

RICARDO SANTAMARIA CORRÊA DA FONSECA

**DESENVOLVIMENTO DE UM *SOFTWARE* PARA A SOLUÇÃO
DE UM PROBLEMA TÍPICO DE *TIMETABLING* EM UMA
EMPRESA INSERIDA NA ERA DO CONHECIMENTO**

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

**São Paulo
2005**

RICARDO SANTAMARIA CORRÊA DA FONSECA

**DESENVOLVIMENTO DE UM *SOFTWARE* PARA A SOLUÇÃO
DE UM PROBLEMA TÍPICO DE *TIMETABLING* EM UMA
EMPRESA INSERIDA NA ERA DO CONHECIMENTO**

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

Orientador:
Prof. Dr. Miguel Cezar Santoro

**São Paulo
2005**

“(..) face aos múltiplos desafios do futuro, a educação surge como um trunfo indispensável à humanidade na construção dos ideais da paz, da liberdade e da justiça social (..)” (DELORS, 2000 apud DA CUNHA; DA SILVA, 2002)

AGRADECIMENTOS

A toda minha família pelo esforço, dedicação e paciência despendidos.

Aos meus pais pelos ensinamentos transmitidos e que foram essenciais para a formação do meu caráter. Não há dúvidas de que levarei para o resto da minha vida o carinho, amor e afeto que eles me deram. Serei eternamente grato por todos os sacrifícios que fizeram para me proporcionar uma vida repleta de felicidades.

Aos meus amigos pelos momentos de alegria e tristeza vividos ao longo desses anos, em especial a uma pessoa maravilhosa e única chamada Ceci Leal a quem devo momentos efêmeros, mas certamente inesquecíveis, de felicidade.

A Joana pelo companheirismo e lealdade demonstrados.

Aos professores do *Departamento de Engenharia Mecânica* e de *Produção* pelos conhecimentos compartilhados.

Aos funcionários da *Escola Politécnica*, Paulo Joaquim Moreira Santiago, Ronni dos Santos Oliveira, Márcia A. Costa, funcionários do xerox do Osni e Sami pela paciência e ajuda.

Aos funcionários e alunos do *Projeto Avizinhar* (CECAE) pela experiência de vida que me proporcionaram.

Aos colegas de trabalho Sílvia A. Hamasaki, Paulo Mancinelli, Hilário Vizintim, Marcos Campos, Augusto Cruz, Tatiana Fogaça, Pablo Martines, Juliana Martins, Fernanda Bellentani, Welington Vieira pela amizade, reconhecimento e ajuda.

Ao Professor Doutor Miguel Cezar Santoro pelos importantes conselhos dados, e, principalmente, por ter desempenhado não somente a figura de orientador mas também a de amigo.

RESUMO

O presente trabalho apresenta o desenvolvimento de um *software* para lidar com o problema de *timetabling* enfrentado por uma empresa que fornece treinamentos. O *software* foi desenvolvido por meio das recomendações de pessoas ligadas a essa atividade como também pelas idéias encontradas na literatura vigente sobre o tema. Finalmente, o *software* foi aplicado em um estudo de caso para comprovar o funcionamento do método de solução proposto: junção de heurística construtiva probabilística e busca tabu (método híbrido). Os resultados encontrados mostraram-se satisfatórios para o problema de médio porte tratado.

ABSTRACT

This paper presents the development of a software to solve timetabling problems, a typical issue that enterprises responsible to train end-users, especially in ERP systems, undergo. This software was created based on recommendations of people who work in this area of knowledge as well on the ideas extracted from the literature about timetabling. To sum up, the software was applied in a case in order to prove the efficiency of the suggested method: junction of constructive heuristics and tabu search (hybrid method). The outcomes were very reasonable for the medium-sized problem solved.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

1. INTRODUÇÃO.....	2
1.1 Organização do trabalho	2
1.2 Contextualização.....	3
1.3 A empresa na era do conhecimento	6
1.4 Formas de conhecimento.....	9
1.5 Tecnologia da informação e a gestão do conhecimento	13
1.5.1 ERP.....	16
1.6 Trabalhador do conhecimento	19
1.6.1 Educação e aprendizado	20
1.6.2 A necessidade de treinamento.....	22
1.6.2.1 Auxílio à capacitação	22
1.6.2.2 Auxílio à mudança	25
1.7 ERP e treinamento	27
2. A EMPRESA E O PROBLEMA	31
3. REVISÃO BIBLIOGRÁFICA	38
3.1 O problema de timetabling	38
3.1.1 University timetabling	41
3.1.2 School timetabling.....	43
3.1.3 Examination timetabling.....	44
3.2 Complexidade e especificidade	46
3.3 Teoria da complexidade	47
3.4 Restrições hard e soft – solução viável e aceitável.....	51
3.5 Resolução manual x automatização.....	59
3.6 Métodos de solução para o problema de timetabling	61
3.7 Estado da arte	65
3.7.1 Abordagem histórica	65

3.7.2	Hiperheurísticas.....	66
3.7.3	Programação matemática.....	67
4.	METODOLOGIA.....	69
5.	<i>SOFTWARE</i> DESENVOLVIDO	74
5.1	Necessidades identificadas.....	74
5.1.1	Restrições do tipo hard	74
5.1.2	Restrições do tipo soft	76
5.1.3	Outras considerações.....	77
5.2	A escolha dos métodos.....	77
5.3	A lógica do software	78
5.3.1	Tratamento do problema em três partes	78
5.3.2	Heurística construtiva probabilística - instrutor.....	79
5.3.2.1	Exceção no sorteio de instrutores.....	80
5.3.3	Heurística probabilística - viagem de instrutor	81
5.3.4	Heurística construtiva probabilística - sala.....	81
5.3.5	Heurística construtiva probabilística e determinística – seleção dos tuples	82
5.3.6	Alocação dos tuples no tempo	83
5.3.7	Preferência dos funcionários.....	85
5.4	Parametrização da heurística construtiva.....	86
5.5	Critérios de parada – heurística construtiva	86
5.5.1	Número de soluções viáveis obtidas	86
5.5.2	Tempo por parametrização	87
5.5.3	Número de iterações por parametrização	87
5.5.4	Outros critérios.....	87
5.5.4.1	Critério do tempo total	87
5.5.4.2	Critério do número de iterações total	88
5.5.4.3	Mistura de critérios.....	88
5.6	Escolha da melhor solução - análise de custo	88
5.7	Busca tabu	90
5.7.1	Vizinhança - movimento de troca	91
5.7.2	Critérios de parada da BT	94

5.7.2.1	Tempo por parametrização	94
5.7.2.2	Número de iterações por parametrização	94
5.7.2.3	Outros critérios.....	95
5.7.2.3.1	Critério do tempo total	95
5.7.2.3.2	Critério do número de iterações total.....	95
5.7.2.3.3	Mistura de critérios	95
6.	EXEMPLO DE APLICAÇÃO	97
7.	ESTUDO DE CASO.....	99
7.1	Descrição do estudo de caso.....	99
7.1.1	A empresa	99
7.1.2	Participação da M & R	101
8.	ANÁLISE DOS RESULTADOS	104
8.1	Execução 1	104
8.1.1	Parametrização	104
8.1.2	Critérios de parada.....	104
8.1.3	Outras considerações	104
8.1.4	Soluções encontradas.....	105
8.1.4.1	Relatório	105
8.1.4.2	Programação da melhor solução encontrada	108
8.1.4.2.1	Observações importantes	109
8.2	Execução 2	111
8.2.1	Parametrização	111
8.2.2	Critérios de parada.....	111
8.2.3	Outras considerações	111
8.2.4	Soluções encontradas.....	112
8.2.4.1	Relatório	112
8.2.4.2	Programação da melhor solução encontrada	115
8.2.4.2.1	Observações importantes	116
8.3	Execução 3	117
8.3.1	Parametrização	117
8.3.2	Critérios de parada.....	117
8.3.3	Outras considerações	118

8.3.4	Soluções encontradas.....	118
8.3.4.1	Relatório	118
8.3.4.2	Programação da melhor solução encontrada	120
8.3.4.2.1	Observações importantes	122
8.4	Considerações sobre a BT	123
8.4.1	Desempenho da BT	128
8.5	Comparação entre as soluções do estudo de caso.....	128
8.6	Comparação da melhor solução do software com a solução manual	129
8.7	Análise da parametrização	129
8.7.1	Execução extra	130
8.7.2	Relatório.....	131
8.7.3	Conclusão.....	133
9.	CRÍTICAS E MELHORIAS	135
10.	CONCLUSÕES.....	138
ANEXO A	140
1.	Classificação das abordagens	140
1.1	Otimizantes.....	142
1.1.1	Enumeração explícita	142
1.1.2	Enumeração implícita.....	142
1.2	Modelos heurísticos	143
1.2.1	Construtivo.....	144
1.2.2	Busca	144
1.3	Heurística construtiva	144
1.3.1	Construída.....	144
1.3.2	Aprendida.....	145
1.4	Busca.....	145
1.4.1	Busca em vizinhança	146
1.4.2	Busca estendida vs busca direta	147
1.5	Detalhamento de métodos de solução.....	148
1.5.1	Metaheurísticas	149
1.5.1.1	Tabu Search ou Busca Tabu	149
1.5.1.2	Simulated annealing	152

1.5.1.3	Algoritmo genético.....	155
1.5.1.3.1	Inicialização.....	155
1.5.1.3.2	Simulação do processo evolutivo	156
1.5.1.4	Procura local	157
1.5.2	CLP.....	158
1.5.3	DSS (Decision suport systems - sistemas de apoio à decisão)	159
1.5.4	Relaxação lagrangiana.....	159
1.5.5	Métodos híbridos.....	161
ANEXO B.....		163
1. Cadastros.....		163
1.1	Cadastro principal.....	163
1.2	Cadastro dos instrutores	164
1.3	Cadastro dos alunos	164
1.4	Cadastro da matérias.....	164
1.5	Cadastro das salas.....	165
1.6	Cadastro das turmas	165
1.7	Cadastro dos tipos de sala	166
1.8	Cadastro de localidades.....	166
1.9	Cadastro de disponibilidade do instrutor.....	167
1.10	Cadastro de disponibilidade de horário das salas	168
1.11	Cadastro de disponibilidade dos alunos.....	169
1.12	Cadastro da relação instrutor-matéria	169
1.13	Cadastro do local dos instrutores.....	170
1.14	Cadastro da regra de precedência entre matérias.....	171
1.15	Cadastro das informações sobre matéria.....	171
1.16	Cadastro da relação aluno-turma	172
1.17	Cadastro do local origem das turmas.....	173
1.18	Cadastro da relação turma-matéria	173
1.19	Cadastro dos locais de simulação das turmas.....	173
1.20	Cadastro de união entre turmas	175
1.21	Cadastro das informações sobre salas.....	176
1.22	Cadastro da distância entre localidades	177

1.23	Cadastro do custo entre localidades.....	178
1.24	Cadastro dos pesos da função objetivo	178
2.	Parâmetros de sorteio	179
3.	Critérios de parada	179
4.	Horizonte desejado.....	179
5.	Solução do exemplo.....	179
6.	Funcionamento da busca tabu.....	183
	ANEXO C	187
1.	Cadastros comuns às soluções 1,2 e 3.....	187
1.1	Relação instrutor-matéria	187
1.2	Regra de precedência	187
1.3	Relação turma-matéria	188
1.4	Informações sobre as matérias.....	189
1.5	Informação sobre as salas.....	189
1.6	Local dos instrutores	190
1.7	Relação turma-matéria-instrutor.....	190
1.8	Local e tamanho das turmas	191
1.9	Relação turma-matéria-local	192
1.10	Relação aluno-turma	193
1.11	Custo e distância entre localidades	196
1.12	Peso dos critérios	196
1.13	Outras considerações	197
2.	Diferença entre as execuções	197
	ANEXO D	201
	LISTA DE REFERÊNCIAS.....	203

LISTA DE TABELAS

Tabela 1: Comparação das três revoluções, adaptado de Paiva (2000).....	3
Tabela 2: Comparação entre a velha e nova realidade, adaptado de ASTD (2005).....	6
Tabela 3: Conversão do conhecimento, adaptado de Da Silva (2005).....	11
Tabela 4: Classificação das ferramentas de TI de acordo com a conversão de conhecimento, adaptado de Carvalho (2000)	16
Tabela 5: Diferença entre <i>planning</i> , <i>scheduling</i> e <i>timetabling</i> (BARTAK; RUDOVA, 2001).....	40
Tabela 6: Exemplo de restrições do tipo <i>hard</i> e <i>soft</i> , elaborada pelo autor	58
Tabela 7: Abordagens para a solução do problema de <i>timetabling</i> , elaborada pelo autor.....	64
Tabela 8: Critérios para a escolha de professor.....	80
Tabela 9: Critérios para escolha de sala.....	82
Tabela 10: Critérios para sortear <i>tuples</i>	83
Tabela 11: Formas de alocar matéria.....	84
Tabela 12: Exemplo de preferência de horário de funcionários (2=muito bom; 1=bom; 0=não).....	85
Tabela 13: Exemplo de preferência de horário de uma turma composta por 3 alunos	85
Tabela 14: Relatório mostrando as 15 melhores soluções (do total de 180) para a primeira execução	106
Tabela 15: Programação da melhor solução (divida em 2 partes)	109
Tabela 16: Relatório mostrando as 15 melhores soluções (do total de 180) para a segunda execução.....	113
Tabela 17: Programação da melhor solução (divida em 2 partes)	116
Tabela 18: Relatório mostrando as 15 melhores soluções (de um total de 36) para a terceira execução.....	119
Tabela 19: Programação da melhor solução (divida em 2 partes)	122
Tabela 20: Programação da solução de número 4 antes da BT.....	125
Tabela 21: Programação da solução de número 4 após a BT	127
Tabela 22: Comparação entre soluções	128

Tabela 23: Contagem dos parâmetros para cada execução.....	130
Tabela 24: Melhores parâmetros	130
Tabela 25: Relatório para a execução extra	132
Tabela 26: Classificação das principais metaheurísticas (GLOVER; LAGUNA, 1997).....	150
Tabela 27: Influência do coeficiente de temperatura.....	154
Tabela 28: Influência da temperatura	154
Tabela 29: Relatório gerado pelo programa.....	181
Tabela 30: Relatório gerado (antes e depois da BT)	184
Tabela 31: Relação instrutor-matéria.....	187
Tabela 32: Regra de precedência entre as matérias.....	188
Tabela 33: Relação turma-matéria (2 significa que a turma deve assistir a matéria em questão).....	188
Tabela 34: Informações sobre as matérias	189
Tabela 35: Informação sobre as salas	189
Tabela 36: Local original e de simulação dos instrutores.....	190
Tabela 37: Especificação forçada de instrutores para determinadas combinações turma-matéria.....	191
Tabela 38: Informação sobre as turmas (local origem e tamanho)	192
Tabela 39: Relação turma-matéria-local.....	193
Tabela 40: Relação aluno-turma.....	195
Tabela 41: Custo entre localidades	196
Tabela 42: Tempo de deslocamento (-1 = dia seguinte).....	196
Tabela 43: Peso dos critérios.....	196
Tabela 44: Preferência das turmas em relação à primeira semana	198
Tabela 45: Preferência das turmas em relação à segunda semana	199
Tabela 46: Preferência das turmas em relação à terceira semana	199
Tabela 47: Resolução manual do Estudo de Caso.....	201

LISTA DE FIGURAS

Figura 1: Processos de conversão do conhecimento, adaptado de Florêncio et al. (2005)	12
Figura 2: Arquitetura de um sistema ERP, adaptado de Davenport (1998, <i>apud</i> BOJIKIAN, 2004)	17
Figura 3: Diferença entre processos produtivos, adaptado de Terra (2005)	20
Figura 4: Exemplo de formulação de um problema de TT (Carter, 1989)	49
Figura 5: Representação das fases em que o problema foi dividido.	78
Figura 6: Critérios de sorteio do professor, sala e <i>tuple</i>	86
Figura 7: Parametrização utilizada na execução 1.	104
Figura 8: Parametrização utilizada na execução 1.	111
Figura 9: Parametrização utilizada na execução 1.	117
Figura 10: Classificação dos modelos de solução, adaptado de Santoro (2003).....	141
Figura 11: Exemplo de busca Tabu , elaborada pelo autor	152
Figura 12: Exemplo de relaxação Lagrangiana (MORTON; PENTICO, 1993).....	160
Figura 13: Cadastro principal do programa	163
Figura 14: Cadastro dos instrutores	164
Figura 15: Cadastro de alunos/funcionários.....	164
Figura 16: Cadastro das matérias	165
Figura 17: Cadastro das salas	165
Figura 18: Cadastro das turmas	166
Figura 19: Cadastro de tipos de sala	166
Figura 20: Cadastro de localidades.....	167
Figura 21: Cadastro de disponibilidade de horários dos instrutores.....	167
Figura 22: Cadastro de disponibilidade das salas.....	168
Figura 23: Disponibilidade dos alunos/funcionários	169
Figura 24: Cadastro da relação instrutor-matéria	170
Figura 25: Cadastro do local do instrutor	170
Figura 26: Cadastro da relação de precedência	171
Figura 27: Cadastro das informações sobre as matérias	171

Figura 28: Cadastro da relação aluno-turma	172
Figura 29: Cadastro do local origem das turmas.....	173
Figura 30: Cadastro da relação turma-matéria	173
Figura 31: Cadastro do local de simulação das turmas.....	174
Figura 32: Cadastro da relação turma-matéria-instrutor.....	175
Figura 33: Cadastro da relação de união entre turmas.....	176
Figura 34: Cadastro das informações sobre salas.....	177
Figura 35: Cadastro de distância entre localidades	177
Figura 36: Cadastro de custo entre localidades	178
Figura 37: Importância relativa entre os critérios.....	178
Figura 38: Parâmetros de sorteio para o exemplo	179
Figura 39: Soluções obtidas para o exemplo em questão	180
Figura 40: Soluções-exemplo para a Busca Tabu	184
Figura 41: Solução (2) onde a BT foi efetiva.....	185

LISTA DE ABREVIATURAS E SIGLAS

BT	Busca Tabu
ERP	<i>Enterprise Resource Planning</i>
GC	Gestão do Conhecimento
MRP	<i>Material Requirements Planning</i>
TI	Tecnologia de Informação
TT	<i>Timetabling</i>

INTRODUÇÃO

1. INTRODUÇÃO

1.1 ORGANIZAÇÃO DO TRABALHO

O presente trabalho foi seccionado em 10 capítulos, conforme explicado a seguir:

- 1) *Introdução*: contextualização do problema tratado. O autor deste trabalho considera extremamente importante explicar o motivo do aparecimento do problema abordado neste trabalho; entretanto, os leitores mais interessados no assunto *timetabling* poderão deixar de ler este capítulo sem comprometimento do resto da leitura.
 - 2) *A Empresa e o Problema*: visão geral da empresa-alvo deste trabalho assim como do problema a ser solucionado.
 - 3) *Revisão Bibliográfica*: apesar de não ser uma pesquisa exaustiva sobre o tema, este capítulo fornece idéias de diversos autores como também fornece subsídios para um entendimento mais profundo do tema *timetabling*.
 - 4) *Metodologia*: explica com detalhes todo o processo seguido para a concepção do *software*.
 - 5) *Software desenvolvido*: este capítulo aborda minuciosamente o funcionamento do *software*.
 - 6) *Exemplo de aplicação*: demonstra a aplicação do *software* em um exemplo bem simples no intuito de preparar o leitor para o capítulo seguinte.
 - 7) *Estudo de Caso*: especifica um problema de médio porte onde o *software* pode ser aplicado.
 - 8) *Análise dos Resultados*: este capítulo segue uma linha de raciocínio que permite mostrar ao leitor os tipos de análises que devem ser feitas ao utilizar-se o *software* desenvolvido.
 - 9) *Críticas e Melhorias*: listagem das deficiências deste trabalho assim como recomendações para a sua melhoria.
-

10) *Conclusões*1.2 **CONTEXTUALIZAÇÃO**

Não há dúvidas de que a história da humanidade é marcada por duas grandes revoluções: a Agrícola, cerca de 10.000 anos atrás; e a Revolução Industrial cujo início deu-se logo após o advento da máquina a vapor, em 1776, e foi responsável por substituir o serviço artesanal pela produção em massa em fábricas emergentes (TOFFLER, 1990 *apud* CAVALCANTI, 1995). Entretanto, segundo um grande número de autores, dentre eles Toffler (1990) e Peter Drucker (1996 *apud* PAIVA, 2000), a humanidade passa por sua terceira onda ou revolução cuja principal característica é a informação e o conhecimento, a chamada Revolução da Informação. A tabela abaixo sintetiza alguns pontos destacados por Toffler (1990 *apud* PAIVA, 2000) relacionados às principais diferenças entre as revoluções (ou ondas, como o autor as chama) citadas.

	Revolução Agrícola	Revolução Industrial	Revolução do Conhecimento
<i>Período</i>	8000 A.C – 1750 D.C.	1750 a 1955.	1955 até os dias de hoje.
<i>Base da Economia</i>	Terra.	Fontes energéticas não renováveis (carvão, gás, petróleo).	Informação e conhecimento.
<i>Energia para mover a Economia</i>	Potência muscular humana e animal e fontes renováveis (sol, vento etc.).	Máquinas.	Forças mentais e intelectuais (ondas cerebrais).
<i>Produção</i>	Artesanal (por encomenda).	Em massa (padronização dos produtos).	Personalizada (foco no cliente).
<i>Meios de Comunicação e de transporte</i>	Difíceis, precários, quase inexistentes.	Estradas de ferro; rodovias; transporte aéreo, correio, telégrafo, telefone.	Velozes, poderosos, diversificados; teia global.
<i>Trabalhador</i>	Camponês.	Operário, alta especialização.	Do conhecimento, generalista.
<i>Ênfase contábil</i>	Partida simples.	Partidas dobradas; custo histórico; elaboração de relatório contábil.	Sistemas de informação e capital intelectual.

Tabela 1: Comparação das três revoluções, adaptado de Paiva (2000)

O advento de termos como Revolução da Informação, Capital Intelectual, Gestão do Conhecimento (GC), Trabalhadores do Conhecimento, Sociedade do Conhecimento etc. não é por acaso. Conforme Paiva (2000) aponta, a sociedade presente e futura não será mais movida por elementos como capital financeiro, máquinas ou ferramentas; mas sim pelo conhecimento e pela informação. Este raciocínio é compartilhado por Drucker (1996 *apud* PAIVA, 2000, p.1) quando diz que “o conhecimento não é mais um recurso, e sim o recurso” ou quando afirma que “os principais grupos sociais da sociedade do conhecimento serão os trabalhadores do conhecimento”, ou seja, pessoas dotadas da competência de converter informações em conhecimento no intuito de usá-lo para gerar valor a uma empresa – incremento da produtividade, inovação etc. (DRUCKER, 1997 *apud* DA CUNHA; DA SILVA, 2002, p. 77).

Paiva (2000) destaca que o conhecimento, principal componente do capital intelectual, não é um tema novo sendo já estudado por Platão (328 A.C. – 347 A.C.), Aristóteles (384 A.C. – 322 A.C.) e outros filósofos que os sucederam. Loureiro (2003) também ressalta que as civilizações mais antigas sempre se preocuparam com o conhecimento adquirido, ou seja, preservar as experiências e reflexões dos seus povos. Este último autor também aponta que a necessidade de captar, armazenar e distribuir conhecimento é um grande incentivo para o desenvolvimento de novas tecnologias. Seguindo este raciocínio têm-se invenções como a escrita proporcionada por poetas, narradores e monges, ou seja, a transição da cultura oral para a escrita; mais tarde o advento da máquina de imprimir que pode ser considerada, segundo Loureiro (2003), uma das maiores invenções da humanidade, pois foi responsável pela acelerada disseminação de informação a custos menores permitindo assim que cidadãos comuns a acessassem.

Ressalta-se, porém, que apesar dos avanços supracitados, até este momento da história a informação fornecida e transmitida ainda era limitada. Somente após o avanço das telecomunicações e da informática que a fase proferida por Bono (*s/d apud* PAIVA, 2005, p.1) torna-se verdade: “houve tempo em que o gargalo estava nas informações; hoje ele está na capacidade de agir com inteligência sobre elas”. A principal ferramenta que proporcionou que a humanidade entrasse em uma nova era onde o problema da disseminação de informações se transferisse para a capacidade

de absorvê-las foi o computador. Primeiramente desenvolvido nas dependências da Universidade da Pensilvânia, nos Estados Unidos, em 1947, o computador ENIAC marcou o início de uma era sem precedentes em relação à agilização da informação (RIOS 1987 *apud* CAVALCANTI, 1995). O computador aliado à telecomunicação possibilitou que informações fossem trocadas em velocidades vertiginosas:

- ◆ O censo americano de 1880 demorou 8 anos para ser realizado sendo que, atualmente, as eleições presidenciais no mesmo país ocorrem em poucas horas (CAVALCANTI, 1995).
- ◆ A informação *on-line* cresce a taxa de 20 milhões de páginas por dia (TERRA, 2005).
- ◆ Diariamente 40 bilhões de mensagens eletrônicas são enviadas (TERRA, 2005)
- ◆ A produção científica passou de 10.000 artigos por dia, em 1986, para 20.000 em apenas quatro anos (WEITZEN, 1991 *apud* CAVALCANTI, 1995);
- ◆ A capacidade média de armazenamento cresce em 50% ao ano (TERRA, 2005);

Justamente o casamento entre a informática e as telecomunicações que possibilitou o surgimento da famigerada rede mundial de computadores e também servirá como subsídio para a construção da chamada *superhighway*. Marques (1994 *apud* CAVALCANTI, 1995) e Martins (1994 *apud* CAVALCANTI, 1995) definem *superhighway* como uma rede de fibras óticas que permitirá o tráfego de sons, imagens e dados em altíssimas velocidades. A principal finalidade desta rede ou estrada será a comunicação entre computadores, ou seja, a disseminação de informação e conhecimento. A importância da troca de informações e conhecimento é destacada por Peters (1992, *apud* CAVALCANTI, 1995, p.5) quando diz que “nada foi de maior importância histórica para o desenvolvimento econômico relativo, nos Estados Unidos e em outros lugares, do que o crescimento de redes (canais, estradas de ferro, auto-estradas, telefones...). O desenvolvimento de redes de informação, dentro da empresa e das nações, como um todo, terá impacto tão grande na saúde

econômica relativa, no século XXI, quanto os sistemas ferroviário e rodoviário tiveram nos séculos XIX e XX”.

Objetivou-se neste item contextualizar o leitor em relação alguns importantes tópicos abordados neste novo milênio. Entretanto, qual será a influência deste novo ambiente no âmbito das organizações e, conseqüentemente, em relação aos seus funcionários ? Estas e outras questões serão abordadas nos tópicos subseqüentes.

1.3 A EMPRESA NA ERA DO CONHECIMENTO

Por quê a *Microsoft*, empresa de Bill Gates e Paul Allen, tem seu valor de mercado avaliado em 100 vezes o seu valor tangível ? Por quê a *IBM*, em 1995, pagou 14 vezes o valor contábil pela compra da *Lotus* ? Por quê a *NOKIA*, filial filandesa, com um número extremamente reduzido de funcionários fatura 200 milhões de dólares ao ano ? Por quê no período de 1981 a 1993 ocorreram 391 aquisições nos EUA sendo que, em média, o valor real das organizações eram 4,5 vezes maior do que os balanços patrimoniais demonstravam ? (PAIVA, 2000).

Talvez a resposta para estas perguntas esteja sintetizada na tabela abaixo:

A Velha Realidade	A Nova Realidade
Pessoas precisam de organizações.	Organizações precisam de pessoas.
Máquinas, capital e localização são vantagens competitivas.	Pessoas talentosas são vantagens competitivas.
Empregados talentosos podem fazer alguma diferença.	Empregados talentosos são essenciais para o sucesso.
Trabalho é escasso.	Pessoas talentosas estão em demanda.
Empregados são leais e os trabalhos oferecem segurança.	Pessoas têm um compromisso de curto prazo com as empresas.
Pessoas aceitam o pacote padrão que as são oferecidos.	Pessoas querem muito mais.

Tabela 2: Comparação entre a velha e nova realidade, adaptado de ASTD (2005)

Entende-se com esta tabela que as pessoas e, conseqüentemente, seu conhecimento estão no centro da Nova Realidade. Stewart (1998 *apud* PAIVA, 2000, p.3) apregoa que “(...) os ativos físicos de uma empresa baseada no conhecimento

contribuem muito menos para o valor de seu produto (ou serviço) final do que os ativos intangíveis – os talentos de seus funcionários, a eficácia de seus sistemas gerenciais, o caráter de seus relacionamentos com os clientes – que, juntos, constituem seu capital intelectual”. O mesmo autor também diz que capital intelectual “é a soma do conhecimento de todos em uma empresa, o que lhe proporciona vantagem competitiva”. Já para Edvinsson & Malone (1998 *apud* PAIVA, 2005b), capital intelectual é a soma do capital humano e do capital estrutural:

- ◆ Capital Humano: soma da capacidade, conhecimento, habilidade e experiência dos empregados de uma organização.
- ◆ Capital Estrutural: constitui-se pela infra-estrutura que suporta o capital humano, ou seja, *softwares*, bancos de dados, equipamentos de informática ou, resumidamente, tudo aquilo que permanece na empresa quando as pessoas vão embora.

Os autores usam a metáfora da árvore para explicar o capital intelectual: os frutos seriam os lucros; as folhas, galhos e tronco a parte visível da empresa e expressa pelo processo contábil e a raiz significaria o capital intelectual, ou seja, o valor oculto.

As características peculiares do capital intelectual como intangibilidade, volatilidade, difícil mensuração, diversidade de formas faz com que haja discrepância entre o valor contábil e o valor real das empresas. Este problema já é abordado por alguns autores. Paiva (2005b) aponta que a contabilidade, da forma que se encontra hoje, apresenta falhas, pois ainda não conta com uma ferramenta para lidar com as informações relacionadas aos ativos intelectuais. Segundo a autora, apesar da importância dada ao capital humano, a maioria das organizações não possui um sistema de informação capaz de contabilizá-lo principalmente porque, ao contrário dos ativos físicos e financeiros, este é extremamente qualitativo.

Portanto, a resposta das perguntas apresentadas no início deste tópico reside no fato de que a contabilidade atual ainda não está preparada para lidar com ativo intangíveis que afetam cada vez mais o valor de uma empresa, principalmente as

empresas inseridas na Era do Conhecimento. Traduz-se por ativos intangíveis o conhecimento e inteligência¹ dos funcionários de uma empresa. Atualmente já existem alguns modelos para medir o capital intelectual. Esta preocupação em descobrir maneiras para medir o intangível, por mais paradoxo que isso pareça, mostra-se clara principalmente quando se estima, segundo a Hay Consultoria (*apud* BERNARDI, 1998 *apud* PAIVA, 2005b), que a saída de um profissional mediano pode custar à empresa entre R\$ 300.000 e R\$ 600.000 enquanto que a perda de um executivo de primeira linha o custo é de no mínimo R\$ 1 milhão.

Apesar da mensuração do capital intelectual ainda ser falha, as empresas sabem que ele vale muito dinheiro e por isso começaram a tentar administrá-lo. Não é por acaso que a partir da década de 90 começou-se a falar em GC. Nesta época muitas empresas viram os seus funcionários mais antigos irem embora e notaram que o prejuízo causado era grande. Os ativos intangíveis das empresas encontram-se principalmente nas pessoas, e quando elas saem os ativos são perdidos em grande parte. No início dos anos 90, isso começou a ser percebido mais claramente quando os funcionários substitutos muitas vezes não conseguiam resolver situações cotidianas que os antigos tinham pleno domínio (CECATTO, 2005).

Segundo uma pesquisa realizada na Universidade de São Paulo por José Cláudio Cyriney Terra, atualmente coexistem três tipos de empresa: empresas que aprendem; empresas tradicionais e pequenas atrasadas. No primeiro tipo encontram-se as empresas que já se adaptaram à Nova Era e que por isso possuem um maior nível de envolvimento com as práticas de GC e conseguem resultados mais expressivos no mercado. Segundo Florêncio et al. (2005), essas organizações voltadas para o conhecimento possuem três pilares: infra-estrutura, pessoal e tecnologia. No segundo grupo estão empresas que não apresentam resultados tão bons, têm pouca participação no mercado externo e menor chance de atrair capital estrangeiro. Já o terceiro tipo, pequenas atrasadas, negligencia as práticas da GC e, conseqüentemente, não se destacam no mercado nacional ou internacional (CECATTO, 2005).

¹ Definido por Paiva (2005, p.72) como sendo “a habilidade de utilizar o conhecimento interpretando novos fatos e variáveis aparentemente não relacionados de forma a atingir os objetivos desejados”.

A pesquisa citada demonstra o que é colocado por Loureiro (2003, p.13): “(...) a competitividade tem vindo a recair sobre os intangíveis, sobre os recursos do conhecimento. Um capital que não é propriedade da organização mas das pessoas que o detêm”. Portanto a GC torna-se um elemento fundamental para as empresas que querem ser bem sucedidas nesta nova Era.

Conforme já citado, o tema conhecimento não é novo. No entanto, segundo Correia; Sarmiento (2005), somente de alguns anos para cá que os princípios, práticas e conceitos da Gestão de Conhecimento tornaram-se relevantes principalmente como um meio de aumentar a capacidade da organização em explorar o conhecimento.

Loureiro (2003, p.13) define GC como sendo “a abordagem sistemática para aumentar o valor e a acessibilidade do capital do conhecimento da organização para alcançar a máxima eficácia nos negócios e para propagar a inovação”. Este autor enfatiza que muitas vezes o conceito de gestão de conhecimento vem ligado ao de tecnologia de informação que é fundamental para facilitar a partilha e uso do conhecimento. Já Florêncio et al. (2005) define a GC como sendo a maneira com que as empresas lidam com o conhecimento pelas suas diversas fases como aquisição, compartilhamento, adaptação ao meio e aprimoramento. Por sua vez, o objetivo da GC, segundo Terra (2005, p.7), é “aumentar a produtividade do trabalhador do conhecimento”.

Destaca-se que a GC ainda não tem um conceito aceito por todos. Entretanto, o seu entendimento depende de outros fatores como as formas de conversão de conhecimento, o significado do termo “trabalhador do conhecimento” e também o papel da TI nesta nova Era. Estes três novos tópicos serão explicados nos itens subseqüentes.

1.4 FORMAS DE CONHECIMENTO

Antes de introduzir as formas como o conhecimento se dá e também as possíveis maneiras de convertê-lo, faz-se necessário apresentar a diferença conceitual entre dado, informação e conhecimento. Paiva (2005) apresenta a visão de dois autores sobre o assunto, Stair (1998 *apud* PAIVA, 2005) e Vivacqua (2001 *apud* Paiva, 2005). O primeiro faz a seguinte definição:

- ◆ Dado: são os fatos em sua forma primária;
- ◆ Informação: são os fatos organizados de tal forma que expressam um valor adicional que extrapola o fato em si;
- ◆ Conhecimento: constitui-se nas regras, diretrizes e procedimentos utilizados para a seleção, organização e manipulação dos dados (ou informações) a fim de torná-los úteis para uma atividade específica.

Já Vivacqua acrescenta uma dimensão a mais:

- ◆ Dado: conjunto de elementos dispersos sem significado próprio;
- ◆ Informação: conjunto de dados organizados, processados ou estruturados;
- ◆ Conhecimento: conjunto de informações não-codificadas e codificadas.
- ◆ Inteligência: capacidade de utilizar conhecimentos para entender novos fatos aparentemente não relacionados.

Este último autor introduz um conceito muito importante para o tema conhecimento ou GC que é a existência de dois tipos de conhecimento: tácito ou não-codificado e o codificado ou explícito.

Segundo Nonaka; Takeushi (1997 *apud* COSTA; GOUVINHAS, 2005) conhecimento tácito pode ser definido como aquele que reside na cabeça das pessoas e não está descrito em lugar nenhum. Já o explícito está disponível para mais pessoas, pois se encontra codificado (registrado) em alguns meios de armazenamento. Os mesmos autores, segundo Da Silva (2005), acrescentam que o conhecimento subjetivo é difícil de ser formalizado, explicado ou transferido para outra pessoa; enquanto que o explícito pode ser formalizado na forma de gráficos, tabelas, desenhos, fluxos, figuras, diagramas etc. e por isso é relativamente fácil de ser transferido ou reutilizado.

A criação do conhecimento, assunto abordado pela GC, baseia-se fundamentalmente na conversão dos dois tipos de conhecimento apresentados. Da Silva (2005) apresenta as formas de conversão que são sintetizadas na próxima tabela:

Classificação	Tipo de conversão	Forma como ocorre	Frase síntese
<i>Socialização</i>	Conversão de parte do conhecimento tácito de uma pessoa para conhecimento tácito de outra.	Diálogo freqüente; <i>brainstorming</i> , trabalho em equipe, relação do tipo “mestre-aprendiz”.	“Troca de conhecimento face a face entre indivíduos”.
<i>Externalização</i>	Conversão de parte do conhecimento tácito de uma pessoa em algum tipo de conhecimento codificado ou explícito.	Descrição do conhecimento utilizando-se planilhas, textos, figuras, regras, scripts etc; relatos orais e filmes, construção de metáforas e analogias por meio da linguagem figurada.	“O registro do conhecimento da pessoa por si própria”.
<i>Combinação</i>	Conversão do conhecimento explícito produzido por uma pessoa ao conhecimento explícito de uma organização.	Agrupamento (classificação e sumarização) de diversos tipos de conhecimento explícitos.	“Agrupamento de conhecimento registrado”.
<i>Internalização</i>	Conversão de parte do conhecimento explícito da organização em conhecimento tácito do indivíduo.	Prática individual (<i>learning by doing</i>); re-experimentar vivências e práticas; estudo individual de diferentes tipos de documentos (imagens, gráficos, textos etc).	“O aprendizado pessoal a partir da consulta dos registros de conhecimento”.

Tabela 3: Conversão do conhecimento, adaptado de Da Silva (2005)

A próxima figura busca mostrar a conversão de conhecimento de uma maneira mais didática.

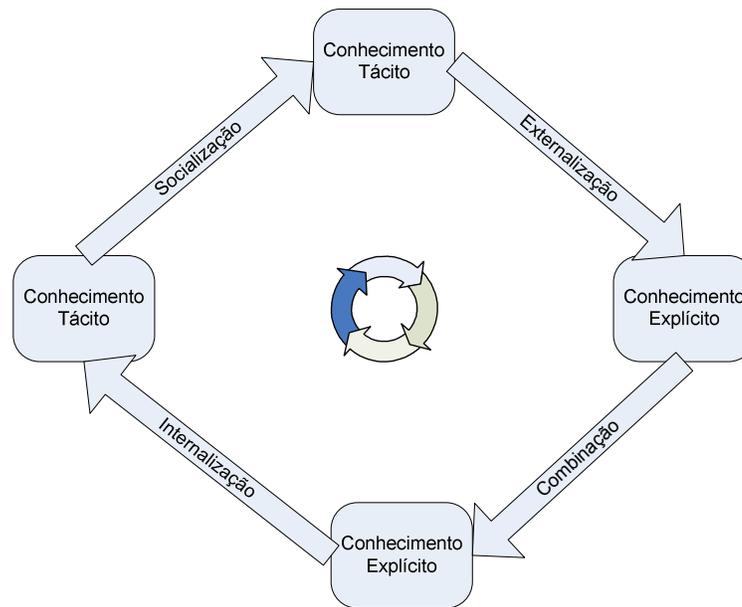


Figura 1: Processos de conversão do conhecimento, adaptado de Florêncio et al. (2005)

As empresas inseridas na Era do Conhecimento precisam aprender a lidar com as formas de conversão de conhecimento que estão intrinsecamente ligadas às práticas e atividades da GC. Estas são apontadas por Bontis e Fitzenz (2002 *apud* CORREIA; SARMENTO, 2005) como sendo:

- ◆ Geração do conhecimento: compreende a forma como os funcionários lidam com a informação, improvisam ou inovam.
- ◆ Integração do conhecimento: inclui a forma como os colaboradores de uma empresa transformam o seu conhecimento tácito em conhecimento explícito por meio da codificação de suas idéias e da inserção destas no âmbito da organização.
- ◆ Partilha do conhecimento: refere-se ao processo de socialização já explicado anteriormente.

As empresas inseridas na Era do conhecimento devem estar preocupadas com as formas de conversão do conhecimento principalmente para poder criar um ambiente onde o valor de seus colaboradores seja potencializado. Exatamente com este intuito, muitas empresas estão investindo em TI para prover o suporte adequado aos seus

funcionários e, principalmente, permitir a criação de meios onde a troca de conhecimento possa ocorrer de maneira flexível e fluente. Esse assunto é abordado com mais detalhe no próximo item.

1.5 TECNOLOGIA DA INFORMAÇÃO E A GESTÃO DO CONHECIMENTO

De acordo com Da Silva (2005), a utilização da TI na GC inicia-se nos anos 70. Nesta época, o foco que antes estava no processamento de dados volta-se para a informação, principalmente com a criação de sistemas de suporte à decisão gerencial. Em seguida, já nos anos 80, a TI direciona-se para os sistemas de informação baseados no conhecimento. Tempos onde a utilização de inteligência artificial e modelos matemáticos e estatísticos começam a ser usados para criar conhecimento por meio de cruzamento e estruturação de dados e informações alocados nos bancos de dados². No entanto, dos anos 90 em diante é que a TI começa a ter maior relevância para a GC, pois além de ter adicionado recursos extras, como os de multimídia e hipertexto, às ferramentas já existentes, também apresentou uma enorme evolução no avanço e na disseminação da Internet e Intranets que possibilitam acesso aos conhecimentos explícitos acumulados na corporação podendo até mesmo ajudar a personalizá-los.

Segundo Da Silva (2005), os recursos da TI podem facilitar em muito a externalização, internalização e combinação do conhecimento explícito. Segundo o autor a TI é fundamental para combinar conhecimentos explícitos; no entanto, não desempenha um papel tão importante em relação ao conhecimento tácito, podendo no máximo ser um instrumento de ligação entre aqueles que o detêm. Já Carvalho (2000) diz que o papel principal da TI é o de aumentar a velocidade com que a transferência de conhecimento ocorre. Este último autor acredita que a TI também auxilia a conversão do conhecimento tácito em explícito.

Atualmente existem diversas ferramentas de TI para auxiliar a transformação e disseminação do conhecimento no âmbito das organizações. Carvalho (2000) cita 8 grupos de *softwares* voltados para este propósito conforme elucidado nos pontos subseqüentes.

² Nessa década há o surgimento do termo *data mining*, ou seja, a garimpagem de dados em busca de informações valiosas que possam gerar conhecimento.

◆ Ferramentas Baseadas na Internet

- Caracterizam-se pela presença de hipertextos que facilitam a navegação por fontes de conhecimento.
- Neste grupo encontram-se as máquinas de busca, sites de navegação e os portais.

◆ Gerenciamento Eletrônico de Documentos (GED)

- Caracterizam-se por serem repositórios de documentos corporativos relevantes, ou seja, atuam como um depósito de conhecimento explícito estruturado. Muito dos conceitos utilizados neste grupo como catalogação e indexação são provenientes da Ciência da Informação.
- Os *softwares* presentes neste grupo procuram diminuir o tempo que os funcionários de uma empresa perdem tentando achar a informação de que precisam³.

◆ *Groupware*

- Neste grupo encontram-se os *softwares* desenhados para aumentar a cooperação e a comunicação interpessoal. Sua principal contribuição é aproximar grupo de pessoas que trabalham em conjunto, porém, fisicamente distantes.
- O groupware comumente funciona como base para aplicações de colaboração como correio eletrônico, grupos de discussão, correio de voz, vídeo-conferência, centrais de suporte e atendimento a clientes.

◆ *Workflow*

- Pode ser definido como um sistema informatizado que fornece suporte para processos padronizados de negócio. Seu objetivo é mostrar as etapas corretas para que um fluxo de processo seja
-

atingido assim como também acompanhar as atividades que o compõem.

- ◆ Sistemas para a construção de bases inteligentes de conhecimento
 - São sistemas que se utilizam de recursos da Inteligência Artificial como sistema especialistas ou redes neurais. Caracterizam-se por usarem conhecimentos prévios para resolver novos problemas.
- ◆ *Business Intelligence*
 - Refere-se a um grupo de ferramentas utilizado para manipular uma massa de dados em busca de informações relevantes para o negócio. Compõe-se em duas partes: *front-end* (interação com usuário) e *back-end* (armazenamento de dados).
- ◆ Mapas do Conhecimento
 - Este grupo de ferramentas é um pouco diferente dos citados. Esses mapas funcionam como uma lista estruturada sobre “quem sabe o quê”. Normalmente cada pessoa possui uma página com seu currículo ou perfil dizendo os assuntos que conhece, projetos dos quais já participou etc.
- ◆ Ferramentas de Apoio à Inovação
 - São ferramentas voltadas para Pesquisa e Desenvolvimento e que procuram colocar as pessoas em contato com conhecimentos tais como patentes, melhores práticas, modelos conceituais para que *insights* e idéias sejam estimuladas.

Segundo Nonaka e Takeuchi (1997 *apud* CARVALHO, 2000) cada um desses grupos favorece um tipo de conversão de conhecimento conforme mostra a tabela a seguir:

³ Segundo a File Net (2000 *apud* CARVALHO ET. AL., 2000) este tempo gasto corresponde de 10 a 40% da jornada diária de trabalho de um funcionário.

<u>De\Para</u>	Tácito	Explícito
<i>Tácito</i>	Socialização Mapas do Conhecimento	Externalização <i>Groupware</i> <i>Workflow</i> Bases Inteligentes de conhecimento
<i>Explícito</i>	Internalização Ferramentas de Apoio à Inovação	Combinação Intranet Gerência Eletrônica de Documentos <i>Business Intelligence</i>

Tabela 4: Classificação das ferramentas de TI de acordo com a conversão de conhecimento, adaptado de Carvalho (2000)

Apesar de não aparecer na classificação anterior, outros autores como Maier (2000 *apud* CORREIA; SARMENTO, 2005) e Da Silva (2005) englobam o ERP (*Enterprise Resource Planning*) explicitamente como parte das ferramentas de TI que auxiliam a GC. Davenport & Prusak (1998 *apud* DA SILVA, 2005, p.149) dizem que “os sistemas ERP procuram padronizar e normalizar os diferentes setores e funções, facilitando o compartilhamento de dados, informação e até conhecimentos”. Destaca-se também que o ERP é muitas vezes um facilitador para a implantação de outras ferramentas já citadas.

