos:

$$\min \{ p_{i1}, p_{j2} \} \leq \min \{ p_{i2}, p_{j1} \} \tag{2.11}$$

Onde a expressão (2.11) é conhecida como regra de Johnson (BAKER, 1974).

A ordenação ótima é obtida através da manipulação dessa expressão, onde o algoritmo testa todas as tarefas identificando suas posições dentro da ordenação. O algoritmo de JOHNSON é detalhado na figura 2.10 a seguir.

Passo 1	Determine o $\min \{ p_{i1}, p_{i2} \}$, com $i = 1, 2, 3, \dots, n$.
Passo 2 a	Se o mínimo tempo de processamento refere-se a uma operação a ser processada na máquina 1, coloque a tarefa J_i na <i>primeira</i> posição disponível na ordem das tarefas. Vá para o passo 3.
Passo 2 b	Se o mínimo tempo de processamento refere-se a uma operação a ser processada na máquina 2, coloque a tarefa J_i na <i>última</i> posição disponível na ordem das tarefas. Vá para o passo 3.
Passo 3	Remova a tarefa designada do conjunto de n tarefas e volte ao passo 1, até que todas as posições na ordem das tarefas sejam ocupadas (os desempates podem ser feitos arbitrariamente).

Figura 2.10 – Algoritmo de JOHNSON

Para exemplificar o algoritmo comentado, tem-se o seguinte problema envolvendo 5 tarefas (peças) e 2 equipamentos (máquinas). A matriz com os tempos de processamento das operações é mostrada na figura 2.11 a seguir.

Tarefas (J_i)	Máquinas (M_j)	
	M1	M2
J1	9	7
J2	2	7
J3	9	3
J4	10	5
J5	4	8

Figura 2.11 – Matriz com os tempos de processamento das operações em 2 máquinas

Conforme detalhado no algoritmo, como primeiro passo, descobre-se o menor tempo de processamento das tarefas entre as máquinas 1 e 2. Esse menor tempo é o $p_{21} = 2$ (tempo de processamento da tarefa 2 na máquina 1). Essa tarefa é retirada da tabela e colocada como primeiro elemento da ordenação ótima $S = (2, \dots)$.

O próximo menor tempo é o $p_{32} = 3$ (tempo de processamento da tarefa 3 na máquina 2). Essa tarefa é retirada da tabela e colocada como último elemento da ordenação ótima $S = (2, \dots, 3)$.

O próximo menor tempo é o $p_{51} = 4$. A tarefa 5 ocupa, agora, a segunda posição na ordenação ótima $S = (2, 5, \dots, 3)$.

A seguir, temos que o menor tempo é $p_{42} = 5$. A tarefa 4 ocupará a quarta posição na ordenação ótima $S = (2, 5, \dots, 4, 3)$

O último tempo de processamento é $p_{12} = 7$. A tarefa 1 ocupará a posição que resta na ordenação ótima, gerando o resultado $S = (2, 5, 1, 4, 3)$.

A solução para este problema também pode ser vista pela figura 2.12 a seguir.

Estágio	Tarefas ainda não programadas	Mínimo p_{ij}	Designação	Ordenação parcial
1	J_1, J_2, J_3, J_4, J_5	t_{21}	$J_2 = J_{[1]}$	J_2
2	J_1, J_3, J_4, J_5	t_{32}	$J_3 = J_{[5]}$	$J_2 - - - J_3$
3	J_1, J_4, J_5	t_{51}	$J_5 = J_{[2]}$	$J_2, J_5 - - J_3$
4	J_1, J_4	t_{42}	$J_4 = J_{[4]}$	$J_2, J_5 - J_4, J_3$
5	J_1	t_{12}	$J_1 = J_{[3]}$	J_2, J_5, J_1, J_4, J_3

Figura 2.12 – Solução do problema pelo algoritmo de JOHNSON

Para a ordenação obtida, o tempo total para a execução das atividade (*makespan*), obtido pelo gráfico de *Gantt* mostrado na figura 2.13, será igual a 37 u.t. (unidade de tempo). Neste gráfico atribuiu-se a cor azul para a peça 2, a cor vermelha para a peça 5, a cor amarela para a peça 1, a cor rosa para a peça 4 e a cor verde para a peça 3.

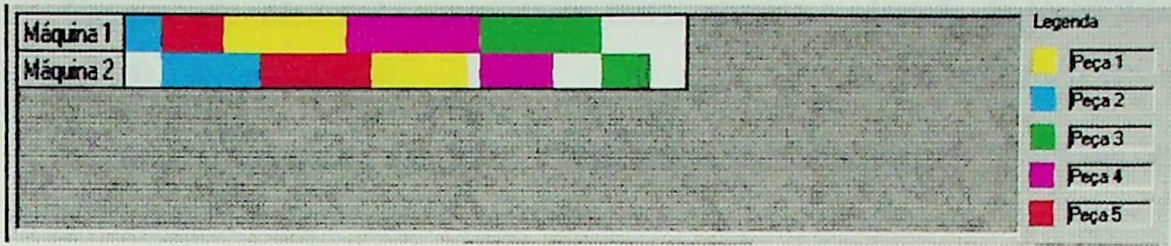


Figura 2.13 – Gráfico de *Gantt* para o problema

2.5 – Considerações sobre incertezas

A maioria dos processos industriais são complexos devido ao número de variáveis envolvidas e a dificuldade de quantificação destas variáveis. Por vezes, diversas destas grandezas têm seus valores estabelecidos através de processos subjetivos, sendo classificados por meios de valores lingüísticos.

Sob tais circunstâncias, a representação destas grandezas, através da lógica *fuzzy*, têm provado ser uma ferramenta efetiva para levar em consideração o impreciso (MONTEVECHI, 1995).

A lógica *fuzzy* é um meio de aproximar a precisão da matemática clássica e a inexatidão do mundo real. Esta teoria consegue manipular e operar quantidades exatas e inexatas, quantificadas através de valores lingüísticos.

2.5.1 – Lógica fuzzy

Proposta por ZADEH (1965), a lógica *fuzzy* passou a ser classificada como um novo ramo da lógica matemática. Nesta lógica, procura-se representar o modo como o ser humano realiza suas avaliações sobre um determinado assunto.

GOMIDE & GUDWIN (1994) afirmam que com o aumento da complexidade de um problema, a habilidade do ser humano de criar declarações precisas sobre um sistema diminui, o que torna mais difícil uma representação da realidade de forma eficiente.

Como alertam MORITA, OKA & WATANABE (1993), as escalas tradicionais muitas vezes não são suficientes para expressar o julgamento humano. Existem ocasiões em que é necessário mais que números para expressar o que queremos, sentimos e julgamos. Assim, compreender a teoria *fuzzy* nos permite representar modelos mais realistas da natureza humana sem alterar o rigor numérico exigido pelo computador, adequando-se a problemas complexos, onde a descrição ou o uso do pensamento intuitivo pelo homem é fundamental.

Um exemplo clássico de representação fuzzy, extraído e adaptado de McNEILL & THRO (1994) é descrito a seguir.

Se, ao olhar-mos para a figura 2.14, fizéssemos a seguinte pergunta: “Este é um cesto de bolinhas azuis?”, a resposta, óbvia, seria *sim*.

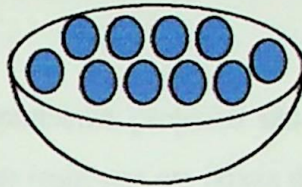


Figura 2.14 – Cesto com bolinhas azuis

Porém, ao olhar-mos para figura 2.15, a resposta para esta mesma pergunta seria obviamente *não*.

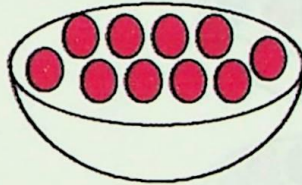


Figura 2.15 – Cesto com bolinhas vermelhas

E qual seria a resposta certa para esta mesma pergunta olhando-se para os cestos da figura 2.16 mostrados a seguir?

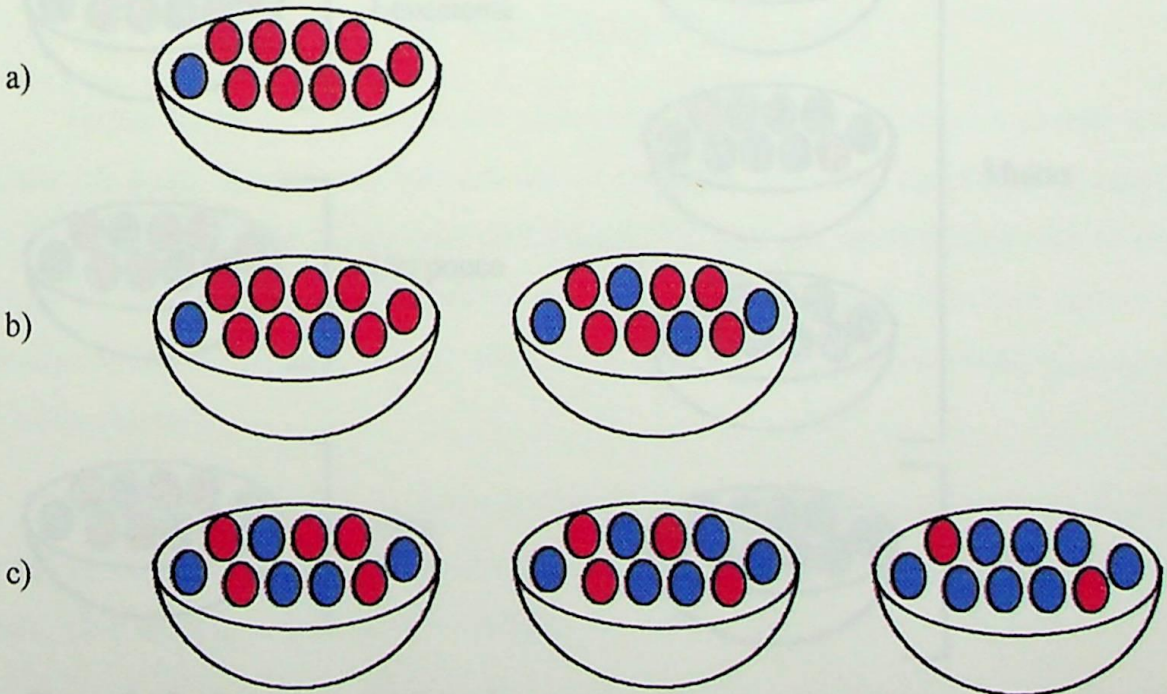


Figura 2.16 – Cestos com bolinhas azuis e vermelhas misturadas

Ao olharmos para os cestos mostrados na figura 2.16, a resposta para a pergunta: “*Este é um cesto de bolinhas azuis?*”, não poderia ser simplesmente sim ou não, pois estas respostas não expressariam, com detalhes, as informações existentes no problema.

Uma resposta mais adequada a este problema seria, para a figura 2.16 a): *contém uma bolinha azul*. Para a figura 2.16 b): *contém algumas bolinhas azuis*. E, para a figura 2.16 c): *contém muitas bolinhas azuis*.

As respostas atribuídas ao problema proposto são conhecidas como representações *fuzzy* do problema. Definindo-se estas respostas em faixas de valores, teremos para a pergunta “*Este é um cesto de bolinhas azuis?*” as seguintes representações *fuzzy* mostradas na figura 2.17 a seguir.

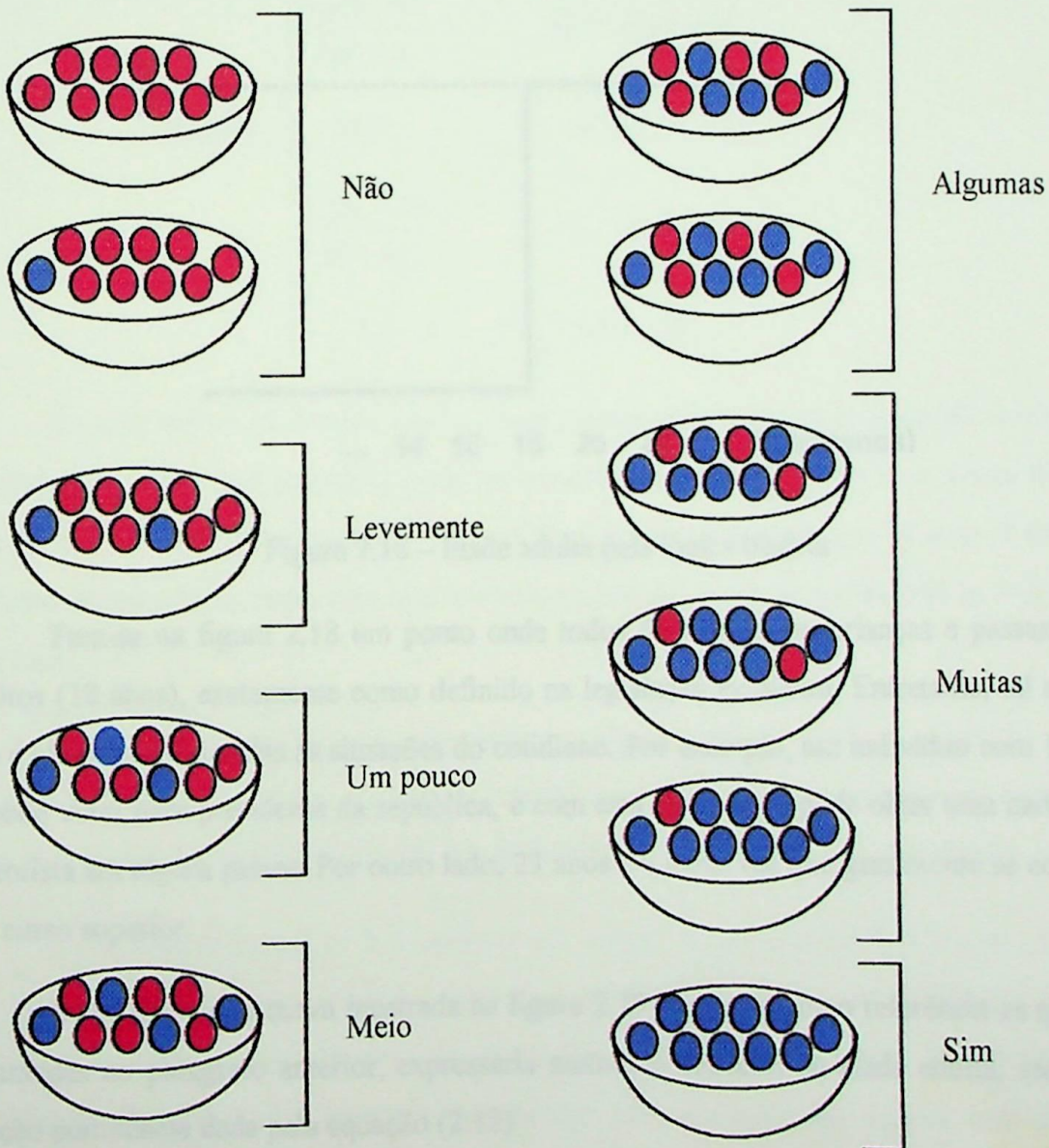


Figura 2.17 – Representação *fuzzy* da pergunta: “Este é um cesto de bolinhas azuis?”

É esta representação em faixas de valores, segundo critérios lingüísticos a grande responsável por alavancar a utilização da teoria fuzzy. Pois, segundo GOMIDE & GUDWIN (1994) nos sistemas lógicos clássicos existem somente os quantificadores existenciais (\exists) e universais (\forall) enquanto a lógica fuzzy admite, uma ampla variedade de quantificadores, por exemplo: pouco, vários, usualmente, freqüentemente, em torno de cinco, etc.

Na lógica binária existem duas únicas situações possíveis: 0 ou 1, que significam falso ou verdadeiro respectivamente. Na lógica *fuzzy*, porém, trabalha-se com um intervalo de transição entre estes valores (0 e 1) denominado *função de pertinência* (μ).

As figuras 2.19 e 2.20 ilustram o conceito de pertinência, através de uma mesma situação abordada pela lógica binária e pela lógica *fuzzy*: *a idade adulta*.

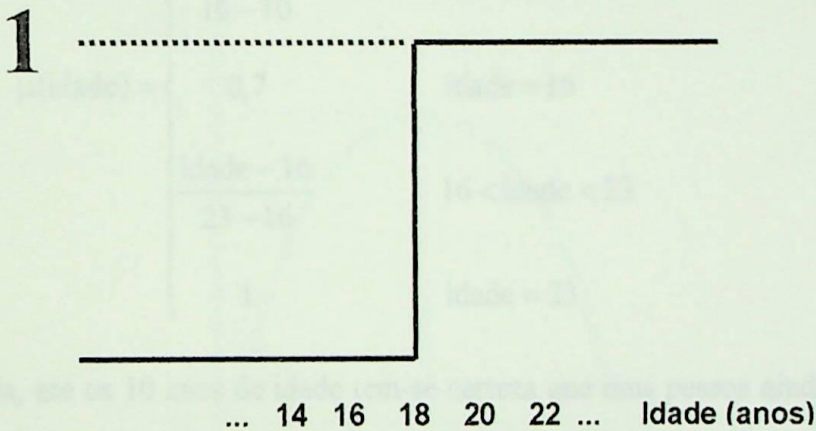


Figura 2.18 – Idade adulta pela lógica binária

Tem-se na figura 2.18 um ponto onde todos deixam de ser crianças e passam a ser adultos (18 anos), exatamente como definido na legislação brasileira. Entretanto, tal aspecto não é observado em todas as situações do cotidiano. Por exemplo, um indivíduo com 16 anos já pode votar para presidente da república, e com esta idade já se pode obter uma carteira de motorista em alguns países. Por outro lado, 23 anos é a idade em que geralmente se completa um curso superior.

Desta maneira a curva mostrada na figura 2.19, tomando como referência as questões levantadas no parágrafo anterior, expressaria melhor o conceito de idade adulta, sendo sua função pertinência dada pela equação (2.12).

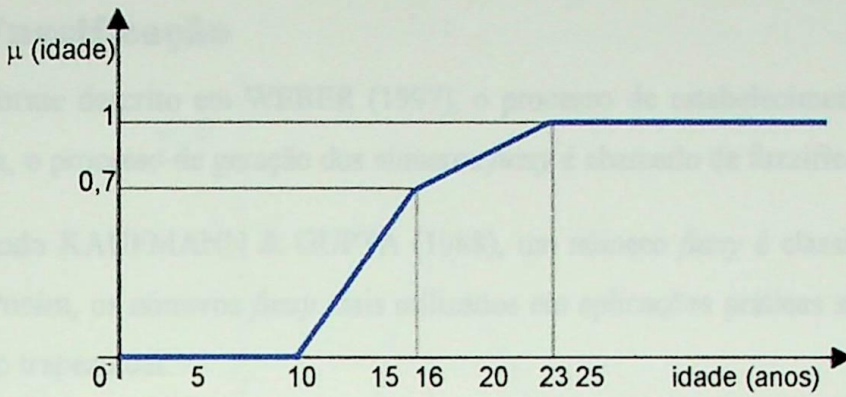


Figura 2.19 - Idade adulta pela lógica *fuzzy*

$$\mu(\text{idade}) = \begin{cases} 0 & \text{idade} \leq 10 \\ \frac{\text{idade} - 10}{16 - 10} & 10 < \text{idade} < 16 \\ 0,7 & \text{idade} = 16 \\ \frac{\text{idade} - 16}{23 - 16} & 16 < \text{idade} < 23 \\ 1 & \text{idade} = 23 \end{cases} \quad (2.12)$$

Ou seja, até os 10 anos de idade tem-se certeza que uma pessoa ainda não é adulta. A partir dos 23 anos, uma pessoa já pode ser considerada adulta, sob qualquer ponto de vista. Dos 10 aos 23 anos, a pessoa passa por uma fase de transição, onde aos 16 anos já está bem próxima de ser adulta, razão pela qual atribuiu-se um alto valor de pertinência ($\mu = 0,7$) para esta idade.

É importante ressaltar que estes valores poderão variar ao ouvir-se a opinião de uma outra pessoa. Assim, surge o conceito de *especialista*: uma pessoa que realmente entende do assunto em questão, e que será ouvida no estabelecimento dos valores de pertinência.

Desta mesma forma, pode-se atribuir aos tempos de processamento de cada atividade, em problemas de programação de operações, uma representação *fuzzy*. Sendo atribuído pertinências maiores, aos tempos de execução da tarefa com maior possibilidade de ocorrência. E pertinências menores, aos tempos de execução das tarefas com menores possibilidades de ocorrência.

2.5.2 – Fuzzificação

Conforme descrito em WEBER (1997), o processo de estabelecimento dos números *fuzzy*, ou seja, o processo de geração dos números *fuzzy* é chamado de fuzzificação.

Segundo KAUFMANN & GUPTA (1988), um número *fuzzy* é classificado quanto à sua forma. Porém, os números *fuzzy* mais utilizados em aplicações práticas são o senoidal, o triangular e o trapezoidal.

COX (1994), afirma que os números *fuzzy* senoidais são mais utilizados ao se estudar situações onde existe incerteza sob a dispersão de uma característica em torno de um valor central. A figura 2.20 mostra um exemplo representando, através de um número *fuzzy* senoidal, a situação de se estimar quais são os números *em torno de cinco*.

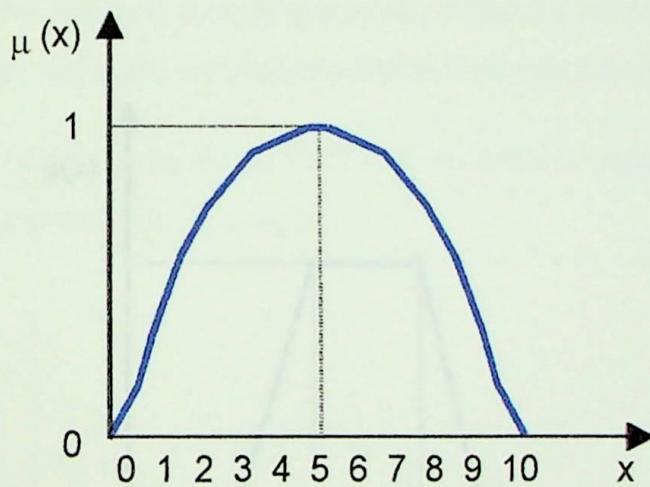


Figura 2.20 – Número *fuzzy* senoidal para “em torno de cinco”

Um número *fuzzy* senoidal se assemelha a uma distribuição de probabilidades normal e por isso é utilizado em alguns casos. Sua diferença principal com a distribuição de probabilidades é que a área sob a curva não precisa ser necessariamente igual a 1.

O número *fuzzy* triangular (TFN – *Triangular Fuzzy Number*) e trapezoidal (TrFN – *Trapezoidal Fuzzy Number*) são, freqüentemente, representados na forma vetorial por (a_1, a_2, a_3) para um TFN e por (a_1, a_2, a_3, a_4) para um TrFN. A figura 2.21 e 2.22 mostram respectivamente este dois números *fuzzy*.

