

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
CENTRO INTERDISCIPLINAR DE TECNOLOGIAS NA EDUCAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA NA EDUCAÇÃO

Andre Zanki Cordenonsi

**Ambientes, Objetos e Dialogicidade: Uma Estratégia de Ensino Superior em  
Heurísticas e Metaheurísticas**

Porto Alegre  
2008

Andre Zanki Cordenonsi

**Ambientes, Objetos e Dialogicidade:  
uma estratégia de ensino superior em heurísticas e metaheurísticas**

Tede de doutorado apresentada ao Programa de Pós-Graduação em Informática na Educação do Centro Interdisciplinar de Novas Tecnologias na Educação da Universidade Federal do Rio Grande do Sul como requisito parcial para obtenção do título de Doutor em Informática na Educação.

Orientador:  
Prof. Dr. Felipe Martins Müller  
Co-Orientadora:  
Profa. Dra Léa da Cruz Fagundes

Porto Alegre  
2008

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)

---

C794a Cordenonsi, Andre Zanki

**Ambientes, objetos e dialogicidade: uma estratégia de ensino superior em heurísticas e metaheurísticas** [manuscrito] / Andre Zanki Cordenonsi; orientador: Felipe Martins Müller; Co-Orientadora: Léa da Cruz Fagundes. – Porto Alegre, 2008.  
f.228 + Anexos.

Tese (doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Informática na Educação, Centro Interdisciplinar de Novas Tecnologias na Educação. Programa de Pós-Graduação em Informática na Educação, 2008, Porto Alegre, BR-RS.

1. Computador na educação – Ambiente de aprendizagem – Ambiente virtual. 2. Pesquisa-ação – Educação. 3. Heurística. 4. Metaheurística. 5. AMEM I. 6. Freire, Paulo. I. Müller, Felipe Martins. II. Fagundes, Léa da Cruz. III. Título.

CDU – **371.694.3:681.3:65.012.122**

---

Bibliotecária Neliana Schirmer Antunes Menezes – CRB 10/939



SERVIÇO PÚBLICO FEDERAL  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
CENTRO INTERDISCIPLINAR DE NOVAS TECNOLOGIAS NA EDUCAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA NA EDUCAÇÃO

**Ata da Sessão de Defesa de Tese de Doutorado de  
ANDRÉ ZANKI CORDENONSI intitulada**

*Ambientes, Objetos e Dialogicidade: Uma Estratégia de Ensino Superior em  
Heurísticas e Metaheurísticas*

Às nove horas e trinta minutos do dia vinte e seis de junho de dois mil e oito, no Laboratório do CINTED, na Universidade Federal do Rio Grande do Sul, realizou-se a Defesa de Tese intitulada *Ambientes, Objetos e Dialogicidade: Uma Estratégia de Ensino Superior em Heurísticas e Metaheurísticas*, de autoria de **André Zanki Cordenonsi**, sob a orientação do Prof. Dr. Felipe Martins Muller e co-orientação da Prof<sup>a</sup>. Dr<sup>a</sup>. Lea da Cruz Fagundes. A Banca Examinadora, composta pelos Professores Doutores Magda Bercht, Luciana Saete Buriol e Fábio da Purificação Bastos, aprovou a Tese de Doutorado do aluno, que cumpriu com todos os requisitos e terá seu título de Doutor em Informática na Educação homologado pela Comissão de Pós-Graduação em Informática na Educação.

Prof. Dr. Felipe Martins Muller  
(Presidente e Orientador)

Prof<sup>a</sup>. Dr<sup>a</sup>. Lea da Cruz Fagundes  
(Co-orientadora)

Prof. Dr. Magda Bercht  
UFRGS - PPGIE

Prof<sup>a</sup>. Dr<sup>a</sup>. Luciana Saete Buriol  
UFRGS - PPGC

Prof. Dr. Fábio da Purificação Bastos  
UFSM - PPGE

*Para Giliane e João Vítor...*

## RESUMO

CORDENONSI, Andre Zanki. **Ambientes, Objetos E Dialogicidade: uma estratégia de ensino superior em heurísticas e metaheurísticas.** – Porto Alegre, 2008. f. + Anexos. Tese (Doutorado em informática na Educação) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Informática na Educação, Centro Interdisciplinar de Novas Tecnologias na Educação. Programa de Pós-Graduação em Informática na Educação, Porto Alegre, 2008.

O ensino de heurísticas e metaheurísticas é disperso na estrutura educacional superior brasileira em disciplinas da graduação e pós-graduação. De acordo com o plano de ensino do professor e sua concepção pedagógica, estas disciplinas podem ser centradas em aspectos práticos, com exemplos reais, ou focalizar as estruturas matemáticas que dão suporte à teoria e concepção dos modelos que são debatidos na literatura. No decorrer desta tese, é apresentada uma metodologia dialógica-problematizadora que, através da concepção, desenvolvimento e teste de duas ferramentas tecnológicas (o ambiente virtual de aprendizagem AMEM – Ambiente Multimídia para Educação Mediada por Computador e o objeto de aprendizagem LOBO – *Learning Object Based on Optimization*), busca uma forma inovadora de discutir os algoritmos heurísticos e metaheurísticos em torno de um problema comum, através de práticas baseadas no diálogo e instigando os alunos a buscar as respostas para os problemas que são apresentados aos mesmos. Concebida como uma proposta para dinamizar o ensino superior de heurísticas e metaheurísticas, esta tese apresenta a metodologia pedagógica, o ferramental tecnológico desenvolvido e, em sua última parte, uma aplicação real destas em uma disciplina de graduação. Considerando as questões de pesquisa previamente definidas na Matriz Dialógica-Problematizadora, é possível concluir que há muito trabalho ainda a percorrer para romper o ciclo positivista que pode agir como um inibidor de novas práticas, servindo como um ponto de apoio aos alunos que, acostumados ao ato de ouvir em uma sala de aula, não se sentem à vontade em *agir*. A utilização de ferramentas que permitam criar um espaço dialógico e questionador, como o AMEM e o LOBO, pode contribuir para quebrar este estigma.

**Palavras-chave:** 1. Computador na educação – Ambiente de aprendizagem – Ambiente virtual. 2. Pesquisa-ação – Educação. 3. Heurística. 4. Metaheurística. 5. AMEM I. 6. Freire, Paulo.

## ABSTRACT

CORDENONSI, Andre Zanki. **Ambientes, Objetos E Dialogicidade: uma estratégia de ensino superior em heurísticas e metaheurísticas.** – Porto Alegre, 2008. f. + Anexos. Tese (Doutorado em informática na Educação) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Informática na Educação, Centro Interdisciplinar de Novas Tecnologias na Educação. Programa de Pós-Graduação em Informática na Educação, Porto Alegre, 2008.

The education of heuristics and metaheuristics is dispersed in the Brazilian higher educational structure in disciplines of the graduation and posgraduation. The discipline can be centered in practical aspects, with real examples, or to focus the mathematical structures which support the theory and conception of the literature models. The professor, considering his education plan and his pedagogical conceptions, can choose one of them. In this thesis, the dialogical-problem methodology is presented, through the conception, development and test of two technological tools (the virtual environment of learning AMEM - Multimedia Educational Environment Mediated for Computer and the learning object LOBO - Learning Object Based on Optimization). The tools and the methodological approach produce an innovative way to the heuristics and metaheuristics educational, based in a common problem, through a practical focused in the dialogue and instigating the students to seek their owner answers. Conceived as a new proposal to the heuristics and metaheuristics higher education, this thesis presents the pedagogical methodology, the technological developed tool and a real application of these in one graduation discipline. Considering the questions of research previously defined in the Dialogical-Problem Matrix, the conclusions of the thesis appoints to a great effort needed to break the pedagogical traditional methodology, especially in the students' behavior. The use of tool to create a dialogical space, as AMEM and the LOBO, can contribute to break this stigma.

**Key-words: 1. Technology in education - the learning environment - Virtual Environment. 2. Research-action - Education. 3. Heurístic. 4. Metaheutístic. 5. AMEM I. 6. Freire, Paulo.**

## LISTA DE FIGURAS

Figura 1 - Dodecaedro regular – adaptado de (CAMPELLO, MACULAN, 1994) .....	24
Figura 2 - Grafo Hamiltoniano e um possível ciclo hamiltoniano para o problema – adaptado de (CAMPELLO, MACULAN, 1994) .....	24
Figura 3 - Formulação matemática do PCV .....	25
Figura 4 - Grafo de busca completo para um PCV de quatro cidades .....	29
Figura 5 - Grafo de busca com poda para um PCV de quatro cidades.....	30
Figura 6 - Heurística construtiva genérica para o PCV .....	32
Figura 7 - Ótimo local e global.....	33
Figura 8 - Heurística de melhoramento genérica para o PCV.....	34
Figura 9 - Busca local: escarpa e platô.....	35
Figura 10 - Matriz Dialógica Problematizadora.....	49
Figura 11 - Entrada de um aluno no AMEM 1.0.....	58
Figura 12 - Entrada de um aluno no AMEM 2.0.....	59
Figura 13 - Aluno entra na <i>Comunicação</i> com o perfil de <i>usuário</i> no AMEM 1.0 .....	60
Figura 14 - Aluno entra nas <i>Disciplinas</i> como <i>usuário</i> no AMEM 1.0 .....	61
Figura 15 - Aluno entra em uma disciplina específica no AMEM 1.0.....	62
Figura 16 - Aluno entra na <i>Comunicação</i> com o perfil de <i>aluno</i> no AMEM 1.0 .....	62
Figura 17 - Aluno entra em <i>Sair</i> com o perfil de <i>aluno</i> no AMEM 1.0.....	63
Figura 18 - Aluno entra no ambiente do AMEM 2.0 .....	64
Figura 19 - Aluno ingressa em uma disciplina no AMEM 2.0 .....	65
Figura 20 - Menu <i>Disciplinas</i> no AMEM 2.0 .....	66
Figura 21 - Menu <i>Comunicação</i> no AMEM 2.0 .....	66
Figura 22 - Aluno visualiza os murais da disciplina e gerais no AMEM 2.0.....	67
Figura 23 - Assistente para criação de um novo objeto de aprendizagem – Passo 1 .....	68
Figura 24 - Assistente para criação de um novo objeto de aprendizagem – Passo 1.1 .....	68
Figura 25 - Assistente para criação de um novo objeto de aprendizagem – Passo 2 .....	69
Figura 26 - Assistente para criação de um novo objeto de aprendizagem – Passo 2.1 .....	69
Figura 27 - Assistente para criação de um novo objeto de aprendizagem – Passo 3 .....	70
Figura 28 - Assistente para criação de um novo objeto de aprendizagem – Passo 3.1 .....	70
Figura 29 - Assistente para criação de um novo objeto de aprendizagem – Passo 3.2 .....	70
Figura 30 - Assistente para criação de um novo objeto de aprendizagem – Passo 4 .....	71
Figura 31 - Assistente para criação de um novo objeto de aprendizagem – Passo 4.1 .....	71
Figura 32 - Assistente para criação de um novo objeto de aprendizagem – Passo 5 .....	72
Figura 33 - Assistente para criação de um novo objeto de aprendizagem – Passo 6 .....	72
Figura 34 - Assistente para criação de um novo objeto de aprendizagem – Passo 7 .....	73
Figura 35 - Assistente para criação de um novo objeto de aprendizagem – Passo 8 .....	73
Figura 36 - Assistente para criação de um novo objeto de aprendizagem – Passo 9 .....	74
Figura 37 - Assistente para criação de um novo objeto de aprendizagem – Passo 9.1 .....	74
Figura 38 - Assistente para criação de um novo objeto de aprendizagem – Passo 10 .....	75
Figura 39 - Operacionalizador <i>programação</i> no AMEM 1.0.....	76
Figura 40 - Registros de uma <i>programação</i> no AMEM 1.0 .....	77
Figura 41 - Operacionalizador <i>atividade extra-classe</i> no AMEM 1.0 .....	77
Figura 42 - Operacionalizador <i>Atividade de Colaboração</i> no AMEM 1.0.....	78
Figura 43 - Bibliografias do operacionalizador <i>Atividade de Colaboração</i> no AMEM 1.079	

Figura 44 - Agendamento de uma <i>atividade</i> no AMEM 1.0.....	80
Figura 45 - Interface para uma <i>disciplina</i> no AMEM 2.0.....	82
Figura 46 - Assistente para criação de uma nova <i>aula</i> – Passo 1 – AMEM 2.0.....	82
Figura 47 - Assistente para criação de uma nova <i>aula</i> – Passo 2 – AMEM 2.0.....	83
Figura 48 - Assistente para criação de uma nova <i>aula</i> – Passo 3 – AMEM 2.0.....	83
Figura 49 - Assistente para criação de uma nova <i>aula</i> – Passo 4 – AMEM 2.0.....	84
Figura 50 - Assistente para criação de uma nova <i>aula</i> – Passo 5 – AMEM 2.0.....	85
Figura 51 - Assistente para criação de uma nova <i>aula</i> – Finalização – AMEM 2.0.....	85
Figura 52 - Assistente para criação de uma nova <i>atividade</i> – Passo1 – AMEM 2.0.....	86
Figura 53 - Assistente para criação de uma nova <i>atividade</i> – Passo2 – AMEM 2.0.....	87
Figura 54 - Assistente para criação de uma nova <i>atividade</i> – Passo3 – AMEM 2.0.....	87
Figura 55 - Assistente para criação de uma nova <i>atividade</i> – Finalização – AMEM 2.0 ...	88
Figura 56 - Criação de uma nova <i>colaboração</i> – AMEM 2.0.....	89
Figura 57 - Agendamento de um operacionalizador – AMEM 2.0.....	89
Figura 58 - Agendamento rápido de um operacionalizador – AMEM 2.0.....	91
Figura 59 - Avaliação de Atividades – AMEM 2.0.....	92
Figura 60 - Edição de comentários às atividades desenvolvidas – AMEM 2.0.....	92
Figura 61 - Fluxo de tarefa Responder Atividades.....	94
Figura 62 - Fluxo de tarefa Agendar atividade.....	95
Figura 63 - Processo de compilação de um programa de computador.....	98
Figura 64 - Processo de compilação de um programa Java.....	98
Figura 65 - Conjunto de especificações do SCORM (adaptado de SCORM, 2004).....	114
Figura 66 - <i>Package Interchange File</i> .....	117
Figura 67 - Tela de boas vindas do objeto LOBO.....	119
Figura 68 - Interface principal do LOBO.....	120
Figura 69 - Apresentação da História do Problema do Caixeiro Viajante.....	121
Figura 70 - Primeira seção da História do Problema do Caixeiro Viajante.....	122
Figura 71 - Tela explicativa da Heurística Vizinho Mais Próximo.....	123
Figura 72 - Tela explicativa sobre os algoritmos construtivos.....	124
Figura 73 - Tela explicativa da Heurística Inserção Mais Distante.....	124
Figura 74 - Algoritmo formal da Heurística Inserção Mais Distante.....	125
Figura 75 - Apresentação sobre a heurística de melhoramento 2-Opt.....	126
Figura 76 - Tela explicativa da Metaheurística Busca Tabu.....	127
Figura 77 - Apresentação sobre a implementação da metaheurística Busca Tabu.....	128
Figura 78 - Apresentação do conjunto de dados berlin52.tsp.....	130
Figura 79 - Funcionalidade Inserir dados.....	131
Figura 80 - Extrato do arquivo de dados no formato TSPLib.....	132
Figura 81 - Extrato do arquivo contendo a rota ótima no formato TSPLib.....	132
Figura 82 - Funcionalidade Abrir arquivos.....	133
Figura 83 - Tela de visualização da opção Sobre.....	134
Figura 84 - Heurística Construtiva: Algoritmo Aleatório.....	135
Figura 85 - Heurística Construtiva: Vizinho Mais Próximo.....	136
Figura 86 - Aplicação da Heurística Vizinho Mais Próximo sobre o problema pr76.tsp .	137
Figura 87 - Heurística Construtiva: Inserção Mais Distante.....	138
Figura 88 - Aplicação da Heurística Inserção Mais Distante sobre o problema pr76.tsp .	139

Figura 89 - Aplicação do algoritmo 3-Opt: rota original com os arcos que serão removidos e as novas rotas possíveis .....	140
Figura 90 - Heurística de Melhoramento: 2-Opt .....	141
Figura 91 - Aplicação de uma rodada da Heurística 2-Opt sobre o problema pr76.tsp ....	142
Figura 92 - Metaheurística: Busca Tabu .....	144
Figura 93 - Comportamento da função probabilística da metaheurística <i>simulated annealing</i> para diferentes graus de temperatura inicial e diferença entre soluções analisadas.....	146
Figura 94 - Metaheurística: <i>simulated annealing</i> .....	147
Figura 95 - Histórico de resultados para uma seqüência de aplicação de heurísticas sobre o problema <i>kroA100.tsp</i> .....	148
Figura 96 - Aplicação das heurísticas de melhoramento 2-Opt e 3-Opt sob um problema cuja solução inicial é aleatória.....	149
Figura 97 - Aplicação das heurísticas de melhoramento 2-Opt e 3-Opt sob um problema cuja solução inicial é dada pelo Vizinho Mais Próximo .....	150
Figura 98 - Aplicação das heurísticas de melhoramento 2-Opt e 3-Opt sob um problema cuja solução inicial é dada pela Inserção Mais Distante .....	150
Figura 99 - Tela de apresentação para execução do algoritmo construtivo aleatório.....	153
Figura 100 - Janela dos resultados para o algoritmo construtivo aleatório sobre o problema <i>berlin52.tsp</i> .....	154
Figura 101 - Gráfico da metaheurística <i>simulated annealing</i> aplicado sobre o problema <i>berlin52.tsp</i> , após a execução construtiva aleatória e da aplicação da heurística de melhoramento 2-Opt.....	155
Figura 102 - Gráfico da metaheurística busca tabu aplicado sobre o problema <i>berlin52.tsp</i> , após a execução construtiva aleatória e da aplicação da heurística de melhoramento 2-Opt .....	156
Figura 103 - Apresentação da simulação passo-a-passo da heurística do vizinho mais próximo.....	157
Figura 104 - Simulação passo-a-passo da heurística vizinho mais próximo sobre o problema <i>berlin52.tsp</i> .....	159
Figura 105 - Simulação de uma rodada 2-Opt passo-a-passo sobre o problema <i>berlin52.tsp</i> .....	160
Figura 106 - Exibição da rota ótima para o problema <i>berlin52.tsp</i> .....	161
Figura 107 - Diferentes visões para o mesmo problema .....	162

## **LISTA DE TABELAS**

<b>TABELA 1 - NÚMERO DE SOLUÇÕES POSSÍVEIS PARA O PROBLEMA DA MOCHILA.....</b>	<b>22</b>
<b>TABELA 2 - A EXPLOSÃO COMBINATÓRIA DO PROBLEMA DO CAIXEIRO VIAJANTE .....</b>	<b>28</b>
<b>TABELA 3 - QUALIDADE DAS SOLUÇÕES X TEMPO COMPUTACIONAL.....</b>	<b>152</b>

## LISTA DE ABREVIATURAS

AMEM – Ambiente Multimídia para Educação Mediada por Computador  
AVEA – Ambiente Virtual de Ensino-Aprendizagem  
LOBO – *Learning Object Based on Optimization*  
BFS – *Breadth first search*  
DA – Desafio Mais Amplo  
DC – *Dublin Core Metadata Initiative*  
DFS – *Depth first search*  
DI – Desafio Inicial  
HTML – *Hyper Text Markup Language*  
IAE – Investigação-Ação Educacional  
IMS – *Instructional Management Systems*  
JRE – *Java Run-time Environment*  
JSP – *Java Servlets Pages*  
JWS – *Java Web Start*  
LOM – *Learning Objects Metadata*  
LTSC – *Learning Technology Standards Committee*  
MDP – Matriz Dialógica Problematizadora  
MSEM – Melhor Solução Educacional no Momento  
PCV – Problema do Caixeiro Viajante  
PEDP – Prática Educacional Dialógica-Problematizadora  
RIVED – Rede Internacional Virtual de Educação  
SCORM – *Sharable Content Object Reference Model*

# SUMÁRIO

<b>1</b>	<b>APRESENTAÇÃO.....</b>	<b>15</b>
<b>2</b>	<b>O ENSINO DE HEURÍSTICAS E METAHEURÍSTICAS .....</b>	<b>20</b>
2.1	O QUE É UM PROBLEMA DE OTIMIZAÇÃO COMBINATÓRIA? .....	20
2.2	O PROBLEMA DO CAIXEIRO VIAJANTE .....	23
2.2.1	<i>Definição</i> .....	23
2.2.2	<i>Formulação matemática</i> .....	25
2.2.3	<i>Aplicações do PCV</i> .....	25
2.2.4	<i>O PCV como Tema Gerador</i> .....	26
2.3	SOLUÇÕES CLÁSSICAS PARA O PROBLEMA DO CAIXEIRO VIAJANTE .....	27
2.3.1	<i>Algoritmos exatos</i> .....	27
2.3.2	<i>Heurísticas</i> .....	31
2.3.3	<i>Metaheurísticas</i> .....	34
2.4	HEURÍSTICAS E METAHEURÍSTICAS COMO OBJETO DE ESTUDO.....	36
2.4.1	<i>Ensino superior de Heurísticas e Metaheurísticas</i> .....	36
2.4.2	<i>Problemas e dificuldades</i> .....	36
2.4.3	<i>A Escolha do Problema do Caixeiro Viajante</i> .....	39
2.5	CONCLUSÕES DO CAPÍTULO .....	40
<b>3</b>	<b>ENSINO-APRENDIZADO DIALÓGICO-PROBLEMATIZADOR.....</b>	<b>42</b>
3.1	PAULO FREIRE E A PROBLEMATIZAÇÃO DA REALIDADE.....	44
3.2	INVESTIGAÇÃO-AÇÃO E A EDUCAÇÃO DIALÓGICA PROBLEMATIZADORA .....	45
3.3	A MATRIZ DIALÓGICA PROBLEMATIZADORA.....	48
3.4	CONCLUSÕES DO CAPÍTULO .....	50
<b>4</b>	<b>AMBIENTES VIRTUAIS DE ENSINO APRENDIZAGEM SOB A ÓTICA DE SUA CONCEPÇÃO PEDAGÓGICA.....</b>	<b>53</b>
4.1	O QUE É UM AVEA?.....	53
4.2	AMEM 2.0 – Um AVEA DIALÓGICO-PROBLEMATIZADOR.....	56
4.2.1	<i>Modelagem da Interface Usuário/Ambiente</i> .....	57
4.2.2	<i>Modelos navegacionais</i> .....	93
4.2.3	<i>Aspectos Tecnológicos</i> .....	95
4.3	CONCLUSÕES DO CAPÍTULO .....	99
<b>5</b>	<b>OBJETOS DE APRENDIZAGEM.....</b>	<b>102</b>
5.1	O QUE SÃO OBJETOS DE APRENDIZAGEM?.....	102
5.2	CATALOGAÇÃO E BUSCA DE OBJETOS DE APRENDIZAGEM: PROPOSTAS E INICIATIVAS .....	106
5.2.1	<i>Padrões de metadados</i> .....	107
5.2.2	<i>Empacotadores de objetos</i> .....	113
5.3	CONCLUSÕES DO CAPÍTULO.....	118
<b>6</b>	<b>LOBO – UM OBJETO DE APRENDIZAGEM PARA O ENSINO DE HEURÍSTICAS E METAHEURÍSTICAS .....</b>	<b>119</b>
6.1	MENU PRINCIPAL .....	120
6.1.1	<i>Sub-Menu: O Problema</i> .....	120
6.1.2	<i>Sub-Menu: Algoritmos</i> .....	123
6.1.3	<i>Sub-Menu: Dados de Teste</i> .....	129
6.1.4	<i>Sub-Menu: Sobre</i> .....	133
6.2	ALGORITMOS IMPLEMENTADOS .....	134
6.2.1	<i>Heurística Construtiva: Algoritmo Aleatório</i> .....	134
6.2.2	<i>Heurística construtiva: Vizinho Mais Próximo</i> .....	135

6.2.3	<i>Heurística construtiva: Inserção Mais Distante</i> .....	137
6.2.4	<i>Heurísticas de melhoria: 2-Opt e 3-Opt</i> .....	139
6.2.5	<i>Metaheurística: Busca Tabu</i> .....	143
6.2.6	<i>Metaheurística: Simulated Annealing</i> .....	144
6.3	HISTÓRICO DOS RESULTADOS .....	147
6.4	BOTÕES DE AÇÃO .....	152
6.4.1	<i>Botão: Executar a Heurística</i> .....	153
6.4.2	<i>Botão: Simulação Passo-a-Passo</i> .....	156
6.4.3	<i>Exibir a Rota Ótima</i> .....	161
6.4.4	<i>Tutorial do LOBO</i> .....	162
6.5	ÁREA DE SIMULAÇÃO .....	162
6.6	ASPECTOS TECNOLÓGICOS .....	163
6.7	CONCLUSÕES DO CAPÍTULO.....	163
<b>7</b>	<b>AS MEDIAÇÕES TECNOLÓGICAS AMEM-LOBO NA PRÁTICA</b> .....	<b>166</b>
7.1	CONCEPÇÃO E METODOLOGIA.....	167
7.1.1	<i>Apresentação da Disciplina</i> .....	167
7.1.2	<i>Inteligência Artificial e suas Inter-relações com a Sociedade da Informação</i> .....	169
7.1.3	<i>Complexidade Computacional</i> .....	172
7.1.4	<i>O Problema do Caixeiro Viajante</i> .....	174
7.2	DESENVOLVIMENTO DE HEURÍSTICAS CONSTRUTIVAS.....	176
7.2.1	<i>Heurísticas Construtivas: Introdução e Exemplos</i> .....	176
7.2.2	<i>Heurísticas Construtivas: Vizinho Mais Próximo e Algoritmos de Inserção</i> .....	178
7.2.3	<i>Heurísticas Construtivas: Algoritmos de Inserção e Heurística das Economias</i> .....	181
7.3	DESENVOLVIMENTO DE HEURÍSTICAS DE MELHORAMENTO.....	184
7.3.1	<i>Heurísticas de Melhoria: Introdução e Exemplos</i> .....	184
7.3.2	<i>Heurísticas de Melhoria: 2-Opt, 3-Opt e Or-Opt</i> .....	185
7.4	DESENVOLVIMENTO DE METAHEURÍSTICAS .....	188
7.4.1	<i>Metaheurísticas: Introdução</i> .....	188
7.4.2	<i>Metaheurísticas: Simulated Annealing</i> .....	191
7.4.3	<i>Metaheurísticas: Busca Tabu</i> .....	193
7.4.4	<i>Metaheurísticas: Algoritmos Genéticos</i> .....	193
7.4.5	<i>Metaheurísticas: Operadores de Recombinação e Mutação para o PCV</i> .....	195
7.4.6	<i>Aulas Práticas</i> .....	198
7.5	RESPONDENDO À MATRIZ DIALÓGICA-PROBLEMATIZADORA .....	198
<b>8</b>	<b>CONCLUSÕES</b> .....	<b>206</b>
<b>9</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>211</b>
	<b>APÊNDICE A – O AMBIENTE AMEM 2.0 E O OBJETO DE APRENDIZAGEM LOBO – DISTRIBUIÇÃO OPEN SOURCE</b> .....	<b>225</b>
	<b>APÊNDICE B – LISTA DE PUBLICAÇÕES DO AUTOR GERADAS DURANTE O DESENVOLVIMENTO DA TESE</b> .....	<b>226</b>

# 1 Apresentação

A educação contemporânea se preocupa cada vez mais com os processos de ensino-aprendizagem que passem da simples *reprodução* do conhecimento para, efetivamente, sua *produção*. Desta forma, o conhecimento científico que deve ser levado ao educando é, essencialmente, uma representação da realidade como ela se apresenta agora, podendo ser alterada constantemente durante o desenvolver do ser humano. Posto isso, é claro afirmar que os desafios dos professores também são aumentados.

As globalizações, tanto tecnológica como comunicacional, também trazem novos desafios e oportunidades. Este movimento econômico e social contribui de forma decisiva na geração dos novos paradigmas interpessoais e profissionais, servindo como base para a modificação dos nossos conceitos acerca do que é informação, como ela se propaga e suas conseqüências na empregabilidade. Não há consenso sobre onde estaremos daqui a dez anos, mas não há como fugirmos da responsabilidade de preparar nossos alunos para poderem mudar com estas transformações que continuarão a surgir nos próximos anos. Novamente, mais importante do que tentar prever com exatidão o que está por acontecer, saber reconhecer as mudanças e adaptar seu conhecimento às novas informações é uma habilidade cada vez mais necessária e primordial na Sociedade da Informação.

A tecnologia mediática e a própria educação estão cada vez mais juntas neste processo, seja por trazer as novas possibilidades de exploração da informação para dentro da sala de aula, seja por que a própria ferramenta computacional se torna cada dia mais indispensável para a grande maioria das profissões. Neste contexto, a *Informática na Educação* tem representado uma revolução na educação tradicional, nas políticas de educação públicas e nos formatos e metodologias propostas para a interação educando – educador. A utilização de suportes telemáticos para as aulas presenciais, semi-presenciais ou a distância é uma alternativa que se torna cada dia mais interessante, tanto em termos tecnológicos como no próprio processo de ensino-aprendizagem.

No entanto, a mera adição de suportes tecnológicos à sala de aula não contribui de forma transformadora, constituindo-se, em muitos casos, de um mero acessório ou de objetos de propaganda institucionais. Novas tecnologias e aparelhagens sofisticadas não são garantia de desempenho acadêmico, seja qual instrumento de avaliação for utilizado. A verdadeira transformação está na mudança metodológica e na avaliação contínua dos professores sobre estes métodos. Dentro deste contexto, as ferramentas computacionais podem ser utilizadas de forma realmente efetiva.

Ou seja, o investimento tecnológico só se justifica caso o objetivo final da implementação do sistema seja plenamente satisfatório, ou seja, a efetiva *aprendizagem* do educando. Além disso, evidencia-se que o processo de ensino precisa mudar para um processo educativo, uma vez que, segundo Moran, “ensino e educação são conceitos diferentes. No ensino organiza-se uma série de atividades didáticas para ajudar os alunos a compreender áreas específicas do conhecimento (ciências, história, matemática). Na educação o foco, além de ensinar é ajudar a integrar ensino à vida, conhecimento e ética, reflexão e ação, a ter uma visão da totalidade”. (MORAN, 2000).