1.5.1 ERP

Segundo Ptak (2000), o avanço da informática acarretou o surgimento das ferramentas de planejamento para as indústrias manufatureiras. Nos meados da década de 60 surgiu o chamado MRP para o planejamento das necessidades dos materiais. Em seguida, com a evolução dos computadores, novas funções foram agregadas ao MRP como o gerenciamento da capacidade de produção. O MRPII foi a ferramenta que integrou o planejamento de materiais e capacidade fabril. Finalmente, com a adição de banco de dados relacional, linguagens de quarta geração e outras funcionalidades, surgiu um sistema capaz de integrar atividades de finanças, controladoria, vendas, produção, logística, *marketing*, RH, entre outras. Apesar de não servir somente para planejamento, a denominação inicial desta ferramenta, *Enterprise Resource Planning* (ERP) desenvolvida na segunda metade dos anos 90, prevaleceu. Alguns autores, como Davenport (2000), preferem usar o termo “sistemas gerenciais” para se referirem ao ERP, por se tratar de uma ferramenta com diversas outras funções como organizar, padronizar e integrar as informações

transacionais de uma organização (DAVENPORT, 1998 *apud* GAMBÔA; CAPUTO; BRESCIANI FILHO, 2004).

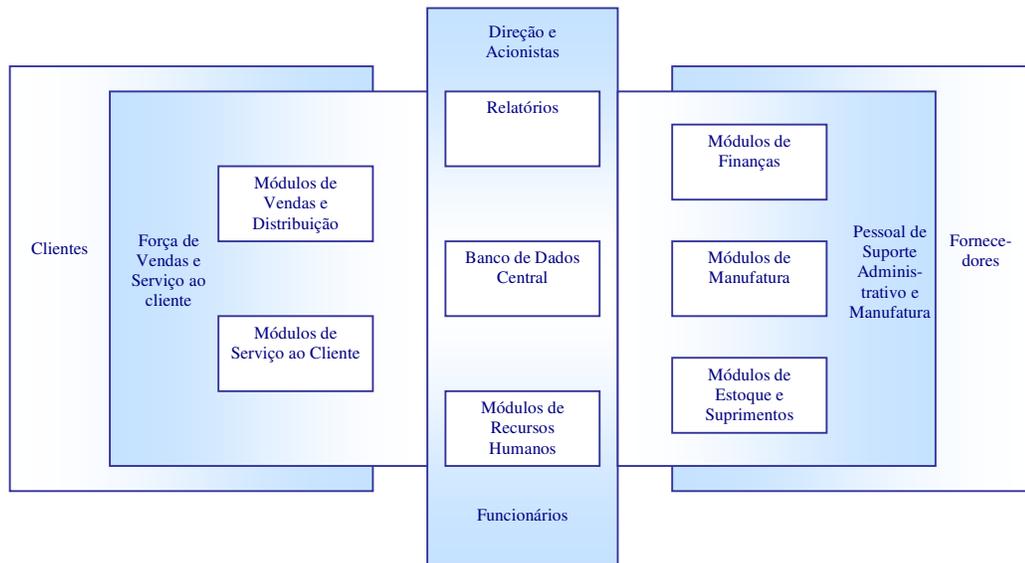


Figura 2: Arquitetura de um sistema ERP, adaptado de Davenport (1998, *apud* BOJIKIAN, 2004)

O reconhecimento do ERP é tão grande que Davenport (1998 *apud* GAMBÔA; CAPUTO; BRESCIANI FILHO, 2004, p.47) caracterizou-o como “o desenvolvimento mais importante do uso corporativo de tecnologia nos anos 90”. No Brasil, um grande número de empresas, cerca de 55 % segundo pesquisa da Yankee Group Brasil (2003 *apud* GAMBÔA; CAPUTO; BRESCIANI FILHO, 2004), implantaram algum tipo de ERP. Hoetzel (2005) aponta como vantagens da implementação de um ERP a integração das áreas de uma empresa, a visão de como as melhores práticas funcionam e também o provimento de importantes informações em tempo real. No entanto, o autor alerta que as empresas que queiram implantar um sistema de gerenciamento, como o ERP, devem saber que se trata de uma mudança radical em infra-estrutura, processos de negócio, estrutura organizacional assim como em papéis, lideranças e habilidades dos seus funcionários (HOETZEL, 2005).

O mais importante a se destacar neste tópico é o papel que os sistemas de gerenciamento, como o ERP, exercem na GC. Davenport (1998 *apud* DA SILVA,

2005) diz que como os ERPs baseiam-se em uma enorme base de dados que percorre toda a organização, eles passam a ser uma ferramenta que impacta a GC. Esse impacto dá-se na medida em que esses sistemas empresariais proporcionam a intermediação da ação de pessoas, aproximando assim quem possui um conhecimento específico de quem o está necessitando. Já Maier (2000 *apud* CORREIA; SARMENTO, 2005) pondera que o ERP pode funcionar como um apoio à codificação, pesquisa e recuperação de informação, à localização de peritos e à construção de comunidades.

Provavelmente o passo adiante em relação ao ERP seja o chamado *e-business*. Já com organização integrada como o primeiro sistema, o segundo auxiliaria no alcance do consumidor e fornecedores, ou seja, os parceiros da empresa em questão. O *e-business* engloba três estágios (NORRIS ET AL., 2000):

- ◆ *E-commerce*: permite que os processos de compra e venda ocorram da maneira mais fluida possível em relação aos processos e cultura organizacionais. Ex: EDI (*Electronic Data Interchange*), *e-procurement*.
- ◆ *E-business*: busca integrar todos os agentes da cadeia de suprimentos a fim de melhorar o nível de serviço do consumidor, reduzir custos e retirar o máximo de interrupções dos processos de negócio.
- ◆ *E-partnering*: responsável por estabelecer uma relação voltada ao consumidor na qual as organizações trabalham juntas para otimizar o valor total da cadeia de suprimentos. Portanto, o *e-partnering* busca incentivar colaboração que resultem em ganho para todos os agentes.

Nota-se portanto que, neste novo milênio, a grande maioria das aplicações de TI estão voltadas para a transmissão, codificação, disseminação e proteção de conhecimento. Entretanto, escondido atrás de todas essas maravilhas tecnológicas está um fator fundamental para o sucesso das organizações inseridas na Era do Conhecimento: o indivíduo. Conforme aponta Senge (1990 *apud* TERRA, 2001, p.65), o idealizador do termo “trabalhador do conhecimento”: “As organizações aprendem somente por intermédio de indivíduos que aprendem. O aprendizado individual não garante o aprendizado organizacional. Mas, sem ele, o aprendizado

organizacional não acontece”. Loureiro (2003, p.14) acrescenta que “numa era em que todos podem ter acesso à tecnologia, são as pessoas que fazem a diferença. Uma empresa que não seja feita por pessoas e para as pessoas, não é nada”.

No próximo item serão elucidadas algumas conseqüências de se ter o fator humano como o centro desta nova Era.

1.6 TRABALHADOR DO CONHECIMENTO

Conforme apontado por Angeloni (2001 *apud* FLORÊNCIO ET AL., 2005), uma organização inserida na Era do Conhecimento apresenta três pilares: infraestrutura, pessoal e tecnologia. Entretanto, na visão de Paiva (2000), um desses pilares se destaca atualmente, pois é inútil ter ativos materiais de alta tecnologia na ausência de pessoas com conhecimento para utilizá-los. A essas pessoas foi dada a denominação de “trabalhadores do conhecimento”. Essa expressão foi cunhada por Drucker, segundo BSJM (2005), para designar pessoas que foram educadas à utilização de conceitos, teorias e conhecimento ao invés da força bruta. BPT (2005) diz também que os trabalhadores do conhecimento lidam primariamente com a criação, uso e disseminação de informação.

As organizações necessitam cada vez mais de pessoas que saibam “navegar em um oceano de incertezas em meio a arquipélagos de certeza” (DA CUNHA; DA SILVA, 2002, p.79). A bússola desses trabalhadores será a capacidade de processar informação, aprender continuamente por meio das experiências vividas nas organizações e pela própria iniciativa pessoal (TERRA, 2005). Greenspan (*s/d apud* ASTD, 2005) também coloca que, além do conhecimento técnico, para os trabalhadores de hoje é fundamental saber criar, analisar e transformar informação além de estar apto a interagir eficientemente com outros. Finalmente, ASTD (2005) completa dizendo que capacitações como comunicar e aplicar conhecimento; sintetizar informação, resolver problemas e adaptar-se a ambientes dinâmicos são fundamentais. A figura a seguir mostra a diferença entre o processo tradicional e com o qual os trabalhadores do conhecimento devem saber lidar.

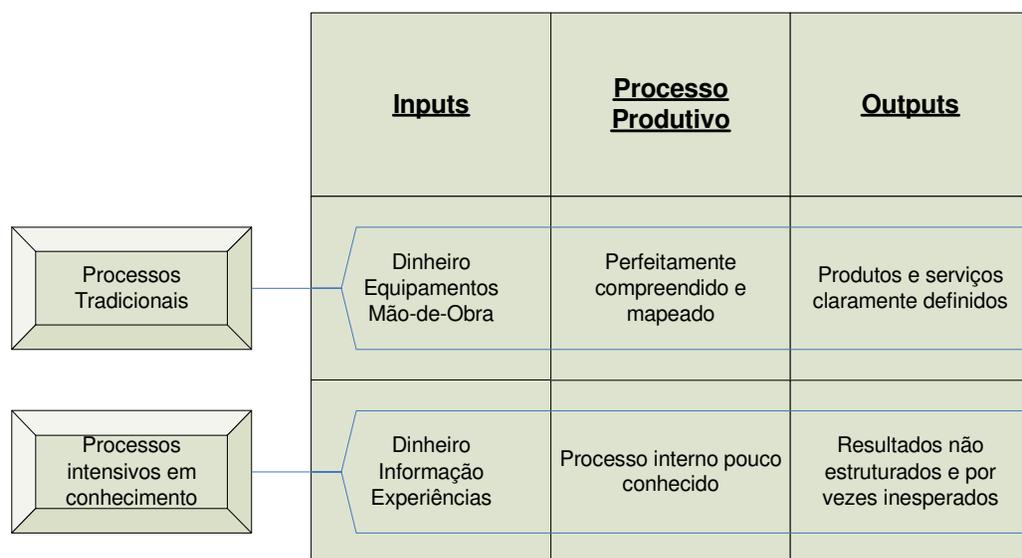


Figura 3: Diferença entre processos produtivos, adaptado de Terra (2005)

Correia; Sarmiento (2005) resume as competências que um trabalhador do conhecimento precisa ter pelo termo Infoliteracia, que nas palavras de Cheuk (2002 *apud* CORREIA; SARMENTO, 2005, p.6) seria a “capacidade dos trabalhadores em reconhecerem quando necessitam de informação, e saberem a forma de localizar, avaliar, organizar e usar efetiva e eficazmente, bem como a competência para criar, organizar e apresentar a informação, de uma forma adequada, à audiência alvo”. Entretanto, será que os trabalhadores deste início de século estão preparados para esta nova etapa de alfabetização ?

1.6.1 EDUCAÇÃO E APRENDIZADO

Da Cunha; Da Silva (2002) preconizam que a educação desempenha um papel primordial nos dias de hoje. Para ele, só a educação irá preparar as pessoas para lidar com os desafios inerentes à Sociedade do Conhecimento. Ao contrário de tempos atrás quando o aprendizado de uma profissão bastava, o cenário não mudava constantemente e a busca por novos conhecimentos não era tão importante; nesta nova Era, o fundamental é aprender a aprender.

Segundo Toffler (1990 *apud* ROSENBERG, 2001, p. 3), “os analfabetos do século 21 não serão os que não conseguem ler ou escrever, mas sim aqueles que não são capazes de aprender, desaprender, e reaprender”. Essa frase ilustra

apropriadamente a tão chamada “Era do Conhecimento” e a crescente necessidade das escolas começarem a ensinar maneiras de manipular a informação e não só transmiti-la indiscriminadamente (TOFFLER, 1984). Essa preocupação também é citada por Cassiolato; Lastres; Szapiro (2003) ao dizer que é extremamente improvável, em um ambiente de mudanças freqüentes, que todas as competências necessárias a uma pessoa possam ser adquiridas no seu primeiro quarto de vida por meio do processo educacional. Terra (2005) também compartilha com a visão que há uma grande probabilidade de que a educação sempre esteja atrasada frente aos avanços acelerados do mundo digitalizado e informatizado. Algumas pesquisas e números levantados podem mostrar esta nova realidade conforme colocado no parágrafo subsequente.

Segundo um estudo, do ano de 2002, da *Information Technology Association of América* (ITAA), companhias americanas têm dificuldade de contratar funcionários no nível gerencial para, em média, 50% dos cargos disponíveis. O motivo seria a falta de habilidades dos candidatos. A situação mostra-se mais grave no setor manufatureiro americano, onde em outro estudo⁴ foi reportado um sério problema de falta de competências. Além dessa inerente defasagem da educação frente os avanços tecnológicos, o Brasil ainda leva desvantagem por nunca ter apresentado um investimento educacional adequado por seguir um modelo orientado a mão-de-obra pouco qualificada como aponta Fleury (2001 *apud* FLORENCIO ET AL., 2005). O resultado de algumas pesquisas brasileiras apontadas por Terra (2001) é no mínimo alarmante:

- ◆ 63,8% dos trabalhadores não tem o ensino fundamental concluído.;
- ◆ 16,3% não apresentam um ano de estudo;
- ◆ apenas 9,16% possuem mais de 12 anos de estudo;
- ◆ apenas 3,7% do PIB são gastos em educação;
- ◆ apenas 4% da população tem diploma universitário.

⁴ Estudo feito pela *National Association of Manufactures*, no ano de 2001

Esses números chegam a ser um paradoxo para um Era onde o fator humano é colocado como primordial para o desenvolvimento e prosperidade de um país. O fator humano, a população, as comunidades, os indivíduos, o homem é o verdadeiro responsável por incorporar, transportar, criar, ampliar, ensinar, transmitir, aperfeiçoar e usar (bem ou mal) o conhecimento (DA CUNHA; DA SILVA, 2002).

Ressalta-se que o trabalhador do conhecimento deve apresentar uma educação sólida voltada para a informação e formas de lidar com o conhecimento, criatividade, empreendedorismo, inovação e principalmente ter a habilidade de aprender (e esquecer) (LEMOS, 2000). Concluí-se este item destacando que em relação à educação, o mais importante é instigar o indivíduo a sempre buscar novos conhecimentos, pois segundo Freitas (2005), vem surgindo um conjunto de tecnologias sem precedentes que irão exigir do trabalhador médio do setor administrativo ou produtivo um aprendizado contínuo em suas qualificações. Não é por acaso que alguns autores como Lundvall e Borras (1998 *apud* LEMOS, 2000); Cassiolato e Lastres (1999 *apud* LEMOS, 2000) também denominam esta nova era como Economia Baseada no Aprendizado.

1.6.2 A NECESSIDADE DE TREINAMENTO

Todos os tópicos tratados até aqui possuem de uma forma ou de outra uma ligação com este tema: necessidade de treinamento por parte das organizações inseridas na Era do Conhecimento. Conforme apontado por Lemos (2000), as inovações e mudanças recentes proporcionam diversas oportunidades. Entretanto, estas oportunidades só são aproveitadas por organizações preparadas. Esta preparação, por sua vez, está exigindo um investimento cada vez maior em pesquisa, desenvolvimento, educação e treinamento. Neste item, o papel do treinamento será focado sob dois pontos de vista: auxílio à capacitação e auxílio à mudança.

1.6.2.1 AUXÍLIO À CAPACITAÇÃO

Conforme mencionado diversas vezes, o conhecimento das pessoas é o que há de mais essencial nessa Era do Conhecimento que tem como foco o capital intelectual. Segundo ASTD (2005), trabalhadores com conhecimento e hábeis o suficiente para realizar as suas tarefas estão em extinção. O autor diz que há um *gap* (defasagem) entre o conhecimento/habilidade apresentado pelos empregados em

relação aos que são exigidos pelas empresas desta nova Era. Ele também cita que essa defasagem está levando às empresas a se preocuparem mais com a seleção, retenção e desenvolvimento contínuo de seus funcionários (os ativos mais valiosos que possuem).

A ilustração do cenário descrito acima é feita por ASTD (2005) ao apresentar um estudo realizado pela empresa de consultoria *PricewaterhouseCoopers*. Neste estudo, 70% das companhias citadas como as 1.000 maiores da revista *Fortune* alegaram que a falta de empregados bem treinados é a principal barreira para o crescimento. Por outro lado, um estudo feito pela *State of the Industry Report (ASTD⁵)* (*apud* ASTD, 2005) mostrou que, apesar da recessão e crise provocada pelos atentados de 11 de setembro de 2001, as empresas mantiveram seus investimentos em treinamento e, em alguns casos, até aumentaram. Diversas empresas levam o treinamento de seus empregados tão a sério que até constroem suas próprias universidades ou cursos como é o caso da *Embraer*, *Bain & Company*, *McDonald's*, *Companhia Vale do Rio Doce* e *GM*, dentre outras. Esta última, por exemplo, ofereceu em 2002 mais de 1.8 milhões de horas de treinamento.

Ressalta-se que muitos destes treinamentos corporativos são feitos de maneira virtual como é o caso da *Bain & Company* com a *Bain Virtual University (BVU)*. O *e-Learning*, definido de maneira geral como o treinamento auxiliado por computador, é um recurso cada vez mais utilizado devido a vantagens como (KRUSE, 2005):

- ◆ Redução de custos: redução de custos em relação ao aluguel de salas, viagens de funcionários, salário de instrutores etc.
- ◆ Aumento de retenção de informação: de acordo com estudo realizado por Fletcher (1991 *apud* KRUSE, 2005) esse aumento é de 25% sobre métodos tradicionais;
- ◆ Individualidade: na maioria dos casos é o estudante quem dita o ritmo do aprendizado.

⁵ ASTD significa *American Society for Training and Development*, ou seja, Sociedade Americana para o Treinamento e Desenvolvimento.

- ◆ Disponibilidade: o estudante pode completar o treinamento em casa, em intervalos no trabalho etc.
- ◆ Interatividade: recursos multimídias presentes no *e-learning* favorecem a interatividade e pró-atividade do aluno.
- ◆ Maior alcance: recursos de som, imagem e vídeo permitem que um número maior de pessoas se interesse por este tipo de aprendizado.

Portanto, a junção desse ambiente de mudanças aos avanços da tecnologia faz com que novos métodos de aprendizado comecem a surgir, como é o caso do *e-learning*, ou de maneira geral, a aquisição de conhecimento auxiliada por computador. No entanto, destaca-se que dificilmente a previsão feita por Tomas Edison, em 1922, possa se realizar. Por esta previsão, os livros usados no colégio e talvez os professores seriam substituídos com o avanço da tecnologia (neste caso, pelo cinema) (ROSENBERG, 2001).

Os métodos tradicionais de treinamento continuam sendo utilizados devido ao contato pessoal que proporcionam. Esse contato é muito rico quando se trata do conhecimento, pois proporciona o potencial para desenvolver todos os processos de conversão de conhecimento: socialização, externalização, combinação e internalização. Os métodos tradicionais são mais eficazes para passar conhecimento tácito assim como experiências pessoais entre os seus participantes, principalmente por meio da socialização. E, conforme apontado por autores como Maskell; Malmberg (1999 *apud* Humphrey, 2003) o conhecimento tácito é a única forma sustentável de vantagem competitiva.

Conforme já apontado em outros tópicos, as empresas nesta nova Era precisam estar sempre um passo a frente de seu concorrente. Isso exige atitudes inovadoras, aquisição de novas tecnologias, adoção de novas práticas, redefinição de processos e, conseqüentemente, profissionais que estejam à altura. Profissionais, estes, abertos às mudanças necessárias e que busquem o aprendizado contínuo através de interações para a troca de informações, conhecimento codificado e tácito (LEMOS, 2000). Portanto, o treinamento, de forma tradicional ou não, é extremamente importante para auxiliar a capacitação dos funcionários de uma empresa. A importância torna-se

ainda maior em um país em que, segundo o Banco Mundial, 40 % das companhias estão preocupadas com a capacitação de seus funcionários (maior índice entre os países pesquisados) (CAIXETA, 2005); mas onde grande parte das pessoas entra despreparada no mercado de trabalho devido ao sistema educacional deficiente.

1.6.2.2 AUXÍLIO À MUDANÇA

Esta necessidade de treinamento está intrinsecamente ligada com a anterior, com a diferença de levar em conta aspectos relativos às idiossincrasias de cada indivíduo. Conforme apontado por Salvador (2000 *apud* COSTA; GOUVINHAS, 2005), em grande parte das organizações três tipos de pessoas coexistem quando o assunto é mudança. Apesar do autor estar se referindo especificamente aos projetos de Gestão de Conhecimento, a idéia é expandida neste trabalho.

- ◆ Indivíduos resistentes: são contra praticamente todos os tipos de mudança por motivos diversos tais como: inaptidão à tecnologia, insegurança em relação ao seu próprio emprego ou simplesmente são “do contra”. Esse grupo deve ser informado e treinado ao máximo para não “contaminarem” o projeto.
- ◆ Indivíduos neutros: compõem a grande maioria das organizações. Necessitam de informações precisas, exemplos de sucesso em outras organizações e demonstrações claras de como a mudança irá auxiliá-los.
- ◆ Indivíduos inovadores: são os grandes patrocinadores de novas iniciativas nas empresas.
- ◆ Duck (1993), por sua vez, acrescenta uma nova categoria de pessoas: os sobreviventes às mudanças. Esse grupo é composto por indivíduos que já passaram por diversas iniciativas que não deram certo, ou seja, já experimentaram diversos programas de mudança que não produziram nenhum resultado prático.

Ansoff (1990 *apud* DEL VAL; FUENTES, 2003) define resistência como sendo um fenômeno que afeta o processo de mudança por meio do atraso de seu início, obstrução ou impedimento de sua implementação e aumento de custos. Outros autores como Maurer (1996 *apud* DEL VAL; FUENTES, 2003) definem resistência como inércia, ou seja, a conduta de querer manter o *status quo*. Há uma ampla gama

de razões para a resistência como as traçadas por Caruth et al. (1985 *apud* DEL VAL; FUENTES, 2003) e Kotter; Schlesinger (1979 *apud* DEL VAL; FUENTES, 2003): percepção de um prestígio menor após o processo de mudança, redução de poder e liberdade, medo, mudança de rotinas, mudança no conteúdo do trabalho, rearranjo de grupos formais ou informais, interesse próprio, má interpretação do processo ou até mesmo a característica de baixa tolerância à mudança de alguns indivíduos. Visto sob a perspectiva de Abraham Maslow, a mudança pode abalar a base da pirâmide das necessidades dos homens. Essa base é composta por segurança, ou seja, a proteção contra perigos e riscos; e pode ficar exposta em processos de mudança. Destaca-se que resistência nem sempre é onerosa, pois nem todas as mudanças agregam valor à organização.

Autores como Duck (1993) colocam a comunicação como fator primordial para a quebra da resistência em processos de mudança. Traduz-se por comunicação a informação que nela está contida. Quanto mais os funcionários souberem onde se pretende chegar, quais são as expectativas por parte da empresa, quais as novas habilidades que os empregados precisarão desenvolver, menor será o tamanho da resistência. O desconhecimento é a principal fonte de medo nesses casos. Não é por acaso que outros autores, como Del Val; Fuentes (2003), Ludmer; Rodrigues Filho (2003), Robey et al. (2002 *apud* HOETZEL, 2005), acreditam que o treinamento é uma importante ferramenta para romper a resistência. Esse treinamento pode apresentar diversos propósitos como: explicar o processo de mudança, clarear as expectativas e exigências na nova configuração, transmitir novos conhecimentos como também introduzir o usuário no novo sistema implementado.

O treinamento faz parte, na maioria das vezes, de um processo maior chamado gerenciamento de mudanças (*Change Management*). Este pode ser definido (BPR, 2005) amplamente como atividades voltadas para:

- ◆ Definir e incorporar novos valores, atitudes, normas e comportamentos que apoiem as novas formas de trabalhar e superem as resistências às mudanças.
 - ◆ Estabelecer um consenso entre consumidores e interessados (*stakeholders*) em relação às mudanças propostas.
-

◆ Planejar, testar e implementar em todos os aspectos a transição de uma estrutura organizacional ou processo para outro patamar.

Portanto, conforme colocado por Hawking; Foster; Stein (2004), *change management* não se trata somente de fornecer o treinamento certo, para a pessoa certa, no momento mais adequado. No entanto, o autor considera o treinamento antes e depois de processos de mudança essencial para o sucesso. O treinamento é colocado por Hawking; Foster; Stein (2004) como uma das melhores práticas⁶ do gerenciamento de mudanças.

A fim de salientar a importância do treinamento em processos de mudança, principalmente em mudanças disruptivas⁷, o exemplo do ERP será mostrado no item subsequente.

1.7 ERP E TREINAMENTO

Segundo um relatório do Gartner Group (*apud* ROBINSON, 2004 *apud* COSTA; GOUVINHAS, 2005), em um projeto de aplicação estratégica, aproximadamente 40% dos custos totais devem estar alocados em questões relacionadas às pessoas. A resposta crucial neste momento é saber se a empresa está realmente pronta.

Em um projeto de implementação de ERP, que é classificado como uma mudança disruptiva, estes aspectos precisam ser levados em conta. Como Turbit (2005) aponta, a própria palavra *Enterprise* (Empresa) mostra que, com o ERP, aquilo que é feito em uma área irá impactar as outras. A empresa só estará realmente pronta quando os usuários finais souberem os impactos de suas ações sobre a organização como um todo. Esta educação, no entanto, não ocorre da noite para o dia e demanda bastante treinamento focado na nova tecnologia com também nas pessoas (resistência). Hoetzel (2005) diz que em um projeto de ERP, a companhia deve focar tanto na educação individual (entender e saber utilizar os módulos do ERP) como na educação coletiva (saber como o ERP é integrado entre os departamentos).

⁶ O termo “melhores práticas” é definido por BPR (2005) como sendo processos, práticas ou sistemas identificados por organizações públicas e privadas por serem extremamente eficientes.

⁷ Mudanças disruptivas são mudanças de cunho estratégico, revolucionárias e que levam a transformações radicais visando, geralmente, a uma nova vantagem competitiva. Já mudanças

Muitos autores como McCredie e Updegrave (1999); Sumner (1999); Nelson e Somers (2001); Sederer et al (2001); Norris et al (2001); Nielsen (2002); Esteves et al (2003) apontam que a educação e treinamento são fatores críticos de sucesso em um projeto de ERP (GAMBÔA; CAPUTO; BRESCIANI FILHO, 2004). Gambôa; Caputo; Bresciani Filho (2004) também colocam a falta de conhecimento do produto/tecnologia e a presença de usuários não capacitados para operar o sistema como fatores de risco que causam grandes impactos no prazo, custo e qualidade da implementação (principais medidas de desempenho de um projeto).

Apesar de sua grande importância, o treinamento muitas vezes é subestimado. Um estudo feito pela META Group mostrou que os custos para o treinamento de usuários são de aproximadamente 9% do orçamento total (DOANE; VOLLMER, 2005).

Doane; Vollmer (2005) apontam duas possíveis causas dessa negligência constatada:

- ◆ A parcela do orçamento dedicada ao treinamento é um alvo fácil quando se tem um orçamento apertado ou que está além do previsto.
- ◆ O treinamento dos usuários finais ocorre normalmente um pouco antes da implementação dos módulos do ERP (*go-live*). Portanto, quando os prazos ou o orçamento estão apertados, esta etapa é feita “às pressas” ou muitas vezes eliminada.

A importância da educação e treinamento em projetos de ERP é destacada por Miller (2001 *apud* HOETZEL, 2005, p.11) quando diz que “pessoas são sempre elementos-chave em qualquer processo de melhoria, então métodos que ajudem-nas a avançarem na curva do conhecimento de uma tecnologia ou processo são extremamente importantes”. No entanto, um dos erros das organizações em projetos de ERP é o pensamento de que somente um treinamento na implementação é o suficiente. Como aponta Doane; Vollmer (2005), esse treinamento deve ser contínuo ou ocorrer em momentos-chave como: atualizações de versão que causam grande impacto, mudanças de processo ou melhorias, substituição ou contratação de

incrementais são pequenas ações que buscam melhorar a situação atual, mas que não levam a uma

funcionários. Portanto, as empresas devem estar cientes e possuírem medidas para evitar que, ao longo do tempo, o conhecimento se degrade, pois a tendência natural é essa.

Conclui-se que *change management* e, conseqüentemente, o treinamento são de fundamental importância para projetos, como a implementação de um sistema ERP.

A EMPRESA E O PROBLEMA

2. A EMPRESA E O PROBLEMA

Este trabalho de conclusão de curso é focado em um problema enfrentado pela empresa *M & R FONSECA CONSULTORIA EM INFORMÁTICA LTDA*⁸, da qual o autor deste trabalho é sócio minoritário. A empresa foi fundada em maio de 2004 visando prestar serviços para suprir algumas das necessidades prementes das organizações inseridas na Era do Conhecimento. Portanto, as áreas de atuação da *M & R*, listadas abaixo, estão intimamente ligadas aos assuntos relatados até o presente momento:

- ◆ Gestão de mudança (*Change Management*):
 - Levantamento e tratamento dos impactos organizacionais.
 - Redistribuição de carga de trabalho.
 - Levantamento e tratamento de riscos.
 - Tratamento de resistência à mudança:
 - Comunicação.
 - Treinamento.
 - Integração.
 - Eventos.

- ◆ Auxílio na implementação de sistemas de gerenciamento de conhecimento e de sistemas integrados:
 - Treinamento de usuário final:
 - Como utilizar a ferramenta.
 - Disseminação dos conceitos envolvidos na ferramenta (módulos do ERP, por exemplo).

O tópico treinamento aparece grifado, pois ele é razão do problema enfrentado atualmente pela empresa *M & R*, conforme apontado no próximo parágrafo.

A programação de treinamentos pode ser um problema extremamente complexo dependendo do tamanho da empresa. Esta pode ter um número extremamente grande de usuários a serem treinados, pode contar com filiais em diferentes regiões (bairro, cidades, municípios, estados) e ter uma demanda grande de cursos a serem ministrados para seus funcionários. O problema que a empresa *M & R* enfrenta atualmente é conseguir conciliar o horário de todos os funcionários, instrutores e de disponibilidade de salas a fim de montar um programa de treinamentos factível, com custo total e tempo (horizonte de programação) menores possíveis.

A programação de treinamento, portanto, envolve diversas considerações que podem ser classificadas como entradas, restrições e saídas. As entradas, listadas a seguir, são informações disponíveis no início do planejamento do *Timetabling* (TT).

Entradas

◆ Instrutores

- Disponibilidade do instrutor, ou seja, horários que podem ou preferem dar aulas.
- Possibilidade de deslocamento entre cidades, ou seja, alguns instrutores poderão dar aulas em diferentes filiais da mesma empresa.
- Curso(s) que cada instrutor está apto a ministrar de acordo com a classificação abaixo:
 - 2 (muito apto), 1 (apto), 0 (não pode dar o curso).

◆ Alunos ou funcionários

- Disponibilidade do aluno, ou seja, horários em que preferem ou podem assistir às aulas.
- Possibilidade de deslocamento entre cidades, ou seja, alguns alunos poderão comparecer em cursos dados em diferentes cidades.
- Designação de qual(ais) curso(s) cada aluno deverá assistir.

⁸ Nas demais partes deste trabalho a empresa será referenciada como *M & R*.

◆ Cursos

- Duração total do curso.
- Duração diária do curso.
- Tipo de sala (equipamentos) exigido para cada curso.
- Regra de precedência entre os cursos.

◆ Salas de aula

- Disponibilidade de cada sala.
- Capacidade máxima de cada sala.
- Equipamentos disponíveis.
- Local (cidade) da sala.
- Custo por hora de cada sala (planejada ou alugada).
- Custo fixo por ter aberto uma sala.

◆ Localidade

- Tempo de deslocamento entre localidades.
- Custo de deslocamento entre localidades (transporte, hospedagem etc.).

◆ Período

- Horizonte de tempo que deverá conter o início do primeiro curso e o término do último (horizonte de planejamento).
- Custo para cada dia extra utilizado em relação ao horizonte de planejamento desejado.

Restrições e/ou preferências

◆ Instrutores

- Disponibilidade (agenda) dos instrutores, ou seja, cada instrutor deverá ter sua preferência de horário respeitada sempre que possível.
 - Conhecimento de cada instrutor, ou seja, os instrutores devem ser alocados para os cursos que estão aptos a ministrar. Ressalta-se
-

que os instrutores deverão ser alocados preferencialmente nos cursos que apresentarem maior nota (ranking).

- Nenhum instrutor poderá dar dois cursos ao mesmo tempo.
- Nenhum instrutor poderá dar dois cursos seguidos que não respeitem o tempo de deslocamento entre salas de aula. Destaca-se que as salas de aula podem estar em cidades diferentes caso a empresa tenha mais de uma filial.

◆ Alunos ou funcionários de uma empresa:

- Disponibilidade (agenda) dos alunos, ou seja, cada aluno deverá ter sua preferência de horário respeitada sempre que possível.
- Alocação de alunos em cursos, ou seja, todos os alunos deverão ser matriculados nos cursos a que foram designados.
- Nenhum aluno poderá assistir dois cursos ao mesmo tempo.
- Preferencialmente, um aluno não deverá ficar trocando de sala frequentemente.
- Nenhum aluno poderá assistir dois cursos seguidos que não respeitem o tempo de deslocamento entre salas de aula. Destaca-se que as salas de aula podem estar em cidades diferentes caso a empresa tenha mais de uma filial.

◆ Cursos

- Alocação dos cursos no tempo de modo a respeitar duração total e diária.
 - Se a sessão de um curso for alocada a uma determinada sala, então esta não deverá mudar nos outros dias.
 - Dois ou mais cursos não podem ocorrer concomitantemente na mesma classe.
 - Cada curso deve ser alocado em uma sala adequada, ou seja, onde existam equipamentos adequados (computador, projetor, *softwares* instalados etc.).
-

- Há cursos que apresentam regra de precedência, ou seja, alguns cursos só poderão começar dado que outro(s) terminou(aram).

◆ Salas de aula

- Cada sala de aula apresenta uma capacidade máxima de alunos que deverá ser respeitada.
- A disponibilidade de cada sala deverá ser respeitada.
- Uma sala não pode comportar dois cursos ocorrendo ao mesmo tempo.

Saídas

- ◆ TT gerado.

Objetivo

A solução do problema deverá tentar minimizar o custo total do TT. Este custo será melhor explicado futuramente.

Destaca-se que os pontos supracitados visam ilustrar o problema a ser tratado neste trabalho. Porém, provavelmente, o leitor já deve ter se deparado com um problema desses, já que TT aparece comumente em escolas e universidades onde há a presença de alunos, professores, salas de aula e cursos que devem ser alocados de maneira a satisfazer (ou melhor atender) as restrições impostas⁹. Estas restrições normalmente referem-se às capacidades das salas de aula (número de pessoas e equipamentos disponíveis), disponibilidade dos professores e suas preferências (não dar aulas sexta-feira à tarde). Algumas escolas ou universidades também levam em conta as preferências dos alunos como, por exemplo, não ter “janelas”¹⁰ extensas.

O motivo que levou o autor deste trabalho a lidar com o problema de TT foi a extrema complexidade em montar uma programação de treinamento de usuários finais (*end users*) para empresas grandes. A quantidade de alunos/funcionários pode

⁹ Três comuns restrições são: um aluno não pode estar em duas aulas ao mesmo tempo, um professor não pode estar alocado em dois cursos ao mesmo tempo e uma sala de aula não pode ter dois cursos ao mesmo tempo.

ultrapassar 1.000, número este que já inviabiliza a solução manual do problema de TT. Este se torna crítico no caso de empresas que possuem mais de uma filial que estão localizadas em cidades diversas. Portanto, a elaboração de um *software* para resolução do problema de TT é de extrema relevância para a empresa da qual o autor deste trabalho é sócio. Pretende-se, neste trabalho, desenvolver um *software* que lide com diversas situações (objetivos) que aparecem nos projetos conduzidos pela empresa *M & R*. Destaca-se também que certas restrições deverão ser totalmente respeitadas e outras atendidas da melhor forma possível (vide nota 8). Destaca-se, porém, que uma melhor definição do problema será dada após a revisão bibliográfica por meio de um estudo de caso, sendo o intuito deste item familiarizar o leitor com o objeto de estudo deste trabalho de conclusão de curso.

¹⁰ O termo “janela”, neste caso, refere-se ao espaço de tempo entre duas aulas consecutivas para um determinado aluno. Muitos alunos não gostam de ter uma aula de manhã e a outra somente no final da tarde.

REVISÃO BIBLIOGRÁFICA

3. REVISÃO BIBLIOGRÁFICA

3.1 O PROBLEMA DE TIMETABLING

O problema de TT, segundo Tripathy (1984), tem sido tratado por pesquisadores usando diversos métodos de solução desde o final da década de 50. O autor aponta que este problema continua sendo de interesse devido à sua complexidade e natureza. De Werra (1997), por sua vez, diz que este interesse vem principalmente de especialistas em Pesquisa Operacional e Matemática devido à natureza combinatória do problema, ou seja, para achar uma solução ótima ou conveniente, um grande número de combinações deve ser analisada. Seguindo este raciocínio, Abramson (1991) enfatiza que apesar deste problema ter sido muito estudado no passado, um limitado sucesso é reportado. A complexidade dá-se principalmente pelo número de variáveis e restrições que este tipo de problema envolve, o que, de certo modo, inviabiliza a utilização de programação matemática para chegar a uma resposta ótima. Entretanto, este assunto será melhor tratado no item sobre as abordagens de solução do TT.

Conforme Lemos (2000) descreve, os problemas publicados a partir de 1960 tratam de diversos tipos de TT, tais como *sports timetabling*, *railway timetabling*, *school timetabling*, *university timetabling*. Carter e Laporte (1997 *apud* ALVAREZ-VALDES; CRESPO; TAMARIT, 2002) acrescentam outras formas com o qual este tipo de problema pode aparecer na literatura: *course timetabling*, *class-teacher timetabling*, *student scheduling*, *teacher assignment* e *classroom assignment*. Chand (2002) enriquece essa lista citando o *project scheduling*, *bus scheduling*, *airflight schedules*. Bastante comum na literatura também são os problemas conhecidos como *examination timetabling problem*, *staff scheduling problem* e o *nurse rostering*, dentre outros. Todos estes problemas, apesar de nomenclaturas distintas, apresentam grande (ou total) similaridade. Essa lista foi citada para mostrar ao leitor a importância do problema de TT dado a variedade em que pode aparecer.

Sob o aspecto conceitual dessa enorme gama de problemas, Bartak; Rudova (2001) faz uma relevante constatação ao mostrar a conexão e similaridade entre problemas classificados como *planning*, *scheduling* e *timetabling*. O autor diz pode haver confusão ou ambigüidade na classificação de um problema em uma dessas

categorias. Isto se dá devido à pequena diferença que há entre *planning*, *scheduling* e *TT* de acordo com a situação ou ambiente tratado. Para o autor, essas três categorias lidam com três entidades – atividades, tarefas e recursos – e que o problema principal é alocar as atividades aos recursos ou vice-versa. A idéia de tempo (alocação das atividades no tempo), muito importante no problema de TT, estaria escondida nos conceitos de tarefa e recurso. A seguir, alguns comentários sobre as três entidades citadas são expostos:

- Atividade: esse conceito aparece nas três categorias sob diferentes nomes – ação (*planning*), operação (*scheduling*) e curso (*timetabling*). Normalmente a atividade está relacionada com uma duração ou uma janela de tempo em que deve ocorrer.

- Recursos: as atividades são normalmente alocadas aos recursos dos quais consomem capacidade. Os recursos podem ser de dois tipos: renováveis ou não-renováveis. Salas de aula são exemplos do primeiro tipo, já que sua capacidade volta ao estado original assim que a atividade acaba (curso). Como exemplo de recurso não-renovável pode-se citar o combustível dado que ao terminar uma atividade este recurso não estará mais no nível original (nota-se que a atividade pode acrescentar ou gastar o recurso, mas em ambos os casos o estado final é diferente do original). Outra característica do recurso é a sua capacidade. No caso de recurso não-renovável, este parâmetro irá indicar o tamanho do recurso (tamanho do tanque, por exemplo). Já no caso do recurso renovável, este parâmetro limita a quantidade de atividades que poderão ocorrer ao mesmo tempo. Em problemas de TT, a sala de aula tem capacidade unitária, ou seja, somente um curso pode ocorrer em um determinado tempo. Destaca-se que uma atividade pode consumir mais de um recurso ao mesmo tempo. No caso do TT, o curso irá consumir, por exemplo, a sala de aula como também o instrutor.

- Tarefas: as atividades normalmente são agrupadas em tarefas. Este agrupamento permite explicitar relações entre as atividades como por exemplo regras de precedência. Portanto, para completar uma tarefa, as atividades devem ser realizadas seguindo um roteiro.

A tabela a seguir busca sintetizar as diferenças entre as três categorias citadas – *planning*, *scheduling* e *timetabling*:

	<i>Planning</i>	<i>Scheduling</i>	<i>Timetabling</i>
<i>Duração</i>	Não	Sim	Sim (normalmente fixa)
<i>Janela de tempo</i>	Não	Sim	Às vezes
<i>Recurso renovável</i>	Sim	Sim	Sim
<i>Recurso não-renovável</i>	Sim	Sim	Não
<i>Restrição de capacidade</i>	Sim	Sim	Sim, freqüentemente unitária

Tabela 5: Diferença entre *planning*, *scheduling* e *timetabling* (BARTAK; RUDOVA, 2001)

Ressalta-se que Bartak; Rudova (2001) dizem que a principal diferença entre *planning* e *scheduling* é que neste último as atividades estão previamente definidas enquanto no planejamento (*planning*) elas devem ser geradas. Completando, Wren (1996 *apud* BARTAK; RUDOVA, 2001) destaca que o problema de TT pode ser considerado um problema particular de *scheduling*. Muller; Bartak (2002) conceitua TT como sendo uma forma de *scheduling* onde o objetivo é alocar atividades em espaços de tempo de recursos respeitando algumas restrições. Seguindo este raciocínio, o problema de TT é semelhante ao enfrentado por empresas de manufatura, as quais precisam realizar o sequenciamento da produção que, grosso modo, significa a alocação de operações em máquinas no tempo. Esta similaridade é constatada por Carter; Tovey (1992) ao citar um problema tratado por Kolen, Lenstra e Papadimitriou – problema de sequenciamento de máquinas, onde as entidades envolvidas são os *jobs* (atividades) com tempos de início e durações e há um conjunto de máquinas com diferentes capacidades que serão consumidas pelas atividades.

Apesar de todas as semelhanças e gamas de problemas supracitados, na literatura pesquisada, apenas dois autores colocam que o problema de TT também é enfrentado por empresas. Enzhe; Sung (2002) dizem que o TT é encontrado principalmente em instituições educacionais e em menor relevância em empresas de

negócio, sendo que Bufe et al. (2001) não faz essa diferença de relevância. Entretanto, o que realmente ocorre é que na literatura os problemas de TT são tratados exclusivamente em instituições educacionais; negligenciando, de certo modo, a necessidade premente de soluções para este mesmo tipo de problema por parte de empresas de negócio inseridas na Era do Conhecimento. Portanto, em face desta deficiência e como meio de gerar *insights* sobre o problema tratado neste trabalho, as constatações e considerações apresentadas daqui por diante referem-se ao problema geral de TT ou a um dos tipos de problema que mais se assemelham ao das empresas e que aparecem na literatura com mais frequência como *university timetabling*¹¹, *school timetabling* e *examination timetabling*. Visando explicitar a semelhança conceitual entre estes problemas, a definição de alguns deles é apresentada nos tópicos subseqüentes.

3.1.1 UNIVERSITY TIMETABLING

Segundo Dakalaki; Birbas (2005), várias vezes durante um período letivo funcionários da administração, responsáveis por recursos e os chamados “schedulers” da universidade ocupam-se do conhecido problema de TT. Decisões em relação aos pontos subcitados precisam ser tomadas:

- ◆ Discentes a grupos de estudantes.
- ◆ Docentes em cursos.
- ◆ Cursos em tempos.
- ◆ Cursos em salas de aula.

As hipóteses e regras aplicadas nestas decisões podem adquirir diversos formatos dependendo da instituição, ou seja, dependendo da estrutura da universidade, do currículo oferecido, da disponibilidade de recursos, dentre outros aspectos.

Daskalaki; Birbas; Housos (2004) definem o *university* TT como sendo um processo de alocação de cursos universitários em períodos específicos ao longo de 5

¹¹Carter; Laporte (1997 *apud* ALVAREZ-VALDES; CRESPO; TAMARIT, 2002) descreve que o *university timetabling* pode conter diversos subproblemas tais como *course timetabling*, *class-teacher timetabling*, *student scheduling*, *teacher assignment* e *classroom assignment*.

dias de aula por semana como também em salas adequadas ao número de estudantes registrados e necessidades de cada curso (equipamentos). O objetivo final seria a construção de grades horárias semanais satisfatórias. Por satisfatório, entende-se uma grade horária que respeite regras operacionais, requisitos da instituição sendo que ao mesmo tempo respeite os requerimentos e tente satisfazer os desejos do corpo docente como também do corpo discente. Algumas considerações colocadas pelo autor e que demonstram a complexidade do problema são citadas abaixo:

Cursos

Um curso oferecido por uma universidade pode conter aulas normais, aulas de exercício e/ou aulas práticas (aula em laboratórios). As aulas podem ser dadas por mais de um professor que se revezam na atividade. As aulas de exercício podem ser dadas por um professor diferente da aula normal e também pode requerer divisão dos alunos em grupos menores. Já as aulas em laboratório ocorrem normalmente em salas equipadas e podem ser ministradas pelos mesmos professores da matéria, professores diferentes da matéria normal e/ou por assistentes.

Os cursos também podem ser divididos em dois tipos: obrigatórios e optativos. Os obrigatórios são aqueles considerados básicos pelo Departamento no ensino de seus alunos. Já os cursos optativos ocorrem normalmente no final do estado no aluno na faculdade quando as matérias básicas já foram cursadas e o foco passa a ser um enriquecimento no currículo do aluno com conhecimentos fora da sua área ou especializações.