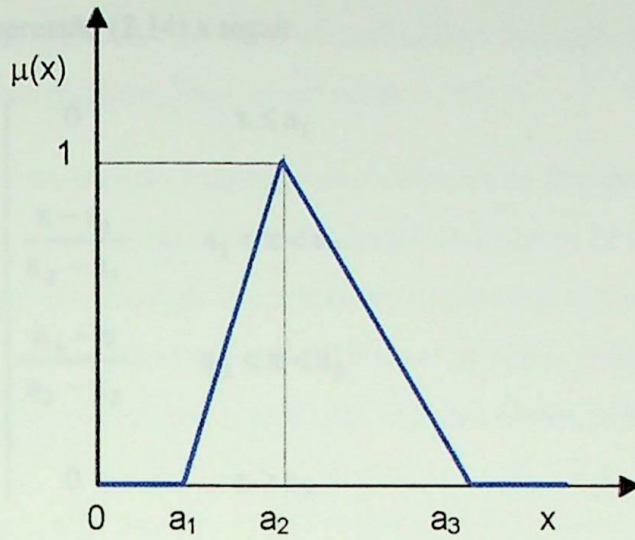


Figura 2.21 – Número *fuzzy* triangular

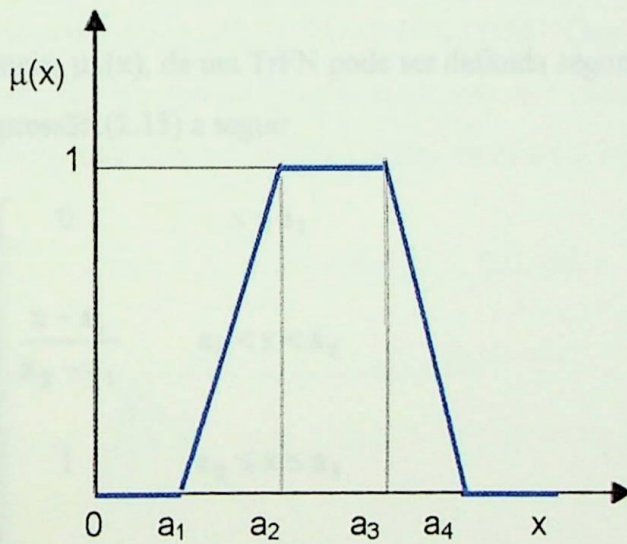


Figura 2.22 – Número *fuzzy* trapezoidal

Pode-se observar que um TFN é, na verdade, considerado um caso particular do TrFN, em que $a_2 = a_3$.

Para um TFN, mostrado na figura 2.21, atribui-se ao valor a_2 a pertinência 1, ou seja, existe possibilidade total de ocorrer a_2 . A valores menores que a_1 e maiores que a_3 são atribuídos pertinência zero, ou seja, não existe possibilidade de ocorrer estes valores. A valores intermediários entre $[a_1, a_2]$ e $[a_2, a_3]$ são atribuídos pertinências entre 0 e 1. O valor a_1 é conhecido como valor otimista do TFN, pois este valor é o menor tempo possível de ocorrência. Já o valor a_3 , é conhecido como valor pessimista, pois este valor é o maior tempo possível de ocorrência para o TFN.

A função pertinência, $\mu_A(x)$, de um TFN pode ser definida segundo, KAUFMANN & GUPTA (1988), pela expressão (2.14) a seguir.

$$\mu_A(x) = \begin{cases} 0 & x \leq a_1 \\ \frac{x - a_1}{a_2 - a_1} & a_1 < x < a_2 \\ \frac{a_3 - x}{a_3 - a_2} & a_2 < x < a_3 \\ 0 & x \geq a_3 \end{cases} \quad (2.14)$$

Já para um TrFN, mostrado na figura 2.22, atribui-se pertinência 1 aos valores entre a_2 e a_3 . A valores menores que a_1 e maiores que a_4 são atribuídos pertinência zero. A valores intermediários entre $[a_1, a_2]$ e $[a_3, a_4]$ são atribuídos pertinências entre 0 e 1.

A função pertinência, $\mu_A(x)$, de um TrFN pode ser definida segundo, KAUFMANN & GUPTA (1988), pela expressão (2.15) a seguir.

$$\mu_A(x) = \begin{cases} 0 & x \leq a_1 \\ \frac{x - a_1}{a_2 - a_1} & a_1 < x < a_2 \\ 1 & a_2 \leq x \leq a_3 \\ \frac{a_4 - x}{a_4 - a_3} & a_3 < x < a_4 \\ 0 & x \geq a_4 \end{cases} \quad (2.15)$$

Ambos, TFN e TrFN, são muito utilizados em ambientes controlados por engenheiros (COX, 1994). Porém, segundo KAUFMANN e GUPTA (1988), os TFNs são mais encontrados nos estudos de problemas de engenharia e administração, pois são de fácil fuzzificação e desfuzzificação. Desta maneira, optou-se em trabalhar com os números fuzzy triangulares (TFN) no desenvolvimento desta dissertação.

2.5.3 – Desfuzzificação

Conforme descrito por COX (1995), desfuzzificação (*defuzzyfication*) é o processo de conversão de um ou vários números *fuzzy* em um número real.

Existem vários processos de desfuzzificação descritos na literatura. COX (1994) cita, entre outros, o centróide, a maior altura e a média do intervalo de maior altura. KAUFMANN e GUPTA (1988) propõem o método do representativo ordinário (*Ordinary representative*) de um número *fuzzy*, onde ao considerarmos que o TFN que se deseja realizar a desfuzzificação seja $A = (a_1, a_2, a_3)$, o seu representativo ordinário (\hat{A}) será obtido pela expressão (2.16), a seguir.

$$\hat{A} = \frac{a_1 + 2a_2 + a_3}{4} \quad (2.16)$$

A figura 2.23 traz o exemplo de desfuzzificação pelos métodos representativo ordinário, centróide e maior altura extraído de SALOMON (1998). Como o TFN do exemplo $A = (1, 2, 6)$ não é simétrico, sua desfuzzificação pelos métodos citados obteve resultados diferentes, embora próximos entre si como mostra a matriz da figura 2.24.

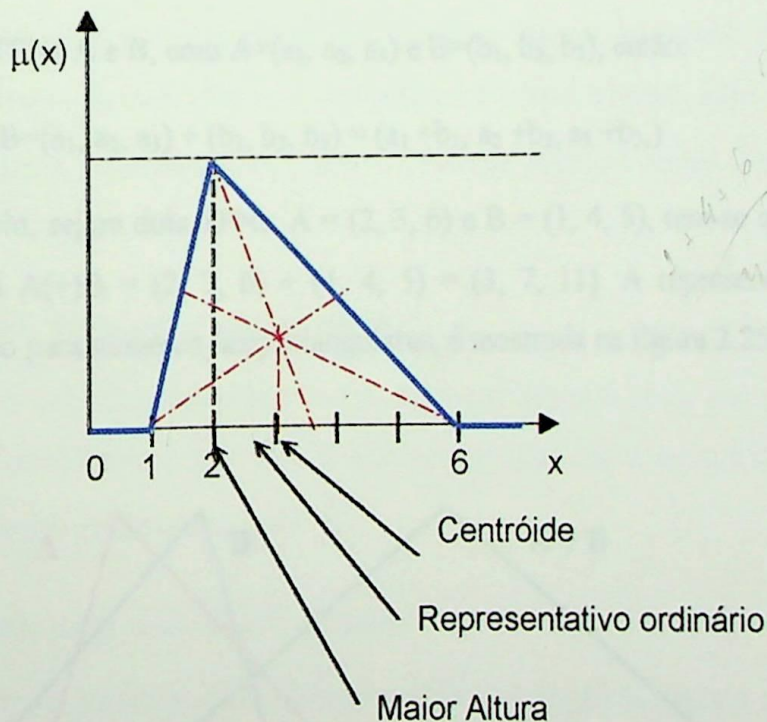


Figura 2.23 – Desfuzzificação do TFN (1, 2, 6)

<i>Método</i>	<i>Resultado</i>
Centróide	3,10
Representativo ordinário	2,75
Maior altura	2,00

Figura 2.24 – Matriz com o resultado da desfuzzificação do TFN (1, 2, 6)

Pela figura 2.24 pode-se perceber que os métodos do centróide e representativo ordinário são, nesta ordem, mais sensíveis à assimetria do TFN. O método representativo ordinário foi utilizado na metodologia proposta pois segundo SALOMON (1998), este método apresenta maior sensibilidade à assimetria que o método maior altura e ao fato de ser mais simples de se calcular que o centróide.

2.5.4 – Operações algébricas com números fuzzy triangulares

Dentre as diversas operações algébricas existente para os TFNs, somente é utilizado nos algoritmos abordados neste trabalho, a operação de adição, sendo esta operação definida pela expressão (2.17) a seguir.

Seja dois TFNs A e B, com $A=(a_1, a_2, a_3)$ e $B=(b_1, b_2, b_3)$, então:

$$A+B=(a_1, a_2, a_3) + (b_1, b_2, b_3) = (a_1 + b_1, a_2 + b_2, a_3 + b_3,) \quad (2.17)$$

Por exemplo, sejam dois TFNs: $A = (2, 3, 6)$ e $B = (1, 4, 5)$, tem-se que a soma entre estes valores será $A(+)B = (2, 3, 6) + (1, 4, 5) = (3, 7, 11)$. A representação gráfica da operação de adição para números *fuzzy* triangulares, é mostrada na figura 2.25.

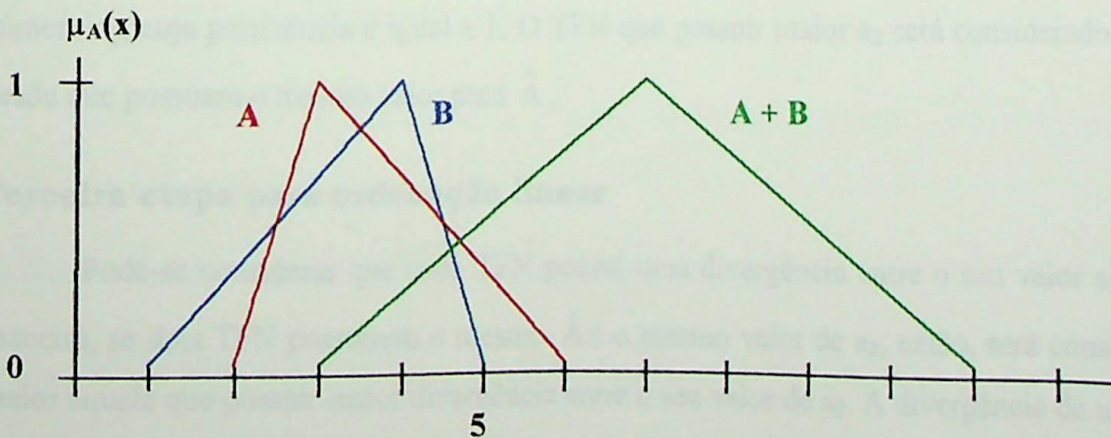


Figura 2.25 - Representação gráfica da operação de adição entre dois TFNs

2.5.5 – Comparação entre números fuzzy triangulares

A comparação entre os TFNs é outra é outra operação realizada em algoritmos de programação de operações com incertezas, pois é necessário, entre vários números triangulares *fuzzy*, definir por exemplo, quais são os maiores, e desta forma, estabelecer uma ordenação linear decrescente, ou seja, estabelecer um *ranking* entre esses números.

Segundo FORTEMPS & ROUBENS (1996) deste que BASS & KWAKERNAAK (1977) escreveram sobre o ordenação de números *fuzzy*, muitos trabalhos têm sido publicados sobre a comparação e ordenação desses números. Neste trabalho, optou-se pelo critério de ordenação proposto por KAUFMANN & GUPTA (1988), pois o método proposto por estes autores é específico para números *fuzzy* triangulares.

A seguir serão mostradas 3 etapas para a ordenação de TFNs. Se a primeira etapa não propiciar uma única ordem linear, a segunda etapa é utilizada, se esta também não for suficiente, utiliza-se, então, a terceira etapa, a fim de se obter uma seqüência ordenada entre os números *fuzzy*.

Primeira etapa para ordenação linear

A primeira etapa consiste em definir, para um número triangular fuzzy, um número real que o represente, ou seja, efetuar a desfuzzificação deste número. Essa desfuzzificação, é calculada conforme a expressão (2.16), onde o TFN que possuir o maior \hat{A} será considerado o maior número.

Segunda etapa para ordenação linear

Sendo um TFN definido como $A=(a_1, a_2, a_3)$, a segunda etapa usa como parâmetro o número a_2 , cuja pertinência é igual a 1. O TFN que possuir maior a_2 será considerado maior, desde que possuam o mesmo valor para \hat{A} .

Terceira etapa para ordenação linear

Pode-se considerar que cada TFN possui uma divergência entre o seu valor a_2 , desta maneira, se dois TFN possuírem o mesmo \hat{A} e o mesmo valor de a_2 , então, será considerado maior aquele que possuir maior divergência entre o seu valor de a_2 . A divergência de um TFN é calculada fazendo-se $a_3 - a_1$.

A fim de exemplificar o método de ordenação entre números triangulares *fuzzy* mostrado anteriormente, considere os seguintes TFNs.

$$A = (-3, 5, 11), B = (-5, 10, 11), C = (-2, 6, 8), D = (0, 6, 7), E = (-1, 6, 7)$$

Pela primeira etapa, tem-se:

$$\hat{A} = \frac{-3+5+11}{4} = 4.5$$

$$\hat{B} = \frac{-5+10+11}{4} = 6.5$$

$$\hat{C} = \frac{-2+6+8}{4} = 4.5$$

$$\hat{D} = \frac{0+6+7}{4} = 4.75$$

$$\hat{E} = \frac{-1+6+7}{4} = 4.5$$

Com a primeira etapa, estabelece-se 3 classes:

Classe 1: A, C e E

Classe 2: D

Classe 3: B

Aplicando-se, agora, a segunda etapa para a classe 1.

Para A = (-3, 5, 11) temos $a_2 = 5$

Para C = (-2, 6, 8) temos $a_2 = 6$

Para E = (-1, 6, 7) temos $a_2 = 6$

Com isso, defini-se as seguintes sub-classes

Sub-Classe 1: A

Sub-Classe 2: C e E

Sub-Classe 3: D

Sub-Classe 4: B

Aplicando a terceira etapa para a sub-classe 2, tem-se:

Para C = (-2, 6, 8) a divergência = $8 + 2 = 10$

Para E = (-1, 6, 7) a divergência = $7 + 1 = 8$

Com isso, defini-se as seguintes sub-sub-classes:

Sub-Sub-Classe 1: A

Sub-Sub-Classe 2: E

Sub-Sub-Classe 3: C

Sub-Sub-Classe 4: D

Sub-Sub-Classe 5: B

Desta maneira, é possível obter a ordenação linear decrescente dos TFNs em questão, mostrada a seguir.

$$A < E < C < D < B$$

2.5.6 – Incerteza em algoritmos de programação de operações

Conforme mencionado na introdução, em algoritmos de programação de operações, a principal variável para os cálculos, é o tempo de processamento de cada operação. Este tempo de processamento estimado pode variar dependendo do operador que executará a atividade, conforme a máquina que será utilizada, do ferramental empregado, etc. Desta maneira pode-se afirmar que o tempo de execução das atividades está sujeito a incertezas.

As incertezas existentes nestes parâmetros de tempo receberam pouca atenção nas décadas de 70 e 80. Somente na década de 90 começaram a surgir trabalhos que levavam em consideração as incertezas utilizando números *fuzzy*.

ISHII, TADA & MASUDA (1992), publicam um artigo que aborda as incertezas existentes na data de entrega das tarefas (*due-date*) o qual eles o denominam de *fuzzy due-date*. Os autores utilizaram dois algoritmos para esta introdução de incertezas. O primeiro, para fluxos de produção do tipo *open shop*, em um caso genérico envolvendo duas máquinas. E o segundo, para fluxos de produção do tipo *máquinas paralelas*.

MCCAHERON & LEE (1992), afirmam que até então os tempos de processamento existentes em problemas de programação da produção eram tratados como sendo tempos exatos, porém na prática raramente o eram. Estes autores são uns dos primeiros a abordar as incertezas existentes nos tempos de processamento de cada tarefa através dos números *fuzzy*. O algoritmo utilizado, pelos autores, foi para fluxo de produção do tipo *flow shop* baseado no CDS (CAMPBELL, DUKEK & SMITH, 1970), onde a função objetivo é minimizar o tempo máximo de duração de um conjunto de n tarefas, ou seja, obter o tempo total de produção (*makespan*). Este algoritmo, entretanto, sofreu algumas modificações para aceitar números *fuzzy* ao invés de números exatos. Os números *fuzzy* utilizados, por estes autores, foram os números *fuzzy* trapezoidais.

ISHIBUCHI, YAMAMOTO, MURATA & TANAKA (1994), investigam a relação existente entre problemas de programação das operações convencionais e problemas de programação *fuzzy* e afirmam que muitos problemas convencionais podem ser abordados como problemas *fuzzy*. Estes autores desenvolveram alguns algoritmos para fluxo de produção do tipo *flow-shop*, onde as incertezas foram inseridas nas datas de entrega das tarefas (*fuzzy due-date*), assim como em ISHII, TADA & MASUDA (1992), visto anteriormente. Na resolução desses problemas, os autores utilizaram duas técnicas combinadas: *algoritmo genéticos* e busca por vizinhança. Sendo, esta última, realizada através da *Busca Taboo* e *Simulated Annealing*. Estes autores utilizaram os números *fuzzy* triangulares e trapezoidais.

FORTEMPS (1997), desenvolveu uma metodologia utilizando alguns algoritmos heurísticos baseados na técnica de *Simulated Annealing*. Esta metodologia foi desenvolvida para problemas de programação do tipo *jobshop*, onde as incertezas foram inseridas nos tempos de processamento de cada operação, sendo que a função objetivo era obter o tempo total de produção (*makespan*).

STÜTZLE (1998), apresenta uma abordagem para problemas de programação *flow-shop* inspirada no comportamento de uma colônia de formigas. Esta abordagem baseia-se na idéia da busca cooperativa existente em colônias de formigas reais.

CURY (1999), apresenta em sua tese de doutorado uma abordagem que considera a fuzzificação dos tempos de processamento e datas de entrega das atividades. O trabalho leva em consideração o fato que o cliente ou fabricante têm um grau de satisfação em relação ao tempo de término e a data de entrega das atividades, onde este grau de satisfação é

considerado uma função de pertinência. A metodologia foi desenvolvida para problemas de programação do tipo *flow-shop* e o algoritmo utilizado foi do tipo *Branch and Bound*.

Notou-se nos algoritmos comentados acima que a etapa de fuzzificação é de total responsabilidade do especialista. Entretanto, quanto maior a quantidade de máquinas e tarefas envolvidas na programação, maior será a quantidade de tempos de operações existentes no problema que se deseja avaliar. Portanto, conforme o problema, a fase de fuzzificação pode ser tornar uma tarefa exaustiva.

Dentre os algoritmos apresentados, alguns são específicos para problemas de programação do tipo *flow-shop*. Nestes problemas, a comparação entre os diversos números *fuzzy* é necessária pois, nos algoritmos de programação de operações *flow-shop*, onde a função objetivo é minimizar o tempo máximo de duração de um conjunto de n tarefas, ou seja, obter o tempo total de produção (*makespan*), é necessário utilizar a comparação entre números *fuzzy* em sua resolução.

Porém, ao se utilizar a técnica de comparação pela desfuzzificação na resolução desses problemas, pode-se correr o risco de gerar resultados não satisfatórios do ponto de vista da programação de operações, conforme será observado e discutido no *Capítulo 3 – Metodologia proposta*.

2.5.7 – Diferenças entre TFNs e a distribuição Beta

A distribuição *Beta* é muito utilizada em gerenciamento de projetos com *PERT* (*Program Evaluation and Review technique* – Técnicas de Avaliação e Revisão de Programas) e *CPM* (*Critical Path Method* – Método do Caminho Crítico). Esta distribuição é utilizada no processo de estimação de tempos permitindo determinar, com boa aproximação, a média e a variância em função de três valores de tempo de fácil compreensão por aqueles encarregados de fazer tais estimações de tempo.

Esses três valores de tempo são a duração otimista ($t_0 = a$), a duração mais provável ($t_{mp} = m$) e a duração pessimista ($t_p = b$), definidas como:

- t_0 é o mínimo tempo que se estima como razoável para a execução da atividade.
 - t_{mp} é o tempo mais provável para a execução da atividade.
 - t_p é o máximo tempo que se estima como razoável para a execução da atividade.
-

Com essas 3 estimativas para a duração de uma atividade, pode-se utilizar as expressões 2.18 e 2.19, mostradas abaixo, para determinar a média $\mu(T)$ e a variância $\sigma^2(T)$, respectivamente, da distribuição da variável aleatória T , que é o tempo de execução da atividade. Essas equações fornecem uma boa aproximação para o verdadeiro valor da média e da variância (erro em torno de 5%).

$$\mu(T) = \frac{a + 4 \cdot m + b}{6} \quad (2.18)$$

$$\sigma^2(T) = \left(\frac{b - a}{6} \right)^2 \quad (2.19)$$

A figura 2.26 apresenta um exemplo da distribuição *beta* de probabilidades.

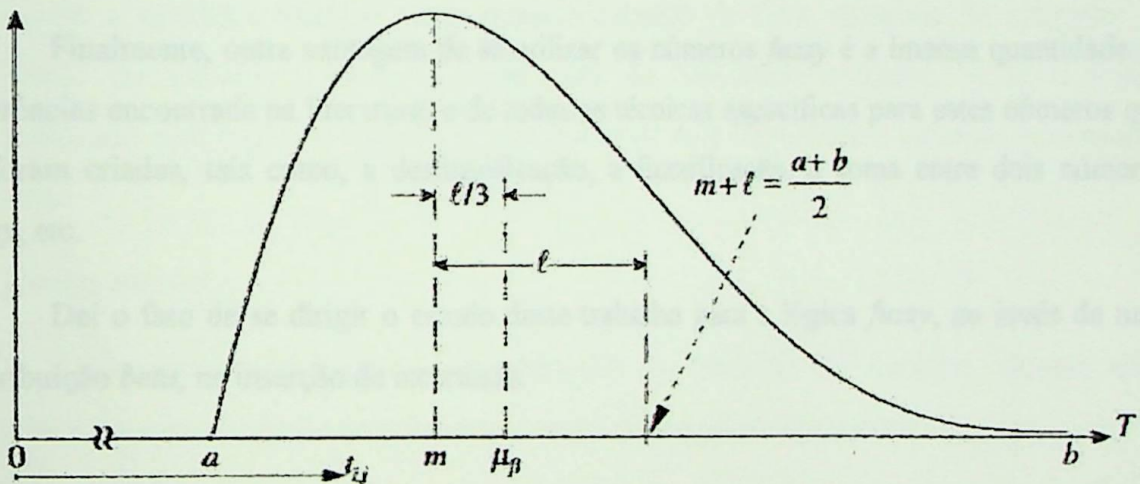


Figura 2.26 – Distribuição *beta* de probabilidades extraída de CONTADOR (1997)

Pode-se observar que o procedimento de estimação dos intervalos de tempo são muito semelhantes entre a distribuição *beta* e os *números triangulares fuzzy*. Em ambos os casos se tem um valor com maior possibilidade de ocorrência, e se opta entre uma opinião otimista e pessimista.

Entretanto, segundo CONTADOR (1997), dentre as distribuições de probabilidades conhecidas, a distribuição *beta* melhor se aplica a duração de uma atividade que se realiza uma *única* vez. Portanto para tempos de processamento das tarefas em sistemas intermitentes, este tipo de distribuição não se aplicaria muito bem, pois estes tempos ocorrem repetidas vezes.

Tem-se, também, o trabalho de HAPKE, JASZKIEWICZ & SLOWINSKI (1994), onde foi criada uma ferramenta computacional para ser empregada em gerenciamento de

projetos (PERT) denominada FPS (*Fuzzy Project Scheduling*). Neste trabalho, os autores criticam a abordagem utilizada normalmente na estimação dos intervalos de tempos das atividades através da distribuição *beta*, e propõem uma nova abordagem na estimação destes intervalos através dos números *fuzzy*.