Ainda em relação ao método, é interessante observar que (POPPER, 1983) afirma que não é possível confirmar a veracidade de uma teoria pela simples constatação de que os resultados oriundos de uma previsão foram previstos pela teoria em questão. Para ele,

ciência só é realizada através de teorias que podem ser *falsificáveis*, ou seja, cuja contraprova possa ser encontrada. Teorias que não podem ser refutadas pertencem à metafísica ou outras especulações. Também afirma que toda a verdade científica é provisória, pois nunca poderemos chegar à verdade, e sim, somente nos aproximar dela.

Diversas metodologias têm sido propostas com a intenção de modificar as práticas pedagógicas dos professores, orientando-os a conduzirem aulas que estejam abertas para os problemas do mundo real e com as transformações dos próprios conteúdos que estão sendo desenvolvidos. A Metodologia Histórico-Crítica (SAVIANI, 1991), a Pedagogia da Alternância (GIMONET, 1998 *apud* BEGNAMI, 2003) e a Metodologia Inovadora (BEHRENS, 1999 *apud* RODRIGUES *et al.*, 2003) são exemplos claros destas preocupações.

No âmbito deste projeto, será destacada a prática educacional orientada pelas teorias da Investigação-Ação Educacional (IAE) e a Prática Educacional Dialógica-Problematizadora (PEDP) (CARR, KEMMIS, 1986). A Investigação-Ação Educacional depende de uma seqüência de julgamentos e ações que constituem as etapas do ciclo de uma espiral, com quatro fases: (a) planejamento: etapa antecessora a ação propriamente dita, onde é necessário refletir sobre a situação educativa, sua complexidade e importância, construindo uma base para as ações futuras; (b) ação: guiada pelo planejamento realizado anteriormente sem, no entanto, incorrer no erro comum de utilizar o planejamento como um guia estático e imutável. A ação deve possuir um propósito criticamente informado; (c) observação: documentação dos efeitos da ação, gerando uma base para a reflexão. A documentação contribui para a melhoria contínua da prática, através da análise da situação contextualizada, o que se pode traduzir em uma ação estratégica mais crítica; (d) reflexão: finalmente, a reflexão tenta interpretar, discursivamente, os acontecimentos oriundos das ações, propondo modificações ao planejamento das mesmas, face às evidências observadas, reconstruindo uma nova ação informada.

Já a Prática Educacional Dialógica-Problematizadora interpreta que a ação pode ser construída metodologicamente através dos seguintes momentos (ANGOTTI, DELIZOICOV, 1990): problematização inicial, organização do conhecimento e aplicação do conhecimento. A problematização inicial é apresentada como um desafio na forma de questões e/ou situações que devem ser discutidas e debatidas pelos alunos. No próximo momento, na organização do conhecimento, o educador deve orientar a sistematização do mesmo, para a completa compreensão do tema e do problema apresentado. Este conhecimento deverá ser usado pelo aluno para analisar o problema inicial e generalizá-lo para outras situações que são explicadas pelo mesmo conhecimento, o que caracteriza o terceiro momento, da aplicação do conhecimento. Estes três momentos foram reorganizados pelo grupo de pesquisa (MÜLLER, DE BASTOS, 2004) para Desafio Inicial, Melhor Solução Educacional no Momento e Desafio Mais Amplo, o que será discutido com mais profundidade no capítulo 3.

Esta metodologia pode ser aplicada em todas as áreas do conhecimento e em todos os graus de ensino, desde os primeiros anos da educação básica até o ensino universitário. O projeto desta tese é delimitado por estas duas características, centrando seus esforços na elaboração de um diálogo problematizador entre alunos e professores, através de um meio telemático, para o ensino de heurísticas e metaheurísticas na educação superior, mais precisamente, na graduação.

Em relação ao tema de ensino, convém destacar que, no ensino de Engenharia, Matemática e Ciência da Computação, a área de heurísticas e metaheurísticas para solução de problemas de otimização combinatória é abordada com ênfase. Em formas gerais, a otimização tem como objetivo a resolução da alocação de recursos, tipicamente limitados, com o intuito de alcançar determinados objetivos. Considerando que existe um conjunto discreto de soluções possíveis, a resolução de um problema de otimização combinatória inclui o processo de geração, avaliação e comparação de soluções, num determinado limite de tempo. Conforme (CORNE *et al.*, 1999), a otimização é um tópico central nas áreas da ciência da computação, inteligência artificial e pesquisa operacional.

Já uma heurística pode ser definida como um algoritmo que encontra uma solução factível, não necessariamente a melhor solução – apesar de poder encontrá-la – para um determinado problema com uma determinada função objetivo, num tempo computacional razoável (DIAZ, 1996). Estas heurísticas também podem ser facilmente adaptáveis a outros tipos de problemas envolvendo as áreas de Inteligência Artificial, Redes de Computadores e Arquitetura de Computadores. No entanto, somente a aplicação de heurísticas pode não resolver todas as classes de problemas. As metaheurísticas, que se caracterizam por guiarem outras heurísticas, têm sido particularmente interessantes na resolução de problemas de grandes dimensões. Em relação às metaheurísticas, podem-se citar como principais métodos o *simulated annealing* (KIRKPATRICK *et al.* 1983), a busca tabu (GLOVER, LAGUNA, 1997) e os algoritmos genéticos (GOLDBERG, 1989).

Este campo do saber é explorado nas mais diversas áreas das ciências naturais e tecnológicas, com especial ênfase nas engenharias e na computação. Desde 1997, tenho atuado como docente em cursos de graduação em diversas instituições de ensino no Rio Grande do Sul, desenvolvendo uma carreira centrada no ensino de disciplinas que trabalham com heurísticas e metaheurísticas. Ministrando as cadeiras de *Inteligência Artificial* na URCAMP, *Sistemas de Informação Inteligentes e Inteligência Artificial* na UNIFRA e, em conjunto com meu orientador, *Inteligência Artificial* na UFSM, esta trajetória pessoal me permitiu conhecer, com razoável profundidade, as dificuldades de trabalhar com o ensino de heurísticas e metaheurísticas, que podem ser explicados sob três aspectos:

- aspecto técnico: o ensino tradicional de heurísticas e metaheurísticas trabalha com a apresentação dos algoritmos clássicos em sala de aula, usualmente através de testes de mesa, que se caracteriza pela *execução*, passo-a-passo, da heurística, realizando todos os cálculos necessários. No entanto, devido ao grande número de passos necessários para executar completamente uma heurística, o professor é obrigado a trabalhar com exemplos de menor dimensão, o que atrapalha a demonstração do algoritmo. Problemas pequenos podem ser resolvidos na otimalidade rapidamente, mesmo com as heurísticas mais simples e, desta forma, a aplicação de heurísticas mais elaboradas perde o sentido. Os alunos, usualmente, tendem a perder o interesse pelas demonstrações passo-a-passo logo após as primeiras aulas, pois o resultado final é praticamente invariante;
- aspecto motivacional: apesar da aplicabilidade prática das heurísticas e metaheurísticas (capítulo 2) ser um aspecto que prende a atenção dos estudantes nas primeiras aulas, a condução da disciplina através da simples demonstração de algoritmos e execução dos mesmos acaba por frustrar as expectativas iniciais

dos estudantes. Ao se trabalhar com números puros, sem significado ou objetivo além da prova de uma teoria, afastamo-nos do aluno. Para trazer o aluno para perto, é necessário dialogar com o mesmo e, para tanto, devemos dar e trabalhar o significado do que está sendo desenvolvido em sala de aula;

- aspecto didático-metodológico: ensinar heurísticas e metaheurísticas é muito mais do que simplesmente apresentar as soluções para um problema. Esta área é baseada, principalmente, na construção de idéias novas para problemas reais, normalmente traçando paralelos com outros saberes, como demonstram os Algoritmos Genéticos, o *Simulated Annealing* e o *Ant Colony* (DORIGO, STÜZLE, 2004). Para incentivar a criatividade dos alunos acerca dos problemas trabalhados, é necessário que eles compreendam o fenômeno e discutam sobre o mesmo. Não há discussão ou argumentação em uma sala de aula tradicional, baseada simplesmente no quadro-negro e giz e na figura autoritária do professor. Não há comunicação que não seja baseada no diálogo, guiado e conduzido pelo professor, mas participativo e estimulado pelas interações entre todos os envolvidos.

Também é importante considerar que, tradicionalmente, o ensino de conteúdos das áreas de engenharia e/ou tecnologia tem sido construído através da transposição dos objetivos e estratégias de resolução, matéria-prima de formação de um engenheiro, em fórmulas e ou algoritmos que podem ser construídos, reproduzidos e confirmados por algum tipo de experiência prática. Diversos autores consideram que a engenharia é a filosofia de resultados, ou seja, a condução dos processos do fazer engenharia deve, necessariamente, levar a algo, físico ou não, que pode ser comprovado e testado de alguma forma.

No entanto, um profissional da área tecnológica ou engenharia não pode ser unicamente *tarefeiro*, ou seja, ele não deve somente ser um solucionador de problemas pois, neste caso, estaríamos formando um profissional incompleto, incapaz de questionar a si ou os métodos que estão sendo empregados, deixando de ser um graduado e se tornando apenas um técnico altamente especializado.

Considerando todo este preâmbulo, o objetivo desta tese é modelar, implementar e testar um objeto de aprendizagem, em conjunto com um ambiente de suporte educacional e uma metodologia pedagógica, de forma a superar as dificuldades do ensino de heurísticas e metaheurísticas discutidas anteriormente. Desta forma, define-se como principal questão de pesquisa se é possível trabalhar o ensino de heurísticas e metaheurísticas mediado por objetos de aprendizagem e na perspectiva da investigação-ação educacional e da concepção educacional dialógico-problematizadora?

Metodologicamente, este trabalho constituiu no estudo do problema em questão, tanto na forma da disciplina propriamente dita como nas questões pedagógicas relacionadas ao ensino da mesma. A partir da concepção da metodologia, foi modelado e implementado um ambiente virtual de ensino-aprendizagem que desse suporte a esta metodologia específica, bem como a especificação de um objeto de aprendizagem específico para a área de heurísticas e metaheurísticas. Finalmente, com a concepção metodológica consolidada, o ambiente virtual em operação e o objeto sendo utilizado pelos alunos, foi construído todo um plano de ensino que operacionalizasse as opções pedagógicas dos professores, potencializadas pelas ferramentas disponíveis.

Vários objetivos específicos foram definidos para este processo:

- estabelecer princípios metodológicos para o ensino de heurísticas e metaheurísticas. Através da adoção da investigação ação-educacional e adotando uma postura que permita e incentive o diálogo-problematizador na sala de aula, esperamos manter os alunos motivados e interessados na disciplina, além de prover mecanismos que permitam uma auto-avaliação das decisões tomadas pelo docente;
- verificar o impacto do ambiente informatizado na condução das disciplinas. A utilização de um arcabouço tecnológico que auxilie na comunicação e no desenvolvimento da disciplina pode trazer ganhos tanto na investigação das práticas educacionais, pela manutenção de relatórios e registros das atividades docentes e discentes, quanto na organização da disciplina em si. O AMEM (apresentado no capítulo 3) permite que as aulas, atividades e colaborações sejam planejadas e disponibilizadas com antecedência, formando um canal de comunicação entre o professor e os alunos;
- verificar o impacto do objeto de aprendizagem na consolidação da disciplina de heurísticas e metaheurísticas. O desenvolvimento do LOBO (capítulo 6) tenta suprir o aspecto técnico apresentado anteriormente, permitindo que exemplos de maiores dimensões sejam apresentados aos alunos em uma interface gráfica de simulação. Através da utilização do objeto em sala de aula e disponibilizando o mesmo para os alunos trabalharem fora do ambiente escolar, espera-se que a compreensão sobre as diferenças entre os algoritmos mais simples e mais complexos, bem com os pontos fortes e pontos fracos de cada um, se tornem mais consolidados;
- trabalhar com um padrão que permita a utilização do objeto de aprendizagem em outras plataformas. Não há sentido em desenvolver um objeto de aprendizagem que seja executado unicamente em uma plataforma. Conceitualmente, *objetos de aprendizagem* podem e devem ser compartilhados entre diferentes professores e alunos. Para tanto, é necessário conhecer os padrões de disponibilização de objetos de aprendizagem, para que o mesmo possa ser concebido de forma a interagir com outros ambientes, além do AMEM.

O trabalho está sistematizado de forma a mostrar o pensamento organizacional de todo o projeto desenvolvido. Inicialmente, no capítulo 2, é apresentado o problema, enfatizando o que são heurísticas e metaheurísticas, quais são os problemas mais comuns da área e traçando uma discussão sobre como o ensino nesta área é, usualmente, abordado. O capítulo 3 apresenta a metodologia pedagógica que norteou o desenvolvimento do Ambiente Virtual de Ensino-Aprendizagem(AVEA) e a preparação das aulas. O capítulo 4 desenvolve questões relativas à modelagem e implementação do AVEA, destacando os ambientes já existentes e a necessidade de implementar um novo ambiente que se adequasse às escolhas pedagógicas. O capítulo 5 apresenta os objetos de aprendizagem focando, principalmente, na catalogação e busca de informações nestas ferramentas, que servem como base para implementação do objeto, que é apresentada no capítulo 6. Finalmente, o capítulo 7 apresenta as conclusões desta tese.

## 2 O Ensino de Heurísticas e Metaheurísticas

O ensino de heurísticas e metaheurísticas está centralizado no desenvolvimento de algoritmos e técnicas para a resolução de problemas de otimização combinatória. Neste capítulo, serão explorados os conceitos acerca desta área da matemática, destacando principalmente, o Problema do Caixeiro Viajante. Com base nestes conceitos, é possível realizar uma discussão sobre como esta disciplina pode ser trabalhada por professores e alunos.

### 2.1 O que é um Problema de Otimização Combinatória?

O termo *otimização combinatória* é um ramo da matemática e da ciência da computação que analisa problemas de otimização em conjuntos, em geral dentro de um limite de tempo. Um *problema de otimização* é um problema para o qual temos diferentes soluções possíveis. Tais soluções podem ser avaliadas e comparadas, efetuando a otimização. Por *otimizar*, se entende encontrar um valor ótimo para um determinado problema, usualmente sob determinadas restrições. Os *conjuntos* representam agrupamentos de itens que só podem ser selecionados através de uma série de regras (*restrições do problema*), formando subconjuntos. Este agrupamento de itens específicos na forma de subconjuntos é denominado uma possível *solução do problema* e, invariavelmente, possui algum *custo* associado. Desta forma, o objetivo central da otimização combinatória é encontrar um subconjunto cujo custo seja mínimo.

A idéia mais simples, e ingênua, para resolver um problema de otimização combinatória é simplesmente *combinar*, ou seja, enumerar, todas as possíveis soluções. Em outras palavras, criar todos os subconjuntos existentes a partir do *conjunto* e das *regras de restrição* e escolher o de menor custo. No entanto, isto se torna impraticável para aplicações práticas, como é fácil provar. Imagine, por exemplo, que o problema a ser resolvido é separar alguns elementos de um determinado conjunto. Cada elemento possui um peso e um valor associado. Estes elementos serão armazenados em uma caixa que possui uma restrição em relação ao peso que ela pode carregar. Deseja-se, desta forma, encontrar o subconjunto de elementos que maximize o valor que será transportado sem que a *restrição* de peso da caixa seja violada.

Considerando que existam cinco itens no conjunto, listado abaixo, e que o peso máximo que a caixa suporta é 10 kg.

A [valor = 10, peso = 3]  
B [valor = 3, peso = 4]  
C [valor = 7, peso = 6]  
D [valor = 9, peso = 4]  
E [valor = 2, peso = 2]

Uma forma de resolver este problema é testar todas as combinações possíveis, como apresentado anteriormente. Desta forma, para o problema anterior, temos as seguintes possibilidades.

S1.	[A]	v.10	p.3	<b>S.17</b>	<b>[A, B, D]</b>	<b>v.22</b>	<b>p.11</b>
S2.	[B]	v.3	p.4	S.18	[A, B, E]	v.15	p.9
S3.	[C]	v.7	p.6	<b>S.19</b>	<b>[A, C, D]</b>	<b>v.26</b>	<b>p.13</b>
S4.	[D]	v.9	p.4	<b>S.20</b>	<b>[A, C, E]</b>	<b>v.19</b>	<b>p.11</b>
S5.	[E]	v.2	p.2	S.21	[A, D, E]	v.21	p.9
S6.	[A, B]	v.13	p.7	<b>S.22</b>	<b>[B, C, D]</b>	<b>v.19</b>	<b>p.14</b>
S7.	[A, C]	v.17	p.9	<b>S.23</b>	<b>[B, C, E]</b>	<b>v.12</b>	<b>p.12</b>
S8.	[A, D]	v.19	p.7	S.24	[B, D, E]	v.14	p.10
S9.	[A, E]	v.12	p.5	<b>S.25</b>	<b>[C, D, E]</b>	<b>v.18</b>	<b>p.12</b>
S10.	[B, C]	v.10	p.10	<b>S.26</b>	<b>[A, B, C, D]</b>	<b>v.29</b>	<b>p.17</b>
S11.	[B, D]	v.12	p.8	<b>S.27</b>	<b>[A, B, C, E]</b>	<b>v.22</b>	<b>p.15</b>
S12.	[B, E]	v.5	p.6	<b>S.28</b>	<b>[A, B, D, E]</b>	<b>v.24</b>	<b>p.13</b>
S13.	[C, D]	v.16	p.10	<b>S.29</b>	<b>[A, C, D, E]</b>	<b>v.28</b>	<b>p.15</b>
S14.	[C, E]	v.9	p.8	<b>S.30</b>	<b>[B, C, D, E]</b>	<b>v.21</b>	<b>p.16</b>
S15.	[D, E]	v.11	p.6	<b>S.31</b>	<b>[A, B, C, D, E]</b>	<b>v.31</b>	<b>p.19</b>
<b>S.16</b>	<b>[A, B, C]</b>	<b>v.20</b>	<b>p.13</b>				

As soluções em negrito não satisfazem a restrição do problema, pois o peso está acima da capacidade da caixa (10 kg). Das demais, a solução ótima do problema é a de número 21, que possui o valor de 21, para um peso de 9 kg. Este é um clássico problema de otimização combinatória, denominado *problema da mochila* (DIAZ *et al.*, 1996), cujo número de soluções é dado por:

$$num\_solucoes = \sum_{i=1}^{n-1} C_{n,i} + 1$$

onde  $n$  é o número de elementos no conjunto e  $C$  representa a operação matemática de combinação. Sabendo que a combinação de  $n$  elementos de um determinado conjunto tomados  $p$  a  $p$ , sem importar a ordem, é dada por:

$$C_{n,p} = \frac{n!}{p!(n-p)!}$$

o número de soluções possíveis para o problema apresentado é calculado como

$$num\_solucoes = \frac{5!}{1!(5-1)!} + \frac{5!}{2!(5-2)!} + \frac{5!}{3!(5-3)!} + \frac{5!}{4!(5-4)!} + 1$$

$$num\_solucoes = \frac{120}{24} + \frac{120}{12} + \frac{120}{12} + \frac{120}{24} + 1$$

$$num\_solucoes = 5 + 10 + 10 + 5 + 1 = 31$$

No entanto, como se trata de um problema *combinatório*, o número de soluções aumenta rapidamente à medida que o tamanho do problema (número de itens a serem alocados) também aumenta. Esta observação pode ser constatada na tabela 1.

Valor de $n$	Número de soluções	Valor de $n$	Número de soluções
5	31	15	32767
6	63	20	1048575

7	127	25	$3.355 * 10^7$
8	255	30	$1.070 * 10^9$
9	511	35	$3.435 * 10^{10}$
10	1023	40	$1.099 * 10^{12}$
11	2047	45	$3.518 * 10^{13}$
12	4095	50	$1.125 * 10^{15}$
13	8191	100	$1.269 * 10^{30}$
14	16383	150	$1.427 * 10^{45}$

Tabela 1 - Número de soluções possíveis para o problema da mochila

Mesmo utilizando computadores de grande capacidade de processamento, a enumeração de todas as possíveis soluções para um problema razoável se torna completamente inviável. É, portanto, necessário resolver os problemas de otimização combinatória através de outras técnicas, ditas *mais inteligentes*, que forneçam atalhos para a descoberta dos valores ótimos para os problemas propostos ou, pelo menos, boas soluções de cunho geral.

A importância dos problemas de otimização combinatória reside na sua aplicação prática. Usualmente, se denomina *Pesquisa Operacional* o conjunto de modelos, técnicas, algoritmos e esforços que centralizam sua atenção no desenvolvimento de respostas eficientes para os problemas de otimização. Este ramo de pesquisa é de fundamental importância para as organizações.

*A utilização do ferramental da “Pesquisa Operacional” na promoção da eficiência e eficácia organizacional em todos os níveis da gestão é uma realidade tornada viável pelo microcomputador e pelo avanço do estado da arte. De fato, a relevância da área de sistemas de apoio à decisão vem crescendo com o advento das estações de trabalho, que a um baixo custo, e com maior proximidade física dos executivos, oferecem capacidade de cálculo, de armazenamento e recursos gráficos antes disponíveis apenas em máquinas de grande porte.*

(GOLDBARG, LUNA, 2000)

Existem diversos tipos de problemas de otimização combinatória, além do *problema da mochila* apresentado neste capítulo, que podem ser utilizados para o desenvolvimento de sistemas de carga de containeres, carregamento de caminhões de entrega, investimento de capital e corte e empacotamento. Entre os principais problemas matemáticos de aplicação prática, é possível citar (GOLDBARG, LUNA, 2000):

- problema de particionamento, utilizado na recuperação de informações em bancos de dados, alocação de tripulações em linhas aéreas, distribuição do tráfego de comunicações em satélites, alocação de serviços de emergência; roteamento de petroleiros, entre inúmeros outros;
- problema da árvore geradora mínima: utilizado no projeto de redes de comunicações, roteamento de meios terrestres na defesa da costa, roteamento de veículos com função multiobjetivo (envolvendo, além da distância, pedágios, condições de risco e quaisquer outras variáveis importantes para o modelo);

- problema de roteamento de veículos: utilizado para definir escalas de tripulação, programação de tarefas e tripulação, rotear veículos com restrições (combustível, janelas de tempo, congestionamento);
- problemas de fluxo: utilizado principalmente em redes de transporte de energia, contribuindo no planejamento da expansão e distribuição do sistema, telecomunicações e escala de motoristas;
- problema do caixeiro viajante: utilizado, principalmente, na descoberta de rotas de custo mínimo.

Ao observar a lista acima, é viável perceber a importância dos modelos de pesquisa operacional, tanto sob os aspectos práticos como teóricos. A necessidade premente da melhoria dos processos de negócio e gestão atuais, aliados às possibilidades de redução de custos, principalmente nas atividades de manufatura e gestão, torna os problemas de otimização combinatória uma área em expansão para os países em desenvolvimento.

Para âmbito desta tese, define-se como tema de ensino as heurísticas e metaheurísticas. No entanto, devido ao grande número de problemas matemáticos que podem ser abordados, foi escolhido um problema-tipo, o Problema do Caixeiro Viajante, que será aplicado durante todo o decorrer da disciplina através de problematizações mediadas pelo apoio de tecnologia. No próximo capítulo, analisaremos com especial cuidado o PCV, uma subárea da pesquisa operacional de grande aplicação prática e que possui uma relação estreita com outros modelos.

## 2.2 O Problema do Caixeiro Viajante

### 2.2.1 Definição

Um caixeiro viajante recebe a informação de um conjunto de cidades e um custo  $c_{ij}$  associado a cada par de cidades  $i$  e  $j$  deste conjunto. Usualmente, este custo representa a distância de partir da cidade  $i$  até a cidade  $j$ . O caixeiro deve partir de uma cidade inicial, passar por todas as demais somente uma vez e retornar à cidade de partida. O problema consiste em encontrar uma trajetória que percorra o menor caminho possível.

Em relação ao custo  $c_{ij}$ , apesar da distância ser a variável mais comum, pode haver variações e outras informações podem ser quantificadas neste custo, tais como pedágios ou condições da rodovia – por exemplo, uma rodovia sem asfalto pode ser penalizada com o acréscimo de uma distância fixa ou proporcional ao trecho.

Esta trajetória, que passa somente uma vez em cada cidade, também é conhecida como *Ciclo Hamiltoniano*, que possui a seguinte definição:

*O grafo... é **Hamiltoniano** se possuir um **Ciclo Hamiltoniano** (circuito), isto é, se admitir um caminho indireto (direto) fechado que passe em cada vértice exatamente uma vez.*

(CAMPELLO, MACULAN,1994)

Os ciclos hamiltonianos são creditados a *Sir William Hamilton*, que investigava a existência destes ciclos em um dodecaedro regular (um poliedro que possui 12 faces

pentagonais e 20 vértices), conforme a figura 1, que apresenta o dodecaedro. A figura 2 apresenta o grafo de Hamilton (a) e um possível ciclo hamiltoniano (b) para este problema.

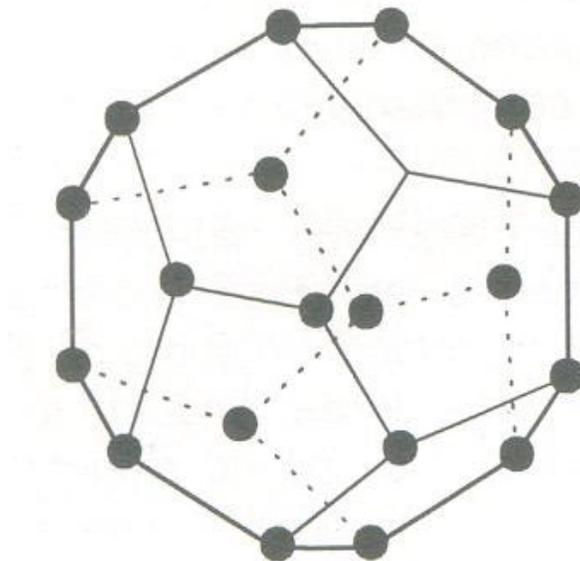


Figura 1 - Dodecaedro regular – adaptado de (CAMPELLO, MACULAN, 1994)

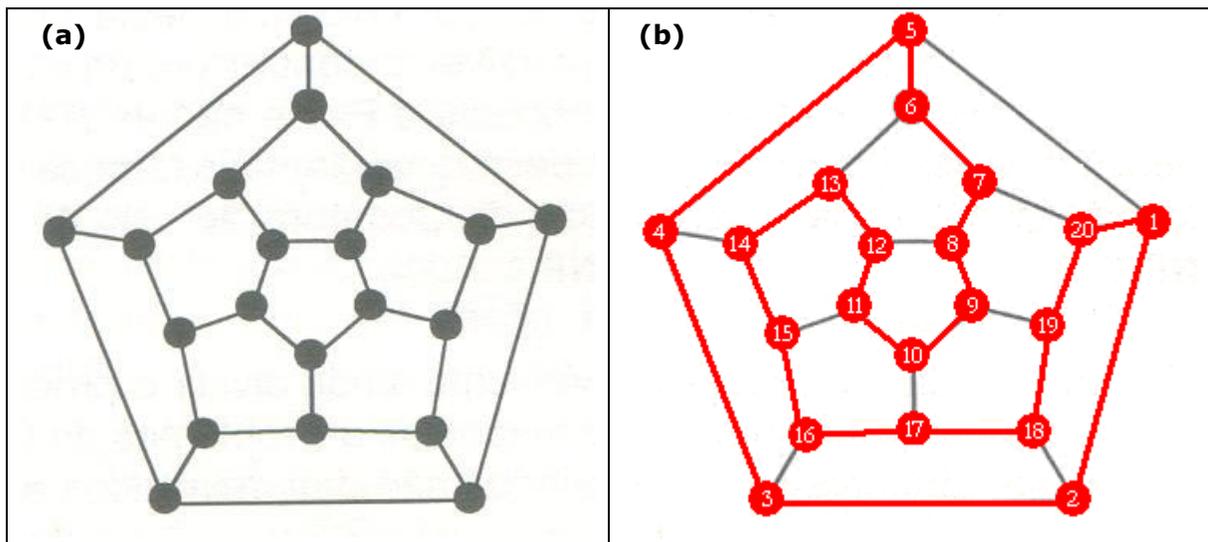


Figura 2 - Grafo Hamiltoniano e um possível ciclo hamiltoniano para o problema – adaptado de (CAMPELLO, MACULAN, 1994)

Mais formalmente, o PCV pode ser definido em relação a um grafo completo  $G=(V, A)$ , onde  $V$  é um conjunto de  $n$  vértices e  $A$  é o conjunto de arcos ou arestas que conectam cada par de cidades  $i$  e  $j \in V$ . Este conjunto  $A$  pode ser entendido como todas as possíveis ligações entre os pares de cidades  $V$ . A cada arco/aresta está associado um custo  $c_{ij}$ . O PCV deve encontrar a rota de menor custo, passando por cada vértice (cidade) uma única vez. Há duas possibilidades de problemas:

- se o custo de ir da cidade  $i$  até a cidade  $j$  é igual ao custo de se deslocar da cidade  $j$  até a cidade  $i$ , ou seja, se  $c_{ij} = c_{ji}$  para toda cidade  $i, j \in V$ , então o problema é conhecido como PCV simétrico;
- caso contrário, ou seja, se  $c_{ij} \neq c_{ji}$  para algum  $i, j \in V$ , então o PCV é assimétrico. Estes problemas podem incluir arcos unidirecionais, por exemplo, ruas de mão única.