Recursos

No problema de *university* TT, recursos referem-se às pessoas ou salas de aula. A disponibilidade de professores depende principalmente de seu contrato com a entidade. Os professores podem ser obrigados a estar disponíveis integralmente ou parcialmente. Vale lembrar que os professores também participam de diversas atividades na faculdade como palestras, graduação, mestrado, doutorado, pesquisa, simpósios, congressos etc. que exercem grande influência nos horários dos docentes. Já a disponibilidade das salas de aula é definida no sistema seguido pela própria universidade. Algumas classes têm disponibilidade total enquanto outras

compartilham aulas com outros departamentos da mesma faculdade (universidade). O compartilhamento de recursos entre departamentos é bastante comum no caso de auditórios ou laboratórios que contenham equipamentos caros, ou seja, recursos escassos.

Regras

A geração de TT para uma universidade deve levar em conta regras institucionais assim como tentar satisfazer desejos expressos por professores (e alunos). Eis alguns exemplos dessas regras e desejos:

- ◆ Não haver colisão: professor dar aula concomitantemente em dois lugares distintos. Alunos assistirem a duas diferentes aulas ao mesmo tempo etc.
- ◆ O TT estar completo: todas as alocações foram realizadas.
- ◆ Preferência de horário de cada professor deve ser respeitada, se possível.
- ◆ Horários dos estudantes devem ser o mais compacto possível: os alunos não devem ter em sua grade horária uma distância muito grande entre aulas consecutivas. Entretanto, a construção da grade horária deverá prever um tempo para almoço.
- ◆ Minimização de mudança de classes: os alunos durante os primeiros anos costumam assistir às aulas em grandes auditórios. Portanto, a mudança de classes deve ser evitada sempre que possível.

Ressalta-se que estas regras são um pequeno exemplo do que pode ser encontrado na literatura.

3.1.2 SCHOOL TIMETABLING

O conceito deste tipo de problema não será muito explorado por ser extremamente semelhante ao *university timetabling*. O *school timetabling* é definido por Santos; Ochi; Souza (2004) como sendo a programação semanal de encontros entre professores e turmas de maneira a assegurar que requerimentos e restrições sejam satisfeitos. Frangouli; Harmandas (1995) apontam que a diferença entre o *school* e o *university timetabling* recai somente nas restrições que devem ser satisfeitas. Um exemplo dado pelo autor é que, em universidade, janelas entre as

aulas, apesar de serem minimizadas, são permitidas; já no colégio esta situação é, geralmente, proibida. Atualmente existem escolas tão flexíveis – escolha de matérias, aulas de laboratório, aulas de exercício, cursos optativos etc. – que a principal diferença com relação ao *university timetabling* diz respeito à escala, pois normalmente problemas de universidade envolvem um maior número de alunos. Entretanto, Frangouli; Harmandas (1995) diz que, em relação à complexidade, os dois tipos de problema – *university* e *school* - apresentam o mesmo grau de dificuldade.

3.1.3 EXAMINATION TIMETABLING

Carter (1986) menciona que grande parte das intuições de educação enfrenta no final de cada período o problema de alocar os exames/provas no tempo. Carter; Laporte (1997 *apud* HUSSIN; KENDAL, 2005, p.2) definem *examination timetabling* como sendo “a alocação de exames em um limitado número de períodos disponíveis de tal maneira que não haja conflitos ou colisão”¹². Merlot et al. (2002) similarmente diz que o *examination timetabling* é a programação de exames em períodos de tempo de tal maneira que satisfaça um conjunto de restrições. Este último autor cita as restrições mais comuns neste tipo de problema:

- ◆ Colisão: nenhum estudante deve ter dois exames ao mesmo tempo.
 - ◆ Capacidade: a quantidade de alunos alocados para todos os exames na mesma sala e no mesmo horário deve ser menor do que a capacidade do local.
 - ◆ Capacidade de exame: o número total de exames no mesmo horário deve ser menor do que o número especificado.
 - ◆ Disponibilidade de exame: alguns exames devem acontecer em um limitado número de horários ou há exames que só devem ocorrer no sábado etc.
 - ◆ Disponibilidade das salas: algumas salas podem não estar disponíveis em todos os horários.
 - ◆ Restrição de exames consecutivos: alguns pares de exames devem respeitar ordens predeterminadas (ex: exame A deve vir depois do B).
-

- ◆ Relação Sala/Exame: alguns exames demandam salas específicas.
- ◆ Restrições dos alunos: pode haver restrições tais como nenhum aluno deverá ter dois exames marcados em três períodos consecutivos ou nenhum aluno deverá ter três ou mais exames consecutivos.
- ◆ Exames comuns: as provas podem ser iguais e serem aplicadas para uma grande quantidade de alunos simultaneamente. Algumas universidades preferem colocar tais provas no início do período de exames.

As instituições educacionais podem utilizar-se completamente ou parcialmente destas restrições ou até acrescentar outras de tal maneira a obter o que classificariam de um TT de boa qualidade. Conforme será visto posteriormente, a qualidade de um TT está intimamente ligada com a capacidade de atender em maior ou menor grau as restrições do tipo “é preferível ter”.

Nota-se pela definição dada que o problema de programação de exames é extremamente parecido com o *university timetabling*. Este raciocínio é compartilhado por Carter (1986) quando diz que é difícil desenhar uma linha separando o *examination* do *course* (ou *university*) *timetabling problem*. Frangouli; Harmandas (1995) aponta que uma das principais diferenças entre estes dois tipos de TT seria o fato de que no *examination* TT há somente uma prova, ao contrário das diversas matérias que há em um semestre. O autor também diz que o *exam scheduling* normalmente é mais complexo do que o *course timetabling problem*.

A principal motivação deste item foi mostrar ao leitor a relevância e riqueza que o assunto TT representa. Os diversos formatos, ambientes e complexidades que esse problema pode assumir fazem com que o TT um assunto cuja demanda de pesquisa e alternativas de solução não deverá ser interrompida tão facilmente. Aliás, tudo leva a crer que o aumento dos cursos demandados por empresas, acréscimo da parcela de estudantes em nível elementar, fundamental e superior assim como oferecimento de cursos complementares resultará em uma intensificação de pesquisas neste assunto, principalmente para suprir o *gap* (lacuna) entre a teoria e a prática. O sucesso da discussão em torno deste assunto já pode ser constatado através do crescimento do

¹² “The assigning of examinations to a limited number of available time periods in such a way that

interesse percebido no grupo de trabalho EURO WATT (*Working group on Automated Timetabling*¹³) ou nas séries de conferências internacionais do PATAT (*Practice and Theory of Automated Timetabling*¹⁴) (AICKELIN ET AL., 2003). Exatamente esse sucesso e a tentativa de suprir a lacuna mencionada, atualmente há até uma competição voltada para a resolução de instâncias de problemas de TT como é o caso da *International Timetabling Competition* (ARNTZEN, H.; LOKKETANGEN, A., 2005). Conforme aponta Aickelin et al. (2003), o futuro do TT é ser um problema do interesse de diversas disciplinas e, conseqüentemente, ser solucionado por abordagens multidisciplinares. Atualmente, o problema moderno de TT já se encontra na interface entre a pesquisa operacional e a inteligência artificial. Outras considerações e conceitos, como a complexidade deste tipo de problema, são tratados nos próximos itens.

3.2 COMPLEXIDADE E ESPECIFICIDADE

Elmohamed; Coddington; Fox (1997) definem TT com sendo um problema combinatório de otimização multi-dimensional, não-euclidiano, multi-restrito e que, conseqüentemente, é muito difícil de ser resolvido. Alvarez-Valdes; Crespo; Tamarit (2002) colocam que a principal dificuldade relacionada com o problema de TT é o seu tamanho. Ele envolve, normalmente, um grande número de entidades como estudantes, professores, cursos e salas ligados por condições e objetivos de tal maneira que um enorme número de variáveis e restrições devam ser analisadas para se chegar a uma resposta. Chand (2002), por sua vez, menciona que a principal dificuldade está ligada ao grande número de restrições conflitantes presentes no problema e que precisam ser resolvidas.

Visando exemplificar a complexidade do problema de TT, Frangouli; Harmandas (1995) apresentam um exemplo. Neste exemplo, considera-se o período de um semestre de uma universidade. Nela há 30 matérias diferentes para serem programadas sendo que cada uma delas é ensinada em 4 períodos distintos por semana. Assume-se também que a semana é composta por 5 dias e que cada dia é subdividido em 10 períodos e que há somente 3 salas de aula para abrigar os cursos.

there are no conflicts or clashes”.

¹³ Grupo de trabalho em *timetabling* automatizado.

Neste caso, que pode ser considerado um problema simples, o espaço de soluções que deve ser analisado para a construção da grade horária semanal é tão grande quanto $(5 \times 10 \times 3)^{4 \times 30}$, ou seja, $1,35 \times 10^{261}$. Fazendo uma analogia, se a grandeza deste número fosse em quilômetros, ele seria 10.253 vezes maior do que a distância do Sol e a Terra. O problema torna-se achar neste espaço de soluções gigantesco uma solução que respeite as restrições do problema (solução viável).

Seguindo o raciocínio apresentado no parágrafo anterior, não é difícil imaginar que um problema imensamente maior é achar a solução ótima para o problema de TT. Badica; Lalescu (2005) mencionam que o algoritmo perfeito para esta tarefa deveria fazer uma procura exaustiva neste espaço de soluções e no final escolher a melhor. No entanto, o mesmo autor diz que isto é, infelizmente, impraticável devido ao crescimento exponencial que o tempo de execução apresentaria. A complexidade é tão grande que o autor diz que até mesmo a comparação de duas soluções para saber qual delas é a mais apropriada para o problema tratado não é uma tarefa trivial. A seguir serão elucidados importantes conceitos presentes na manipulação de problemas de notável complexidade.

3.3 TEORIA DA COMPLEXIDADE

Segundo DICTIONARYLABORLAWTALK (2005), a Teoria da Complexidade é parte da Teoria da Computação e trata dos recursos necessários durante os procedimentos necessários para resolver um dado problema. Os recursos mais comuns são o tempo e espaço que significam respectivamente a quantidade de passos e memória necessários para resolver o problema em questão. Outros recursos como número de processadores em paralelo também podem ser considerados.

Nesta teoria encontram-se os chamados problemas do tipo P, NP, NP-Complete (NP-Completo) e NP-*Hard* (NP-Difícil):

Problemas do tipo P

A classe de problemas do tipo P são os problemas de decisão que podem ser solucionados em um tempo polinomial por algoritmos (determinísticos). Um algoritmo resolve um problema em tempo polinomial se no pior caso a eficiência

¹⁴ Prática e teoria do *timetabling* automatizado. Essas conferências ocorrem a cada dois anos (ASAP,

com relação ao tempo pertence a uma função do tipo $O(p(n))$ onde $p(n)$ é um polinômio cujo parâmetro de entrada é o tamanho do problema “n”. Problemas que podem ser resolvidos em tempo polinomial são classificados de tratáveis, caso contrário de intratáveis. Já a natureza do algoritmo, segundo Reingold et al. (1977 *apud* IVANQUI, 2005), pode ser classificada em duas categorias:

- ◆ Determinístico: um algoritmo é determinístico caso ele explore uma alternativa de solução do problema em questão por vez. Ao terminar esta alternativa ele começa a explorar outras e assim por diante.
- ◆ Não-determinístico: um algoritmo é dito não determinístico caso ele explore mais de uma alternativa de solução por vez criando cópias de si em cada uma delas. Se uma cópia encontrar uma alternativa improdutiva, ela pára. Se encontrar uma solução, a cópia avisa as demais cópias que param a execução.

Problemas do tipo NP

A classe NP inclui os problemas que podem ser resolvidos por algoritmos polinomiais não-determinísticos. Nesta categoria encontram-se os problemas classificados como NP-Completo e como NP-Difícil¹⁵. Apesar da diferença entre estes dois últimos tipos de problema não ser tratada neste trabalho, interessa saber que problemas classificados como NP-Completo ou NP-Difícil apresentam, segundo George; Cunha (2002), ordem de complexidade exponencial. Em outras palavras, isto quer dizer que o esforço computacional cresce exponencialmente de acordo com o tamanho do problema.

Dada a conceituação apresentada acerca da complexidade dos problemas de decisão, pode-se dizer que, segundo Daskalaki; Birbas; Housos (2004), os problemas de TT, em quase todas as suas formas, são classificados como NP-Completo. Portanto, o tempo de solução cresce drasticamente de acordo como o tamanho do problema (tamanho do espaço de solução). Outros autores compartilham, de uma forma ou de outra, desta idéia:

2005).

¹⁵ A diferença entre esses dois tipos pode ser encontrada nesta referência: <http://en.wikipedia.org/wiki/NP-complete>.

- ◆ Even; Atai; Shamir (1976 *apud* BUFE ET AL., 2001) apontam que o problema de TT é NP-Completo sempre que há indisponibilidade de professores, classes ou salas de aula. Nota-se que a escassez de recursos aparece praticamente em todos os problemas reais.
- ◆ Cooper; Kingston (1995 *apud* CHAND, 2002) referindo-se ao problema de programação de exames dizem que devido à grande quantidade de escolhas na alocação de provas e locais o problema é NP-Completo.
- ◆ Enzhe; Sung (2002) coloca que o problema de TT é NP-Completo e que por esse motivo a solução satisfatória deste tipo de problema é extremamente difícil.
- ◆ Even; Atai; Shamir (1976 *apud* SANTOS; OCHI; SOUZA, 2004) afirmam que o *school timetabling* problem é considerado NP-Hard para quase todas as formas que ele pode aparecer.
- ◆ CMU (2005) também coloca, referindo-se ao *class timetabling problem*, que uma procura exaustiva de todas as soluções possíveis é NP-Hard.
- ◆ Carrasco; Pato (2004) mencionam que em termos de complexidade, os autores Even et al. (1976 *apud* CARRASCO; PATO, 2004) demonstram que o problema de *class-teacher timetabling* é NP-Hard mesmo para versões simplificadas do problema.

Reforçando a idéia supracitada, Carter (1989) apresenta um exemplo de problema considerado NP-Hard. Este problema é composto de uma função objetivo e duas equações representando as restrições:

$$\min \sum_{i=1}^n \sum_{j=1}^m C_{ij} x X_{ij} \quad (\text{função objetivo})$$

sujeito a :

$$\sum_{j=1}^m X_{ij} = 1 \quad \text{para } i = 1, 2, \dots, n$$

$$\sum_{i \in P_k} X_{ij} \leq 1 \quad \text{para } j = 1, 2, \dots, m \quad \text{para todo os períodos } k$$

Figura 4: Exemplo de formulação de um problema de TT (Carter, 1989)

Esta formulação é classificada como programação binária, pois a variável de decisão X_{ij} pode assumir os valores 1 (caso a aula “i” seja alocada à sala “j”) ou 0 (caso contrário). Com relação às restrições¹⁶, a primeira assegura que toda aula é alocada a uma sala e a segunda garante que nenhuma sala de aula terá duas aulas alocadas para o mesmo período. Apesar deste problema parecer simples, Carter; Tovey (1989 *apud* CARTER, 1989) classificam-no como NP-*Hard* mesmo quando somente dois períodos são considerados ($k=2$).

Observar-se-á, no capítulo sobre abordagens de solução, que a classificação do problema de TT na classe NP-*Hard* terá grande influência nos métodos de solução para este tipo de problema. Outra característica que também exerce tal influência e que cabe ressaltar neste item, pois é bastante ressaltada na literatura, é a especificidade dos problemas de TT.

Conforme apontado por Santos; Ochi; Souza (2004), os problemas de TT envolvendo instituições educacionais são bastante numerosos. Isto se deve ao fato de que são altamente dependente de fatores como o sistema educacional de um país ou ao tipo de instituição tratada. Merlot et al. (2002) diz que as diferenças também podem ser explicadas por meio da política e combinação única de restrições adotadas pela instituição. Outro fator apontado pelo autor é que cada instituição terá um julgamento diferente do que constitui um TT de boa qualidade. Reis (2000 *apud* CHAND, 2002) e Burke et al (1997 *apud* CHAND, 2002) colocam que uma das dificuldades de se pesquisar problemas de TT é a variedade de forma que as restrições e o formato dos dados são encontrados dependendo da instituição analisada.

A especificidade do problema de TT resulta na constatação feita por Al-Yakoob; Sherali (2005): apesar dos problemas de TT e de *class scheduling* estejam sendo pesquisados amplamente na literatura e, conseqüentemente, diversos métodos de solução existam, nenhum modelo é universalmente aceito para a resolução deste tipo de problema. O autor aponta como a causa principal a natureza específica do problema de TT. Sob este ponto de vista, De Werra (1997) diz que os problemas de

¹⁶ Nota-se que na segunda restrição P_k significa o conjunto dos cursos alocados para o mesmo período k .

TT são numerosos, pois diferem no tipo e densidade das restrições. No entanto, paradoxalmente, o mesmo autor enfatiza que mesmo que haja dois problemas do mesmo “tamanho” (complexidade) com os mesmos tipos de restrição, eles serão bem diferentes um do outro. O motivo apontado é que apesar das restrições serem da mesma natureza, a maneira como elas são consideradas podem exercer grande influência no método de solução. Esta afirmação fica mais clara nas palavras de Manfrin; Sampels; Socha (2005) quando diz que a especificidade do problema de TT reside principalmente no tipo e número de restrições do tipo *soft* e *hard* que comumente variam bastante entre um problema e outro ou entre uma instituição e outra. Estes dois tipos de restrição são bastante importantes na formulação do problema de TT e por isso serão discutidos de maneira isolada no próximo item. Entretanto, as restrições do tipo *hard* e *soft* configuram em um problema de TT o que deve ser realmente obedecido e o que deve ser, na medida do possível, respeitado. Exatamente esta escolha do que é compulsório ou não que torna os problemas tão específicos como já apontado.

3.4 RESTRIÇÕES HARD E SOFT – SOLUÇÃO VIÁVEL E ACEITÁVEL

Conforme colocado no tópico anterior, o problema de TT é bastante complexo e por isso classificado como NP-Hard. Essa complexidade aliada ao fato de que TT lida com o horário das pessoas faz com que a formulação do problema conte com diversas restrições ou desejos. Pode-se imaginar por exemplo que alunos e professores não queiram ter ou dar aula em uma sexta-feira à noite ou que o professor prefira concentrar suas aulas em poucos dias durante a semana ou, ao contrário, que suas aulas sejam espaçadas na semana preferencialmente na parte da manhã. Também podem haver casos em que os alunos não querem ter um espaço de tempo muito grande entre duas aulas consecutivas. A lista de preferências ou restrições pode ser tão extensa quanto se queira. Neste momento, então, que a diferença entre restrição e desejo torna-se fundamental para a resolução de um problema de TT.

Imagine que todas as preferências dos professores e alunos fossem encaradas como restrições a serem respeitadas impreterivelmente¹⁷. Neste caso, como grande

¹⁷ Apesar desta frase parecer redundante, ela foi colocada desta maneira para enfatizar a criticidade da situação comentada.

parte das restrições seria conflitante, achar uma solução viável seria extremamente difícil ou raramente encontrada como colocado por Stallaert (1997)¹⁸. Zhao; Goebel (1999) colocam que a maioria dos problemas reais de TT são super-restritos (*over-constrained*). Neste caso, conforme colocado por Frangouli; Harmandas (1995), torna-se necessário relaxar algumas dessas restrições, ou seja, fazer com que não sejam compulsórias, mas uma preferência ou desejo, simplesmente. O autor diz que esta artimanha é adequada à realidade, pois em grande parte das vezes, as entidades – alunos e professores – envolvidas no problema expressam suas preferências e não obrigações a serem cumpridas. Stallaert (1997) compartilha desta idéia mencionando que algumas restrições são claramente classificadas como “*nice to have*”, ou seja, desejáveis; enquanto outras precisam ser obrigatoriamente respeitadas.

Seguindo o raciocínio exposto acima, Kolski; Piechowiak (2004) colocam que as restrições podem ser classificadas em dois grupos: restrições físicas ou restrições de preferência. Na literatura, estas duas restrições são mais conhecidas como *hard constraints* (restrições duras) e *soft constraints* (restrições *soft* ou frágeis), respectivamente. Glibovets; Medvid (2003) denominam, por sua vez, estas restrições de rígidas e não-rígidas e diz que quando as restrições rígidas são violadas o TT resultante é incorreto e impossível de ser aplicado na prática. O autor conclui, portanto, que o número de restrições rígidas deve ser o menor possível. Kolski; Piechowiak (2004) colocam que a violação de restrições do tipo *hard* provoca o que chama de confrontos como, por exemplo, o fato de um aluno ter duas aulas em duas salas diferentes concomitantemente. O interessante a notar neste ponto é que as restrições do tipo *hard* estão intimamente ligadas com a solução dita viável, ou seja, dado que foram respeitadas, o TT resultante é válido – passível de ser aplicado na prática. Entretanto, apesar de sua aplicabilidade, o TT gerado pode apresentar uma qualidade ruim, ou seja, desrespeita grande parte das preferências dos professores como também não oferece uma qualidade pedagógica adequada (sobrecarrega os alunos, por exemplo). A restrição do tipo *soft*, vista no próximo parágrafo, é a que vai determinar a aceitabilidade ou qualidade de um TT. Portanto, em relação a um mesmo problema, quanto mais uma alternativa de solução considerar as restrições do tipo *soft*, melhor ela será.

¹⁸ Problemas deste tipo são chamados de super-restritos (*over-constraint*).

Alvarez-Valdes; Crespo; Tamarit (2002) definem as restrições do tipo *soft* como sendo aquelas que quanto mais forem respeitadas, melhor será o TT. Zhao; Goebel (1999) colocam que essas restrições devem ser satisfeitas, no entanto, em alguns casos são parcialmente atendidas ou até mesmo violadas, sendo que ambas as situações são aceitáveis. Frangouli; Harmandas (1995) dão um exemplo citando o caso de um professor que não gostaria de dar aula antes das 11h todos os dias. Encarando este pedido como uma restrição do tipo *soft*, caso um TT melhor¹⁹ seja encontrado com aulas para este professor antes das 11h da manhã então esta preferência será violada.

Ainda em relação às restrições *soft*, Kolski; Piechowiak (2004) ressaltam uma característica relevante dizendo que como são normalmente expressas por palavras como “evitar”, “deveria”, “se possível” etc. esse tipo de restrição torna-se mais difícil de formalizar do que as restrições do tipo *hard*. O autor coloca que as restrições não-rígidas são extremamente importantes na determinação da qualidade do TT; entretanto são numerosas e de natureza mais empírica o que, conseqüentemente, dificulta a sua determinação e transcrição para a formulação matemática. Outra dificuldade apontada, agora por Stallaert (1997), diz respeito à importância de cada restrição *soft*. Esta preocupação aparece na medida em que esse tipo de restrição pode receber prioridades, ou seja, a preferência demandada pelos professores pode ser mais importante do que a preferência dos alunos²⁰. Esta problemática revela uma constatação colocada por Carter (2001 *apud* DIMOPOULO ET AL., 2004) quando diz que o *timetabling* é em grande parte político. Essa política revela-se na determinação da organização do que é um TT de boa qualidade, pois, colocando em outras palavras, qualidade refere-se ao atendimento de restrições do tipo *soft*, conseqüentemente, ao atendimento das preferências que foram escolhidas como as mais prioritárias. Nos parágrafos subseqüentes, mostrar-se-á como as restrições *soft* aparecem na formulação matemática.

O importante a se destacar neste momento é que as restrições *soft* como foram chamadas não são, a rigor, restrições. Dakalaki; Birbas (2005) tomam o cuidado, em

¹⁹ O termo melhor é bastante subjetivo, mas entende-se aqui como um *timetabling* que melhora a situação da maioria das entidades envolvidas – professores e alunos.

²⁰ O autor deste trabalho reserva-se o direito de deixar um protesto nesta nota.

seu artigo, de não utilizar o termo “restrições *soft*”, mas sim regras *soft*. O autor menciona que estas regras não são cobertas pelo modelo de restrições, mas servem para guiar o TT na direção de soluções mais atrativas. Essa função é alcançada por meio da utilização de pesos na função-objetivo da formulação. Alguns autores como Abramson (1991) chamam a função-objetivo de função custo e que, portanto, deve ser minimizada. Sendo uma parcela da função objetivo com um peso (custo) associado, pode-se imaginar que as restrições *soft* tenham importância (pesos) diferentes. Elmohamed; Coddington; Fox (1997) coloca que as restrições de um problema podem ser classificadas em três grupos: *hard* (difícil), *médium* (média) e *soft* (fraca):

- Restrições *hard*: restrições que devem ser respeitadas para que a solução seja classificada como viável (útil). Normalmente, essas são consideradas restrições do problema e, portanto, não aparecem na função objetivo (não permitir que duas aulas sejam marcadas no mesmo horário para a mesma sala de aula). Entretanto, será mostrado mais tarde que em alguns métodos de solução isso nem sempre é verdadeiro.

- Restrições *médium*: são as restrições/regras que se situam na área cinzenta entre as *hard* e *médium*. Este tipo de restrição possui um peso ou penalidade bem alta. Espera-se que a solução consiga respeitar todas as restrições médias (evitar colocar aulas no mesmo horário que possuam alunos em comum).

- Restrições *soft*: possuem uma penalidade associada menor que a média e não se espera que a solução obtida respeite-as integralmente (ex: preferência dos professores).

Um exemplo de função-objetivo onde os pesos são aplicados é fornecido abaixo:

$$F(Q) = \sum_{i=1}^I \alpha_i x b_i + \beta_i x c_i + \omega_i * d_i \text{ onde "t" é a quantidade de professores}$$

Nesta função, por exemplo, as letras gregas representam o peso correspondente a cada tipo de restrição/regra *soft*. Traduzindo, b_i pode ser a quantidade de aulas que o professor “i” terá que dar sexta-feira à noite, enquanto c_i poderia ser a quantidade de aulas seguidas que o professor “i” dá em um mesmo dia. Nota-se, portanto, que é

possível estabelecer importâncias relativas entre as restrições do tipo *soft* e que cada professor pode ter a sua prioridade específica já que neste caso o peso depende de qual professor está sendo considerado. Sobre este assunto, Stallaert (1997) aponta que muitas vezes é difícil estabelecer essa importância relativa, pois nem sempre ela é bem evidenciada pelas entidades – professores e alunos – envolvidas no problema. O autor diz que um processo de tentativa e erro pode ser necessário para obter um bom conjunto de parâmetros/penalizações. Finalmente, o valor dessa função objetivo, dependendo da alternativa de solução considerada, será predominante para dizer se o TT é bom ou ruim em relação à outro.

Visando deixar o leitor mais familiarizado com os tipos de restrições utilizadas em um problema de TT, a tabela subsequente enumera as restrições *hard* e *soft* mais comuns:

	Restrições <i>Hard</i>	REF.	Restrições <i>Soft</i>	REF.
<i>I</i>	Colisões não são permitidas (professores darem aulas ao mesmo tempo, uma sala ter duas ou mais aulas ao mesmo tempo, alunos terem duas ou mais aulas ao mesmo tempo etc.)	(DASKALAKI; BIRBAS; HOUSOS, 2004) (ELMOHAMED; CODDINGTON; FOX, 1997) (MANFRIN; SAMPELS; SOCHA, 2005) (DEMAZEAU ET AL., 2005) (BUFE ET AL., 2001) (ENZHE; SUNG, 2002) (CARRASCO; PATO, 2004) (ALVAREZ-VALDES; CRESPO; TAMARIT, 2002) (FRANGOULI; HARMANDAS, 1995) (CHENG; KRUK; LIPMAN, 2002)	Preferência dos professores seja obedecida, se possível	(DASKALAKI; BIRBAS; HOUSOS, 2004) (ELMOHAMED; CODDINGTON; FOX, 1997) (DEMAZEAU ET AL., 2005) (ABDENNADHER; RINGEISSEN, 2000) (ENZHE; SUNG, 2002) (CARRASCO; PATO, 2004) (ALVAREZ-VALDES; CRESPO; TAMARIT, 2002) (FRANGOULI; HARMANDAS, 1995)

2	O <i>timetable</i> deve estar completo (todos os cursos aparecem no horário, todos os alunos e professores foram alocados em aulas)	(DASKALAKI; BIRBAS; HOUSOS, 2004) (ENZHE; SUNG, 2002) (ALVAREZ-VALDES; CRESPO; TAMARIT, 2002)	Horário dos estudantes deve ser o mais compacto possível	(DASKALAKI; BIRBAS; HOUSOS, 2004) (ELMOHAMED; CODDINGTON; FOX, 1997) (ABDENNADHER; RINGEISSEN, 2000) (ALVAREZ-VALDES; CRESPO; TAMARIT, 2002)
3	A capacidade da sala de aula não deve ser excedida.	(MULLER; BARTAK, 2002) (ELMOHAMED; CODDINGTON; FOX, 1997) (MANFRIN; SAMPELS; SOCHA, 2005) (ENZHE; SUNG, 2002) (CARRASCO; PATO, 2004) (FRANGOULI; HARMANDAS, 1995)	Minimizar a mudança freqüente de sala de aula por parte dos estudantes	(DASKALAKI; BIRBAS; HOUSOS, 2004) (ELMOHAMED; CODDINGTON; FOX, 1997) (CARRASCO; PATO, 2004) (ALVAREZ-VALDES; CRESPO; TAMARIT, 2002)
4	Deve existir um intervalo de 1h entre aulas que são ministradas para a mesma turma só que ocorre em prédios diferentes.	(MULLER; BARTAK, 2002)	Uma turma não deveria ter mais de 10h de aula em um dia como também ter mais de 6 horas sem um intervalo.	(MULLER; BARTAK, 2002)
5	As aulas devem ser alocadas nas salas apropriadas (equipamentos)	(ELMOHAMED; CODDINGTON; FOX, 1997) (BUFE ET AL., 2001) (ENZHE; SUNG, 2002) (CARRASCO; PATO, 2004)	Um professor não deve dar aulas por mais de 8h por dia como também mais de 6h sem intervalo.	(MULLER; BARTAK, 2002)

		(ALVAREZ- VALDES; CRESPO; TAMARIT, 2002)		
6	Cursos que são pré-requisitos de outros não devem ocorrer no mesmo período.	(DEMAZEAU ET AL., 2005)	As aulas devem ocorrer preferencialmente no mesmo local que o professor trabalha.	(MULLER; BARTAK, 2002) (ENZHE; SUNG, 2002) (CARRASCO; PATO, 2004)
7	O número de cursos que ocorrem em um determinado tempo deve ser limitado ao número de salas disponíveis.	(DEMAZEAU ET AL., 2005)	Minimizar o espaço de tempo livre entre a primeira e a última aula do dia para todos os professores e alunos.	(MULLER; BARTAK, 2002) (ENZHE; SUNG, 2002) (CARRASCO; PATO, 2004)
8	Na construção do TT, aulas que já tinham períodos predefinidos para ocorrerem devem ser respeitadas.	(ENZHE; SUNG, 2002) (ALVAREZ- VALDES; CRESPO; TAMARIT, 2002)	Os alunos, preferencialmente, devem ter um horário para almoço.	(ELMOHAMED; CODDINGTON; FOX, 1997)
9	As matérias devem estar bem distribuídas.	(ENZHE; SUNG, 2002) (CARRASCO; PATO, 2004)	Preferência dos professores por salas de aula precisa ser respeitada.	(ELMOHAMED; CODDINGTON; FOX, 1997)
10	Os alunos e professores devem ter um período livre para almoçar.	(CARRASCO; PATO, 2004)	As aulas devem ocorrer preferencialmente no mesmo departamento em que o aluno estuda.	(ELMOHAMED; CODDINGTON; FOX, 1997) (CARRASCO; PATO, 2004)
11	Os períodos em que os professores e alunos não estão	(CARRASCO; PATO, 2004) (ALVAREZ- VALDES; CRESPO;	Número de aulas no último período do dia deve ser minimizado,	(MANFRIN; SAMPELS; SOCHA, 2005)

	disponíveis devem ser respeitados.	TAMARIT, 2002) (FRANGOULI; HARMANDAS, 1995)	para cada aluno.	
12			Um aluno não deveria ter somente uma aula no dia.	(MANFRIN; SAMPELS; SOCHA, 2005)
13			Alunos, preferencialmente, não devem ter duas aulas no mesmo período.	(MANFRIN; SAMPELS; SOCHA, 2005)
14			Cursos opcionais de cada aluno não devem cair no mesmo período que aulas compulsórias.	(DEMAZEAU ET AL., 2005)
15			Aulas devem ser concentradas preferencialmente de manhã.	(BUFE ET AL., 2001)
16			Professores devem ter um mínimo de dias livres por semana.	(BUFE ET AL., 2001)
17			Para cada período de aula é desejável ter um professor substituto disponível.	(BUFE ET AL., 2001)
18			Aulas devem ocorrer em salas que possuem a capacidade adequada.	(ALVAREZ-VALDES; CRESPO; TAMARIT, 2002)

Tabela 6: Exemplo de restrições do tipo *hard* e *soft* , elaborada pelo autor

A lista apresentada não é para ser uma pesquisa exaustiva das restrições do tipo *soft* e *hard*. Entretanto, mesmo com essa amostra pequena, percebe-se alguns fatos relevantes como:

- ◆ Quais são as restrições do tipo *hard* (1, 3 e 5) e *soft* (1,2 e 3) mais comuns.
- ◆ A diversidade do problema de TT. Percebe-se que há restrições que um autor coloca como *hard* enquanto outro coloca como *soft*.
- ◆ As restrições do tipo *soft* têm uma variedade muito maior do que as *hard* e, normalmente, aparecem em quantidade superior nos problemas de TT.

Buscou-se com este tópico familiarizar ainda mais o leitor no problema de TT. No próximo item, considerações sobre a automatização do TT são feitas.

3.5 RESOLUÇÃO MANUAL X AUTOMATIZAÇÃO

Muitas instituições educacionais enfrentam o problema de programar cursos e exames todo semestre ou ano. Segundo Hussin; Kendal (2005), com o aumento da complexidade dos problemas – grande número de estudantes, cursos, exames, salas e restrições – a automatização do TT para produzir respostas de boa qualidade é necessária. O autor diz que o problema de TT é resolvido nas instituições ou de maneira manual, ou semi-automática ou completamente automática. Entretanto, o autor apresenta uma pesquisa conduzida por Burke et al (1996) em 56 universidades britânicas. Esta pesquisa revelou que somente 58% dos respondentes utilizam computador em alguma fase do processo. Também foi constatado que uma parcela pequena – 21% (11 universidade) - possuía um sistema propriamente dito sendo que duas utilizavam *software* comercial e as outras haviam desenvolvido seu próprio sistema. Essa disparidade pode ser explicada pela especificidade do problema de TT que faz com que as universidades construam seu próprio sistema. A pesquisa de Burket et al (1996) mostra que para as empresas adotarem um *software* ele deve produzir TTs de boa qualidade permitindo intervenção do usuário, fácil de usar, abrangente e compatível com os outros sistemas usados na instituição.

A automatização do TT encontra sucesso mesmo no início do seu desenvolvimento. Bloomfield; McSharry (1979 *apud* STALLAERT, 1997) estão entre os primeiros a terem resolvido o problema de alocação de cursos. Eles reportam

que o problema foi resolvido em menos de dois dias enquanto que o processo manual levava de duas a três semanas. De acordo com Kleiger (1992 *apud* STALLAERT, 1997), um TT construído por meio de um sistema produz uma solução no mínimo tão boa quanto uma manual, mas com um esforço muito menor. Stallaert (1997) ressalta que a automatização apresenta duas vantagens: economiza tempo dos planejadores e melhora tanto o processo de coleta de dados como a qualidade da solução. Carter (1986) diz que a complexidade dos problemas está levando os processos manuais a ficarem intratáveis. Para o autor, a maioria dos grandes problemas tem sido automatizada em algum nível.

Muitos casos de sucesso da automatização do processo de alocação aparecem na literatura. Nos itens abaixo são apresentados três deles:

◆ Stallaert (1997) diz ter vivenciado essas vantagens ao automatizar o TT de uma faculdade de Los Angeles. O autor diz que, no processo anterior, o aparente simples trabalho de achar os cursos conflitantes era penoso sendo que, mais difícil ainda, era realocar os cursos em períodos válidos. Após a automatização, o *software* gera uma lista dos cursos conflitantes, isso quando existe algum. O autor também diz que como o sistema demanda informações precisas, o processo de coleta de dados na universidade ficou mais organizado após a automatização.

◆ Demazeau et al. (2005), por sua vez, conta que no seu caso, automatização do TT de uma faculdade de engenharia francesa, o processo manual levava em torno de uma semana para gerar uma resposta. Após a automatização, 10 respostas diferentes para o problema foram geradas em aproximadamente 10 minutos sendo que a qualidade delas é superior em comparação com a manual. O autor menciona que a capacidade de simular TT em diferentes condições também foi um dos resultados de grande interesse para o planejador da faculdade.

◆ Carter (1989) menciona o sucesso da automatização do TT na Universidade de Waterloo (Canadá). O autor diz que o tempo e esforço para gerar a solução foram diminuídos substancialmente dado que a resolução manual do problema consumia cerca de 27 semanas no ano. Carter (1989) diz que o maior sucesso do

software foi que nenhum aluno notou a troca de processos mostrando que o TT gerado era no mínimo melhor que o manual. Destaca-se que, segundo o autor, o sistema manual de Waterloo era considerado, em média, melhor do que os das universidades norte-americanas.

Demazeau et al. (2005) menciona que os problemas de TT são bastante estudados devido a sua gama de aplicações e se tornam críticos em organizações de grande porte. Paradoxalmente, muitas organizações, atualmente, ainda resolvem os problemas de alocação de forma manual devido à falta de uma ferramenta apropriada. Mas também é fato, como apontado por Hussin; Kendal (2005), que o processo de automatização tem se intensificado ao longo dos anos e que há uma pesquisa continua sobre novas técnicas e métodos para resolver os problemas de TT e assim produzir melhores soluções.

Conclui-se este tópico com a constatação de que automatização ou semi-automatização²¹ é uma tendência irrefutável para a solução de problemas de TT, principalmente em grandes instituições (de ensino ou não), dado o avanço da informática e das abordagens/métodos de solução. No próximo capítulo, diversos destes métodos serão apresentados.

3.6 MÉTODOS DE SOLUÇÃO PARA O PROBLEMA DE TIMETABLING

Segundo Cole (1994 *apud* ENZHE; SUNG, 2002) e Wood (1968 *apud* ENZHE; SUNG, 2002), o problema de TT vem sendo investigado por um grande número de pesquisadores principalmente devido à programação de recursos em instituições educacionais. Como resultado, conforme apontam Carter; Laporte (1995 *apud* ENZHE; SUNG, 2002) e Cumming (1995 *apud* ENZHE; SUNG, 2002) diversas técnicas foram e estão sendo desenvolvidas para resolver este tipo de problema. Segundo Enzhe; Sung (2002), a técnica mais antiga e popular é o chamado *graph colouring algorithm*. Entretanto, Alvarez-Valdes; Crespo; Tamarit (2002) apontam que os métodos de solução são extremamente amplos partindo desde o método mencionado até complexos algoritmos de metaheurística passando por programação

²¹ Demazeau et al. (2005) é defensor da idéia de que as ferramentas computacionais para solução do problema de *timetabling* precisam ser abertas e interativas de modo a aproveitar-se da inteligência (*Knowhow*) do planejador.

matemática²² e heurísticas específicas para o problema em questão. Daskalaki; Birbas; Housos (2004) descrevem, no entanto, que no início da década de 60, observava-se na literatura sobre a resolução de problemas de TT que os métodos mais comuns eram simulação, heurística e métodos de *graph colouring*.

Conforme foi colocado, os problemas de TT já apresentam uma dificuldade inata (conforme colocado por Daskalaki; Birbas; Housos (2004), os problemas de TT são NP-Completo em quase todas as suas formas). Entretanto, com o passar do tempo, os problemas de TT, que já apresentavam essa dificuldade, tornaram-se ainda mais complexos principalmente pelo aumento do número de recursos (pessoas) envolvidos, ou seja, pelo crescimento das instituições onde há problemas de TT a serem resolvidos.

O resultado do cenário traçado no parágrafo acima foi expresso nas palavras de Alvarez-Valdes; Crespo; Tamarit (2002) ao dizer que os métodos mais eficientes para resolver este tipo de problema foram os desenvolvidos recentemente, sendo que a principal característica deles é a utilização de metaheurísticas. Este pensamento é compartilhado por Bellanti et al. (2004) ao dizer que as abordagens atuais são baseadas principalmente em metaheurísticas²³. Já Abramson; Dang; Krisnamoorthy (1999) enfatizam que, devido ao tamanho dos problemas reais de TT encontrados já no final da década de 90, as soluções mais eficientes eram as que se baseavam em heurísticas. O leitor pode estar se perguntando se estes métodos baseados em heurísticas ou metaheurísticas não apresentam nenhuma desvantagem em relação aos anteriores. Sob um ponto de vista limitado, o principal ponto negativo das heurísticas ou metaheurísticas é que, conforme será exposto, esses métodos não resultam necessariamente na solução ótima do problema como seria o caso da programação matemática. Entretanto, conforme Talbi (2002) constata, a obtenção de uma resposta ótima é intratável em alguns tipos de problemas combinatórios como é o caso daqueles classificados como *NP-Hard*, justamente o caso dos problemas de TT. O autor diz que para este tipo de problema, na prática, uma “boa” solução obtida via metaheurística já é satisfatório.

²² Programação Linear, Programação Mista, Programação Inteira, Programação Não-Linear são exemplos de programação matemática.

Ressalta-se, porém, que, conforme colocado por Mausser; Magazine (1996) e Daskalaki; Birbas; Housos (2004), ainda não existe um método considerado unanimemente como o melhor para resolver problemas de TT. A melhor explicação para este fato é a já mencionada característica de especificidade que os problemas de TT apresentam. Todavia, conforme será exposto, novas abordagens também baseadas em metaheurísticas, as hiperheurísticas, surgiram justamente para lidar com esta deficiência. A seguir é apresentada uma lista contendo as principais abordagens para a resolução de um problema típico de TT seguido de uma ou mais referências²⁴.

²³ O autor também menciona a programação por restrição como sendo também uma base importante para as abordagens atuais.

²⁴ Enfatiza-se que esta lista não apresenta a pretensão de ser uma pesquisa exaustiva dos métodos presentes na literatura.

Métodos de Solução para o Problema de Timetabling	
<i>Local Search</i>	Ppadimitriou; Steiglitz (1982 apud TALBI, 2002); (ARNTZEN, H.; LOKKETANGEN, 2005)
<i>Greedy Heuristic</i>	Lawler (1976 apud TALBI, 2002)
<i>Simulted Annealing</i>	Kirkpatrick; Gelatt; Vecchi (1983 apud TALBI, 2002); Abramson (1991 apud ABRAMSON; DANG; KRISNAMOORTHY, 1999); Dowsland (1990 apud ALVAREZ-VALDES; CRESPO; TAMARIT, 2002); Elmohamed et al. (1997 apud ALVAREZ-VALDES; CRESPO; TAMARIT, 2002); (ABRAMSON; DANG; KRISNAMOORTHY, 1999); (ABRAMSON, 1991)
<i>Tabu Search</i>	Glover (1989 apud TALBI, 2002); Hertz (1991 apud ALVAREZ-VALDES; CRESPO; TAMARIT, 2002); Dowsland (1998 apud BELLANTI ET AL., 2004); Aickelin (2000 apud BELLANTI ET AL., 2004); (ALVAREZ-VALDES; CRESPO; TAMARIT, 2002); (SANTOS; OCHI; SOUZA, 2004)
<i>Genetic Algorithm</i>	Holland (1975 apud TALBI, 2002); Burke et al. (1996 apud ALVAREZ-VALDES; CRESPO; TAMARIT, 2002); Corne; Ross et al. (1996 apud ALVAREZ-VALDES; CRESPO; TAMARIT, 2002); Paetcher et al. (1996 apud ALVAREZ-VALDES; CRESPO; TAMARIT, 2002); Colomi (1990 apud FRANGOULI; HARMANDAS, 1995); (ENZHE; SUNG, 2002); (GLIBOVETS; MEDVID, 2003); (CMU, 2005); (BADICA; LALESCU, 2005)
<i>Evolution Strategies</i>	Rechenberg (1973 apud TALBI, 2002)
<i>Genetic Programming</i>	Koza (1992 apud TALBI, 2002)
<i>Ant Colonies</i>	Colomi; Dorigo; Maniezzo (1991 apud TALBI, 2002); (MANFRIN; SAMPELS; SOCHA, 2005)
<i>Scatter Search</i>	Glover (1977 apud TALBI, 2002)
<i>Graph Colouring</i>	Carter (1986 apud MAUSSER; MAGAZINE, 1996); Mehta (1981 apud FRANGOULI; HARMANDAS, 1995); Welsh; Powell (1967 apud DASKALAKI; BIRBAS; HOUSOS, 2004); Brelaz (1979 apud CARTER, 1986); Dunstan (1976 apud CARTER, 1986)
<i>Neural Network</i>	Pellerin; Héroult (1994 apud MAUSSER; MAGAZINE, 1996); (CARRASCO; PATO, 2004)
<i>Hyperheuristics</i>	(BURKE; KENDALL; SOUBEIGA, 2005); (HUSSIN; KENDALL, 2005)
<i>Constraint Logic Programming</i>	Guéret (1995 apud ALVAREZ-VALDES; CRESPO; TAMARIT, 2002); Azevedo (1994 apud FRANGOULI; HARMANDAS, 1995); Boizumault (1994 apud FRANGOULI; HARMANDAS, 1995); (FRANGOULI; HARMANDAS, 1995); (HENZ; WURTZ, 1996); (MÜLLER; RUDOVÁ, 2004); (ABDENNADHER; RINGEISSEN, 2000); Henz; Wurtz (1995 apud STALLAERT, 1997)
<i>Network model</i>	Mulvey (1982 apud FERLAND; FLEURENT, 1994); Chahal; Werra (1989 apud STALLAERT, 1997)
<i>Lagrangian Relaxation</i>	Tripathy (1980 apud FERLAND; FLEURENT, 1994); (CARTER, 1989)
<i>Programação Matemática</i>	Tripathy (1980 apud FRANGOULI; HARMANDAS, 1995); Ferland (1985 apud FRANGOULI; HARMANDAS, 1995); Lawrie (1969 apud DASKALAKI; BIRBAS; HOUSOS, 2004); Akkoyunlu (1973 apud DASKALAKI; BIRBAS; HOUSOS, 2004); (DASKALAKI; BIRBAS; HOUSOS, 2004); (DAKALAKI; BIRBAS, 2005); (TRIPATHY, 1984)
<i>Database Management System</i>	Johnson (1993 apud FRANGOULI; HARMANDAS, 1995)
<i>Simulation Techniques</i>	Schmidt; Strohlein (1979 apud DASKALAKI; BIRBAS; HOUSOS, 2004)
<i>Decision Support System</i>	(FERLAND; FLEURENT, 1994); (KOLSKI; PIECHOWIAK, 2004); (MULLER; BARTAK, 2002); (DIMOPOULO ET AL., 2004); Johnson (1993 apud STALLAERT, 1997); (DINKEL; MOTE; VENKATARAMANAN, 1989)
<i>Flow Formulation</i>	(CHENG; KRUK; LIPMAN, 2002); (DE WERRA, 1969)
<i>Hybrid approach</i>	(BUFE ET AL., 2001); (MERLOT ET AL, 2002); (DEMAZEAU ET AL., 2005); (STALLAERT, 1997)

Tabela 7: Abordagens para a solução do problema de *timetabling*, elaborada pelo autor

A última linha, referente à abordagem híbrida (*Hybrid Approach*), contém as abordagens que se utilizam de dois ou mais métodos conjuntamente²⁵. Conforme já

²⁵ Talbi (2002) apresenta uma proposta de taxonomia das principais metaheurísticas híbridas.

mencionado, esta lista não pretende ser uma relação exaustiva das técnicas de solução do problema de TT. Todavia, o autor deste trabalho acredita que ela engloba os métodos mais populares e cumpre com a tarefa de passar ao leitor a idéia de que o TT possui uma ampla variedade de métodos e que os mais utilizados são as (meta)heurísticas. O leitor, porém, terá um maior contato com algumas dessas abordagens no ANEXO A, onde será dado maior detalhe sobre a lógica, funcionamento e características de alguns destes métodos listados²⁶.