GOMIDE & GUDWIN (1994), afirmam que a grande simplicidade de implementação de sistemas *fuzzy* pode reduzir a complexidade de um projeto a um ponto em que problemas anteriormente intratáveis passam a ser solúveis.

Sabe-se que a distribuição *beta* é uma distribuição de probabilidades, enquanto que os números *fuzzy* triangulares são uma distribuição de possibilidades. No primeiro a área sob a curva tem que ser sempre igual a 1, enquanto que no segundo, não necessariamente a área sobre a curva será igual a 1.

Finalmente, outra vantagem de se utilizar os números *fuzzy* é a imensa quantidade de referências encontrada na literatura, e de todas as técnicas específicas para estes números que já foram criadas, tais como, a desfuzzificação, a fuzzificação, a soma entre dois números *fuzzy*, etc.

Daí o fato de se dirigir o estudo deste trabalho para a lógica *fuzzy*, ao invés de uma distribuição *beta*, na inserção de incertezas.

2.6 – Considerações sobre tempo padrão

A metodologia tradicional do cálculo de tempo padrão, proposta por BARNES (1986), tem sido utilizada em situações reais do cotidiano fabril. A formulação de cálculo para obtenção deste tempo padrão, mostrada na expressão (2.20), foi extraída de ROSA (1996).

$$tp = t_1 f_1 [1 + (z_1 + z_2) f_2 + z_3 + z_4 + z_5] \quad (2.20)$$

Onde:

tp - Tempo padrão	f_2 - Fator de recuperação de fadiga
t_1 - Tempo médio cronometrado	z_3 - Abono por monotonia
f_1 - Ritmo de trabalho	z_4 - Tolerância pessoal
z_1 - Fadiga mental	z_5 - Ajustes e trocas de ferramentas
z_2 - Fadiga física	

t₁ - Tempo médio cronometrado

Média dos tempos cronometrados na realização de uma unidade de produção segundo um padrão de trabalho previamente determinado.

f₁ - Ritmo de trabalho

Fator relacionado com a habilidade e esforço do operador no sentido de corrigir o tempo cronometrado para as condições normais de execução da unidade de produção. Segundo BARNES (1986), tempo normal ou normalizado é o tempo necessário para a execução do trabalho em ritmo normal. Se o operador que está sendo cronometrado é extremamente habilidoso e esforçado, então o tempo cronometrado será menor que aquele de um operador com habilidade e esforço considerados normais.

Se o operador situa-se numa posição extrema de falta completa de empenho na realização do trabalho e demonstra uma falta de habilidade evidente, então o tempo cronometrado será maior que aquele de um operador com habilidade e esforço considerados normais.

z₁ - Fadiga mental

Abono relacionado com a complexidade de execução do trabalho. Quanto mais complexa for a execução do trabalho maior será o abono por fadiga mental.

z₂ - Fadiga física

Abono relacionado com o esforço físico na realização da tarefa. Quanto maior for o esforço maior será o abono.

f₂ - fator de recuperação de fadiga

A expressão $(a_1 + a_2)$ representa o abono bruto por fadiga. A expressão $(a_1 + a_2)f_2$, o abono líquido por fadiga. O fator f_2 , fator de recuperação de fadiga, se encarrega de diminuir o abono bruto por fadiga quando existe tempo automático e o operador pode recuperar-se da fadiga. Quando não existe tempo automático então $f_2 = 1$ e o abono líquido de fadiga coincide com o abono bruto.

z₃ - Abono por monotonia

O abono por monotonia está relacionado com a repetibilidade das ações executadas pelo operador na realização da unidade de produção. Quanto menor é o ciclo dessa

repetibilidade mais monótona é a tarefa e conseqüentemente maior será o abono concedido por monotonia.

z₄ - Tolerâncias pessoais

O abono por tolerâncias pessoais está associado ao tempo destinado ao atendimento das necessidades fisiológicas do trabalhador. Todo operário deve ter um tempo reservado para a satisfação de suas necessidades pessoais. A duração deste abono pode ser determinado através de um levantamento contínuo ou até mesmo por amostragem do trabalho.

z₅ - Ajustes e trocas de ferramentas

Este abono está relacionado com os ajustes e as trocas de ferramentas necessárias no decorrer do trabalho de produção. É função das condições dos equipamentos disponíveis para a execução da produção.

Os diversos fatores existentes na obtenção de tempo padrão foram apresentados, neste item, de forma sucinta. O objetivo foi demonstrar que, entre os diversos fatores utilizados na obtenção de tempo padrão, apenas o fator de ritmo de trabalho (f_1), está diretamente relacionado com o operador que executará a tarefa. Os demais fatores estão relacionados com o tipo de processo em questão, não sofrendo variações com a mudança de operador.

Este conceito será utilizado para o desenvolvimento da metodologia proposta no capítulo 3.

2.7 – Considerações finais

Neste capítulo foram abordados os principais conceitos e definições relevantes ao problema de programação de operações para sistemas de produção intermitentes do tipo *flow-shop* permutacional. Vários algoritmos foram apresentados na intenção de se mostrar a relevância que esse estudo possui na Engenharia de Produção.

Como foi constatado, nesta última década, foram publicados vários trabalhos sobre incertezas em algoritmos de programação de operações utilizando a lógica *fuzzy*. Entretanto, em nenhum dos trabalhos analisados percebeu-se uma preocupação na etapa de fuzziificação da matriz de tempos de operação das atividades, uma vez que esta fuzziificação fica sempre de responsabilidade específica do especialista.

A maioria dos algoritmos apresentados, utilizam a comparação entre números *fuzzy* em sua resolução. Como observado no item 2.5.4, para se efetuar a comparação entre dois números *fuzzy* é necessário efetuar a desfuzzificação desses números. Entretanto, o resultado da programação de operações obtido dessas comparações nem sempre geram a melhor resolução do problema, conforme será visto no próximo capítulo.

A metodologia desenvolvida no capítulo 3 a seguir, baseia-se nesses dois processos, fuzzificação e comparação pela desfuzzificação, necessários na resolução de problemas de programação de operações com incertezas, isto é, na redução das desvantagens comentadas nos dois parágrafos anteriores.

3.2 – Fuzzificação da matriz com os tempos de operações

Conforme mencionado na introdução, a programação de produção é fortemente dependente das particularidades do problema em questão e da distribuição dos dados, que neste caso são os tempos de processamento das tarefas em diversas máquinas.

A teoria *fuzzy* propõe que a introdução de incertezas nos tempos de operação das atividades envolvidas no processo, que o especialista avalia as condições que envolvem as atividades e seus resultados, e estabelece entre as divergências que podem ocorrer no tempo de operação. Esta teoria pode ser vista de forma simples, especialmente quando existem vários tempos de operação envolvidos em atividades que se deseja avaliar.

Dessa maneira, desenvolve-se um procedimento com o intuito de avaliar e associar as condições entre os tempos de operação existentes de atividades em questão. Com esse procedimento, importa-se que qualquer divergência de tempo possa ser avaliada por LINDENBERG apresentada no item 2.4 – Considerações sobre o método.

Capítulo 3: Metodologia proposta

3.1 – Considerações iniciais

Neste capítulo, primeiramente, será apresentado uma metodologia para a fuzzificação da matriz com os tempos de processamento das operações. Esta nova proposta, fornece uma sugestão inicial para o problema de programação de operações com considerações de incertezas, baseada na teoria de obtenção do tempo padrão de uma atividade.

Também será realizado, neste capítulo, o mesmo procedimento utilizado por alguns autores descritos na revisão bibliográfica para a obtenção da programação de operações, em problemas envolvendo incertezas. Este procedimento é efetuado pela comparação entre os diversos números *fuzzy* através da desfuzzificação.

Finalmente, uma nova abordagem será apresentada para resolução desses problemas. Esta nova abordagem será testada com o processo de comparação pela desfuzzificação, na intenção de se demonstrar a eficácia nos resultados obtidos.

3.2 – Fuzzificação da matriz com os tempos de operações

Conforme mencionado na introdução, a programação da produção é fortemente dependente das particularidades do problema em questão e da distribuição dos dados, que neste caso são os tempos de processamento das tarefas nas diversas máquinas.

A teoria fuzzy propõe para a introdução de incertezas, nos tempos de operação das atividades envolvidas no processo, que o especialista avalie as condições que envolvem as atividades a serem analisadas, e pondere entre as divergências que podem ocorrer no tempo de operação. Entretanto esta é uma tarefa árdua, especialmente quando existem vários tempos de operações envolvidos na atividade que se deseja avaliar.

Desse modo, desenvolveu-se um procedimento com o intuito de auxiliar o especialista a ponderar entre os tempos de operações existentes na atividade em questão. Este procedimento baseia-se nas técnicas de obtenção de tempo padrão proposta por BARNES (1986) apresentada no item 2.6 – *Considerações sobre tempo padrão*.

Deve-se deixar claro que a metodologia desenvolvida neste trabalho refere-se apenas a atividades manuais, onde o operador é determinante na definição do tempo padrão. Diferente do caso onde em uma linha de produção existem máquinas automáticas. A utilização de números *fuzzy* em atividades automáticas é perfeitamente possível, porém sua estimativa seria diferente da abordagem utilizada nesta dissertação.

A obtenção do tempo padrão, em atividades manuais, consiste em determinar o tempo necessário para uma pessoa qualificada e bem treinada, trabalhando em ritmo normal, executar uma tarefa. Ou seja, o resultado deste estudo é o tempo gasto por um operador adaptado ao trabalho e completamente treinado no método específico para executar uma atividade trabalhando em um ritmo considerado normal.

A técnica de obtenção de tempo padrão determina que cada tempo de operação é obtido através de uma média realizada entre diversas tomadas de tempo, e a esta média são acrescentadas várias ponderações como pôde ser visto na expressão (2.19).

Entretanto, dentre os diversos fatores existentes na expressão (2.19), apenas o fator de ritmo de trabalho (f_1) varia proporcionalmente ao operador que executará a tarefa, pois este fator está diretamente relacionado com o esforço e a habilidade do operador. Os demais fatores (fadigas, tolerâncias pessoais, etc.) são referentes ao processo em si, e não se alteram em função do operador.

Portanto, podemos rescrever a expressão (2.19) considerando apenas o tempo médio cronometrado (t_1), o ritmo de trabalho (f_1) e os demais fatores, representados por uma constante K . Desta forma obtém-se a expressão (3.1).

$$tp = t_1 \cdot f_1 \cdot K \quad (3.1)$$

Segundo BARNES (1986), o tempo padrão para uma atividade *não* pode variar, entretanto o tempo médio cronometrado (t_1) varia conforme o operador que executará a tarefa. Daí a utilização de um fator de correção para este tempo médio cronometrado em função da habilidade e esforço do operador (f_1).

Para um tempo médio cronometrado (t_1) obtido por um operador com habilidade e esforço *Normal* (tabela A.1 do Anexo A), tem-se um fator de ritmo de trabalho (f_1) igual a 1 (tabela A.2 do Anexo A). Neste caso o tempo padrão será obtido pela expressão (3.2) a seguir.

$$tp = t_1 \cdot f_1 \cdot K = t_1 \cdot 1 \cdot K = t_1 \cdot K \quad (3.2)$$

Já para um tempo médio cronometrado obtido por um operador com habilidade e esforço considerado *Boa* (tabela A.1), tem-se um fator de ritmo de trabalho igual a 1.11 (tabela A.2). Neste caso o tempo padrão será obtido pela expressão (3.3), onde percebe-se que o tempo padrão é aumentado, ou seja, corrigido, pois um operador habilidoso e esforçado executaria a atividade em um tempo menor do que um operador normal.

$$t_p = t_1 \cdot f_1 \cdot K = t_1 \cdot 1.11 \cdot K \quad (3.3)$$

Entretanto, para um tempo médio cronometrado obtido por um operador com habilidade e esforço considerado *Regular* (tabela A2.1), tem-se um fator de ritmo de trabalho igual a 0.82 (tabela A2.2). Neste caso o tempo padrão será obtido pela expressão (3.4), onde nota-se que o tempo médio cronometrado é reduzido (corrigido), pois um operador com habilidade e esforço regular necessitará de um tempo maior do que um operador normal para executar uma mesma atividade.

$$t_p = t_1 \cdot f_1 \cdot K = t_1 \cdot 0.82 \cdot K \quad (3.4)$$

Conforme mencionado anteriormente o tempo padrão de uma atividade não pode variar, portanto ao atribuir-mos ao tempo médio cronometrado para o operador com habilidade e esforço *Boa* o valor t_B , para um operador com habilidade e esforço *Normal* o valor t_N , e para um operador com habilidade e esforço *Regular* o valor t_R , teríamos, com base nas expressões (3.2), (3.3) e (3.4) a expressão (3.5) a seguir.

$$t_p = \begin{cases} t_B \cdot 1.11 \cdot K \\ t_N \cdot K \\ t_R \cdot 0.82 \cdot K \end{cases} \quad (3.5)$$

Da expressão (3.5), obtém-se as expressões (3.6) e (3.7).

$$t_B \cdot 1.11 \cdot K = t_N \cdot K \quad (3.6)$$

$$t_R \cdot 0.82 \cdot K = t_N \cdot K \quad (3.7)$$

Da expressão (3.6), obtém-se a expressão (3.8).

$$t_B \cdot K = \frac{t_N \cdot K}{1.11} \quad (3.8)$$

E da expressão (3.7), obtém-se a expressão (3.9).

$$t_R \cdot K = \frac{t_N \cdot K}{0.82} \quad (3.9)$$

Desta forma, ao considerarmos um tempo padrão para determinada atividade, poderíamos estimar a variação do tempo médio cronometrado em função do K para os três casos possíveis de esforço e habilidade (*Boa*, *Normal* e *Regular*), ou seja ($t_B \cdot K$, t_p , $t_R \cdot K$).

Por exemplo, para um tempo padrão de 10 minutos, teríamos, pela expressão (3.5), a expressão (3.10).

$$t_p = t_N \cdot K = 10 \quad (3.10)$$

E pelas expressões (3.8), (3.9) e (3.10), teríamos as expressões (3.11) e (3.12) a seguir.

$$t_B \cdot K = \frac{t_N \cdot K}{1.11} = \frac{t_p}{1.11} = \frac{10}{1.11} = 9.01 \quad (3.11)$$

$$t_R \cdot K = \frac{t_N \cdot K}{0.82} = \frac{t_p}{0.82} = \frac{10}{0.82} = 12.20 \quad (3.12)$$

Desta forma é possível afirmar que para uma atividade que possua um tempo padrão de 10 minutos, os tempos médios cronometrados em função do K podem variar de 9.01 a 12.20 minutos, dependendo do operador que executará a atividade.

O procedimento desenvolvido, neste trabalho, considera que a fuzzificação dos tempos de operação determinísticos seja realizada atribuindo-se ao valor a_2 de um TFN (item 2.5.2), a expressão (3.10), ou seja, o próprio valor do tempo padrão. Isto é factível, pois existe grande possibilidade de qualquer outro operador efetuar a mesma tarefa em um tempo com as características de habilidade e esforço consideradas normais.

Para o valor a_1 (otimista) de um TFN, deverá ser atribuído o valor obtido pela expressão (3.11), ou seja, o valor do tempo padrão dividido pelo fator de ritmo de trabalho para um operador com esforço e habilidade *Boa*. Isto também é factível, pois um operador com essas características realizará o trabalho no menor tempo possível.

Seguindo a mesma linha de raciocínio, para o valor a_3 (pessimista) de um TFN, deverá ser atribuído o valor obtido pela expressão (3.12), ou seja, o valor do tempo padrão dividido pelo fator de ritmo de trabalho para um operador com esforço e habilidade *Regular*, pois um operador com essas características necessitará de um tempo maior para realizar tais tarefas.

Portanto, ao realizarmos a fuzzificação de um tempo de operação determinístico de 10 minutos, este passaria a ser representado pelo número *fuzzy* triangular (9,10, 12,20) minutos.

Utilizando-se o procedimento descrito acima para todos os tempos de processamentos mostrado na figura 3.1 a seguir, obtém-se a matriz com os tempos de operações com considerações de incertezas que podem ser vistos na figura 3.2. Nesta matriz, todos os tempos de operação estão representados por um número *fuzzy* triangular, onde, para a peça 1 processada na máquina 1, tem-se o tempo de operação de (8,1, 9, 11), o que significa que existe grande possibilidade da peça 1 ser processada em um tempo de 9 u.t. (unidades de tempo), entretanto este tempo pode variar entre os tempos otimista e pessimista (8.1 e 11 respectivamente) conforme a variação da habilidade e esforço do operador.

Peças (J _i)	Máquinas (M _j)			
	M1	M2	M3	M4
J1	9	4	5	8
J2	9	3	10	1
J3	5	9	8	10

Figura 3.1 – Matriz com os tempos de processamento das operações determinísticos

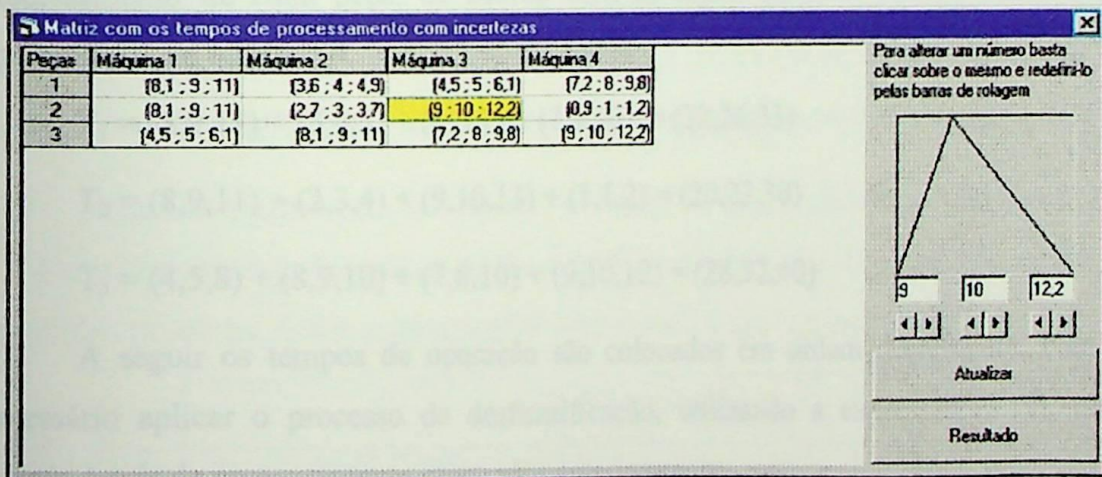


Figura 3.2 - Matriz com os tempos de processamento após processo de fuzzificação

Nota-se que para facilitar os cálculos, foi desenvolvido um software para a metodologia proposta. Basta digitar o tempo padrão para cada operação que o software gera pela fuzzificação os números triangulares *fuzzy*, automaticamente. Cabe ressaltar que o especialista pode interagir com o sistema, alterando qualquer um dos tempos que julgar necessário, conforme descrito no Apêndice A – Modelo computacional.

3.3 – Resolução de problemas de programação de operações com incertezas pela desfuzzificação

A seguir será apresentado uma abordagem para resoluções de problemas de programação de operações baseada na comparação através da desfuzzificação do número *fuzzy*. Esta abordagem é a mesma apresentada por MCCAHERN & LEE (1992), ISHIBUCHI, YAMAMOTO, MURATA & TANAKA (1994), STÜTZLE (1998) e CURY (1999).

Primeiramente será apresentado a solução pelo algoritmo NEH e posteriormente pelo algoritmo de JOHNSON.

3.3.1 – Resolução pelo algoritmo NEH

O algoritmo NEH (NAWAZ, ENSCORE JR & HAM, 1983), foi apresentado no item 2.4.3 da revisão bibliográfica.

Para o desenvolvimento deste algoritmo, utilizou-se a matriz com os tempos de processamento das operações mostrada anteriormente na figura 3.1. A fuzzificação dos tempos de processamento de cada operação foi realizada segundo o procedimento descrito no item 3.2. O resultado da fuzzificação desta matriz foi mostrado anteriormente na figura 3.2.

Conforme o algoritmo, como primeiro passo, calcula-se os tempos totais de processamento de cada peça, de acordo com as regras de somatório de números *fuzzy*, apresentados no item 2.5.4.

$$T_1 = (8,9,11) + (3,4,5) + (4,5,7) + (7,8,10) = (22,26,33)$$

$$T_2 = (8,9,11) + (2,3,4) + (9,10,13) + (1,1,2) = (20,23,30)$$

$$T_3 = (4,5,8) + (8,9,10) + (7,8,10) + (9,10,12) = (28,32,40)$$

A seguir os tempos de operação são colocados em ordem decrescente. Para tal, é necessário aplicar o processo de desfuzzificação, utilizando a expressão (2.17), nos três tempos totais de processamento mostrados anteriormente.

$$\text{Para } T_1, \text{ temos que } \hat{T}_1 = \frac{22 + 2 \cdot 26 + 33}{4} = 26.75$$

$$\text{Para } T_2, \text{ temos que } \hat{T}_2 = \frac{20 + 2 \cdot 23 + 30}{4} = 24$$

$$\text{Para } T_3, \text{ temos que } \hat{T}_3 = \frac{28 + 2 \cdot 32 + 40}{4} = 33$$

Após os resultados da desfuzzificação, obtém-se a ordenação decrescente J3, J1 e J2 para os tempos totais de processamento de cada peça. Com isto, calcula-se a melhor ordenação parcial para as peças J1 e J3, conforme mostrado nas figuras 3.3 e 3.4 a seguir.

		Máquinas (M _j)			
Peças (J _i)		M1	M2	M3	M4
J1		(8,9,11) / (8,9,11)	(3,4,5) / (11,13,16)	(4,5,7) / (15,18,23)	(7,8,10) / (22,26,33)
J3		(4,5,8) / (12,14,19)	(8,9,10) / (20,23,29)	(7,8,10) / (27,31,39)	(9,10,12) / (36,41,51)

Figura 3.3 - Tempo de fluxo de operação para ordenação parcial J1-J3

		Máquinas (M _j)			
Peças (J _i)		M1	M2	M3	M4
J3		(4,5,8) / (4,5,8)	(8,9,10) / (12,14,18)	(7,8,10) / (19,22,28)	(9,10,12) / (28,32,40)
J1		(8,9,11) / (12,14,19)	(3,4,5) / (15,18,24)	(4,5,7) / (23,27,35)	(7,8,10) / (35,40,50)

Figura 3.4 - Tempo de fluxo de operação para ordenação parcial J3-J1

É importante ressaltar que os tempos de fluxo de operação para as ordenações parciais apresentadas na figura 3.3 e 3.4 foram obtidas através da soma e comparação entre os números *fuzzy* triangulares, sendo esta comparação realizada através de sua desfuzzificação.

Pode-se observar que a melhor ordenação parcial obtida acima foi a das peças J3-J1. Esta ordenação possui o tempo de fluxo de operação igual a (35,40,50). Nos próximos passos a posição relativa das peças J1 e J3 deverá ser, então, primeiro a peça J3 e depois a peça J1.

Seguindo o algoritmo, seleciona-se a 3ª peça da lista de ordem decrescente de tempo, que é a peça número 2. Calcula-se, então, a melhor ordenação parcial colocando a peça J2 nas 3 posições possíveis. Estas ordenações são demonstradas nas figuras 3.5, 3.6, 3.7.