### 2.2.2 Formulação matemática

A formulação matemática para o PCV é dada pela seguinte função objetivo, definida na figura 3, e sujeita às restrições que se seguem:

$$\begin{aligned} \text{Função objetivo :} \quad & \text{MIN} \sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot x_{ij} \\ \text{Sujeito a :} \quad & \sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n \quad (1) \\ & \sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \quad (2) \\ & \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subset V, S \neq \emptyset \quad (3) \\ & x_{ij} \in \{0,1\} \quad i, j = 1, \dots, n \quad (4) \end{aligned}$$

Figura 3 - Formulação matemática do PCV

A variável  $x_{ij} = 1$  representa que a cidade  $j$  é imediatamente posterior à cidade  $i$  na rota que está sendo apresentada. Em caso contrário,  $x_{ij} = 0$ . A variável  $n$  representa o número de cidades do problema,  $S$  é um subconjunto do conjunto  $\{1, 2, \dots, n\}$  e o símbolo “ $| \cdot |$ ” representa a cardinalidade do conjunto. Desta forma, a função objetivo é a representação matemática da minimização do somatório das distâncias entre as cidades da rota, considerando somente os vértices que pertencem à rota específica ( $x_{ij} = 1$ ). A primeira e a segunda restrições garantem que para cada cidade  $i$  e  $j$  só participam da rota uma única vez como conexão de saída e uma única vez como conexão de chegada. A restrição 3 garante que não haja subrotas, ou seja, uma rota que não inclua todas as cidades que devem ser visitadas e a restrição 4 garante, por sua vez, que  $x_{ij}$  deve ser uma variável binária, ou seja, que só admita valores 0 ou 1.

### 2.2.3 Aplicações do PCV

Além da aplicação direta do PCV em situações reais que exijam a construção de rotas, como as viagens de caixeiros viajantes ou as rotas de entrega de um único veículo, outros problemas também podem ser relatados como de interesse dos pesquisadores da área.

### *Fabricação de placas de circuitos eletrônicos*

Diversas camadas de uma placa de circuito eletrônico impresso devem ser assentadas e perfurações são realizadas utilizando máquinas de controle numérico que controlam um braço robótico automatizado. Este braço pode realizar furos de diversos diâmetros mas, para tanto, é necessário trocar a broca, o que consome tempo. Desta forma, é interessante que, para uma determinada broca, sejam realizadas todas as perfurações necessárias antes da troca para outra broca. Cada seqüência de perfurações com uma determinada broca é considerada um PCV, pois o braço parte de uma posição inicial, se desloca uma única vez até cada ponto específico, realiza a perfuração e, ao final, retorna a sua caixa de ferramentas.

### *Roteamento de veículos*

O problema de roteamento de veículos (PRV) consiste na alocação de um determinado número de veículos que deve entregar ou coletar encomendas em um determinado número de clientes. A ordem da entrega dos clientes pode ser realizada através de um PCV. Usualmente, restrições em relação à capacidade dos veículos e as janelas de tempo – horário inicial e final que um determinado cliente está esperando a visita do veículo – são necessárias para a correta modelagem do problema.

### *Seqüenciamento de tarefas*

O seqüenciamento de tarefas em uma determinada máquina, por exemplo, em uma indústria de manufatura, pode ser modelado como um PCV. Considerando que  $n$  represente o número de tarefas que devem ser processadas em uma dada seqüência que espera ser obtida pelo algoritmo, é possível estabelecer  $c_{ij}$  como o custo de preparação para processar a tarefa  $j$  imediatamente após a execução da tarefa  $i$ . Quanto menor for o custo total de preparação da máquina, que é considerado tempo perdido dentro da empresa, mais eficiente será a execução das tarefas. O custo de preparação pode ser considerar a calibragem da máquina, a troca de componentes (por exemplo, a ferramenta de um braço mecânico), limpeza e quaisquer outros procedimentos necessários para que a nova tarefa  $j$  inicie. Usualmente, este custo é calculado como o tempo necessário para realizar estas operações.

#### 2.2.4 O PCV como Tema Gerador

Dentro da concepção pedagógica, que será detalhada no capítulo três, o Tema Gerador ganha destaque em relação à escolha do PCV como problema-tipo. O Tema Gerador pode ser caracterizado como o contexto, o que emerge da sabedoria dos próprios alunos, o ponto de partida para o processo de construção da descoberta do conhecimento científico. Os temas geradores são extraídos “da prática de vida dos educandos” (TOZONI-REIS, 2006), através da análise investigativa da sua realidade, definida como a compreensão dos fatos como partes de um todo maior, conhecido como *totalidade concreta* (STRECK *et al.* 2008). A realidade, desta forma, não é simplesmente a soma de todos os fatos ou conceitos e, sim, a interligação entre os mesmos e, da mesma maneira,

estudar a realidade não é simplesmente coletar dados, mas investigar a realidade através dos mesmos. A concepção da realidade deve ser construída pelos atores do processo educativo e não repassada de um ator (professor) para outro (alunos).

Pode o PCV ser utilizado como um problema tipo através dos temas geradores que são desenvolvidos com os alunos? A resposta, positiva, para esta indagação é definida como um dos pressupostos desta tese. Metodologicamente, são propostos desafios aos alunos, durante os atos educativos, onde o vocabulário utilizado no desenvolvimento das soluções apresentadas – investigação da realidade – é utilizado para a construção dos conceitos relativos às heurísticas e metaheurísticas, dentro do contexto do PCV ou de suas variantes. Ao trabalhar com o vocabulário proveniente do tema gerador, pode-se ampliar o potencial reflexivo e intensificar a metodologia problematizadora. Convém salientar que este tema gerador deve ser utilizado para irradiar a concepção de uma pedagogia comprometida com a compreensão e transformação da realidade e não, somente, visando à resolução pontual de um problema específico que foi definido na sala de aula. Para tanto, as aplicações do PCV na vida real e o diálogo que pode ser definido a partir destas constatações devem nortear o ato educativo.

### 2.3 Soluções Clássicas para o Problema do Caixeiro Viajante

A resolução do PCV pode abranger duas opções: (a) o desenvolvimento de algoritmos exatos; (b) o desenvolvimento e teste de heurísticas e metaheurísticas. Os *algoritmos exatos*, usualmente desenvolvidos através de cálculos de programação matemática, têm como principal ponto a garantia de encontrar a solução ótima, ou seja, a solução com o menor custo possível. No entanto, normalmente o tempo necessário para a resolução completa de um problema na exatidão é proibitivo, o que não ocorre com as heurísticas e metaheurísticas, que usualmente encontram *boas* soluções em um tempo razoável. No entanto, não há como garantir a otimalidade da solução obtida.

#### 2.3.1 Algoritmos exatos

Os algoritmos exatos, como o próprio nome implica, buscam a solução *exata*, ou seja, ótima, de um problema dado. Para o PCV, um algoritmo exato deve encontrar o ciclo hamiltoniano de menor custo possível para um dado problema. Diversas implementações de algoritmos exatos têm sido propostas na literatura, sendo que os algoritmos baseados na busca em profundidade em grafos (*depth first search – DFS*) ou em largura (*breadth first search – BFS*) se apresentam como casos particulares dos algoritmos *Branch and Bound* (HOROWITZ *et al.*, 1988). Estes algoritmos têm como princípio básico explorar, mesmo que de forma implícita, todas as possibilidades de rotas, construindo uma solução completa para o problema em questão.

É fácil perceber que, ao construir *todas* as possíveis rotas de um determinado problema, a melhor rota será encontrada com facilidade. No entanto, o número de rotas que devem ser analisadas aumenta exponencialmente na seguinte razão:

$$\text{PCV simétrico: número de soluções} = \frac{(n-1)!}{2}$$

$$\text{PCV assimétrico: número de soluções} = (n-1)!$$

Para efeitos de comparação, é possível imaginar um computador arbitrário que gere e avalie 1000 rotas por segundo. Neste caso, é possível construir a tabela 2, que apresenta o número de cidades, o número de rotas possíveis (considerando o problema simétrico) e o tempo gasto no computador supra citado, dado em *s* – segundos, *m* – minutos, *h* – horas, *d* – dias, *a* – anos, *c* – séculos.

<i>N</i>	Número de rotas possíveis	Tempo gasto
3	1	0,001 s
4	3	0,003 s
5	12	0,012 s
6	30	0,030 s
7	360	0,360 s
8	2.520	2,520 s
9	20.160	20,160 s
10	181.440	181,440 s 3,024 m
11	1.814.400	1.814,400 s 30,240 m
12	19.958.400	19.958,400 s 332,640 m 5,544 h
13	239.500.800	239.500,800 s 3.991,680 m 66,528 h 2,772 d
14	3.113.510.400	3.113.510,400 s 51.891,840 m 864,864 h 36,036 d
15	43.589.145.600	43.589.145,600 s 726.485,760 m 12.108,096 h 504,504 d 1,382 a
20	60.822.550.204.416.000	60.822.550.204.416,000 s 1.013.709.170.073,600 m 16.895.152.834,560 h 703.964.701,440 d 1.928.670,414 a 19.286,704 c
25	310.224.200.866.619.719.680.000	310.224.200.866.619.719.680,000 s 5.170.403.347.776.995.328,000 m 86.173.389.129.616.588,800 h 3.590.557.880.400.691,200 d 9.837.144.877.810,112 a 98.371.448.778,101 c

Tabela 2 - A explosão combinatória do Problema do Caixeiro Viajante

É muito fácil perceber que qualquer algoritmo que resolva o problema na exatidão gerando *todas* as possibilidades de rotas possíveis só pode ser utilizado em problemas muito pequenos. No entanto, há formas de diminuir o tempo computacional gasto através dos métodos *branch and bound* citados anteriormente, utilizando a *poda* de partes do grafo que não interessam. Um exemplo de poda pode ser visualizado na figura 5, que representa um grafo de geração de rotas arbitrário para um problema com quatro cidades. Os nós representam as cidades, e os arcos descendentes, as possibilidades de partida para o nó em

questão, iniciando na cidade A. O grafo em questão é completo, ou seja, é possível partir de uma cidade  $i$  para uma cidade  $j$  para todos  $i$  e  $j \in V$  e o problema é assimétrico.

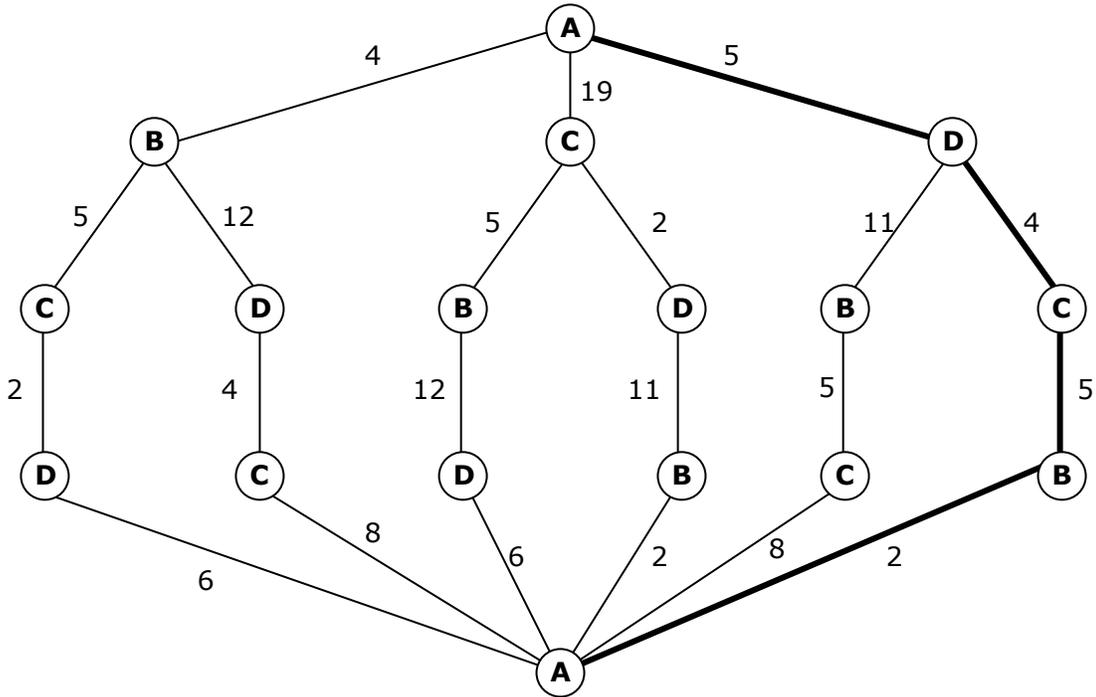


Figura 4 - Grafo de busca completo para um PCV de quatro cidades

O algoritmo de busca em profundidade (*depth first search – DFS*) é muito simples de ser entendido. A partir da cidade inicial, ele gera todas as possibilidades, partindo sempre da esquerda para a direita, e gerando a rota completa, ou seja, até retornar novamente para a cidade inicial. A aplicação do DFS no problema da figura 4 geraria as seguintes rotas, nesta ordem:

- (A,B,C,D,A) com custo de 17
- (A,B,D,C,A) com custo de 28
- (A,C,B,D,A) com custo de 42
- (A,C,D,B,A) com custo de 34
- (A,D,B,C,A) com custo de 29
- (A,D,C,B,A) com custo de 16

A rota ótima encontrada é a (A,D,C,B,A) com o custo total de 16. No entanto, este procedimento pode ser melhorado em relação a sua performance computacional através da *poda* de partes do grafo. Há várias formas para realizar a poda de um grafo (GOODRICH, 2002 e NARAHARI, 1997), sendo que a mais simples de ser implementada é através da análise do custo da melhor rota. A idéia é bastante simples: ao encontrar uma subrota cujo custo atual seja superior ao da melhor rota completa já descoberta, desconsidere as subrotas subseqüentes. É fácil perceber que, se uma rota ainda não completa, ou seja, faltando cidades para serem inseridas, já contempla um custo superior a melhor rota atual,

não é necessário prosseguir no caminhamento desta parte do grafo, pois não é possível que a rota diminua seu valor.

A aplicação desta poda no grafo da figura 4 geraria um grafo menor, apresentado na figura 5, e de menor custo computacional em termos de tempo.

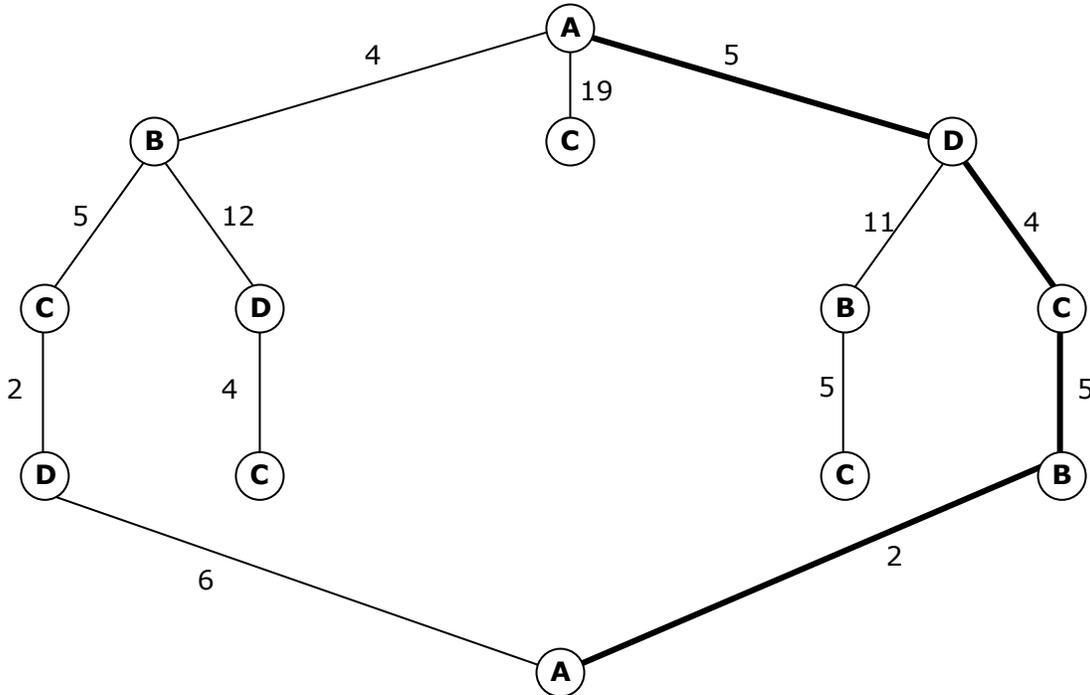


Figura 5 - Grafo de busca com poda para um PCV de quatro cidades

O algoritmo inicia sua análise gerando a primeira rota pela esquerda, ou seja, (A,B,C,D,A) com custo de 17. Esta rota é considerada a melhor rota até o momento e o custo dela é armazenado como *custo incumbente*. Uma rota *incumbente* representa o melhor *tour* encontrado até o momento. Se ele não for melhorado até o final, o mesmo se torna o *tour* ótimo do sistema. A segunda rota parte de A, vai até B, depois até D e finalmente até C. No entanto, ao chegar em C, a rota analisada já possui o custo de 18 (4+12+4). Como este custo é maior que o custo incumbente, não é gerado o último passo e o algoritmo segue para o próximo passo.

A próxima rota inicia em A e vai até C. Novamente, o custo deste pequeno trecho já é maior que o custo incumbente. Desta forma, todo o subgrafo resultante das possibilidades de rotas a partir da subrota A-C são ignorados e o algoritmo continua. A penúltima opção é gerada, formando uma rota partindo de A para D, depois para B e finalmente chegando a C, com um custo de 21. Esta subrota também é finalizada e o último *tour* possível é analisado, formando a rota completa (A, D, C, B, A), cujo custo é inferior ao incumbente. A nova rota é armazenada e o custo incumbente é atualizado. Como esta era a última folha do grafo a ser gerada, o algoritmo finaliza, apresentando a solução final, de custo 16.

Existem diversos modelos e algoritmos exatos desenvolvidos para o PCV, onde é possível citar os trabalhos de (LITTLE *et al.*, 1963), (BALAS, GUIGNARD, 1979), (HELD, KARP, 1970), (CHRISTOFIDES, 1970) e (VOLGENANT, JONKER, 1982).

Uma lista bastante completa dos algoritmos exatos e suas variantes está disponível em (LAWLER *et al.*, 1985).

No entanto, apesar do avanço tecnológico, principalmente relacionado ao poder computacional disponível, mesmo os melhores algoritmos exatos têm dificuldade em encontrar as soluções ótimas para problemas maiores em um tempo computacional razoável. Desta forma, uma alternativa deve ser buscada para que os limites computacionais e de tempo sejam satisfeitos. No ramo da pesquisa operacional, as heurísticas têm cumprido este papel, tentando fornecer boas soluções em tempos menores.

### 2.3.2 Heurísticas

O termo heurístico provém do grego *heuriskein* = descobrir, do mesmo radical que deu origem a palavra *heureka*, imortalizada pelo matemático e filósofo grego Arquimedes. Uma heurística é um procedimento algorítmico desenvolvido através de um modelo cognitivo, usualmente através de regras baseadas na experiência dos desenvolvedores. Ao contrário dos métodos exatos, que buscam encontrar uma forma algorítmica de achar uma solução ótima através da combinação ou busca de todas as soluções possíveis, as heurísticas normalmente tendem a apresentar um certo grau de *conhecimento* acerca do comportamento do problema, gerando um número muito menor de soluções.

Os métodos heurísticos englobam estratégias, procedimentos e métodos aproximativos com o objetivo de encontrar uma *boa* solução, mesmo que não seja a ótima, em um tempo computacional razoável. Algumas definições de heurísticas encontradas na literatura são citadas a seguir:

*...procedimientos simples, a menudo basados en el sentido común, que se supone ofrecerán una buena solución (aunque no necesariamente la óptima) a problemas difíciles, de un modo fácil y rápido.*

(ZANAKIS, EVANS, 1981 *apud* DIAZ, 1996)

*Para resolver eficientemente muitos problemas difíceis, geralmente é necessário comprometer as exigências de mobilidade e sistematicidade e construir uma estrutura de controle que não garanta encontrar a melhor resposta, mas que quase sempre encontre uma resposta muito boa. ... a heurística é uma técnica que melhora a eficiência de um processo de busca, possivelmente sacrificando pretensões de completeza.*

(RICH, KNIGHT, 1993)

Entre os fatores que tornam interessante a utilização de algoritmos heurísticos na resolução de um determinado problema (DIAZ *et al.*, 1996), pode-se citar a possibilidade de oferecer soluções *boas* para problemas cujos métodos exatos sejam inexistentes ou que requeram um tempo muito alto de processamento e/ou quando o tempo de resposta necessário seja definido e finito.

As heurísticas podem ser divididas, em termos pedagógicos, em *construtivas*, *melhoramento* e *metaheurísticas*. As duas primeiras serão abordadas nesta seção e as metaheurísticas serão consideradas a parte, na seção 2.3.3.

Uma heurística *construtiva*, ou míope, para o PCV, consiste em tentar encontrar uma *boa* rota, considerando a cada interação somente o próximo passo, ou seja, o critério de escolha é basicamente local (CAMPELLO, MACULAN, 1994). Ela parte de uma solução vazia e constrói a rota, inserindo sempre uma cidade de cada vez, até atingir a rota completa. Algoritmos construtivos não possuem nenhum esquema de *backtracking*, ou seja, após inserir uma cidade, não é possível retirá-la da rota.

Um algoritmo genérico para uma heurística construtiva pode ser visualizado na figura 6. Nele, a entrada, representada pela solução parcial  $Sp$ , pode representar tanto um vértice quanto um conjunto de vértices unidos em um ciclo. A expressão *melhor componente guloso de P* garante que, para os vértices candidatos do conjunto, seja escolhido aquele que tenha o melhor valor da função de avaliação. O símbolo  $\otimes$  representa um operador genérico que, para o algoritmo em questão, é capaz de adicionar um vértice  $p$  à solução parcial  $Sp$ .

*Entrada : uma solução parcial  $Sp$*   
*Saída : um ciclo hamiltoniano  $[v_1, \dots, v_n]$*   
 $C_{ij}$  = custo entre os vértices  $i$  e  $j$   
 $N = \{a, b, \dots, n\} \rightarrow$  conjunto de vértices  
 $P$  = conjunto de vértices candidatos  
 ENQUANTO  $Sp$  não for uma solução completa  
      $p \leftarrow$  melhor componente guloso de  $P$   
      $Sp \leftarrow Sp \otimes p$   
     Atualize  $P$   
 FIM – ENQUANTO  
 Saída =  $Sp$

Figura 6 - Heurística construtiva genérica para o PCV

As heurísticas de melhoramento, por sua vez, iniciam já com uma solução factível. Uma rota é considerada *factível* se ela satisfaz todas as restrições presentes na formulação matemática representada na figura 3, ou seja, se ela contém todos os vértices do problema; se há somente um arco de entrada e um de saída para cada cidade; e se a rota forma um ciclo completo, retornando novamente para o ponto de partida. Desta forma, a heurística trabalha no *melhoramento* da solução atual, através da realização de passos sucessivos.

Usualmente, estes passos realizam a exclusão e inclusão de arcos ou vértices a esta rota, de forma a pesquisar a vizinhança da solução em busca de uma rota de melhor qualidade. O termo *vizinhança* se refere a rotas que se encontram *próximas* no espaço de busca das soluções, que podem ser alcançadas através de um movimento. Por movimento em um espaço de busca, se entende a aplicação de uma regra ou função que altere a solução atual, gerando uma nova solução. Um dos movimentos mais simples para o caixeiro viajante é o *swap*, representado pela simples troca de posição entre duas cidades. Por exemplo, em um *tour* de quatro cidades, as rotas  $\{A, B, C, D, A\}$  e  $\{A, B, D, C, A\}$  são consideradas vizinhas, pois é possível obter a cidade vizinha pela simples troca das posições das cidades  $C$  e  $D$  (*swap*). No entanto, as rotas  $\{A, B, C, D, A\}$  e  $\{A, D, C, B, A\}$

exigem um maior número de trocas para que sejam consideradas vizinhas através do movimento *swap*. É importante observar que, para movimentos diferentes, a estrutura de vizinhança também se altera.

As heurísticas de melhoramento param, usualmente, quando nenhuma outra troca possível melhora o resultado atual, o que é considerado um ótimo local. Este ótimo local pode, eventualmente, ser o ótimo global, ou seja, a melhor solução possível para o problema, mas não há garantias em relação a este fato, como em todos os procedimentos heurísticos. O gráfico da figura 7 apresenta esta relação. A heurística de melhoramento inicia a partir de uma solução inicial, que é representada por um ponto no gráfico. A partir deste momento, através do critério de vizinhança adotado, ele busca soluções ao seu redor e, no caso do gráfico apresentado, reduz o custo da solução até o segundo ponto em destaque, o *ótimo local*. No entanto, neste momento a heurística cessa seu funcionamento. Como é possível observar, todas as soluções vizinhas à rota do ótimo local possuem um valor superior à própria solução. Desta forma, o algoritmo pára e exibe como resposta o ótimo local, ignorando as possibilidades que poderiam levar o algoritmo a, mais tarde, encontrar um *ótimo global*, representado graficamente como o último ponto em destaque da figura 7. Cabe salientar, novamente, que não há garantias que o primeiro ponto, o *ótimo local*, não seja realmente o ótimo global, ou seja, se a busca continuasse indefinidamente, existe sempre a possibilidade de não haver outras melhorias a serem encontradas.

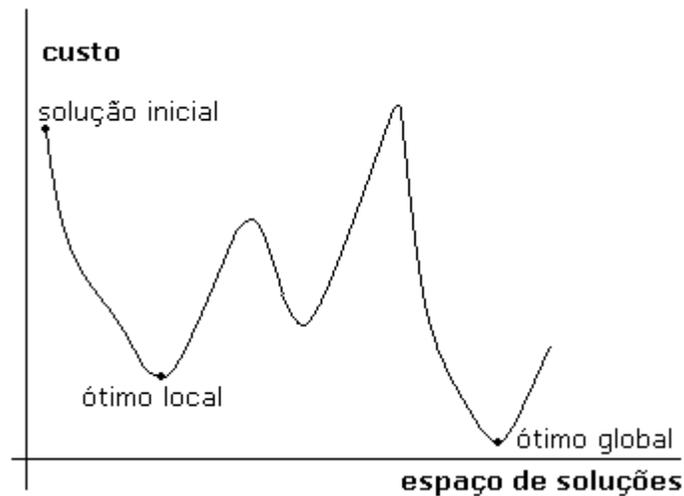


Figura 7 - Ótimo local e global

Um algoritmo genérico para as heurísticas de melhoramento pode ser visualizado na figura 8.

*Entrada* : um ciclo hamiltoniano  $[v_1, \dots, v_n]$   
*Saída* : um ciclo hamiltoniano  $[v_1, \dots, v_n]$   
 $C_{ij}$  = custo entre os vértices  $i$  e  $j$   
 $\beta$  = critério de vizinhança  
*solucao\_incumbente* = *Entrada*  
*controle* = verdadeiro  
 ENQUANTO *controle* = verdadeiro FAÇA  
     *controle* = falso  
     *SOL* = gerar\_todas\_solucoes\_vizinhas( $\beta$ , *solucao\_incumbente*)  
     ENQUANTO *SOL*  $\neq \emptyset$  FAÇA  
          $\alpha$  = *SOL*[topo]  
         SE custo( $\alpha$ )  $\leq$  custo(*solucao\_incumbente*) ENTÃO  
             *solucao\_incumbente* =  $\alpha$   
             *controle* = verdadeiro  
         FIM – SE  
     *SOL*  $\leftarrow$  *SOL* – [topo]  
     FIM \_ ENQUANTO  
 FIM – ENQUANTO

Figura 8 - Heurística de melhoramento genérica para o PCV

### 2.3.3 Metaheurísticas

As metaheurísticas são procedimentos heurísticos que guiam outras heurísticas, usualmente de busca local, experimentando o espaço de soluções além do ótimo local, buscando explorar boas características das soluções encontradas e explorar novas regiões promissoras. Existem várias formas de classificar as metaheurísticas, sendo que as mais comuns são explicitadas a seguir:

- com relação à existência de uma memória: as metaheurísticas com memória, como a *busca tabu* (GLOVER, LAGUNA, 1997), apresentam estruturas de dados que armazenam temporariamente características ou soluções inteiras de bom aproveitamento; enquanto que as demais não utilizam nenhum tipo de memória, como o *simulated annealing* (KIRKPATRICK *et al.*, 1983) e o GRASP (FEO, RESENDE, 1994);
- exploração de um elemento da vizinhança ou exploração de uma população de soluções: a utilização de algoritmos populacionais, tais como os *algoritmos genéticos* (GOLDBERG, 1989), *algoritmos meméticos* (MOSCATO, 1989) e *scatter search* (GLOVER, 1977), prevê a exploração de várias regiões do espaço de busca a cada iteração, expandindo a possibilidade de selecionar boas rotas. Diversas trajetórias são testadas e combinadas, gerando novas soluções através dos operadores de recombinação dos algoritmos apresentados. As heurísticas não populacionais exploram somente um elemento da vizinhança a

cada nova interação, como a *busca tabu* e o *simulated annealing*, gerando somente uma trajetória de soluções.

As metaheurísticas são utilizadas para tentar suprir os problemas encontrados com a mera utilização de heurísticas construtivas e de melhoramento. Além do problema citado anteriormente em relação aos ótimos locais, duas outras situações são bastante comuns na aplicação das técnicas puramente heurísticas, conforme apresentado nos gráficos da figura 9.