3.7 ESTADO DA ARTE

Após todas essas considerações sobre o problema de TT, resta fazer a seguinte pergunta: “Em qual direção as futuras pesquisas seguirão?”. Essa pergunta está intrinsecamente ligada com uma das características do TT que já foi citada neste trabalho: a especificidade.

Sobre este assunto, Carter e Laporte (1996 *apud* BURKE; PETROVIC, 2002) apontam que deve haver centenas de pesquisas e milhares de *softwares* (muitos amadores) sobre TT. Também há uma alta variedade de abordagens para resolver o famigerado problema e muitos deles ajudaram a entender a dificuldade e natureza dos problemas reais. Entretanto, os autores dizem que a maioria das soluções refere-se a problemas extremamente específicos. Carter e Laporte dizem que muitos pesquisadores acreditam que o próximo passo estará ligado a desenvolver métodos que inteligentemente escolham a melhor maneira de solucionar um determinado problema ao invés de se preocupar com um algoritmo específico para um problema específico. Portanto, os métodos de solução da nova geração deverão trabalhar em um nível de abstração maior do que as abordagens atuais. Seguindo este raciocínio, Burke; Petrovic (2002) apresentam duas tendências conforme será exposto nos itens subseqüentes.

3.7.1 ABORDAGEM HISTÓRICA

Conforme Burke et al. (2002) apontam, a idéia por trás desta abordagem é a utilização de experiência passada para resolver novos problemas. Burke; Petrovic (2002) colocam que a idéia é montar um banco de dados com diversos casos

²⁶ Para os métodos que não forem detalhados neste trabalho, recomenda-se que o leitor busque mais informações nas referências fornecidas na tabela sobre abordagens.

solucionados onde cada registro apresentaria o problema de TT junto com o algoritmo que foi utilizado para resolvê-lo e o resultado obtido. Burke et al. (2002) coloca que esta idéia não é nova, sendo já utilizada como sucesso em diversas áreas nos últimos 15 anos, entretanto pouca pesquisa foi feita com o problema de TT. Portanto, ao encontrar um novo problema, o banco de dados seria pesquisado e os TTs semelhantes²⁷ seriam selecionados. O método de solução mais adequado seria selecionado puramente ou como alguma adaptação.

Claramente este método de solução possui algumas dificuldades como organização do banco de dados para que a procura seja eficiente e rápida; como adaptar o algoritmo escolhido para o novo problema; como aprender com as falhas e sucessos obtidos e, finalmente, como controlar o crescimento do banco de dados.

3.7.2 **HIPERHEURÍSTICAS**

As hiperheurísticas são fortemente cotadas para serem a tendência de solução dos problemas de TT. Segundo Burke; Kendall; Soubeiga (2005), hiperheurísticas podem ser definidas como sendo uma (meta)heurística que escolhe (meta)heurísticas de maneira a resolver um problema de otimização. A grande diferença entre a hiperheurística e a metaheurística é que a hiperheurística lida com métodos de solução (as heurísticas ou metaheurísticas, por exemplo) enquanto que as metaheurísticas lidam com solução. Destaca-se, porém, que a hiperheurística pode ser na verdade uma metaheurística como normalmente acontece. O termo hiperheurística é usado para mostrar que se trata de um nível maior de abstração.

Ressalta-se que o termo hiperheurística foi aplicado recentemente, mas que a idéia não é nova. Burke; Kendall; Soubeiga (2005) dizem que os primeiros trabalhos sobre hiperheurísticas foram divulgados em meados dos anos 60 por Fisher e Thompson (1961, 1963 *apud* BURKE; KENDALL; SOUBEIGA, 2005) e Crowston et al (1963 *apud* BURKE; KENDALL; SOUBEIGA, 2005) e tratavam de problemas de *job-shop scheduling* (problema de sequenciamento da produção). Entretanto, há pouca pesquisa sobre hiperheurística em problema de TT. A idéia por trás desta tendência é extremamente simples conforme é explicado no próximo parágrafo.

²⁷ Nota-se que “semelhança” é um termo extremamente subjetivo e constitui-se em um dos “problemas” desta abordagem.

A principal vantagem da hiperheurística é que ela não precisa se preocupar com o tipo de problema que está resolvendo. Inicialmente a informação que ela precisa é a quantidade de (meta)heurísticas que estão no nível inferior. A hiperheurística não precisa nem saber o nome, propósito ou implementação detalhada das (meta)heurísticas inferiores. Ela só precisa saber qual (meta)heurística específica (H_i , por exemplo) chamar. Esta, por sua vez, irá modificar a atual solução do problema e retornará um resultado via uma função de avaliação. A hiperheurística deverá tomar a decisão de chamará outra (meta)heurística específica ou se continuará com a atual e assim por diante. A principal dificuldade desta abordagem será construir hiperheurísticas que sejam competitivas quando comparadas com as (meta)heurísticas ou outras abordagens específicas para o problema em questão. Com exemplo de aplicação cita-se Burke; Kendall; Soubeiga (2005) e Hussin; Kendal (2005) que mostram e comentam a utilização da hiperheurística com Busca Tabu.

3.7.3 PROGRAMAÇÃO MATEMÁTICA

Torna-se importante destacar que com o avanço da informática, não está descartada a utilização de programação linear, mista, binária etc. na resolução de problemas de TT. Atualmente, as utilizações destes métodos encontram como principal problema o tempo que levam para encontrar a solução ótima. Entretanto, o avanço dos microprocessadores poderá colocar estas abordagens em destaque para a solução de problemas de combinação mesmo sendo de grande porte como é o caso da maioria dos problemas de TT.

METODOLOGIA

4. METODOLOGIA

O método utilizado para o desenvolvimento do *software* foi dividido em 5 etapas, conforme explicado nos pontos subseqüentes.

1) Primeiro foi feita uma pesquisa bibliográfica sobre o assunto em questão, que foi realizada para o entendimento dos conceitos presentes no estudo do problema de TT, principalmente em universidades e escolas. Essa etapa justifica-se, pois é extremamente importante conhecer as idéias de outros autores como, por exemplo, entender que métodos utilizaram, quais as dificuldades encontradas na resolução do problema assim como quais recomendações são fornecidas pelos autores.

2) A segunda etapa consistiu-se na realização de uma pesquisa bibliográfica específica sobre autores que realmente desenvolveram *softwares* para o problema de TT em questão. Essa pesquisa foi importante para analisar aspectos como: linguagem de programação utilizada, tempo de resposta do *software*, formas de representação de entradas de informação, relatórios normalmente gerados e formas de representar a solução propriamente dita.

3) A terceira etapa refere-se à pesquisa com pessoas envolvidas em empresas que realizam treinamento de usuários finais assim como cursos de capacitação.

3.1) Questionário

O desenvolvimento do questionário foi realizado de acordo com as etapas descritas por Garber (2001):

3.1.1) Construção do questionário

Esta etapa refere-se à elaboração das perguntas em uma seqüência lógica e que esteja de acordo com o meio escolhido para realização da pesquisa – correio, internet, pessoalmente (terceiros ou o próprio idealizador da pesquisa). No questionário foram utilizados três tipos de perguntas conforme a classificação proposta por Garber (2001):

◆ Perguntas fechadas: perguntas que oferecem um número finito de possibilidade de respostas previamente conhecidas, onde uma delas será

necessariamente escolhida. Destaca-se que em algumas perguntas deste tipo houve a utilização da escala Likert para as alternativas de resposta.

- *Escala Likert*: escala proposta por Rensis Likert, em 1932, que busca identificar, além da concordância ou discordância de afirmações, o grau ou intensidade em que elas ocorrem. Nesta escala, um número é atribuído a cada opção de resposta para que seja possível aferir a direção de atitude do entrevistado em relação à alternativa (MATTAR, 1997 *apud* GODOY; MOURA; SANTOS, 2004). Este tipo de escala foi amplamente empregado no questionário como um meio de quantificar as diversas características qualitativas que há em um cluster.

◆ Perguntas abertas: refere-se ao tipo de pergunta em que o entrevistado não tem que encaixar a sua resposta em uma lista de possibilidades.

◆ Pergunta aberta/fechada: pergunta que, além de oferecer possibilidades de respostas predefinidas, permite que o entrevistado acrescente outras através da opção “Outro. Qual?”, por exemplo.

3.1.2) Aplicação do questionário

A aplicação do questionário foi feita com duas pessoas envolvidas há bastante tempo com programação de treinamentos de empresas:

- ◆ Um funcionário da *M & R* que já trabalha há mais de 15 anos com o treinamento de empresas.
- ◆ Uma funcionária de uma empresa parceira da *M & R* que também tem bastante *expertise* nesta área.

3.2) Entrevistas

Conforme apontado por TRGU (2004), há diversos tipos de entrevistas – estruturada, semi-estruturada, não-estruturada, discussões informais ou grupo focado. Neste trabalho foram utilizados os dois tipos explicados a seguir:

◆ Entrevista semi-estruturada: este tipo de entrevista é mais flexível, pois encoraja o entrevistado a discutir e fornecer respostas mais amplas, ao invés de ser forçado a seguir um padrão, como ocorre na entrevista estruturada. Destaca-se, porém, que na entrevista semi-estruturada o entrevistador ainda possui um roteiro ou um *checklist* de questões conjuntamente com um local apropriado para anotação das respostas, isto é feito para que a entrevista não se desvirtua.

◆ Discussões informais: refere-se às conversas realizadas com uma ou mais pessoas onde não há um propósito ou objetivo aparente e, conseqüentemente, inexistência de roteiros ou *checklist*. No entanto, tais discussões também podem ser fonte de dados, questões ou detecção de problemas importantes.

Portanto, além do questionário também foram realizadas entrevistas com as duas pessoas já citadas com o intuito de investigar algumas características da programação de treinamento de maneira mais profunda, como também para adquirir informações sobre outras funcionalidades que o *software* deveria ter e que não foram abordadas no questionário desenvolvido.

4) Desenvolvimento do *software*

Esta etapa refere-se ao desenvolvimento do *software* propriamente dito onde o principal objetivo foi fazer o programa atender o maior número de necessidades explicitadas/detectadas no questionário e nas entrevistas realizadas.

Apesar de grande parte dos autores pesquisados na revisão bibliográfica terem desenvolvido os *softwares* de TT em linguagem C, decidiu-se neste trabalho utilizar a programação VBA (*Visual Basic for Applications*) que está presente nos programas do pacote Office. No caso deste trabalho, o Excel foi escolhido como front end por permitir uma entrada de dados mais amigável (por meio de formulários) e também por ser um ambiente onde se pode gerar relatórios e gráficos de forma rápida, concisa e agradável esteticamente.

5) Aplicação do *software* desenvolvido: estudo de caso.

Posteriormente ao desenvolvimento do *software*, este foi aplicado em um estudo de caso para comprovar a sua eficácia e eficiência. Em seguida, foi feita uma análise

da solução obtida manualmente com a solução gerada pelo *software* em termos de custo e preferência de horário dos professores/instrutores²⁸ e alunos/funcionários²⁹.

²⁸ Os termos “professor” e “instrutor” serão utilizados neste trabalho indistintamente.

²⁹ Os termos “funcionário” e “aluno” serão utilizados neste trabalho indistintamente.

SOFTWARE DESENVOLVIDO

5. SOFTWARE DESENVOLVIDO

5.1 NECESSIDADES IDENTIFICADAS

Por meio do questionário e entrevistas realizadas foram identificadas as necessidades com as quais o *software* desenvolvido neste trabalho deveria lidar. Conforme já apontado na revisão bibliográfica, o problema de TT normalmente lida com dois tipos de restrições: *hard* e *soft constraints* (restrições duras e restrições flexíveis). A seguir são apresentadas as restrições apontadas nas entrevistas e questionários aplicados de acordo com qual categoria pertence.

5.1.1 RESTRICÇÕES DO TIPO HARD

As restrições consideradas *hard*, ou seja, aquelas sem as quais o TT não tem utilidade, foram:

- 1) Um mesmo professor/instrutor dando duas ou mais aulas ao mesmo tempo, contrariando, assim, as leis da física.
- 2) Uma mesma turma com duas aulas concomitantes (caso parecido com a restrição anterior).
- 3) A capacidade da sala de aula deve ser respeitada, ou seja, se em uma sala cabem no máximo 50 pessoas, este limite deve ser respeitado.
- 4) Muitas das matérias³⁰ destes programas de cursos empresariais seguem uma seqüência lógica, ou seja, há regras de precedência entre elas exatamente como há em universidades, ou seja, um aluno só pode assistir a aula de “Trabalho de Formatura II” caso já tenha passado pela aula de “Trabalho de Formatura I”.
- 5) Não é difícil imaginar que há dias ou horários em que os professores/instrutores não estão disponíveis para ministrar o curso. Os instrutores normalmente fazem parte do quadro de funcionários da empresa ou são terceiros. Neste último caso, o problema de indisponibilidade costuma ser maior, já que os contratados têm um horário específico a cumprir.

³⁰ Os termos “matéria” e “curso” serão utilizados neste trabalho indistintamente.

-
- 6) Também não é difícil imaginar que os alunos/funcionários também apresentam períodos em que estão indisponíveis para o treinamento, mesmo porque eles continuam com as suas funções cotidianas.
 - 7) Outra restrição que deve ser respeitada é o horário em que as salas estão disponíveis. As salas utilizadas nestes cursos empresariais são normalmente salas da própria empresa ou salas alugadas (principalmente em hotéis). Apesar deste último caso a restrição de horário de sala ser mais fácil de imaginar, em salas da própria empresa muitas vezes já existem eventos marcados como reuniões de projeto, reuniões de staff (equipe) etc. gerando, assim, indisponibilidades.
 - 8) Com relação às salas de aula, também foi identificado que o tipo de sala deve ser uma restrição do tipo *hard*, ou seja, cursos que precisam de computador devem ser alocados em salas que possuam computador; do mesmo modo, cursos que precisam de projetor devem ser alocados em salas que possuam tal equipamento.
 - 9) Outra necessidade identificada e que se não respeitada também contrariaria as leis da física é o tempo de deslocamento entre diferentes locais. Muitas empresas que estão recebendo o curso contam com filiais em mais de um local (bairro, cidade, estado), portanto, pode ocorrer que um ou mais instrutores precisem viajar. Não é difícil perceber, então, que neste meio tempo (tempo de deslocamento), nenhum curso pode ser alocado para tal instrutor, dado que isto seria inviável.
 - 10) Assim como os instrutores, também pode ocorrer que turmas inteiras viajem para filiais da empresa diferentes das quais trabalham. Neste caso, o tempo de deslocamento também deve ser computado para que nenhum curso seja alocado para a turma em questão neste período.
 - 11) Finalmente, a última restrição do tipo *hard* está relacionada à sala de aula que uma determinada combinação curso-turma é alocada. Este ponto foi discutido exaustivamente com os dois entrevistados e a conclusão que se chegou foi que uma determinada combinação curso-
-

turma, ao ser alocada em uma sala, suas outras sessões³¹ devem estar nesta mesma sala para não causar o inconveniente das turmas ficarem trocando de sala freqüentemente dado que o curso não mudou.

5.1.2 **RESTRICÇÕES DO TIPO SOFT**

Notou-se por meio dos questionários e entrevistas que as “restrições” desejáveis, ou seja, aquelas que o TT deve *tentar* resolver são³²:

- 1) Caso o preço de aluguel de salas seja muito alto, deseja-se que o TT gerado não apresente muitas salas.
- 2) Destaca-se que, em relação às salas, caso haja diferença entre o custo/hora, o operador do TT pode desejar que salas mais baratas sejam utilizadas.
- 3) Caso exista o desejo de que a programação de treinamentos no horizonte de um mês, então a utilização de um número de dias que exceda este prazo deve ser evitada.
- 4) Caso o peso e/ou custo de viagem de um instrutor seja muito alto, então o TT gerado também deverá minimizar/evitar que estes deslocamentos sejam freqüentes, ao menos que isto seja necessário.
- 5) Deseja-se que as preferências de horário dos instrutores sejam respeitadas. Destaca-se que a restrição do tipo *hard* associada a este quesito (horário dos professores) refere-se somente ao período em que o instrutor realmente não pode dar o curso. Entretanto, nos horários em que o mesmo está disponível, o programa deve permitir que haja a diferenciação entre horários em que o professor prefere lecionar e horários em que o professor não de ser alocado, mas caso isto aconteça não haverá problema.

³¹ Imagine que há um curso cuja duração total é de 10h, mas cuja carga máxima de curso em um dia seja de 2h. Para este caso, então, uma das possibilidades é este curso ter 5 sessões de duas horas.

³² Destaca-se que cada uma dessas restrições terá um peso associado para indicar qual é a sua importância relativa. Esta possibilidade de associar pesos permite que diversas simulações do programa possam ser realizadas com, por exemplo, cenário onde alugar sala é muito caro ou cenários onde a viagem de instrutores para diferentes locais é proibitivo etc.

- 6) Deseja-se também que a preferência dos alunos por horários específicos também seja respeitada assim como os professores.

5.1.3 OUTRAS CONSIDERAÇÕES

- 1) Apesar da possibilidade da turma poder viajar, esta informação será uma entrada do *software* e não uma decisão tomada por ele. Entretanto, o custo da viagem assim como um peso relativo foi acrescentado ao *software* como auxílio à decisão (saber o custo total) caso o operador do *software* deseje construir o TT considerando diversos cenários (turma 1 viaja ou não viaja, por exemplo).
- 2) Observa-se que o TT também deve lidar com casos em que a empresa possui mais de uma filial em diferentes localidades. Portanto, a combinação turma-matéria só poderá ser alocada em salas situadas no mesmo local em que a turma está sendo simulada. A diferenciação entre local de simulação e local original da turma faz-se necessário já que o operador do TT pode obrigar que turmas viajem de um local para outro caso ache tal estratégia vantajosa. Vale lembrar, porém, que a função objetivo do problema irá contabilizar o custo de tal viagem.
- 3) Os instrutores também contam com um local original (local em que residem) e um local de simulação. Portanto, os instrutores poderão viajar com conseqüente ônus da função objetivo.

5.2 A ESCOLHA DOS MÉTODOS

Conforme mostrado na Revisão Bibliográfica, há inúmeros métodos para a resolução do problema de TT. Neste trabalho o método escolhido foi a utilização da heurística construtiva com a Busca Tabu (BT); portanto, um método híbrido. Entretanto, diferentemente do que o leitor possa estar imaginando, a heurística construtiva não desempenhará um papel meramente secundário de gerar sementes para a BT, mas é por meio dela que se tentará achar uma resposta com o custo melhor possível. As soluções geradas pela HC serão então passadas para a BT que, por sua vez, sem mexer no custo, irá tentar melhorar a preferência de horário das entidades envolvidas no TT (instrutores e alunos).

O leitor pode estar estranhando a escolha da heurística construtiva para uma finalidade tão nobre quanto a escolha do melhor custo. Entretanto, conforme já foi comentado, em problemas envolvendo TT, normalmente a obtenção de uma solução razoavelmente boa já é aceitável. A heurística construtiva, por sua vez, é reconhecida pela facilidade de implementação e rapidez com que consegue gerar respostas. Esta capacidade de gerar diversas respostas em um curto espaço de tempo é desejável em um problema tão complexo como o de TT. Já a escolha da BT não será muito comentada, pois suas vantagens e funcionamento estão bem detalhados no ANEXO A deste trabalho.

Nos itens subseqüentes, o funcionamento dos métodos escolhidos é descrito com mais detalhe.

5.3 A LÓGICA DO SOFTWARE

5.3.1 TRATAMENTO DO PROBLEMA EM TRÊS PARTES

Tratando-se de um problema extremamente complexo e com várias decisões a serem tomadas, resolveu-se dividi-lo em três etapas. A figura a seguir ilustra as diferentes fases:

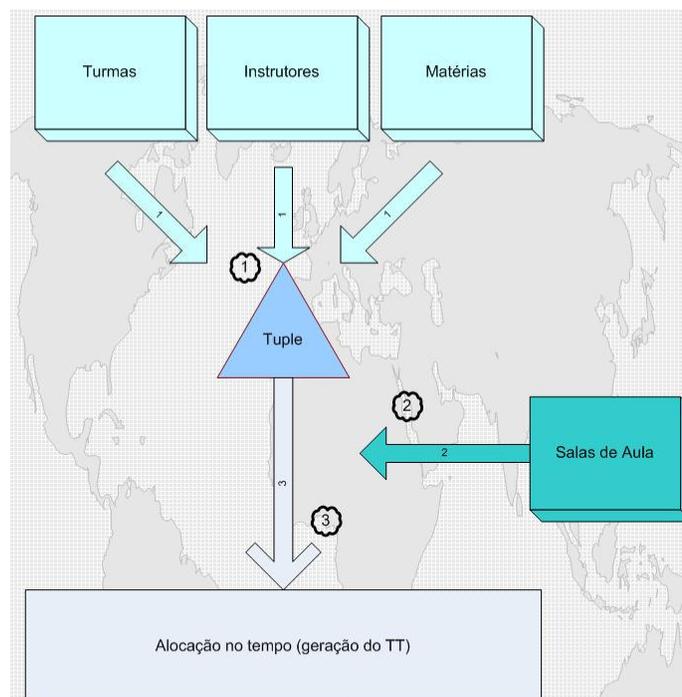


Figura 5: Representação das fases em que o problema foi dividido.

As três etapas ou fases referem-se, portanto, às seguintes decisões:

Etapa 1: dado a combinação turma-matéria, decidir qual instrutor será alocado.

Etapa 2: dado o *tuple* (turma-matéria-instrutor), decidir em que sala de aula ele será alocado.

Etapa 3: dado o *tuple* + sala, decidir em que tempo/horário (conhecido na literatura como *slot*) esta combinação será alocada.

Portanto, o TT estará completo quando todos os *tuples* forem alocados no tempo, em alguma sala de aula. Nos itens que se seguem, a idéia por trás de cada uma dessas etapas será explanada.

5.3.2 HEURÍSTICA CONSTRUTIVA PROBABILÍSTICA - INSTRUTOR

Uma das deficiências encontradas na literatura foi o fato de que em poucos artigos pesquisados o(s) autor(es) em questão mostra(m) ou entra(m) em mais detalhe sobre como foi formada a combinação aluno-professor-matéria. Pode-se afirmar que em nenhum artigo consultado para este trabalho o(s) autor(es) demonstra(m) o procedimento para se chegar nesta tripla que é mais conhecida na literatura pelo termo *tuple*. Esta questão é dita, na maioria dos casos, como já resolvida anteriormente ou tratada como mais um parâmetro de entrada do *software* ou da formulação desenvolvida.

Neste trabalho, porém, não há como se esquivar deste assunto, já que o *tuple* não é um parâmetro previamente definido. Na verdade, a matéria que uma turma irá assistir já está definida, entretanto o instrutor que irá dar esta matéria para esta turma deve ser definido pelo programa para assim completar o *tuple*. Portanto, esta questão de definição do instrutor acaba sendo um novo fator de complexidade do problema visto que cada curso pode ser dado por um ou mais instrutores e pode até haver casos em que instrutores preferem dar um curso a outro dependendo de sua especialidade.

Visando definir o instrutor que irá dar uma determinada matéria para uma turma específica, decidiu-se por utilizar a heurística construtiva probabilística por ser uma ferramenta rápida, de fácil implementação e que por utilizar a técnica de simulação de Monte-Carlo, produz diversas respostas diferentes. Por esta técnica, pode-se fazer o sorteio de professores de maneira totalmente aleatória ou também de maneira

viesada. A seguir os diferentes tipos de sorteio de um professor para determinada combinação de matéria-turma são explicados.

Critério	Explicação
<i>Afinidade</i>	Por este critério, quando for feito o sorteio do instrutor para uma determinada combinação de turma-matéria, aqueles instrutores que tem mais conhecimento (especialistas) no assunto envolvido terão preferência, ou seja, maior probabilidade de serem sorteados.
<i>Carga</i>	Por este critério, à medida que os professores são sorteados, aqueles que estão com uma carga didática mais baixa têm maior probabilidade de serem escolhidos. Este critério busca igualar a carga de aula de cada instrutor para que não haja super ou subutilização do mesmo.
<i>Revesado</i>	Por este critério, à medida que os professores são sorteados, os instrutores que foram escolhidos mais recentemente terão menor probabilidade de serem sorteados. Este critério também busca deixar os instrutores com uma carga didática parecida.
<i>Aleatório</i>	Este critério é na verdade um não-critério, ou seja, não há nenhum tipo de viés no sorteio. Portanto, os professores terão igual probabilidade de serem sorteados.
<i>Observação</i>	Obviamente, dado uma matéria, os instrutores que efetivamente podem ser sorteados para ensiná-las são somente aqueles cadastrados como sendo detentores deste conhecimento, ou como especialista ou como alguém que conhece o assunto. Portanto, um instrutor que está somente familiarizado com a matéria Y jamais poderá ser sorteado para ensinar a matéria X, independentemente do critério escolhido.

Tabela 8: Critérios para a escolha de professor

5.3.2.1 EXCEÇÃO NO SORTEIO DE INSTRUTORES

Caso o operador do TT tenha fixado que determinada turma-matéria tenha que ter um professor em específico, então nenhum sorteio será realizado.

5.3.3 HEURÍSTICA PROBABILÍSTICA - VIAGEM DE INSTRUTOR

Podem ocorrer que em determinados locais não haja instrutor para uma matéria específica. Neste caso algumas decisões podem ser tomadas:

- 1) Fazer com que a turma viaje para um local onde haja instrutor. Esta decisão, por ser extremamente importante, fica a cargo do operador do TT. Conforme será visto na seção de cadastro das informações, o operador pode decidir que determinada combinação turma-matéria ocorra em um local diferente daquele em que os membros da turma residem.
- 2) Fazer com que algum instrutor viaje para o local em que está faltando pessoal com conhecimento para ensinar determinada matéria. Esta decisão, diferentemente da anterior, pode ser deixada para que o programa faça. Entretanto, o operador do TT deverá indicar no cadastro quais professores podem viajar para ensinar determinada matéria para determinada turma. Caso o operador especifique mais de um instrutor, então o programa irá sortear um deles para ser deslocado. Nota-se que este sorteio é viesado, pois sempre o instrutor que estiver mais perto da localidade onde há carência apresentará vantagem em sua probabilidade de ser escolhido.

5.3.4 HEURÍSTICA CONSTRUTIVA PROBABILÍSTICA - SALA

Antes de alocar o *tuple* definido na etapa anterior, torna-se necessário saber em qual sala colocá-lo. Esta decisão também foi resolvida por meio da heurística construtiva probabilística. Na tabela a seguir, os diferentes critérios utilizados são explanados.

Critério	Explicação
<i>Proximidade</i>	Por este critério, busca-se alocar um determinado <i>tuple</i> (combinação matéria-turma-instrutor) na sala em que ele melhor se adequa, ou seja, se um <i>tuple</i> conta com 10 alunos, as salas com 10 alunos terão uma probabilidade maior de serem escolhidas. Portanto, este critério busca otimizar a capacidade das salas tentando alocar turmas grandes em salas grandes e turmas pequenas em salas menores.
<i>Carga</i>	Este critério almeja uma distribuição igualitária entre as salas de aula em relação a sua ocupação. Portanto, à medida que o sorteio de sala é realizado para os <i>tuples</i> , aquelas que já estão mais ocupadas terão menor probabilidade de serem escolhidas do que as que estão sendo menos utilizadas.
<i>Aleatório</i>	Esta falta de critério faz com que não haja nenhum tipo de viés na escolha das salas de aula. Portanto, todas as salas têm a mesma probabilidade de ser sorteada.
<i>Observação</i>	Vale lembrar que para um determinado <i>tuple</i> , as salas que poderão ser sorteadas são aquelas que se encontram no mesmo local que o <i>tuple</i> em questão. Conforme já mencionado, o <i>tuple</i> , ou melhor, a turma, possui um local original (onde os alunos constituintes residem) e um local de simulação, ou seja, localidade que o programa irá realmente considerar.

Tabela 9: Critérios para escolha de sala

5.3.5 HEURÍSTICA CONSTRUTIVA PROBABILÍSTICA E DETERMINÍSTICA – SELEÇÃO DOS TUPLES

Dado um *tuple*, ou seja, uma combinação matéria-turma-instrutor com a sua sala já sorteada, este deverá ser alocado no tempo. Nota-se pelas restrições do tipo *hard* que as regras que esta alocação deverá respeitar são capacidade da sala e regra de precedência entre as matérias. Todavia, também foram criadas heurísticas construtivas para esta alocação sendo uma determinística e duas probabilísticas (com e sem viés) conforme é elucidado na tabela subsequente.

Critério	Explicação
<i>Mais crítica (determinístico)</i>	Este critério visa alocar os <i>tuples</i> mais críticos primeiro. Esta criticidade é calculada observando-se a disponibilidade do professor e da turma, ou seja, os <i>tuples</i> constituídos de turmas e instrutores que possuem restrições de horário mais acentuadas certamente serão alocados primeiro. A criticidade é calculada multiplicando-se a quantidade de indisponibilidade do professor pela quantidade de indisponibilidade da turma. Portanto, <i>tuples</i> mais críticos de serem alocados terão um índice maior de criticidade do que aqueles mais folgados. Este critério busca facilitar a alocação dos <i>tuples</i> , já que se um <i>tuple</i> crítico for deixado para o final, a probabilidade de ela ser alocado será menor. Esta é uma heurística construtiva determinística, pois não há probabilidade em jogo, ou seja, as <i>tuples</i> mais críticas serão certamente escolhidas primeiro.
<i>Probabilidade da mais crítica.</i>	Este critério utiliza-se do mesmo índice explicado no critério anterior. No entanto, a principal diferença é que os <i>tuples</i> mais críticos terão maior probabilidade de serem escolhidos, portanto não é determinístico já que há probabilidade envolvida.
<i>Aleatória</i>	Novamente foi incluída uma maneira de sortear onde não há nenhum tipo de viés.

Tabela 10: Critérios para sortear *tuples*

5.3.6 ALOCAÇÃO DOS TUPLES NO TEMPO

Conforme visto no tópico anterior, há 3 maneiras de selecionar os *tuples*, ou seja, uma combinação contendo matéria-turma-instrutor. Também já foram explicadas as maneiras de se escolher a sala onde o *tuple* será alocado. Portanto, dado a combinação (*tuple* + sala), a única etapa que falta é a alocação no tempo. Esta, por sua vez, poderá ocorrer de três formas diferentes dependendo do cadastro da matéria feita pelo operador do TT. O cadastro permite que a matéria seja de 4 tipos diferentes:

Tipo	Explicação
<i>Aleatória</i>	Neste caso, o próprio programa irá sortear com igual probabilidade uma das três maneiras restantes.
<i>Mesmo Slot</i>	A melhor denominação para este critério seria “Mesmo Horário”. Caso este tipo seja escolhido para uma determinada matéria, então o programa irá fazer com que ela tenha o seu instante de início e término em cada dia igual aos outros. Explicando melhor, imagine que haja uma matéria com duração total de 8h e que a duração diária desta seja 2h, portanto 4 dias deverão ser utilizados (8/2). Imagine agora que o programa tenha conseguido alocar uma sessão dessa matéria começando às 2h (conseqüentemente, terminando às 4h) de um dia qualquer. Este critério, por sua vez, garantirá que nos outros dias em que a matéria foi alocada, ela começará às 2h e terminará às 4h impreterivelmente.
<i>Encaixar</i>	Caso uma determinada matéria tenha recebido este critério para alocação, então dois parâmetros devem ser fornecidos: duração total e duração diária assim como o critério anterior. Imagine que uma matéria tenha a duração total de 8h e duração diária de 2h, portanto esta matéria pode ser dividida em 4 blocos de 2h. Por este critério, cada um desses blocos será encaixado onde for possível no <i>timetabling</i> , logo o horário de começo e término terá grande probabilidade de não ser respeitado.
<i>Seguida</i>	Neste critério, somente um parâmetro adicional deverá ser fornecido: duração total da matéria. Este critério irá alocar a matéria em um ou mais dias sem interrupção. Imagine que cada dia tenha 8h disponíveis e que uma matéria possua 10h de duração total e foi configurada para ser ensinada de maneira “seguida”. Uma alocação possível respeitando este critério seria ocupar um dia interior as duas primeiras horas do dia seguinte, pois deste modo não há interrupção.

Tabela 11: Formas de alocar matéria

Portanto, cada combinação (*tuple* + sala) terá que obedecer ao tipo de alocação (aleatória, seguida, mesmo *slot*, encaixar) da matéria em questão.

5.3.7 PREFERÊNCIA DOS FUNCIONÁRIOS

Este tópico visa esclarecer como a preferência dos usuários é tratada pelo programa. No intuito de não comprometer o desempenho do programa, foi acordado entre o autor deste trabalho e as partes envolvidas neste projeto que a preferência de horário a ser respeitada pelo programa deveria ser a da turma e não de cada funcionário isoladamente. Imagine, portanto, que uma determinada turma seja composta por 3 funcionários e que a preferência de cada um seja dada pela tabela abaixo:

Funcionário	segunda-feira	terça-feira	quarta-feira	quinta-feira	sexta-feira
<i>Ricardo</i>	2	2	1	1	1
<i>Santoro</i>	1	1	2	2	2
<i>Joana</i>	1	2	1	2	0

Tabela 12: Exemplo de preferência de horário de funcionários (2=muito bom; 1=bom; 0=não)

Para resolver o conflito entre as preferências de cada funcionário, o programa irá, para cada dia (coluna), calcular a mediana e utilizá-la para os fins adequados. Portanto, o resultado seria dado pela seguinte tabela:

Turma	segunda-feira	terça-feira	quarta-feira	quinta-feira	sexta-feira
<i>Turma 1</i>	1	2	1	2	0

Tabela 13: Exemplo de preferência de horário de uma turma composta por 3 alunos

Pode-se dizer, então, que o programa desenvolvido é democrático, pois respeita a preferência da maioria. Entretanto, deve-se destacar o que aconteceu na sexta-feira. Neste caso um dos funcionários pertencentes à turma em questão não podia assistir a nenhuma aula, logo a turma inteira acabou sendo prejudicada, pois para uma turma ser alocada em um determinado dia, nenhum dos seus componentes pode apresentar indisponibilidade.

Nota-se que este conflito não ocorre para os instrutores, já que eles não estão agrupados. As salas de aula, por sua vez, também não contam com este tipo de agrupamento e também não possuem um estratégia de preferência de horário, pois ou elas podem ser utilizadas (valor 1) em determinado horário ou estão indisponíveis (valor 0).

5.4 *PARAMETRIZAÇÃO DA HEURÍSTICA CONSTRUTIVA*

Conforme já mencionado, a construção dos *tuples* assim como a escolha das salas é feita de maneira probabilística. A seleção de qual *tuple* será alocado primeiro também pode ser feita de maneira probabilística. Assim tem-se:

Instrutor	Afinidade
	Aleatório
	Carga
	Revesado
Sala	Proximidade
	Aleatório
	Carga
<i>Tuple</i>	Primeira Crítica
	Aleatório
	Probabilidade Crítica

Figura 6: Critérios de sorteio do professor, sala e *tuple*

Portanto, caso a combinação de critério (afinidade, proximidade, primeira crítica) seja escolhida, como os dois primeiros critérios envolvem probabilidade, então fica claro que a cada vez que uma solução (TT completo) é encontrada ela tem grande chance de ser diferente da anterior mesmo usando critérios iguais para construir e selecionar *tuple* e escolher sala. O resultado do uso de probabilidade é que para um mesmo critério pode-se gerar diversas respostas diferentes que só irão terminar quando algum critério de parada for atingido. Os critérios de parada estão explicados nos itens subseqüentes.

5.5 *CRITÉRIOS DE PARADA – HEURÍSTICA CONSTRUTIVA*

5.5.1 *NÚMERO DE SOLUÇÕES VIÁVEIS OBTIDAS*

Imagine que o operador do TT tenha escolhido as seguintes parametrizações (afinidade-proximidade-primeira crítica; afinidade-proximidade-aleatório). Para cada uma dessas combinações há um número muito grande de respostas. O critério do número de soluções viáveis faz com que as respostas geradas com uma combinação (afinidade-proximidade-primeira crítica, por exemplo) cessem quando um número N de soluções viáveis for atingido e comece a usar outra parametrização ou terminar o programa caso não haja mais nenhuma. Pode-se imaginar que nem todos os sorteios

e, conseqüentemente, soluções geradas sejam factíveis; neste caso, esse critério irá ignorar estas soluções dado que não irá contabilizá-las.

5.5.2 TEMPO POR PARAMETRIZAÇÃO

Suponha que novamente o operador do TT tenha escolhido duas formas de parametrizar os sorteios (afinidade-proximidade-primeira crítica {1}; afinidade-proximidade-aleatório {2}). Caso este critério seja usado, as respostas geradas com a {1} irá cessar assim que um tempo definido seja atingido. Ao alcançar este tempo, o cronômetro é zerado e o programa parte para as soluções geradas com a parametrização {2}. Finalmente, quando o critério de tempo for atingido novamente, as respostas geradas com a parametrização {2} irão acabar e o programar irá terminar.

5.5.3 NÚMERO DE ITERAÇÕES POR PARAMETRIZAÇÃO

Dado que o operador do TT tenha escolhido novamente as parametrizações {1} e {2}, ele pode definir quantas iterações o programa irá tentar em cada uma. Portanto, caso seja escolhido 10 como número de iterações máximo para cada parametrização, no final, 20 iterações serão obtidas e o programa irá terminar. Destaca-se que como o número de iterações está sendo limitado e não o número de soluções viáveis. Portanto, pode ser que no final das 20 iterações nenhuma solução factível tenha sido alcançada.

5.5.4 OUTROS CRITÉRIOS

Também foram desenvolvidos critérios de parada que independem da parametrização, ou seja, critérios globais. A seguir, os dois critérios criados são explanados.

5.5.4.1 CRITÉRIO DO TEMPO TOTAL

Caso o operador do TT queira utilizar-se deste critério, ele deverá estipular o tempo total que o programa deverá rodar. Ao ser atingido este tempo, o programa pára. Portanto, se o critério for de 30 min, após este tempo o programa irá parar de achar respostas e terminará repentinamente independente se há mais parametrizações a serem feitas. O programa irá mostrar ao operador do TT quais respostas ele conseguiu desenvolver em 30 min. Destaca-se que este tempo pode não ser exato, já

que se o programa estiver no meio de uma solução ao dar os 30min, ele irá completá-la para depois ser interrompido.

5.5.4.2 CRITÉRIO DO NÚMERO DE ITERAÇÕES TOTAL

Este critério permite ao operador de TT estipular o número de iterações total que o programa poderá realizar. Ao ser atingido este número, o programa parará. Portanto, por este critério há a possibilidade de nem todas as parametrizações serem utilizadas.

5.5.4.3 MISTURA DE CRITÉRIOS

Destaca-se que todos os critérios supracitados podem ser utilizados concomitantemente. Neste caso, o programa irá parar assim que o primeiro deles for atingido. Segue alguns exemplos do que pode ser feito:

Imagine que dois tipos de parametrização para os sorteios foram escolhidos (afinidade-aleatório-aleatório {1} e afinidade-carga-aleatório {2}). Pode-se então limitar cada uma dessas parametrizações pelo número de iterações ou pelo tempo. Se o operador colocar 10 iterações ou 1 min, assim que o primeiro for atingido o programa irá parar de gerar soluções para a parametrização {1} e começará a gerar para a {2}.

Imagine agora que as duas parametrizações do exemplo anterior também foram escolhidas e que os critérios utilizados foram “número de soluções viáveis” (10) e “critério de tempo total” (15 min). Neste caso, se o programa alcançar 20 soluções viáveis (10 para cada parametrização escolhida) antes de 15 min, então o critério de tempo não será utilizado. Entretanto, caso esteja sendo muito difícil encontrar soluções factíveis para o problema proposto, então os 15 min serão alcançados e o programa terminará.

5.6 ESCOLHA DA MELHOR SOLUÇÃO - ANÁLISE DE CUSTO

Escolhida as parametrizações de sorteio, tudo leva a crer que várias soluções viáveis serão escolhidas para cada tripla de sorteio (critério para instrutor-critério para sala – critério para *tuple*). Todavia, será vantajoso guardar todas as soluções achadas ?

No programa o usuário tem a opção de escolher quantas respostas ela irá armazenar por parametrização. Portanto, caso o operador escolha 10 respostas e duas parametrizações ($\{1\}$ e $\{2\}$) o programa irá guardar 20 respostas viáveis (10 para cada uma). Nota-se, porém, que as 20 respostas serão guardadas somente se o programa conseguir encontrá-las com o critério de parada proposto. Se o problema cadastrado for muito difícil de ser resolvido, há probabilidade de o programa não encontrar respostas viáveis.

O critério de escolha para as M melhores soluções é o custo. O custo de cada solução é composto das seguintes parcelas:

- ◆ Custo de abrir sala (custo fixo): caso uma sala seja usada, mesmo que seja para somente uma aula, haverá um custo associado (este custo será chamado de custo de abertura de sala).
 - ◆ Custo de utilizar a sala (custo variável): além do custo anterior, também há um custo por hora de utilização da sala (este custo será chamado de custo de utilização da sala). Este custo foi utilizado para diferenciar salas que apresentam custo/hora diferentes.
 - ◆ Custo dos dias excedentes: o operador de TT terá que cadastrar o horizonte máximo e o horizonte desejado da programação de cursos. O horizonte máximo é a quantidade de dias que o TT poderá utilizar para alocar os cursos, já o horizonte desejado refere-se à quantidade de dias que o operador de TT gostaria que fossem realmente utilizados. Toda vez que uma solução de TT for desenvolvida, o programa irá calcular o custo de dias excedentes fazendo a seguinte conta [(dias utilizados na solução – dias desejados pelo operador) * custo dos dias excedentes].
 - ◆ Custo de viagem do instrutor: toda vez que um instrutor tiver que viajar para dar algum curso, o custo de viagem onerará a função objetivo.
 - ◆ Custo de viagem da turma: toda vez que a turma viajar para assistir a algum curso, o custo associado será somado à função objetivo. Destaca-se que a decisão da turma viajar ou não é do operador do TT, portanto ele já sabe
-

previamente que as soluções geradas terão este custo. Ressalta-se também que o custo de viagem da turma é proporcional ao seu tamanho³³.

Conforme já mencionado, cada um destes custos apresenta um peso correspondente. Este peso serve para medir a importância relativa entre os fatores que podem onerar a função objetivo. Portanto, se o operador de TT quiser soluções com o menor número de dias possível, um peso grande deverá estar associado ao custo de dias excedentes. Em outra situação, caso o operador queira que o número de salas seja o menor possível, então o peso para o custo de salas deverá ser grande. Nota-se, portanto, que o operador do TT terá sempre que lidar com *tradeoffs* já que se escolher um peso relativamente grande para a abertura de salas, provavelmente as soluções obtidas terão um horizonte mais extenso do que o cenário onde o peso para o custo de dias excedentes fosse muito maior do que o de abertura de sala.

5.7 BUSCA TABU

O leitor atento deve ter percebido que nada foi mencionado até agora sobre a preferência de horário dos instrutores e funcionários. A tentativa de achar uma resposta considerada “boa” até o momento só levou em consideração as diversas parcelas de custo conforme já explicado. Entretanto, nessa parte do trabalho será apresentada uma estratégia para melhorar a solução de acordo com a preferência dos alunos sem alterar o custo³⁴.

A estratégia adotada foi a utilização da BT, com a qual o leitor já deve estar familiarizado (vide ANEXO A). O modelo de BT aplicado neste trabalho é exatamente o mesmo modelo simples explicado na Revisão Bibliográfica. Diz-se simples, pois a BT implementada apresenta somente memória de curto prazo. Na literatura específica sobre o assunto, observa-se que outras estratégias ou melhorias³⁵ podem ser aplicadas à BT como, por exemplo, memória de longo prazo, matriz de residência etc.

³³ Nota-se que o custo individual entre localidades é exatamente o mesmo tanto para o instrutor quanto para um funcionário. Não foi acrescentado nenhum tipo de política de desconto no caso de turma. Portanto, o custo de viagem da turma é igual ao custo individual entre as localidades em questão multiplicado pelo número de pessoas.

³⁴ A preferência dos instrutores não será tratada nesta versão do *software*.

³⁵ O termo “melhoria” foi aplicado, mas ressalva-se que nem sempre a adição de recursos na Busca Tabu gera resultados melhores.

A idéia por trás da BT aplicada no *software* desenvolvido neste trabalho é elucidada nos itens seguintes:

1. Semente inicial: solução obtida com as diversas heurísticas construtivas já citadas.
2. Partindo-se dessa semente, procurar novas soluções, ou seja, soluções vizinhas por meio de um movimento de troca (este passo é melhor explicado no próximo item).
3. Se houver alguma solução melhor do que a solução incumbente (a melhor solução obtida até o momento), então escolhê-la, independentemente se o movimento aplicado for proibido (movimento tabu).
4. Caso não haja nenhuma solução melhor do que a incumbente, então montar um subconjunto com todas as vizinhas geradas a partir de movimentos que não sejam proibidos.
5. Se o tamanho do conjunto anterior for maior do que zero, então escolher a melhor vizinha encontrada e atualizar a lista tabu, ou seja, acrescentar o movimento que gera esta vizinha à lista. Este acréscimo pode se dar tanto por meio da retirada de um outro movimento (caso a lista já esteja cheia) ou pela inclusão do movimento caso a lista tabu não esteja ainda totalmente cheia (provável início do problema).
6. Atualizar a semente com a nova solução encontrada.
7. Verificar se o critério de parada foi atingido, caso contrário, voltar ao passo 2.