		Máquinas (M _j)			
Peças (J _i)		M1	M2	M3	M4
J3		(4,5,8) / (4,5,8)	(8,9,10) / (12,14,18)	(7,8,10) / (19,22,28)	(9,10,12) / (28,32,40)
J1		(8,9,11) / (12,14,19)	(3,4,5) / (15,18,24)	(4,5,7) / (23,27,35)	(7,8,10) / (36,40,50)
J2		(8,9,11) / (19,23,30)	(2,3,4) / (21,26,34)	(9,10,13) / (31,37,48)	(1,1,2) / (36,41,52)

Figura 3.5 - Tempo de fluxo de operação para a ordenação parcial J3-J1-J2

Peças (J_i)	Máquinas (M_j)			
	M1	M2	M3	M4
J3	(4,5,8) / (4,5,8)	(8,9,10) / (12,14,18)	(7,8,10)/(19,22,28)	(9,10,12)/(28,32,40)
J2	(8,9,11) / (12,14,19)	(2,3,4) / (14,17,23)	(9,10,13)/(28,32,41)	(1,1,2)/(29,33,43)
J1	(8,9,11) / (20,23,30)	(3,4,5) / (23,27,35)	(4,5,7) / (32,37,48)	(7,8,10)/(39,45,58)

Figura 3.6 - Tempo de fluxo de operação para a ordenação parcial J3-J2-J1

Peças (J_i)	Máquinas (M_j)			
	M1	M2	M3	M4
J2	(8,9,11) / (8,9,11)	(2,3,4) / (10,12,15)	(9,10,13)/(19,22,28)	(1,1,2)/(20,23,30)
J3	(4,5,8) / (12,14,19)	(8,9,10) / (20,23,29)	(7,8,10)/(27,31,39)	(9,10,12)/(36,41,51)
J1	(8,9,11) / (20,23,30)	(3,4,5) / (23,27,35)	(4,5,7) / (31,36,46)	(7,8,10)/(43,49,61)

Figura 3.7 - Tempo de fluxo de operação para a ordenação parcial J2-J3-J1

Pelos cálculos obtidos no desenvolvimento do algoritmo acima, observa-se que a melhor ordenação para a produção das peças é a J3-J1-J2, que determina um tempo de fluxo de operação igual a (36,41,52).

O procedimento descrito anteriormente poderia ser simplificado, se logo do início, aplicasse-mos o processo de desfuzzificação na matriz de tempos de operação *fuzzy* mostrada na figura 3.2. O resultado dessa desfuzzificação pode ser visto na figura 3.8 a seguir.

Peças (J_i)	Máquinas (M_j)			
	M1	M2	M3	M4
J1	9.3	4.2	5.2	8.3
J2	9.3	3.1	10.3	1.0
J3	5.2	9.3	8.3	10.3

Figura 3.8 – Matriz com os tempos de processamento após o processo de desfuzzificação

Da mesma forma que anteriormente, são realizados os procedimentos para a resolução de problemas de programações de operações pelo algoritmo NEH descritos no item 2.4.3.

Como primeiro passo, calcula-se os tempos totais de processamento de cada peça.

$$T_1 = 9.3 + 4.2 + 5.2 + 8.3 = 27.0$$

$$T_2 = 9.3 + 3.1 + 10.3 + 1.0 = 23.7$$

$$T_3 = 5.2 + 9.3 + 8.3 + 10.3 = 33.1$$

A seguir os tempos de operação são colocados em ordem decrescente, resultando na ordenação J3, J1, J2. Com isto, calcula-se a melhor ordenação parcial para as peças J1 e J3, conforme mostrado nas figuras 3.9 e 3.10

	Máquinas (M _j)			
Peças (J _i)	M1	M2	M3	M4
J1	9.3 / 9.3	4.2 / 13.5	5.2 / 18.7	8.3 / 27
J3	5.2 / 14.5	9.3 / 23.8	8.3 / 32.1	10.3 / 42.4

Figura 3.9 - Tempo de fluxo de operação para a ordenação parcial J1-J3

	Máquinas (M _j)			
Peças (J _i)	M1	M2	M3	M4
J3	5.2 / 5.2	9.3 / 14.5	8.3 / 22.8	10.3 / 33.1
J1	9.3 / 14.5	4.2 / 18.7	5.2 / 28	8.3 / 41.4

Figura 3.10 - Tempo de fluxo de operação para a ordenação parcial J3-J1

Pode-se observar que a melhor ordenação parcial obtida acima também foi a das peças J3-J1. Esta ordenação possui o tempo de fluxo de operação é igual a 41.4. Nos próximos passos a posição relativa das peças J1 e J3 deverá ser, então, primeiro a peça J3 e depois a peça J1.

Seguindo o algoritmo, seleciona-se a 3ª peça da lista de ordem decrescente de tempo, que é a peça número 2. Calcula-se, então, a melhor ordenação parcial colocando a peça J2 nas 3 posições possíveis. Estas ordenações são demonstradas nas figuras 3.11, 3.12, 3.13.

	Máquinas (M _j)			
Peças (J _i)	M1	M2	M3	M4
J3	5.2 / 5.2	9.3 / 14.5	8.3 / 22.8	10.3 / 33.1
J1	9.3 / 14.5	4.2 / 18.7	5.2 / 28	8.3 / 41.4
J2	9.3 / 23.8	3.1 / 26.9	10.3 / 38.3	1.0 / 42.4

Figura 3.11 - Tempo de fluxo de operação para a ordenação parcial J3-J1-J2

		Máquinas (M _j)			
Peças (J _i)		M1	M2	M3	M4
J3		5.2 / 5.2	9.3 / 14.5	8.3 / 22.8	10.3 / 33.1
J2		9.3 / 14.5	3.1 / 17.6	10.3 / 33.1	1.0 / 34.1
J1		9.3 / 23.8	4.2 / 28	5.2 / 38.3	8.3 / 46.6

Figura 3.12 - Tempo de fluxo de operação para a ordenação parcial J3-J2-J1

		Máquinas (M _j)			
Peças (J _i)		M1	M2	M3	M4
J2		9.3 / 9.3	3.1 / 12.4	10.3 / 22.7	1.0 / 23.7
J3		5.2 / 14.5	9.3 / 23.8	8.3 / 32.1	10.3 / 42.4
J1		9.3 / 23.8	4.2 / 28	5.2 / 37.3	8.3 / 50.7

Figura 3.13 - Tempo de fluxo de operação para a ordenação parcial J2-J3-J1

Pelos cálculos obtidos no desenvolvimento do algoritmo acima, observa-se que a melhor ordenação para a produção das peças é a J3-J1-J2, que determina um tempo de fluxo de operação igual a 42.4. O gráfico de *Gantt* para esta ordenação pode ser visto na figura 3.14, onde a cor verde foi atribuída a peça J3, a cor amarela foi atribuída a peça J1 e a cor azul foi atribuída a peça J2.

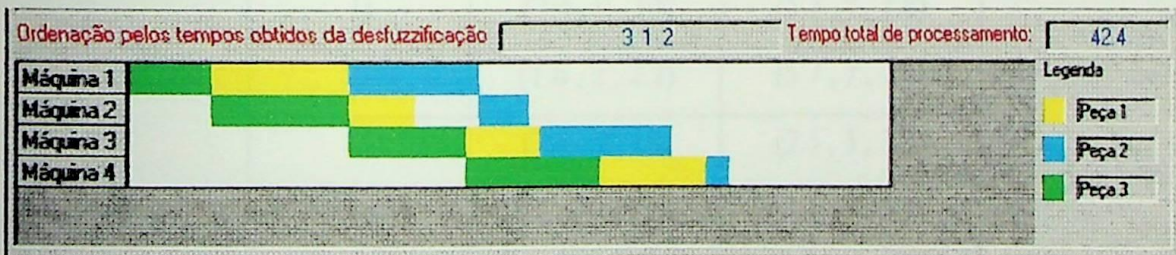


Figura 3.14 – Gráfico de *Gantt* obtido pela desfuzzificação dos tempos de processamento

Os resultados obtidos através dos dois procedimentos descritos acima geram o mesmo resultado. Ao aplicarmos o processo de desfuzzificação no tempo de processamento *fuzzy* (36,41,52), apresentado pelo primeiro procedimento, obteríamos o valor 42.5, que é praticamente o mesmo valor gerado pelo segundo procedimento (42.4). Esta diferença de 0.1 ocorre em função do arredondamento realizado na desfuzzificação de cada tempo de operação *fuzzy* da matriz da figura 3.2.

3.3.2 - Resolução pelo Algoritmo de JOHNSON

O algoritmo de JOHNSON (JOHNSON, 1954) foi detalhado no item 2.4.4 da revisão bibliográfica, onde, para o desenvolvimento do algoritmo, utilizou-se a matriz com os tempos de processamento das operações mostrada na figura 3.15 a seguir.

Tarefas (J_i)	Máquinas (M_j)	
	M1	M2
J1	9	7
J2	2	7
J3	9	3
J4	10	5
J5	4	8

Figura 3.15 – Matriz com os tempos de processamento das operações em 2 máquinas

A fuzzificação dos tempos de processamento de cada operação será realizada conforme o procedimento descrito no item 3.2. A matriz com a inserção de incertezas é mostrada na figura 3.16.

Tarefas (J_i)	Máquinas (M_j)	
	M1	M2
J1	(7.4 , 9 , 10)	(5.7 , 7 , 7.8)
J2	(1.6 , 2 , 2.2)	(5.7 , 7 , 7.8)
J3	(7.4 , 9 , 10)	(2.5 , 3 , 3.3)
J4	(8.2 , 10 , 11.1)	(4.1 , 5 , 5.5)
J5	(3.3 , 4 , 4.4)	(6.7 , 8 , 8.9)

Figura 3.16 – Matriz com os tempos de processamento com incertezas

Para facilitar os cálculos necessários para a obtenção da programação de operações, assim como no item 3.3.1, será realizado o processo de desfuzzificação utilizando a expressão (2.16) em todos os tempos de processamento *fuzzy* mostrados na figura 3.16. O resultado dessa desfuzzificação pode ser visto na figura 3.17.

Tarefas (J_i)	Máquinas (M_j)	
	M1	M2
J1	8.85	6.88
J2	1.95	6.88
J3	8.85	2.95
J4	9.82	4.9
J5	3.93	7.88

Figura 3.17 – Matriz com os tempos de processamento após a desfuzzificação

A seguir são realizados os procedimentos descritos para a resolução de problemas de programações de operações pelo algoritmo JOHNSON descritos no item 2.4.4.

Conforme este algoritmo, como primeiro passo, descobre-se o menor tempo de processamento das tarefas entre as máquinas 1 e 2. Esse menor tempo é o $p_{21} = 1.95$. Esta tarefa é então retirada da tabela e colocada como primeiro elemento da ordenação ótima $S = (2, \dots)$. O próximo menor tempo é o $p_{32} = 2.95$. Portanto, essa tarefa é retirada da tabela e colocada como último elemento da ordenação ótima $S = (2, \dots, 3)$. A seguir, tem-se que o menor tempo é o $p_{51} = 3.93$. Então a tarefa 5 ocupará, agora, a segunda posição na ordenação ótima $S = (2, 5, \dots, 3)$. Na seqüência, tem-se que o menor tempo é $p_{42} = 4.9$. Desta forma a tarefa 4 ocupará a quarta posição na ordenação ótima $S = (2, 5, \dots, 4, 3)$. O último tempo de processamento é $p_{12} = 6.88$. Portanto, a tarefa 1 ocupará a posição que resta na ordenação ótima, gerando o resultado $S = (2, 5, 1, 4, 3)$.

A solução para este problema também pode ser visualizada pela tabela apresentada na figura 3.18 a seguir.

Estágio	Tarefas ainda não programadas	Mínimo p_{ij}	Designação	Ordenação parcial
1	J_1, J_2, J_3, J_4, J_5	p_{21}	$J_2 = J_{[1]}$	J_2
2	J_1, J_3, J_4, J_5	p_{32}	$J_3 = J_{[5]}$	$J_2 - - - J_3$
3	J_1, J_4, J_5	p_{51}	$J_5 = J_{[2]}$	$J_2, J_5 - - J_3$
4	J_1, J_4	p_{42}	$J_4 = J_{[4]}$	$J_2, J_5 - J_4, J_3$
5	J_1	p_{12}	$J_1 = J_{[3]}$	J_2, J_5, J_1, J_4, J_3

Figura 3.18 – Solução do problema pelo algoritmo de JOHNSON através da desfuzzificação

O tempo total para a execução das atividade (*makespan*) pode ser obtido pelo gráfico de Gantt mostrado na figura 3.19, onde a cor azul foi atribuída a peça J2, a cor vermelha foi atribuída a peça J5, a cor amarela foi atribuída a peça J1, a cor rosa foi atribuída a peça J4 e a cor verde foi atribuída a peça J3.

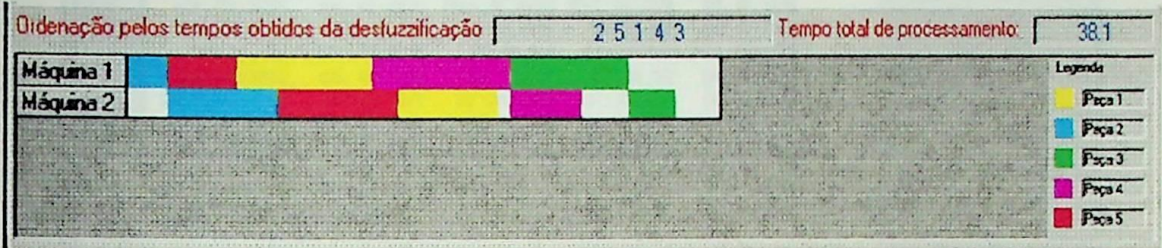


Figura 3.19 – Gráfico de *Gantt* obtido para o problema

3.4 – Nova abordagem proposta para a resolução de problemas de programação de operações com incertezas

A seguir será apresentada uma nova abordagem para a resolução de problemas de programação de operações com incertezas. Esta nova abordagem leva em consideração as programações obtidas para cada um dos valores do TFN (otimista, maior possibilidade e pessimista), além da programação gerada pela comparação através da desfuzzificação dos tempos de operação, mostrada no item anterior.

3.4.1 – Resolução pelo algoritmo NEH

Considere o seguinte problema de programação de operações envolvendo 7 tarefas e 5 máquinas. A matriz com os tempos de operação em minutos para cada atividade é mostrada na figura 3.20.

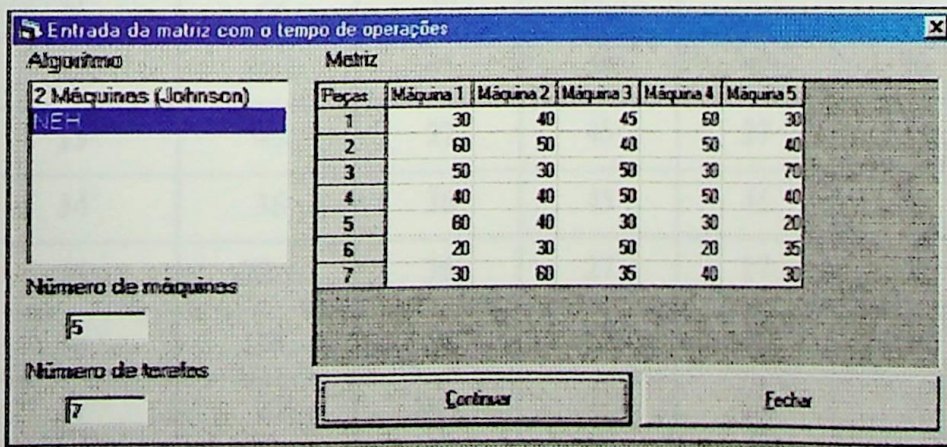
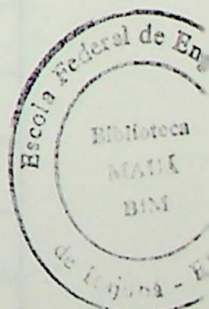


Figura 3.20 – Matriz com os tempos de processamento



O processo de fuzzificação desta matriz é automaticamente realizado pelo software, conforme o procedimento descrito no item 3.2. A matriz com os tempos de operação *fuzzy* é mostrada na figura 3.21.

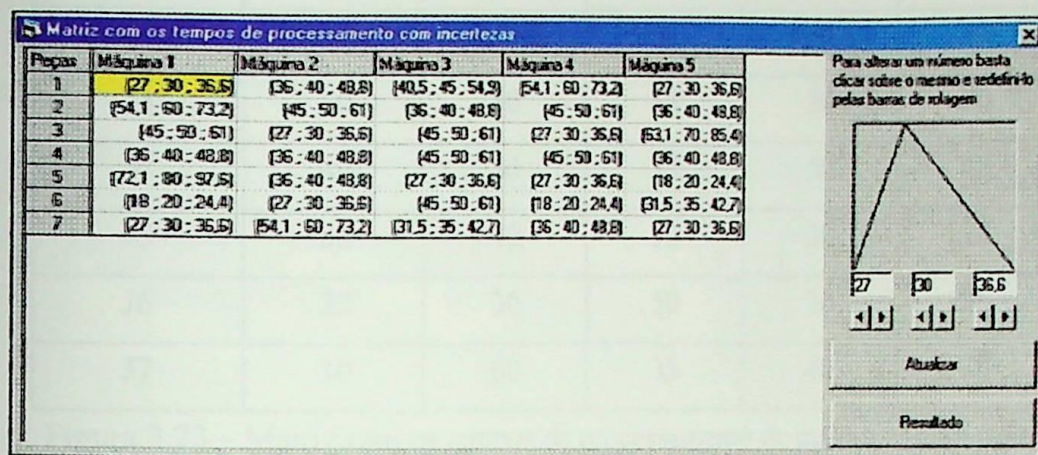


Figura 3.21 - Matriz com os tempos de operação *fuzzy*

Como este exemplo é puramente didático, não será efetuada nenhuma mudança nos números *fuzzy* apresentados pelo sistema.

Nesta etapa, além de efetuar a programação de operações nos tempos obtidos pela comparação através da desfuzzificação da matriz da figura 3.21, será também efetuada a programação de operações separadamente para cada um dos valores do TFN (otimista, maior possibilidade e pessimista) apresentados pelo sistema.

Ao considerar-se estes valores separadamente, tem-se, então, quatro matrizes com os tempos de processamento de cada operação mostrados nas figuras 3.22, 3.23, 3.24 e 3.25.

Peças (J _i)	Máquinas (M _j)				
	M1	M2	M3	M4	M5
J1	27	36	40.5	54.1	27
J2	54.1	45	36	45	36
J3	45	27	45	27	63.1
J4	36	36	45	45	36
J5	72.1	36	27	27	18
J6	18	27	45	18	31.5
J7	27	54.1	31.5	36	27

Figura 3.22 – Matriz com os tempos de processamento otimistas

Peças (J_i)	Máquinas (M_j)				
	M1	M2	M3	M4	M5
J1	30	40	45	60	30
J2	60	50	40	50	40
J3	50	30	50	30	70
J4	40	40	50	50	40
J5	80	40	30	30	20
J6	20	30	50	20	35
J7	30	60	35	40	30

Figura 3.23 – Matriz com os tempos de processamento de maior possibilidade

Peças (J_i)	Máquinas (M_j)				
	M1	M2	M3	M4	M5
J1	36.6	48.8	54.9	73.2	36.6
J2	73.2	61	48.8	61	48.8
J3	61	36.6	61	36.6	85.4
J4	48.8	48.8	61	61	48.8
J5	97.6	48.8	36.6	36.6	24.4
J6	24.4	36.6	61	24.4	42.7
J7	36.6	73.2	42.7	48.8	36.6

Figura 3.24 – Matriz com os tempos de processamento pessimistas

Peças (J_i)	Máquinas (M_j)				
	M1	M2	M3	M4	M5
J1	30.9	41.2	46.4	61.8	30.9
J2	61.8	51.5	41.2	51.5	41.2
J3	51.5	30.9	51.5	30.9	72.1
J4	41.2	41.2	51.5	51.5	41.2
J5	82.4	41.2	30.9	30.9	20.6
J6	20.6	30.9	51.5	20.6	36.0
J7	30.9	61.8	36.0	41.2	30.9

Figura 3.25 – Matriz com os tempos de processamento representados pela sua desfuzzificação

As programações das operações sugeridas pelo sistema, para cada uma dessas matrizes com os tempos de operação apresentadas anteriormente, são realizadas através dos procedimentos descritos no item 2.4.3 – *Algoritmo NEH*.

O resultado dessas programações, com a visualização dos gráficos de *Gantt*, são apresentados na figura 3.26 a seguir.

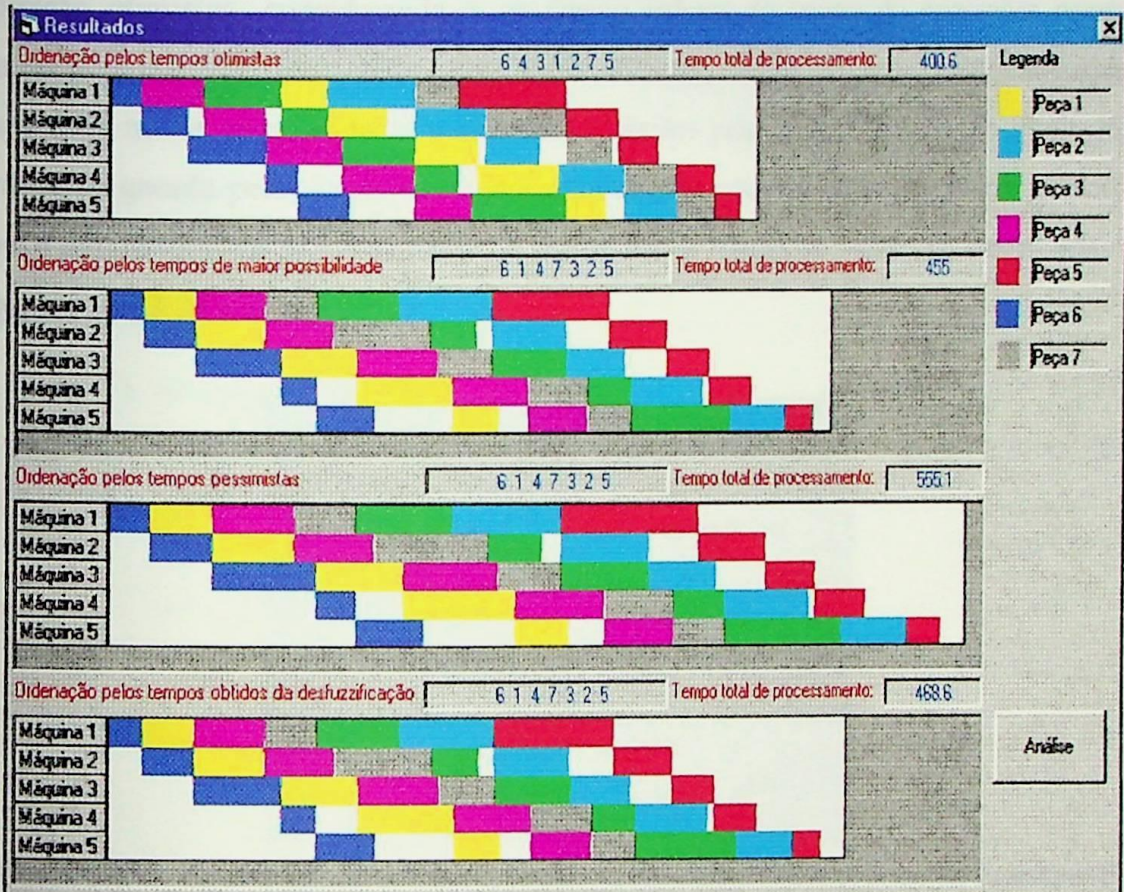


Figura 3.26 – Resultado da programação das operações

Pode-se reparar que o sistema apresentou dois resultados distintos. A mesma ordenação foi obtida para as matrizes de tempos de operações considerados de maior possibilidade, pessimistas e pela defuzzificação (6, 1, 4, 7, 3, 2, 5). Já para a matriz de tempos otimistas a ordenação foi (6, 4, 3, 1, 2, 7, 5).