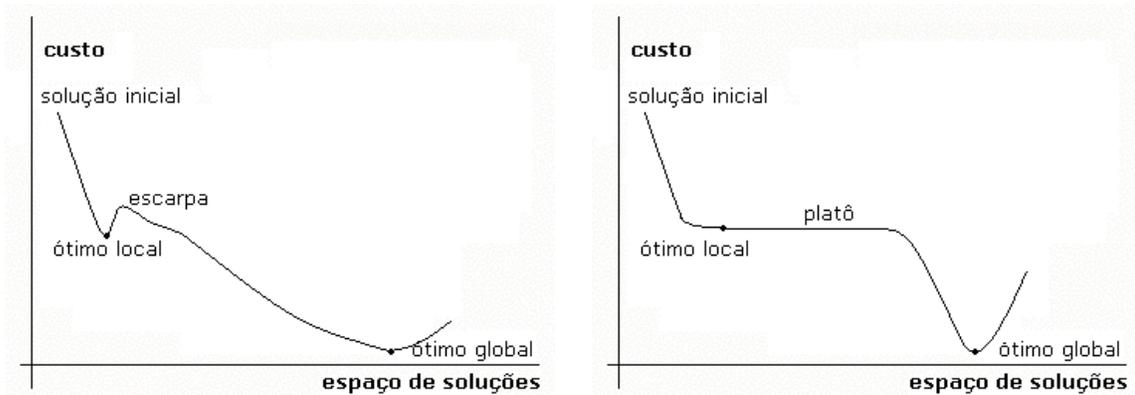


Figura 9 - Busca local: escarpa e platô

A *escarpa* é um problema na busca do espaço de soluções caracterizado pela piora nos resultados, após encontrar o ótimo local, por um curto espaço de tempo. Considerando o algoritmo da figura 8, se nenhum dos vizinhos do ótimo local for aprimorante ou, pelo menos, possuir o mesmo valor, o algoritmo pára. Como pode se observar na parte esquerda da figura 9, em determinadas situações isso conduz ao *problema da escarpa*, pois é possível que a trajetória da curva entre novamente em descendente até um outro ótimo local ou um possível ótimo global. No entanto, os algoritmos de melhoramento simples não conseguem testar possibilidades que não sejam sempre aprimorantes. O mesmo algoritmo da figura 8 pode conduzir a outro problema, definido como *platô*, que ocorre devido aos critérios de finalização dos algoritmos. Quando as soluções vizinhas à solução incumbente não são aprimorantes, mas também não pioram a solução atual, usualmente se estabelece um limite para a busca no *platô*, ou seja, nos vizinhos que possuem soluções de custo estagnado. Este limite pode ser de tempo ou de número de iterações. Em ambos os casos, se o limite for muito curto, é possível perder um novo ótimo local, como aparece na figura 9, à direita. No entanto, se o limite for muito longo, o tempo computacional cresce sem nenhuma garantia de que o mesmo encontrará alguma solução melhor.

As metaheurísticas podem guiar as heurísticas de melhoramento – busca local, fugindo dos ótimos locais, escarpas e platôs. No entanto, assim como as demais heurísticas, os critérios e parâmetros dos algoritmos devem ser explorados com cuidado com o intuito de gerar soluções factíveis e de boa qualidade em um tempo computacional razoável.

## 2.4 Heurísticas e Metaheurísticas como objeto de estudo

### 2.4.1 Ensino superior de Heurísticas e Metaheurísticas

O desenvolvimento dos conteúdos agregados ao ensino de heurísticas e metaheurísticas ocorrem nos mais diversos momentos e disciplinas dos currículos básicos da graduação e pós-graduação. Usualmente, esta disciplina recebe nomes tão diversos como *Metaheurísticas* (UFRN), *Pesquisa Operacional* (UFMG, UNIFRA), *Inteligência Computacional para Otimização* (UFOP), *Otimização Combinatória* (UFOP), *Introdução à Pesquisa Operacional* (UFSC), *Seminários de Otimização* (USP), *Otimização de Sistemas* (UNICAMP) e *Inteligência Artificial* (UFSM). Todas estas disciplinas ocorrem dentro dos cursos de graduação em Ciência da Computação, Engenharia da Computação ou Sistemas de Informação, bacharelados ligados à área do saber relacionada à computação.

Segundo a Resolução n. 4, de 13 de julho de 2005, do Conselho Nacional de Educação – Câmara de Educação Superior (CNE, 2005), os cursos de bacharelado em Administração devem prover uma ou mais disciplinas com o tema *pesquisa operacional*. A forma da aplicação desta resolução é de inteira responsabilidade das instituições de ensino. Da mesma forma, a Resolução n. 2, de 11 de março de 2002 (CNE, 2002) institui que, para todos os cursos de Engenharia, o núcleo de conteúdos profissionalizantes, que deve conter no mínimo 15% de carga horária, deverá versar sobre um subconjunto coerente de diversos tópicos e, entre estes, está a disciplina *Pesquisa Operacional*.

Os currículos dos cursos de Matemática também oferecem, em alguns casos, tópicos relacionados à otimização de sistemas e heurísticas, principalmente no que concerne à resolução de problemas através de algoritmos heurísticos (SOUZA, PEREIRA, 2005). Disciplinas como *Seminários de Resolução de Problemas* e *Epistemologia da Matemática* (USP) são apenas alguns exemplos.

Conforme o exposto, o ensino superior na área de heurísticas e metaheurísticas se encontra enraizado nas matrizes curriculares de diversos cursos de graduação e pós-graduação das áreas tecnológicas e administrativas.

### 2.4.2 Problemas e dificuldades

O ensino na área das ciências exatas e tecnológicas é um desafio recorrente. Em (TULLIO, 1995) se constatava que o método predominante no ensino das engenharias era o expositivo, apoiado no quadro-negro e uso do retroprojetor, caracterizando o professor como centro do processo de ensino-aprendizagem.

Em (LINSINGEN *et al.* 1999 *apud* FLEMMING, LUZ, 2000), a formação específica de engenheiros é destacada em relação a dois pontos que, segundo os autores, indicariam uma formação de profissionais ultrapassados pelas universidades. O primeiro ponto está centrado nas práticas bancárias do professor centralizador (FREIRE, 1987), onde o aluno *escuta aulas e armazena* conhecimentos. O aluno como depósito oco a ser preenchido tem sido uma análise constante nos mais diversos trabalhos (ASSIS, 2002), (VALENTE, ALMEIDA, 2001), (ABREU, 2006) e (GUEDES, 2004). O segundo ponto seria o desconhecimento da pesquisa como uma ferramenta de aprendizagem e renovação do próprio conhecimento.

Onde há abundância de recursos, uma forte pressão por resultados se interpôs no processo, ocasionando uma migração em massa na forma da simples transposição dos materiais impressos para os eletrônicos, sem nenhuma preocupação sobre as metodologias de ensino que continuaram a ser empregadas de forma tradicional (PETTRO, 1999). É notório que, no modelo pedagógico centralizado no professor – o que sabe *tudo* –, qualquer forma de diálogo é constrangida pela própria postura do mestre em relação a seus “discípulos”. Se, por outro lado, o diálogo torna-se elemento precursor e necessário do processo, a concepção dialógica-problematizadora pode ser utilizada. É nesta linha que a utilização do objeto de aprendizagem LOBO (capítulo 6) e o ambiente AMEM (capítulo 4). Ao propor uma metodologia baseada na apresentação de desafios centrada, muitas vezes, na utilização do objeto de aprendizagem, este adquire potencial de mediar tecnologicamente às relações entre professores e alunos, propondo uma base comum (vocabulário) onde as discussões acerca da realidade podem ser desenvolvidas. Por sua vez, o AMEM centraliza as mediações fora do contexto físico escolar, onde as atividades à distância, tais como tarefas extraclasse, são desenvolvidas, entregues e analisadas por todos os participantes.

Outros estudos apontam para direções semelhantes, como em (COLLIS, CARLEER, 1993 *apud* EICHLER, DEL PINO, 2006), onde os resultados foram obtidos de forma massificada em 15 mil alunos do Canadá, Estados Unidos da América, Irlanda, Holanda, Israel e Inglaterra. Os autores frisaram a dificuldade dos professores em utilizar a tecnologia, necessidade de uma constante remotivação de alunos e professores, além de problemas apontados pelos docentes que consideravam a avaliação através de meios eletrônicos muito difícil de documentar.

No entanto, algumas considerações devem ser feitas em relação ao tema específico deste trabalho, o ensino de heurísticas e metaheurísticas:

- uma heurística nada mais é do que uma *boa* idéia aplicada a um problema difícil e que leve a uma *boa* solução. Desta forma, usualmente não são necessários equipamentos especiais ou máquinas de grande valor para a realização de experimentos ou mesmo desenvolvimento de pesquisas científicas de ponta. O desenvolvimento dos modelos matemáticos exige muito mais criatividade e concentração do que propriamente recursos financeiros;
- como qualquer outra área ligada à engenharia e tecnologia, a falta de alunos é um problema constante. Segundo o Censo Demográfico de 2000 (IBGE, 2003), o Brasil possuía 5.585.835 graduados com 25 anos ou mais, distribuídos da seguinte forma: 639.900 graduados em Educação (11,45%), 630.285 em Artes, Humanidades e Letras (11,28%), 2.222.149 em Ciências Sociais, Administração e Direito (39,78%), 504.085 em Ciências, Matemática e Computação (9,02%), 527.662 em Engenharia, Produção e Construção (9,44%), 115.787 em Agricultura e Veterinária (2,07%), 835.179 em Saúde e Bem Estar Social (14,95%), 53.760 em serviços (0,96%) e 57.029 em áreas não especificadas (1,02%). Em termos comparativos, o Mercado Comum Europeu possuía, em 2005, 16 milhões de *estudantes* de ensino superior, sendo que cerca de 26% estudavam as áreas de ciência e engenharia (PIRES, 2005), ou seja, quase 4,1 milhões.

Outras considerações devem ser realizadas sobre o tema específico. Apesar de extremamente relevante no que concerne à aplicabilidade, o ensino de heurísticas e metaheurísticas é, pedagógica e tecnologicamente, normalmente apresentado de forma desinteressante e dependente da fórmula *giz + quadro-negro* (método bancário definido por FREIRE, 1987). O ensino de heurísticas e metaheurísticas, que usualmente apresenta um embasamento matemático forte e necessita um grande número de passos para que o processo faça sentido, torna-se desestimulante se realizado através de um processo didático tradicional.

Quando se fala no ensino de heurísticas e metaheurísticas, a atenção se volta, principalmente, para a modelagem, solução e análise de problemas decisórios, sendo que um estudo de caso completo corresponde à realização de experimentos numéricos com modelos lógico-matemáticos. Estes experimentos envolvem geralmente grande volume de cálculos repetitivos, fazendo-se necessário o uso intensivo do computador. Também se torna necessário o emprego de um conjunto de fórmulas e técnicas matemáticas que, se não forem ilustradas de forma aplicada, corre-se o risco de que o alcance destas não seja compreendido pelos alunos (DÁVALOS, 2002).

Existe um vácuo tecnológico e pedagógico que precisa ser preenchido para que o ensino das técnicas de heurísticas e metaheurísticas se torne mais acessível para o corpo discente. Estas técnicas usualmente são discutidas dentro dos cursos de engenharia, administração e informática. No entanto, a simples exposição de cada técnica não provê ao aluno a aprendizagem significativa necessária para a real compreensão do fenômeno que está sendo investigado. Para que uma técnica seja eficientemente explorada e compreendida pelo aluno, ele precisa verificar como seu funcionamento altera as soluções encontradas no decorrer do processo, assim como os parâmetros alteram as mesmas. Entretanto, as questões envolvendo a implementação das diferentes técnicas inviabilizam que todas as mesmas possam ser desenvolvidas por completo pelos alunos.

Uma forma interessante proveria que o aluno testasse as técnicas e compreendesse seu funcionamento para que, posteriormente, utilizasse este aprendizado no desenvolvimento de suas próprias técnicas e funções heurísticas. Dentre as técnicas que podem ser utilizadas para o incremento da real compreensão dos alunos, a animação pode ser considerada a mais viável e eficaz. A animação por computador pode ser definida como uma seqüência temporal de mudanças visuais em uma determinada cena. Adicionando a mudança de posição dos objetos com translações ou rotações, a animação baseada em computador pode mostrar variações de tempo no tamanho do objeto, cor, transparência ou mesmo textura. Quando utilizada na educação, a animação pode ter duas abordagens.

A primeira, denominada passiva, ocorre quando o usuário é um mero espectador da animação. Esta metodologia é comparada à experiência de assistir um filme ou uma seqüência pré-definida de eventos. Ela pode ser útil para a complementação dos aspectos vistos nas aulas presenciais, abordando somente o conhecimento pré-estabelecido pelo desenvolvedor da animação. Usualmente, sistemas que trabalham com este tipo de abordagem têm como característica principal a alta qualidade gráfica.

No entanto, a abordagem passiva possui uma limitação clara no que concerne à experimentação de novos conjuntos de dados, pois os parâmetros da animação são estabelecidos pelo desenvolvedor e não podem ser alterados pelos alunos. Com a eliminação desta restrição e a incorporação de facilidades no manuseio das diretivas da animação, é possível para o aluno realizar suas próprias experiências, direcionando o seu

conhecimento de acordo com o seu ritmo de aprendizagem. Esta abordagem é denominada ativa. Sistemas que trabalham com esta metodologia geram animações com uma qualidade gráfica mais baixa, com o intuito de não sobrecarregar a máquina. Como exemplos, é possível citar os trabalhos de (MARTINS *et al.*, 2003) e (ZACHARY, 2004). Além disso, ao possibilitar que os alunos tenham acesso ao código de desenvolvimento da animação realizada, permitindo que os mesmos possam alterar o núcleo do sistema, este passa a representar uma forma de simulação pois, agora, modelos diferentes podem ser implementados ao promover alterações mais profundas no sistema.

Do mesmo modo que a experimentação durante a operação das animações enriquece o aprendizado mais do que a mera observação passiva delas, é de se esperar que, com um sistema onde a própria implementação das animações gráficas é facilitada a ponto de poder ser realizada pelo estudante, a compreensão do funcionamento das heurísticas seja ainda mais intensa.

### 2.4.3 A Escolha do Problema do Caixeiro Viajante

A escolha do problema, ou problemas, que podem ser utilizados como suporte as aulas de heurísticas e metaheurísticas é crucial para o bom desenvolvimento da disciplina. A escolha do problema a ser resolvido pelos alunos deve ser bastante criteriosa. Se, por um lado, é importante que o mesmo seja o mais realista possível, para que os participantes se sintam naturalmente atraídos pela situação que, possivelmente lhes é familiar, o professor também deve ter em mente que o problema deve, necessariamente, ter soluções viáveis. Como critérios de escolha, foi estabelecido que o problema deveria ser de aplicação real, para incentivar e motivar os alunos na sala de aula, e com uma boa generalização em relação aos demais problemas.

Como a área de heurísticas e metaheurísticas é bastante ampla e a gama de problemas que pode ser resolvido é extremamente densa e rica, a primeira tarefa do professor é o estabelecimento de uma meta ou problema objetivo (problema-tipo), que vai ser discutido durante toda a disciplina. Mesmo que, aparentemente, a exploração de um único problema possa condensar sobremaneira o séqüito de problemas possíveis dentro da área de pesquisa operacional, a intenção de uma disciplina introdutória de heurísticas e metaheurísticas deve prover meios dos alunos repensarem suas próprias idéias acerca de conceitos básicos como *melhor escolha* ou *melhor rota* em um dado problema.

O aprendizado de pesquisa operacional não pode estar centrado no conhecimento de um ou outro algoritmo, e sim nos obstáculos e preceitos básicos que permeiam todas as situações, independente do problema que está sendo atacado. Desta forma, é possível que um aluno consiga generalizar os conhecimentos construídos para resolver os mais diferentes problemas.

Considerando as asserções apresentadas acima, a decisão do grupo de pesquisa foi de utilizar um único problema para as aulas de heurísticas e metaheurísticas, sendo que o escolhido foi o Problema do Caixeiro Viajante (PCV), apresentado na seção 2.2. O PCV é, provavelmente, o mais conhecido e estudado dos problemas de otimização. Sua fácil aplicação nos mais diversos campos do saber o levaram a um patamar diferenciado dos demais, pois inúmeros pesquisadores, das mais diferentes áreas, convergiram seus esforços para a resolução adequada do mesmo. Apesar da complexidade bem conhecida do mesmo, sua formulação, mesmo matemática, é simples e intuitiva, o que o torna um problema ideal

para estudantes de graduação que o reconhecem imediatamente como uma questão a ser resolvida, apesar de muitos não enxergarem, em um primeiro momento, a explosão combinatória de soluções possíveis e sua conseqüente influência no tempo computacional gasto para resolver os problemas na otimalidade. Além disso, o PCV pode ser generalizado ou especializado para outros problemas da literatura, como o de sequenciamento de tarefas, o roteamento de veículos, o controle de robôs autônomos e linhas de montagens de componentes eletrônicos. (CUNHA *et al.* 2002).

## 2.5 Conclusões do Capítulo

Revisitando as práticas bancárias definidas por (FREIRE, 1987), onde a escola é apenas a repetição do ato de depositar, onde alunos são os depositários e o professor é o depositante, definimos o papel do professor como mero comunicador que, através de explanações (comunicados) e depósitos, estimula o aluno a receber, memoriza e repetir, sem nenhuma reflexão. Logo, a colaboração, a comunicação e a criatividade não são estimuladas (DE BASTOS, ALBERTI, MAZZARDO, 2005).

Hoje em dia, é muito *simples* buscar informação na *Internet*, apesar de não ser *fácil* encontrar o que realmente se procura. Por outro lado, esta expansão da informação disponível desmitifica a figura do professor em relação à sua disciplina específica. É muito difícil, para um aluno de graduação, reconhecer o professor como único detentor do conhecimento quando os tópicos abordados em aula se refletem em milhares de sítios na *web*, onde artigos, resenhas, dissertações, teses e até mesmo livros se encontram *on-line*, apresentando uma gama muito mais profunda do que é possível para um professor estabelecer dentro da sala de aula.

Desta forma, é premente o professor se auto-avaliar, buscando, primeiramente em si, onde seus métodos já não condizem mais com a realidade observada. Sem um processo de reconhecimento prévio do professor com suas próprias aulas, nenhuma metodologia, por mais inovadora ou estimulante que pareça, arcará com o ônus de modificar as práticas ultrapassadas e renovar sua metodologia frente aos novos desafios. Pronto para se aventurar sob novos ares, o professor pode, então, investigar seus métodos, estudando *como* ele realmente ministra suas aulas. A metodologia da investigação-ação é apresentada no próximo capítulo com o intuito de fornecer subsídios para o processo de ação-reflexão-ação, necessário para a transformação da realidade do professor.

Contudo, reconhecemos o conhecimento científico do professor sobre a área específica, saberes estes necessários para que ele se reconheça como professor de uma disciplina, como essencial e de importância ímpar na definição de suas novas práticas. Dentro da perspectiva do ensino de ciências e tecnologia sob a ótica da investigação-ação educacional e do diálogo-problematizador, aprender o conhecimento científico é prioritário.

Na área de heurísticas e metaheurísticas, alguns pontos importantes podem ser analisados sob esta perspectiva:

- a disciplina é, essencialmente, a definição e construção de métodos matemáticos para a solução de problemas, ou seja, o conhecimento pode ser problematizado;
- o formalismo e abstração estão no cerne da construção de uma boa heurística. Estes pontos não podem ser descartados ou sub-dimensionados;

- muito além de ensinar as metodologias e algoritmos atuais, uma disciplina desta natureza deve ter como objetivo explícito o estímulo criativo dos alunos frente aos novos problemas, ou seja, realmente inculcar no aluno a necessidade de *compreender os fenômenos observados* (no caso específico, o resultado da aplicação dos algoritmos sobre os problemas), discutindo teorias e propondo soluções. Em um campo tão vasto, tanto cientificamente quanto em suas aplicações práticas, esta *necessidade* deve ser trabalhada durante todo o desenrolar do curso.

Considerando as características da área em questão, e aceitando como fato que as práticas tradicionais e bancárias não se adequam à nossa realidade atual, é necessário buscar uma metodologia de ensino através de uma concepção pedagógica dentro do ensino das ciências e tecnologias. No próximo capítulo, a *Pedagogia do Oprimido* de Paulo Freire e os métodos dialógico-problematizadores são apresentados como uma resposta a estas inquietações.

### 3 Ensino-Aprendizado Dialógico-Problematizador

O diálogo e suas múltiplas percepções sobre a sociedade já era reconhecido como elemento fundamental e (re)formador dos indivíduos desde a Grécia antiga. A *dialética* fora estabelecida como a *arte do diálogo*, da contraposição de idéias que levaria a outras idéias. Segundo (KONDER, 1987), a dialética grega, aos poucos, se transformou em uma verdadeira arte para, a partir do diálogo, demonstrar uma determinada tese através de argumentos capazes de definir e distinguir claramente todos os conceitos envolvidos na discussão. Apesar de importância ímpar na vida política e científica Grega, tanto quanto a retórica, há dúvidas sobre como a dialética surgiu nos campos helenísticos.

*Aristóteles considerava Zênon de Eléa (aprox. 490-430 a.C.) o fundador da dialética. Outros consideraram Sócrates (469-399 a. C.).*  
(KONDER, 1987)

Segundo a filosofia Grega, a dialética envolvia a técnica de perguntar, responder e refutar através de argumentações. Para Platão (1997), somente através do *diálogo* o filósofo estaria apto a buscar o verdadeiro conhecimento, partindo do que ele denominava mundo *sensível* (onde é possível captar as sensações através dos sentidos do corpo), até o mundo das *idéias*.

No entanto, é importante observar que o próprio conceito de dialética se transformou em relação ao tempo e a própria sociedade inserida no contexto de seu uso, utilizando diferentes doutrinas filosóficas e significados distintos, como é possível notar na dialética marxista e na dialética de Hegel. Contudo, a base da dialética ainda constitui, em última análise, o diálogo necessário entre os indivíduos.

Há muitas formas de se conceituar o diálogo, dependentes sempre do contexto histórico e social de quem os define. Ele pode ser definido a partir do seu pressuposto, a linguagem, canal de comunicação que possibilita a interação entre os sujeitos, gerando o entendimento, informação e conhecimento (PORTUGAL, 2002). Em (FREITAS, 1996), a autora expõe as idéias de Vigotsky sobre a linguagem como verdadeira construtora do sujeito. A linguagem em si já foi considerada mais importante que o diálogo, principalmente em relação ao seu uso na *retórica*, arte de convencer o interlocutor através da oratória. Usualmente, os adeptos da retórica clássica procuram conduzir o interlocutor a convencer-se da verdade das palavras do emissor através do seu próprio raciocínio. É importante observar, nesta asserção, que tais definições estabelecem como ponto de partida a existência de um *emissor* – sujeito que está, efetivamente, tentando convencer outrem – e um receptor, o interlocutor, que é levado a raciocinar sobre suas próprias convicções de tal forma a atingir o objetivo do emissor.

Um conceito em direção oposta ao diálogo proposto pela dialética é observado em (MARIOTTI, 2001), que explicita o significado em relação a sua percepção comunicativa, sem objetivos pré-definidos e, em última análise, sem levar a argumentações decisivas e teses defendidas.

*...diálogo (reflexão conjunta e observação cooperativa da experiência) é uma metodologia de conversação que visa melhorar a comunicação entre as pessoas e a produção de idéias novas e significados compartilhados. Ou, posto de outra forma: é uma metodologia que permite que as pessoas pensem juntas e compartilhem os dados que surgem dessa interação sem procurar analisá-los ou julgá-los de imediato.*

(MARIOTTI, 2001)

Desta forma, o diálogo surge como uma técnica de conversação que objetiva a construção de uma visão de mundo mais ampliada, sem necessariamente chegar a conclusões objetivas. A simples ampliação das dúvidas do sujeito já é alvo de reflexões mais agudas por parte deste, como já observava Sócrates (1996). Outros autores buscam a concepção do diálogo como técnica de comunicação.

*El dialogo es un juego al que podríamos calificar como ‘ganar-o-ganar’ (diferencia de lo que ocurre en la discusión, un juego del tipo ‘yo-gano-tu-pierdes’). Pero el hecho es que diálogo es algo más que una participación común en la que no estamos juzgando contra los demás sino con ellos.*

(BOHM, 1996 apud SANTOS, 2003)

Neste aspecto, o autor define um diálogo como um *jogo* onde todos ganham, a discussão pressupõe que exista sempre um perdedor e um ganhador, e a opinião é simplesmente uma suposição de um sujeito. Logo, o diálogo pode ser afirmado como algo compartilhado, que promove a criatividade individual e coletiva, sem preconceitos. No entanto, o autor ainda ressalta que as diferenças culturais podem acabar por promover diferenças de crenças e opiniões, transformando o diálogo em debates ou discussões.

Considerando a perspectiva do diálogo como preceptor e impulsionador das práticas pedagógicas, Paulo Freire (19/09/1921 – 02/05/1997) estabelece um eixo norteador, baseado nas suas experiências como alfabetizador de adultos e influenciado fortemente por uma visão abrangente das classes oprimidas.

Para Freire, o diálogo representa uma categoria central do processo pedagógico crítico e libertador.

*...através do diálogo, podemos olhar o mundo e a nossa existência em sociedade como processo, algo em construção, como realidade inacabada e em constante transformação*

(STRECK *et al.* 2008)

Os Temas Geradores, desta forma, estão imbuídos no contexto do diálogo freireano. O alicercamento do conteúdo programático nos saberes coletivos dos alunos é o objetivo que conduz à prática do diálogo propositivo, desafiador e transformador. A coerência metodológica do professor com seu discurso implica na postura dialógica. E, a partir do diálogo, é possível problematizar a realidade.

### 3.1 Paulo Freire e a Problematização da Realidade

A proposta de Paulo para a alfabetização de adultos criticava o sistema tradicional de ensino baseado nas *cartilhas didáticas* para o ensino da leitura e escrita. Usualmente, as cartilhas tinham como método central a repetição de palavras soltas ou frases criadas, completamente fora do contexto dos alunos para as quais elas seriam utilizadas. Para o educador, as cartilhas colocavam os homens e mulheres como meros observadores da realidade e não incluíam o aluno no processo pedagógico. Desta forma, o diálogo inexistia, na prática.

Em (FREIRE, 1987), é apresentado os dois grandes princípios do método que ficou popularmente conhecido como *Método Paulo Freire*, apesar do próprio educador não concordar com o nome, pois não reconhecia seu trabalho como uma metodologia, mas como uma teoria acerca do conhecimento.

O primeiro princípio do *método* diz respeito à politicidade do ato educativo. Não há educação neutra, pois toda educação é um ato político bem como qualquer prática política é pedagógica. Ao considerar a educação como um (re)construir dos significados de uma realidade, Paulo Freire prevê a ação do homem sobre esta realidade. Desta forma, ao se educar, o homem pode transformar sua realidade. A intervenção da realidade é o ponto central deste princípio, onde os alunos passam a se considerarem como atores do contexto onde vivem, atuando de diversas formas nesta realidade. Os alunos são convidados a debater sobre sua própria realidade, através das palavras que são lidas, escritas e discutidas a partir dos seus próprios saberes. O movimento de observação da realidade; a reflexão sobre seu papel social, histórico e político nesta realidade; a readmiração da importância das suas ações e; finalmente, a intervenção dentro do seu contexto, molda o *método* como uma metodologia eminentemente política.

O segundo princípio apresenta o diálogo do ato educativo como base da pedagogia. Acima de tudo, a relação pedagógica deve ser baseada na dialogicidade, cujo objetivo final se concentraria na luta pelas transformações necessárias na sociedade em que os indivíduos interagem. Para que os alunos se tornassem capazes de transformar o seu próprio mundo politicamente, é necessária uma ampliação de sua visão de mundo e este processo só poderia ser alcançado através do diálogo (FREIRE, 1987). Os alunos são desafiados a refletir sobre seu papel na sociedade enquanto aprendem a ler e escrever. Este processo político de criação de uma consciência crítica é mediada pelo diálogo entre alunos e professores, alunos e alunos, alunos e o objeto de conhecimento.

A sistematização do *método Paulo Freire*, como ficou universalmente conhecido, era dividido por ele em três grandes momentos, sendo estes concebidos para uma proposta de alfabetização de adultos. O primeiro momento constituía a etapa de *investigação temática*, uma busca do professor com os alunos sobre temas e palavras mais significativos para os próprios alunos, considerando o universo vocabular regional e a sua própria localidade. O estudo da realidade pelos professores era traduzido no *tema gerador geral*, interdisciplinar, capaz de gerar um ensino dos mais diversos campos do saber de forma integrada e não fragmentada.

O segundo momento representa a *tematização*, que constitui a seleção de temas e palavras geradoras, recortes da realidade investigada, realizando a codificação e decodificação dos temas em busca de suas significâncias sociais. Dos temas geradores, as *palavras geradoras* são extraídas, usualmente ilustradas para incentivar a capacidade de

gerar debates entre os alunos. Esta ilustração representa a *codificação*, o aspecto de uma determinada realidade.

Finalmente, o terceiro momento representa a *problematização*, a busca incessante pela construção crítica de uma consciência acerca dos temas debatidos. A problematização deve ser composta de situações desafiadoras, usualmente apresentando problemas locais que abrem perspectivas para a análise de problemas regionais e nacionais.

Em (FREIRE, 1989), o autor propõe a seguinte execução do seu método, através de cinco fases distintas:

- 1ª fase: levantamento do universo vocabular do grupo de alunos, onde ocorrem as primeiras interações de aproximação entre alunos e professores, de (re)conhecimento mútuo, além da anotação das palavras da linguagem dos membros do grupo, respeitando seu linguajar típico;
- 2ª fase: escolha das palavras selecionadas, seguindo os critérios de *riqueza fonética e dificuldades fonéticas*. Esta escolha deve estabelecer seqüências gradativas das mais simples para as mais complexas;
- 3ª fase: criação de situações existenciais/problematizadoras que sejam características do grupo. Estas situações devem ser inseridas na realidade local. A discussão tem como objetivo abrir perspectivas para a análise crítica consciente de problemas locais, regionais e nacionais;
- 4ª fase: criação das *fichas-roteiro*, que funcionam como um guia para os debates, as quais deverão servir como subsídios;
- 5ª fase: criação de fichas de palavras para a decomposição das famílias fonéticas correspondentes às palavras geradoras.

Desta forma, Freire sistematiza sua pedagogia de forma a implementar a problematização através do diálogo. Muitos autores, desde então, tem contribuído e reconstruído sob estas questões, buscando ampliar o trabalho de Freire. Entre estes, é possível citar (CARR, KEMMINS, 1986) e a *Investigação-Ação Educacional*; e (ANGOTTI, DELIZOICOV, 1990), focando na espiral reflexiva dos três momentos pedagógicos. Na próxima seção, nos deteremos na investigação-ação e na prática freireana, sob a ótica da investigação-ação educacional.