5.7.1 VIZINHANÇA - MOVIMENTO DE TROCA

Em qualquer estratégia de busca, uma das configurações mais importantes refere-se à maneira de gerar soluções vizinhas dado uma solução completa. Conforme mencionado, no caso do *software* desenvolvido neste trabalho, a BT será responsável por melhorar a preferência dos usuários realizando movimentos que não alterem o custo.

Esses movimentos, por sua vez, também podem fazer com que “soluções” vizinhas inactíveis sejam criadas. Caso o leitor se interesse por buscas e pesquise mais a fundo este tema, observará que alguns autores permitem que vizinhas inactíveis sejam obtidas. O motivo é a crença de que esta estratégia poderá fornecer alguma vantagem, ou melhor, permitir que a busca chegue em regiões do espaço de soluções que ela jamais iria, caso somente vizinhas factíveis fossem aceitas. Apesar do autor deste trabalho acreditar nessa idéia, a BT desenvolvida não conta com esta liberdade. A razão para este conservadorismo reside no fato de que no problema de TT, o movimento adotado poderá perturbar em muito a solução de maneira que o seu conserto seja mais caro que a busca propriamente dita. Seguindo este raciocínio, o movimento desenvolvido foi o mais cuidadoso possível e os passos para obtê-lo são descritos nos itens subseqüentes:

- 1) Ache um *tuple* i (combinação instrutor-matéria-turma) qualquer.
 - 2) Criar um subconjunto L com os *tuples* que estão na mesma localidade que o *tuple* i .
 - 3) Criar um subconjunto M de L contendo os *tuples* cuja matéria seja igual a do *tuple* i .
 - 4) Para cada par (i,j) dado que j são os *tuples* pertencentes ao subconjunto M , analisar se a capacidade da sala desses dois *tuples* admitiria uma troca, ou seja, a seguinte pergunta deve ser feita: a capacidade utilizada da sala do *tuple* i no instante analisado menos o tamanho da turma do *tuple* i mais o tamanho da turma do *tuple* j é menor que a capacidade da referida sala. A mesma pergunta deve ser feita trocando-se o índice i por j e vice-versa.
 - 5) Se a análise anterior der certo, outra análise deverá ser feita: caso a troca fosse realmente efetivada, haveria algum problema temporal ? O significado de problema temporal pode ser traduzido nos seguintes pontos:
 - ◆ Um professor dando duas matérias ao mesmo tempo.
 - ◆ Uma turma assistindo duas matérias ao mesmo tempo.
-

-
- ◆ Matéria invadindo o tempo de seu predecessor ou sucessor.
 - ◆ Turma não tendo disponibilidade para assistir aulas no período alocado após a troca.
- 6) Se a análise anterior for positiva, então se deve verificar se os *tuples* que serão trocados estão unidos ou não. As seguintes situações podem ocorrer:
- ◆ Casos favoráveis à troca:
 - i. Um dos *tuples* (i ou j) está unido com outros, mas o que está desunido tem permissão para se juntar com outros *tuples*.
 - ii. Os dois *tuples* estão unidos.
 - iii. Os dois *tuples* estão desunidos. Neste caso não importa se eles podem se juntar ou não a outros *tuples*.
 - ◆ Casos desfavoráveis à troca:
 - i. Um dos *tuples* (i ou j) está unido com outros, mas o que está desunido não tem permissão para se juntar com outros *tuples*.
- 7) Caso a análise anterior resulte em alguma situação favorável à troca então deverá ser observado se após a troca a combinação turma-professor é viável. Isto é necessário, pois na troca, somente as turmas serão mexidas. Portanto, os professores é que, após a troca, receberão uma nova turma. Entretanto, conforme será visto posteriormente, a combinação professor-turma resultante nem sempre é possível, pois o operador do TT pode ter especificado previamente que um professor deverá dar aula para determinada turma ou que determinada turma pode ter qualquer professor desde que esteja em um subconjunto dado pelo operador de TT.
-

- 8) Finalmente, se todas as ressalvas anteriores foram favoráveis à troca, então o movimento é viável e deverá ser considerado para a geração de soluções vizinhas.

5.7.2 CRITÉRIOS DE PARADA DA BT

Conforme explicado no ANEXO A, a única maneira de uma BT cessar é o alcance de algum critério de parada. Neste trabalho os seguintes critérios foram desenvolvidos:

5.7.2.1 TEMPO POR PARAMETRIZAÇÃO

Suponha que o operador do TT tenha escolhido duas formas de parametrizar os sorteios das heurísticas construtivas probabilísticas já explicadas (afinidade-proximidade-primeira crítica {1}; afinidade-proximidade-aleatório {2}). Se o operador do TT também configurou o *software* para escolher as duas melhores soluções de cada parametrização, então ao final da heurística construtiva, se tudo der certo³⁶, 4 soluções serão achadas.

Seguindo o raciocínio, se o operador do TT configurou o *software* para realizar a BT, então esta será executada para 4 soluções independentemente. Entretanto, o operador do TT conta com a opção de limitar o tempo em que a BT irá gastar em cada uma dessas soluções. Portanto, esta é a finalidade deste critério: fazer com que a BT gaste o mesmo tempo para melhorar a preferência de horário dos instrutores e dos alunos em cada uma das 4 soluções obtidas.

5.7.2.2 NÚMERO DE ITERAÇÕES POR PARAMETRIZAÇÃO

Dado as condições do item anterior, ou seja, 4 soluções no total foram obtidas, o operador do TT conta com a opção de limitar o número de iterações que a BT irá tentar em cada uma. Portanto, caso seja escolhido 10 como número de iterações máximo para cada solução, no final, 40 iterações serão obtidas e o programa irá terminar.

³⁶ Essa observação foi feita, pois dependendo do problema e do critério de parada, existe a possibilidade de um número menor de soluções, em relação ao especificado pelo operador de TT, seja obtido. Pensando em um caso extremo, ou seja, se o problema for muito restrito, pode acontecer de, no final da execução do *software*, nenhuma solução viável tenha sido encontrada.

5.7.2.3 OUTROS CRITÉRIOS

Também foram desenvolvidos critérios de parada que independem das soluções, ou seja, critérios globais. A seguir, os dois critérios criados são explanados.

5.7.2.3.1 CRITÉRIO DO TEMPO TOTAL

Caso o operador do TT queira utilizar-se deste critério, ele deverá estipular o tempo total que a BT passará em execução. Ao ser atingido este tempo, o programa pára. Portanto, se o critério for de 30 min, após este tempo o programa irá parar de achar respostas e terminará repentinamente, independente se há mais soluções a serem melhoradas. O programa irá mostrar ao operador do TT quais respostas ele conseguiu desenvolver em 30 min. Destaca-se que este tempo pode não ser exato, já que se o programa estiver no meio de uma solução ao dar os 30min, ele irá completá-la para depois ser interrompido.

5.7.2.3.2 CRITÉRIO DO NÚMERO DE ITERAÇÕES TOTAL

Este critério permite ao operador de TT estipular o número de iterações total que a BT poderá realizar. Ao ser atingido este número, o programa parará. Portanto, por este critério há a possibilidade de nem todas as soluções iniciais (sementes) serem melhoradas.

5.7.2.3.3 MISTURA DE CRITÉRIOS

Destaca-se que todos os critérios supracitados podem ser utilizados concomitantemente. Neste caso, o programa irá parar assim que o primeiro deles for atingido.

EXEMPLO DE APLICAÇÃO

6. EXEMPLO DE APLICAÇÃO

Apesar de comentários, explicações e observações sobre o *software* desenvolvido já tenham sido dados, o autor deste trabalho acredita que a utilização do programa em um exemplo simples possa suprir dúvidas que o leitor porventura possa ainda ter. Como esta explicação direciona-se aos leitores mais curiosos, não comprometendo o restante da leitura deste trabalho, ela será apresentada no ANEXO B. O próximo item já apresenta um estudo de caso um pouco mais complexo baseado no que realmente ocorre na prática.

ESTUDO DE CASO

7. ESTUDO DE CASO

No capítulo anterior, o leitor deparou-se com um exemplo simples para o qual algumas funcionalidades do *software* desenvolvido foram demonstradas. Neste capítulo, porém, o leitor poderá analisar o funcionamento do programa com um exemplo mais complexo. Como a primeira programação de treinamentos de grande porte da empresa *M & R* só irá ocorrer no próximo ano, buscou-se construir um exemplo fictício, mas que contenha situações que realmente ocorrem na prática. No intuito de dar mais realismo ao exemplo, sua elaboração foi feita por um dos sócios da *M & R* que possui aproximadamente 10 anos de experiência nesta área. Portanto, o leitor pode imaginar o Estudo de Caso como uma amostra de problemas verossímeis.

No item subsequente será relatado o estudo de caso em questão.

7.1 DESCRIÇÃO DO ESTUDO DE CASO

Inicialmente, visando situar o leitor quanto à ordem de grandeza deste estudo de caso, o exemplo colocado no ANEXO B possui 6 *tuples* para serem alocados no total. Este estudo de caso, por sua vez, conta com 64.

7.1.1 A EMPRESA

O estudo de caso trata da empresa *SRV*. Esta empresa, fundada no ano de 2000, é atualmente uma exímia fabricante de guitarras, violões acústicos e teclados. Entretanto, como grande parte dos fabricantes de instrumentos, a *SRV* começou com poucos funcionários (10) e a fabricação era essencialmente artesanal. Situada, naquela época, em um casebre, no bairro de Pinheiros (SP), a empresa *SRV* começou suas atividades vendendo guitarras de marca própria com também instrumentos consignados. As vendas dos instrumentos de marca própria eram inicialmente voltadas para iniciantes no aprendizado de guitarra, pois a estratégia da empresa estava voltada para custo. Portanto, os aspirantes a músicos podiam encontrar na Av. Teodoro Sampaio, uma loja em que os instrumentos apresentavam preços acessíveis ao contrário de marcas como *Fender®* ou *Gibson®*.

O dono da loja, Mr. Clapton, não estava contente com aquela situação e resolveu, no ano de 2002, dar uma guinada em sua vida e seguir o que realmente sonhava: construir uma fábrica de guitarras de classe mundial e que servisse como

benchmarking para fábricas emergentes, de pessoas como ele. A primeira decisão de Mr. Clapton foi de seguir os conselhos dados por gurus da administração como Michael Porter dentre outros. Mr. Clapton vislumbrou que sua fábrica só poderia dar certo caso idéias como foco no cliente, diferenciação, gestão do conhecimento e responsabilidade social fossem levados em conta. A partir daquele dia, 21 de novembro de 2002, aquela pequena fábrica de caráter inicialmente familiar começou a se transformar.

No meio do ano seguinte, 2003, Mr. Clapton obteve uma ajuda essencial e que marcou profundamente sua carreira como empreendedor. Sua mulher, Mrs. Raitt, PhD em Engenharia Química, descobriu um novo material sintético que poderia substituir as madeiras atuais utilizadas nas guitarras e violões. A sorte de Mr. Clapton foi que, além de contribuir com o meio-ambiente, o novo material produzia um som inigualável. Esta notícia, assim como as novas guitarras fabricadas por Mr. Clapton, espalharam-se rapidamente pelas comunidades do ramo artístico. Em pouco tempo, a fábrica de Mr. Clapton se expandiu em uma velocidade vertiginosa. Para encurtar a história, no começo do ano de 2005, Mr. Clapton havia profissionalizado o comando da empresa e adotado práticas como *just-in-time* (o cliente monta sua guitarra ou violão em um portal na rede mundial de computadores, sendo a produção puxada), QFD (*Quality Function Deployment*), FMEA (*Failure Mode and Effect Analysis*), BSC (*Balanced Scorecard*) etc. guinando, desta maneira, sua fábrica para uma estratégia calcada na diferenciação e foco no cliente.

Atualmente, Mr. Clapton possui, no total, três unidades de negócio (guitarra, violão e teclado) em diferentes locais. A empresa, por ter crescido muito, está implementando um sistema de ERP para melhorar o fluxo de informações e processos internos. Entretanto, a empresa precisará, perto do final do projeto (*go-live*), treinar funcionários de diferentes áreas para capacitação nos conceitos e também na manipulação do sistema. Este treinamento deverá ocorrer no menor tempo possível para que os funcionários, assim que o sistema de ERP entrar em funcionamento, ainda estejam com o conhecimento adquirido na memória. Abaixo estão relacionados alguns números referentes à empresa:

- ◆ No total 120 funcionários, entre gerentes e pessoas de nível operacional deverão ser treinados. Eles trabalham em áreas como logística, recursos humanos, vendas, produção, finanças e marketing.
- ◆ Os funcionários estão distribuídos pelas 3 unidades de negócio da empresa, sendo 70 no local 1, 25 no local 2 e 25 no local 3.
- ◆ Em cada local há dois tipos de sala: uma de pequeno porte e uma de médio porte (chamada de auditório). As salas estão distribuídas da seguinte maneira:
 - Local 1: 3 salas de pequeno porte e 2 auditórios.
 - Local 2: 1 sala e 1 auditório.
 - Local 3: 2 salas e 1 auditório.
- ◆ Ao todo, 9 funcionários da empresa *SRV*, que estão participando do projeto, estão aptos a ensinar as matérias para os demais colaboradores. Estes funcionários são os chamados multiplicadores de conhecimento e estão dispersos entre as localidades da seguinte maneira:
 - Local 1: 5 instrutores.
 - Local 2: 3 instrutores.
 - Local 3: 1 instrutor.

7.1.2 PARTICIPAÇÃO DA M & R

Tendo o cenário descrito no item anterior, a empresa *M & R* foi contratada para estabelecer um plano de treinamento dos usuários do novo sistema. Esta atividade envolve as seguintes tarefas:

- ◆ Estabelecer quais matérias cada funcionário deverá assistir.
 - ◆ Montar as turmas de acordo com algum padrão notado ao final da tarefa anterior.
 - ◆ Montar uma seqüência lógica entre as matérias (regra de precedência).
 - ◆ Definir quantos instrutores serão necessários, assim como indicar quantos e quais cursos cada um deles irá ensinar.
-

◆ Fazer a programação de treinamentos propriamente dita, ou seja, alocar a combinação (matéria-turma-instrutor) em salas e no tempo. Conforme foi colocado, essa programação deverá ser feita no menor intervalo de tempo possível.

Dispondo agora de um *software* que permite a geração de TT de uma maneira mais fácil, a empresa *M & R* decidiu utilizá-lo como forma de testar a sua confiabilidade e ver como ele se porta em problemas, de pequeno para médio porte, e que apresentam certas dificuldades como falta de instrutor em algumas localidades e tipos de salas diferentes. Entretanto, antes de utilizar o *software*, algumas considerações ou ações devem ser tomadas:

◆ O *software* não decide (e nem poderia) quais cursos os funcionários deverão assistir assim como a regra de precedência entre as matérias envolvidas. Portanto, tais informações devem ser estudadas e elaboradas previamente, pois servirão como parâmetro de entrada do programa.

◆ Conforme comentado, algumas localidades estão carentes de instrutores que possam ensinar determinadas matérias. Como já foi explicado neste trabalho (ANEXO B), o operador do TT deverá fornecer, para estes casos, uma lista de quais instrutores poderão se deslocar para suprir tal carência.

◆ A disponibilidade, assim como a preferência de horário dos funcionários e dos instrutores, devem ser levantadas por meio de questionários (preferencialmente, via e-mail), pois também são aspectos pessoais que servem como parâmetros de entrada para o *software*.

A *M & R* executou 4 vezes o *software* com diferentes configurações e também, paralelamente, montou uma solução manual para o estudo de caso em questão a fim de comparar o horizonte total da programação de treinamentos. Os cadastros utilizados para estas soluções podem ser encontrados no ANEXO C. Já as soluções obtidas, assim como suas análises, são apresentadas no próximo capítulo.

ANÁLISE DOS RESULTADOS

8. ANÁLISE DOS RESULTADOS

8.1 EXECUÇÃO 1

Nesta primeira tentativa de achar uma solução para o problema, o *software* foi executado³⁷ com as seguintes configurações:

8.1.1 PARAMETRIZAÇÃO

Prof	afinidade	1
	aleatorio	1
	carga	1
	revesado	1
Sala	proximidade	1
	aleatorio	1
	carga	1
Tuple	primCritica	1
	aleatorio	1
	probCritica	1

Figura 7: Parametrização utilizada na execução 1.

Portanto, todos os parâmetros foram utilizados. No total há 36 combinações diferentes (4 x 3 x 3).

8.1.2 CRITÉRIOS DE PARADA

O seguinte critério de parada foi utilizado:

- ◆ 5 minutos ou 100 soluções para cada parametrização, o que limitar primeiro.

8.1.3 OUTRAS CONSIDERAÇÕES

Importante dizer que:

- ◆ A BT não foi executada, pois com não há prioridade para a preferência de horário dos funcionários, tal medida não seria útil.
- ◆ Configurou-se o programa para guardar as 5 melhores soluções de cada parametrização. Portanto, ao todo, 180 (5 x 36) soluções podem ser achadas.

³⁷ Essa execução assim como as outras foram realizadas em um computador com processador AMD ATHLON 2.400+ 1.45 GHz e 448MB de RAM.

- ◆ O programa pode tentar achar respostas no período de 3 semanas, entretanto, todas as alocações que passarem de 2 semanas (horizonte desejado) serão penalizadas (os pesos estão no ANEXO C).

8.1.4 SOLUÇÕES ENCONTRADAS

8.1.4.1 RELATÓRIO

A seguir, o relatório gerado pelo programa é mostrado junto com uma breve explicação.

O relatório mostra as soluções encontradas em ordem crescente de custo. Neste caso, pela grande extensão da lista de soluções, somente as 15 primeiras foram reproduzidas. As informações importantes que podem ser extraídas deste relatório são:

- ◆ A melhor solução foi encontrada após 16 min da execução do programa (o tempo da coluna “TempoMelhor” está em segundos). Também se observa que as três melhores soluções foram encontradas em menos de 1h.
 - ◆ O critério que realmente limitou as soluções mostradas foi o número de iterações por parametrização. Observa-se na coluna “NumSolCompleta” que 100 iterações foram alcançadas em cada caso.
 - ◆ A melhor solução foi a de número 42 (de 100) da sua parametrização. Explicando melhor, esta solução pertence à parametrização {afinidade, aleatório, primCrítica}. Para esta parametrização, 100 soluções completas (viáveis) foram encontradas e a melhor foi a de número 42 conforme visto na coluna “NumSol”.
 - ◆ As 3 primeiras soluções apresentam o horizonte de programação exatamente igual ao horizonte desejado (2 semanas).
 - ◆ Os custos incorridos pela melhor solução foram: 11.000 (custo por utilizar as salas) e 620 (custo de viagem do instrutor).
 - ◆ O valor da função objetivo da melhor solução (3.111.000) pode ser visto na coluna denominada “fit”. Destaca-se que este valor corresponde à soma de cada custo multiplicado pelo peso correspondente.
 - ◆ O índice que registra o atendimento às preferências de horário dos funcionários e dos instrutores registra o número 352 para ambos.
 - ◆ Conforme foi comentado, a BT não foi executada justificando a seção sem dados do relatório.
-

Seq21	Seq22	Seq23	Seq24	Ter21	Ter22	Ter23	Ter24	Qua21	Qua22	Qua23	Qua24	Qui21	Qui22	Qui23	Qui24	Sex21	Sex22	Sex23	Sex24
Sonia-PROD3-T1E//	Sonia-PROD3-T1E//	Sonia-PROD3-T1E//	Sonia-PROD3-T1E//	João-LOG2-T16//João-LOG2-T17//João-LOG2-T18//	João-LOG2-T16//João-LOG2-T17//João-LOG2-T18//	--	--	João-LOG2-T16//João-LOG2-T17//João-LOG2-T18//	João-LOG2-T16//João-LOG2-T17//João-LOG2-T18//	Antonio-LOG3-T17//	Antonio-LOG3-T17//	Antonio-LOG3-T17//	Antonio-LOG3-T17//	--	--	--	--	--	--
Carlos-RH2-T1C//	Carlos-RH2-T1C//	Carlos-RH2-T1C//	Carlos-RH2-T1C//	Maria-REL-T1D//	Maria-REL-T1D//	Maria-REL-T1D//	Maria-REL-T1D//	Maria-FIN2-T12//	Maria-FIN2-T12//	Maria-FIN4-T14//	Maria-FIN4-T14//	Maria-FIN2-T12//	Maria-FIN2-T12//	Antonio-LOG4-T18//	Antonio-LOG4-T18//	Antonio-LOG4-T18//	Antonio-LOG4-T18//	--	--
Maria-FIN3-T13//	Maria-FIN3-T13//	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Maria-REL-T25//	Maria-REL-T25//	Maria-REL-T25//	Maria-REL-T25//
--	--	--	--	Carlos-RH2-T36//	Carlos-RH2-T36//	Carlos-RH2-T36//	Carlos-RH2-T36//	--	--	--	--	--	--	--	--	--	--	--	--
Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Sonia-REL-T35//	Sonia-REL-T35//	Sonia-REL-T35//	Sonia-REL-T35//
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Tabela 15: Programação da melhor solução (divida em 2 partes)

8.1.4.2.1 OBSERVAÇÕES IMPORTANTES

Vale destacar que:

- ◆ Essa execução demorou 2h10min registrando um total de 3.005 soluções viáveis (23 soluções por minuto, em média).

◆ Todos os requisitos preenchidos nos cadastros foram respeitados (capacidade da sala, aptidão do instrutor, tempo de deslocamento etc.). Exemplificando, se o leitor reparar na instrutora Ana, ela viaja do seu local original (local 2) para o local 3 no intuito de ensinar a matéria FIN1 para a turma T33. Ela termina de dar aula no local 2 na “Sex12” e começa a dar aula no local 3 na “Seg21”. Caso o leitor tenha lido o ANEXO C, reparou que no tempo de deslocamento entre locais está cadastrado “-1”. Este número negativo significa que o instrutor só está disponível para dar aula em outro local no dia seguinte. Portanto, é justamente isto que ocorre, a Ana está disponível para dar aula no dia seguinte (neste caso o primeiro *slot* da segunda-feira da segunda semana – Seg21) e assim o faz.

◆ Os custos de viagem de instrutor são justificados pelos seguintes deslocamentos:

○ Carlos: 1-3-1-3-1³⁸

○ Sônia: 3-1-3

○ Ana: 2-3-2

○ Raul: 2-3-2

○ Marcos: 1-2-1

○ Maria: 1-2-1

▪ Lembre-se que o programa conta que no final da programação o instrutor volta para o seu local de origem.

▪ O custo de deslocamento do local 3 para 2 é de \$10. Os demais são 50 (inclusive 2 para 3). Total: \$620.

◆ Os custos de utilização da sala são justificados pelos seguintes fatos:

◆ Foram utilizados 156 *slots* em salas pequenas (\$50/*slot*). Total: \$7.800.

◆ Foram utilizados 16 *slots* em salas grandes (\$200/*slot*). Total: \$3.200.

8.2 EXECUÇÃO 2

Nesta tentativa de achar uma solução para o problema, o *software* foi executado com as seguintes configurações:

8.2.1 PARAMETRIZAÇÃO

Prof	afinidade	1
	aleatorio	1
	carga	1
	revesado	1
Sala	proximidade	1
	aleatorio	1
	carga	1
Tuple	primCritica	1
	aleatorio	1
	probCritica	1

Figura 8: Parametrização utilizada na execução 1.

Portanto, todos os parâmetros foram utilizados. No total há 36 combinações diferentes ($4 \times 3 \times 3$).

8.2.2 CRITÉRIOS DE PARADA

O seguinte critério de parada foi utilizado:

- ◆ 3 minutos ou 100 soluções para cada parametrização, o que limitar primeiro.

8.2.3 OUTRAS CONSIDERAÇÕES

Importante dizer que:

- ◆ Configurou-se o programa para guardar as 5 melhores soluções de cada parametrização. Portanto, ao todo, 180 (5×36) soluções podem ser achadas.
- ◆ O programa pode tentar achar respostas no período de 3 semanas, entretanto, todas as alocações que passaram de 2 semanas (horizonte desejado) serão penalizadas (os pesos estão no ANEXO C).

³⁸ Leia-se: foi do local 1 para 3, depois para o 1, voltou para o 3 e finalmente chegou a seu local de origem novamente.

8.2.4 SOLUÇÕES ENCONTRADAS

8.2.4.1 RELATÓRIO

A seguir, o relatório gerado pelo programa é mostrado junto com uma breve explicação.

O relatório mostra as soluções encontradas em ordem crescente de custo. Neste caso, pela grande extensão da lista de soluções, somente as 15 primeiras foram reproduzidas. As informações importantes que podem ser extraídas deste relatório são:

- ◆ A melhor solução foi encontrada após 1h30min da execução do programa (o tempo da coluna “TempoMelhor” está em segundos). Também se observa que as três melhores soluções foram encontradas em menos de 2h.
 - ◆ O critério que realmente limitou as soluções mostradas foi o tempo por parametrização. Observa-se na coluna “NumSolCompleta” que nenhuma das parametrizações utilizadas conseguiu atingir 100 soluções completas em 3 min.
 - ◆ A melhor solução foi a de número 26 (de 78) da sua parametrização. Explicando melhor, esta solução pertence à parametrização {revesado, aleatório, primCrítica}. Para esta parametrização, 78 soluções completas (viáveis) foram encontradas e a melhor foi a de número 26 conforme visto na coluna “NumSol”.
 - ◆ Somente a primeira solução apresenta o horizonte de programação exatamente igual ao horizonte desejado (2 semanas).
 - ◆ Os custos incorridos pela melhor solução foram: 11.200 (custo por utilizar as salas) e 520 (custo de viagem do instrutor).
 - ◆ O valor da função objetivo da melhor solução (2.611.200) pode ser visto na coluna denominada “fit”. Destaca-se que este valor corresponde à soma de cada custo multiplicado pelo peso correspondente.
 - ◆ O índice que registra o atendimento às preferências de horário dos funcionários e dos instrutores registra o número 352 para ambos.
 - ◆ Conforme esperado, a BT não conseguiu melhor nenhuma solução, pois isso seria impossível dado que as prioridades de horário para os alunos não diferem umas das outras.
-

Seq21	Seq22	Seq23	Seq24	Ter21	Ter22	Ter23	Ter24	Qua21	Qua22	Qua23	Qua24	Qui21	Qui22	Qui23	Qui24	Sex21	Sex22	Sex23	Sex24
Sonia- PROD3- T1E//	Sonia- PROD3- T1E//	Sonia- PROD3- T1E//	Sonia- PROD3- T1E//	Maria- FIN3- T13//	Maria- FIN3- T13//	--	--	--	--	Antonio- LOG4- T18//	Antonio- LOG4- T18//	Antonio- LOG4- T18//	Antonio- LOG4- T18//	--	--	--	--	--	--
Maria- REL- T1D//	Maria- REL- T1D//	Maria- REL- T1D//	Maria- REL- T1D//	João- LOG2- T18//	João- LOG2- T18//	--	--	João- LOG2- T18//	João- LOG2- T18//	Sonia- VEN1- T1B//	Sonia- VEN1- T1B//	Sonia- VEN1- T1B//	Sonia- VEN1- T1B//	Sonia- VEN1- T1B//	Sonia- VEN1- T1B//	Sonia- VEN1- T1B//	Sonia- VEN1- T1B//	--	--
--	--	João- LOG2- T18//João- LOG2- T17//	João- LOG2- T18//João- LOG2- T17//	Antonio- LOG3- T17//	Antonio- LOG3- T17//	Antonio- LOG3- T17//	Antonio- LOG3- T17//	Maria- FIN2- T12//	Maria- FIN2- T12//	Maria- FIN4- T14//	Maria- FIN4- T14//	Maria- FIN2- T12//	Maria- FIN2- T12//	--	--	--	--	--	--
Marcos- PROD1- T23//Marc os- PROD1- T24//	Marcos- PROD2- T23//	Maria- REL-T25//	Maria- REL-T25//	Maria- REL-T25//	Maria- REL-T25//														
Carlos- RH1- T32//Carl os-RH1- T36//	Carlos- RH1- T32//Carl os-RH1- T36//	Carlos- RH1- T32//Carl os-RH1- T36//	Carlos- RH1- T32//Carl os-RH1- T36//	Carlos- RH2-T36//	Carlos- RH2-T36//	Carlos- RH2-T36//	Carlos- RH2-T36//	--	--	--	--	--	--	--	--	--	--	--	--
Ana-FIN1- T33//	Ana-FIN1- T33//	Raul- LOG1- T34//	Raul- LOG1- T34//	Ana-FIN1- T33//	Ana-FIN1- T33//	Raul- LOG1- T34//	Raul- LOG1- T34//	Ana-FIN1- T33//	Ana-FIN1- T33//	Raul- LOG1- T34//	Raul- LOG1- T34//	Ana-FIN1- T33//	Ana-FIN1- T33//	Raul- LOG1- T34//	Raul- LOG1- T34//	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Tabela 17: Programação da melhor solução (divida em 2 partes)

8.2.4.2.1 OBSERVAÇÕES IMPORTANTES

Vale destacar que:

- ◆ Essa execução demorou 1h48min registrando um total de 1.543 soluções viáveis (14 soluções por minuto, em média).
- ◆ Todos os requisitos preenchidos nos cadastros foram respeitados (capacidade da sala, aptidão do instrutor, tempo de deslocamento etc.).
- ◆ Os custos de viagem de instrutor são justificados pelos seguintes deslocamentos:

- Carlos: 1-3-1
- Sônia: 3-1-3

- Ana: 2-3-2
 - Raul: 2-3-2
 - Marcos: 1-2-1
 - Maria: 1-2-1
 - Lembre-se que o programa conta que no final da programação o instrutor volta para o seu local de origem.
 - O custo de deslocamento do local 3 para 2 é de \$10. Os demais são 50 (inclusive 2 para 3). Total: \$520.
- ◆ Os custos de utilização da sala são justificados pelos seguintes fatos:
 - ◆ Foram utilizados 160 *slots* em salas pequenas (\$50/*slot*). Total: \$8.000.
 - ◆ Foram utilizados 16 *slots* em salas grandes (\$200/*slot*). Total: \$3.200.

8.3 EXECUÇÃO 3

Nesta tentativa de achar uma solução para o problema, o *software* foi executado com as seguintes configurações:

8.3.1 PARAMETRIZAÇÃO

Prof	afinidade	1
	aleatorio	1
	carga	1
	revesado	1
Sala	proximidade	1
	aleatorio	1
	carga	1
Tuple	primCritica	1
	aleatorio	1
	probCritica	1

Figura 9: Parametrização utilizada na execução 1.

Portanto, todos os parâmetros foram utilizados. No total há 36 combinações diferentes (4 x 3 x 3).

8.3.2 CRITÉRIOS DE PARADA

O seguinte critério de parada foi utilizado:

- ◆ 4 minutos para cada parametrização (o número de soluções não foi limitado).

8.3.3 OUTRAS CONSIDERAÇÕES

Importante dizer que:

- ◆ Nesta execução do *software* foram feitas algumas modificações no cadastro da preferência de horário dos alunos (vide ANEXO C). Esta modificação foi feita para que o leitor percebesse melhor como a BT funciona. Portanto, nesta execução o trabalho da BT não foi em vão. A BT foi limitada a 1 min ou 500 iterações para cada solução encontrada.
- ◆ Configurou-se o programa para guardar somente a melhor solução de cada parametrização. Portanto, ao todo, 36 (1 x 36) soluções podem ser achadas.
- ◆ O programa pode tentar achar respostas no período de 3 semanas, entretanto, todas as alocações que passaram de 2 semanas (horizonte desejado) serão penalizadas (os pesos estão no ANEXO C).

8.3.4 SOLUÇÕES ENCONTRADAS

8.3.4.1 RELATÓRIO

A seguir, o relatório gerado pelo programa é mostrado junto com uma breve explicação.

num	prof	sala	tuple	TempoMelhor	NumSol	NumSol/Completa	custoAbrirSala	PesoSala	custoUsarSala	peso	viagemProf	peso	viagemTurma	peso	fit
1	afinidade	carga	primCritica	1543	27	68	0	1	11000	1	500000	520	5000	0	5000
2	aleatorio	aleatorio	primCritica	3041	37	58	0	1	11000	1	500000	520	5000	0	5000
3	afinidade	aleatorio	primCritica	834	33	72	0	1	11000	1	500000	620	5000	0	5000
4	carga	carga	primCritica	5961	65	79	0	1	11200	1	500000	620	5000	0	5000
5	revesado	aleatorio	primCritica	7432	84	91	0	1	11000	1	2500000	520	5000	0	5000
6	carga	aleatorio	primCritica	5099	16	71	0	1	11000	1	4500000	520	5000	0	5000
7	aleatorio	carga	primCritica	3799	71	91	0	1	11200	1	4500000	520	5000	0	5000
8	revesado	carga	primCritica	8034	47	113	0	1	11200	1	4500000	520	5000	0	5000
9	carga	proximidade	primCritica	4364	1	21	0	1	11400	1	4500000	520	5000	0	5000
10	aleatorio	carga	aleatorio	3895	15	78	0	1	11000	1	4500000	620	5000	0	5000
11	afinidade	carga	aleatorio	1910	74	75	0	1	11000	1	4500000	620	5000	0	5000
12	aleatorio	aleatorio	probCritica	3366	1	67	0	1	11400	1	4500000	620	5000	0	5000
13	revesado	aleatorio	aleatorio	7595	46	66	0	1	11000	1	4500000	720	5000	0	5000
14	afinidade	proximidade	primCritica	171	17	21	0	1	11000	1	6500000	520	5000	0	5000
15	revesado	proximidade	primCritica	6623	24	37	0	1	11200	1	6500000	520	5000	0	5000

preprof	peso	prefTurma	peso	fitpref	prefProf	peso	prefTurma	peso	fitpref	% BT
352	1	446	1	798	352	1	446	1	798	0,00%
352	1	442	1	794	352	1	442	1	794	0,00%
352	1	444	1	796	352	1	444	1	796	0,00%
352	1	440	1	792	352	1	448	1	800	1,01%
352	1	446	1	798	352	1	446	1	798	0,00%
352	1	442	1	794	352	1	442	1	794	0,00%
352	1	444	1	796	352	1	452	1	804	1,01%
352	1	442	1	794	352	1	450	1	802	1,01%
352	1	452	1	804	352	1	452	1	804	0,00%
352	1	482	1	834	352	1	482	1	834	0,00%
352	1	450	1	802	352	1	450	1	802	0,00%
352	1	464	1	816	352	1	480	1	832	1,96%
352	1	478	1	830	352	1	478	1	830	0,00%
352	1	446	1	798	352	1	446	1	798	0,00%
352	1	442	1	794	352	1	450	1	802	1,01%

Tabela 18: Relatório mostrando as 15 melhores soluções (de um total de 36) para a terceira execução

O relatório mostra as soluções encontradas em ordem crescente de custo. Neste caso, pela grande extensão da lista de soluções, somente as 15 primeiras foram reproduzidas. As informações importantes que podem ser extraídas deste relatório são:

- ◆ A melhor solução foi encontrada após, aproximadamente, 26 min da execução do programa (o tempo da coluna “TempoMelhor” está em segundos). Também se observa que as três melhores soluções foram encontradas em menos de 1h.
- ◆ O critério que limitou as soluções mostradas foi o tempo por parametrização, pois foi o único utilizado.
- ◆ A melhor solução foi a de número 27 (de 68) da sua parametrização. Explicando melhor, esta solução pertence à parametrização {afinidade, carga, primCrítica}. Para esta parametrização, 68 soluções completas (viáveis) foram encontradas e a melhor foi a de número 27 conforme visto na coluna “NumSol”.
- ◆ As 4 primeiras soluções apresentam o horizonte de programação exatamente igual ao horizonte desejado (2 semanas).
- ◆ Os custos incorridos pela melhor solução foram: 11.000 (custo por utilizar as salas) e 520 (custo de viagem do instrutor).
- ◆ O valor da função objetivo da melhor solução (2.611.000) pode ser visto na coluna denominada “fit”. Destaca-se que este valor corresponde à soma de cada custo multiplicado pelo peso correspondente.
- ◆ Os índices que registram o atendimento às preferências de horário dos funcionários e dos instrutores indicam o número 352 e 446.
- ◆ Observa-se agora que a BT conseguiu melhorar algumas soluções (vide última coluna do relatório com as porcentagens).

8.3.4.2 PROGRAMAÇÃO DA MELHOR SOLUÇÃO ENCONTRADA

Eis a programação da melhor solução encontrada. Ela está dividida em duas partes (2 semanas) devido ao seu tamanho.

Seq21	Seq22	Seq23	Seq24	Ter21	Ter22	Ter23	Ter24	Qua21	Qua22	Qua23	Qua24	Qui21	Qui22	Qui23	Qui24	Sex21	Sex22	Sex23	Sex24	
--	--	--	--	--	--	--	--	--	--	Antonio-LOG3-T17//	Antonio-LOG3-T17//	Antonio-LOG3-T17//	Antonio-LOG3-T17//	--	--	--	--	--	--	
Sonia-PROD3-T1E//	Sonia-PROD3-T1E//	Sonia-PROD3-T1E//	Sonia-PROD3-T1E//	João-LOG2-T16//João-LOG2-T17//João-LOG2-T18//	João-LOG2-T16//João-LOG2-T17//João-LOG2-T18//	Maria-FIN4-T14//	Maria-FIN4-T14//	João-LOG2-T16//João-LOG2-T17//João-LOG2-T18//	João-LOG2-T16//João-LOG2-T17//João-LOG2-T18//	Maria-FIN3-T13//	Maria-FIN3-T13//	--	--	Antonio-LOG4-T18//	Antonio-LOG4-T18//	Antonio-LOG4-T18//	Antonio-LOG4-T18//	--	--	
Maria-REL-T10//	Maria-REL-T10//	Maria-REL-T10//	Maria-REL-T10//	Maria-FIN2-T12//	Maria-FIN2-T12//	--	--	Maria-FIN2-T12//	Maria-FIN2-T12//	Sonia-VEN1-T18//	--	--								
Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Maria-REL-T25//	Maria-REL-T25//	Maria-REL-T25//	Maria-REL-T25//
Carlos-RH1-T32//Carlos-RH1-T36//	Carlos-RH1-T32//Carlos-RH1-T36//	Carlos-RH1-T32//Carlos-RH1-T36//	Carlos-RH1-T32//Carlos-RH1-T36//	Carlos-RH2-T36//	Carlos-RH2-T36//	Carlos-RH2-T36//	Carlos-RH2-T36//	--	--	--	--	--	--	--	--	--	--	--	--	
Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	--	--	--	--	
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	

Tabela 19: Programação da melhor solução (divida em 2 partes)

8.3.4.2.1 OBSERVAÇÕES IMPORTANTES

Vale destacar que:

- ◆ Essa execução demorou 2h24min registrando um total de 2.086 soluções viáveis (15 soluções por minuto, em média).
- ◆ Todos os requisitos preenchidos nos cadastros foram respeitados (capacidade da sala, aptidão do instrutor, tempo de deslocamento etc.).

◆ Os custos de viagem de instrutor são justificados pelos seguintes deslocamentos:

- Carlos: 1-3-1
- Sônia: 3-1-3
- Ana: 2-3-2
- Raul: 2-3-2
- Marcos: 1-2-1
- Maria: 1-2-1
 - Lembre-se que o programa conta que no final da programação o instrutor volta para o seu local de origem.
 - O custo de deslocamento do local 3 para 2 é de \$10. Os demais são 50 (inclusive 2 para 3). Total: \$520.

◆ Os custos de utilização da sala são justificados pelos seguintes fatos:

◆ Foram utilizados 156 *slots* em salas pequenas (\$50/*slot*). Total: \$7.800.

◆ Foram utilizados 16 *slots* em salas grandes (\$200/*slot*). Total: \$3.200.

8.4 CONSIDERAÇÕES SOBRE A BT

Caso o leitor tenha notado, o último relatório apresenta duas soluções destacadas. A segunda solução destacada (de número 4) serve para demonstrar o funcionamento da BT. Percebe-se por meio do relatório que antes da BT a preferência dos alunos (funcionários) era de 440 e passou para 448. Esta melhora pode ser explicada pela troca de horário entre duas turmas conforme visto nas figuras subsequentes.

Seg21	Seg22	Seg23	Seg24	Ter21	Ter22	Ter23	Ter24	Qua21	Qua22	Qua23	Qua24	Qui21	Qui22	Qui23	Qui24	Sex21	Sex22	Sex23	Sex24
Maria-FIN4-T14//	Maria-FIN4-T14//	Maria-FIN3-T13//	Maria-FIN3-T13//	--	--	--	--	--	--	Antonio-LOG3-T17//	Antonio-LOG3-T17//	Antonio-LOG3-T17//	Antonio-LOG3-T17//	--	--	--	--	--	--
Carlos-RH2-T1C//	Carlos-RH2-T1C//	Carlos-RH2-T1C//	Carlos-RH2-T1C//	Maria-REL-T1D//	Maria-REL-T1D//	Maria-REL-T1D//	Maria-REL-T1D//	Maria-FIN2-T12//	Maria-FIN2-T12//	João-LOG2-T18//	João-LOG2-T18//	Maria-FIN2-T12//	Maria-FIN2-T12//	João-LOG2-T18//	João-LOG2-T18//	Antonio-LOG4-T18//	Antonio-LOG4-T18//	Antonio-LOG4-T18//	Antonio-LOG4-T18//
Sonia-PROD3-T1E//	Sonia-PROD3-T1E//	Sonia-PROD3-T1E//	Sonia-PROD3-T1E//	João-LOG2-T16//João-LOG2-T17//	João-LOG2-T16//João-LOG2-T17//	--	--	João-LOG2-T16//João-LOG2-T17//	João-LOG2-T16//João-LOG2-T17//	Sonia-VEN1-T1B//	--	--							
Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Maria-REL-T25//	Maria-REL-T25//	Maria-REL-T25//	Maria-REL-T25//							
Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Carlos-RH2-T36//	Carlos-RH2-T36//	Carlos-RH2-T36//	Carlos-RH2-T36//
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Tabela 20: Programação da solução de número 4 antes da BT

Seg21	Seg22	Seg23	Seg24	Ter21	Ter22	Ter23	Ter24	Qua21	Qua22	Qua23	Qua24	Qui21	Qui22	Qui23	Qui24	Sex21	Sex22	Sex23	Sex24
Maria-FIN4-T14//	Maria-FIN4-T14//	Maria-FIN3-T13//	Maria-FIN3-T13//	--	--	--	--	--	--	Antonio-LOG3-T17//	Antonio-LOG3-T17//	Antonio-LOG3-T17//	Antonio-LOG3-T17//	--	--	--	--	--	--
Carlos-RH2-T1C//	Carlos-RH2-T1C//	Carlos-RH2-T1C//	Carlos-RH2-T1C//	Maria-REL-T1D//	Maria-REL-T1D//	Maria-REL-T1D//	Maria-REL-T1D//	Maria-FIN2-T12//	Maria-FIN2-T12//	João-LOG2-T16//	João-LOG2-T16//	Maria-FIN2-T12//	Maria-FIN2-T12//	João-LOG2-T16//	João-LOG2-T16//	Antonio-LOG4-T18//	Antonio-LOG4-T18//	Antonio-LOG4-T18//	Antonio-LOG4-T18//
Sonia-PROD3-T1E//	Sonia-PROD3-T1E//	Sonia-PROD3-T1E//	Sonia-PROD3-T1E//	João-LOG2-T18//João-LOG2-T17//	João-LOG2-T18//João-LOG2-T17//	--	--	João-LOG2-T18//João-LOG2-T17//	João-LOG2-T18//João-LOG2-T17//	Sonia-VEN1-T1B//	--	--							
Marcos-PROD1-T23//Marcos-PROD1-T24//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Marcos-PROD2-T23//	Maria-REL-T25//	Maria-REL-T25//	Maria-REL-T25//	Maria-REL-T25//							
Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Ana-FIN1-T33//	Ana-FIN1-T33//	Raul-LOG1-T34//	Raul-LOG1-T34//	Carlos-RH2-T36//	Carlos-RH2-T36//	Carlos-RH2-T36//	Carlos-RH2-T36//
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Tabela 21: Programação da solução de número 4 após a BT

Portanto, conforme destacado em amarelo, os 8 pontos de melhora na preferência das turmas são resultado da troca de horário que houve entre a turma T16 e a turma T18. Olhando-se no ANEXO C, percebe-se que a turma T16 prefere o horário da tarde enquanto a T18 prefere de manhã. Logo, a troca agradou às duas turmas.

8.4.1 DESEMPENHO DA BT

O leitor pode ter considerado o desempenho da BT insuficiente. Entretanto, antes de fazer qualquer julgamento é importante refletir sobre alguns pontos:

◆ O problema deste Estudo de Caso não é favorável à BT. O movimento para gerar vizinhas feito por esta última lida justamente com a combinação turma-matéria. A BT tenta trocar turmas diferentes, mas que estão assistindo a mesma matéria, de horário. Entretanto, neste Estudo de Caso, quando há a mesma matéria para diferentes turmas, essas podem ser unidas para uma melhor utilização da capacidade da classe. A união mencionada ocorreu diversas vezes na programação gerada. Tendo isto em mente, a BT ficou praticamente sem vizinhas, pois trocar de horário turmas diferentes para a mesma matéria dado que elas estão no mesmo horário e ainda por cima na mesma sala não leva a lugar nenhum.

◆ O Estudo de Caso conta com instrutores que viajam de um lugar para outro assim como regras de precedência. A BT desenvolvida neste trabalho tenta sempre fazer trocas que não desestabilizem o programa já gerado. Portanto, fazer trocas respeitando viagem de instrutor, regra de precedência etc. é extremamente complicado. Se o leitor analisar a troca da solução de número 4 mostrada, observará que o movimento feito não prejudicou em nada o programa gerado assim como não alterou o custo total.