De imediato o decisor é tentado a optar pela ordenação (6, 1, 4, 7, 3, 2, 5), uma vez que esta ordenação foi obtida em três das quatro matrizes de tempos de operações possíveis.

Entretanto esta nova abordagem prevê que quando ocorre qualquer tipo de variações nas ordenações geradas pelo sistema para as matrizes de tempos de operações analisadas, é necessário realizar uma comparação dessas ordenações.

Esta comparação é efetuada *mantendo-se fixas* as ordenações obtidas e variando-se os demais tempos das matrizes de processamentos possíveis para essa ordenação. Esta análise é efetuada com o intuito de se buscar o melhor resultado dentre as ordenações apresentadas, como resposta, pelo sistema.

Portanto, inicialmente compara-se a ordenação obtida pela matriz de tempos de operação otimistas, considerando-se os demais valores de tempo de operações possíveis (maior possibilidade, pessimista e pela desfuzzificação). Isto significa que serão utilizados os tempos de maior possibilidade, pessimistas e obtidos pela desfuzzificação, no resultado da ordenação gerada pela matriz de tempos otimistas, na intenção de se verificar o que ocorre com o tempo total de processamento. A análise desses resultados podem ser visualizadas nas figuras 3.27, 3.28 e 3.29 a seguir.

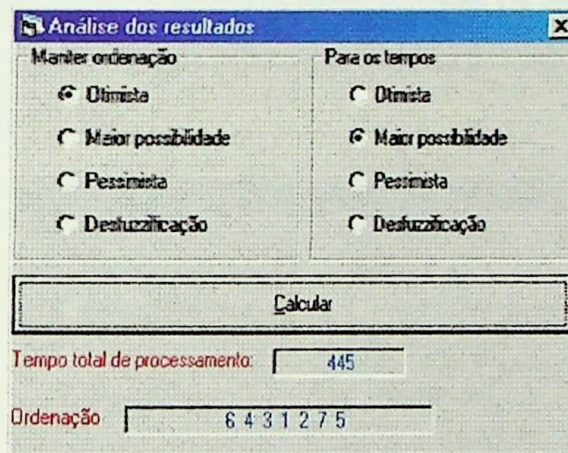


Figura 3.27 – Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação otimistas com os valores de tempos de maior possibilidade

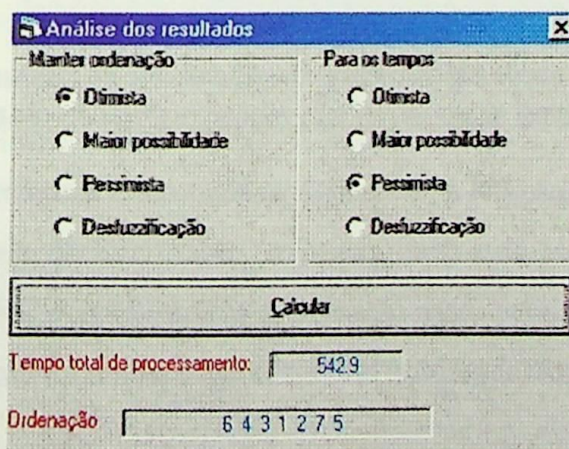


Figura 3.28 – Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação otimistas com os valores de tempos pessimistas

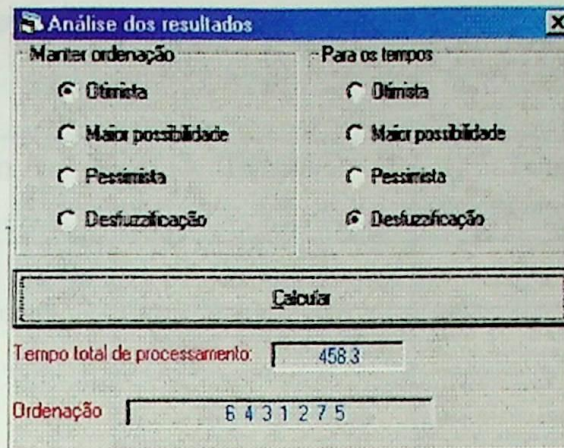


Figura 3.29 – Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação otimistas com os valores de tempos obtidos da desfuzzificação

Dando continuidade a análise, posteriormente comparou-se a ordenação obtida pela matriz de tempos de operação de maior possibilidade, pessimistas e pela desfuzzificação (6, 1, 4, 7, 3, 2, 5), considerando-se os valores de tempo otimistas. Ou seja, será utilizado o valor de tempo otimista na ordenação gerada pelas outras matrizes de tempo. O resultado desta análise pode ser visto na figura 3.30.

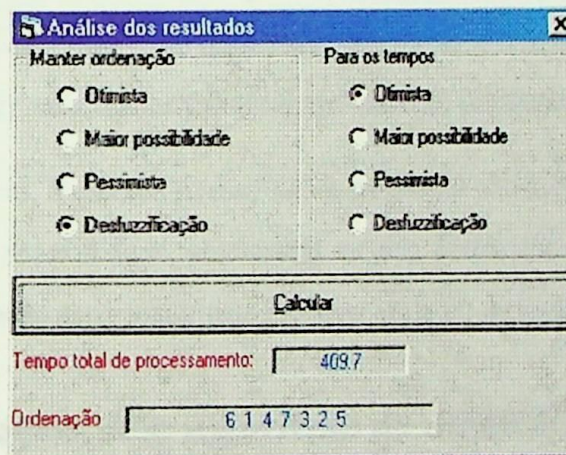


Figura 3.30 – Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos otimistas

Na análise realizada e mostrada na figura 3.30, a ordenação selecionada foi a obtida pela comparação pela desfuzzificação, entretanto poder-se-ia escolher a ordenação obtida pelos valores de maior possibilidade ou a ordenação obtida pelos valores pessimistas, pois o resultado da ordenação foi o mesmo para essas três matrizes de tempo de operação.

Observa-se que o resultado final, ou seja o tempo total de produção aumentou quando considerou-se a ordenação obtida pela comparação pela desfuzzificação da matriz com os valores de tempos otimistas. Entretanto diminuiu quando considerou-se a ordenação obtida pela matriz de tempos otimistas com os demais valores de tempos de processamento. Estas

comparações entre os resultados podem ser melhor visualizadas na tabela apresentada na figura 3.31.

Ordenação	Tempos	Tempo total de processamento (makespan)
Considerando a matriz de tempo otimista <i>6, 4, 3, 1, 2, 7, 5</i>	Otimistas	400.6
	Maior possibilidade	445
	Pessimistas	542.9
	Pela desfuzzificação	458.3
Considerando as demais matrizes de tempo <i>6, 1, 4, 7, 3, 2, 5</i>	Otimistas	409.7
	Maior possibilidade	455
	Pessimistas	555.1
	Pela desfuzzificação	468.6

Figura 3.31 – Análise dos resultados pelo algoritmo NEH

Portanto, a melhor ordenação para a resolução desse problema, seria a ordenação obtida considerando-se a matriz de tempo otimistas (6, 4, 3, 1, 2, 7, 5), onde o tempo total de produção (*makespan*) seria menor para qualquer variação de tempo que pudesse ocorrer.

Se ao contrário, a escolha tivesse sido a ordenação obtida considerando-se as matrizes de tempo de maior possibilidade, pessimistas ou pela desfuzzificação (6, 1, 4, 7, 3, 2, 5), o tempo total de produção seria maior para qualquer variação de tempo que pudesse ocorrer.

3.4.2 – Resolução pelo algoritmo de JOHNSON

Considere-se o seguinte problema de programação da produção envolvendo 8 tarefas e 2 máquinas. A matriz com os tempos de operação em minutos para cada atividade é mostrada na figura 3.32.

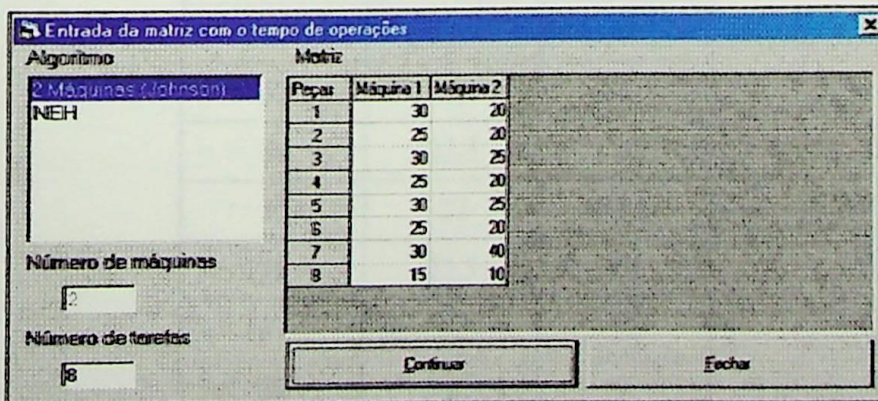


Figura 3.32 – Matriz com os tempos de processamento

O processo de fuzzificação desta matriz é automaticamente realizado pelo software, conforme o procedimento descrito no item 3.2. A matriz com os tempos de operação *fuzzy* é mostrada na figura 3.33.

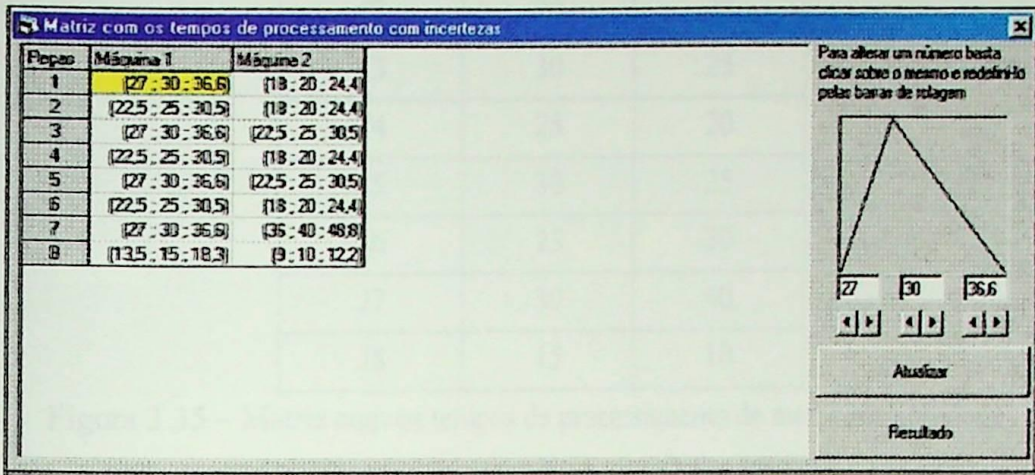


Figura 3.33 - Matriz com os tempos de operação fuzzy

Da mesma forma que para a nova abordagem do algoritmo NEH com incertezas discutido no item 3.4, o exemplo acima também é puramente didático, portanto não será efetuado nenhuma mudança nos números *fuzzy* apresentados pelo sistema.

Nesta etapa, será realizado o mesmo procedimento apresentado no item 3.4, ou seja, será efetuado a programação de operações separadamente para cada um dos valores do TFN (otimista, maior possibilidade e pessimista), além dos valores do TFN representados pela sua desfuzzificação.

Ao considerar-se estes valores separadamente, tem-se, então, quatro matrizes com os tempos de processamento de cada operação mostrados nas figuras 3.34, 3.35, 3.36 e 3.37.

Peças (J _i)	Máquinas (M _j)	
	M1	M2
J1	27	18
J2	22.5	18
J3	27	22.5
J4	22.5	18
J5	27	22.5
J6	22.5	18
J7	27	36
J8	13.5	9

Figura 3.34 – Matriz com os tempos de processamento otimistas

Peças (J_i)	Máquinas (M_j)	
	M1	M2
J1	30	20
J2	25	20
J3	30	25
J4	25	20
J5	30	25
J6	25	20
J7	30	40
J8	15	10

Figura 3.35 – Matriz com os tempos de processamento de maior possibilidade

Peças (J_i)	Máquinas (M_j)	
	M1	M2
J1	36.6	24.4
J2	30.5	24.4
J3	36.6	30.5
J4	30.5	24.4
J5	36.6	30.5
J6	30.5	24.4
J7	36.6	48.8
J8	18.3	12.2

Figura 3.36 – Matriz com os tempos de processamento pessimistas

Peças (J_i)	Máquinas (M_j)	
	M1	M2
J1	30.9	20.6
J2	25.8	20.6
J3	30.9	25.8
J4	25.8	20.6
J5	30.9	25.8
J6	25.8	20.6
J7	30.9	41.2
J8	15.5	10.3

Figura 3.37 – Matriz com os tempos de processamento representados pela sua desfuzzificação

As programações das operações sugeridas pelo sistema, para cada uma dessas matrizes com os tempos de operação apresentadas anteriormente, são realizadas através dos procedimentos descritos no item 2.4.4 – *Algoritmo de JOHNSON*.

O resultado dessas programações, com a visualização dos gráficos de *Gantt*, são apresentadas na figura 3.38 a seguir.

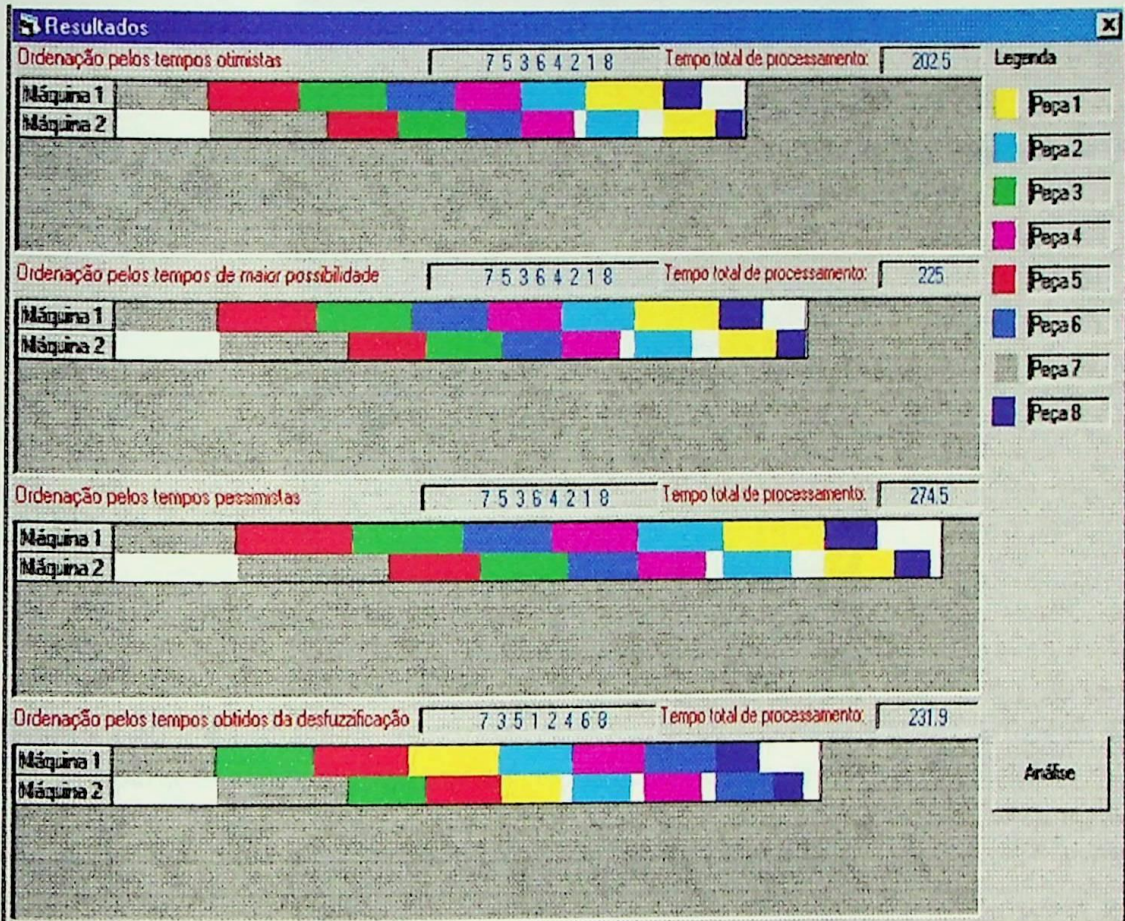


Figura 3.38 – Resultado da programação das operações

Pode-se reparar, que para este exemplo o sistema apresentou 2 resultados distintos. A mesma ordenação foi obtida para as matrizes de tempos de operações otimistas, de maior possibilidade e pessimistas (7, 5, 3, 6, 4, 2, 1, 8). Já para a matriz gerada pela desfuzzificação, a ordenação foi (7, 3, 5, 1, 2, 4, 6, 8).

Devido a existência de variações nas ordenações geradas pelo sistema para as matrizes de tempos de operações analisadas, é necessário efetuar uma comparação dessas ordenações.

Inicialmente compara-se a ordenação obtida pela matriz de tempos de operações obtidos da desfuzzificação, com os demais valores de tempos possíveis (otimistas, maior possibilidade e pessimistas). A análise desses resultados podem ser vistas nas figuras 3.39, 3.40 e 3.41.

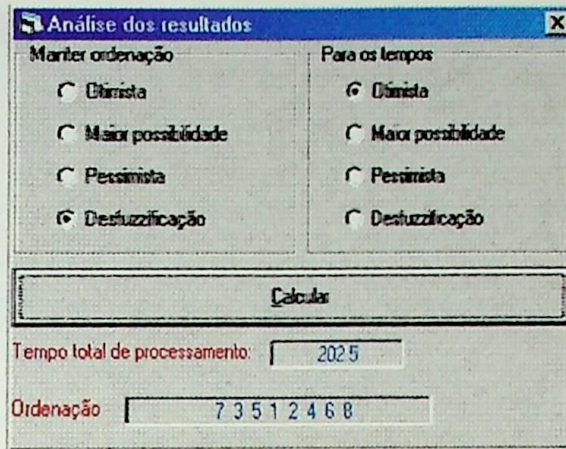


Figura 3.39 – Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos otimistas

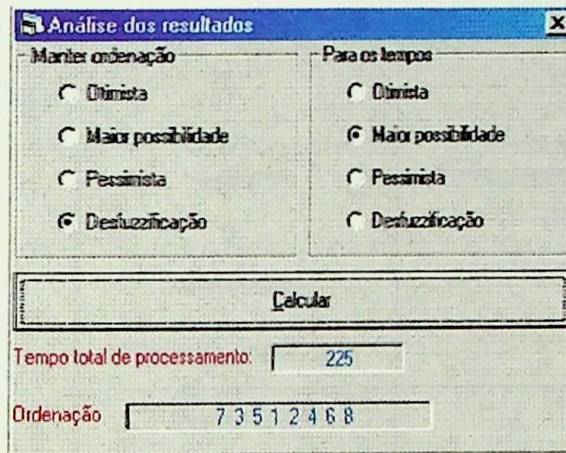


Figura 3.40 – Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos de maior possibilidade

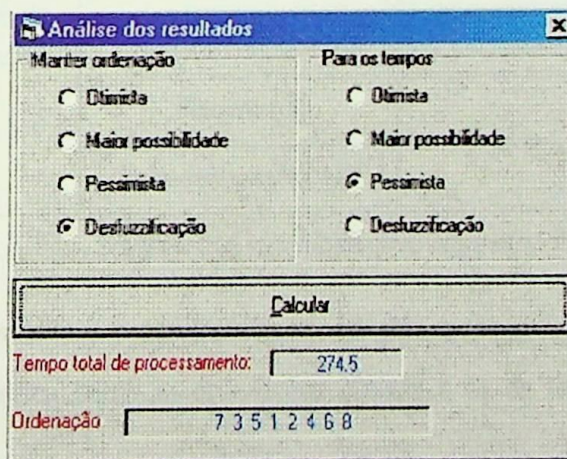


Figura 3.41 – Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos pessimistas

Dando continuidade a análise, posteriormente compara-se a ordenação obtida pela matriz de tempos de operações otimistas, de maior possibilidade e pessimistas (7, 5, 3, 6, 4, 2, 1, 8), considerando-se os valores de tempos obtidos pela desfuzzificação. O resultado desta análise pode ser visto na figura 3.42.

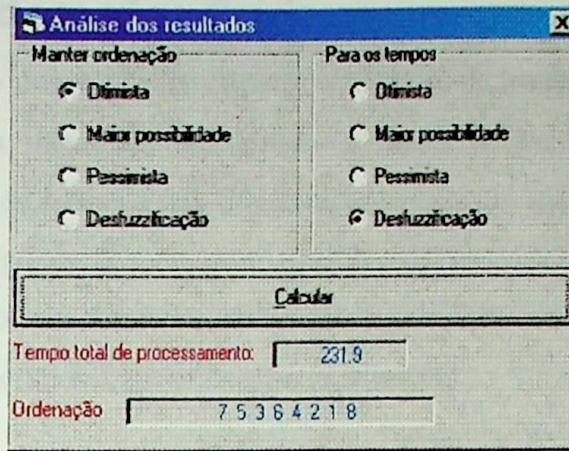


Figura 3.42 – Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação otimistas com os valores de tempos da desfuzzificação

A comparação entre os resultados da análise realizados podem ser visualizadas na tabela apresentada na figura 3.43.

Ordenação	Tempos	Tempo total de processamento (makespan)
Considerando a matriz obtida pela desfuzzificação 7, 3, 5, 1, 2, 4, 6, 8	Otimistas	202.5
	Maior possibilidade	225
	Pessimistas	274.5
	Pela desfuzzificação	231.9
Considerando as demais matrizes de tempo de operação 7, 5, 3, 6, 4, 2, 1, 8	Otimistas	202.5
	Maior possibilidade	225
	Pessimistas	274.5
	Pela desfuzzificação	231.9

Figura 3.43 – Análise dos resultados pelo algoritmo JOHNSON

Pode-se observar que o resultado final, ou seja o tempo total de produção, não se alterou com as variações entre as ordenações obtidas e os de tempos de processamento possíveis.

Portanto, a melhor ordenação para a resolução desse problema, poderia ser qualquer uma das duas ordenações apresentadas pelo sistema.

3.5 – Considerações finais

A utilização dos conceitos relativos a obtenção de tempo padrão de uma operação, para uma sugestão inicial na fuzzificação da matriz de tempos de processamento das atividades mostrou-se bastante útil. Esta abordagem permitiu criar uma ferramenta capaz de auxiliar o decisor a ponderar entre as variações de todos os tempos de operações existentes no processo em estudo, simplesmente considerando as variações de esforço e habilidade que ocorrem entre os operadores que executarão as atividades.

Como foi constatado, o resultado obtido pela comparação entre os números triangulares *fuzzy* através da desfuzzificação nem sempre é o melhor resultado (item 3.4.1), principalmente quando são utilizados algoritmos heurísticos na resolução desses problemas, pois estes não apresentam a melhor solução para o problema proposto, mas sim uma boa solução ao problema. Daí a necessidade de se avaliar todas as ordenações, obtidas das quatro matrizes de tempo de operação, na tentativa de se obter a melhor solução dentre aquelas apresentadas pelo sistema.