### **3.2 Investigação-ação e a Educação Dialógica Problematizadora**

A investigação-ação está centrada nos problemas reais enfrentados pelos professores, buscando compreender e sendo compreendida como uma forma dos mesmos pensarem sobre suas aflições, adotando uma postura investigativa dentro de suas salas de aula (MÜLLER, DE BASTOS, 2004). Nesta metodologia, a pesquisa e o ensino podem ocorrer concomitantemente. Ao desenvolver suas aulas, o professor é seu próprio observador e não somente dos alunos. Analisando sua prática, ela pode *agir* sobre a mesma, compreendendo os problemas decorridos, o que está diretamente relacionado ao entendimento dos mesmos, e buscando soluções.

O professor se transforma em um investigador de seus próprios métodos e agora, além do conhecimento, que já não mais lhe pertence de forma sozinha e autocrática, o próprio fazer educacional também não é mais algo intocável. A partir do seu próprio

escrutínio e das observações que podem ser coletadas por seus pares e alunos, o professor pode tomar consciência da metodologia que ele *realmente* pratica em aula, e não mais aquela que ele *imagina* estar realizando.

Em (CARR, KEMMINS, 1986), os autores apresentam a Investigação-Ação Educacional como uma seqüência de julgamentos e ações que constituem as etapas do ciclo de uma espiral, com quatro fases:

- planejamento: etapa antecessora a ação propriamente dita, onde é necessário refletir sobre a situação educativa, sua complexidade e importância, construindo uma base para as ações futuras;
- ação: guiada pelo planejamento realizado anteriormente sem, no entanto, incorrer no erro comum de utilizar o planejamento como um guia estático e imutável. A ação deve possuir um propósito criticamente informado;
- observação: documentação dos efeitos da ação, gerando uma base para a reflexão. A documentação contribui para a melhoria contínua da prática, através da análise da situação contextualizada, o que se pode traduzir em uma ação estratégica mais crítica;
- reflexão: finalmente, a reflexão tenta interpretar, discursivamente, os acontecimentos oriundos das ações, propondo modificações aos planejamentos das mesmas, face às evidências observadas, reconstruindo uma nova ação informada.

A aplicação prática destas idéias ressurge com a necessidade do diálogo, onde (ELLIOT, 1978) ressalta como de importância singular.

*...importância do diálogo como ferramenta constitutiva do processo de Investigação-Ação Educacional; a imersão do investigador na realidade com os participantes; e a necessidade de uma rede de acordos éticos entre os envolvidos, definindo novos aspectos na investigação-ação.*

*(ELLIOT, 1978 apud MÜLLER, DE BASTOS, 2004)*

Neste contexto, o diálogo e, mais especificamente, o diálogo-problematizador, pode contextualizar e sustentar a Investigação-Ação Educacional (IAE). Para ocorrer o verdadeiro diálogo, é necessário agir sobre o objeto e os alunos, usualmente de forma colaborativa, pois o professor, neste caso, não dialoga através da imposição dos seus conceitos. Não é uma simples troca de idéias, mas também não ocorre aqui o formalismo tradicional da educação bancária, onde um sujeito – o professor – deposita suas idéias em outro – os alunos. A reflexão conjunta entre os sujeitos, professor e alunos, respeitando suas respectivas vivências, o contexto deve servir para que ocorra um auxílio do professor, no que consiste do surgimento da consciência crítica sobre o objeto, abandonando a consciência ingênua dos alunos, conforme já discutido no capítulo anterior.

Se o professor deseja realizar uma investigação das suas ações, o diálogo problematizador se torna uma ferramenta de vital importância, pois é através deste que o objeto problematizado toma forma de uma solução construída. A reflexão sobre este processo é a base que sustenta a IAE.

Em (ANGOTTI, DELIZOICOV, 1990), a espiral reflexiva da investigação-ação foi desenvolvida através do aporte metodológico dos três momentos pedagógicos, implementado pelos autores no contexto do ensino de ciências naturais. O primeiro

momento é o *Problematização Inicial*, apresentado, normalmente, como uma questão e/ou situação que deve ser discutida e debatida pelos alunos. A problematização inicial deve problematizar, de forma dialógica, o tema a ser tratado durante a atividade educativa. Durante uma aula, este problema é colocado para os alunos na forma de uma questão, por exemplo, onde os mesmos podem expor seu conhecimento prévio sobre o assunto, discutindo suas implicações. Este momento é primordial para que ocorra o clima dialógico, comunicativo e participativo, onde as idéias podem ser livremente debatidas – considerando que o professor, mesmo assim, ainda é o foco dirigente do processo e não pode se abster desta responsabilidade.

No segundo momento, a *Organização do Conhecimento*, o professor deve orientar a sistematização do mesmo, para a completa compreensão do tema e do problema apresentado. Esta organização visa ampliar o diálogo do desafio inicial, introduzindo uma nova visão de conhecimento aos alunos, visão esta oriunda do mundo escolar e científico.

Finalmente, o terceiro e último momento, a *Aplicação do conhecimento*, a visão ampliada e crítica do aluno deverá ser usada por estes para analisar o problema inicial, generalizando-o para outras situações que também são explicadas pelo mesmo conhecimento. Novos desafios são colocados para a análise dos alunos, explicitando também as limitações do novo conhecimento prévio e científico.

Em (DE BASTOS e MULLER, 1999), os três momentos pedagógicos foram reorganizados em: *Desafio Inicial*, *Melhor Solução Educacional no Momento* e *Desafio Mais Amplo*. Para os autores, o *desafio inicial* deveria envolver uma atividade a realizar e não uma questão, transformando o momento em uma efetiva resolução de um problema, diferenciando, assim, da *questão* apresentada anteriormente. A *melhor solução educacional no momento* está relacionada à resolução do problema e o *desafio mais amplo* amplia o horizonte da questão específica, trabalhando sob o conhecimento geral da disciplina.

De forma prática, o modelo é ativado no início da aula com um *Desafio Inicial*, que instiga os alunos a investigarem suas próprias visões de mundo através de um desafio concreto e/ou um problema a ser resolvido. Através de múltiplas interações entre aluno/professor e aluno/aluno, deve-se consolidar o conhecimento científico através da codificação/decodificação da *Melhor Solução Educacional no Momento*, sistematizada pelo professor. Desta forma, este pode confrontar as visões de mundo dos alunos, tensionando as visões científicas e cotidianas, rompendo-as e problematizando o *desafio mais amplo*, “que busca avaliar processualmente a universalidade, validade e limitação do conhecimento científico-tecnológico abordado na aula.” (DE BASTOS e MULLER, 1999).

### 3.3 A Matriz Dialógica Problematizadora

A Matriz Dialógica Problematizadora (MDP) possibilita a criação de uma estrutura sistemática envolvendo educador, aluno, tema de estudo e contexto, favorecendo o exame e discussão da preocupação temática. Ela foi desenvolvida com origens na *Tábua Aristotélica de Invenção*, que objetivava criar uma estrutura para examinar e discutir sobre um determinado tema de forma sistemática (KEMMIS, MCTAGGART, 1988 *apud* MÜLLER, DE BASTOS, 2004). O objetivo da matriz é auxiliar os professores a definir seus problemas sobre as orientações educativas. Considerando a IAE, é primordial para o professor entender/descobrir quais são os problemas e/ou os objetos de pesquisa que ele deverá observar durante seus atos educativos. Com a utilização da matriz, o professor tem a possibilidade de sistematizar suas questões centrais, utilizando a mesma como um guia para a aplicação da metodologia investigativa na sala de aula.

A MDP orienta o componente investigativo da Investigação-Ação Educacional. Sua finalidade é guiar a programação das aulas, colaborações e aulas, estabelecendo objetivos de pesquisa para cada ato educativo, ajudando professores a definirem os seus problemas acerca das orientações metodológicas. Desta forma, não só as aulas são preparadas com antecedência, mas a própria investigação (pesquisa) é construída a partir de um plano de observações previamente estabelecido. Para cada ato educativo, são escolhidas as questões da matriz que podem, sob o ponto de vista do professor, ser respondidas, mesmo que parcialmente, com aquela ação específica. Dois pontos cruciais devem ser citados: a preparação de *objetivos de pesquisa* está no cerne de qualquer metodologia científica e a IAE, com seu pressuposto investigativo, não pode prescindir desta técnica (KOCHE, 2006); o estabelecimento de uma determinada pergunta (objetivo) para um ato educativo não deve ser considerado como estanque, ou seja, é possível que os objetivos sejam alterados dinamicamente, sempre que o professor considerar necessário.

Em (MÜLLER, DE BASTOS, 2004), a Matriz Dialógica Problematizadora é apresentada na forma de tabela, onde os quatro lugares comuns educativos são colocados em ambos os eixos, formando a matriz. Estes quatro elementos, *professores*, *alunos*, *tema* e *contexto*, são utilizados de forma a estabelecer questões investigativas entre os próprios objetos. A formulação destas questões parte do primeiro elemento, *professores*, em relação a ele mesmo, ou seja, o que é esperado dos professores com relação aos demais professores que participam das aulas? Da mesma forma, os demais elementos da tabela são preenchidos, sempre utilizando como guia o primeiro elemento e construindo a pergunta em relação a este: o que se espera dos alunos em relação aos professores, o que se espera do tema em relação aos professores, o que se espera do contexto em relação aos professores.

Ao finalizar a primeira linha, o professor inicia a construção da segunda linha, realizando as mesmas perguntas, mas mantendo agora os *alunos* em foco. Da mesma forma, as demais linhas e questões são desenvolvidas de maneira semelhante. É importante salientar que as questões construídas pelos professores não são estáticas em relação ao tempo. É possível realizar refinamentos nas questões, partindo de situações práticas, se for o caso, para preocupações temáticas e teóricas. O próprio ato de responder as questões durante e depois de uma aula dada contribui para a modificação dos questionamentos em si. Um exemplo de Matriz Dialógica Problematizadora a ser construída aparece na figura

10, que foi implementada para a disciplina *Inteligência Artificial* na Universidade Federal de Santa Maria.

	[A] Professores	[B] Alunos	[C] Tema	[D] Contexto
[1] Professores	Os professores possuem conhecimentos básicos sobre heurísticas e metaheurísticas para o Problema do Caixeiro Viajante, sobre a investigação ação-educacional que forma a base do AMEM e sobre o objeto de aprendizagem LOBO?	Os alunos efetivamente estão aproveitando os operacionalizadores pedagógicos do AMEM e o LOBO para o entendimento das heurísticas e metaheurísticas?	A abordagem dialógica-problematizadora potencializa o ensino de heurísticas e metaheurísticas?	As aulas de heurísticas e metaheurísticas utilizando a investigação ação-educacional, o AMEM e o LOBO favorecem no processo de ensino-aprendizagem?
[2] Alunos	Como os professores podem instigar os alunos, utilizando a problematização, o LOBO e o AMEM?	Os alunos efetivamente estão compreendendo o que são heurísticas e metaheurísticas através da abordagem baseada em problemas?	Quais são as dificuldades e os avanços do ensino de heurísticas e metaheurísticas utilizando o Problema do Caixeiro Viajante?	Como verificar se os alunos estão compreendendo o conteúdo de heurísticas e metaheurísticas que está sendo desenvolvido através da investigação ação-educacional, do LOBO e do AMEM?
[3] Tema	Como os professores têm construído a organização didática de suas aulas nas práticas de investigação-ação educacional de heurísticas e metaheurísticas?	Os estudantes têm assumido que os problemas vistos em aula realmente contribuem no processo de aprendizagem de heurísticas e metaheurísticas?	Será que o Problema do Caixeiro Viajante é adequado para o ensino de heurísticas e metaheurísticas?	As aulas de heurísticas e metaheurísticas contribuem para a evolução do conhecimento científico dos problemas propostos?
[4] Contexto	Os professores conseguem adotar a prática de investigação ação-educacional no ensino de heurísticas e metaheurísticas?	Os alunos têm compreendido as questões de heurísticas e metaheurísticas com a prática de investigação-ação educacional, o LOBO e o AMEM?	De que forma o Problema do Caixeiro Viajante, o LOBO e o AMEM contribuem para o ensino de heurísticas e metaheurísticas?	Quais são as contribuições do ensino de heurísticas e metaheurísticas sob a perspectiva da investigação-ação educacional, o LOBO e o ambiente AMEM?

Figura 10 - Matriz Dialógica Problematizadora

Outra consideração importante é que a Matriz Dialógica Problematizadora não precisa, e usualmente não é, analisada integralmente a cada aula. Para cada ato educativo, são escolhidos, de acordo com os pontos definidos para a aula, atividade ou colaboração, as questões que o professor espera responder durante e ao final da aula. No entanto, é possível que, durante a realização da aula, o professor perceba que outras questões são tanto ou mais importantes, da mesma maneira que conclua que uma determinada questão não foi possível de responder, mesmo que parcialmente, na aula corrente. No capítulo 4, a MDP é discutida em termos de sua implementação tecnológica como parte constituinte de um Ambiente Virtual de Ensino-Aprendizagem, o AMEM.

### 3.4 Conclusões do capítulo

A pedagogia de Paulo Freire e a metodologia dialógica-problematizadora podem ser empregadas em quaisquer áreas do conhecimento, não se restringindo a alfabetização de adultos, apesar da importância desta dentro do contexto a qual o autor inseriu e reconstruiu suas idéias. Atualmente, é notável a aplicação da teoria freireana nas mais diversas áreas do conhecimento, como é possível vislumbrar nos congressos específicos da área, como o Encontro Internacional do Fórum Paulo Freire (organizado, na sua quarta edição, pela Federação Nacional dos Professores – Portugal – [www.fenprof.pt](http://www.fenprof.pt)), o Colóquio Internacional Paulo Freire (organizado pelo Centro Paulo Freire de estudos e pesquisas – UFPE – [www.paulofreire.org.br](http://www.paulofreire.org.br)), o Fórum de Estudos e Leituras de Paulo Freire (organizado, na sua nona edição, pela Fundação Universidade do Rio Grande – [serv2.ceamecim.furg.br/paulofreire](http://serv2.ceamecim.furg.br/paulofreire)), e o Seminário Nacional Ensinando/Aprendendo Paulo Freire (2000) e Seminário Nacional Paulo Freire: Ensinando/Aprendendo a ser Cidadão (2001 - ambos organizados pela Secretária da Educação e Cultura/RS – Porto Alegre). Nestes congressos, bem como em diversas outras publicações nacionais e internacionais, se destaca a aplicação multifacetada das metodologias propostas anteriormente nos mais diversos campos do saber.

O processo de ensino-aprendizagem do objeto em questão – heurísticas e metaheurísticas, apresentando no capítulo 2 – pode ser realizado totalmente através do ensino-aprendizagem dialógico-problematizador, pois o próprio cerne das questões levantadas sob o aspecto dos algoritmos desenvolvidos e as teorias matemáticas que os embasam foram construídos através de problemas muito bem definidos. Desta forma, o casamento entre os conteúdos e a educação dialógico-problematizadora é perfeitamente factível. No entanto, não é a simples exposição de problemas que garante o efetivo aprendizado do aluno nem a eficácia desta metodologia. A correta aplicação da metodologia dialógica-problematizadora decorre, também, da condução das etapas através do dirigente do processo, o professor. A discussão das teorias e idéias dos alunos só tem significado científico dentro de um processo curricular através da *sistematização* daquilo que foi construído coletivamente, ou seja, pelo desenvolvimento da *melhor solução educacional no momento*. A MSEM é construída através das interações *aluno-aluno*, *aluno-professor* e interatividade *aluno-objeto* e *aluno-ambiente*. O objeto de aprendizagem LOBO, como parte do ambiente e o AMEM, como ambiente de *locus* desterritoralizado, mas permanente, da disciplina, fornecem mediação tecnológica que permite o desenvolvimento da prática estabelecida através de uma organização cícli-espiralada bem definida e coerente com as teorias-guia explicitadas. Os operacionalizadores pedagógicos

desenvolvidos no AMEM (capítulo 4) são utilizados para a caracterização dos atos educativos, que incluem a utilização do LOBO. Por outro lado, a mediação tecnológica que inicia no LOBO, através da observação dos dados experimentais, continua na MSEM através dos fóruns de discussão, onde a análise é realmente desenvolvida.

Outro fator importante a observar está no fato de que alguns alunos não estão prontos para esta mudança, por medo ou acomodação. Afinal, sempre foi mais *fácil* estudar alguns dias antes para uma dada prova formal do que desenvolver um trabalho mais complexo durante um semestre inteiro, participando diligentemente do processo. Nestes casos, cabe ao professor dirigir o processo, explicando as vantagens da construção do conhecimento pelo fazer e pensar do que simplesmente pelo decorar. Não faltam exemplos na sociedade moderna, dita sociedade da informação, que corroboram a tese de que *aprender a aprender* é muito mais importante do que saber alguma técnica ou metodologia de cor. Por outro lado, as discussões em sala de aula podem se tornar abrangentes demais e, da mesma forma, cabe ao professor, dirigente do processo – apesar de não ser mais o único detentor do conhecimento – trazer novamente os alunos para o foco da disciplina. Desta forma, há espaço para discussão e debates de idéias novas e, potencialmente, profícuas, mas também há a necessidade de se consolidar um determinado conteúdo que faz parte de uma disciplina de graduação.

Um aspecto crucial nesta metodologia, é que o professor, ou grupo de professores, continua sendo essencial para o desenvolvimento das aulas. Apesar de não ser o centralizador do conhecimento, pois ele já o partilha desde o início, deixando que os próprios alunos possam construí-los e desenvolvê-los a partir de suas idéias individuais e coletivas, o professor deve dirigir as aulas, conduzindo os alunos ao conteúdo que está sendo desenvolvido.

Por último, é também interessante apresentar um olhar um pouco diverso sobre a obra de Paulo Freire, no que concerte à sua analogia aos processos neurológicos. Em (DAMÁSIO, 1996) e (DAMÁSIO, 2000), o autor explicita uma teoria diversa da que é usualmente aceita na comunidade neurocientífica, que busca a explicação sobre o funcionamento do corpo, do cérebro e da mente através da dissociação e classificação das suas partes, pois, em teoria, é mais fácil estudar e compreender a mente como uma entidade separada do corpo, assim como cada órgão do ser humano tem sua função única e inequívoca. A dicotomia entre mente e corpo, apresentada por Descartes (*Penso, logo existo*), estaria, para o autor, errada. Ele considera que o corpo e sua representação no cérebro constituem-se em um quadro de referência indispensável para os processos neurais que, eventualmente, nós consideramos sendo a mente. Ou seja, a mente pode ser considerada um "processo", muito mais do que uma realidade material que pode aferida. Apresentando esta idéia de forma a romper com a dicotomia entre o cérebro e o corpo, o autor reconhece que suas idéias não são unânimes entre os pesquisadores da área, mas também explicita que nenhuma idéia o é, pois a neurociência ainda está engatinhando na busca por respostas.

Desta forma, Damásio apresenta a metáfora de *sair à luz*, o ato dos artistas se preparando para entrar no palco, como a tomada de consciência biológica, pela primeira vez. Quando o ser humano toma consciência sobre sua própria inteligência, sobre seu corpo, sobre quem ele é, ele sai à luz. É interessante notar que, em um nível metodológico e pedagógico, esta metáfora também pode ser utilizada para a pedagogia libertadora de Paulo Freire. Considerando o ponto principal da teoria freireana, temos como princípio à

politicidade do ato educativo. Não há educação neutra, pois toda educação é um ato político bem como qualquer prática política é pedagógica. Desta forma, ao se educar, o homem pode transformar sua realidade. A intervenção da realidade é o ponto central deste princípio, onde os alunos passam a se considerarem como atores do contexto onde vivem, atuando de diversas formas nesta realidade. Desta forma, ele parte de um conhecimento ingênuo para uma consciência crítica e, desta forma, o indivíduo pode *sair à luz*, tornando-se ativo na sociedade, crítico da mesma, e transformador de sua realidade.

Em relação à prática da investigação-ação educacional e da metodologia dialógica problematizadora através de um ambiente tecnológico (de suporte presencial, semi-presencial ou completamente à distância), é interessante que o mesmo apresente ferramentas que suportem a visão metodológica do docente, bem como facilitem seu trabalho de organização do espaço docente e das atividades cooperativas que deverão ser realizadas. Desta forma, é necessário compreender como os ambientes informatizados de ensino e aprendizagem através de meios telemáticos estão posicionados em relação a suas teorias pedagógicas (se possuírem) e, principalmente, o suporte que as mesmas fornecem em relação à aplicação das metodologias problematizadoras desenvolvidas neste capítulo.

## 4 Ambientes Virtuais de Ensino Aprendizagem sob a ótica de sua Concepção Pedagógica

### 4.1 O que é um AVEA?

Para compreender o que é de um Ambiente Virtual de Ensino Aprendizagem – AVEA, é interessante investigar sua etimologia.

Inicialmente, todos os AVEA se referem a si próprios como, antes de mais nada, um *ambiente*, o que leva a concepção mental de um espaço, um lugar, um recinto ou, em última análise, um meio. Partindo deste pressuposto, é viável concluir que os participantes destes *ambientes* interagirão entre si de tal forma que todos se encontrem no mesmo *espaço* ou *lugar* sem, necessariamente, estarem no mesmo tempo cronológico. Esta concepção é de fácil assimilação, pois diariamente nos confrontamos com os mais diversos *ambientes* onde nos defrontamos com diferentes pessoas em tempos diferentes. Mesmo onde haja um compromisso formal de socialização no ambiente, como em salas de aula presenciais ou o compartilhar de escritórios, não há garantias que todos os envolvidos estarão, invariavelmente, nos mesmos horários percorrendo os mesmos *meios*. Desta forma, a transposição de um ambiente real para o virtual, se bem sucedida, levará como arcabouço epistemológico a complexa gama de iterações humanas que acompanham o dia-a-dia de qualquer *locus* onde se fazem presentes duas ou mais pessoas. Mas o que é mesmo *virtual*?

Se os AVEA se definem como ambiente, também o fazem, e com grande alarde, como virtuais. A definição de *virtualidade* tem se alterado em relação a sua compreensão, apesar de suas bases terem sido lançadas há um bom tempo. No início da década de 60, os investigadores da Inteligência Artificial já se preocupavam acerca da definição das palavras *inteligência*, *natural*, *artificial*, *sintético* e *virtual*.

*A ciência natural é o cerne do conhecimento sob os objetos e fenômenos do mundo que nos cerca, sob suas características e propriedades inerentes, sob seus comportamentos e suas iterações intrínsecas. A tarefa principal das ciências naturais é transformar os prodígios da natureza em lugar-comum, mostrando-nos que a complexidade é uma máscara para a simplicidade, descobrindo os padrões escondidos no meio do caos aparente.*

(SIMON, 1969).

O mundo que nos cerca atualmente é formado, basicamente, por elementos construídos por seres humanos (artificiais). Praticamente todos os componentes de nosso ambiente se mostram, ou seja, possuem características, construídas por seres humanos. A temperatura onde gastamos a maior parte do nosso tempo é mantida artificialmente, assim como a umidade e até mesmo as impurezas que respiramos são controladas. Mesmo sendo controladas artificialmente, estes fenômenos trabalham com algum tipo de matéria-prima natural.

O termo natural é definido como algo que provêm da natureza (FEHÉR, 2007), ou seja, em que não há trabalho ou intervenção do homem. É preciso ter muito cuidado para não confundir um fenômeno *natural* com um fenômeno *biológico*. Uma floresta pode ser entendida como um fenômeno natural, ao passo que uma fazenda certamente é um

fenômeno biológico, mas nunca natural. As várias espécies das quais o homem é dependente, tais como o gado e o trigo, em última instância, são artefatos de sua engenhosidade. Um campo arado, não faz mais nem menos parte da natureza do que uma estrada asfaltada. O ser humano vive em um mundo cercado de objetos naturais e artificiais e seu ambiente é definido pelas iterações entre esses elementos e seus próprios propósitos.

Uma definição do termo *artificial* pode ser descrita como algo que foi produzido pelo ser humano (FEHÉR, 2007) Apesar disso, os elementos que nós habitualmente denominamos artificiais não podem ser consideradas em separado da natureza. Eles não possuem nenhum tipo de isenção que os faça ignorar, ou mesmo violar, as leis da natureza, as leis físicas que regem o nosso mundo. Ao mesmo tempo, estes artefatos são adaptados aos objetivos e desejos humanos. Eles são o que são, desde que possam conduzir objetos ou seres humanos mais rapidamente (aeroplano) ou serem apreciados pelo paladar (criação de gado). Se os objetivos dos seres humanos mudarem, seus artefatos também mudam – e vice-versa.

Por outro lado, como exemplificou (SIMON, 1969), uma gema produzida por um vidro colorido imitando uma safira, é um objeto artificial. Mas uma gema que fosse construída pelo ser humano, mas cuja composição química fosse indistinguível de uma gema natural, é um objeto *sintético*. Tais artefatos são imitações da natureza, em todos os seus detalhes, e esta imitação pode ou não utilizar os mesmos componentes básicos do objeto natural a ser imitado. O termo sintético é comumente usado como sinônimo para projetado ou composto.

A engenharia trata do sintético, da síntese, enquanto que a ciência trabalha com a análise. O engenheiro preocupa-se como os objetos devem ser, como deve ser sua funcionalidade, ou seja, quais são seus objetivos, enquanto que a ciência concentra-se no que os objetos são.

Em relação direta à virtualidade, a Inteligência Artificial investigava a definição e criação de *autômatos* reais ou *virtuais*. Em poucos anos, dois grandes grupos de pesquisa se formaram nos mais diversos laboratórios ao redor do mundo. O primeiro se concentrava na automação, ou seja, no desenvolvimento de mecanismos *reais*, cibernéticos, que imitassem os mais diversos tipos de comportamentos mecânicos encontrados na natureza. Os pesquisadores destes sistemas, denominados *embodied systems*, acreditavam que a *verdadeira* inteligência não poderia existir desprovida de um corpo físico. Os avanços da computação robótica (TORRES, 2003) são exemplos notórios das atividades de pesquisa nesta área. No entanto, um segundo grupo de pesquisadores investiu seus esforços na descoberta de metodologias e algoritmos que emulassem, de alguma forma, um comportamento dito *inteligente*. Estes comportamentos, na grande maioria dos casos, ocorria em ambientes puramente *virtuais*, ou seja, onde o *corpus* não mais existia. Das linhas de pesquisa nesta área (sistemas multiagentes, redes neurais e sistemas especialistas), os *chatterbots* têm tido uma grande expansão nos últimos anos e pode ser considerado como um ícone da *virtualização* da inteligência artificial.

Um *chatterbot* é um programa de computador que aceita como entrada comandos verbais (sonoros ou textuais) e fornece uma resposta verbalizada e contextualizada. Usualmente, a entrada e a saída dos comandos em um *chatterbot* utilizam a linguagem natural, ou seja, não há necessidade de traduzir o que se quer para uma *linguagem* própria do computador. Atualmente, os *chatterbots* mais bem sucedidos são aqueles que *não* necessitam simular fielmente uma conversa inteligente e cooperativa (McTEAR, 2004).

Analisando a virtualidade além dos parâmetros da inteligência artificial, encontramos a definição de Pierre Lévy:

*...desterritorialização de uma entidade, que, embora real, não pode ser fixada em nenhuma coordenada espaço-tempora; porém, ao ser atualizada, é capaz de gerar manifestações concretas em diferentes momentos e locais determinados. A digitalização da informação em códigos de computador e sua posterior exibição em tela e papel exemplificam o conceito.*

(LÉVY, 1999)

Desta forma, a virtualidade é expandida em seu contexto, utilização e mesmo enunciação para algo capaz de vivenciar aos seus utilizadores experiências similares às *reais*, incorporando elementos destas. O ápice da virtualidade seria, então, o ambiente sintético. E os ambientes virtuais, definidos como *locus* sem território, devem proporcionar a exteriorização dos sentimentos similares aos ambientes reais apesar de sua natureza ilimitada, como conclui Lucia Leão.

*Essa virtualidade ... enquanto potencialidade manifesta do existir, não tem limite em sua extensão em qualquer direção que queiramos nos movimentar, ou seja, não tem um limite de profundidade assim como não tem uma forma (limites) definida.*

(LEÃO, 2003)

E o processo de ensino-aprendizagem? Como se insere no contexto de um ambiente virtual? A combinação destes elementos se dá pelos novos paradigmas que sustentam o processo do ensinar a aprender. Desta forma, aprender não é mais adquirir uma informação e sim raciocinar sobre esta, atuar sobre seu mundo a partir desta e correlacionar com outras informações. Para Zabala, este é “um processo articulado às características singulares de cada aluno, traduzindo experiências, motivações e interesses pessoais explicitados num dado contexto”(ZABALA, 1998). E, como coloca Merieau, este “processo... somente ocorre pela interação entre informações disponíveis e um processo singular de apropriar-se das informações, configurando-se sentidos, significados”(MERIEU,1998). Para (FREIRE, 1987, p.31), “educador e educandos... se encontram numa tarefa em que ambos são sujeitos no ato, não só de desvelá-la e, assim, criticamente conhecê-la, mas também no de recriar este conhecimento”. A recriação do conhecimento faz parte do processo de conhecer o seu próprio mundo para que, a partir deste reconhecimento, se possa realizar a transformação do mesmo.

Desta correlação, é possível definir um AVEA como um *locus* sem território, atuando como um objeto capaz de conduzir as interações entre os diversos sujeitos que compõem os processos de ensino-aprendizagem presentes naquele ambiente específico. Muito mais que um mero artefato tecnológico, um AVEA passa a ser um instrumento pedagógico. Em uma última análise, o próprio ambiente pode ser considerado um objeto de aprendizagem; mais complexo e muito mais interativo que a maioria dos objetos desenvolvidos mas, em suma, um objeto que é utilizado como meio de comunicação e de suporte as atividades desenvolvidas pelos alunos e professores. Estas atividades podem ser traduzidas desde o

simples apoio aos encontros presenciais até como, potencialmente, único meio de comunicação e interação entre comunidades puramente virtuais.