8.5 COMPARAÇÃO ENTRE AS SOLUÇÕES DO ESTUDO DE CASO

A tabela seguinte sintetiza o desempenho apresentado pelas 3 soluções geradas (Sol1 = solução da execução 1, assim por diante).

	custoUsarSala	peso	viagemProf	peso	fit	prefTurma	fitpref	TempoTotal	Custo igual
Sol1	11000	1	620	5000	3111000	352	704	7779	1
Sol2	11200	1	520	5000	2611200	352	704	6514	1
Sol3	11000	1	520	5000	2611000	446	798	8677	2

Tabela 22: Comparação entre soluções

Esta tabela é um resumo do que já foi visto com o acréscimo de uma coluna (a última) que expressa quantas soluções foram encontradas com exatamente os mesmos parâmetros de custo. Destaca-se, porém, que apesar destas soluções

apresentarem o mesmo custo em todos os quesitos, a programação não é igual (são soluções diferentes para o mesmo problema). A preferência após a BT não foi levada em conta, pois somente a terceira execução do *software* utilizou-a.

Percebe-se que a solução vencedora é a terceira por apresentar um custo total de 2.611.000. Portanto, a melhor solução, como o bom-senso indica, deve apresentar os seguintes fatores:

- ◆ Dado que um instrutor viajou, fazer com que ele execute todas as tarefas no destino para não ter que voltar futuramente o que acarretaria custos desnecessários.
- ◆ Juntar todas as turmas sempre que possível para maximizar a utilização das salas e, conseqüentemente, incorrer um custo de utilização de sala desnecessário.

Esses dois fatores são em grande parte os responsáveis pela diferença de custo entre as soluções analisadas.

8.6 COMPARAÇÃO DA MELHOR SOLUÇÃO DO SOFTWARE COM A SOLUÇÃO MANUAL

Até o momento nada foi dito sobre a solução manual feita para o Estudo de Caso por uma pessoa envolvida com treinamento de empresas. Esta solução pode ser encontrada no ANEXO D.

Analisando a solução manual, percebe-se que a melhor solução desenvolvida pelo *software* é superior, pois ganha no quesito “viagem de instrutor”. A pessoa que fez o TT cometeu o mesmo erro que uma das soluções encontradas pelo *software*, ou seja, exagerou no número de viagens. Assim como a solução 1 (execução 1), esta pessoa fez o instrutor Carlos viajar nesta seqüência 1-3-1-3-1, quando na verdade, um TT que apresente a seqüência 1-3-1 é viável de ser encontrado.

Portanto, o *software* desenvolvido mostrou ser mais eficiente.

8.7 ANÁLISE DA PARAMETRIZAÇÃO

Analisando as três execuções do *software* assim como os relatórios gerados, uma interessante comparação é possível de ser feita. A tabela a seguir foi gerada com as 15 melhores soluções de cada execução.

		Exec1	Exec2	Exec3		
Instrutor	afinidade	4	2	2	8	17,78%
	aleatorio	5	4	4	13	28,89%
	carga	2	2	2	6	13,33%
	revesado	4	7	7	18	40,00%
		15	15	15	45	100,00%
Sala	proximidade	0	1	3	4	8,89%
	aleatorio	7	10	6	23	51,11%
	carga	8	4	6	18	40,00%
		15	15	15	45	100,00%
Tuple	primCritica	12	11	11	34	75,56%
	aleatorio	2	2	3	7	15,56%
	probCritica	1	2	1	4	8,89%
		15	15	15	45	100,00%

Tabela 23: Contagem dos parâmetros para cada execução

Esta tabela mostra quantas vezes as regras das heurísticas construtivas apareceram considerando as 15 melhores soluções de cada execução. A penúltima e a última coluna mostram respectivamente o número total de aparições de cada parâmetro assim com sua porcentagem em relação ao grupo (*Instrutor*, *Sala*, *Tuple*). A percepção de que alguns parâmetros não apareceram muito não significa que eles sejam ruins, apenas não são adequados para o Estudo de Caso em questão. Aliás, o motivo por existirem diversas regras de heurística construtiva foi justamente tentar desenvolver um programa que seja adequado ao maior número possível de casos.

8.7.1 EXECUÇÃO EXTRA

Constatada a observação anterior, resolveu-se executar o *software* mais uma vez utilizando somente os melhores parâmetros.

Prof	afinidade	0
	aleatorio	1
	carga	0
	revesado	1
Sala	proximidade	0
	aleatorio	1
	carga	0
Tuple	primCritica	1
	aleatorio	0
	probCritica	0

Tabela 24: Melhores parâmetros

Portanto, existem duas parametrizações [(aleatório, aleatório, primCrítica), {1}] e [(revesado, aleatório, primCrítica), {2}]. No caso do instrutor, dois parâmetros foram escolhidos, pois foi o único grupo que não apresentou um parâmetro vencedor

com maioria absoluta. O critério de parada foi tempo, 1h para cada parametrização, e foram armazenadas 5 soluções de cada.

8.7.2 RELATÓRIO

Para esta execução somente o relatório será apresentado e não a programação propriamente dita.

num	prof	sala	tuple	TempoMelhor	NumSol	NumSoCompleta	custoAbrirSala	PesoSala	custoUsarSala	peso	custoExc	peso	viagemProf	peso	viagemTurma	peso	fit
1	revesado	aleatorio	primCritica	3979	272	2773	0	1	11000	1	0	500000	520	5000	0	5000	2611000
2	revesado	aleatorio	primCritica	4275	486	2773	0	1	11000	1	0	500000	520	5000	0	5000	2611000
3	revesado	aleatorio	primCritica	4312	517	2773	0	1	11200	1	0	500000	520	5000	0	5000	2611200
4	aleatorio	aleatorio	primCritica	2000	1080	1970	0	1	11000	1	0	500000	620	5000	0	5000	3111000
5	aleatorio	aleatorio	primCritica	1474	799	1970	0	1	11000	1	2	500000	520	5000	0	5000	3611000
6	aleatorio	aleatorio	primCritica	2749	1491	1970	0	1	11000	1	2	500000	520	5000	0	5000	3611000
7	aleatorio	aleatorio	primCritica	3101	1700	1970	0	1	11000	1	2	500000	520	5000	0	5000	3611000
8	revesado	aleatorio	primCritica	4003	285	2773	0	1	11000	1	2	500000	520	5000	0	5000	3611000
9	revesado	aleatorio	primCritica	4091	342	2773	0	1	11000	1	2	500000	520	5000	0	5000	3611000
10	aleatorio	aleatorio	primCritica	681	348	1970	0	1	11000	1	4	500000	520	5000	0	5000	4611000

preprof	peso	prefTurma	peso	fitpref	preprof	peso	prefTurma	peso	Fitpref
352	1	442	1	794	352	1	442	1	794
352	1	444	1	796	352	1	444	1	796
352	1	440	1	792	352	1	448	1	800
352	1	442	1	794	352	1	442	1	794
352	1	442	1	794	352	1	442	1	794
352	1	442	1	794	352	1	442	1	794
352	1	448	1	800	352	1	448	1	800
352	1	448	1	800	352	1	448	1	800
352	1	442	1	794	352	1	442	1	794
352	1	448	1	800	352	1	448	1	800

Tabela 25: Relatório para a execução extra

8.7.3 CONCLUSÃO

Analisando-se todos os relatórios gerados até o momento, percebe-se que o da execução extra é o melhor. Este último relatório mostra que 4 soluções foram encontradas com um horizonte de 2 semanas. As 3 primeiras soluções apresentam um custo total igual ao muito próximo ao melhor obtido com as 3 primeiras execuções do *software*. Concluindo, a execução extra possui um desempenho melhor, mas a diferença não é tão grande, principalmente quando comparado com a terceira execução.

CRÍTICAS E MELHORIAS

9. CRÍTICAS E MELHORIAS

O autor deste trabalho acredita que os principais pontos deficientes deste trabalho são:

- ◆ Nesse trabalho preferiu-se dar prioridade a uma BT que só considerasse a preferência dos funcionários, que neste tipo de problema é mais importante do que a dos instrutores. Entretanto, uma próxima versão do *software* desenvolvido irá englobar uma BT semelhante à apresentada neste trabalho, só que para os instrutores. A idéia será utilizar a BT primeiro para melhorar a preferência dos funcionários e depois passar para o refinamento do horário dos instrutores; seguindo, assim, a prioridade neste tipo de problema (TT para empresas).
 - ◆ O *software* deste trabalho foi desenvolvido em uma linguagem interpretada e não compilada. Este fato faz com que a velocidade (desempenho) não seja tão bom quanto outras linguagens como C++ etc. Um próximo passo que será seguido pelo autor deste trabalho é a codificação do *software* em uma linguagem compilada, mas que também permita uma interface amigável como a mostrada neste trabalho.
 - ◆ As heurísticas construtivas para a construção do TT não param por aqui. Novas heurísticas podem ser acrescentadas ao programa de maneira a tornar o tempo de solução menor ou caso isso não aconteça, oferecer uma gama maior de sorteios que poderão ser úteis dependendo do problema tratado.
 - ◆ Muitas vezes, o problema de TT desenvolvido para empresas conta não só com instrutores, mas também com co-instrutores. Estes, por sua vez, são empregados, principalmente, em aulas práticas junto ao instrutor para ajudar na manipulação dos computadores. O acréscimo de co-instrutores torna o problema e o programa ainda mais complexos, principalmente em questões envolvendo viagens e conflitos de horários.
 - ◆ Outra melhoria proposta seria deixar o *software* decidir quando as turmas devem viajar. Entretanto, o autor deste trabalho acredita que tirar a autonomia do operador de TT em demasia seja uma estratégia arriscada. O autor acredita
-

que o gerador de TT deva servir mais como um auxílio à decisão do que uma caixa-preta.

◆ Finalmente, seria interessante acrescentar uma funcionalidade ao *software* que permitisse fixar certas alocações que o operador de TT tenha achado de boa qualidade. Deste modo, dada uma solução, o operador do TT poderia fixar alguns *tuples* alocados e executar o programa novamente para aquelas que não tenham ficado do agrado do operador.

CONCLUSÕES

10. CONCLUSÕES

O autor deste trabalho crê piamente que este trabalho colaborou com a atualização da literatura acerca de problemas de TT assim como resultou em um *software* que será de grande valia para a empresa estudada (*M & R FONSECA CONSULTORIA EM INFORMÁTICA LTDA*). Também vale destacar que o autor deste trabalho não encontrou nenhum artigo ou referência bibliográfica em que o problema de TT tenha sido tratado da maneira explicitada, ou seja, com uso majoritário de heurísticas construtivas para construção dos *tuples*, assim como no auxílio da alocação dos mesmos. Entretanto, observou-se que para problemas de pequeno e médio portes, tal método gera boas respostas e deveria ser melhor investigado. Finalmente, espera-se que este trabalho evidencie a negligência da literatura com relação aos novos problemas enfrentados pelas empresas inseridas na Era do Conhecimento, como é o caso do problema de *Timetabling*.

ANEXO A

ANEXO A

1. CLASSIFICAÇÃO DAS ABORDAGENS

Santoro (2003) construiu uma classificação dos modelos de solução para o problema de sequenciamento da produção. Conforme já citado neste trabalho, este problema é bastante similar ao de TT. Portanto, essa classificação pode ser usada também neste trabalho com pequenas modificações. Apesar da classificação feita por Santoro (2003) englobar diversos métodos, na figura abaixo serão mostradas somente onde se inserem algumas abordagens que serão detalhadas neste trabalho³⁹.

³⁹ Nem todas as abordagens descritas neste capítulo encontram uma classificação adequada no trabalho feito por Santoro (2003).

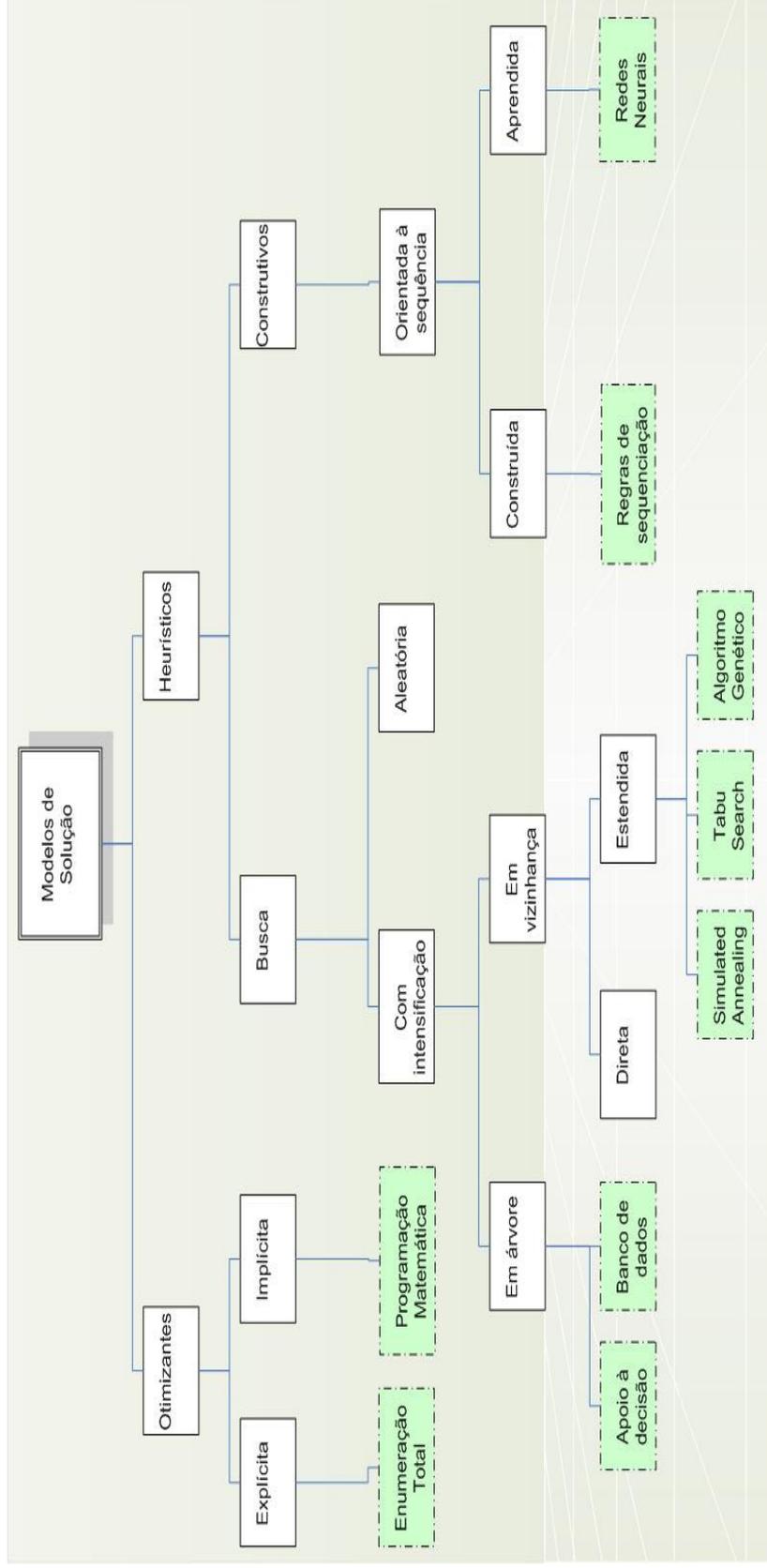


Figura 10: Classificação dos modelos de solução, adaptado de Santoro (2003)

A primeira divisão dos modelos de solução leva em conta o critério de otimalidade da solução, segundo Santoro (2003). Dois grupos resultam desta classificação: otimizantes e os heurísticos.

1.1 OTIMIZANTES

Modelos otimizantes são aqueles que fornecem uma resposta ótima para o problema analisado, ou seja, é a melhor resposta que se pode encontrar, desde que haja uma. Destaca-se que a resposta encontrada é a melhor para a modelagem feita e não necessariamente para o problema real (SANTORO, 2003). Os modelos otimizantes podem ser subclassificados em relação à forma de enumerar as alternativas

1.1.1 ENUMERAÇÃO EXPLÍCITA

Analisa todas as alternativas de solução para saber qual é a melhor. Exemplificando, no caso de 10 ordens esperando para serem processadas em uma máquina, a enumeração explícita iria analisar as 10! (fatorial) combinações possíveis a fim de encontrar a melhor solução. Este método de solução torna-se impraticável no caso de problemas combinatórios muito grandes como é o caso deste trabalho.

1.1.2 ENUMERAÇÃO IMPLÍCITA

Neste caso, as combinações são analisadas utilizando-se uma forma mais inteligente do que a enumeração anterior. Ferramentas como programação linear, mista ou programação dinâmica, técnicas de *branch and bound*, métodos de caminho mais curto e mais longo em redes fazem parte desta classificação. Como exemplo desta inteligência, uma explicação superficial do funcionamento do *branch and bound* é dada a seguir:

Branch-and-bound é um modelo otimizante classificado como de numeração implícita, ou seja, há a probabilidade de que nem todas as alternativas precisem ser analisadas para alcançar a resposta ótima. Exemplificando, caso o problema tratado seja de uma máquina com 10 ordens, tem-se, no total, 10! combinações possíveis de serem analisadas. No entanto, utilizando alguns artifícios, pode-se fazer a numeração implícita o que resultará, em geral, em um universo menor de combinações para ser analisado. Os mecanismos mais comuns são:

◆ *Lower bound* ou limite inferior: solução mais otimista para o nó considerado, dado o nível da árvore em que se encontra.

◆ *Trial solution*: solução completa que permite comparar o valor obtido com os *lower bounds* do outros nós que ainda não foram completamente analisados.

Algoritmo *best first search*

Esse algoritmo procura ir descendo os níveis da árvore do *branch-and-bound* aos poucos. A filosofia deste algoritmo é escolher sempre o nó que apresentar a solução mais promissora independente do nível em que ele se encontra. Portanto, esse algoritmo vai analisando a árvore na horizontal e sempre que possível passa para o nível seguinte. No entanto, caso o nível seguinte não seja bom, nós em níveis mais superficiais voltam a ser analisados. Esse algoritmo garante que assim que se obtiver uma solução no nível mais baixo, ela será ótima.

Algoritmo *depth first search*

Esse algoritmo preconiza encontrar uma solução completa (*trial solution*) o mais rápido possível. No entanto, após achá-la o algoritmo tentará provar que ela é ótima ou que há soluções melhores. Para isso, nós que possuem *lower bounds* mais promissores são analisados até que fiquem piores do que a *trial solution* ou se torne a melhor solução atual. Portanto, esse algoritmo tenta sempre chegar após achar a *trial solution* ao nível mais profundo da árvore sempre que esse movimento for vantajoso. O principal problema deste algoritmo apontado por Morton; Pentico (1993) é que se a solução ótima estiver muito longe da solução inicial (*trial solution*), o algoritmo pode gastar muito tempo na parte errada da árvore.

Destaca-se que a programação matemática (programação linear, inteira, mista etc.) normalmente utiliza o *branch and bound* como ferramenta para alcançar a solução ótima. Conforme já foi comentado, mesmo a utilização de métodos de enumeração implícita torna-se impraticável em problemas classificados como NP-*Hard* devido ao número de combinações e variáveis envolvidas.

1.2 MODELOS HEURÍSTICOS

Os modelos heurísticos caracterizam-se por fornecerem uma resposta que não é necessariamente a melhor. Santoro (2003) divide estes modelos em dois grupos –

busca e construtivo - de acordo com um critério que denomina de número de passos. A principal diferença entre eles é a maneira com que acham as soluções (ou alternativas de solução). Enquanto o modelo heurístico construtivo, como diz o próprio nome, vai construindo a solução no decorrer do tempo; os modelos de busca caracterizam-se por partir de uma solução já pronta e a partir dela ir explorando outras alternativas até que algum critério seja satisfeito. Nota-se que a solução inicial da busca (semente) pode ser encontrada utilizando um modelo heurístico construtivo. Portanto, utilizando-se o caso de 10 ordens em uma máquina onde o objetivo fosse minimizar o máximo atraso havia os seguintes casos:

1.2.1 CONSTRUTIVO

Neste caso, uma regra é estabelecida para a escolha das ordens. Pode-se imaginar, portanto, que a fila de atividades seja ordenada decrescentemente pelo atraso. Desta maneira, a ordem mais atrasada é a primeira a ser processada.

1.2.2 BUSCA

Neste caso, uma solução inicial é obtida (de maneira aleatória ou usando qualquer outro método) e depois certas perturbações nesta fila são feitas a fim de analisar o que ocorre no objetivo buscado. Essa perturbação pode ser, por exemplo, troca de ordem entre duas ordens adjacentes.

Essas duas classificações podem ser subdividas como será mostrado a seguir.

1.3 HEURÍSTICA CONSTRUTIVA

Na tabela mostrada, a heurística construtiva se subdivide em duas outras categorias: construída e aprendida.

1.3.1 CONSTRUÍDA

Essa classe conta com ferramentas, tais como as regras de sequenciamento, que são usadas para ir construindo a solução a cada passo dado. Como exemplo de regras de sequenciamento para aquele caso de 10 ordens em uma máquina tem-se:

- ◆ EDD (Earliest Due Date) – Essa heurística sequencia as atividades de acordo com a data de entrega. Por esta heurística, os *jobs* com datas mais próximas de entrega são seqüenciados primeiro.

◆ Hodgson - A heurística criada por Hodgson é boa para a minimização do número de *jobs* atrasados. Quando as prioridades dos *jobs* são iguais, essa heurística resulta no sequenciamento ótimo. Os passos seguidos por essa heurística são:

- Ordenar a lista de *jobs* pelo critério EDD a fim de minimizar o máximo atraso.
- Achar o primeiro *job* atrasado.
- Remover o *job* até este ponto que tenha o maior tempo de processamento.
- Repetir os passos 2 e 3 até que não haja mais *jobs* atrasados.

Obviamente, a regra de sequenciamento deve ser escolhida de acordo com o objetivo a ser alcançado.

A heurística construtiva conta com duas grandes vantagens: fácil implementação e rapidez na geração de soluções. A heurística construtiva quando bem aplicada pode gerar ótimos resultados.

1.3.2 APRENDIDA

Conforme será visto nas redes neurais, este tipo de sequenciamento é capaz de aprender com os problemas aos quais já teve contato.

1.4 BUSCA

Os modelos de busca podem ser subdivididos em busca aleatória e busca com intensificação. A primeira, conforme o próprio nome diz, não utiliza nenhum critério para a escolha das soluções que irá analisar. Por exemplo, no caso das 10 ordens uma primeira seqüência é originada (semente). Após isso, trocas de ordem feitas de maneira aleatória são feitas, ou seja, outras soluções são achadas (soluções vizinhas) de maneira que a melhor seja sempre guardada. Na busca com intensificação, alguns critérios de seleção são utilizados para escolher a próxima solução a ser analisada. Desta maneira a procura por uma boa solução não é totalmente “às cegas” como é o caso da busca aleatória.

1.4.1 BUSCA EM VIZINHANÇA

Busca em vizinhança é um modelo heurístico que procura a partir de uma solução inicial explorar outras alternativas de maneira a sempre guardar a melhor. Segundo Morton; Pentico (1993), os elementos básicos de uma busca em vizinhança são:

- Uma solução inicial para o problema de interesse – a semente original
- Todas as soluções próximas à solução original – as vizinhas da semente original
- Um método para selecionar a nova semente (solução melhorada) – critério de seleção
- Um método para terminar o procedimento – critério de término

Para Morton; Pentico (1993) o procedimento geral de uma busca em vizinhança envolve os seguintes passos:

- Obter solução semente (se for o começo do problema essa semente é chamada de semente inicial ou original) usando critério de proximidade em relação ao ótimo ou intensificação/diversificação.
- Se o critério de parada é alcançado então parar, caso contrário continuar.
- Escolha a vizinha considerando o *tradeoff* entre qualidade e tempo de computacional.
- Escolha estratégia para avaliar as soluções possíveis.
- Após avaliar todas ou uma parte das soluções vizinhas, escolha uma ou mais soluções baseadas em proximidade/intensificação/diversificação.
- Volte para o passo 1.

Segundo os mesmos autores, a semente original pode ser obtida de diversas maneiras. Um método comum seria escolher essa semente de maneira aleatória. Entretanto a utilização de uma heurística condizente com o objetivo medido pode ser um melhor critério para determinar essa solução inicial. Os autores dizem que tendo uma semente boa a chance de se chegar à solução ótima é maior.

Destaca-se que o termo “vizinha” é extremamente relativo, ou seja, para qualquer problema de busca a definição do que é uma solução vizinha é muito importante. Dependendo dessa definição, a cada iteração teremos uma pequena ou enorme quantidade de vizinhas. Exemplificando:

- Troca de pares adjacentes: por essa regra teremos a cada iteração $(n-1)$ vizinhas, sendo “ n ” o número de *jobs* considerados na seqüência. Portanto, partindo da solução inicial (1,2,3) teremos as seguintes vizinhas (1,3,2) e (2,1,3).
- Troca generalizada de pares: por essa regra chega-se a $n(n-1)/2$ vizinhas a cada iteração. Portanto, partindo da solução inicial (1,2,3) tem-se as seguintes vizinhas (1,3,2);(2,1,3) e (3,2,1).
- Última operação em primeiro: por essa regra há $(n-1)$ vizinhas. Exemplo: partindo da solução (1,2,3) tem-se (3,1,2) e (2,3,1).
- Ordem com maior prioridade na frente: por essa regra alcança-se cerca de $[n(n-1)/2]$ vizinhas.
- Troca “K-way”: por essa regra pode-se atingir uma quantidade de n^k vizinhas a cada iteração.

Segundo Atkinson (1999), nas heurísticas de busca, a maneira com que serão escolhidas as soluções vizinhas afeta diretamente o desempenho computacional, pois para cada vizinha deve-se simular o problema objetivando a obtenção do valor da função objetivo. O autor também enfatiza que o próprio procedimento de geração de vizinhas pode comprometer o desempenho, pois sua implementação pode ser muito trabalhosa. Neste sentido, Morton; Pentico (1993) dizem que para problemas grandes, a geração de vizinhas é tão custosa que muitas vezes vale a pena desenvolver métodos para eliminar vizinhas ruins ou de calcular aproximadamente cada vizinha gerada. Atkinson (1999) conclui que os parâmetros das heurísticas de busca devem ser escolhidos criteriosamente de maneira a chegar na melhor relação entre tempo computacional consumido e qualidade da solução obtida.

1.4.2 BUSCA ESTENDIDA VS BUSCA DIRETA

A busca em vizinhança simples (ou direta) apesar de ter um grande número de vantagens como facilidade de programação, rapidez e boa acurácia dependendo do

tipo de problema, pode ser chamada de limitada. A busca simples tem movimento de intensificação apenas, ou seja, ela procura sempre alternativas melhores e porventura pode acabar ficando presa em um ótimo local. Essa busca só resultará na resposta ótima se houver muita sorte ou, como Morton; Pentico (1993) acreditam, se a semente inicial partir de uma heurística boa para o problema em questão. Como exemplo da busca direta tem-se a busca local (*local search*).

Já as buscas estendidas incorporam o que é chamado de diversificação, ou seja, soluções piores do que a atual podem ser aceitas. Esse aparente contra-senso explica-se na medida em que para sair de um ótimo local e explorar novas regiões, soluções que resultam em um valor desfavorável para função objetivo devem ser aceitas. No rol das buscas estendidas encontram-se a BT (objeto de estudo deste projeto), algoritmo genético e o *simulated annealing* dentre outras.

Conforme já foi dito, há diversas maneiras de gerar a semente inicial para os modelos de busca. Essas maneiras se subdividem em duas grandes classificações:

- 1) Probabilística - envolve o uso de sorteio que pode ser de duas maneiras:
 - a) Estratificação sem viés: cada *job* apresenta a mesma probabilidade de ser escolhido. Exemplificando: se houver 24 *jobs*, cada um terá a probabilidade de $p=1/24$ de ser sorteado. Por não ter critério algum, essa semente pode ser extremamente pobre. No entanto, pensando na diversificação ela pode apresentar vantagens, ou seja, iniciar a exploração de novas regiões.
 - b) Estratificação com viés: cada *job* apresenta uma determinada probabilidade de ser sorteado.

Este tópico teve como objetivo mostrar ao leitor a ampla variedade de abordagens para resolver o problema de TT. Entretanto, só foi passada uma visão superficial delas assim como uma idéia de como podem ser classificadas. No próximo tópico será feito um maior detalhamento do funcionamento e características das abordagens mais promissoras para o problema de TT.

1.5 DETALHAMENTO DE MÉTODOS DE SOLUÇÃO

Nesta seção algumas abordagens para a solução do problema de TT serão explicadas com mais detalhe.

1.5.1 METAHEURÍSTICAS

1.5.1.1 TABU SEARCH OU BUSCA TABU

Segundo Glover; Laguna (1997), a filosofia da BT é investigar e explorar os princípios de resolução de problemas de maneira inteligente. Os autores dizem que esse modelo de busca estendida é baseado em conceitos que unem os campos de otimização e inteligência artificial. Segundo Barnes; Laguna (1993 *apud* Glover et al; 1993) a BT foi totalmente descrita em 1996 como uma estratégia para resolver problemas de difícil otimização. Já em 1988, a BT já era apontada, junto como o *simulated annealing* e o algoritmo genético, como um modelo promissor para problemas práticos. Pode-se dizer que essa previsão realizou-se devido à grande utilização desses modelos atualmente. Segundo Woodruff, Zemel (1993 *apud* GLOVER ET AL., 1993), a BT provou ser um método muito efetivo para resolver problemas difíceis.

Destaca-se que a BT muitas vezes é caracterizada como uma metaheurística (ou meta-heurística). Segundo Glover; Laguna (1997), essa denominação vem do fato de que inicialmente o conceito de heurística era de ter uma inteligência ou regra iterativa que parasse assim que fosse achada uma solução que não pudesse ser melhorada (um provável ótimo local). Apesar dos autores acharem que esse conceito prevalece, eles enfatizam que atualmente muitos dos métodos heurísticos não são mais puramente locais. Entretanto, como a BT conta com elementos que a permitem sair de um ótimo local, ela foi denominada metaheurística. Segundo Moscato (1992 *apud* GLOVER ET AL., 1993), a BT assim como o *simulated annealing* compartilham das mesmas vantagens que recaem na popularidade dos dois métodos. Além disso, são metaheurísticas com lógicas que objetivam fugir de ótimo locais.

Glover; Laguna (1997) dizem que as metaheurísticas podem ser classificadas de acordo com três critérios principais:

- O uso de memória adaptativa, ou seja, uma memória baseada em atributos.
 - Usa (código A) ou não usa memória (código M)
- O tipo de exploração de vizinhança usado

- Explora a vizinhança de acordo com algum critério (N) ou utiliza-se de aleatoriedade (S).
- O número de soluções correntes carregadas de uma iteração para outra
 - Carrega somente uma (1) ou há uma população de tamanho P (código P)

De acordo com os códigos dados, os autores apresentam a seguinte classificação:

Meta-heurística	Classificação 1	Classificação 2
<i>Algoritmo Genético</i>	M/S/P	M/N/P
<i>Simulated annealing</i>	M/S/1	M/N/1
<i>Tabu Search</i>	A/N/1	A/N/P

Tabela 26: Classificação das principais metaheurísticas (GLOVER; LAGUNA, 1997)

Glover; Laguna (1997) explicam que há duas classificações porque a primeira é condizente com “conceitos populares”, enquanto que a segunda representa a visão de uma minoria de pesquisadores. Destaca-se que a segunda classificação da BT é condizente com os primeiros *papers* que incorporaram o uso de população como uma estratégia da BT de explorar boas soluções.

Ressalta-se pela classificação que uma ou a principal característica da BT é a utilização de memória adaptativa. Esse fator é um dos grandes responsáveis por fazer com que a BT apresente movimentos de intensificação e diversificação. Grosso modo, movimentos de intensificação olham somente a função objetivo, ou seja, somente as vizinhas que são melhores que a solução atual são aceitas (busca o ótimo local). No entanto, movimentos de diversificação muitas vezes aceitam soluções piores para que outras regiões sejam pesquisadas, o que pode resultar em ganhos (outros ótimos locais melhores ou o ótimo global). Esses movimentos de diversificação levam às seguintes observações feitas por Morton; Pentico (1993):

- Sempre que uma solução melhor é encontrada ela é salva, assim nenhuma informação relevante é perdida com a diversificação.

- O maior trabalho da diversificação é manter atualizada a lista dos últimos “m” (Glover propõe $m=7$) movimentos e também checar se os novos movimentos são “Tabus” ou não. Movimentos “tabus” são movimentos já feitos e que por isso não devem ser repetidos (podem prender a busca em um ótimo local) a menos que valha a pena, ou seja, obedeça ao critério de aspiração. Este critério pode ser, por exemplo, “aceite movimentos tabus desde que a solução obtida seja melhor do que a melhor solução encontrada até o momento”.
- A BT não pára mesmo se o ótimo global for encontrado, deve haver algum mecanismo de parada como número de iterações ou tempo de processamento.

Destaca-se que se em um determinado momento todos os movimentos forem tabus, o programa pode usar algum critério de “desempate”: critério de aspiração ou escolher a nova solução ao acaso. O critério de aspiração pode ser usado também quando há um ou mais movimentos tabus coexistindo com um ou mais movimentos não-tabus. Neste caso, um critério de aspiração válido seria escolher algum movimento tabu caso o valor da função objetivo obtido por ele seja o menor até o momento (caso minimização). Visando um maior esclarecimento de como funciona a BT, um exemplo simples é descrito a seguir. Neste exemplo, têm-se quatro atividades (1, 2, 3, 4) a serem seqüenciadas em uma máquina. O objetivo é minimizar a função objetivo, seja ela qual for. Abaixo seguem algumas explicações para o exemplo dado:

- ◆ Função objetivo: esta função deverá ser minimizada. Para cada seqüência, o valor da função objetivo é dado entre parênteses.
 - ◆ Semente ou solução inicial: Essa solução pode ser obtida de maneira aleatória, por regra de sequenciamento etc., ou seja, é o ponto de partida da busca.
 - ◆ Atributo de movimento: movimento que é necessário se fazer com a solução corrente para se chegar na candidata à próxima solução.
 - ◆ Lista Tabu: lista dos movimentos proibidos desde que não atendam ao critério de aspiração. Neste exemplo o tamanho da lista tabu é 4, ou seja, ela guarda no máximo 4 movimentos proibidos. Ao atingir o número
-

máximo, a lista tabu segue a regra PEPS⁴⁰ (primeiro a entrar, primeiro a sair).

- ◆ Critério de aspiração: se um movimento estiver na lista tabu, mas se o resultado de sua aplicação levar a obtenção de uma resposta melhor do que a atual, então ele poderá ser executado.
- ◆ Soluções vizinhas: Entende-se por soluções vizinhas, soluções obtidas através da troca da ordem de atividades adjacentes.
- ◆ Critério de parada: atingir o final do sexto passo.

Exemplo	Passo 1	Passo 2	Passo 3	Passo 4	Passo 5	Passo 6
Solução Corrente	2341	3241	3214	3124	1324	1234
Melhor solução até o momento	2341(32)	3241(10)	3214(9)	3214(9)	3214(9)	1234(6)
Lista tabu	[]	[2 troca com 3]	[4 com 1; 2 com 3]	[2 com 1; 4 com 1; 2 com 3]	[3 com 1; 2 com 1; 4 com 1; 2 com 3]	[3 com 2; 3 com 1; 2 com 1; 4 com 1]
soluções vizinhas	3241(10) 2314(31) 2431(32)	3214(9) 2341(32) 3421(26)	3241(10) 2314(31) 3124(37)	3214(9) 3142(53) 1324(44)	3124(37) 1234(6) 1342(60)	2134(28) 1324(44) 1243(7)
Candidata à próxima solução	3241	3214	3241	3214	3214	1243
Atributo de movimento	[3 troca com 2]	[1 troca com 4]	[4 com 1]	[2 com 1]	[2 com 3]	[4 com 3]
Movimento é tabu ?	não	não	sim	sim	sim	não
Atende ao critério de aspiração ?	-	-	não (10 >= 9)	não (9 >= 9)	sim (6 < 9)	-
Próxima solução corrente	3241	3214	passo não completado	passo não completado	1234	1243
Critério de parada atingido?	não	não	não	não	não	sim
Solução Corrente			3214	3124		
Melhor solução até o momento			3214(9)	3214(9)		
Lista tabu			[4 com 1; 2 com 3]	[2 com 1; 4 com 1; 2 com 3]		
soluções vizinhas			3241(10) 2314(31) 3124(37)	3214(9) 3142(53) 1324(44)		
Candidata à próxima solução			2314	1324		
Atributo de movimento			[2 com 3]	[1 com 3]		
Movimento é tabu ?			sim	não		
Atende ao critério de aspiração ?			não (31 >= 9)	-		
Próxima solução corrente			passo não completado	1324		
Solução Corrente			3214			
Melhor solução até o momento			3214(9)			
Lista tabu			[4 com 1; 2 com 3]			
soluções vizinhas			3241(10) 2314(31) 3124(37)			
Candidata à próxima solução			3124			
Atributo de movimento			[1 com 2]			
Movimento é tabu ?			não			
Atende ao critério de aspiração ?						
Próxima solução corrente			3124			

Figura 11: Exemplo de busca Tabu , elaborada pelo autor

1.5.1.2 SIMULATED ANNEALING

Esse modelo de busca estendida tem uma origem bem diferente da BT. O *simulated annealing* foi concebido a partir de idéias aplicadas ao processo de

⁴⁰ O autor declara que esta sigla não busca fazer menção a nenhum produto comumente no mercado. Em inglês, a sigla ficaria FIFO (*first in, first out*).

Dj	0	5	10	15	20	25	27	29
K	10							
Pj	8,75651E-27	1,93E-22	4,25E-18	9,36E-14	2,06E-09	4,54E-05	0,002479	0,135335

Tabela 27: Influência do coeficiente de temperatura

k	0,5							
Dbest	30							
Dj	0	5	10	15	20	25	27	29
K	200							
Pj	0,927	0,939	0,9512	0,963	0,975	0,987	0,992	0,997

k	0,5							
Dbest	30							
Dj	0	5	10	15	20	25	27	29
K	50							
Pj	0,740	0,778	0,818	0,860	0,904	0,951	0,970	0,990

k	0,5							
Dbest	30							
Dj	0	5	10	15	20	25	27	29
K	10							
Pj	0,223	0,286	0,367	0,472	0,606	0,778	0,860	0,951

Tabela 28: Influência da temperatura

Destaca-se que à medida que o algoritmo do *simulated annealing* vai buscando novas soluções, a temperatura e/ou o coeficiente também podem mudar de valor dependendo da estratégia que se queira adotar (diversificação ou intensificação). Nota-se pelas tabelas apresentadas que quanto menor a temperatura (menor coeficiente) maior é a diversificação, ou seja, a probabilidade de aceitar soluções piores é grande. A esta estratégia de mudar o valor da temperatura é chamada de *cooling schedule* (programação de resfriamento), conforme aponta Abramson; Dang; Krisnamoorthy (1999). Elmohamed; Coddington; Fox (1997) dizem que os resultados obtidos dependem muito do *cooling schedule* adotado. Existem diversas estratégias de resfriamento sendo que o mais popular é o resfriamento geométrico. Neste caso a próxima temperatura é dada pela equação $T_{k+1} = \alpha \times T_k$, sendo α um parâmetro constante cujo valor típico, segundo Abramson; Dang; Krisnamoorthy (1999), é entre 0,90 e 0,99.

A grande diferença entre o *simulated annealing* e a BT é que na primeira a diversificação se dá por sorteio, enquanto que na segunda a diversificação ocorre predominantemente por meio da imposição (Lista Tabu). Destaca-se também que os fatores do *simulated annealing* podem ser dinâmicos, ou seja, podem variar com o decorrer do tempo. Portanto, uma estratégia seria colocar um k alto no começo (privilegiar a intensificação) e depois ir diminuindo (diversificação).

Elmohamed; Coddington; Fox (1997) colocam como vantagens do *simulated annealing* a facilidade de implementação, a aplicabilidade do algoritmo para quase todos os problemas combinatórios, a robustez e a facilidade de combiná-lo com outras técnicas. Entretanto, o autor diz que a utilização do *Simulated Annealing* para a obtenção de bons resultados requer uma análise cuidadosa dos parâmetros como a definição de vizinhança e a taxa de resfriamento.

1.5.1.3 ALGORITMO GENÉTICO

Segundo Glibovets; Medvid (2003), o algoritmo genético é um método de sucesso para solucionar problemas combinatórios. Seu princípio deu-se em 1975 com J. Holland da Universidade de Michigan. Morton; Pentico (1993) dizem que o algoritmo genético pode ser considerado como uma generalização dos dois últimos métodos apresentados. A lógica utilizada pelo algoritmo genético é muito semelhante ao princípio da seleção natural proposto por Charles Darwin em 1859. Basicamente, segundo CMU (2005), o algoritmo genético começa com uma população inicial onde cada indivíduo⁴¹ apresenta um grau de adaptação (fitness) que representa o quão bom ele é. Esta medida baseia-se na quantidade de conflitos que o TT apresenta e dará origem à probabilidade que este indivíduo terá em se reproduzir e passar seus genes adiante. A seguir as etapas do algoritmo genético são explicadas com mais detalhes conforme colocado por Glibovets; Medvid (2003):

1.5.1.3.1 INICIALIZAÇÃO

Uma população individual é gerada para que o algoritmo genético possa iniciar-se. Esta população inicial pode ser obtida de maneira totalmente aleatória ou por heurísticas específicas para o problema. A população é composta de cromossomos

⁴¹ Cada indivíduo representa uma solução completa para o problema e é chamado, na literatura, de cromossomo.

(indivíduos) que carregam a informação de uma solução para o problema. Cada cromossomo é composto de diversos genes que carregam a menor informação do problema envolvido (combinação sala de aula-professor-turma, por exemplo).

1.5.1.3.2 SIMULAÇÃO DO PROCESSO EVOLUTIVO

Esta fase é a mais importante, pois nela o processo evolutivo é aplicado à população inicial. Hereditariedade (*crossover*), mutação e seleção estarão presentes nesta etapa do algoritmo:

1) O operador de mutação é aplicado em cada um dos genes de todos os cromossomos da primeira geração. Essa mutação irá ocorrer ou não segundo uma probabilidade estipulada no início do problema que poderá ou não mudar no tempo. Glibovets; Medvid (2003) afirmam que a mutação desempenha um papel-chave na natureza, no entanto deve ser aplicada rigorosamente para não estragar soluções boas, ou seja, causar um caos na população. Badica; Lalescu (2005) colocam que a mutação serve para introduzir uma mudança aleatória na população atual de maneira a prevenir que ela se concentre em somente um lugar no espaço de soluções (fique preso em algum ótimo local). A mutação pode ser, por exemplo, a troca aleatória de dois genes em um cromossomo.

2) *Crossover* ou cruzamento de dois ou mais pais ocorrem para gerar a segunda geração. Quanto maior a adaptação do indivíduo da primeira geração, maior será sua probabilidade de poder se reproduzir. O processo de *crossover* consiste em pegar dois ou mais pais e estabelecer pontos de corte; em seguida, os genes resultantes são recombinados para formar a próxima geração. Segundo Enzhe; Sung (2002), o mecanismo de reprodução pode resultar em TTs inaceitáveis. Portanto, a conferência de restrições do tipo *hard* e sua eventual reparação tornam-se necessárias em cada um dos indivíduos gerados. Às vezes também há necessidade de fazer esta reparação após a mutação.

3) O critério de seleção é aplicado. Glibovets; Medvid (2003) colocam que duas formas de seleção podem acontecer. No primeiro tipo, quando a segunda geração chegar no número de indivíduos igual ao da primeira (N, por exemplo) eles irão substituir completamente os indivíduos da primeira geração (tipo *generation*). Já o

segundo tipo, chamado *steady state reproduction* junta a primeira e a segunda geração e os indivíduos menos adaptados são descartados.

4) Finalmente, há o encerramento do algoritmo genético que pode ser por tempo de processamento, número de iterações, valor de adaptação alcançado etc.

Segundo CMU (2005), além do critério de seleção, dois outros parâmetros tem importância crucial do algoritmo genético: tamanho da população e taxa de mutação. Segundo o autor, se o tamanho da população é muito pequeno então a variabilidade é baixa; entretanto, se a população é muito grande a variabilidade é tão alta que soluções realmente boas terão menor possibilidade de serem escolhidas. Em relação à mutação, escolher uma taxa muito alta pode ser desfavorável, pois pode neutralizar o fato dos melhores pais terem sido escolhidos. No limite, com uma taxa de mutação muito alta, a formação de novos indivíduos equivaleria a criar uma geração não a partir de reprodução, mas aleatoriamente.

Enzhe; Sung (2002) apontam três itens como vantagens do algoritmo genético: não envolve matemática complicada, apresenta grande eficiência na procura por soluções e pode ser facilmente integrada com heurísticas. Sua principal desvantagem em comparação com as outras metaheurísticas é o tempo de processamento.

1.5.1.4 PROCURA LOCAL

A *Local Search* ou *Hill Climbing*⁴² é uma metaheurística extremamente simples, pois não conta com nenhum mecanismo para escapar de ótimos locais caso encontre um. Normalmente, em problemas de TT, a procura local é usada com outras metaheurísticas para auxiliar o movimento de intensificação. A seguir os passos que esta metaheurística segue são apresentados (ARNTZEN, H.; LOKKETANGEN, A. (2005)) (MERLOT ET AL., 2002).

- 1) Começar com uma solução s ⁴³.
- 2) Analisar as soluções s' nas vizinhanças $N(s)$ de s . Caso todas as vizinhas sejam piores que a solução s , parar.

⁴² O termo *Hill Climbing* é apropriado para problemas onde se quer maximizar a função objetivo, ou seja, o objetivo é achar o maior (“mais alto”) valor possível. Entretanto este termo também é usado para problemas de minimização devido à sua consagração.

- 3) Mover para a melhor solução s' pertencente a $N(s)$.
- 4) Faça $s = s'$.
- 5) Verificar critério de parada: número de iterações máximo, por exemplo. Se este for atingido parar. Caso contrário voltar para passo 2.

1.5.2 CLP

O CLP ou *Constraint Logic Programming* é a descrição de um tipo de linguagem desenvolvida para facilitar a formulação de problemas de TT. Assim como JAVA, essa linguagem também apresenta Classes o que a torna mais poderosa. Segundo Henz; Wurtz (1996), ela combina a flexibilidade de abordagens ligadas à inteligência artificial com a eficiência dos algoritmos da pesquisa operacional. O autor diz que esta linguagem, em muitos problemas reais, é competitiva com outras abordagens. O poder do CLP foi mostrado com linguagens tais como CHIP, Prolog III ou CLP(R). Segundo Frangouli; Harmandas (1995), o CLP é frequentemente chamado de CLP(X), onde X significa o conjunto sobre o qual as restrições do problema são formuladas. Há linguagens onde X representa os números reais, números racionais, valores binários etc. A mais popular para problemas de TT é o CLP(FD) onde FD significa Domínio Finito (*Finite Domain*). Esta linguagem é caracterizada por facilitar a representação de problemas de TT via restrições que envolvem variáveis que assumam um número finito de valores.