Não foi observado comportamento semelhante quando utilizado algoritmos ótimos na resolução de problemas de programação de operações com incertezas. Isto ocorre pois, como comentado no item 2.4, os algoritmos ótimos apresentam a melhor solução para o problema em estudo, ao contrário dos algoritmos heurísticos, que não apresentam a melhor, porém, uma boa solução ao problema. Daí a necessidade de se avaliar todas as ordenações obtidas na tentativa de se obter a melhor solução dentre as apresentadas pelo sistema.

A metodologia desenvolvida neste capítulo permitiu a elaboração de um software que representa o protótipo de uma nova ferramenta. Este software é apresentado no *Apêndice A – Modelo computacional* desta dissertação.

Neste capítulo, preocupou-se, também, em demonstrar a facilidade em se utilizar a ferramenta de análise desenvolvida.

Capítulo 4: Estudo de caso

4.1 – Considerações iniciais

Neste capítulo é mostrado um estudo de caso real utilizando a metodologia proposta no capítulo 3 e o software descrito no Apêndice A. Este estudo tem a finalidade de consolidar um melhor entendimento dos vários elementos discutidos na metodologia proposta neste trabalho.

O estudo de caso real foi baseado em informação obtidas na Indústria de Material Bélico do Brasil – IMBEL, Fábrica 5, localizada em Itajubá, MG. Esta empresa, vinculada ao Ministério do Exército, é tradicional fabricante de armas militares leves e principal fornecedora das Forças Armadas do Brasil, além dos principais exércitos de países aliados. Entre os produtos que fabrica destacam-se pistolas e fuzis.

4.2 – Estudo de caso real

Através de contatos realizados na empresa, conseguiu-se obter dados sobre alguns produtos que são produzidos obedecendo as regras de *flow-shop* permutacional descritas no *Capítulo 2 – Revisão bibliográfica*. Deste modo, chegou-se a seguinte matriz, mostrada na figura 4.1, com os tempos de operação para 5 tarefas e 2 máquinas.

<i>Tarefas</i>	<i>Maq1 – Torno automático</i>	<i>Maq2 – Torno revólver</i>
31	5490	1815
32	4690	4820
37	3690	1815
74	3690	2215
117	8490	3015

Figura 4.1 – Matriz com os tempos de processamento fornecida pela empresa

Os tempos de operação apresentados na figura 4.1 estão descritos em minutos e são os tempos necessários para a fabricação de um lote de 2000 peças. Os números apresentados na coluna *Tarefas* são os códigos de cada uma das peças fabricadas na empresa. Cabe ressaltar

também que os tempos de preparação para cada máquina (*setup-times*) foram acrescentados nos tempos de processamento de cada operação.

Por apresentar somente duas máquinas, estes dados podem ser utilizados pelos dois algoritmos descritos no *Capítulo 3 – Metodologia proposta*.

4.2.1 – Resolução pelo algoritmo NEH

A entrada de dados para a resolução do problema pelo algoritmo NEH se dá conforme mostrado na figura 4.2, onde os tempos de processamento, número de máquinas e números de tarefas foram mostrados anteriormente na figura 4.1.

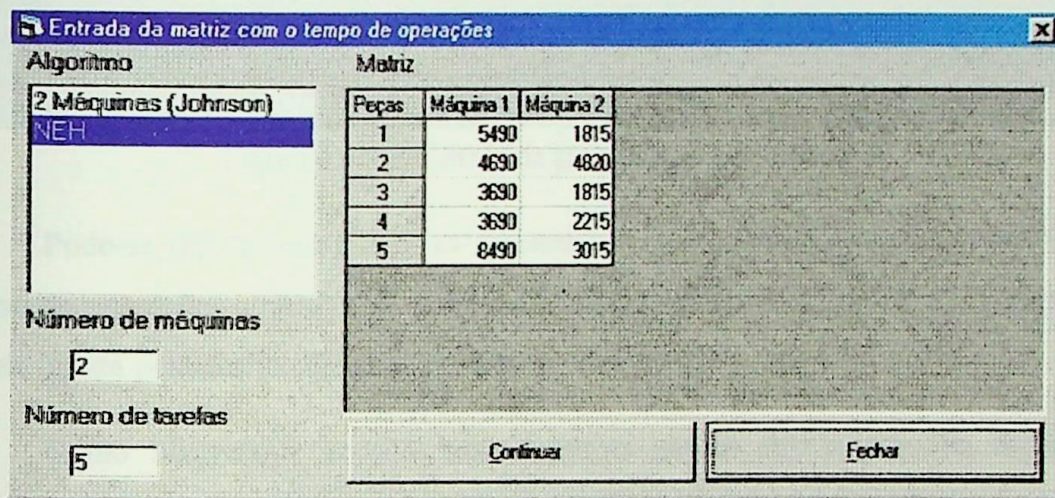


Figura 4.2 – Entrada da matriz com os tempos de operação

Esses tempos de processamento são transformados em números fuzzy, conforme o processo de fuzzificação descrito no item 3.2 e mostrados na figura 4.3 abaixo.

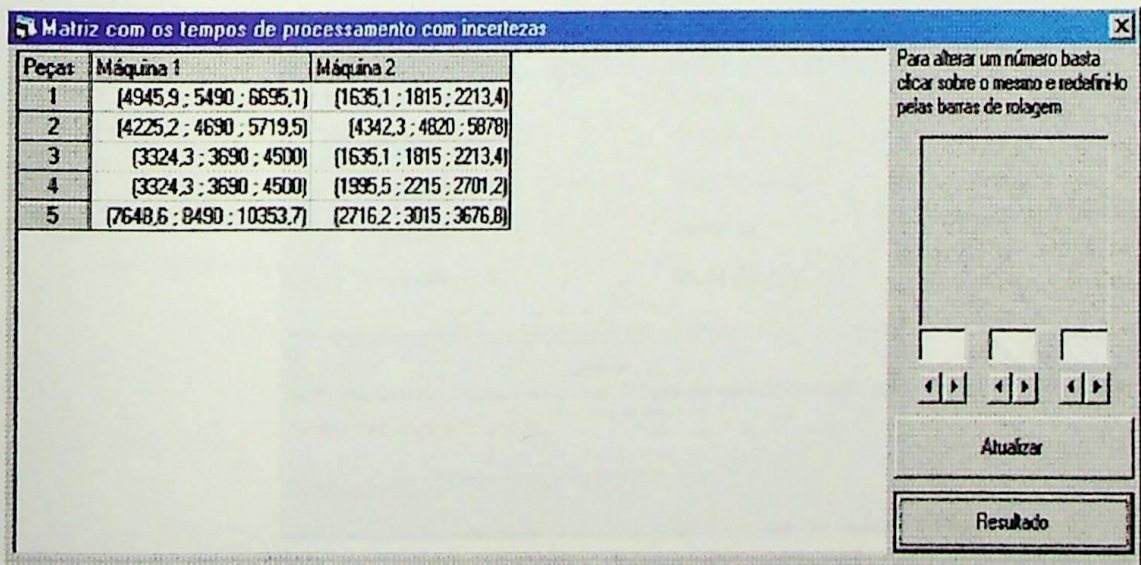


Figura 4.3-Matriz com os tempos de operação fuzzy

Foi questionado ao gerente de produção do setor onde os dados foram coletados se a variação dos tempos de operação entre os tempos otimistas e pessimistas estavam satisfatórias com a realidade da linha de produção. Este gerente, que no caso é o nosso especialista, confirmou os dados apresentados pelo sistema e não sugeriu nenhuma mudança neste tempos de operação.

A programação da produção sugerida pelo sistema, sem a visualização dos gráficos de *Gantt* é mostrada na figura 4.4 abaixo.

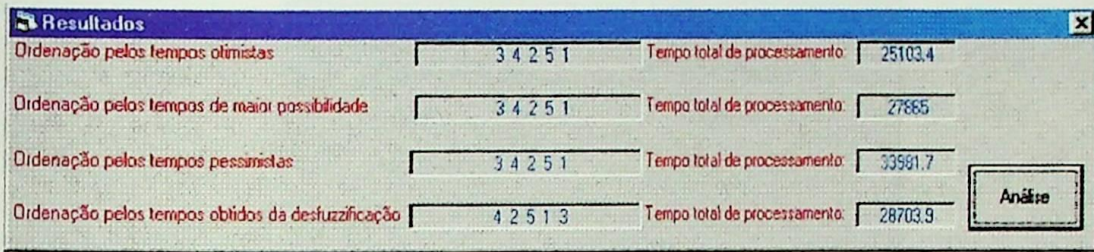


Figura 4.4 – Resultado da programação das operações

Pode-se reparar que o sistema apresentou a mesma ordenação para as matrizes de tempo de operações otimistas, de maior possibilidade e pessimistas (3, 4, 2, 5, 1). Já para a matriz obtida pela desfuzzificação, a ordenação foi diferente (4, 2, 5, 1, 3).

Como ocorreram variações nas ordenações geradas pelo sistema, foi necessário realizar a análise dessas ordenações conforme descrito no item 3.4 da metodologia.

Inicialmente, comparou-se a ordenação obtida pelas matrizes de tempos de operação otimistas, maior possibilidade e pessimista (3, 4, 2, 5, 1), considerando-se os valores de tempos de operação obtidos pela desfuzzificação. O resultado pode ser visto na figura 4.5.

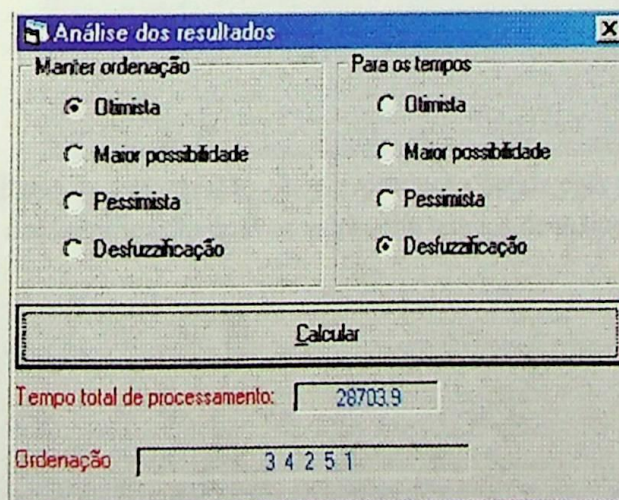


Figura 4.5 - Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação otimistas com os valores de tempos da desfuzzificação

Observa-se que no resultado final o tempo total de produção não se alterou, mesmo considerando-se a ordenação otimista para os tempos obtidos pela desfuzzificação.

Posteriormente, comparou-se a ordenação gerada pela matriz de tempos de operação obtida pela desfuzzificação (4, 2, 5, 1, 3), considerando-se os valores de tempo de operação otimista, maior possibilidade e pessimista. Os resultados podem ser vistos nas figuras 4.6, 4.7 e 4.8.

Análise dos resultados	
Manter ordenação <input type="radio"/> Otimista <input type="radio"/> Maior possibilidade <input type="radio"/> Pessimista <input checked="" type="radio"/> Desfuzzificação	Para os tempos <input checked="" type="radio"/> Otimista <input type="radio"/> Maior possibilidade <input type="radio"/> Pessimista <input type="radio"/> Desfuzzificação
Calcular	
Tempo total de processamento:	25103.4
Ordenação	4 2 5 1 3

Figura 4.6 – Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos otimistas

Análise dos resultados	
Manter ordenação <input type="radio"/> Otimista <input type="radio"/> Maior possibilidade <input type="radio"/> Pessimista <input checked="" type="radio"/> Desfuzzificação	Para os tempos <input type="radio"/> Otimista <input checked="" type="radio"/> Maior possibilidade <input type="radio"/> Pessimista <input type="radio"/> Desfuzzificação
Calcular	
Tempo total de processamento:	27865
Ordenação	4 2 5 1 3

Figura 4.7 – Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos de maior possibilidade

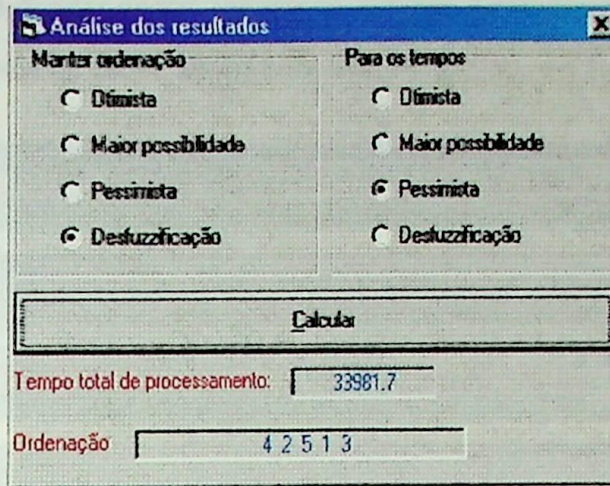


Figura 4.8 – Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos pessimistas

Observa-se que o resultado final, ou seja o tempo total de produção, não se alterou para nenhum dos tempos analisados, mesmo com as mudanças efetuadas nos valores de tempos de operação. Estas comparações entre os resultados podem ser melhor visualizadas na tabela apresentada na figura 4.9.

Ordenação	Tempos	Tempo total de processamento (makespan)
Considerando a matriz de tempo obtido pela desfuzzificação 3, 4, 2, 5, 1	Otimistas	25103.4
	Maior possibilidade	27865
	Pessimistas	33981.7
	Pela desfuzzificação	28703.9
Considerando as demais matrizes de tempo 4, 2, 5, 1, 3	Otimistas	25103.4
	Maior possibilidade	27865
	Pessimistas	33981.7
	Pela desfuzzificação	28703.9

Figura 4.9 – Análise dos resultados pelo algoritmo NEH

Concluiu-se, assim que para esta análise qualquer uma das duas ordenações apresentadas como resposta, não interfeririam no resultado final, ou seja, no tempo total de produção (*makespan*).

Quando isto ocorre, o gerente de produção, poderá tomar uma decisão baseando-se no tempo de ocupação de cada máquina, levando-se em consideração outras atividades que são realizadas ao mesmo tempo na linha de produção.

Deste modo, o decisor poderá solicitar ao *software*, a programação de operações com visualização dos gráficos de *Gantt*, mostrado na figura 4.10.

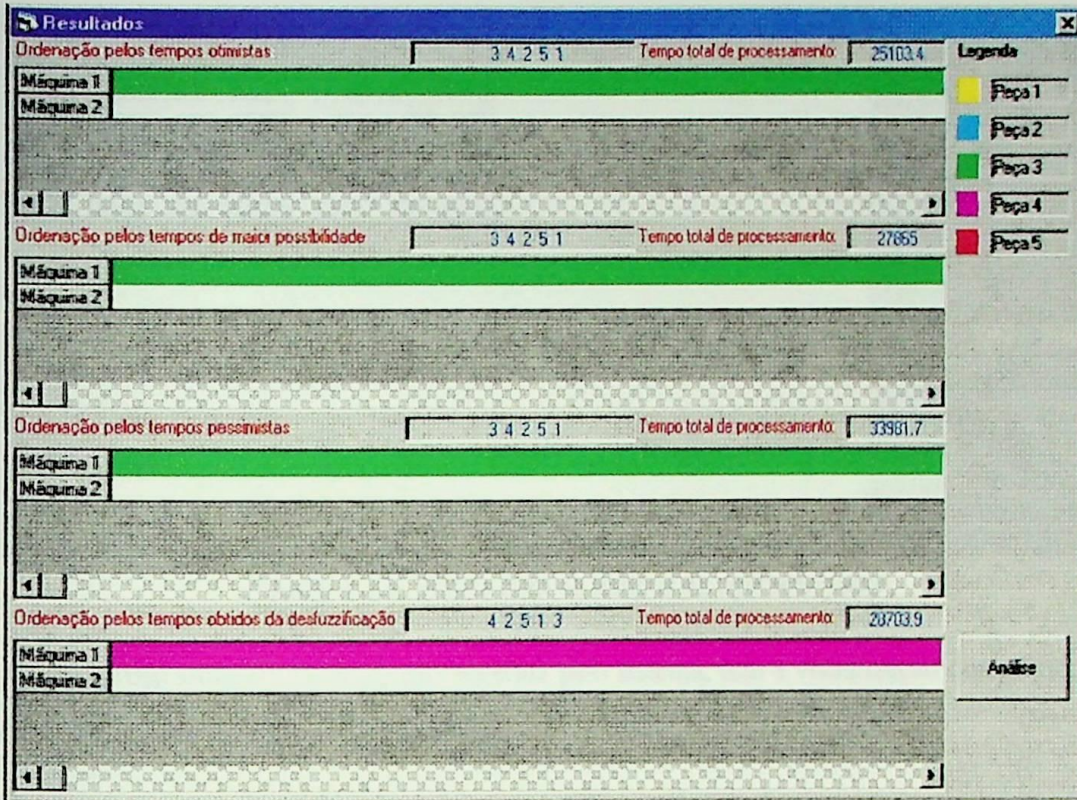


Figura 4.10 – Programação de operações com a visualização dos gráficos de *Gantt*

4.2.2 – Resolução pelo algoritmo de JOHNSON

Os valores dos tempos de processamento de cada operação com a inclusão dos tempos de *setup* utilizados como entrada de dados para a resolução do problema pelo algoritmo de JOHNSON foi apresentado no item 4.2.1, e está mostrada na figura 4.11 abaixo.

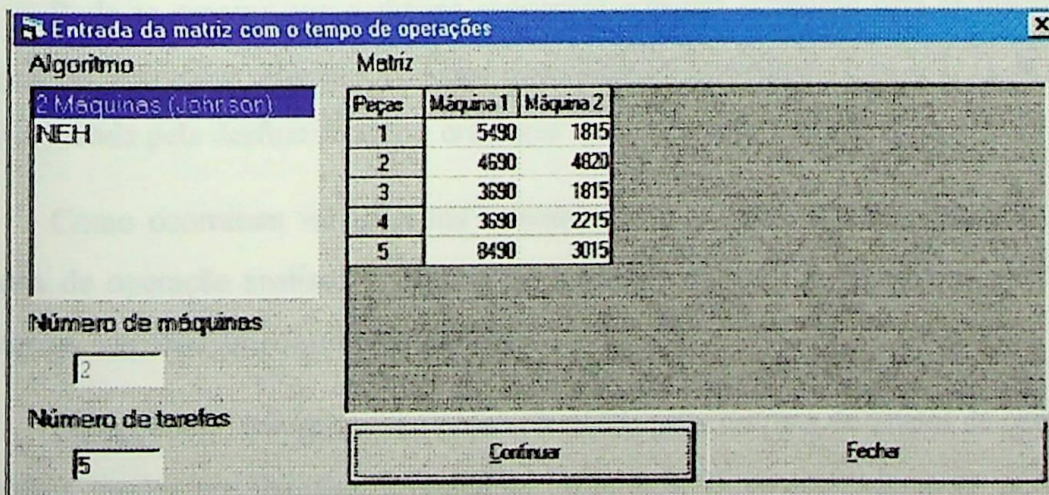


Figura 4.11 - Entrada da matriz com os tempos de operação

Esses tempos de processamento são transformados em números fuzzy, conforme descrito no item 3.2 e são mostrados na figura 4.12 abaixo.

Peças	Máquina 1	Máquina 2
1	{4945,9 ; 5490 ; 6695,1}	{1635,1 ; 1815 ; 2213,4}
2	{4225,2 ; 4690 ; 5719,5}	{4342,3 ; 4820 ; 5878}
3	{3324,3 ; 3690 ; 4500}	{1635,1 ; 1815 ; 2213,4}
4	{3324,3 ; 3690 ; 4500}	{1995,5 ; 2215 ; 2701,2}
5	{7648,6 ; 8490 ; 10353,7}	{2716,2 ; 3015 ; 3676,8}

Figura 4.12- Matriz com os tempos de operação fuzzy

A programação da produção sugerida pelo sistema, sem a visualização dos gráficos de Gantt é mostrada na figura 4.13 a seguir.

Ordenação pelos tempos otimistas	2 5 4 3 1	Tempo total de processamento:	25103,4
Ordenação pelos tempos de maior possibilidade	2 5 4 3 1	Tempo total de processamento:	27865
Ordenação pelos tempos pessimistas	2 5 4 3 1	Tempo total de processamento:	33981,7
Ordenação pelos tempos obtidos da desfuzzificação	2 5 4 1 3	Tempo total de processamento:	28703,9

Figura 4.13 – Resultado da programação das operações

Pode-se reparar que o sistema apresentou a mesma ordenação para as matrizes de tempos de operações otimistas, de maior possibilidade e pessimistas (2, 5, 4, 3, 1). Já para a matriz gerada pela desfuzzificação a ordenação foi sucintamente diferente (2, 5, 4, 1, 3).

Como ocorreram variações nas ordenações geradas pelo sistema para as matrizes de tempos de operação analisados, também foi necessário realizar a análise dessas ordenações conforme efetuado anteriormente no item 4.2.1.

Inicialmente, comparou-se a ordenação obtida pelas matrizes de tempos de operações otimistas, maior possibilidade e pessimista (2, 5, 4, 3, 1), considerando-se os valores de tempos de operação obtidos pela desfuzzificação. O resultado pode ser visto na figura 4.14.

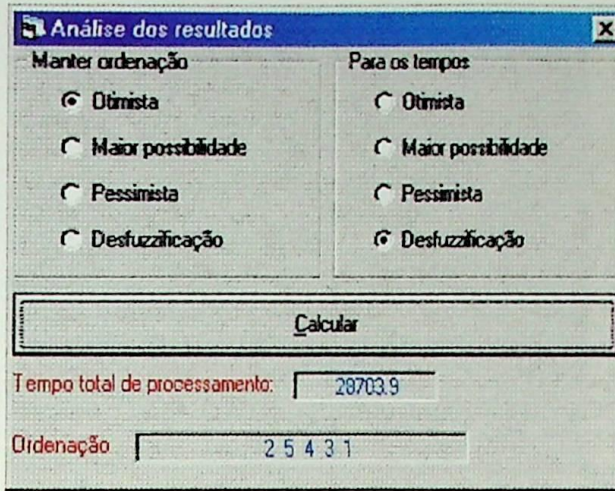


Figura 4.14 – Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação otimistas com os valores de tempos da desfuzzificação

Observa-se que o resultado final, o seja o tempo total de produção não se alterou, mesmo considerando-se a ordenação otimista para os tempos obtidos pela desfuzzificação.

Posteriormente, comparou-se a ordenação obtida da matriz de tempos de operação gerados pela desfuzzificação (2, 5, 4, 1, 3), com os tempos de operações otimistas, maior possibilidade e pessimistas. Os resultados podem ser vistos nas figuras 4.15, 4.16 e 4.17.