Diversas soluções têm sido propostas para a construção de AVEA, tanto por instituições públicas como privadas, agregando tecnologias a um vasto mercado de *e-learning* que se expande de forma vertiginosa (em termo de valores, estima-se que o mercado de aprendizagem baseada por computador movimentou cerca de US\$ 6,6 bilhões em 2002 e as previsões é que para que em quatro anos o montante cresça para US\$ 23,7 bilhões – HADDAD, GRACIANO, 2004). Os AVEA incorporam ferramentas que permitem a construção de conteúdos (colaborativos ou não), canais diversos de comunicação, controle total da informação circulante pelo ambiente, a disponibilização de material instrucional e a possibilidade da incorporação de instrumentos interativos e cooperativos que podem ser construídos especificamente para uma determinada área de saber. Tais instrumentos, comumente conhecidos como objetos de aprendizagem, serão explorados em detalhes no capítulo 5.

## 4.2 AMEM 2.0 – Um AVEA Dialógico-Problematizador

Analisando os principais AVEA disponíveis na literatura até o presente momento – TelEduc (ROMANI, ROCHA, SILVA, 2000), Moodle (NASCIMENTO, LEIFHEIT, 2005), AulaNet (FUKS *et al.* 2006), Rooda (BEHAR, KIST, BITTENCOURT, 2001a e 2001b), Claroline (NUÑO *et al.*, 2005), WebCT (LU, LIY, 2003), Ensinet (BASTOS *et al.*, 2003), AVA (GRIGNS *et al.*, 2000), dotLRN (SOLARTE, URBANO, 2005) e webAula (MORONI, 2004) – é possível considerar que a pedagogia dialógica-problematizadora, como apresentada no capítulo 3, não é suportada em sua plenitude nas ferramentas tecnológicas disponíveis. Obviamente, com um pouco de esforço intelectual de imaginação, é possível adaptar uma ou outra ferramenta existente nos sistemas apresentados para a operacionalização de uma prática educacional dialógica-problematizadora. No entanto, tais soluções *forçadas* exigem do professor esforços sobressalentes. Em (VRASIDAS, 2004), o autor argumenta que o maior problema dos ambientes está na mera replicação dos operacionalizadores pedagógicos tradicionais, que ocorrem em muitos deles. Para o autor, há uma necessidade de reavaliar e readequar a forma de atuação do professor para trabalhar com um AVEA, bem como o próprio ambiente precisa ser construído de forma a suportar tais necessidades. Diversos ambientes têm sido desenvolvidos de tal forma a manter em suas bases o maior conjunto possível de ferramentas – operacionalizadores pedagógicos – sendo que a decisão de qual subconjunto utilizar se torna exclusivamente do professor.

Com base na constatação acima e considerando que o pensar sobre o seu próprio atuar é pressuposto básico para a melhoria na qualidade de ensino das instituições, o AMEM (Ambiente Multimídia para Educação Mediada por Computador), foi concebido em sua primeira versão. Seu desenvolvimento foi baseado solidamente em uma concepção metodológica clara e objetiva, onde o professor é levado a adotar uma prática pedagógica que retrate a teoria investigação-ação fundamentadora da definição do sistema. Desta forma, os docentes imbuídos da necessidade de estudar suas próprias práticas através deste marco teórico reconhecem o ambiente como uma ferramenta única e inovadora para suas próprias ações. É importante salientar que, ao contrário de outros ambientes educacionais tradicionais, o AMEM não é apenas uma coleção de ferramentas disponíveis. Através dos

operacionalizadores pedagógicos que podem ser escolhidos (programação, atividade extraclasse e atividade de colaboração), o professor pode organizar sua prática docente através da metodologia pedagógica proposta.

A primeira versão do ambiente foi desenvolvida em 2003 (FERNÁNDEZ, 2003 e LAUERMAN, 2002). Sua concepção deixava claro que o professor, ao desenvolver suas aulas, deveria utilizar como preceito pedagógico a educação dialógica-problematizadora. No entanto, era possível que professores com outras filosofias pedagógicas utilizassem o ambiente, se assim o desejassem. Todo o ambiente fora construído sob esta ótica e, apesar de ser possível a utilização de outras metodologias por parte do educador, ficava claro para este que o principal objetivo do sistema era prover um ferramental que suportasse, além da aprendizagem do aluno, a investigação de suas próprias práticas docentes, ou seja, do ensino em si.

Tal abordagem trazia como principal ponto positivo prover a um professor acostumando à metodologia dialógica-problematizadora, ferramentas para que ele desenvolvesse suas atividades no ambiente; no entanto, seu ponto negativo era exatamente o oposto da asserção acima – professores cujas metodologias divergiam da proposta pelo ambiente consideravam o sistema de difícil compreensão e uso complexo. Através da experiência acumulada durante a utilização dos professores por diversos anos (CORDENONSI *et al.*, 2005), uma nova versão do ambiente foi proposta.

#### 4.2.1 Modelagem da Interface Usuário/Ambiente

Um dos pontos chave da modelagem da nova versão do ambiente AMEM foi a readequação da interface, principalmente no que concerne a sua navegabilidade (HOELZEL, 2004). Após anos utilizando a primeira versão, se tornou bastante premente que uma grande gama de usuários achava a navegação no ambiente confusa e pouco intuitiva, principalmente no grupo dos professores. Uma análise preliminar pode ser realizada ao observar a tela inicial do ambiente em suas duas versões, conforme as figuras 11 e 12.

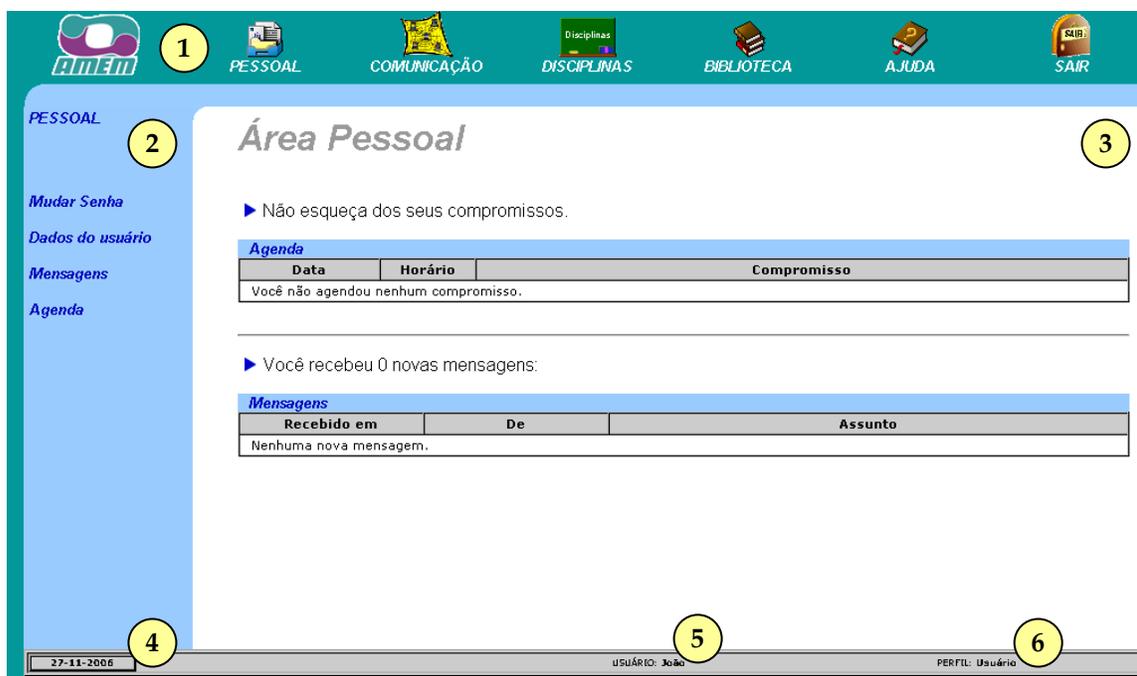


Figura 11 - Entrada de um aluno no AMEM 1.0

Ao entrar no AMEM em sua primeira versão, o aluno tem seis áreas visuais onde se concentra a interface do sistema. A primeira área (número 1 da figura 11) apresenta o menu principal do ambiente. A partir desta, o aluno pode escolher visualizar suas informações pessoais (mudar senha, dados do usuário, mensagens e agenda), acessar as ferramentas de comunicação (mural, notícias, salas de discussão, fóruns e mensagens), entrar em uma disciplina, acessar a biblioteca, obter ajuda sobre o sistema ou sair (de uma turma ou de todo o ambiente). No lado esquerdo do ambiente, se encontra o segundo menu, que se altera conforme a escolha realizada no menu principal. Esta área (número 2) tem a cor alterada conforme a opção do aluno. A área número 3 representa a interface principal do sistema. É nesta que o aluno realmente vai operar no ambiente. De acordo com as opções selecionadas no menu principal e no segundo menu, a ferramenta escolhida é apresentada ao usuário que pode, então, realizar as ações desejadas. Na parte inferior da interface, é apresentada uma barra de estado com três informações. A primeira (número 4) representa a data do sistema. A informação central, número 5, apresenta o nome do usuário. Finalmente, a área número 6 apresenta o perfil do utilizador, que pode assumir quatro papéis: usuário, aluno, professor ou administrador (um aluno é considerado um *usuário* se o mesmo ainda não *entrou* explicitamente em uma disciplina. Ao realizar esta ação, seu perfil passa automaticamente para aluno. Um administrador permanece neste perfil até entrar em uma disciplina, o que acarretará sua mudança para *professor*. É possível também um administrador acessar uma disciplina como aluno, onde seu perfil assumirá este papel.

A segunda versão do AMEM também contempla seis áreas visuais ao ingressar no sistema. A interface principal do ambiente pode ser visualizada na figura 12.

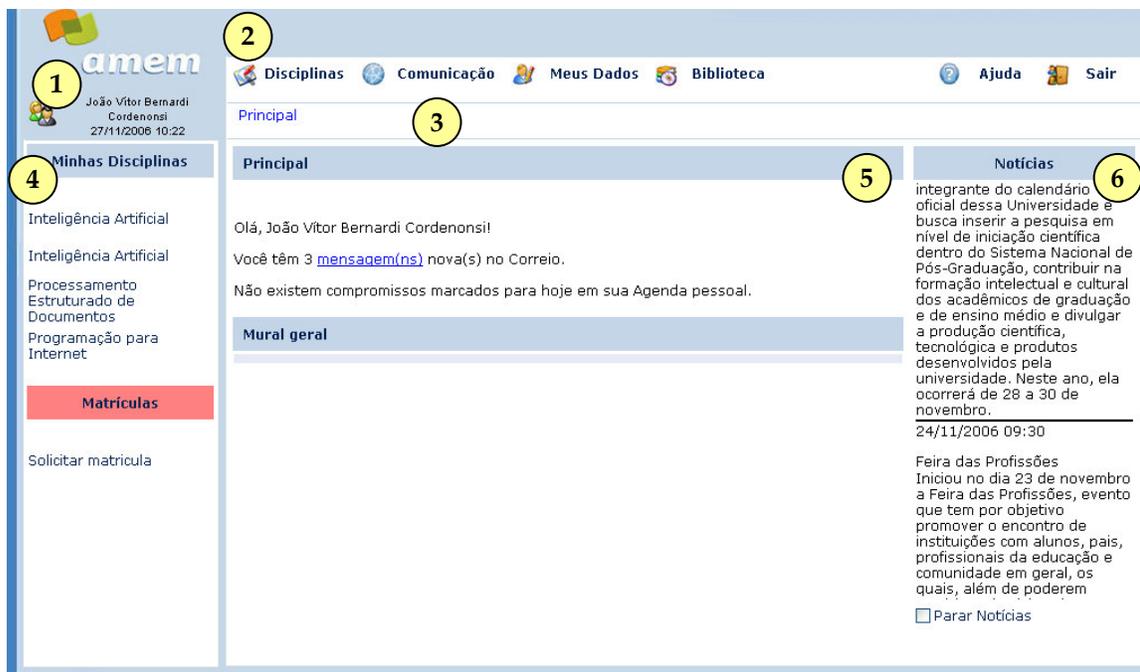


Figura 12 - Entrada de um aluno no AMEM 2.0

O aluno, ao ingressar no ambiente, tem sua atenção focada para o canto superior esquerdo, ação esta comprovada por vários estudos (PIRES, ALVES, 2002; OSF, 1990; PARIZOTTO, 1997; RIBEIRO, CHIARAMONTE, 2003). Desta forma, se optou por levar todas as informações mais importantes do ambiente para esta região. A área número 1 da figura 12 apresenta as três principais informações disponíveis para o utilizador: seu nome, a data e hora do sistema e seu perfil, representado graficamente por um ícone que, ao ser selecionado pelo clicar de um apontador gráfico, apresenta uma mensagem na tela informando o perfil do usuário. A segunda área (número 2) revela o menu principal do sistema. A partir deste, o aluno poderá acessar todas as funcionalidades do mesmo: disciplinas (onde são apresentadas todas as disciplinas que o aluno está cursando atualmente), comunicação (ferramentas de correio eletrônico, mural, sala de discussão, notícias e fóruns), meus dados (gerenciamento da senha pessoal, alteração dos dados do usuário, agenda pessoal e gerenciamento de uma pasta virtual, onde arquivos e endereços eletrônicos podem ser cadastrados e compartilhados), biblioteca (busca de bibliografias cadastradas no ambiente, virtuais ou não), ajuda do sistema e a opção de sair do ambiente. A área três representa o que comumente é conhecido como *migalhas de pão*. Este menu dinâmico apresenta ao usuário sua real posição dentro do ambiente, informando rapidamente onde o aluno se encontra e fornecendo uma forma eficaz do usuário retornar ao(s) estado(s) anterior(es). A área número 4 representa o segundo menu do sistema, que é alterado de acordo com a escolha no menu principal. Ao selecionar uma opção no menu principal ou no segundo menu, a interface se altera para representar a seleção escolhida pelo aluno. A área principal da interface é representada pela união das subáreas 5 e 6. A subárea 6 apresenta as notícias cadastradas no ambiente. Ao entrar no sistema, o aluno visualiza as notícias do ambiente e, ao selecionar uma disciplina, as notícias particulares da turma em questão. Para as demais opções, há uma fusão das duas áreas, formando a interface principal, responsável pela verdadeira interface entre o aluno e o ambiente.

Após analisar como os usuários realizavam suas tarefas no ambiente e as sugestões apontadas por estes, alguns aspectos foram levantados, servindo como base para a nova implementação.

### *Entrada/saída explícita de uma determinada turma*

Na primeira versão do ambiente AMEM, ao escolher uma determinada turma no menu *Disciplinas*, todo o sistema se alterava e passava a conter as ferramentas/informações exclusivas daquela turma. Por exemplo, no menu *Comunicação*, seria disponibilizado para o usuário as notícias, murais, fóruns e salas de discussão da turma selecionada. Caso o usuário, independente de qual seja, buscasse uma informação de outra turma, era necessário *sair* explicitamente daquela turma através do menu *Sair* e escolher uma nova turma no menu *Disciplinas*. Esta troca constante de perfil acarretava uma sobrecarga cognitiva que se tornava onerosa para os usuários. Dependendo da necessidade, caso o utilizador não tivesse certeza de qual turma ele necessitava a informação, era necessário *entrar e sair* de cada uma das disciplinas que o mesmo se encontrava matriculado (ou era responsável, no caso do *professor*) até encontrar o que buscava. Além disso, foi relatado por diversos docentes que, apesar da informação de seu perfil estar presente na interface durante todo o tempo, muitas vezes eles se encontravam perdidos no ambiente, sem ter certeza se o mesmo estava dentro de uma disciplina ou de uma turma. Tal fato ocorria, pois a barra de informações se localizava no canto inferior do ambiente, o que tornava a visualização precária.

A sobrecarga cognitiva pode ser representada pela seqüência de passos representados nas figuras 13 a 16. Nestas, é possível visualizar como um aluno deve navegar pelo ambiente AMEM 1.0 para visualizar o mural de diversas disciplinas.



Figura 13 - Aluno entra na *Comunicação* com o perfil de *usuário* no AMEM 1.0

Se o aluno, ao entrar no ambiente, se dirigir diretamente para o menu *Comunicação*, seu perfil se manterá como *usuário*, como é visualizado na parte circulada da figura 13. Neste caso, ele apenas tem acesso aos fóruns, salas de discussão, notícias e murais do *ambiente*, ou seja, aqueles cadastrados pelos professores como de cunho geral. Para que o aluno tenha acesso as notícias, murais, fóruns e salas de discussão de sua disciplina, ele deve se dirigir ao menu *Disciplinas* e escolher a disciplina específica, conforme a figura 14.

Após selecionar uma disciplina (no exemplo da figura 14, o aluno selecionará a opção circulada – Processamento da Informação Digital), a interface sofrerá uma mudança. O segundo menu apresentará as opções *Atividades da disciplina* (lista de todas as aulas, atividades extra-classe e de colaboração agendadas para a disciplina específica) e *Meu grupo de trabalho* (lista com todos os alunos e professores da disciplina). A interface principal (figura 15, letra a) apresentará automaticamente as atividades da disciplina; e a barra de estado no canto inferior da tela apresentará uma mudança do perfil para *aluno* (letra b) e duas novas informações serão visualizadas: a disciplina em que o aluno está inserido (letra c) e a turma em que ele está matriculado (letra d).

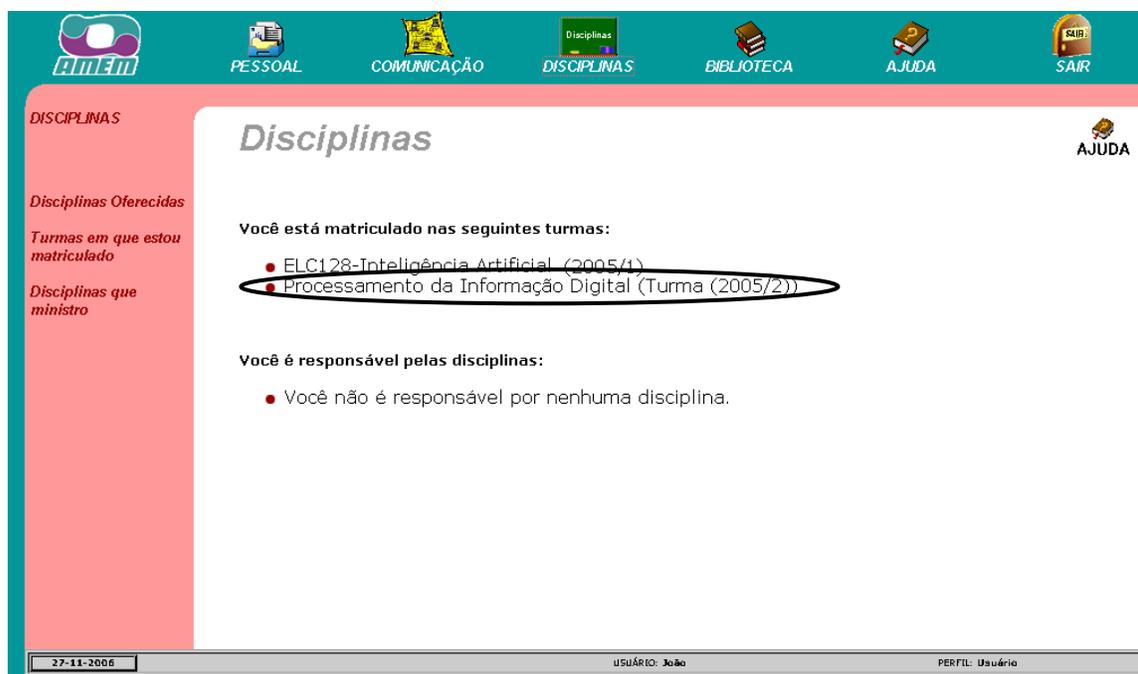


Figura 14 - Aluno entra nas *Disciplinas* como *usuário* no AMEM 1.0

**Atividades da disciplina**

*Lista de Atividades*

Atividade	Data/Hora Início	Ativ.Extra-classe	Ativ.Colaboração
Aula 01 - Documentos Multimídia I	23/01/2006 - 13:30:00	Não	Não
Aula 02 - Documentos Multimídia II	30/01/2006 - 13:30:00	Sim (Encerrada)	Sim (Encerrada)
Aula 03 - Informação Digital	06/02/2006 - 13:30:00	Não	Não
Metadados para Arquivos Digitais	22/02/2006 - 00:00:00	Sim (Encerrada)	Não

27-11-2006    USUÁRIO: João    PERFIL: Aluno    DISCIPLINA: Processamento de Informação Digital    TURMA: Turma (2005/2)

Figura 15 - Aluno entra em uma disciplina específica no AMEM 1.0

Desta forma, se o aluno pretende, agora, visualizar o mural ou as notícias de sua turma, ele pode retornar ao menu *Comunicação*. Nota-se, então, uma mudança no comportamento do sistema, que passa a exibir também os fóruns, salas de discussão, notícias e murais da turma, como pode ser visualizado na figura 16, letra a. A mudança no perfil do aluno e sua condição de imersão à disciplina específica se mantém no ambiente (letras b, c, d da figura 16).

**Comunicação**

*Fóruns*

Condição	Assunto	Proponente
Encerrado	Software Livre e Educação Dialógica-Problematicadora	fbastos

*Notícias*

Não há notícias.

*Mural*

**Mural da Turma**

► **Aviso :Atividade Extra-Classe**

Alunos! Foi definida a atividade extra-classe para a próxima aula. Entrem na Disciplina, cliquem na última aula (Documentos Multimídia - Parte II) e acessem a Atividade Extra-Classe. Lá se encontram as instruções para responder a pergunta deixada.

27-11-2006    USUÁRIO: João    PERFIL: Aluno    DISCIPLINA: Processamento de Informação Digital    TURMA: Turma (2005/2)

Figura 16 - Aluno entra na *Comunicação* com o perfil de *aluno* no AMEM 1.0

Completando o ciclo, se o aluno deseja visualizar o mural de outra disciplina, ele deve *sair* explicitamente da disciplina onde ele está inserido. Para tanto, ele deve selecionar o menu *Sair* (figura 17, letra a) e escolher a opção *sair da turma* (figura 17, letra b). Desta forma, ele será automaticamente levado ao menu *Disciplinas* (figura 14), onde o aluno poderá escolher uma nova disciplina, repetindo o ciclo. Cabe salientar que a segunda opção apresentada na figura 17 representa uma saída total do ambiente. Neste caso, o aluno deveria fornecer novamente o seu *login* e *senha* para retornar ao sistema.

Para evitar esta sobrecarga de informações e diminuir o número de passos necessários para se chegar à informação desejada, a nova versão teve como opção de projeto a exclusão definitiva da entrada/saída explícita de uma determinada disciplina. No sistema AMEM 2.0, o aluno pode passar de uma disciplina para a outra sem necessariamente ter que *sair* da anterior, economizando tempo. Utilizando a metodologia da *migalha de pão*, ele sempre tem a informação de onde está e como pode retornar a um estado anterior para chegar a outra opção. As figuras 18 a 22 expõem a mesma situação descrita anteriormente para a versão 1.0, apresentando a nova solução.

Há duas formas de um aluno verificar os murais/notícias de uma determinada disciplina. A figura 18 apresenta a interface do ambiente quando o aluno faz a sua entrada no sistema. No lado esquerdo, o segundo menu (letra a), são visualizadas todas as disciplinas que o mesmo está matriculado. A primeira opção de visualização do mural é simplesmente selecionar uma das disciplinas. Neste caso, o aluno será transportado para a interface da figura 19.

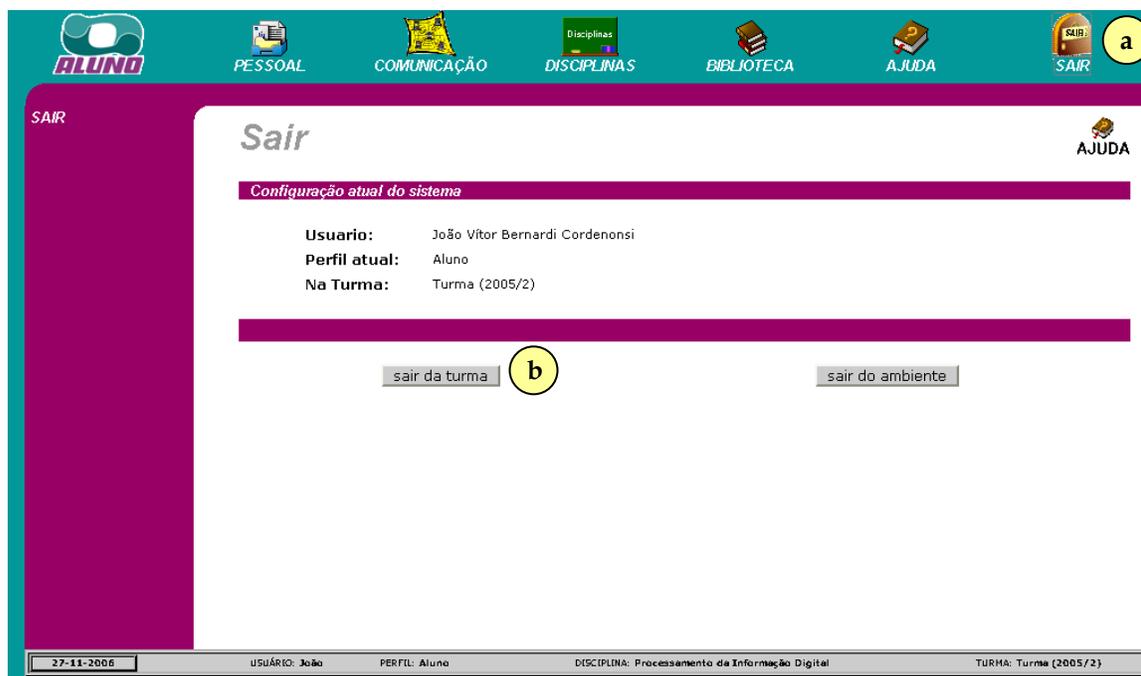


Figura 17 - Aluno entra em *Sair* com o perfil de *aluno* no AMEM 1.0

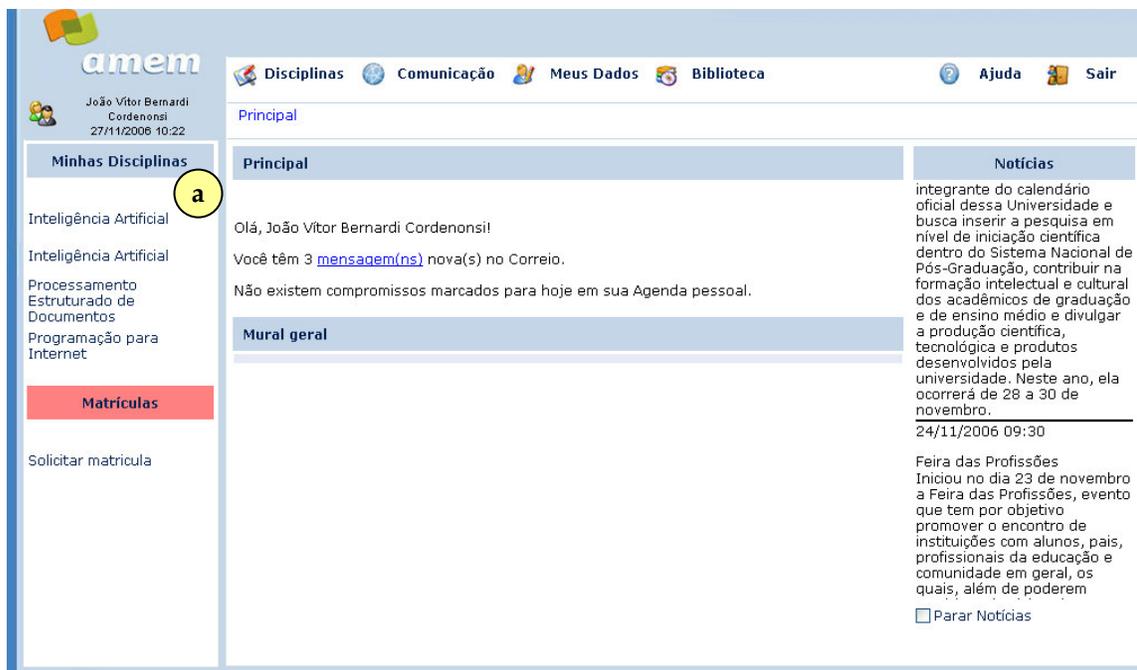


Figura 18 - Aluno entra no ambiente do AMEM 2.0

Ao selecionar uma disciplina específica, três áreas são alteradas substancialmente. O segundo menu (letra a, figura 19) apresenta as opções do aluno referentes à disciplina, a saber, *Início* (tela inicial de boas vindas, representada na figura 19), *Aulas* (aulas agendadas para esta disciplina), *Atividades* (atividades agendadas para esta disciplina), *Colaborações* (atividades de colaboração agendadas para esta disciplina), *Lista de participantes* (contendo o nome, perfil, endereço eletrônico externo, possibilidade de enviar uma mensagem interna e visualização das pastas virtuais de todos os alunos e professores da disciplina), *Bibliografia* (cadastrada para a disciplina, dividida em básica e complementar) e *Ementa e programa*. O menu *migalha de pão* passa a representar a localização do aluno (figura 19, letra b). No caso, ele informará ao mesmo que ele se encontra dentro de uma disciplina, de nome *Processamento Estruturado de Documentos* e que, ainda, está na opção *Início*. Finalmente, a tela principal (figura 19, letra c) apresentará uma mensagem de boas vindas, um aviso informando se há atividades pendentes para a disciplina em questão. Logo abaixo, está o esperado mural da disciplina e, à sua direita, as notícias gerais do ambiente e específicas da disciplina.