A lógica do CLP pode ser encontrada em CLIP LAB (2005) e IC PARC (2005), mas basicamente, a resolução do problema ocorre com a análise das restrições e do domínio (valores) que as variáveis podem assumir. A cada iteração tenta-se diminuir o tamanho do domínio das variáveis, ou seja, o espaço de soluções vai sendo cortado até se chegar em uma solução viável caso ela exista. O método de solução do CLP pode ser considerado como pertencente à classe de “enumeração implícita”. Atualmente há diversas linguagens que se utilizam desta lógica como a ECLiPSe desenvolvida pelo ECRC que é um centro de pesquisa de computação industrial europeu.

⁴³ Solução semente que pode ter sido originada aleatoriamente ou através de alguma heurística ou metaheurística.

1.5.3 DSS (DECISION SUPORT SYSTEMS - SISTEMAS DE APOIO À DECISÃO)

Alguns autores clamam por uma ferramenta que seja mais genérica e tenha uma maior flexibilidade do que métodos totalmente computadorizados. Kolski; Piechowiak (2004) dizem que a solução deve ser focada no usuário e não no problema ou no algoritmo usado como muitas abordagens preconizam. O autor também critica as soluções totalmente computadorizadas pelos seguintes motivos: requer computadores poderosos, são mais adequadas para problemas de colégios devido ao menor número de variáveis, não são eficientes quando há um TT super-restringido, ou seja, não há uma solução viável e, finalmente, precisam que todas as restrições estejam formuladas sendo que isto nem sempre é tão fácil de fazer.

Ferland; Fleurent (1994) dizem que os DSSs normalmente não incluem uma ferramenta de otimização por trás. Entretanto, o autor acredita que o DSS pode ser muito mais útil caso haja um procedimento que gere respostas de boa qualidade. O autor acredita que um bom DSS deve ter 5 categorias de funções: manipulação de dados, otimização automática, otimização interativa, geração de relatórios e configuração de parâmetros.

Portanto, pode-se imaginar uma ferramenta que tenha um mecanismo de busca que pode ser interrompido a qualquer momento pelo usuário. Após a interrupção o usuário pode editar a solução com o auxílio do computador. Exemplificando, caso o usuário queira mover uma turma de horário, o computador pode mostrar quais horários são mais favoráveis (não causam conflitos). Outra grande vantagem é em relação aos problemas que não possuem uma solução livre de conflitos, neste caso o usuário experiente, pode adequar a solução da melhor maneira possível. O desenvolvimento de um DSS também auxilia o usuário a lidar com mudanças de “última hora”, neste caso há a possibilidade de edição sem ter que rodar o processo de otimização novamente. A geração de relatórios também é bastante interessante, pois permite mostrar aos usuários todos os conflitos envolvidos assim como também já mostrar um cronograma detalhado do TT.

1.5.4 RELAXAÇÃO LAGRANGIANA

Apesar de não consistir em uma abordagem totalmente separada (normalmente está associada à programação linear), o conceito da relaxação Lagrangiana é bastante

usado em TT, seja na abordagem de programação matemática (linear, binária, mista etc.) ou em metaheurísticas.

No caso da programação matemática, já foi dito que pela complexidade do problema (NP-*Hard*) dificilmente esta abordagem é usada. Entretanto, alguns autores como Tripathy (1984), dentre outros, tentam resolver o problema através da relaxação. O conceito por trás da relaxação é permitir que o problema tenha maior liberdade e assim fique mais fácil encontrar uma solução. A relaxação significa tirar restrições do problema e alocá-las na função objetivo, deste modo a restrição torna-se uma condição não-obrigatória mas que o programa tentará satisfazer caso isso seja possível. Um exemplo dado por Fisher (1985 *apud* MORTON; PENTICO (1993)) é mostrado abaixo:

$$\begin{array}{ll}
 \min cx \text{ (função objetivo)} & \min cx - u(b - Ax) \text{ (função objetivo)} \\
 \text{sujeito a :} & \text{sujeito a :} \\
 Ax \leq b & Dx \leq e \\
 Dx \leq e & x \geq 0 \text{ e INTEIRO} \\
 x \geq 0 \text{ e INTEIRO} &
 \end{array}$$

Figura 12: Exemplo de relaxação Lagrangiana (MORTON; PENTICO, 1993)

Neste caso, a primeira restrição foi retirada e acrescentada à função objetivo. Nota-se que a restrição dizia que “ $b - Ax$ ” deve ser maior que zero. Com a penalidade proposta na função objetivo (- u) esta condição acaba sendo estimulada, pois se “ $b - Ax$ ” for negativo esse fato irá onerar a função objetivo que é de minimização. A maior diferença é que agora, apesar de ser desfavorável, esta condição poderá ocorrer, entretanto a solução do problema torna-se mais fácil.

A principal função da relaxação é dar um limite inferior para a solução do problema original, ou seja, a resposta da solução do problema relaxado certamente é melhor do que a solução do problema original (caso ela exista). Outra utilização é o uso da solução Lagrangiana como a solução do problema original após algumas modificações. No entanto, destaca-se que a relaxação não é algo trivial de ser feito, pois se deve tomar bastante cuidado com a restrição que será eliminada assim como

as penalidades que serão colocadas, pois uma relaxação mal feita pode tornar o problema tão difícil de ser resolvido quanto o problema original.

O autor deste trabalho acredita que o conceito da relação Lagrangiana é muito semelhante ao conceito de restrições *soft* e *hard*, pois as restrições *soft* nada mais são do que restrições que foram incorporadas à função objetivo conforme já foi observado.

1.5.5 MÉTODOS HÍBRIDOS

Os métodos híbridos nada mais são do que a junção de duas ou mais abordagens. Estes métodos são bastante encontrados na literatura de TT principalmente porque permite aproveitar-se das vantagens de cada método especificamente. Como exemplo de métodos híbridos pode-se ter heurística com métodos de busca, programação binária (com relaxação Lagrangiana) com métodos de busca, heurística com programação matemática etc.

ANEXO B

ANEXO B

O exemplo dado neste anexo é bem simples e didático, pois seu intuito é demonstrar algumas das funcionalidades do *software* desenvolvido. Os parâmetros principais do exemplo são:

- ◆ 9 funcionários organizados em 3 turmas (3 funcionários/turma).
- ◆ 3 matérias com relação de precedência entre elas.
- ◆ 3 professores.

Nos tópicos subseqüentes, os cadastros necessários para o *software* “rodar” serão apresentados.

1. CADASTROS

1.1 CADASTRO PRINCIPAL

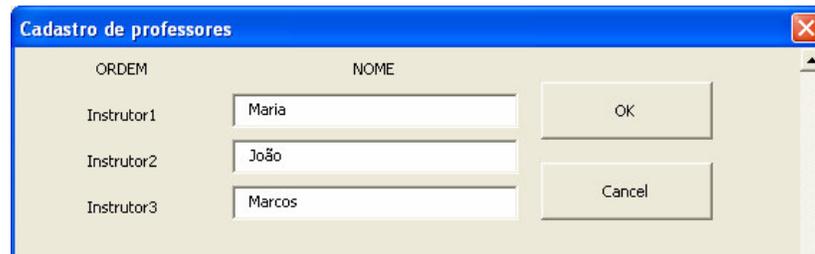
O cadastro principal visa informar o programa sobre algumas quantidades básicas, conforme pode ser visto na figura abaixo.

Parâmetro	Valor
Número de slots/dia	4
Número de dias/semana	5
Horizonte máximo (dias)	5
Número de instrutores	3
Número de alunos	9
Número de matérias	3
Número de salas	3
Número de turmas	3
Número de tipos/sala	1
Número de locais	1

Figura 13: Cadastro principal do programa

1.2 CADASTRO DOS INSTRUTORES

Este cadastro permite ao usuário do TT configurar os nomes dos instrutores que serão utilizados. Abaixo estão os 3 instrutores do exemplo.



A janela "Cadastro de professores" apresenta uma interface com uma barra de título azul e um botão de fechar no canto superior direito. O conteúdo principal é dividido em duas colunas: "ORDEM" e "NOME".

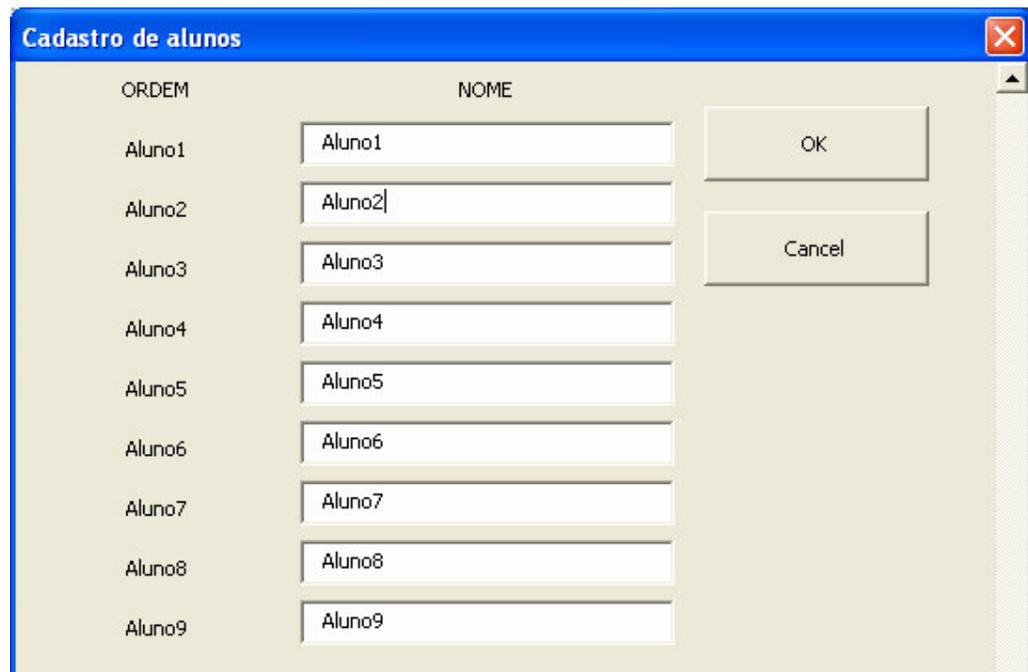
ORDEM	NOME
Instrutor1	Maria
Instrutor2	João
Instrutor3	Marcos

À direita do formulário, há dois botões: "OK" e "Cancel".

Figura 14: Cadastro dos instrutores

1.3 CADASTRO DOS ALUNOS

Neste cadastro é possível colocar o nome de cada aluno. Neste exemplo, preferiu-se deixar a sugestão de nome fornecida pelo programa.



A janela "Cadastro de alunos" possui uma barra de título azul e um botão de fechar no canto superior direito. O formulário é organizado em duas colunas: "ORDEM" e "NOME".

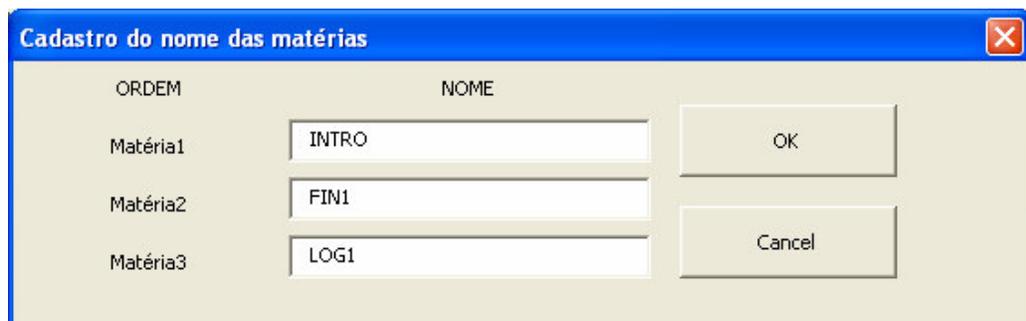
ORDEM	NOME
Aluno1	Aluno1
Aluno2	Aluno2
Aluno3	Aluno3
Aluno4	Aluno4
Aluno5	Aluno5
Aluno6	Aluno6
Aluno7	Aluno7
Aluno8	Aluno8
Aluno9	Aluno9

Do lado direito, há dois botões: "OK" e "Cancel".

Figura 15: Cadastro de alunos/funcionários.

1.4 CADASTRO DA MATÉRIAS

Neste cadastro, por sua vez, é possível colocar o nome real das matérias que serão utilizadas na programação de treinamentos.



ORDEM	NOME
Matéria1	INTRO
Matéria2	FIN1
Matéria3	LOG1

Figura 16: Cadastro das matérias

1.5 CADASTRO DAS SALAS

Muitas empresas dão nomes às suas salas como, por exemplo, sala Ruy Barbosa. Este cadastro serve então para personalizar os nomes das salas. Neste exemplo, preferiu-se deixar a sugestão dada pelo próprio programa.



ORDEM	NOME
Sala1	Sala1
Sala2	Sala2
Sala3	Sala3

Figura 17: Cadastro das salas

1.6 CADASTRO DAS TURMAS

Assim como os anteriores, as turmas também podem receber um nome mais adequado que o sugerido pelo programa. Neste exemplo, utilizou-se os nomes fornecidos pelo programa.

ORDEM	NOME
Turma1	<input type="text" value="Turma1"/>
Turma2	<input type="text" value="Turma2"/>
Turma3	<input type="text" value="Turma3"/>

OK
Cancel

Figura 18: Cadastro das turmas

1.7 CADASTRO DOS TIPOS DE SALA

Conforme já mencionado, as salas podem ser de vários tipos, ou seja, apresentar retroprojeter e computador, apresentar computador e quadro-negro ou computador e quadro-branco ou não ter computador etc. Neste exemplo, assume-se que todas as sala são do mesmo tipo, conforme o cadastro mostrado abaixo.

ORDEM	NOME
TipoSala1	<input type="text" value="TipoSala1"/>

OK
Cancel

Figura 19: Cadastro de tipos de sala

1.8 CADASTRO DE LOCALIDADES

As salas, os instrutores e os alunos/funcionários podem estar em diferentes localidades. Este cadastro serve para informar o programa sobre as localidades utilizadas. O nome dado poderia ser do bairro, cidade, município etc.; entretanto, neste simples exemplo, aceitou-se deixar o nome dado pelo programa.

ORDEM	NOME
Local1	Local1

Figura 20: Cadastro de localidades

1.9 CADASTRO DE DISPONIBILIDADE DO INSTRUTOR

Este cadastro serve para indicar ao programa em que horários os instrutores poderão ou preferem ser alocados.

Cadastrar Disponibilidade dos Instrutores

	Seg11	Seg12	Seg13	Seg14	Ter11	Ter12	Ter13	Ter14	Qua11
Maria	1	1	1	1	1	1	1	1	1
João	1	1	1	1	1	1	1	1	1
Marcos	1	1	1	1	1	1	1	1	1

Cadastrar Disponibilidade dos Instrutores

	Qua12	Qua13	Qua14	Qui11	Qui12	Qui13	Qui14	Sex11	Sex12	Sex13	Sex14
	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1

Figura 21: Cadastro de disponibilidade de horários dos instrutores

Portanto, neste simples exemplo os instrutores podem ser alocados em qualquer horário com a mesma preferência. Observa-se que a preferência precisa ser indicada por *slot*, como no cadastro principal foi indicado que cada dia possui 4 *slots* então tem-se que um dia como segunda-feira será subdividido em “Seg11”, “Seg12”, “Seg13” e “Seg14”. O significado do primeiro número é a semana, ou seja, trata-se da segunda-feira da primeira semana, já o segundo número indica qual *slot* está sendo referido. Como este exemplo está limitado a uma semana, o primeiro número de todos os dias será 1.

1.10 CADASTRO DE DISPONIBILIDADE DE HORÁRIO DAS SALAS

The figure consists of two screenshots of a software window titled "Cadastro de disponibilidade de horário das salas".

The top screenshot shows the first part of the grid. It has a title bar with "OK" and "Cancel" buttons. The grid has columns for Seg11, Seg12, Seg13, Seg14, Ter11, Ter12, Ter13, Ter14, and Qua11. The rows are Sala1, Sala2, and Sala3. The values in the cells are:

	Seg11	Seg12	Seg13	Seg14	Ter11	Ter12	Ter13	Ter14	Qua11
Sala1	1	1	1	1	1	1	1	1	1
Sala2	1	1	1	1	1	1	1	1	1
Sala3	0	0	1	1	1	1	1	1	1

The bottom screenshot shows the second part of the grid. It has a title bar with a close button. The grid has columns for Qua12, Qua13, Qua14, Qui11, Qui12, Qui13, Qui14, Sex11, Sex12, Sex13, and Sex14. The rows are Sala1, Sala2, and Sala3. The values in the cells are:

	Qua12	Qua13	Qua14	Qui11	Qui12	Qui13	Qui14	Sex11	Sex12	Sex13	Sex14
Sala1	1	1	1	1	1	1	1	1	1	1	1
Sala2	1	1	1	1	1	1	1	1	1	1	1
Sala3	1	1	1	1	1	1	1	1	1	1	1

Figura 22: Cadastro de disponibilidade das salas

Nota-se neste exemplo que no primeiro meio-período da segunda-feira a sala3 está indisponível.

1.11 CADASTRO DE DISPONIBILIDADE DOS ALUNOS

Abaixo segue a figura mostrando o cadastro da disponibilidade dos alunos.

Cancel	Seg11	Seg12	Seg13	Seg14	Ter11	Ter12	Ter13	Ter14	Qua11	Qu
Aluno1	2	2	2	2	1	1	1	1	1	
Aluno2	2	2	2	2	1	1	1	1	1	
Aluno3	2	2	2	2	1	1	1	1	1	
Aluno4	1	1	1	1	1	1	1	1	1	
Aluno5	1	1	1	1	1	1	1	1	1	
Aluno6	1	1	1	1	1	1	1	1	1	
Aluno7	1	1	1	1	1	1	1	1	1	

Figura 23: Disponibilidade dos alunos/funcionários

Destaca-se que não há nenhuma indisponibilidade, mas os alunos da turma 1 preferem ter aula na segunda-feira quando comparado com os outros dias.

1.12 CADASTRO DA RELAÇÃO INSTRUTOR-MATÉRIA

Dado que há três instrutores e três matérias, o operador do TT deve indicar para o programa qual instrutor dá qual aula. Nota-se, porém, que um instrutor pode dar mais de uma aula, mas é, na verdade, especialista em somente uma delas. Esta questão pode ser indicada no cadastro por meio de uma escala (0 = não ensina a matéria; 1 = ensina a matéria, mas não é a especialidade; 2 = especialidade).

A figura a seguir mostra esta diferença de escala.

	INTRO	FIN1	LOG1
Maria	1	1	1
João	1	2	2
Marcos	1	0	2

Figura 24: Cadastro da relação instrutor-matéria

Portanto, pode-se dizer que a instrutora Maria é uma generalista, enquanto os outros instrutores são aptos a dar a matéria introdutória e possuem uma ou mais especialidades.

1.13 CADASTRO DO LOCAL DOS INSTRUTORES

Conforme já foi mencionado, o instrutor pode ter um local original e um local de simulação, ou seja, local que o programa realmente considerará. Esta diferença de local é feita para que em cada TT gerado pelo programa o custo de deslocamento possa ser calculado. Neste exemplo, não haverá diferenciação entre local original e local de simulação já que foi dito no cadastro principal que só há um local.

	Local Sim	Local Origem
Maria	Local1	Local1
João	Local1	Local1
Marcos	Local1	Local1

Figura 25: Cadastro do local do instrutor

1.14 CADASTRO DA REGRA DE PRECEDÊNCIA ENTRE MATÉRIAS

Pode ocorrer de algumas matérias dependerem de outras para serem ensinadas. Neste cadastro, deve-se indicar para cada matéria qual a sua sucessora imediata.

	INTRO	FIN1	LOG1
INTRO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
FIN1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LOG1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 26: Cadastro da relação de precedência

Portanto, conforme pode-se observar na figura anterior, a matéria INTRO tem como sucessoras imediatas as matérias FIN1 e LOG1, logo estas matérias jamais poderão ser ensinadas sem que antes as turmas em questão não tenham assistido à matéria INTRO.

1.15 CADASTRO DAS INFORMAÇÕES SOBRE MATÉRIA

A figura seguinte mostra as informações necessárias para completar os dados das matérias.

	Duração	Agrupamento	Tipo	TipoSala	SalaExclusiva	Juntar?	Prof
INTRO	4		seguida	TipoSala1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Maria
FIN1	4	2	mesmoSlot	TipoSala1	<input type="checkbox"/>	<input type="checkbox"/>	
LOG1	2	2	encaixar	TipoSala1	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 27: Cadastro das informações sobre as matérias

Tomando-se com exemplo a matéria INTRO, observa-se que sua duração total é 4 *slots* e que ela deverá ser alocada de maneira seguida, por este motivo não foi especificado nenhum agrupamento para ela. Nota-se também que o tipo de sala necessário para ensinar esta matéria deve ser o TipoSala1 ou superior. Caso o TipoSala1 fosse o único necessário, ou seja, tipos superiores não deveriam ser escolhidos, então o *flag* SalaExclusiva deveria estar marcado. Finalmente, o cadastro indica que a matéria INTRO deverá ser obrigatoriamente ensinada pela instrutora Maria.

1.16 CADASTRO DA RELAÇÃO ALUNO-TURMA

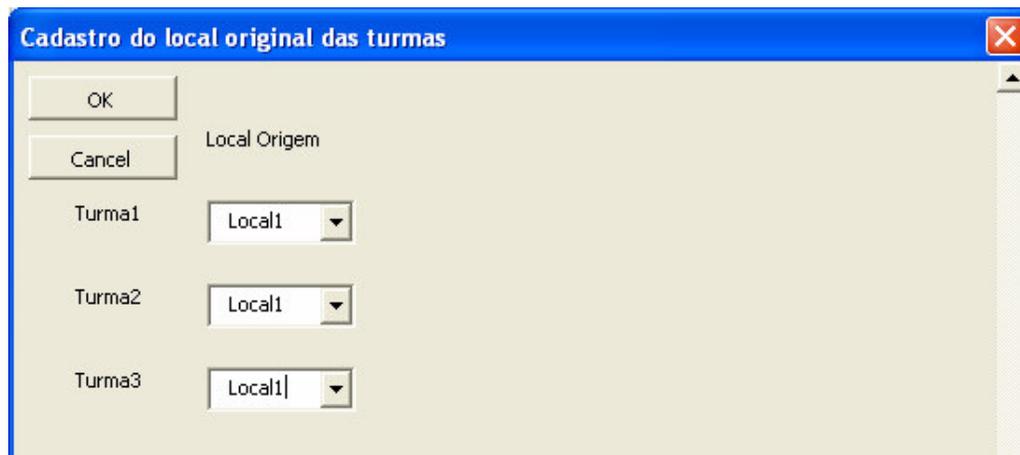
Conforme já foi colocado, o exemplo em questão conta com 9 alunos divididos igualmente entre 3 turmas. Esta lógica é fornecida neste cadastro e pode ser observada na próxima figura.

	Turma1	Turma2	Turma3
Aluno1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aluno2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aluno3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aluno4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Aluno5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Aluno6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Aluno7	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Aluno8	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 28: Cadastro da relação aluno-turma

1.17 CADASTRO DO LOCAL ORIGEM DAS TURMAS

Assim como os instrutores, também deve ser cadastrado um local origem para as turmas. O intuito deste cadastro é o cálculo das viagens que, porventura, as turmas possam realizar.



Turma	Local Origem
Turma1	Local1
Turma2	Local1
Turma3	Local1

Figura 29: Cadastro do local origem das turmas

1.18 CADASTRO DA RELAÇÃO TURMA-MATÉRIA

Até o momento, não foi especificado quais matérias uma determinada turma deverá assistir. O cadastro seguinte serve justamente para indicar esta relação.



Turma	INTRO	FIN1	LOG1
Turma1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Turma2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Turma3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 30: Cadastro da relação turma-matéria

1.19 CADASTRO DOS LOCAIS DE SIMULAÇÃO DAS TURMAS

Este cadastro não faz muito sentido neste exemplo, pois só um local existe. Entretanto, este local de simulação serve para indicar onde uma turma assistirá uma

determinada matéria. Portanto, o operador do TT pode fazer com uma turma viaje caso ache esta estratégia vantajosa. Imagine que o local de origem de uma turma seja muito próximo do local onde fica a matriz da empresa e que naquele local não haja nenhum instrutor apto a dar uma determinada matéria, neste caso ou algum instrutor deverá viajar ou a turma inteira deverá se deslocar. Esta última opção deverá ser indicada neste cadastro. Destaca-se que em ambas as situações (viagem da turma ou viagem do instrutor) o valor da função objetivo será aumentado (custo).

	INTRO	FIN1	LOG1
Turma1	Local1	Local1	Local1
Turma2	Local1	Local1	Local1
Turma3	Local1	Local1	Local1

Figura 31: Cadastro do local de simulação das turmas

Sabe-se que a Turma1 deve assistir à matéria LOG1 e que a matéria LOG1 pode ser ensinada pelos 3 professores sendo que dois deles são especialistas. Imagine que a Turma1 só queira assistir a matéria LOG1 com especialistas; portanto, ou com João ou com Marcos. Explicando melhor, LOG1 tem um conjunto C de professores que estão aptos a dar esta matéria; neste cadastro, então, pode-se restringir este conjunto C em um subconjunto S unitário ou não com os professores que realmente deverão ensinar uma determinada matéria para uma turma.

Outra utilidade deste questionário é forçar com que instrutores viajem para dar uma matéria para determinada turma. Conforme apontado no tópico anterior, há casos em que ou a turma deverá viajar ou o instrutor deverá se deslocar para suprir a carência de instrutores em determinados locais. Portanto, se há uma turma-matéria-local sem nenhum instrutor, neste cadastro pode-se fornecer um conjunto de instrutores (unitário ou não) que podem viajar para o local em questão. Caso este

conjunto não seja unitário, então a probabilidade de cada instrutor ser sorteado estará viesada de acordo com sua distância do local de origem em relação ao local de destino.

	INTRO	FIN1	LOG1
Turma1	<input type="text"/>	<input type="text"/>	João;Marcos
Turma2	<input type="text"/>	<input type="text"/>	<input type="text"/>
Turma3	<input type="text"/>	<input type="text"/>	<input type="text"/>

Figura 32: Cadastro da relação turma-matéria-instrutor

Neste exemplo, portanto, o conjunto de professores que podem ensinar a matéria LOG1 é {Maria; João; Marcos}. Entretanto, neste caso este conjunto foi limitado para {João;Marcos} somente para uma determinada turma. Logo, se outra turma também assistisse a matéria LOG1 e nenhuma limitação fosse feita, a Maria seria uma instrutora como probabilidade diferente de zero de ser escolhida.

Deve-se destacar a diferença entre este cadastro e o cadastro onde as informações sobre as matérias são fornecidas. O leitor atento deve-se lembrar de que no cadastro das informações da matéria INTRO, Maria foi colocada como instrutora. Isto significa que, independente da turma, sempre que aparecer a matéria INTRO, ela será ensinada pela instrutora Maria. Caso essa exclusividade de instrutor fosse somente para uma determinada turma, então o cadastro apresentado neste tópico deveria ser utilizado.

1.20 CADASTRO DE UNIÃO ENTRE TURMAS

Da mesma maneira que o cadastro anterior, este cadastro sofre interferência das opções colocadas sobre as matérias, mas especificamente no *flag* que indica união. Caso o leitor se lembre, o *flag* correspondente a esta opção na matéria INTRO foi marcado. Isto significa que sempre que a capacidade da sala permitir, diferentes

turmas serão unidas para assistir a essa matéria. Entretanto, casos em que algumas turmas não devem ser unidas podem ocorrer. Imagine, por exemplo, que há três turmas e que uma delas é composta somente por diretores enquanto as outras são funcionários menos abastados na hierarquia da empresa. Pode ser que esta turma ilustre não deva ser unida com as outras. Neste caso, o *flag* localizado no cadastro das matérias não deve ser ativado, mas sim o cadastro mostrado a seguir, pois ele permite dizer quais turmas poderão ser unidas.

	INTRO	FIN1	LOG1
Turma1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Turma2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Turma3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 33: Cadastro da relação de união entre turmas

As duas matérias que possuem mais de uma turma assistindo são INTRO e FIN1. No cadastro da matéria INTRO o *flag* correspondente a união já está marcado, portanto tanto faz a opção que é colocada no cadastro deste item, pois o *flag* da matéria é dominante. Já a matéria FIN1 não tem o *flag* de união marcado e optou-se neste cadastro não juntar as turmas 1 e 2 (*flag* neste cadastro não está ativo).

1.21 CADASTRO DAS INFORMAÇÕES SOBRE SALAS

Neste cadastro, questões como capacidade, tipo, local e custos da sala são acrescentados, conforme é observado na figura abaixo.

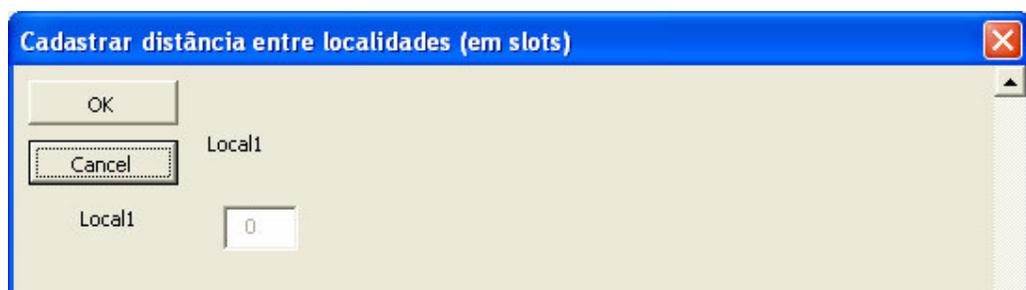


	Capacidade	Tipo	Local	Custo/Hora	CustoAbrir
Sala1	6	TipoSala1	Local1	50	10
Sala2	6	TipoSala1	Local1	40	10
Sala3	6	TipoSala1	Local1	50	10

Figura 34: Cadastro das informações sobre salas

1.22 CADASTRO DA DISTÂNCIA ENTRE LOCALIDADES

Este cadastro não será utilizado para este exemplo, já que há somente uma localidade. O próprio programa não permite mudança no valor apresentado (valor 0). Nota-se que a distância é dada em número de *slots*, ou seja, se para o operador cada *slot* corresponde a uma hora, então a distância será dada em horas. Entretanto, há um caso especial de distância (valor -1). Quanto este valor for digitado como distância entre duas localidades, o programa interpretará que sempre que houver viagem entre um lugar e outro, a disponibilidade da entidade que viajou será sempre o dia seguinte. Exemplificando melhor, imagine que o operador de *timetabling* tenha colocado a distância entre a localidade 1 e 2 como sendo -1. Agora, imagine que um instrutor irá viajar do local 1 para o local 2, e que a última atividade do instrutor no local 1 tenha sido em uma segunda-feira de manhã. Neste caso, o instrutor só poderá ensinar alguma matéria no local 2 a partir de terça-feira. Esta conclusão também seria válida caso a última atividade do instrutor no local 1 tivesse sido na segunda-feira à tarde, pois como foi dito, a distância -1 significa disponível no dia seguinte.



Cadastrar distância entre localidades (em slots)

OK

Cancel

Local1

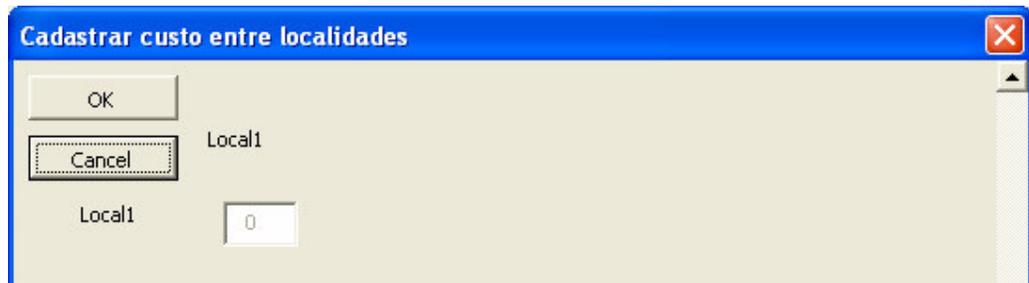
Local1

0

Figura 35: Cadastro de distância entre localidades

1.23 CADASTRO DO CUSTO ENTRE LOCALIDADES

Outro cadastro que não será útil neste exemplo é o de custo entre localidades.

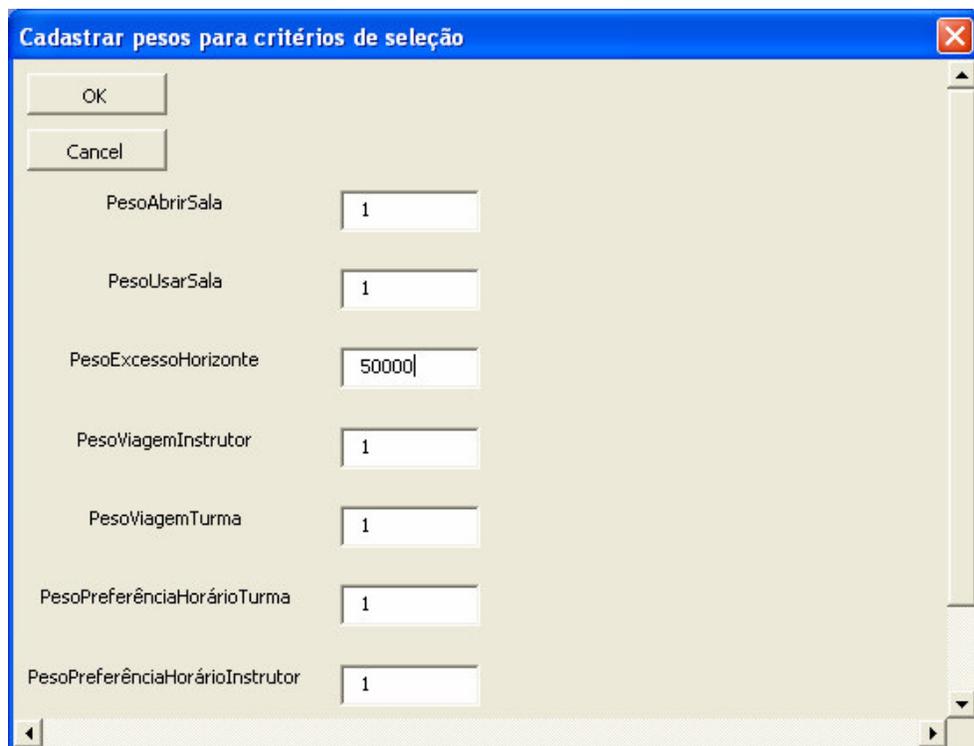


The dialog box titled "Cadastrar custo entre localidades" features a blue title bar with a close button (X) in the top right corner. Below the title bar, there are two buttons: "OK" and "Cancel". To the right of the "Cancel" button is the text label "Local1". Below this, there is another "Local1" label followed by a text input field containing the number "0".

Figura 36: Cadastro de custo entre localidades

1.24 CADASTRO DOS PESOS DA FUNÇÃO OBJETIVO

Esse cadastro fornece ao programa informações sobre qual é a importância relativa dos critérios utilizados. Neste exemplo, assim como no estudo de caso, o objetivo maior é a minimização do horizonte do programa. Este objetivo explica a diferença de pesos da figura abaixo.



The dialog box titled "Cadastrar pesos para critérios de seleção" has a blue title bar with a close button (X) in the top right corner. It contains two buttons: "OK" and "Cancel". Below these are seven rows, each with a label and a text input field:

Label	Value
PesoAbrirSala	1
PesoUsarSala	1
PesoExcessoHorizonte	50000
PesoViagemInstrutor	1
PesoViagemTurma	1
PesoPreferênciaHorárioTurma	1
PesoPreferênciaHorárioInstrutor	1

Figura 37: Importância relativa entre os critérios

2. PARÂMETROS DE SORTEIO

Conforme já foi mencionado, as heurísticas construtivas possuem algumas opções de sorteio. As alternativas escolhidas para este exemplo foram:

Prof	afinidade	1
	aleatorio	1
	carga	0
	revesado	0
Sala	proximidade	1
	aleatorio	0
	carga	0
Tuple	primCritica	0
	aleatorio	0
	probCritica	1

Figura 38: Parâmetros de sorteio para o exemplo

Portanto, fazendo-se as combinações, o exemplo contará com duas parametrizações ($\{1\}$ = afinidade, proximidade, probCrítica; $\{2\}$ =aleatório, proximidade, probCrítica).

3. CRITÉRIOS DE PARADA

Para o exemplo em questão, somente um critério de parada foi adotado: tempo por parametrização igual a 10s. Portanto, o programa irá ser executado em 20s, no total.

4. HORIZONTE DESEJADO

No intuito de medir o desempenho das respostas obtidas, dado que o principal peso é para o horizonte de programação, um horizonte desejado de 3 dias foi escolhido.

5. SOLUÇÃO DO EXEMPLO

Conforme já foi comentado, duas parametrizações ($\{1\}$ = afinidade, proximidade, probCrítica; $\{2\}$ =aleatório, proximidade, probCrítica) foram escolhidas para a solução deste simples exemplo. Também foi configurado no programa para que somente a melhor solução de cada parâmetro fosse mostrada, portanto, duas soluções no total. A BT também foi acionada para tentar melhorar a preferência dos funcionários. As respostas obtidas foram:

1	Seg11	Seg12	Seg13	Seg14	Ter11	Ter12	Ter13	Ter14	Qua11	Qua12	Qua13	Qua14
Sala1	--	--	--	--	Maria-INTRO-T3//	Maria-INTRO-T3//	Maria-INTRO-T3//	Maria-INTRO-T3//	--	--	--	--
Sala2	Maria-INTRO-T1//Maria-INTRO-T2//	Maria-INTRO-T1//Maria-INTRO-T2//	Maria-INTRO-T1//Maria-INTRO-T2//	Maria-INTRO-T1//Maria-INTRO-T2//	João-FIN1-T2//	João-FIN1-T2//	João-FIN1-T1//	João-FIN1-T1//	João-FIN1-T2//	João-FIN1-T2//	João-FIN1-T1//	João-FIN1-T1//
Sala3	--	--	--	--	Marcos-LOG1-T1//	Marcos-LOG1-T1//	--	--	--	--	--	--

2	Seg11	Seg12	Seg13	Seg14	Ter11	Ter12	Ter13	Ter14	Qua11	Qua12	Qua13	Qua14
Sala1	--	--	--	--	Maria-INTRO-T3//	Maria-INTRO-T3//	Maria-INTRO-T3//	Maria-INTRO-T3//	--	--	--	--
Sala2	Maria-INTRO-T1//Maria-INTRO-T2//	Maria-INTRO-T1//Maria-INTRO-T2//	Maria-INTRO-T1//Maria-INTRO-T2//	Maria-INTRO-T1//Maria-INTRO-T2//	--	--	Marcos-LOG1-T1//	Marcos-LOG1-T1//	--	--	--	--
Sala3	--	--	--	--	João-FIN1-T1//	João-FIN1-T1//	João-FIN1-T2//	João-FIN1-T2//	João-FIN1-T1//	João-FIN1-T1//	João-FIN1-T2//	João-FIN1-T2//

Figura 39: Soluções obtidas para o exemplo em questão

O programa ordena as soluções por ordem crescente de custo, portanto, ou essas duas soluções apresentam o mesmo custo ou a primeira é melhor do que a segunda solução exposta. Antes de esclarecer esta dúvida, é importante notar que, quanto ao quesito principal (horizonte de programação), as duas soluções são iguais, pois utilizam 3 dias exatamente. Pode-se perceber também que todas as condições cadastradas foram respeitadas:

- ◆ A sala3 não foi utilizada no primeiro meio-período da segunda-feira, momento que estava indisponível.
- ◆ Os instrutores alocados às matérias estão corretos. Vale lembrar que havia uma restrição especial para a combinação turma1-LOG1. Esta matéria poderia ter sido ensinada por todos os professores, entretanto conforme cadastrado, o operador do TT quis que somente especialistas fossem escolhidos (João ou Marcos), o que foi respeitado. No caso da matéria FIN1, só a Maria e o João poderiam ministrá-las; em ambas as soluções o João foi escolhido. Nota-se que isto foi necessário, pois se a Maria fosse escolhida o horizonte de programação teria que necessariamente ser maior, pois, conforme cadastrado, somente ela poderia dar a matéria INTRO.

- ◆ Percebe-se que a regra de precedência foi respeitada na medida em que as matérias FIN1 e LOG1 para uma determinada turma foram alocadas em instantes de tempo superiores aos da matéria INTRO1.
- ◆ As durações total, diária assim como o tipo de alocação (seguida, mesmoSlot, encaixar) estão corretos em ambas as respostas. Destaca-se que nas duas soluções houve junção de turmas para assistir a matéria INTRO. A terceira turma só não foi unida, pois a capacidade da sala (6 pessoas) não permitiria que isto fosse feito (cada turma tem 3 pessoas).

Apesar dessas observações, ainda resta a dúvida se as soluções são iguais em relação ao custo ou há alguma diferença. Como o leitor pode perceber, cada solução possui uma numeração no canto superior esquerdo. Essa numeração serve para que o operador do TT possa consultar o relatório de TT gerado pelo programa de forma prática. A seguir é mostrado o relatório, que devido a sua extensão foi dividido em 3 partes.

num	prof	sala	tuple	TempoMelhor	NumSol	NumSolCompleta	custoAbrirSala	PesoSala	custoUsarSala
1	afinidade	proximidade	probCritica	3.00	49	482	30	1	780
2	aleatorio	proximidade	probCritica	20.00	563	654	30	1	840

peso	custoExc	peso	viagemProf	peso	viagemTurma	peso	fit	preprof	peso	prefTurma	peso	fitpref
1	0	50000	0	1	0	1	810	22	1	26	1	48
1	0	50000	0	1	0	1	870	22	1	26	1	48

prefProf	peso	prefTurma	peso	Fitpref
22	1	26	1	48
22	1	26	1	48

Tabela 29: Relatório gerado pelo programa

Nota-se, portanto, que há diferença entre as soluções, pois a primeira apresenta um custo total menor (coluna denominada fit⁴⁴). Como as duas soluções terminaram no horizonte desejado (3 dias, conforme cadastrado), nenhuma delas sofreu a maior penalidade⁴⁵. Entretanto, a primeira solução se destacou por ter utilizado mais vezes a sala mais barata (sala2), razão pela qual seu custo é menor.

A seguir, as demais colunas do relatório são explicadas:

⁴⁴ Alusão à palavra *fitness* que é usada por muitos autores para indicar quão boa é uma solução.

⁴⁵ Cada *slot* que ultrapassasse os três dias desejados contribuiria com 50.000 no custo total.

-
- ◆ Coluna 1: numeração que relaciona o relatório com TT gerado. Este último possui uma identificação no canto superior esquerdo.
 - ◆ Coluna 2: maneira de sortear o professor escolhido.
 - ◆ Coluna 3: maneira de sortear a sala escolhida.
 - ◆ Coluna 4: maneira de selecionar os *tuples*.
 - ◆ Coluna 5: tempo em que a solução em questão foi encontrada. O instante inicial para a medição desse intervalo de tempo é o momento em que o operador do TT acionou o programa.
 - ◆ Colunas 6 e 7: a coluna 7 indica quantas soluções viáveis foram encontradas como a parametrização em questão. Portanto, com a primeira tripla (afinidade-proximidade-probCrítica) foram encontradas 482 soluções com o critério de parada escolhido (10s). Desses 482 TTs completos, o melhor foi o de número 49 (coluna 6). Já em relação à segunda tripla (aleatório-proximidade-probCrítica), observa-se que a melhor solução foi encontrada somente no final (563 de 654). A explicação desta diferença é que, provavelmente, o sorteio do professor de maneira viesada é melhor do que a ausência de critério (aleatório).
 - ◆ Colunas 8 e 9: custo das salas abertas. Como o custo por sala era de 10 e nas duas soluções as três salas foram utilizadas, o custo foi de 30 em ambos os casos. Já a coluna 9 é o peso correspondente (cadastrado) do custo de abrir uma sala.
 - ◆ Colunas 10 e 11: custo de utilização das salas (coluna 10) e peso deste critério (coluna 11). Analisando a primeira solução como exemplo, tem-se que 12 *slots* foram utilizados na sala mais barata (40 por *slot*) e 6 *slots* nas outras salas mais caras (50 por *slots*). Logo, o custo deste quesito é $(12*40 + 6*50) = 780$.
 - ◆ Colunas 12 e 13: correspondem ao custo e peso do quesito “excesso de horizonte”. Conforme explicado, nenhuma das soluções ultrapassou o limite de 3 dias, o que ocasionou um custo zero.
-

- ◆ Colunas 14, 15, 16 e 17: correspondem ao custo e peso de viagem de instrutores e de alunos. Como esse exemplo utilizou somente um local, este custo é zero, em ambos os casos.
- ◆ Coluna 18: esta coluna significa o custo total incorrido pela alternativa de solução em questão. No caso da primeira solução, tem-se o custo de abertura de salas (30) acrescido do custo de utilização de salas (780) resultando em um custo de 810.
- ◆ Colunas 18, 20, 21 e 22: estas colunas apresentam um índice relacionado à aderência do TT gerado em relação às preferências de horário tanto dos professores como das turmas envolvidas. Já foi mencionado, que esta preferência não é tratada na primeira parte do problema, mas sim com a BT. Entretanto, por sorte, a única turma que havia indicado algum tipo de preferência (Turma 1, na segunda-feira) foi alocada no dia certo.
- ◆ Coluna 23: refere-se à soma do índice de preferência de horário dos professores com o das turmas.
- ◆ Colunas 24, 25, 26, 27 e 28 (marrom): estas colunas tem o mesmo significado que as cinco anteriores, entretanto, elas mostram o desempenho da BT em relação à melhora da preferência de horário das turmas⁴⁶. Como a única turma que havia indicado preferência foi alocada no dia correto, a BT não pode melhorar a solução. No entanto, o próximo item irá dar um exemplo de como a BT implementada funcionaria neste caso.