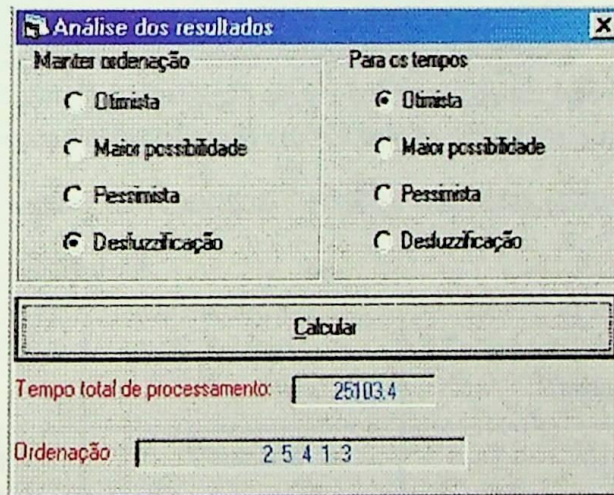


Figura 4.15 - Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos otimistas

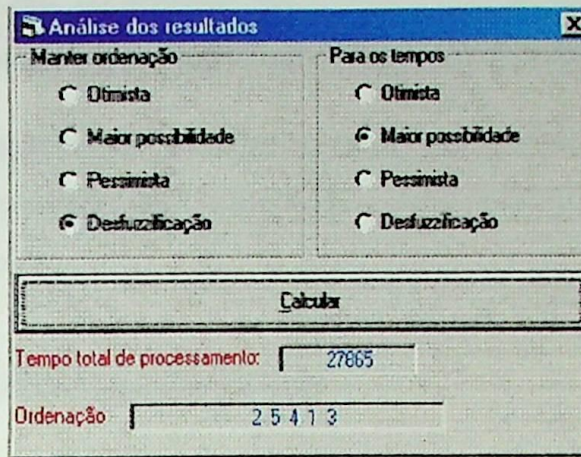


Figura 4.16 - Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos de maior possibilidade

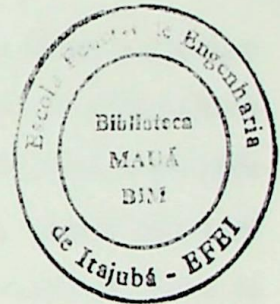
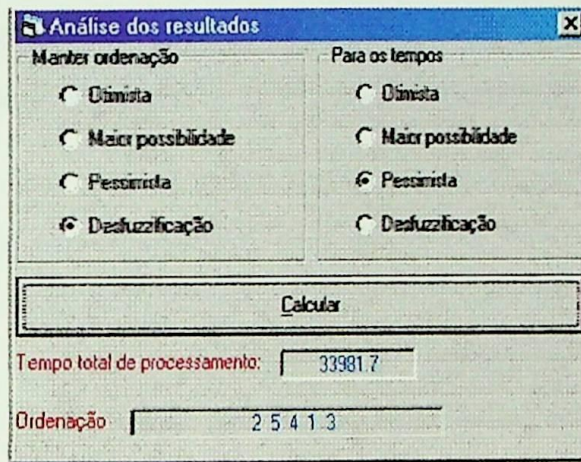


Figura 4.17 - Resultado da comparação entre a ordenação obtida pela matriz de tempos de operação da desfuzzificação com os valores de tempos pessimistas

A comparação entre os resultados da análise realizados podem ser visualizadas na tabela apresentada na figura 4.18.

Ordenação	Tempos	Tempo total de processamento (makespan)
Considerando a matriz obtida pela desfuzzificação 2, 5, 4, 1, 3	Otimistas	25103.4
	Maior possibilidade	27865
	Pessimistas	33981.7
	Pela desfuzzificação	28703.9
Considerando as demais matrizes de tempo de operação 2, 5, 4, 3, 1	Otimistas	25103.4
	Maior possibilidade	27865
	Pessimistas	33981.7
	Pela desfuzzificação	28703.9

Figura 4.18 – Análise dos resultados pelo algoritmo JOHNSON

Observa-se que no resultado final o tempo total de produção não se alterou para nenhum dos tempos analisados mesmo com as mudanças dos tempos de operação.

Concluiu-se, assim que para esta análise qualquer uma das ordenações apresentadas como solução pelo sistema não interferiram no resultado final, ou seja, no tempo total de produção (*makespan*).

4.3 – Considerações finais

Apesar do estudo de caso não abordar toda a potencialidade da metodologia desenvolvida, conforme apresentado no *Capítulo 3 – metodologia proposta*, o resultado gerado pelo sistema foi utilizado pela empresa, uma vez que esta não possuía uma ordenação definida na fabricação dos produtos que foram analisados neste capítulo.

Para problemas envolvendo duas máquinas é sempre interessante utilizar o algoritmo de JOHNSON, pois este método possui uma abordagem ótima. MOCELLIN (1992) afirma que para poucas máquinas o algoritmo NEH gera resultados próximos do ótimo, o que pôde ser visto no estudo de caso apresentado neste capítulo. Apesar do sistema apresentar ordenações diferentes, o tempo total de processamento das tarefas (*makespan*) foram os mesmos para as duas abordagens.

Capítulo 5: Conclusões e recomendações

5.1 – Considerações iniciais

Este último capítulo tem a finalidade de sintetizar os resultados obtidos nesta dissertação. Inicialmente serão abordadas as contribuições realizadas por esse trabalho. A seguir serão relacionados temas que podem ser desenvolvidos em trabalhos futuros.

Cabe ressaltar que, durante o desenvolvimento desta dissertação, alguns artigos sobre as diversas partes do tema abordado, foram publicados e apresentados em importantes eventos e congressos. Isto permitiu uma avaliação parcial deste trabalho de tal modo a comprovar a validade das propostas apresentadas. Uma relação dessas publicações encontra-se no *Anexo B – Artigos originados dessa dissertação*.

5.2 – Contribuições do trabalho

Segundo MONTEVECHI (1995), muitas vezes o conhecimento sobre os processos de manufatura está na “*cabeça*” das pessoas que tratam diretamente com este assunto, sendo muito difícil a formalização destes conhecimentos.

KUSIAK (1990), afirma que um dos atributos humanos considerados é a habilidade em ponderar informações incertas. Deste modo, uma metodologia que se preocupe com este atributo, está dando um passo para a concepção de sistemas que possam usar a maneira do homem em resolver problemas.

Portanto, a utilização de um processo automático para a fuzziificação da matriz com os tempos de processamento das atividades, com base nas abordagens de obtenção de tempo padrão de operação, mostra-se bastante útil. Este processo reduz substancialmente o trabalho do decisor, além de ainda permitir total interação com os tempos de operação *fuzzy* gerados pelo sistema, possibilitando, assim, uma formalização do conhecimento destes especialistas que ficarão arquivados no sistema, e facilitando a utilização das técnicas de incertezas no chão de fábrica.

A metodologia desenvolvida, neste trabalho, para a fuzziificação da matriz com os tempos de processamento das atividades, sujeito a postos de trabalho exclusivamente manuais, pode ser utilizada em problemas de programação de operações que envolvem

qualquer tipo de padrão de fluxo de produção (*flow-shop*, *job-shop*, máquina única, etc.), e não só ficar restrita a problemas de *flow-shop* permutacional em sistemas intermitentes.

A separação dos tempos de processamento *fuzzy* em quatro matrizes de tempos de operação (otimista, maior possibilidade, pessimista e pela desfuzzificação), quando trabalha-se com incertezas, através dos números *fuzzy* triangulares, em problemas de programação de operações, mostrou-se mais eficiente que simplesmente utilizar a comparação entre os números *fuzzy* através da desfuzzificação desses números, principalmente nos algoritmos heurísticos.

Isto ocorre pois os algoritmos com abordagem heurísticas não apresentam a melhor solução para o problema proposto, mas sim uma boa solução ao problema. Daí a necessidade de se avaliar todas as ordenações, obtidas das quatro matrizes de tempo de operação, na tentativa de se obter a melhor solução dentre aquelas apresentadas pelo sistema.

Ao contrário, ao se utilizar algoritmos ótimos, a comparação entre os números *fuzzy* através da desfuzzificação mostrou-se tão eficiente quanto a nova abordagem desenvolvida para a resolução de problemas de programação de operações com incertezas.

Entretanto MOCCELLIN (1992) afirma que não existe um algoritmo eficiente que forneça uma solução ótima, a não ser para o caso de 2 ou 3 máquinas como o algoritmo de JOHNSON (JOHNSON, 1954). Assim muitos métodos heurísticos têm sido propostos para a solução de problemas de programação de operações *flow-shop*.

Portanto a nova abordagem desenvolvida neste trabalho se torna relevante em problemas de programação de operação com incertezas que utilizam um tratamento heurístico em sua solução.

O *software* desenvolvido, batizado de *Fuzzy Scheduling*, descrito no Apêndice A – *Modelo computacional*, mostrou-se bastante capaz de realizar os problemas de programação de operações apresentados ao longo desta dissertação. Sua interação prática, fácil e amigável contribuíram para uma melhor compreensão da metodologia desenvolvida.

Este modelo computacional desenvolvido apresenta também uma vantagem ao permitir a visualização dos gráficos de *Gantt* para a ordenação obtida como resultado da programação de operações, pois estes gráficos possibilitam uma melhor visualização das capacidades produtivas de cada máquina.

Durante a fase de coleta de dados e realização do estudo de caso, constatou-se a facilidade de atribuição de incertezas, nos dados reais coletados, através do processo de fuzzificação proposto por este trabalho. O resultado gerado pelo estudo de caso foi utilizado pela empresa, pois esta não possuía nenhum critério de ordenação definido para a fabricação de seus produtos.

5.3 – Propostas para trabalhos futuros

Uma extensão natural deste trabalho seria desenvolver um processo de fuzzificação para gerar outros tipos de números *fuzzy*. Por exemplo, um número *fuzzy* trapezoidal, onde aos dois valores de maior possibilidade (a_2 e a_3), poderiam ser atribuídos os tempos médios cronometrados em função do K (expressão 3.1) para um operador com habilidade e esforço considerados *Boa* e *Normal* (Figura A2.1) respectivamente. Para o valor otimista (a_1), seria atribuído o tempo cronometrado em função do K para um operador com habilidade e esforço considerado *Excelente*. E, para o valor pessimista de um TrFN (a_4), seria atribuído o tempo cronometrado em função do K para um operador com habilidade e esforço considerado *Regular*.

Outros processos de desfuzzificação poderiam ser avaliados, na resolução de problemas de programação de operações, com esta nova abordagem. Por exemplo, o método do centróide, descrito em COX (1995).

A implementação de outros algoritmos de programação de operações com considerações de incertezas no modelo computacional desenvolvido é, também, uma fonte para novas pesquisas. E uma vez realizada estas implementações, a comparação entre estes algoritmos complementariam, também, este trabalho.

Apêndice: Modelo computacional

A.1 – Considerações iniciais

Será apresentado, neste capítulo, um modelo computacional protótipo que permitirá a implementação da metodologia proposta no capítulo 3 desta dissertação. Esta ferramenta recebeu o nome de *Fuzzy Scheduling* e foi desenvolvida pelo programa *Microsoft Visual Basic 5.0*¹.

Inicialmente é apresentado um diagrama com o intuito de facilitar o entendimento da lógica funcional do sistema implementado. E, para ilustrar a facilidade de interação como o usuário as principais telas de interface do *software* protótipo também são mostradas.

Cabe ressaltar que esta ferramenta foi apenas utilizada no estudo de caso apresentado no capítulo 4. Portanto, o desenvolvimento deste *software* terminou quando se chegou a uma ferramenta suficientemente capaz de testar a metodologia proposta.

Desta maneira, esta primeira versão, aqui apresentada, é a base para que novas versões, com um maior número de algoritmos com inserção de incertezas, possam ser implementados em trabalhos futuros.

A.2 – O software e suas principais telas

O fluxograma apresentado na figura A1.1 representa de maneira generalizada as principais etapas realizadas pelo modelo computacional proposto.

Inicialmente, o usuário pode optar por começar um novo estudo ou abrir um estudo existente. Se a opção escolhida for um novo estudo de caso, o usuário deverá fornecer a matriz com os tempos de operação de cada atividade e escolher qual algoritmo será utilizado na resolução do problema.

Vencida esta etapa, o sistema realiza automaticamente a fuzziificação da matriz com os tempos de operação, transformando-os em números triangulares *fuzzy*, sendo que o usuário tem total liberdade de alterar qualquer um desses números *fuzzy* gerados.

¹ Linguagem de programação comercializada pela Microsoft® Corporation

Finalmente, então, o sistema gera os resultados para o problema proposto. Se ocorrer mais de uma ordenação como resposta, deverá ser realizada uma análise dos possíveis resultados das ordenações geradas pelo sistema, conforme visto no item 3.4.

Se inicialmente o usuário optar por abrir um estudo de caso já existente, este poderá modificar as incertezas já existentes e/ou efetuar uma nova análise dos possíveis resultados gerados pelo *software*.

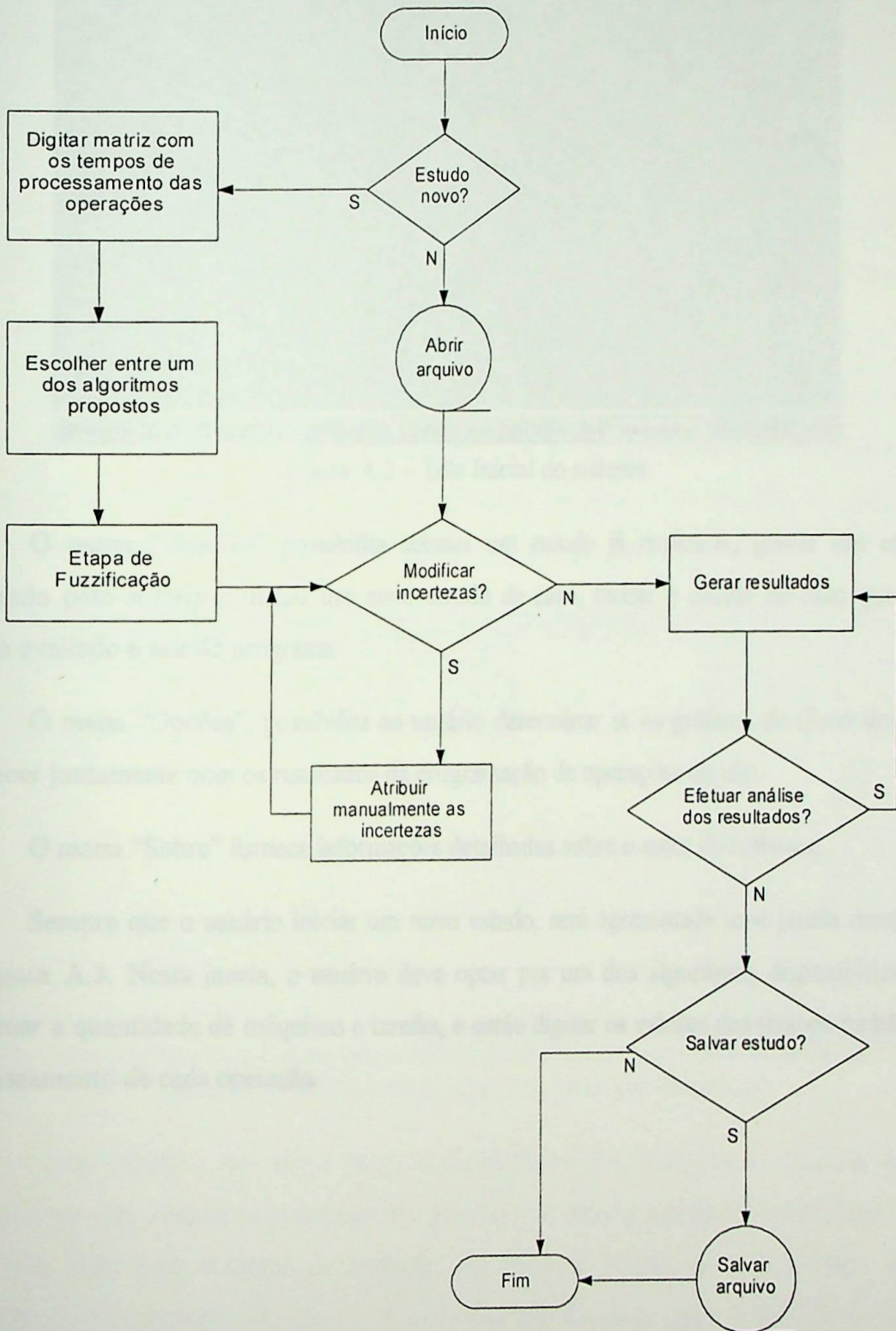


Figura A.1 – Fluxograma do modelo computacional proposto

A figura A.2 mostra a tela inicial do sistema. Nota-se que o programa possui três submenus, sendo estes: Arquivo, Opções e Sobre.

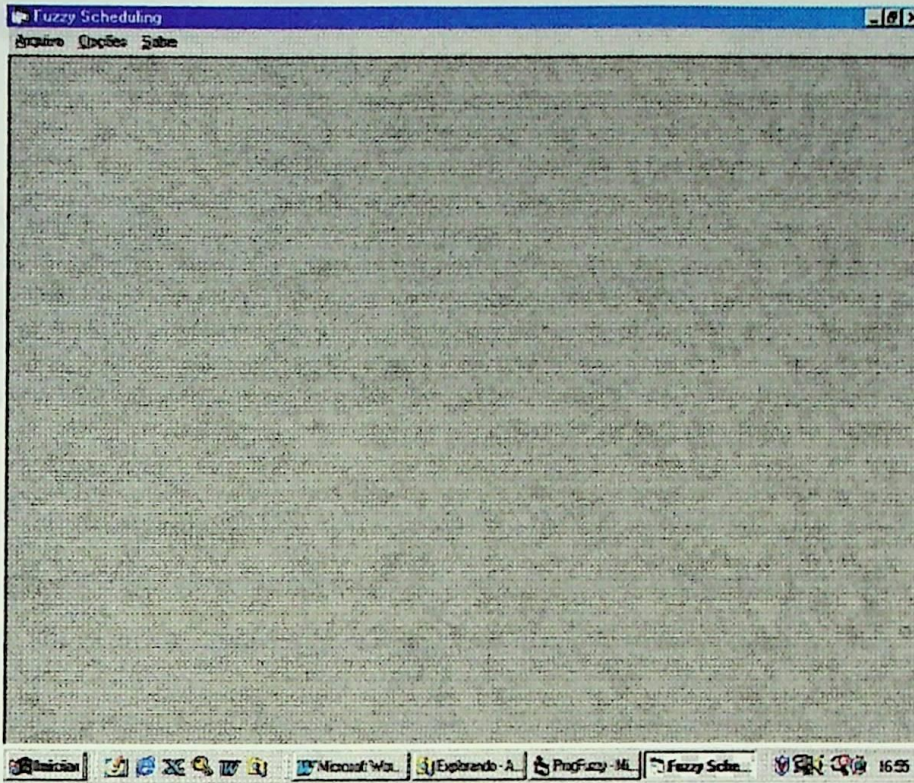


Figura A.2 – Tela Inicial do sistema

O menu “Arquivo” possibilita acessar um estudo já realizado, gravar um estudo realizado pelo *software*, iniciar um novo estudo de caso, fechar o estudo de caso que está sendo avaliado e sair do programa.

O menu “Opções”, possibilita ao usuário determinar se os gráficos de *Gantt* deverão aparecer juntamente com os resultados da programação de operações ou não.

O menu “Sobre” fornece informações detalhadas sobre o autor do software.

Sempre que o usuário iniciar um novo estudo, será apresentada uma janela mostrada na figura A.3. Nesta janela, o usuário deve optar por um dos algoritmos disponibilizados, informar a quantidade de máquinas e tarefas, e então digitar os valores dos tempos padrão de processamento de cada operação.

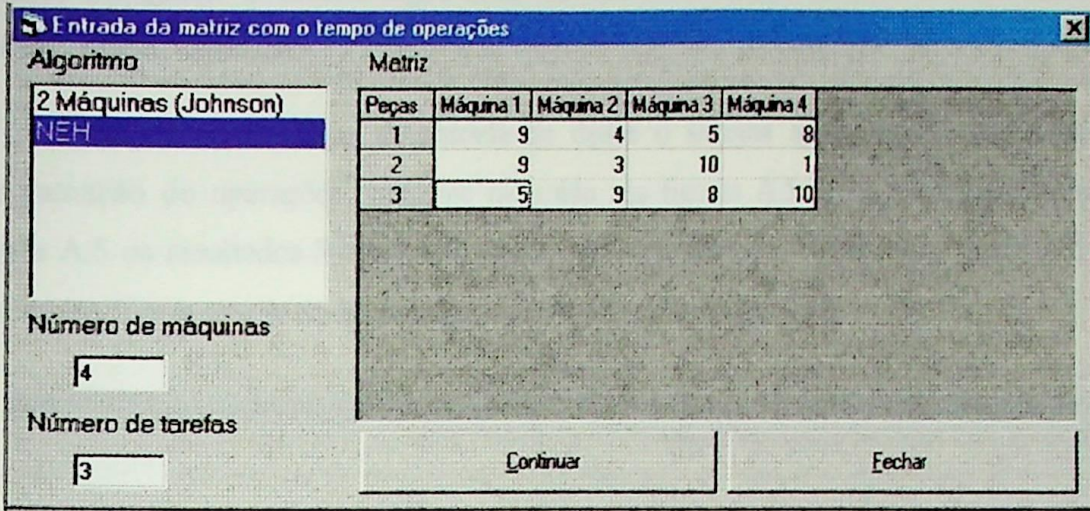


Figura A.3 – Janela para iniciar um novo estudo

É importante mencionar que atualmente o modelo trabalha com 2 algoritmos (NEH e JOHNSON), que foram analisados e desenvolvidos na metodologia deste trabalho. Entretanto o software foi elaborado prevendo-se uma expansão para vários outros algoritmos.

Vencida esta etapa, o sistema gera automaticamente os números triangulares *fuzzy* para todos os tempos de operação. Esta fuzzificação é realizada com base na teoria de obtenção de tempo padrão discutida no item 3.2 da metodologia proposta. A matriz com os tempos de processamento de cada operação *fuzzy* pode ser vista na figura A.4.

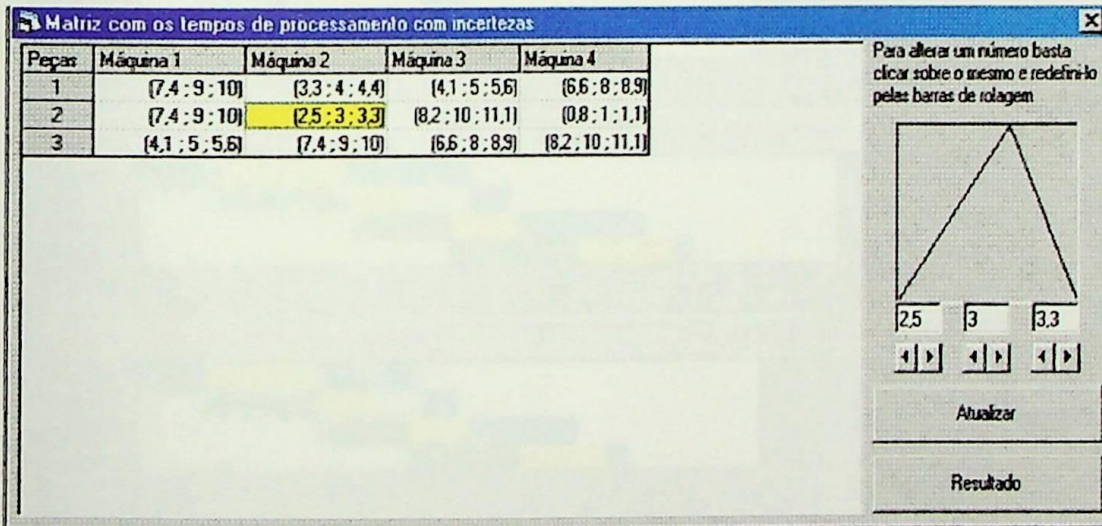


Figura A.4 – Janela com os tempos de processamento *fuzzy*

Como previsto pela teoria *fuzzy*, ainda na figura A.4, o decisor é capaz de alterar qualquer um dos tempos de processamento gerados pelo sistema através da fuzzificação. Para tal, basta selecionar o tempo de operação que deseja-se alterar, onde este tempo ficará marcado com a coloração amarela e será exibido ao lado direito da janela sobre a forma de um

número triangular. O usuário digita os novos valores nas caixas de texto correspondentes e clica no botão *Atualizar*.