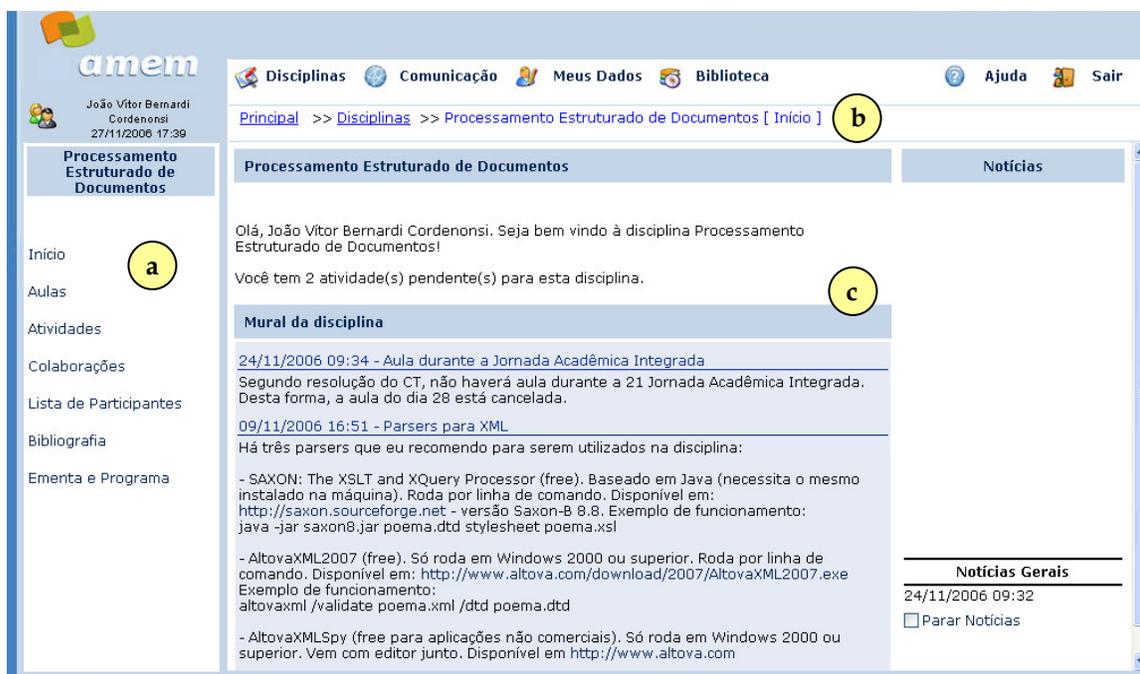


Figura 19 - Aluno ingressa em uma disciplina no AMEM 2.0

Para que o aluno acesse o mural de outra disciplina, ele deve retornar à sua lista de disciplinas matriculadas. Esta ação é realizada através do menu *migalha de pão* (figura 19, letra b) ou através da opção *Disciplinas* do menu principal. Ao acessar qualquer uma destas opções, o segundo menu apresentará novamente as disciplinas onde o aluno está matriculado (figura 20, letra a) e a interface principal estará preenchida com as atividades pendentes para cada disciplina (figura 20, letra b). Desta forma, ao selecionar qualquer disciplina no segundo menu, o ambiente apresentará uma interface semelhante ao da figura 19, onde serão apresentados todos os murais e notícias da disciplina escolhida. Esta seqüência de ações elimina a necessidade de *sair* explicitamente de uma disciplina. Ao retornar à interface que contém a lista de disciplinas (figura 20), o aluno automaticamente está se desligando de uma disciplina específica. A principal forma de percepção do aluno para o que ocorreu está no menu *migalha de pão*, onde, agora, não há mais o nome da disciplina exposto.

Além da seqüência estabelecida anteriormente, o aluno também tem uma nova opção para visualizar as notícias e murais das disciplinas. Para tanto, basta ele selecionar o menu *Comunicação* e o ambiente terá uma nova configuração, como apresentada na figura 21.

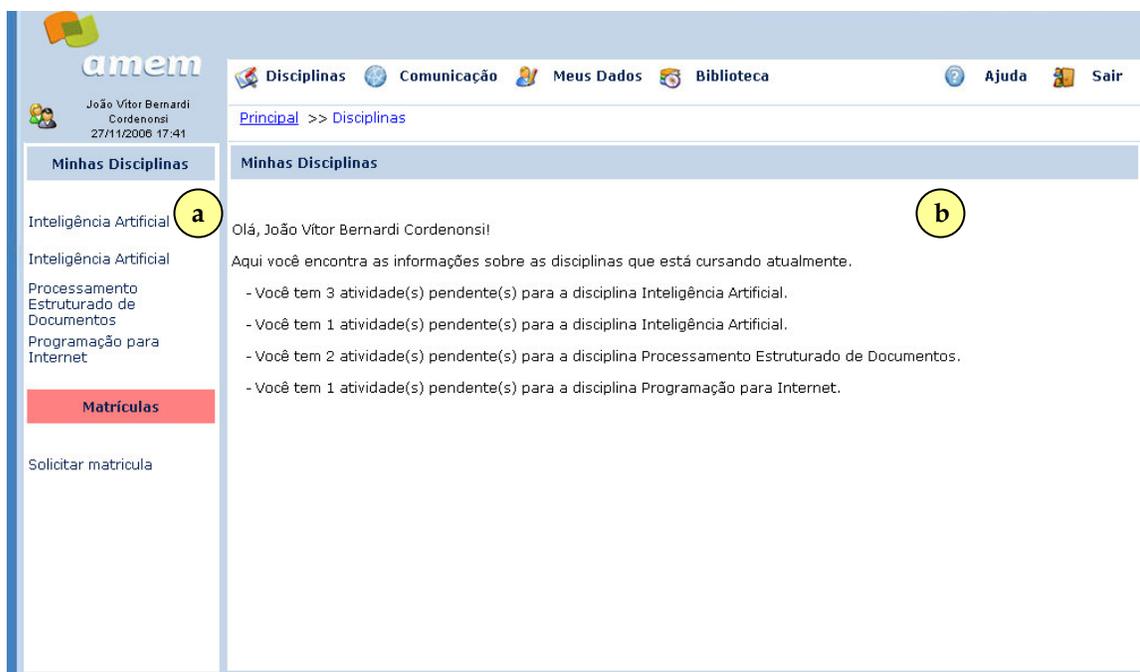


Figura 20 - Menu *Disciplinas* no AMEM 2.0



Figura 21 - Menu *Comunicação* no AMEM 2.0

Nesta interface, basta ele selecionar uma das opções do segundo menu (*notícias* ou *mural*) e o aluno poderá visualizar todas as notícias ou murais expostos para todas as disciplinas, bem como as notícias e murais gerais. A figura 22 apresenta a interface para a seleção da opção *mural*. Primeiramente, o aluno poderá visualizar os murais gerais. Seguindo na lista, ele poderá visualizar todos os murais de todas as disciplinas onde o aluno está matriculado, separados por disciplina.

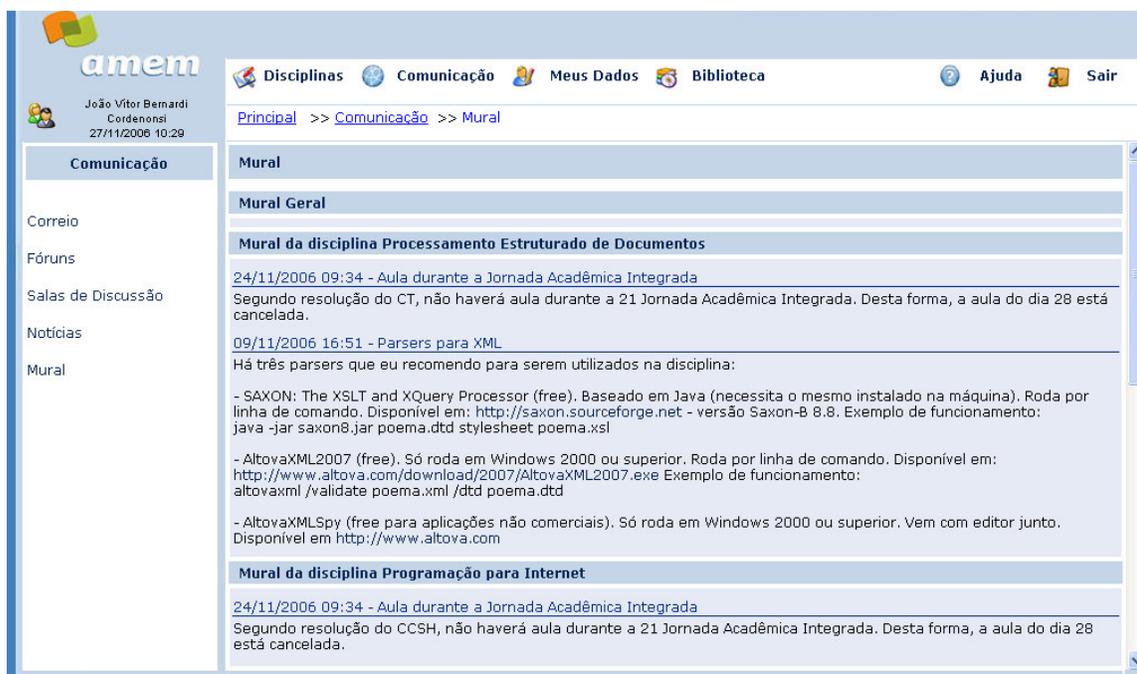


Figura 22 - Aluno visualiza os murais da disciplina e gerais no AMEM 2.0

### *Novos Operacionalizadores Pedagógicos*

A necessidade de construir novos operacionalizadores pedagógicos, incluindo também formas de utilização do ambiente que não estejam estritamente baseadas na concepção pedagógica original, se mostrou uma preocupação constante dos utilizadores. A opção de projeto para a nova implementação do AMEM 2.0 focou na possibilidade da criação, inserção e manutenção de novos operacionalizadores pedagógicos através da implementação de objetos educacionais manipuláveis (objetos de aprendizagem). Esta possibilidade é um objetivo interessante por si mesmo e fornece subsídios e ferramental importante para os diversos aspectos das ações educacionais dos professores. Ao mesmo tempo, propicia aos docentes, cujo embasamento teórico-metodológico diverge do escolhido para o sistema, uma gama maior de práticas educacionais que não são completamente atreladas ao sistema original. Da mesma forma, a redefinição dos três operacionalizadores pedagógicos como uma ferramenta importante, mas não única e não obrigatória, auxilia os docentes que desejam experimentar outras metodologias pedagógicas. Sendo assim, os operacionalizadores escolhidos pelos professores refletirão suas escolhas pedagógicas, e não o contrário.

O cadastramento de um objeto de aprendizagem no ambiente AMEM 2.0 é realizado através do preenchimento de um cadastro baseado no *BaseScheme* do *Learning Objects Metadata*, dividido em dez etapas, que são explicadas em detalhes no capítulo 5. Assim como os demais operacionalizadores, os objetos de aprendizagem são cadastrados através de um assistente. Primeiramente, conforme demonstra a figura 23, o professor e/ou administrador precisa informar as características gerais do objeto, tais como um identificador, nome e linguagem.

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 1 - Informações Gerais**

Identificador:\*  ?

Título:\*  ?

Linguagem:\*  ?

Descrição:\*  ?

Palavras-chave:\*  ?

Cobertura:\*  ?

Estrutura:\*  ?

Nível de agregação:\*  ?

Figura 23 - Assistente para criação de um novo objeto de aprendizagem – Passo 1

O próximo passo (figura 24), é detalhar as características do objeto em relação a um esquema de catalogação formal.

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 1.1 - Informações Gerais - Catalogação**

Título:

Catálogo:\*  ?

Entrada:\*  ?

**Catálogos do objeto de aprendizagem**

#	Catálogo	Entrada	Ações

Figura 24 - Assistente para criação de um novo objeto de aprendizagem – Passo 1.1

Iniciando a segunda etapa, é necessário informar a versão e o estado atual do objeto em relação a sua completude(figura 25).

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 2 - Ciclo de Vida**

Título: Lobo - Learning Object Based on Optimization

Versão:\* 1.0

Estado:\* Final

[Próximo >>](#)

Figura 25 - Assistente para criação de um novo objeto de aprendizagem – Passo 2

As entidades, físicas e jurídicas, que atuaram na concepção e desenvolvimento do objeto, devem ser cadastradas na guia de *Contribuições* (figura 26).

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 2.1 - Ciclo de Vida - Contribuições**

Título: Lobo - Learning Object Based on Optimization

Papel:\* Autor

Entidade:\*

Data da contribuição:\* dd/mm/aaaa

[Cadastrar contribuição](#) [Próximo >>](#)

**Catálogos do objeto de aprendizagem**

#	Entidade	Papel	Data	Ações
1	<a href="#">Andre Zanki Cordenonsi</a>	Autor	15/01/2007	 
2	<a href="#">Felipe Martins Müller</a>	Autor	15/01/2007	 

Figura 26 - Assistente para criação de um novo objeto de aprendizagem – Passo 2.1

A figura 27 representa a terceira etapa no cadastramento, onde os dados sobre os metadados utilizados devem ser informados.

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 3 - Meta metadados**

Título: Lobo - Learning Object Based on Optimization

Identificador:\*  ?

Esquema:\*  ?

Linguagem:\*  ?

**Próximo >>**

Figura 27 - Assistente para criação de um novo objeto de aprendizagem – Passo 3

Esta etapa ainda necessita registrar a catalogação dos metadados utilizados (figura 28) e os principais contribuidores que criaram os metadados utilizados (figura 29).

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 3.1 - Meta metadado - Catalogação**

Título: Lobo - Learning Object Based on Optimization

Catálogo:\*  ?

Entrada:\*  ?

**Cadastrar catálogo** **Próximo >>**

**Catálogos do metadados**

#	Catálogo	Entrada	Ações
1	<a href="#">IEEE Standard for Learning Object Metadata</a>	IEEE 1484.12.1-2002	

Figura 28 - Assistente para criação de um novo objeto de aprendizagem – Passo 3.1

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 3.2 - Meta metadado - Contribuições**

Título: Lobo - Learning Object Based on Optimization

Papel:\*  ?

Entidade:\*  ?

**Cadastrar contribuição** **Próximo >>**

**Catálogos do metadados**

#	Entidade	Papel	Ações
1	<a href="#">IEEE</a>	Criador	

Figura 29 - Assistente para criação de um novo objeto de aprendizagem – Passo 3.2

A quarta etapa informa os requisitos técnicos do sistema (figura 30). É importante ressaltar, nesta parte do assistente, o campo *Localização*. Se o usuário já possui o objeto disponível em algum servidor, ele pode preencher este campo com o caminho completo do objeto e o AMEM automaticamente apontará o sistema para o servidor especificado. Se este campo permanecer vazio, ao final do cadastramento do objeto é aberta uma opção para carregar o objeto para o servidor.

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 4 - Técnica**

Título: Lobo - Learning Object Based on Optimization

Formato:\*  ou

Tamanho:\*  em bytes

Localização:

Duração:\*  em minutos

Instalação: Não é necessário instalar o LOBO, mas o sistema operacional deve possuir o Java Runtime Enviroment.

Outras plataformas requeridas:

**Próximo >>**

Figura 30 - Assistente para criação de um novo objeto de aprendizagem – Passo 4

A figura 31 apresenta os requerimentos técnicos em relação a dependência de sistema operacional ou navegador *internet*.

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 4.1 - Técnica - Requerimentos Técnicos**

Título: Lobo - Learning Object Based on Optimization

Tipo:\*

Nome:\*

Relação lógica:\*

Versão mínima:  Versão máxima:

**Cadastrar requerimento** **Próximo >>**

**Requerimentos técnicos**

#	Tipo	Nome	Mínimo	Máximo	Relação	Ações
1	<a href="#">Navegador</a>	NV - qualquer			E	

Figura 31 - Assistente para criação de um novo objeto de aprendizagem – Passo 4.1

A quinta etapa do cadastramento de um objeto baseado na especificação LOM relaciona os aspectos educacionais (figura 32).

Assistente para Cadastramento de Objetos de Aprendizagem	
Passo 5 - Educacional	
Título:	Lobo - Learning Object Based on Optimization
Tipo de interação:*	Ambos ?
Nível de interatividade:*	Alto ?
Papel do usuário:*	Estudante ?
Dificuldade:*	Médio ?
Tempo de aprendizagem:*	120 minutos ?
Tipo de recurso:*	Simulação ?
Densidade semântica:*	Média ?
Contexto educacional:*	Graduação/Pós-graduação ?
Idade típica:*	18 a 25 ?
Linguagem:*	pt ?
Descrição:	<div style="border: 1px solid black; height: 30px; width: 100%;"></div> ?

[Próximo >>](#)

Figura 32 - Assistente para criação de um novo objeto de aprendizagem – Passo 5

A figura 33 apresenta a sexta etapa, relativa aos direitos e custos de utilização do objeto que está sendo cadastrado.

Assistente para Cadastramento de Objetos de Aprendizagem	
Passo 6 - Direitos	
Título:	Lobo - Learning Object Based on Optimization
Custos:*	Não ?
Restrições de copyright:*	Não ?
Descrição:	Este objeto está disponível livremente através da GNU/GENERAL PUBLIC LICENSE, Versão 2, de junho de 1991.   ?

[Próximo >>](#)

Figura 33 - Assistente para criação de um novo objeto de aprendizagem – Passo 6

Um objeto pode possuir relações de dependência, interdependência ou até mesmo se basear, em maior ou menor grau, com outros objetos cadastrados. Na sétima etapa do cadastramento, é possível informar tais relações, conforme a figura 34.

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 7 - Relações**

Título: Lobo - Learning Object Based on Optimization

Tipo:\*  ?

Identificador do recurso:\*  ?

Catálogo do recurso:  ?

Entrada no catálogo:  ?

Descrição:  ?

**Relações com o objeto de aprendizagem**

#	Tipo	Identificador	Ações
---	------	---------------	-------

Figura 34 - Assistente para criação de um novo objeto de aprendizagem – Passo 7

Na figura 35, o usuário pode armazenar informações adicionais sobre o objeto, no espaço das *Anotações*.

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 8 - Anotações**

Título: Lobo - Learning Object Based on Optimization

Entidade:\*  ?

Data da anotação:\*  ?

Descrição:\*  ?

**Anotações do objeto de aprendizagem**

#	Entidade	Data	Ações
---	----------	------	-------

Figura 35 - Assistente para criação de um novo objeto de aprendizagem – Passo 8

Finalmente, na última etapa, são cadastradas as classificações possíveis do objeto(figura 36) e a taxonomia utilizada nestas classificações, se existirem(figura 37).

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 9 - Classificação**

Título: Lobo - Learning Object Based on Optimization

Propósito:\*  ?

Palavras-chave:\*  ?

Descrição:  ?

**Classificações do objeto de aprendizagem**

#	Propósito	Palavras-chave	Ações
1	Disciplina	<a href="#">Heurísticas e Metaheurísticas</a>	
2	Referencia	<a href="#">Graduação, Pós-Graduação</a>	

Figura 36 - Assistente para criação de um novo objeto de aprendizagem – Passo 9

**Assistente para Cadastramento de Objetos de Aprendizagem**

**Passo 9.1 - Classificação - Taxonomia**

Título: Lobo - Learning Object Based on Optimization

Classificação:\*

Fonte:\*  ?

Identificador:\*  ?

Entrada:\*  ?

**Taxonomias do objeto de aprendizagem**

#	Classificação	Fonte	Identificador	Entrada	Ações
---	---------------	-------	---------------	---------	-------

Figura 37 - Assistente para criação de um novo objeto de aprendizagem – Passo 9.1

Após a realização de todos os passos para o cadastramento de um objeto LOM no ambiente, ainda é necessário um passo adicional, utilizado apenas para a estrutura interna do AMEM. A figura 38 apresenta este passo, onde o usuário pode armazenar os parâmetros utilizados para iniciar a aplicação representada por um objeto. É possível armazenar até quatro parâmetros, além de um pequeno texto de ajuda que é exibido ao usuário que estiver realizando a ligação do objeto com uma aula ou uma atividade. É importante observar que, uma vez cadastrado um objeto no AMEM, este fica disponível para todos os usuários que possuam o perfil professor e administrador. Logo, o usuário que cadastra o objeto pode não

ser o único utilizador do mesmo e, deste modo, é importante que ele forneça uma explicação sobre os possíveis parâmetros de configuração do seu objeto, caso eles existam.

A segunda parte deste cadastro representa os dois arquivos que podem ser carregados para o servidor. O primeiro representa o objeto propriamente dito e o segundo representa a possibilidade de visualização da ligação do ambiente com o objeto através de uma imagem iconográfica. Caso o usuário decida não utilizar um ícone específico, o AMEM possui uma imagem padrão que é colocada no seu lugar. Ao preencher estes últimos dados, o objeto é carregado no servidor, se for o caso, e o assistente finaliza.

Assistente para Cadastramento de Objetos de Aprendizagem	
Passo 10 - Parâmetros e arquivos	
Título:	Lobo - Learning Object Based on Optimization
Parâmetro Um [Nome]:	<input type="text"/>
Parâmetro Um [Ajuda]:	<input type="text"/> 
Parâmetro Dois [Nome]:	<input type="text"/>
Parâmetro Dois [Ajuda]:	<input type="text"/> 
Parâmetro Três [Nome]:	<input type="text"/>
Parâmetro Três [Ajuda]:	<input type="text"/> 
Parâmetro Quatro [Nome]:	<input type="text"/>
Parâmetro Quatro [Ajuda]:	<input type="text"/> 
Ícone:	<input type="text"/> <input type="button" value="Procurar..."/> 
Objeto:	<a href="http://openpower-amem.ce.ufsm.br:8080/amem/objetos/lobo.jnlp">http://openpower-amem.ce.ufsm.br:8080/amem/objetos/lobo.jnlp</a> 
<input type="button" value="Próximo &gt;&gt;"/>	

Figura 38 - Assistente para criação de um novo objeto de aprendizagem – Passo 10

### Nomenclatura, Atividades e Agendamentos

A nomenclatura do ferramental disponível para os professores e alunos se tornou um empecilho ao entendimento do sistema e à visualização e leitura dos operacionalizadores pedagógicos. Para compreender tal fenômeno, salienta-se a seguir de que forma os docentes poderiam estruturar suas atividades dentro do ambiente através dos operacionalizadores disponíveis.

O AMEM 1.0 possuía três operacionalizadores: *programação*, *atividade extra-classe* e *colaboração*. O primeiro operacionalizador, *programação*, se referia ao programa estabelecido para uma determinada aula/encontro (figura 39). Neste, era possível cadastrar o nome e os elementos da Matriz Dialógica Problematicadora que o docente pretendia

responder em relação aquela aula/encontro(figura 39, letra a). Após este cadastro básico, o professor poderia *registrar* a aula(figura 39, letra b).



Figura 39 - Operacionalizador *programação* no AMEM 1.0

Um registro de uma programação possuía um tempo (em minutos), um assunto e uma ordem (em relação aos demais registros daquela programação) (figura 40). Uma vez definido um *programa* para uma disciplina, este poderia ser utilizado em todas as turmas da mesma, poupando o retrabalho de definir novamente a estrutura da aula/encontro novamente. Logo, um professor só pode definir uma *programação* quando estiver inserido no *locus* virtual da uma determinada disciplina. Ao ingressar em uma turma, ela não pode gerenciar as programações definidas anteriormente. A sua localização em relação a disciplina e/ou turma em que o docente se encontra está disponível na barra de estado (figura 39, letra c). A premissa de reutilização dos operacionalizadores também é verdadeira para as *atividades extra-classe* e a *colaboração*.

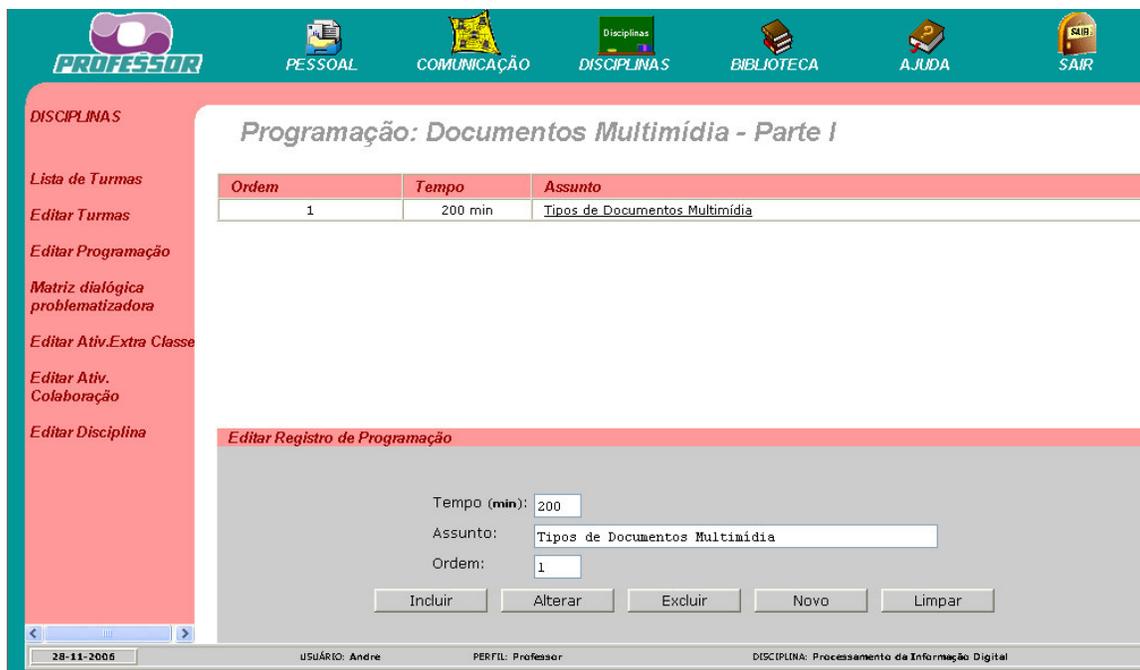


Figura 40 - Registros de uma *programação* no AMEM 1.0

O operacionalizador *Atividade extra-classe* segue a mesma estrutura básica da *programação*. Inserido em uma disciplina, o docente pode definir a atividade através do cadastro do seu nome e das observações referentes à mesma, onde o professor poderá detalhar as questões que ele pretende serem respondidas pelos alunos. A figura 41 apresenta a interface de cadastro deste operacionalizador.

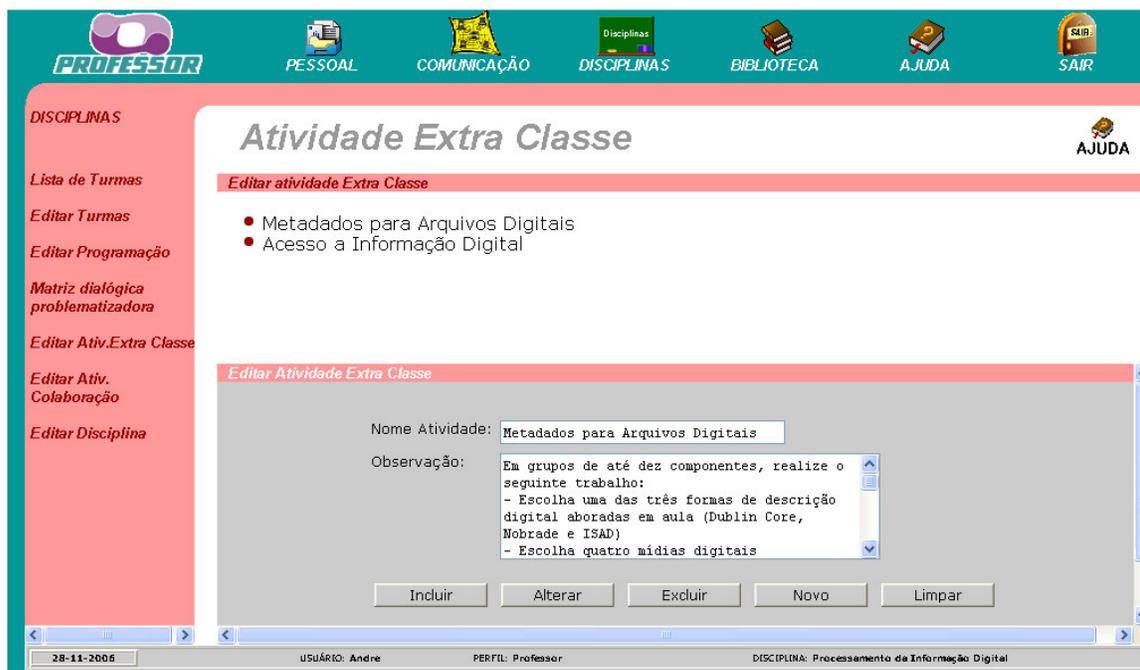


Figura 41 - Operacionalizador *atividade extra-classe* no AMEM 1.0

Finalmente, o terceiro e último operacionalizador da primeira versão do ambiente foi denominado *atividade de colaboração*. Esta atividade compreendia a possibilidade do aluno contribuir de forma decisiva com a próxima aula/encontro e/ou responder alguma questão que remetesse aos conteúdos didáticos que seriam discutidos posteriormente. Novamente, seguindo a mesma premissa dos anteriores, o docente deveria estar inserido em uma disciplina para definir/gerenciar uma atividade de colaboração. O cadastro da mesma envolvia a definição do seu nome, a escolha da *programação* a qual a colaboração estava vinculada (ou seja, era necessário ter uma programação futura prévia para que a colaboração pudesse ser efetuada) e duas escolhas: se o aluno responderia com a colaboração diretamente no sistema (*área para colaboração*) e se o docente pretendia anexar uma bibliografia para análise (*literatura recomendada*) (figura 42, letra a). É importante salientar que somente o operacionalizador *atividade de colaboração* possuía a opção de inserir bibliografias (virtuais ou não) para o aluno. Os demais operacionalizadores não possuíam esta opção.