6. FUNCIONAMENTO DA BUSCA TABU

Já foi comentado que o exemplo anterior não utilizou a BT porque a turma 1 preferia ser alocada na segunda-feira e assim o foi nas duas alternativas de solução. Entretanto, imagine que isto não acontecesse, ou seja, a combinação turma1-Intro-Maria não fosse colocada em uma segunda-feira, mas sim em uma terça-feira. O autor deste trabalho “rodou” o *software* mais uma vez para o mesmo exemplo, só que dessa vez o critério de parada foi o número de iterações. Por ter sido executado por

⁴⁶ Conforme já explicado, a preferência dos instrutores será tratada em um outra versão deste mesmo *software*.

menos tempo⁴⁷, a probabilidade de achar uma solução pior é maior. As duas soluções obtidas foram:

1	Seg11	Seg12	Seg13	Seg14	Ter11	Ter12	Ter13	Ter14	Qua11	Qua12	Qua13	Qua14	Qui11	Qui12
Sala1	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Sala2	Maria-INTRO-T1//Maria-INTRO-T2//	Maria-INTRO-T1//Maria-INTRO-T2//	Maria-INTRO-T1//Maria-INTRO-T2//	Maria-INTRO-T1//Maria-INTRO-T2//	Maria-INTRO-T3//	Maria-INTRO-T3//	Maria-INTRO-T3//	Maria-INTRO-T3//	João-FIN1-T1//	João-FIN1-T1//	--	--	João-FIN1-T1//	João-FIN1-T1//
Sala3	--	--	--	--	João-LOG1-T1//	João-LOG1-T1//	--	--	Maria-FIN1-T2//	Maria-FIN1-T2//	--	--	Maria-FIN1-T2//	Maria-FIN1-T2//

2	Seg11	Seg12	Seg13	Seg14	Ter11	Ter12	Ter13	Ter14	Qua11	Qua12	Qua13	Qua14	Qui11	Qui12
Sala1	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Sala2	Maria-INTRO-T2//Maria-INTRO-T3//	Maria-INTRO-T2//Maria-INTRO-T3//	Maria-INTRO-T2//Maria-INTRO-T3//	Maria-INTRO-T2//Maria-INTRO-T3//	João-FIN1-T2//	João-FIN1-T2//	--	--	João-FIN1-T2//	João-FIN1-T2//	João-LOG1-T1//	João-LOG1-T1//	--	--
Sala3	--	--	--	--	Maria-INTRO-T1//	Maria-INTRO-T1//	Maria-INTRO-T1//	Maria-INTRO-T1//	Maria-FIN1-T1//	Maria-FIN1-T1//	--	--	Maria-FIN1-T1//	Maria-FIN1-T1//

Figura 40: Soluções-exemplo para a Busca Tabu

Conforme pode ser notado, a primeira solução obedeceu, por sorte, a preferência da turma 1 de ser alocada na segunda-feira. Entretanto, a segunda alternativa não obteve o mesmo destino glorioso e, por isso, poderá ser melhorada pela BT. Já foi explanado que a BT busca trocar pares turma-matéria, onde a matéria é a mesma dado que a troca não causará nenhum impacto no restante do TT gerado. Neste caso, fica bem claro que a combinação T1-INTRO poderia ser trocada com a combinação T3-INTRO sem consequência alguma para o TT. Esta troca faria com que a turma 1 ficasse mais contente com o novo horário. A melhora proporcionada pela busca tabu pode ser vista no relatório gerado (amarelo = Antes da BT, marrom = Depois da BT):

preprof	peso	prefTurma	peso	fitpref
22	1	26	1	48
22	1	22	1	44

prefProf	peso	prefProf	peso	Fitpref
22	1	26	1	48
22	1	26	1	48

Tabela 30: Relatório gerado (antes e depois da BT)

⁴⁷ Pela precisão do programa (centésimos de segundo) o tempo de execução foi zero.

Nota-se, portanto, que a solução 1 não foi melhorada (e nem poderia), ao contrário da solução 2, onde a troca explicada foi efetuada (vide próxima figura).

2	Seg11	Seg12	Seg13	Seg14	Ter11	Ter12	Ter13	Ter14	Qua11	Qua12	Qua13	Qua14	Qui11	Qui12
Sala1	--	--	--	--	--	--	--	--	--	--	--	--	--	--
Sala2	Maria-INTRO-T2//Maria-INTRO-T1//	Maria-INTRO-T2//Maria-INTRO-T1//	Maria-INTRO-T2//Maria-INTRO-T1//	Maria-INTRO-T2//Maria-INTRO-T1//	João-FIN1-T2//	João-FIN1-T2//	--	--	João-FIN1-T2//	João-FIN1-T2//	João-LOG1-T1//	João-LOG1-T1//	--	--
Sala3	--	--	--	--	Maria-INTRO-T3//	Maria-INTRO-T3//	Maria-INTRO-T3//	Maria-INTRO-T3//	Maria-FIN1-T1//	Maria-FIN1-T1//	--	--	Maria-FIN1-T1//	Maria-FIN1-T1//

Figura 41: Solução (2) onde a BT foi efetiva

ANEXO C

ANEXO C

1. CADASTROS COMUNS ÀS SOLUÇÕES 1,2 E 3

Nesta parte os cadastros comuns às três soluções do Estudo de Caso serão apresentados.

1.1 *RELAÇÃO INSTRUTOR-MATÉRIA*

Abaixo está a relação de qual instrutor pode ensinar qual matéria. Nota-se que não houve diferenciação de graus de conhecimento (0 = não ensina; 1 = ensina; 2 = especialista).

prof/matéria	INTRO	FIN1	LOG1	PROD1	RH1	RH2	VEN1	FIN2	PROD2	LOG2	LOG4	LOG3	REL	FIN3	FIN4	PROD3
Maria	1	1	0	0	0	0	0	1	0	0	0	0	1	1	1	0
João	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
Marcos	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
Carlos	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
Sonia	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1
Ana	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
José	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Antonio	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
Raul	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabela 31: Relação instrutor-matéria

1.2 *REGRA DE PRECEDÊNCIA*

O cadastro a seguir refere-se à regra de precedência entre as matérias. A indicação da relação entre as matérias está de acordo com sucessores imediatos. Tomando-se a primeira linha da próxima tabela como exemplo, sabe-se que a matéria FIN1 só poderá ser ensinada para uma determinada turma, caso ela já tenha aprendido o conteúdo da matéria INTRO.

matéria	Sucessor
INTRO	FIN1
INTRO	LOG1
INTRO	PROD1
INTRO	RH1
INTRO	VEN1
FIN1	FIN2
PROD1	PROD2
LOG1	LOG2
LOG2	LOG4
LOG2	LOG3
INTRO	REL
FIN1	FIN3
FIN1	FIN4
PROD2	PROD3
RH1	RH2

Tabela 32: Regra de precedência entre as matérias

1.3 RELAÇÃO TURMA-MATÉRIA

A tabela a seguir indica as matérias que cada turma deverá assistir. Ressalta-se que esta decisão é extremamente importante e deve ser tomada a partir de uma análise de competências por profissionais experientes e qualificados neste assunto.

t/c	INTRO	FIN1	LOG1	PROD1	RH1	RH2	VEN1	FIN2	PROD2	LOG2	LOG4	LOG3	REL	FIN3	FIN4	PROD3
T11	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	2	2	0	0	0	0	0	2	0	0	0	0	0	0	0	0
T13	2	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0
T14	2	2	0	0	0	0	0	0	0	0	0	0	0	0	2	0
T15	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
T16	2	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0
T17	2	0	2	0	0	0	0	0	0	2	0	2	0	0	0	0
T18	2	0	2	0	0	0	0	0	0	2	2	0	0	0	0	0
T19	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
T1A	2	0	0	2	0	0	0	0	2	0	0	0	0	0	0	0
T1B	2	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
T1C	2	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0
T1D	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
T1E	2	0	0	2	0	0	0	0	2	0	0	0	0	0	0	2
T21	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T22	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
T23	2	0	0	2	0	0	0	0	2	0	0	0	0	0	0	0
T24	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
T25	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
T31	2	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
T32	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
T33	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T34	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
T35	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
T36	2	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0

Tabela 33: Relação turma-matéria (2 significa que a turma deve assistir a matéria em questão).

1.4 INFORMAÇÕES SOBRE AS MATÉRIAS

No cadastro a seguir, é possível saber algumas informações sobre as matérias como por exemplo duração total, diária (agrupamento), tipo de matéria, tipo de sala que precisa, se a matéria permite que mais de uma turma sejam unidas e se há algum professor específico para lecioná-la.

Mat	DuraçãoTotal	agrupamento	tipo	salaPrecisa	SalaExclusiva?	pode Juntar?	prof?
INTRO	4	1	seguida	1	1	1	
FIN1	8	2	mesmoSlot	2	0	1	
LOG1	8	2	mesmoSlot	2	0	1	
PROD1	8	1	seguida	2	0	1	
RH1	8	1	seguida	2	0	1	
RH2	4	1	seguida	2	0	1	
VEN1	8	1	seguida	2	0	1	
FIN2	4	2	mesmoSlot	2	0	1	
PROD2	8	1	seguida	2	0	1	
LOG2	4	2	mesmoSlot	2	0	1	
LOG4	4	1	seguida	2	0	1	
LOG3	4	1	seguida	2	0	1	
REL	4	1	seguida	2	0	1	
FIN3	2	1	seguida	2	0	1	
FIN4	2	1	seguida	2	0	1	
PROD3	4	1	seguida	2	0	1	

Tabela 34: Informações sobre as matérias

1.5 INFORMAÇÃO SOBRE AS SALAS

Na tabela subsequente, informações sobre as salas como capacidade, tipo, local e custos são fornecidos. Destaca-se que “tipo 2” significa sala de pequeno porte com computadores e projetor e “tipo 1” significa uma sala maior onde não há computadores, somente projetor. O custo de abrir (última coluna) refere-se ao custo fixo da sala por estar utilizando-a. Este custo é mais apropriado para salas alugadas e não salas da própria empresa como é o caso em questão.

	capacidade	Tipo	Local	custo/hora	custoAbrir
Sala1	15	2	1	50	0
Sala2	25	2	1	50	0
Sala3	10	2	1	50	0
Sala4	10	2	2	50	0
Sala5	10	2	3	50	0
Sala6	10	2	3	50	0
Auditorio1	40	1	1	200	0
Auditorio2	40	1	1	200	0
Auditorio3	30	1	2	200	0
Auditorio4	30	1	3	200	0

Tabela 35: Informação sobre as salas

1.6 LOCAL DOS INSTRUTORES

Este cadastro mostra o local de origem e o local onde o instrutor começará o problema (localSimul). No estudo de caso, todos os instrutores começam o problema no local de origem.

proflocal	LocalSimul	LocalOriginal
Maria	1	1
João	1	1
Marcos	1	1
Carlos	1	1
Sonia	3	3
Ana	2	2
José	2	2
Antonio	1	1
Raul	2	2

Tabela 36: Local original e de simulação dos instrutores

1.7 RELAÇÃO TURMA-MATÉRIA-INSTRUTOR

Este cadastro serve para forçar algumas combinações de turma-matéria-instrutor. Sempre que não houver um determinado instrutor para dar um curso em alguma localidade, este cadastro provavelmente será utilizado.

Caso haja uma combinação turma-matéria em um local onde não há instrutor apto a dar esta matéria e o operador do TT não quer deslocar a turma de lugar, então este cadastro deverá obrigatoriamente ser utilizado. Se isto não ocorrer, o programa irá falhar.

turma	prof	INTRO	FIN1	LOG1	PROD1	RH1	RH2	VEN1	FIN2	PROD2	LOG2	LOG4	LOG3	REL	FIN3	FIN4	PROD3
T11																	
T12																	
T13																	
T14																	
T15																	
T16																	
T17																	
T18																	
T19																	
T1A																	
T1B								Sonia									
T1C																	
T1D																	
T1E																	Sonia
T21																	
T22																	
T23					Marcos					Marcos							
T24					Marcos												
T25														Maria			
T31																	
T32						Carlos											
T33			Ana														
T34				Raul													
T35														Sonia			
T36						Carlos	Carlos										

Tabela 37: Especificação forçada de instrutores para determinadas combinações turma-matéria

1.8 LOCAL E TAMANHO DAS TURMAS

O cadastro seguinte possui uma informação de entrada e uma calculada. O tamanho da turma é calculado de acordo com a quantidade de alunos que lhe pertencem. Já o local (de origem) é um parâmetro direto de entrada do operador do TT.

	tamanho	Local
T11	7	1
T12	5	1
T13	6	1
T14	5	1
T15	6	1
T16	5	1
T17	4	1
T18	4	1
T19	6	1
T1A	4	1
T1B	5	1
T1C	4	1
T1D	6	1
T1E	3	1
T21	6	2
T22	5	2
T23	5	2
T24	4	2
T25	5	2
T31	6	3
T32	3	3
T33	4	3
T34	4	3
T35	5	3
T36	3	3

Tabela 38: Informação sobre as turmas (local origem e tamanho)

1.9 RELAÇÃO TURMA-MATÉRIA-LOCAL

Este cadastro serve essencialmente para fazer com que turmas viajem. Esta decisão, por ser estratégica, deve ser tomada pelo operador de TT após análises de custo pertinentes. Como no estudo de caso não há nenhum deslocamento forçado, o local de cada turma coincide com o seu local de origem independentemente da matéria em questão.

	INTRO	FIN1	LOG1	PROD1	RH1	RH2	VEN1	FIN2	PROD2	LOG2	LOG4	LOG3	REL	FIN3	FIN4	PROD3
T11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T1A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T1B	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T1C	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T1D	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T1E	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T21	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
T22	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
T23	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
T24	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
T25	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
T31	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
T32	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
T33	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
T34	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
T35	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
T36	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Tabela 39: Relação turma-matéria-local

1.10 RELAÇÃO ALUNO-TURMA

A tabela subsequente dispensa maiores comentários, pois a única informação contida é a relação de qual aluno pertence a que turma. Justamente esta relação que permitirá o *software* calcular o tamanho das turmas.

1.11 CUSTO E DISTÂNCIA ENTRE LOCALIDADES

As duas tabelas subseqüentes explicitam os custos de viagem entre localidades assim como o tempo (em número de *slots*) que leva para ir de um local a outro. Destaca-se que o custo de ida pode ser diferente do custo da volta conforme ocorre entre as localidades 2 e 3. Vale lembrar que um tempo “-1” significa que independente do horário em que o instrutor ficou livre no dia anterior, no dia seguinte ele já estará na outra localidade.

custoviagem	1	2	3
1	0	50	50
2	50	0	50
3	50	10	0

Tabela 41: Custo entre localidades

	1	2	3
1	0	-1	-1
2	-1	0	-1
3	-1	-1	0

Tabela 42: Tempo de deslocamento (-1 = dia seguinte)

1.12 PESO DOS CRITÉRIOS

A tabela subseqüente mostra os pesos dos critérios utilizados:

Critério	Peso	Abreviatura	Método
custo de abrir sala	1	abrirSala	Heurística
custo por utilizar a sala	1	usarSala	Heurística
custo dos dias que passou do horizonte desejado	500000	excHorizonte	Heurística
custo de viagem de instrutor	5000	viagemInstrutor	Heurística
custo de viagem da turma	5000	viagemTurma	Heurística
preferência dos alunos	1	prefTurma	Busca
preferência dos professores	1	prefProf	Busca

Tabela 43: Peso dos critérios

Portanto, o objetivo principal do estudo de caso está sendo respeitado já que o maior peso está relacionado com os dias que passaram do horizonte desejado (2 semanas neste caso).

1.13 OUTRAS CONSIDERAÇÕES

Outras considerações acerca dos cadastros:

◆ Os cadastros contendo as disponibilidades de horário dos professores e das salas não serão colocados neste trabalho, pois todas as células estão preenchidas com o valor unitário. Portanto, as salas não apresentam nenhuma indisponibilidade e os professores não indicaram nenhuma preferência de horário, podendo ser alocados em todos os dias, sem distinção.

◆ O cadastro contendo a informação turma-matéria-união, por não ter sido utilizado neste Estudo de Caso, não foi colocado neste anexo. Para mais informações sobre esta tabela, consultar o anexo anterior.

◆ Nas soluções 1 e 2, o cadastro da disponibilidade dos alunos foi preenchido com o valor unitário. Portanto, os alunos não possuem nenhuma indisponibilidade e os horários apresentam igual prioridade. Já na solução 3, houve uma alteração neste cadastro fazendo com que alguns alunos preferissem assistir às aulas de manhã e outros à tarde. Este ponto está melhor explicado no próximo item. Destaca-se que o cadastro de disponibilidade de horários dos alunos é a base para a construção da tabela de disponibilidade de horário das turmas. No caso das soluções 1 e 2, o cadastro de disponibilidade das turmas também apresentou valores unitários para todos os *slots* já que o cadastro dos alunos não apresenta nenhuma variação.

2. DIFERENÇA ENTRE AS EXECUÇÕES

Conforme comentado, o *software* foi executado três vezes para resolver o estudo de caso. As duas primeiras vezes diferem somente pelos critérios de parada utilizados, ou seja, não há nenhuma divergência no cadastro das entidades (aluno ou professor). Entretanto, na terceira solução houve uma mudança. Na primeira e na segunda, os alunos não contavam com preferência de horários, ou seja, daquela escala já explicada (0 = não pode; 1 = pode; 2 = horário preferencial), somente o número 1 foi utilizado. Já na terceira execução do *software*, algumas preferências (número 2) foram utilizadas fazendo com que algumas turmas preferissem aulas de manhã e outras à tarde, enquanto que outras continuaram indiferentes. Essa diferença

cadastral é extremamente importante quando se refere à Busca Tabu desenvolvida. Nas duas primeiras execuções, a BT jamais conseguiria achar uma resposta melhor, pois sua função, melhorar a alocação de acordo com a preferência de horário dos funcionários, é inócua em um cenário onde nenhum aluno/funcionário indicou horários com maior prioridade (número 2). Já na terceira solução, a BT teve oportunidade de melhorar algumas soluções, já que nesta execução do *software* alguns funcionários se manifestaram em relação às suas preferências. O cadastro da disponibilidade de horário das turmas para a terceira execução do *software* é reproduzido na seqüência.

	S11	S12	S13	S14	T11	T12	T13	T14	Qr11	Qr12	Qr13	Qr14	Qn11	Qn12	Qn13	Qn14	Sx11	Sx12	Sx13	Sx14
T11	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1
T12	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1
T13	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1
T14	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
T15	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
T16	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
T17	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
T18	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1
T19	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1
T1A	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1
T1B	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1
T1C	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
T1D	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
T1E	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
T21	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
T22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T32	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T33	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T34	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T35	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T36	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Tabela 44: Preferência das turmas em relação à primeira semana

ANEXO D

ANEXO D

A tabela a seguir mostra resolução manual do Estudo de Caso por uma pessoa que lida com treinamento de empresas.

1	Seq11	Seq12	Seq13	Seq14	Ter1	Ter2	Ter3	Ter4	Qua11	Qua22	Qua33	Qua44	Qui11	Qui22	Qui33	Qui44	Sex11	Sex22	Sex33	Sex44
Sala1					FIN1 MARIA T11,T12,T13,T14	FIN1 MARIA T11,T12,T15,T16,T17,T18	LOG1 JOAO T15,T16,T17,T18	LOG1 JOAO T15,T16,T17,T18	FIN1 MARIA T11,T12,T13,T14	FIN1 MARIA T11,T12,T13,T14	LOG1 JOAO T15,T16,T17,T18	LOG1 JOAO T15,T16,T17,T18	FIN1 MARIA T11,T12,T13,T14	FIN1 MARIA T11,T12,T13,T14	LOG1 JOAO T15,T16,T17,T18	LOG1 JOAO T15,T16,T17,T18	FIN1 MARIA T11,T12,T13,T14	FIN1 MARIA T11,T12,T13,T14	LOG1 JOAO T15,T16,T17,T18	LOG1 JOAO T15,T16,T17,T18
Sala2					PROD1 MARCOS T19,T1A,T1E	PROD1 MARCOS T19,T1A,T1E	PROD1 MARCOS T19,T1A,T1E	PROD1 MARCOS T19,T1A,T1E	PROD1 MARCOS T19,T1A,T1E	PROD1 MARCOS T19,T1A,T1E	PROD1 MARCOS T19,T1A,T1E	PROD1 MARCOS T19,T1A,T1E	PROD2 MARCOS T1A,T1E	PROD2 MARCOS T1A,T1E	PROD2 MARCOS T1A,T1E	PROD2 MARCOS T1A,T1E	PROD2 MARCOS T1A,T1E	PROD2 MARCOS T1A,T1E	PROD2 MARCOS T1A,T1E	PROD2 MARCOS T1A,T1E
Sala3					RH1 CARLOS T1C	RH1 CARLOS T1C	RH1 CARLOS T1C	RH1 CARLOS T1C	RH1 CARLOS T1C	RH1 CARLOS T1C	RH1 CARLOS T1C	RH1 CARLOS T1C	VEN1 SONIA T1B	VEN1 SONIA T1B	VEN1 SONIA T1B	VEN1 SONIA T1B	VEN1 SONIA T1B	VEN1 SONIA T1B	VEN1 SONIA T1B	VEN1 SONIA T1B
Sala4					FIN1 ANA T21	FIN1 ANA T21	LOG1 RAUL T22	LOG1 RAUL T22	FIN1 ANA T21	FIN1 ANA T21	LOG1 RAUL T22	LOG1 RAUL T22	FIN1 ANA T21	FIN1 ANA T21	LOG1 RAUL T22	LOG1 RAUL T22	FIN1 ANA T21	FIN1 ANA T21	LOG1 RAUL T22	LOG1 RAUL T22
Sala5					VEN1 SONIA T31	VEN1 SONIA T31	VEN1 SONIA T31	VEN1 SONIA T31	VEN1 SONIA T31	VEN1 SONIA T31	VEN1 SONIA T31	VEN1 SONIA T31	RH1 CARLOS T32,T36	RH1 CARLOS T32,T36	RH1 CARLOS T32,T36	RH1 CARLOS T32,T36	RH1 CARLOS T32,T36	RH1 CARLOS T32,T36	RH1 CARLOS T32,T36	RH1 CARLOS T32,T36
Sala6																				
Auditorio1	INTRO MARIA T11,T12,T13,T14,T15,T16,T17	INTRO MARIA T11,T12,T13,T14,T15,T16,T17	INTRO MARIA T11,T12,T13,T14,T15,T16,T17	INTRO MARIA T11,T12,T13,T14,T15,T16,T17																
Auditorio2	INTRO JOAO T18,T19,T1A,T1B,T1C,1D,1E	INTRO JOAO T18,T19,T1A,T1B,T1C,1D,1E	INTRO JOAO T18,T19,T1A,T1B,T1C,1D,1E	INTRO JOAO T18,T19,T1A,T1B,T1C,1D,1E																
Auditorio3	INTRO JOSE T21,T22,T23,T24,T25	INTRO JOSE T21,T22,T23,T24,T25	INTRO JOSE T21,T22,T23,T24,T25	INTRO JOSE T21,T22,T23,T24,T25																
Auditorio4	INTRO SONIA T31,T32,T33,T34,T35,T36	INTRO SONIA T31,T32,T33,T34,T35,T36	INTRO SONIA T31,T32,T33,T34,T35,T36	INTRO SONIA T31,T32,T33,T34,T35,T36																

Seq21	Seq22	Seq23	Seq24	Ter21	Ter22	Ter23	Ter24	Qua21	Qua22	Qua23	Qua24	Qui21	Qui22	Qui23	Qui24	Sex21	Sex22	Sex23	Sex24
FIN2 MARIA T12	FIN2 MARIA T12	LOG2 JOAO T16,T17,T18	LOG2 JOAO T16,T17,T18	FIN2 MARIA T12	FIN2 MARIA T12	LOG2 JOAO T16,T17,T18	LOG2 JOAO T16,T17,T18	LOG3 ANTONIO T17	LOG3 ANTONIO T17	LOG3 ANTONIO T17	LOG3 ANTONIO T17	FIN3 MARIA T13	FIN3 MARIA T13	FIN4 MARIA T14	FIN4 MARIA T14				
REL SONIA T1D	REL SONIA T1D	REL SONIA T1D	REL SONIA T1D					PROD3 SONIA T1E	PROD3 SONIA T1E	PROD3 SONIA T1E	PROD3 SONIA T1E	LOG4 ANTONIO T18	LOG4 ANTONIO T18	LOG4 ANTONIO T18	LOG4 ANTONIO T18				
RH2 CARLOS T1C	RH2 CARLOS T1C	RH2 CARLOS T1C	RH2 CARLOS T1C					PROD3 SONIA T1E	PROD3 SONIA T1E	PROD3 SONIA T1E	PROD3 SONIA T1E								
PROD1 MARCOS T23,T24	PROD1 MARCOS T23,T24	PROD1 MARCOS T23,T24	PROD1 MARCOS T23,T24	PROD1 MARCOS T23,T24	PROD1 MARCOS T23,T24	PROD1 MARCOS T23,T24	PROD1 MARCOS T23,T24	PROD2 MARCOS T23	REL MARIA T25	REL MARIA T25	REL MARIA T25	REL MARIA T25							
FIN1 ANA T33	FIN1 ANA T33	LOG1 RAUL T34	LOG1 RAUL T34	FIN1 ANA T33	FIN1 ANA T33	LOG1 RAUL T34	LOG1 RAUL T34	FIN1 ANA T33	FIN1 ANA T33	LOG1 RAUL T34	LOG1 RAUL T34	FIN1 ANA T33	FIN1 ANA T33	LOG1 RAUL T34	LOG1 RAUL T34	REL SONIA T35	REL SONIA T35	REL SONIA T35	REL SONIA T35
				RH2 CARLOS T36	RH2 CARLOS T36	RH2 CARLOS T36	RH2 CARLOS T36												

Tabela 47: Resolução manual do Estudo de Caso

LISTA DE REFERÊNCIAS

LISTA DE REFERÊNCIAS

ABDENNADHER, S.; RINGEISSEN, C. (2000) *University* course timetabling using constraint handling rules. **Applied Artificial Intelligence**, v. 14, p. 311-325, 2000.

ABRAMSON, D. A. (1991) Constructing *school* timetables using simulated annealing: sequential and parallel algorithms, **Management Science**, v. 37, n. 1, pp 98 – 113, jan. 1991.

ABRAMSON, D. A.; DANG, H.; KRISNAMOORTHY, M. (1999) Simulated annealing cooling schedules for the *school* timetabling problem, **Asia-Pacific Journal of Operational Research**, v. 16, n.1, p. 1-22, 1999.

AICKELIN, U. ET AL. (2003) Timetabling and rostering. **European Journal of Operational Research**, v. 156, n. 2, 2003.

ALVAREZ-VALDES R.; CRESPO E.; TAMARIT J.M. (2002) Design and implementation of a course scheduling system using tabu search. **European Journal of Operational Research**, v. 137, p. 512 -523, 2002.

AL-YAKOOB, S. M.; SHERALI, H. D. (2005) Mathematical programming models and algorithms for a class-faculty assignment problem. **European Journal of Operational Research**, 2005.

ARNTZEN, H.; LOKKETANGEN, A. (2005) **A local search heuristic for a university timetabling problem.** Disponível em: <www.idsia.ch/Files/ttcomp2002/arntzen.pdf>. Acesso em: 05 de jul. 2005

ASAP (2005). Watt history. Disponível em: <<http://www.asap.cs.nott.ac.uk/ASAP/watt/history.html>>. Acesso em: 01 de abr. 2005.

ASTD (2005) The human capital challenge. Disponível em: <http://www.astd.org/NR/rdonlyres/EF96C137-94DA-40CD-8B3F-FA18662DDEAD/0/human_capital_press_release.pdf>. Acesso em: 19 de ago. 2005.

ATKINSON, P. A. (1999). **Estudo de técnicas para minimização do makespan no job shop clássico com base no modelo em grafo disjuntivo**. Dissertação (Mestrado). Escola Politécnica – Universidade de São Paulo. São Paulo, 1999.

BADICA, C.; LALESCU, L. (2005) *University of Craiova*. **Timetabling experiments using genetic algorithms**. Disponível em: <http://is.comp-craiova.ro/~badica_costin/cercetare/papers/tainn03_p2.pdf>. Acesso em: 08 de jul. 2005.

BARTAK, R.; RUDOVA, H. (2001) Integrated modelling for planning, scheduling, and timetabling problems. IN: **PROCEEDINGS OF THE TWENTIETH WORKSHOP OF THE UK PLANNING AND SCHEDULING SPECIAL INTEREST GROUP (PLANSIG 2001)**. Edinburgh (UK): Old College, *University of Edinburgh*, p. 19-31, 2001.

BELLANTI, F. ET AL. (2004) A greedy-based neighborhood search approach to a nurse rostering problem. **European Journal of Operational Research**, v. 153, p. 28-40, 2004.

BOJIKIAN, P. P. (2004). **Estudo sobre a implantação de um sistema ERP como apoio ao processo de mudança da estrutura organizacional**. 2004. 134p. Dissertação (Graduação) – Escola Politécnica, Universidade de São Paulo. São Paulo, 2004.

BPR (2005). Glossary of Terms. Disponível em: <<http://www.gao.gov/special.pubs/bprag/bprgloss.htm>>. Acesso em: 01 de mar. 2005.

BPT (2005). Disponível em: < http://www.bptrends.com/resources_glossary.cfm>. Acesso em: 01 de abr. 2005.

BSJM (2005). Disponível em: <<http://www.indiainfoline.com/bisc/jmek.html>>. Acesso em: 01 de abr. 2005.

BUFE, M ET AL. (2001) Automated solution of a highly constrained *school* timetabling problem - preliminary results. IN: **BOERS, E. J. W. ET AL. APPLICATIONS OF EVOLUTIONARY COMPUTING**, p. 431-440, 2001.

BURKE, E. K. ET AL. (2002) A Case Based Heuristic Selection Investigation of Hill Climbing, Simulated Annealing and Tabu Search for Exam Timetabling Problems” (abstract). In: **PROCEEDINGS OF THE 4TH INTERNATIONAL CONFERENCE ON THE PRACTICE AND THEORY OF AUTOMATED TIMETABLING (PATAT 2002)**, p. 408-410, 2002.

BURKE, E. K.; KENDALL. G.; SOUBEIGA. E. (2005) A tabu-search hyperheuristic for timetabling and rostering. **Journal of Heuristics**, v. 9, n. 6, p.451-470, dez. 2003.

BURKE, E. K.; PETROVIC, S. (2002) Recent research directions in automated timetabling. **European Journal of Operational Research**, v. 140, p. 266-280, 2002.

CAIXETA, N. (2005) Procura-se mão-de-obra qualificada. **Revista Exame**. Disponível em: <
http://portalexame.abril.com.br/edicoes/836/economia/conteudo_61070.shtml>.
Acesso em: 09 de jul. 2005.

CARRASCO, M. P.; PATO, M., V. (2004) A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem. **European Journal of Operational Research**, 153(1):65–79, 2004.

CARTER, M. W. (1986) A survey of practical applications of *examination* timetabling algorithms. **Operations Research**, v.34, n.2, p.193-202, 1986.

CARTER, M. W. (1989) A Lagrangian Relaxation Approach to the Classroom Assignment Problem. **Infor**, v.27, n.2, p.230-246, 1989.

CARTER, M. W.; TOVEY, C. A. (1992) When is the classroom assignment problem hard? **Operations Research**, v.40, n.1, p.28-39, jan.-fev. 1992.

CARVALHO, R. B. ET. AL. (2000) **Acelerando a espiral do conhecimento com a tecnologia da informação**. 2000. Programa de Pós-Graduação em Ciências - UFMG - Departamento de Ciência da Computação - Centro Universitário FUMEC - Banco de Desenvolvimento de Minas Gerais - BDMG, Belo Horizonte, 2000.

CASSIOLATO, J. E; LASTRES, H. M. M.; SZAPIRO, M. (2003). **Arranjos e sistemas produtivos locais e proposições de políticas de desenvolvimento industrial e tecnológico**. Disponível em: <<http://www.ie.ufrj.br/redesist/P2/textos/NT27.PDF>>. Acesso em: 05 de dez. 2003.

CAVALCANTI, E. P. (1995) **Revolução da informação: algumas reflexões**. **Caderno de pesquisas em administração**, São Paulo, v.1, n. 1, 1995.

CECATTO, C. (2005) **A importância da gestão do conhecimento nas empresas**. Disponível em: <<http://www.sebraepb.com.br:8080/bte/index.asp?vdir=Gest%E3o%5CGest%E3o+d o+Conhecimento>>. Acesso em: 15 de jul. 2005.

CHAND, A. (2002) A heuristic approach to constraint optimization in timetabling. **South Pacific Journal of Natural Science**, vol. 20, p. 64-67, 2002.

CHENG, E.; KRUK, S.; LIPMAN, M. J. (2002) Flow Formulations for the Student Scheduling Problem. **PATAT 2002**, p. 299-309, 2002.

CLIP LAB (2005). How Does a CLP System Work? Disponível em: <http://clip.dia.fi.upm.es/~vocal/public_info/seminar_notes/node15.html>. Acesso em: 20 de jul. 2005.

CMU (2005). A Parallel Genetic Solution to the Class Timetabling Problem. Disponível em: <<http://www.andrew.cmu.edu/~dpatel/final.htm>>. Acesso em: 01 de jul. 2005.

COMPUTATIONAL COMPLEXITY THEORY (2005). Disponível em:<<http://www.csc.liv.ac.uk/~ravi/COMP108/part5/notes5.pdf>>. Acesso em: 21 de jul. 2005.

CORREIA, A. M. R.; SARMENTO, A. (2005) **Gestão do conhecimento: competências para a inovação e competitividade**. Disponível em: <http://www.isegi.unl.pt/docentes/acorreia/preprint%5CAPSIOT_Correia_Sarmento_260303.pdf>. Acesso em: 06 de jul. 2005.

COSTA, P. E. C.; GOUVINHAS, R. P. (2004) Gestão do conhecimento: quebrando o paradigma cultural para motivar o compartilhamento do conhecimento dentro das organizações. In: XXIV Encontro Nacional de Engenharia de Produção (ENEGEP), 2004, Florianópolis. **Anais de Resumo - ENEGEP 2004**. Porto Alegre : ABEPRO, 2004. p. 288-288.

DA SILVA, E. L.; DA CUNHA, M. V. (2002) A formação profissional no século XXI: desafios e dilemas. **Ciência da Informação**, Brasília, v. 31, n. 3, p. 77-82, set./dez. 2002.

DA SILVA, S. L. (2005) **Gestão do conhecimento: uma revisão crítica orientada pela abordagem da criação do conhecimento**. Disponível em: <<http://www.scielo.br/pdf/ci/v33n2/a15v33n2.pdf>>. Acesso em: 02 de ago. 2005.

DAKALAKI, S.; BIRBAS, T. (2005) Efficient solutions for a *university* timetabling problem through integer programming. **European Journal of Operational Research**, v.160, p.106 – 120, 2005.

DASKALAKI, S.; BIRBAS, T.; HOUSOS, E. (2004) An integer programming formulation for a case study in *university* Timetabling. **European Journal of Operational Research**, v. 153, n. 1, p. 117–135, 2004.

DAVENPORT, T. H. (2000) **Mission Critical: realizing the promise of enterprise systems**. Boston: Harvard Business School Press, 330 p., 2000.

DE WERRA, D. (1969) Construction of *school* timetables by flow methods. **Infor**. v.1, n.1, p. 12-22, 1969.

DE WERRA, D. (1997) The combinatorics of timetabling. **European Journal of Operational Research**, v. 96, p. 504-513, 1997.

DEL VAL, P.; FUENTES, C. M. (2003) Resistance to change: a literature review and empirical study. **Journal of Management Decision**. v. 41, n. 2, p.148-155, 2003.

DEMAZEAU, F. ET AL. (2005) Interactively solving *school* timetabling problems using extensions of constraint programming. **Practice and Theory of Automated Timetabling V**, Lecture Notes in Computer Science, n. 3616, primavera, 2005

DICTIONARYLABORLAWTALK (2005). Enciclopédia: Complexity Theory. Disponível em: <http://encyclopedia.laborlawtalk.com/Computational_complexity>. Acesso em: 01 de ago. 2005.

DIMOPOULO, M. ET AL. (2004) An automated *university* course timetabling system developed in a distributed environment: a case study. **European Journal of Operational Research**, v. 153, p.136-147, 2004.

DINKEL, J. J.; MOTE, J.; VENKATARAMANAN, M. A. (1989) An Efficient Decision Support System for Academic Course Scheduling. **Operations Research**, v. 37, n6, p. 853- 864, 1989.

DOANE, M.; VOLLMER, J. (2005) **Gaining continuous value from ERP through a comprehensive, continuous education strategy**. Disponível em: <http://www.enableit.ubc.ca/documents/fms/erp_user_and_team_training_white_paper.pdf>. Acesso em: 27 de jul. 2005.

DUCK, J.D. (1993) Managing change: The art of balancing. **Harvard Business Review**, November 1, 1993.

ELMOHAMED, M. A. S.; CODDINGTON, P.; FOX, G. (1997) A comparison of annealing techniques for academic course scheduling. **PATAT 1997**, p.92-114, 1997.

ENZHE, Y.; SUNG, K. (2002) A Genetic Algorithm for a *University* Weekly Courses Timetabling Problem. **International transactions in Operational Research**, v. 9, p. 703 – 717, 2002.

FERLAND, J. A.; FLEURENT, C. (1994). SAPHIR: A Decision Support System for Course Scheduling. **Interfaces**, v. 24, n. 2, p. 105-115, 1994.

FLORÊNCIO, D. G. L. ET AL. (2005) **A dimensão Humana da gestão do Conhecimento**. Disponível em: <<http://www.aceso.net/.../Biblioteca%20Digital/Biblioteca%20Sebrae/Gest%3o/NT000663F6.pdf>>. Acesso em: 01 de jul. 2005..

FRANGOULI, H.; HARMANDAS, V. (1995) STAMATOPOULOS, P. UTSE: Construction of optimum timetables for *university* courses: a CLP based approach. In: **PROCEEDINGS OF THE THIRD INTERNATIONAL CONFERENCE ON THE PRACTICAL APPLICATIONS OF PROLOG**, p. 225-243, 1995.

FREITAS, A. M. (2005) **A empresa do novo milênio**. Disponível em: <http://www.aceso.net/fanese/gestao_da_qualidade_e_produtividade/Biblioteca%20Digital/Biblioteca%20Sebrae/Gest%3o/NT0003EE66.pdf>. Acesso em: 27 de jul. 2005.

GAMBÔA, F. A. R.; CAPUTO, M. S.; BRESCIANI FILHO, E. (2004) Métodos para gestão de riscos em implementações de sistemas ERP baseado em fatores críticos de sucesso. **Revista de Gestão da Tecnologia e Sistemas de Informação**, v. 1, n. 1, p. 46 – 63, 2004.

GEORGE, V. ; CUNHA, C. B. (2002) Metaheurísticas e sua Aplicação em Problemas de Otimização Combinatória. **Produção em Iniciação Científica da Escola Politécnica da Universidade de São Paulo**, São Paulo, v. 1, 2002.

GLIBOVETS N. N.; MEDVID S. A. (2003) Genetic algorithms used to solve scheduling problems. **Cybernetics and Systems Analysis**, v. 39, n. 1, p. 81-90, 2003.

GLOVER, F; LAGUNA, M. (1997). **Tabu Search**. Massachusetts: Kluwer Academic Publishers, 381 p., 1997.

HAWKING P.; FOSTER S.; STEIN A. (2004). **ERP II & Change Management: the real struggle for ERP systems practices**. Managing Business with SAP: Planning, Implementation and Evaluation, Idea Group, 2004.

HENZ, M.; WURTZ, J. (1996) Constraint-based timetabling: a case study. **Applied Artificial Intelligence**, vol. 10, p.439-453, 1996.

HOETZEL, B. R. (2005) **Change Management within ERP projects**. Disponível em: <http://www.mittelstand-competence-center.de/standardsoftware.nsf/f1b7ca69b19cbb26c12569180032a5cc/d5c75164726dbe10c1256ffc006d1567!OpenDocument>>. Acesso em: 15 de ago. 2005.

HUMPHREY, J. (2003) **Opportunities for SME's in developing countries to upgrade in a global economy**. Geneva: International Labour Office, 2003 (Working Paper 43).

HUSSIN, N. M.; KENDALL, G. (2005) Tabu search hyper-heuristic approach to the *examination* timetabling problem at *university* technology MARA. **ASAP Research Group**. Disponível em: <http://www.asap.cs.nott.ac.uk/patat/patat04/patat04-full-papers.shtml>>. Acesso em: 01 de jul. 2005.

IC PARC (2005). The ECLIPSE^e Constraint Logic Programming System (2005). Disponível em: < <http://www.icparc.ic.ac.uk/eclipse/>>. Acesso em: 20 de jul. 2005.

IVANQUI, I. L. (2005) **Um modelo para solução do problema de arranjo físico de instalações interligadas por corredores**. Disponível em: < <http://www.eps.ufsc.br/teses98/ivanqui/>>. Acesso em: 01 de abr. 2005.

KOLSKI, C.; PIECHOWIAK, S. (2004) Towards a generic object oriented decision support system for *university* timetabling: an interactive approach. **International Journal of Information Technology and Decision Making**, v. 3, n. 1, p. 179-208, 2004.

KRUSE, K. (2005). **The benefits and drawbacks of e-learning**. Disponível em: http://www.e-learningguru.com/articles/art1_3.htm>. Acesso em: 01 de mar. 2005.

LEMOS, C. (2000) Inovação na Era do Conhecimento. **Parcerias Estratégicas**, n. 8, p. 157-179, MCT, maio 2000.

LOUREIRO, J. L. (2003) **Gestão do conhecimento**. Portugal: Centro Atlântico, 208 p., 2003.

LUDMER, G.; RODRIGUES FILHO, J. (2003) ERP e teoria crítica: alertas para os riscos de uma disparada para um novo tipo de Iron Cage. In: **THIRD INTERNATIONAL CONFERENCE OF THE IBEROAMERICAN ACADEMY OF MANAGEMENT**, São Paulo, 2003.

MANFRIN, M.; SAMPELS, M.; SOCHA, K. (2005) Ant Algorithms for the *university* timetabling problem with regard to the-state-of-the-art. **Universidade Livre de Bruxelas**. Disponível em: <<http://www.iihe.ac.be/internal-report/2003/index-2003.html>>. Acesso em: 07 de set. 2005.

MAUSSER H. E.; MAGAZINE M. J. (1996) Comparison of neural and heuristic methods for a timetabling problem. **European Journal of Operational Research**, v.93, p. 271-87, 1996.

MERLOT, L. T. G. ET AL. (2002). A Hybrid Algorithm for the *Examination* Timetabling Problem. **The Practice and Theory of Automated Timetabling**, v. 2740, p. 207-231, 2002.

MORTON, T; PENTICO, D. (1993). **Heuristic Scheduling Systems**: with applications to Productions Systems and Project Management. New York: John Wiley & Sons, 1993, 695p.

MULLER, T.; BARTAK, R. (2002) Interactive Timetabling: Concepts, Techniques, and Practical Results. IN: **PROCEEDINGS TO PATAT'02 CONFERENCE**, Gent, 2002.

MÜLLER, T.; RUDOVÁ, H. (2004) Minimal Perturbation Problem in Course Timetabling. In **PATAT 2004 - PROCEEDINGS OF THE 5TH**

INTERNATIONAL CONFERENCE ON THE PRACTICE AND THEORY OF AUTOMATED TIMETABLING. Pittsburgh, p. 283-303, 2004.

NORRIS, G. ET AL. (2000). **E-Business and ERP:** transforming the enterprises. New York: John Wiley & Sons, 2000, 194p.

PAIVA, S. B. (2000) Capital intelectual: um novo paradigma para a gestão dos negócios. In: CONGRESSO BRASILEIRO DE CONTABILIDADE, 16., Goiânia, 2000. **Anais.** Goiânia: CFC, 2000.

PAIVA, S. B. (2005) **Algo novo no mundo empresarial:** do turbilhão de dados ao requinte da inteligência. Disponível em: [http://www.biblioteca.sebrae.com.br/bte/bte.nsf/8060FD1C2222BC5503256F9A005533F9/\\$File/NT000A3676.pdf](http://www.biblioteca.sebrae.com.br/bte/bte.nsf/8060FD1C2222BC5503256F9A005533F9/$File/NT000A3676.pdf)>. Acesso em: 20 de jul. 2005.

PAIVA, S. B. (2005b) **O capital humano no contexto da contabilidade gerencial.** Disponível em: <http://www.biblioteca.sebrae.com.br>>. Acesso em: 05 de jul. 2005

PTAK, C. A. (2000). **ERP:** tools, techniques, and applications for integrating the supply chain. Boca Raton: St. Lucie Press, 2000, 424p.

ROSENBERG, M. J. (2001). E-learning: strategies for delivering knowledge in the digital age. McGraw-Hill, 2001, 243p.

SANTORO, M. C. (2003). **Planejamento, Programação e Controle da Produção.** Apostila, v. 2, p. 118-131, 2003.

SANTOS, H. G.; OCHI, L. S.; SOUZA, M. J. F. (2004) An Efficient Tabu Search Heuristic for the *School* Timetabling Problem. **Lecture Notes in Computer Science**, v. 3059, p. 468-481, 2004.

STALLAERT, J. (1997) Automated timetabling improves course scheduling at UCLA. **Interfaces**, v. 27, p. 67-81, 1997.

TALBI, E. (2002) A taxonomy of hybrid metaheuristics. **Journal of Heuristics**, vol. 8, n° 5, p. 541-564, 2002.

TERRA, J. C. C. (2005) **Os desafios da produtividade: novas habilidades na era da informação e do conhecimento e o papel central da gestão do conhecimento.**

Disponível em: <<http://www.cgee.org.br>>. Acesso em: 07 de jul. 2005.

TERRA, J. C. C. (2001) **Gestão do conhecimento: o grande desafio empresarial.** São Paulo: Negócio Editora, 2001, 313p.

TOFFLER, A. (1980) **Third Wave: the classic study of tomorrow.** Nova Iorque: Bantam Books, 536 p., 1980.

TRIPATHY, A. (1984) *School timetabling – a case in large binary integer linear programming.* **Management Science.** v. 30, n. 12, dez. 1984.

TURBIT, N. (2005) **ERP implementation – the traps.** Disponível em: <<http://www.projectperfect.com.au>>. Acesso em: 27 de jul. 2005.

ZHAO, Q.; GOEBEL, R. (1999) Algorithms for handling soft constraints and their application to staff scheduling problems. **International Journal of Intelligent Systems in Accounting, Finance, and Management**, v.8, n. 1, p.119-129, 1999.