Completada a etapa de entrada de dados o sistema apresenta os resultados da programação de operações conforme mostrado nas figuras A.5 e A.6. Percebe-se que na figura A.5 os resultados foram apresentados sem os gráficos de *Gantt*. Já na figura A.6, os resultados foram apresentados juntamente com seus gráficos de *Gantt*

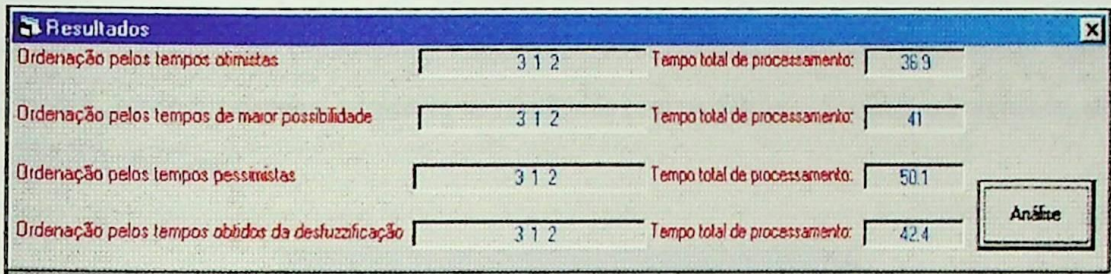


Figura A.5 – Janela de resultados sem os gráficos de *Gantt*

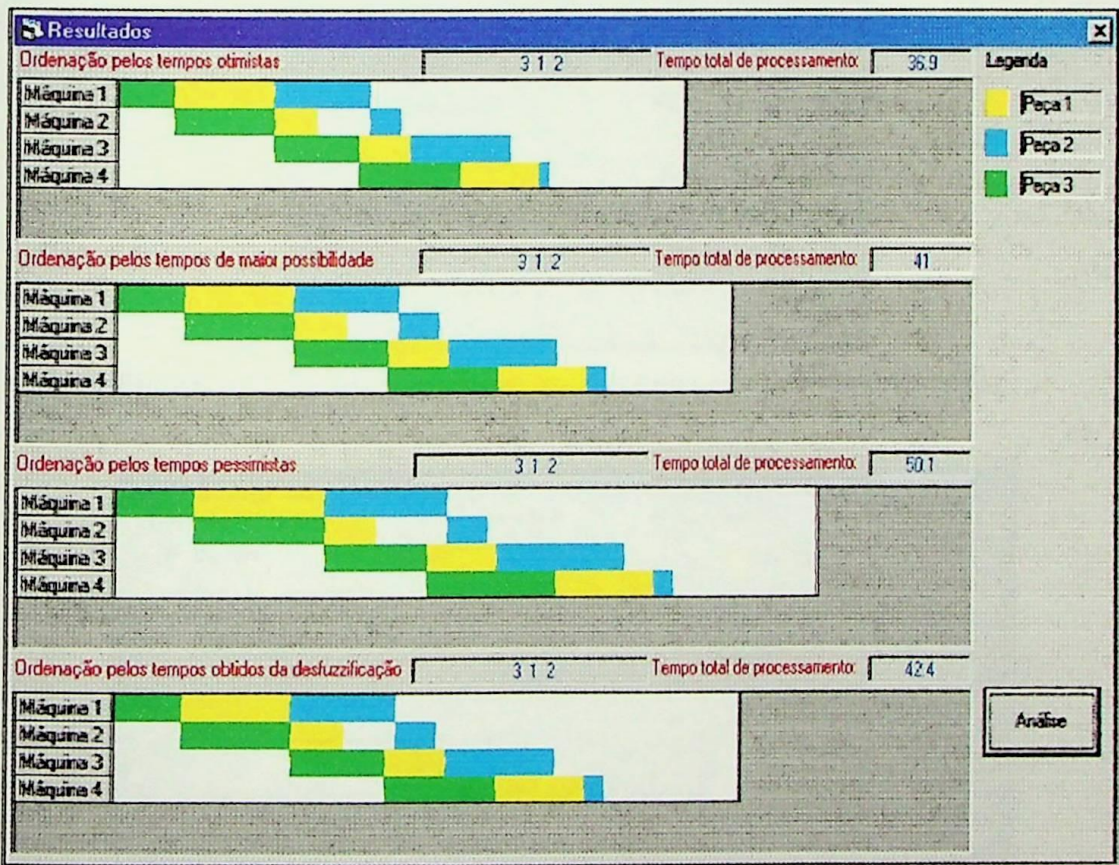


Figura A.6 – Janela de resultados com os gráficos de *Gantt*

Optou-se por possibilitar a apresentação dos resultados através dos gráficos de *Gantt*, pois estes permitem uma melhor idéia do tempo de ocupação de cada máquina.

Conforme apresentado no capítulo 3 – *Metodologia proposta*, percebe-se que existem quatro resultados de programação distintos. O primeiro é para a matriz de tempos considerada

otimistas, o segundo, para a matriz de tempos de maior possibilidade, o terceiro, para a matriz de tempos pessimistas, e por último, para a matriz de tempos obtidos pela desfuzzificação dos números *fuzzy*.

Nesta janela, os quatro resultados da programação de operações apresentaram a mesma ordenação. Entretanto, nem sempre as ordenações obtidas são as mesmas (item 3.4). Neste caso é necessário efetuar a análise das possíveis ordenações e escolher dentre as diversas possibilidades, aquela que apresenta um melhor resultado do ponto de vista do tempo total de processamento (*makespan*). Esta análise é realizada na janela apresentada nas figuras A.7 e A.8, onde a primeira apresenta os resultados com os gráficos de *Gantt* e a segunda, sem os gráficos de *Gantt*.

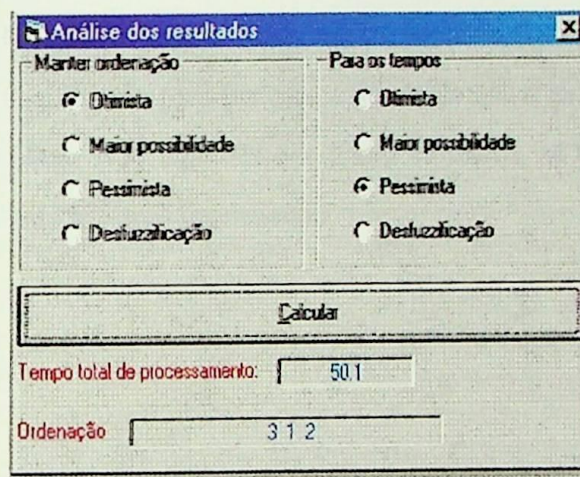


Figura A.7 – Janela para análise dos resultados sem os gráficos de *Gantt*

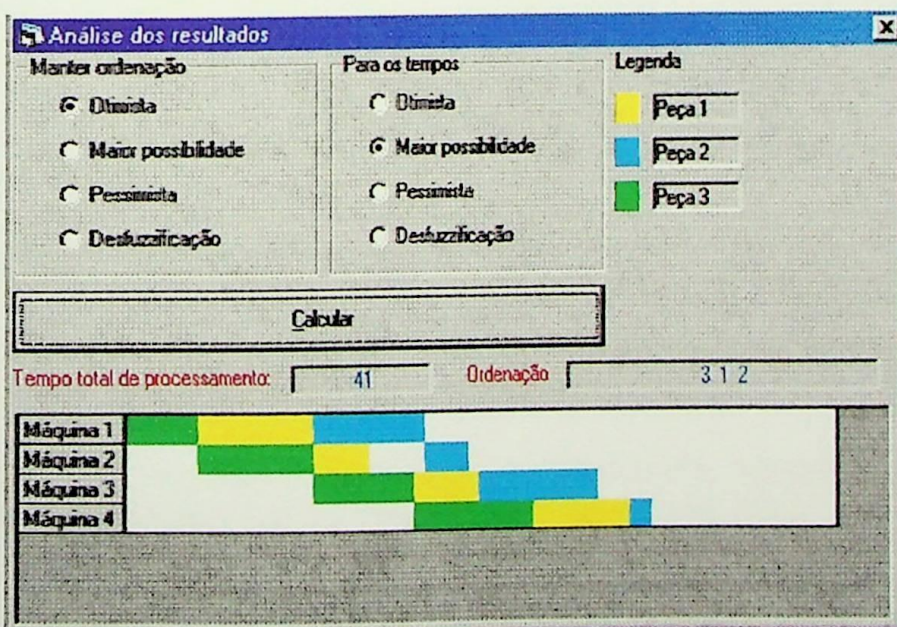


Figura A.8 – Janela para análise dos resultados com os gráficos de *Gantt*

A.3 – Considerações finais

O *Fuzzy Scheduling* foi desenvolvido utilizando-se a técnica de programação por janelas. Ou seja, na prática, cada fase do fluxograma mostrado na figura A.1 corresponde a uma, ou mais, janela do programa. Com isto busca-se uma maior concentração do usuário, através do controle de suas opções.

Percebe-se, também, que existiu uma grande preocupação no desenvolvimento desta ferramenta em possibilitar uma interação com usuário bastante amigável. Isto pôde ser observado nas figuras apresentadas neste capítulo.

O modelo computacional apresentado neste capítulo foi desenvolvido em um processador *Pentium* de 200 MHz, 32 Mb de RAM (*Random Access Memory*) e monitor SVGA (*Super Video Graphics Array*). Porém, um processador *486 DX2* (de apenas 66 MHz) com 8 Mb de RAM é suficiente para sua utilização, desde que o sistema operacional seja o *Windows 9x* (ou *Windows NT*) uma vez que foram utilizados, na elaboração do *software*, recursos presentes em tais sistemas operacionais.

Referências bibliográficas

- BASS, S.; KWAKERNAAK, H. *Rating and Ranking of Multiple Aspect Alternatives Using Fuzzy Sets*. Automatica, Vol. 13, pp. 47-58, 1977.
- BAKER, K. R. *Introduction to Sequencing and Scheduling*. John Wileys & Sons, Inc., New York, 1974.
- BARNES, R. M. *Estudo de Movimentos e de Tempos: Projeto e Medida de Trabalho*. Tradução da 6ª edição americana. Editora Edgard Blücher Ltda, 1986
- BETIOLI, M. A.; BELHOT, R. V. *Uma Base de Conhecimento para o Problema de Máquinas Paralelas*. 16º Encontro Nacional de Engenharia de Produção, Piracicaba, outubro, 1996.
- BURBIDGE, J. L. *The Introduction of Group Technology*. William Heinemann Ltd., London, 1975.
- CAMPBELL, H. G.; DUDEK, R. A.; SMITH, M. L. *A Heuristic Algorithm for the n-Job, m-Machine Sequencing Problem*. Management Science, Vol 16/B, pp. 630-637, 1970.
- CONTADOR, J. L. *Gerenciamento de Projetos com PERT e CPM*. Gestão de Operações – A Engenharia de Produção a Serviço da Modernização da Empresa, Fundação Vanzolini e Editora Edgard Blücher Ltda., 1ª edição. 1997.
- CONWAY, R. W.; MAXWELL, W. L.; MILLER, L.W. *Theory of Scheduling*. Addison-Wesley, 1967.
- COX, E. *The Fuzzy Systems Handbook*. Academic Press Limited, Estados Unidos, 1994.
- COX, E. *Fuzzy Logic for Business and Industry*. Incorporated Charles River Media, Rockland, Estados Unidos, 1995.
- CURY, R. M. *Uma Abordagem Difusa para o Problema de Flow Shop Scheduling*. Tese de Doutorado. Universidade Federal de Santa Catarina, Florianópolis, 1999.
- DANNENBRING, D. G. *An Evaluation of Flow-shop Sequencing Heuristics*. Management Science, Vol 23, pp. 1174-1182, 1977.
- EMMONS, H. *One Machine Sequencing to Minimize Mean Flow Time with Minimum Number Tardy*. Naval Research Logistics Quarterly, Vol. 22, Nº 3, September, 1975.
- FORTEMPS, P. *Jobshop Scheduling with Imprecise Durations: A Fuzzy Approach*. IEEE Transactions on Fuzzy Systems, Vol. 5, NO. 4, November, 1997.
- FORTEMPS, P.; ROUBENS, M. *Ranking and Defuzzification Methods Based on Area Compensation*. Fuzzy Sets and Systems, Vol. 82, pp. 319-330, 1996.
- GLOVER, F.; *Tabu Search – Part I*. ORSA Journal on Computing, Vol. 1, pp. 190-206, 1989.
- GOMIDE, F. A. C.; GUDWIN, R. R. *Modelagem, Controle, Sistemas e Lógica Fuzzy*. SBA Controle & Automação, Vol. 4, Nº 3, Setembro/Outubro, 1994.

- HAPKE, M.; JASKIEWICZ, A.; SLOWINSKI, R. *Fuzzy Project Scheduling System for Software Development*. Fuzzy Sets Systems., Vol. 67, pp. 101-117, 1994.
- HECK, H.; ROBERTS, S. *A Note on the Extension of Result on Scheduling with Secondary Criteria*. Naval Research Logistics Quarterly, Vol. 19, Nº 2, June, 1972.
- HOLLAND, J. H. *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, Mich, 1975.
- IGNALL, E. and SCHRAGE, L.E. *Application of Branch and Bound Technique to some Flow-Shop Problem*, Operations Research Vol. 13, pp. 400-412, 1965.
- ISHII, H.; TADA, M.; MASUDA, T. *Two Scheduling Problems with Fuzzy Due Dates*. Fuzzy Sets Syst., Vol. 46, pp. 339-347, 1992.
- ISHIBUCHI, H.; YAMAMOTO, N.; MURATA, T.; TANAKA, H. *Genetic Algorithms and Neighborhood Search Algorithms for Fuzzy Flow-shop Scheduling Problems*. Fuzzy Sets System., Vol. 67, pp. 81-100, 1994.
- JOHNSON, S. M. *Optimal Two and Three Stage Production Scheduling with Setup Times Included*. Naval Research Logistics Quarterly, Vol. 1, pp. 61-68, 1954.
- JOHNSON, L. A.; MONTGOMERY, D. C. *Operations Research in Production Planning, Scheduling and Inventory Control*. Wiley, New York, 1974.
- KAUFMANN, A. *Introduction to the Theory of Fuzzy Subsets*. Academic Press Limited, Estados Unidos, 1975.
- KAUFMANN, A. and M. M. GUPTA, *Fuzzy Mathematical Models in Engineering and Management Science*. Elsevier Science Publishers, New York/NY, USA, 1988.
- KIRKPATRICK, S.; GELATT, C. D. Jr.; VECCHI, M. P. *Optimization by Simulated Annealing*. Science Vol. 220, pp. 671-680, 1983.
- KUSIAK, A. *Intelligent Manufacturing System*. New Jersey – Prentice Hall, Inc, 1990.
- LÓPES, O. C.; BARCIA, R. M.; EYADA, O. *Problema de Programação da Produção – um Esquema de Classificação*. Produção, Vol. 5, Nº 2, novembro, 1995.
- MACCARTHY, B. L.; LIU, J. *Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling*. International Journal of Production Research, Vol. 31, NO. 1, pp. 59-79, 1993.
- MAZZUCCO JR, J *Uma abordagem Híbrida do Problema da Programação da Produção Através dos algoritmos Simulated Annealing e Genético*. Tese de Doutorado, Universidade Federal de Santa Catarina, Florianópolis, 1999.
- MCCAHERN, C. S.; LEE, E. S. *Fuzzy Job Sequencing for a Flow Shop*. European. Journal Operational Research, Vol. 62, pp. 294-301, 1992.
- MCNEILL, F.M.; THRO, E. *Fuzzy Logic a Practical Approach*. Academic Press Limited, Estados Unidos, 1994.
- MOCCELIN, J. V. *Uma Contribuição à Programação de Operações em Sistemas de Produção Intermitente Flow-Shop*. Tese de livre docência, Escola de Engenharia de São Carlos. São Carlos, SP: EESC-USP, 1992.

- MOCCELIN, J. V. *Técnicas de seqüenciamento e Programação de Operações em Máquinas*. Apostila, Escola de Engenharia de São Carlos. São Carlos, SP: EESC-USP, 1994.
- MOCCELIN, J. V.; NAGANO, M. S. *A New Constructive Heuristic Method for Minimizing Makespan in Permutation Flow Shop Scheduling*. 19º Encontro Nacional de Engenharia de Produção, Rio de Janeiro, novembro, 1999.
- MONTEVECHI, J.A.B. *Contribuição para Identificação de Similaridades entre Peças - Abordagem Baseada na Lógica Fuzzy em Sistemas de Apoio Computadorizados*. Tese de Doutorado. São Paulo - SP, 1995.
- MORITA, Y.; OKA, Y.; WATANABE, H. *A Theory of Evaluation Applied to Human-Environment System*. J. Therm. Biol., Vol. 18, Nº 5/6, pp. 443-447, 1993.
- MOORE, J. M. *An n Job, one Machine Sequencing algorithm for Minimizing the Number of Late Jobs*. Management Science, Vol. 15, Nº 1, September. 1968.
- NAWAZ, M.; ENSCORE, E. & HAM, I. "A Heuristic Algorithm for the m-Machine, n-Job Flow-Shop Sequencing Problem". *OMEGA - the int. J. of Mgnt S.* Vol. 11 No. 1 pp. 91-95, 1983.
- OSMAM, I. H.; POTTS, C. N. *Simulated Annealing for Permutation Flow-Shop Scheduling*. *OMEGA - the int. J. of Mgnt S.* Vol. 17, pp. 551-557, 1989.
- PALMER, D. S. *Sequencing Jobs Through a Multi-Stage Process in the Minimum Total Time - A Quick Method of Obtaining a Near Optimum*. Operational Research Quarterly, Vol. 16, pp.101-107, 1965.
- ROSA, E. B. *Parâmetros de Desempenho: A Vantagem competitiva das Empresas - Estudo de Caso*. Dissertação de Mestrado, EFEI, Itajubá, 1996.
- SALOMON, V. A. P. *Contribuição para auxílio à decisão em Processo de Compras*. Dissertação de Mestrado, EFEI, Itajubá, 1998.
- SIDNEY, J. B. *Na Extension of Moore's Due Date Algorithm*. Symposium on the Theory of Scheduling and its Applications, Edited by S.E. Elmaghraby, Springer-Verlag, 1973.
- SMITH, W. E. *Various Optimizers for Single-Stage Production*. Naval Research Logistics Quarterly, Vol. 3, Nº 1, March, 1956.
- STÜTZLE, T. *An Ant Approach to the Flow Shop Problem*. 6th European Congress on Intelligent Techniques and Soft Computing, Aachen, Alemanha, 1998
- TERANO, T.; ASAI, K.; SUGENO, M. *Applied Fuzzy Systems*. Academic Press Limited, Estados Unidos, 1994.
- WEBBER, R., *Data Analysis Applications of Neural and Fuzzy Systems in Industry*. 5th European Congress on Intelligent Techniques and Soft Computing, Aachen, Alemanha, 1997
- WIDMER, M.; HERTZ, A. *A New Heuristic Method for the Flow Shop Sequencing Problem*. European Journal of Operation Research, Vol. 41, pp. 186-193, 1989.
- ZADEH, L. A. *Information and Control*. Fuzzy Sets, Vol. 8, pp. 338-353, 1965.
-

Anexo A: Tabelas para o fator eficiência

<i>Classificação</i>	<i>Habilidade</i>	<i>Esforço</i>
<i>Fraca</i>	Não adaptado ao trabalho, comete erros e seus movimentos são inseguros.	Falta de interesse ao trabalho e utiliza métodos inadequados.
<i>Regular</i>	Adaptado ao trabalho, comete erros e seus movimentos são inseguros.	As mesmas tendências, porém com menos intensidade.
<i>Normal</i>	Trabalha com exatidão satisfatória e seu ritmo se mantém razoavelmente constante.	Trabalha com constância e se esforça razoavelmente.
<i>Boa</i>	Tem confiança em si mesmo e seu ritmo se mantém razoavelmente constante.	Trabalha com constância e confiança muito pouco ou nenhum tempo perdido.
<i>Excelente</i>	Precisão nos movimentos, nenhuma hesitação e ausência de erros.	Trabalha com rapidez e com movimentos precisos.
<i>Superior</i>	Movimentos sempre iguais, mecânicos, comparáveis ao de uma máquina.	Se lança numa marcha impossível de manter. Não serve para estudo de tempos.

Tabela A.1 – Avaliação do ritmo de trabalho extraído de ROSA (1996)

Valores FE			Habilidade										
			S		E		B		N	R		F	
			A1	A2	B1	B2	C1	C2	D	E1	E2	F1	F2
Esforço			0,15	0,13	0,11	0,08	0,06	0,03	0,00	-0,05	-0,10	-0,16	-0,22
S	A1	0,13	1,28	1,26	1,24	1,21	1,19	1,16	1,13	1,08	1,03	0,97	0,91
	A2	0,12	1,27	1,25	1,23	1,20	1,18	1,15	1,12	1,07	1,02	0,96	0,90
E	B1	0,10	1,25	1,23	1,21	1,18	1,16	1,13	1,10	1,05	1,00	0,94	0,88
	B2	0,08	1,23	1,21	1,19	1,16	1,14	1,11	1,08	1,03	0,98	0,92	0,86
B	C1	0,05	1,20	1,18	1,16	1,13	1,11	1,08	1,05	1,00	0,95	0,89	0,83
	C2	0,02	1,17	1,15	1,13	1,10	1,08	1,05	1,02	0,97	0,92	0,86	0,80
N	D	0,00	1,15	1,13	1,11	1,08	1,06	1,03	1,00	0,95	0,90	0,84	0,78
R	E1	-0,04	1,11	1,09	1,07	1,04	1,02	0,99	0,96	0,91	0,86	0,80	0,74
	E2	-0,08	1,07	1,05	1,03	1,00	0,98	0,95	0,92	0,87	0,82	0,76	0,70
F	F1	-0,12	1,03	1,01	0,99	0,96	0,94	0,91	0,88	0,83	0,78	0,72	0,66
	F2	-0,17	0,98	0,96	0,94	0,91	0,89	0,86	0,83	0,78	0,73	0,67	0,61

Tabela A.2 – valores do fator de ritmo de trabalho extraído de ROSA (1996)

Anexo B – Artigos originados desta dissertação

“Incertezas em Algoritmos de Programação de Operações” 19º Encontro Nacional de Engenharia de Produção, Rio de Janeiro, novembro - 1999.

“Triangular Fuzzy Numbers Utilization in a Flow-Shop Sequence Algorithm”. Sixth European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, setembro - 1998.

“Aplicação de Números Fuzzy Triangulares em Análises de Investimentos em Situações de Incerteza - Método Baseado na Teoria dos Jogos”. 17º Encontro Nacional de Engenharia de Produção, Gramado, outubro - 1997.

“Utilização de Números Triangulares Fuzzy em Algoritmos de Sequenciamento tipo Flow-Shop”. 16º Encontro Nacional de Engenharia de Produção, Piracicaba, outubro - 1996.

“Um Modelo Computacional, Baseado no Método AHP, para Análise Multicriterial de Decisão”. 16º Encontro Nacional de Engenharia de Produção, Piracicaba, outubro - 1996.