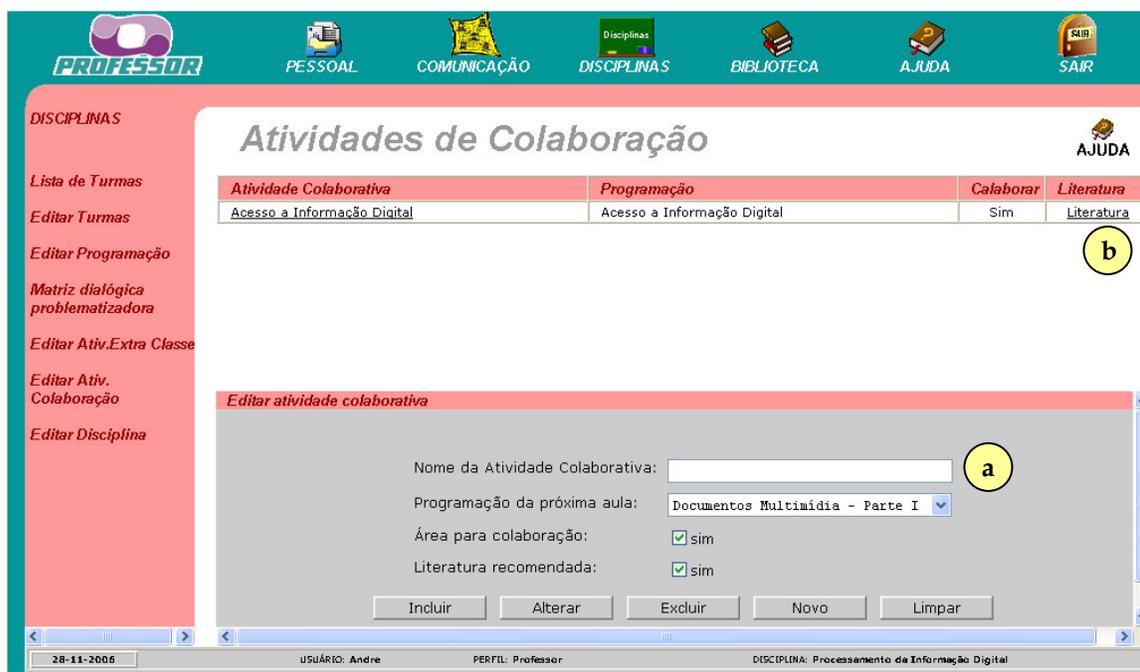


Figura 42 - Operacionalizador *Atividade de Colaboração* no AMEM 1.0

Após cadastrar uma atividade de colaboração, o docente poderia anexar as bibliografias previamente cadastradas na biblioteca do ambiente. Para realizar esta ação, ele deveria selecionar a opção *literatura* (figura 42, letra b) da atividade desejada. O ambiente modificaria a janela principal, apresentando a interface da figura 43. Nesta, era possível incluir uma nova bibliografia, após selecionar a lista de títulos disponíveis para um determinado assunto (figura 43, letra a), ou excluir uma bibliografia já cadastrada, removendo a mesma da *atividade de colaboração* (figura 43, letra b).

**Atividade Colaborativa: Acesso a Informação Digital**  
**Literatura Recomendada**

Textos selecionados			Finalizar	
Autor	Título	Fonte		
Andressa Piconi e Humberto Innarelli	Padrões de Formatos de Documentos Digitais Adotados pelo Arquivo Permanente do Sistema de Arquivos..	Link		Excluir
Börje Karlsoon, Luciano Barbosa, Mariano Cravo e Thiago Santos	Musical Information Retrieval e Bibliotecas Virtuais	Link		Excluir
Carlos Xavier de Azevedo Netto, Bernardina Maria Juvenal Freire e Perpétua Pereira	A Representação de Imagens no Acervo da Biblioteca Digital Paulo Freire - Proposta e Percursos	Link		Excluir
Clarissa Costa e Lima, Gláucia da Silva Tavares, Fernanda Martins Vieira, Claudina Dutra Moresi...	Sistema de Obtenção de Dados Multimídia para a Pesquisa Guignard	Link		Excluir

**Seleção os textos desejados na lista abaixo**  
**Assunto: AMBIENTES VIRTUAIS DE APRENDIZAGEM**

- ALMEIDA, Maria Elizabeth Bianconcini de, *TECNOLOGIA E EDUCAÇÃO A DISTÂNCIA: ABORDAGENS E CONTRIBUIÇÕES DOS AMBIENTES DIGITAIS E INTERATIVOS*, 2003. [Incluir](#)
- Felipe Martins Müller e outros, *Projeto FINBP - Laboratório para Educação Mediada por Tecnologia*, 2003. [Incluir](#)

28-11-2006    USUÁRIO: André    PERFIL: Professor    DISCIPLINA: Processamento da Informação Digital

Figura 43 - Bibliografias do operacionalizador *Atividade de Colaboração* no AMEM 1.0

No entanto, o mero cadastramento dos operacionalizadores para uma disciplina não implica diretamente na disponibilização dos mesmos para os alunos. Considerando a premissa que, uma vez cadastrado um operacionalizador, este pode ser utilizado diversas vezes em diversas turmas da mesma disciplina, é necessário um passo adicional para que os alunos realmente tenham acesso à estrutura preparada pelo professor. No AMEM 1.0, os três elementos poderiam ser agendados, em grupo ou individualmente, sob um encapsulador de alto nível, que era apresentando aos alunos e professores como *atividades*. Desta forma, ao entrar no ambiente, os mesmos tinham acesso às *atividades* da turma e, ao selecionar uma determinada *atividade*, ele poderia navegar pelos três operacionalizadores agendados. Cabe salientar que é possível, também, realizar o agendamento de apenas um operacionalizador, sem a co-relação com os demais.

Independente do número de operacionalizadores agendados concomitantemente, esta ação só era permitida para o professor que ingressasse em uma determinada turma. Após escolher a opção *Lista de Turmas* (figura 43, letra c), era apresentado ao docente o rol de turmas da disciplina em questão, onde o mesmo poderia escolher aquela que lhe interessasse. Uma vez dentro da turma em questão, a opção correspondente era a *Agendamento de atividade* (figura 44, letra a), que apresentaria a interface do agendamento dos operacionalizadores tal como demonstrado na figura 44, letra b. Para realizar um determinado agendamento, era necessário cadastrar o nome da atividade e a data e hora em que a mesma estaria disponível para visualização dos alunos. O *conteúdo* de uma atividade foi definido como os três operacionalizadores, que deviam estar previamente cadastrados. Desta forma, era possível escolher uma programação existente, uma atividade extra-classe (onde era necessário fornecer a data e hora de início e de finalização da mesma, sendo que o sistema impedia o envio de resposta fora do prazo) e uma atividade de colaboração (com o cadastramento de início e hora, nos mesmos moldes da atividade extra-classe).

Finalmente, após o preenchimento deste formulário, o aluno matriculado na turma específica desta disciplina (figura 44, letra c e letra d) teria acesso aos operacionalizadores definidos pelo docente.

A formatação das atividades em operacionalizadores reutilizáveis evitava o retrabalho dos docentes, quando da aplicação dos mesmos em diversas turmas no mesmo semestre ou em turmas diferentes em semestres subsequentes. Após a definição do operacionalizador, o mesmo poderia ser reutilizado em diversas turmas, poupando trabalho e agilizando o trabalho docente.

**Atividades**

**Lista de Atividades**

Atividade	Data/Hora Início	Ativ.Extra-classe	Ativ.Colaboração
<a href="#">Aula 01 - Documentos Multimídia I</a>	23/01/2006 - 13:30:00	Não	Não
<a href="#">Aula 02 - Documentos Multimídia II</a>	30/01/2006 - 13:30:00	Sim (Encerrada)	Sim (Encerrada)
<a href="#">Aula 03 - Informação Digital</a>	06/02/2006 - 13:30:00	Não	Não
<a href="#">Metadados para Arquivos Digitais</a>	22/02/2006 - 00:00:00	Sim (Encerrada)	Não

**Editar Atividade**

**Início da Atividade**

Nome da Atividade:

Data (dd/mm/aaaa):  Hora (hh:mm):

**Conteúdo**

Programação:

Ativ.Extra Classe:

Data início (dd/mm/aaaa):  Data fim (dd/mm/aaaa):

Hora início (hh:mm):  Hora fim (hh:mm):

Colaboração:

Data início (dd/mm/aaaa):  Data fim (dd/mm/aaaa):

Hora início (hh:mm):  Hora fim (hh:mm):

28-11-2005 USUÁRIO: Andre PERFIL: Professor DISCIPLINA: Processamento da Informação Digital TURMA: Turma (2005/2)

Figura 44 - Agendamento de uma atividade no AMEM 1.0

A opção estrutural do AMEM 1.0 em dividir o cadastramento dos operacionalizadores de sua real utilização se mostrou bastante interessante. Este foi um dos pontos positivos mais citados pelos professores da instituição que utilizaram o sistema. No entanto, duas preocupações também apareciam constantemente para os docentes. Primeiramente, a nomenclatura, como citada anteriormente, acabava por confundir os professores. A existência de uma atividade genérica, encapsuladora, que cobria o agendamento de outras duas atividades e de uma programação trazia uma sobrecarga de conhecimentos que sobrepujava os balizadores pedagógicos. O docente trazia mais questões em relação ao fluxo tecnológico do que as preocupações pedagógicas existentes no ambiente. De acordo com as asserções acima, o próprio agendamento foi criticado por

fornecer informações sobressalentes, caso o professor não precisasse ou tivesse a intenção de utilizar todos os operacionalizadores pedagógicos.

Como decisão de projeto para a segunda versão do ambiente, os operacionalizadores foram renomeados para *aula*, *atividade* e *colaboração*, sendo que os mesmos são completamente independentes entre si. O professor, se desejar, pode realizar a conexão entre os mesmos, mas tal via não é obrigatória. Para evitar que os professores se confundam com a utilização de nomes parecidos para coisas distintas (*atividade extra-classe* como um possível sub-elemento de *atividade*), a nomenclatura foi alterada. Da mesma forma, os *registros* de uma *programação* foram substituídos por *pontos* de uma *aula*. Ao aproximar a nomenclatura à ontologia utilizada usualmente pelos docentes, a sobrecarga cognitiva causada pelo treinamento da ferramenta tecnológica pode se tornar menos árdua, levando o professor a pensar mais no seu fazer educacional do que na utilização do ambiente por si só. Mais uma vez é importante lembrar que um AVEA, independente de qual seja, é apenas uma ferramenta nas mãos do professor e do aluno.

No AMEM 2.0, o cadastramento dos operacionalizadores (bem como grande parte dos demais formulários) é feito através de *assistentes*. Estas pequenas ferramentas guiam o docente, passo a passo, no preenchimento dos formulários e na correta formatação de suas estruturas. Assim como a primeira versão do ambiente, se optou pela distinção entre a criação dos operacionalizadores e seu agendamento. Desta forma, quando um professor se encontra em uma *disciplina*, ele pode cadastrar *aulas*, *colaborações* e *atividades*. Mais tarde, ao ingressar em uma *turma* desta *disciplina*, ele poderá realizar o agendamento, permitindo, deste modo, a reutilização dos operacionalizadores por diversas turmas. A localização do professor em relação ao seu estado se encontra no menu *migalha de pão* (figura 45, letra a). Nesta mesma figura, é apresentada toda a interface do professor quando este se encontra em uma disciplina. No segundo menu (letra b), é possível acessar as opções de cadastramento dos operacionalizadores, informar a bibliografia básica da disciplina (comum para todas as turmas), cadastrar a ementa e programa da disciplina, alocar novos professores para a disciplina, definir a Matriz Dialógica Problematizadora, cadastrar/gerenciar turmas desta disciplina e imprimir relatórios. A figura 45, letra c, também apresenta na sua interface principal a visualização de todas as aulas desta disciplina, pela ocasião da escolha da opção *Cadastrar aulas*. Note que é possível excluir ou alterar uma determinada aula, criar uma nova aula ou realizar um agendamento rápido, que será discutido posteriormente.

A figura 46 apresenta o primeiro passo do assistente para o cadastramento de uma nova *aula* no ambiente. Inicialmente, é necessário fornecer a descrição da aula. Esta descrição não poderá ser semelhante a nenhuma outra já cadastrada na disciplina específica. Disciplinas diferentes podem conter aulas com o mesmo nome. Ao preencher a descrição e apertar o botão próximo, o assistente passará automaticamente para o próximo passo caso todas as informações necessárias estejam corretas. Caso contrário, uma mensagem de alerta mostrando o equívoco é apresentada ao docente.



Figura 45 - Interface para uma disciplina no AMEM 2.0

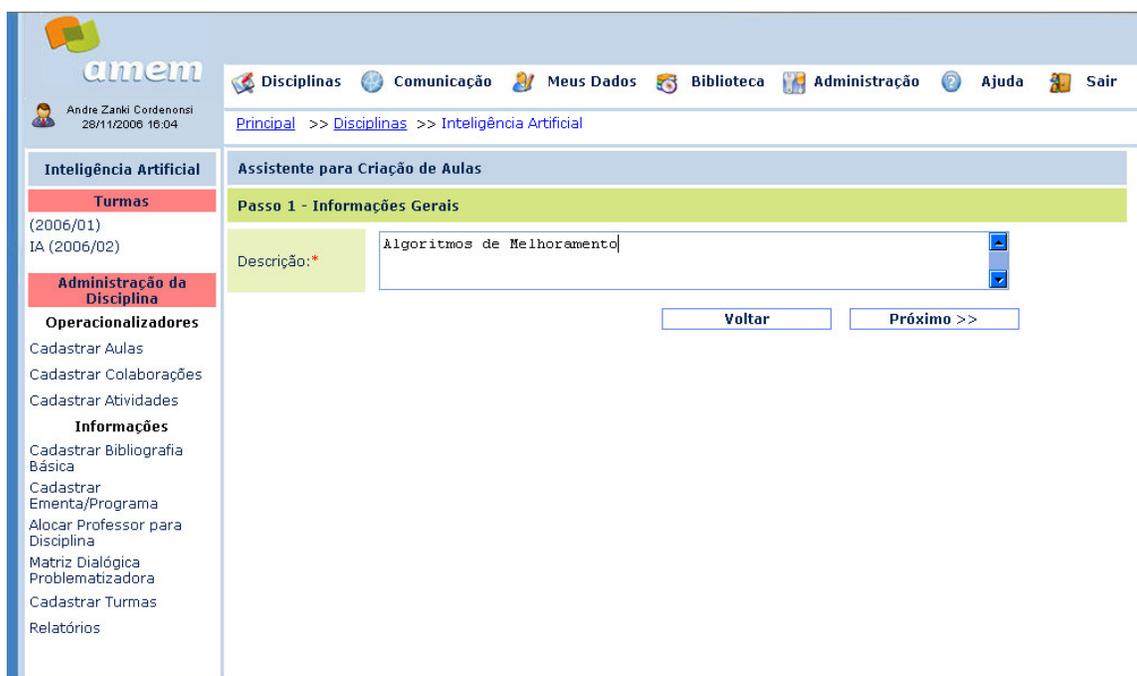


Figura 46 - Assistente para criação de uma nova aula – Passo 1 – AMEM 2.0

O segundo passo para a criação de uma aula é a definição dos *pontos* da mesma. Cada ponto deve conter uma descrição e um tempo (cuja granularidade pode ser escolhida entre minutos, horas, dias, semanas, meses, anos e encontros)(figura 47, letra a). A posição relativa dos pontos de uma mesma aula pode ser alterada graficamente pelas flechas presentes na lista dos pontos da aula (figura 47, letra b).

**Assistente para Criação de Aulas**

**Passo 2 - Pontos da Aula**

Aula: Algoritmos de Melhoria

Descrição: \*

Tempo: \* [ ] Horas

Salvar ponto Próximo >>

**Pontos da aula Algoritmos de Melhoria**

#	Descrição	Tempo	Ações
1	<a href="#">Problemas de um algoritmo construtivo</a>	30 minutos	[Ícone de edição] [Ícone de exclusão] [Ícone de adicionar] [Ícone de remover]
2	<a href="#">Escarpas, platôs e ótimos locais</a>	60 minutos	[Ícone de edição] [Ícone de exclusão] [Ícone de adicionar] [Ícone de remover]
3	<a href="#">Algoritmos de melhoria: 2-OPT</a>	30 minutos	[Ícone de edição] [Ícone de exclusão] [Ícone de adicionar] [Ícone de remover]
4	<a href="#">Algoritmos de Melhoria: 3-OPT</a>	30 minutos	[Ícone de edição] [Ícone de exclusão] [Ícone de adicionar] [Ícone de remover]

Figura 47 - Assistente para criação de uma nova *aula* – Passo 2 – AMEM 2.0

O terceiro passo define os objetos de aprendizagem anexados aos pontos da aula (figura 48). Não é necessário que um ponto de aula possua um objeto de aprendizagem anexado, assim como é possível que um ponto possua mais de um objeto de aprendizagem anexado. Todos os objetos utilizados pelos pontos da aula devem estar disponíveis no ambiente antes de serem anexados.

**Assistente para Criação de Aulas**

**Passo 3 - Anexar Objetos de Aprendizagem**

Aula: Algoritmos de Melhoria

Pontos: \* Problemas de um algoritmo construtivo

Objetos de aprendizagem: \* Ambiente Multimídia para Educação Mediada por Computador

Anexar Objeto de Aprendizagem Próximo >>

**Pontos da aula e seus objetos de aprendizagem**

Ponto de Aula	Objeto de Aprendizagem	Ações
Algoritmos de melhoria: 2-OPT	Sistema de Simulação de Algoritmos Construtivos e de Melhoria para o PCV	[Ícone de exclusão] [Ícone de adicionar]
Algoritmos de Melhoria: 3-OPT	Sistema de Simulação de Algoritmos Construtivos e de Melhoria para o PCV	[Ícone de exclusão] [Ícone de adicionar]

Figura 48 - Assistente para criação de uma nova *aula* – Passo 3 – AMEM 2.0

O quarto passo é utilizado para a escolha da bibliografia da aula. Diferentemente da primeira versão, o AMEM 2.0 permite a anexação de bibliografias tanto para o operacionalizador *aula* como para o *atividade*. A escolha da literatura é feita de forma gráfica através da lista de bibliografias disponíveis previamente na biblioteca. Em destaque é fornecida a bibliografia básica da disciplina (figura 49, letra a). Após esta listagem, todas as bibliografias cadastradas na biblioteca, separadas por assuntos, são fornecidas pelo sistema e podem ser escolhidas pelo docente (figura 49, letra b).



Figura 49 - Assistente para criação de uma nova *aula* – Passo 4 – AMEM 2.0

O próximo passo possibilita a escolha dos elementos da Matriz Dialógica Problematizadora que o docente espera responder durante/após a aula em questão. Ele deve marcar os elementos e anexá-los à referida aula (figura 50). É importante salientar que, caso o docente não queira se utilizar da MDP ou prefira não anexar elementos a esta aula específica, ele pode simplesmente passar para o próximo passo, deixando todas as opções desmarcadas.

Finalmente, após os cinco passos concluídos, é apresentado ao docente um resumo da aula cadastrada e uma mensagem de felicitações, com a opção de *Finalizar o assistente*. Esta interface é visualizada na figura 51.

**amem**

Andre Zanki Cordenonsi  
28/11/2006 16:36

Disciplinas Comunicação Meus Dados Biblioteca Administração Ajuda Sair

Principal >> Disciplinas >> Inteligência Artificial

**Assistente para Criação de Aulas**

**Passo 5 - Matriz Dialógica Problematizadora**

Selecione as questões da Matriz Dialógica Problematizadora que deseja anexar a esta aula:

Aula: Algoritmos de Melhoramento

	(A) Professores	(B) Estudantes	(C) Tema	(D) Contexto
(1) Professores	<input type="checkbox"/> Os professores possuem conhecimentos básicos sobre heurísticas e meta-heurísticas para o binpacking e sobre a investigação ação-educacional que forma a base do AMEM?	<input type="checkbox"/> Os alunos efetivamente estão aproveitando os operacionalizadores pedagógicos para o entendimento das heurísticas e meta-heurísticas?	<input checked="" type="checkbox"/> A abordagem dialógica-problematizadora potencializa o ensino de Heurísticas e Meta-heurísticas?	<input type="checkbox"/> As aulas de heurísticas e meta-heurísticas utilizando a investigação ação-educacional e o AMEM favorecem no processo de ensino-aprendizagem dos professores?
(2) Estudantes	<input type="checkbox"/> Como os professores podem instigar os alunos, utilizando a problematização e o AMEM?	<input type="checkbox"/> Os alunos efetivamente estão compreendendo o que são heurísticas e meta-heurísticas através da abordagem baseada em problemas?	<input checked="" type="checkbox"/> Quais são as dificuldades e os avanços do ensino de Heurísticas e Meta-heurísticas utilizando PCV?	<input type="checkbox"/> Como verificar se os alunos estão compreendendo o conteúdo de Heurísticas e Meta-heurísticas que está sendo desenvolvido através da investigação ação-educacional, presente no AMEM?
(3) Tema	<input type="checkbox"/> Como os professores têm construído a organização didática de suas aulas nas práticas de investigação-ação educacional de Heurísticas e Meta-Heurísticas?	<input type="checkbox"/> Os estudantes tem assumido que os problemas vistos em aula realmente contribuem no processo de aprendizagem de Heurísticas e Meta-Heurísticas?	<input type="checkbox"/> Será que o PCV é adequado para o ensino de Heurísticas e Meta-Heurísticas?	<input type="checkbox"/> As aulas de heurísticas e meta-heurísticas contribuem para a evolução do conhecimento científico dos problemas propostos?

Figura 50 - Assistente para criação de uma nova aula – Passo 5 – AMEM 2.0

**amem**

Andre Zanki Cordenonsi  
28/11/2006 16:26

Disciplinas Comunicação Meus Dados Biblioteca Administração Ajuda Sair

Principal >> Disciplinas >> Inteligência Artificial

**Assistente para Criação de Aulas**

**Finalização do Assistente**

Parabéns, Andre Zanki Cordenonsi, a sua nova aula foi cadastrada com sucesso!

Confira a seguir o resumo da aula cadastrada:

**Finalizar Assistente**

**Disciplina:** Inteligência Artificial

**Descrição da aula:** Algoritmos de Melhoramento

**Pontos:**

Problemas de um algoritmo construtivo	30 minutos	
Escarpas, platôs e ótimos locais	60 minutos	
Algoritmos de melhoramento: 2-OPT	30 minutos	Sistema de Simulação de Algoritmos Construtivos e de Melhoramento para o PCV
Algoritmos de Melhoramento: 3-OPT	30 minutos	Sistema de Simulação de Algoritmos Construtivos e de Melhoramento para o PCV

**Bibliografia:**

- Heuristics: Intelligent search strategies for computer problem solving, *Judea Pearl*, Addison Wesley, 1985.
- O Problema do Caixeiro Viajante, *J.F. Porto da Silveira* UERGS 2000

Figura 51 - Assistente para criação de uma nova aula – Finalização – AMEM 2.0

Como é possível perceber, a implementação do operacionalizador *aula* no novo ambiente trouxe novas possibilidades ao docente. Agora, é facultativo ao professor anexar bibliografias ou objetos de aprendizagem à sua *aula*, apesar de que uma aula desprovida destes elementos de mediação se tornar vazia e incoerente com a prática pedagógica que está sendo desenvolvida.

Contudo, este ferramental mais complexo traz, inconfundivelmente, uma necessidade mais premente no que concerne ao aprendizado da tecnologia. Cientes deste fato, o *assistente* surge como um possível auxiliar no treinamento dos docentes. Ao vislumbrar as informações necessárias para sua aula de forma metódica e passo-a-passo, é minimizado no docente a sensação de *pânico* tecnológico ao perceber a quantidade imensa de informações necessárias para a realização de uma tarefa que, ao primeiro momento, se mostrava tão simples.

Em relação às *atividades*, estas são caracterizadas como exercícios ou tarefas que os alunos devem realizar dentro de um prazo determinado e que, oportunamente, podem receber algum tipo de avaliação. Para o cadastramento de uma atividade, também foi utilizado o recurso do *assistente*, como comprovam as figuras 52 a 55. O primeiro passo do assistente (figura 52) apresenta o cadastramento básico da *atividade*. O docente deve fornecer um nome para a atividade em questão, a descrição da mesma e escolher uma entre as *possibilidades de resposta do aluno*: somente envio de arquivo, somente envio de texto (neste caso, é aberto um formulário para o aluno preencher sua resposta no próprio ambiente) ou ambos.

Figura 52 - Assistente para criação de uma nova *atividade* – Passo1 – AMEM 2.0

O segundo passo é muito semelhante ao terceiro passo do operacionalizador *aula* (figura 48). Aqui, o docente deverá escolher (se for o caso) o objeto (ou objetos) de aprendizagem que deverão ser utilizados pelo aluno para a realização da atividade (figura 53). Uma atividade poderá conter vários objetos ou nenhum. Como não há *pontos* em uma atividade, todos os objetos estarão anexados diretamente à atividade cadastrada.

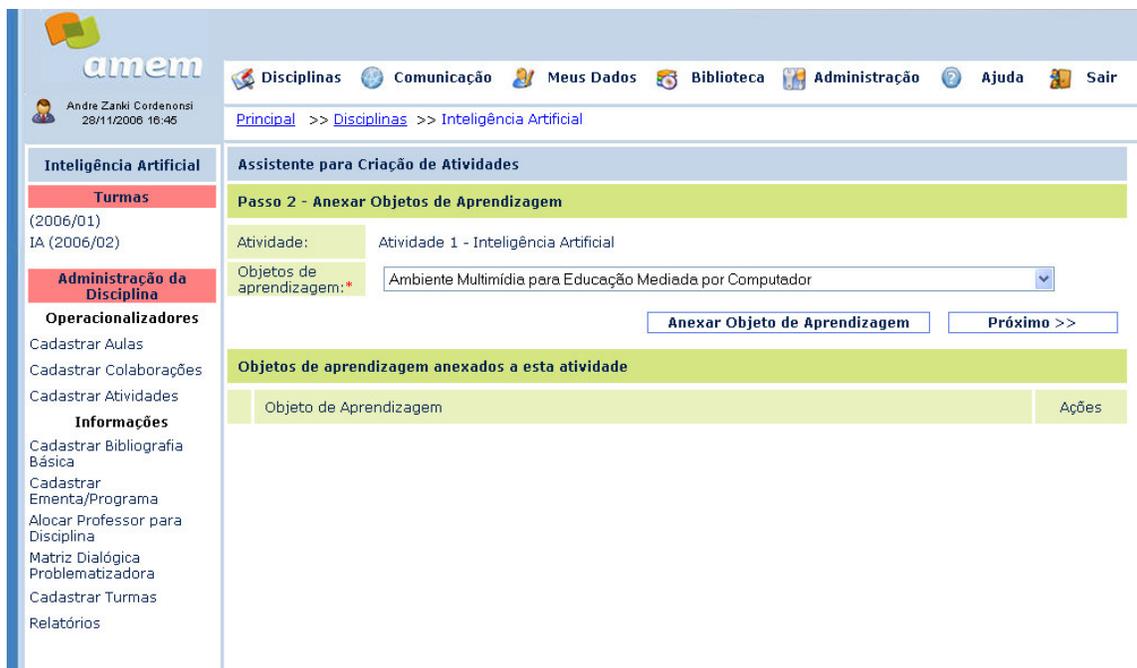


Figura 53 - Assistente para criação de uma nova *atividade* – Passo2 – AMEM 2.0

O terceiro passo, por sua vez, é semelhante ao quarto passo do operacionalizador *aula* (figura 49). O professor deve escolher a bibliografia que ele pressupõe adequada para a realização da atividade (figura 54). Caso não seja necessária nenhuma bibliografia, ele poderá passar diretamente para a finalização do assistente.



Figura 54 - Assistente para criação de uma nova *atividade* – Passo3 – AMEM 2.0

Finalmente, o último passo apresenta somente o resumo da atividade cadastrada com uma mensagem de felicitações (figura 55). O docente poderá, então, finalizar o assistente com segurança.

The screenshot shows the AMEM 2.0 web interface. At the top left is the 'amem' logo and user information for Andre Zanki Cordenonzi. The navigation bar includes links for Disciplinas, Comunicação, Meus Dados, Biblioteca, Administração, Ajuda, and Sair. The breadcrumb trail is 'Principal >> Disciplinas >> Inteligência Artificial'. The sidebar on the left has 'Inteligência Artificial' selected, with sub-links for Turmas, Administração da Disciplina, Operacionalizadores, and Informações. The main content area is titled 'Assistente para Criação de Atividades' and shows the 'Finalização do Assistente' step. A message congratulates the user and provides details about the created activity, including the discipline name, description, response type, and bibliography. A 'Finalizar Assistente' button is visible.

Figura 55 - Assistente para criação de uma nova *atividade* – Finalização – AMEM 2.0

Da mesma forma que o operacionalizador *aula*, a *atividade* também trouxe possibilidades não implementadas na primeira versão, incluindo a anexação de objetos de aprendizagem e a escolha de uma bibliografia selecionada.

O último operacionalizador implementado no ambiente foi o *colaboração*. Diferente dos demais, este não possuía um cadastro complexo ou multifacetado. Baseado nesta constatação, foi implementado um cadastro simples para o usuário, representado na figura 56. O docente deverá fornecer um nome para a *colaboração* que está sendo cadastrada, uma questão orientadora (que guiará o aluno na sua concepção em relação ao que o professor pretende trabalhar na próxima aula/encontro) e, finalmente, a programação da próxima *aula*, ou seja, com qual aula esta *colaboração* deverá ser ligada quando do agendamento das mesmas. Ao preencher os dados, o docente pode, simplesmente, gravar a *colaboração*.



Figura 56 - Criação de uma nova *colaboração* – AMEM 2.0

Após o cadastramento dos operacionalizadores, assim como na primeira versão, é necessário que o docente *agende* os mesmos para que os alunos tenham acesso a programação. Para realizar tal ação, o docente deve estar ingressado em uma turma. A lista de turmas da disciplina em questão aparece em destaque, no canto superior do segundo menu, como demonstra a figura 57, letra b. Ao selecionar uma turma, o menu *migalhas de pão* acrescenta esta informação, como demonstra a figura 57, letra a.



Figura 57 - Agendamento de um operacionalizador – AMEM 2.0

O segundo menu, para uma turma, apresenta diversas funcionalidades. No canto superior, o docente verifica qual a disciplina e turma em que ele está trabalhando (figura 57, letra b). Imediatamente abaixo (letra c), estão disponíveis para o docente todas as opções que os *alunos* daquela disciplina visualizam quando do seu ingresso no sistema. Desta forma, o professor tem acesso a visualização do ambiente tal como ele se apresentaria para o aluno. A terceira parte do segundo menu (letra d) corresponde as ferramentas da administração da turma, disponíveis somente para os docentes e administradores do ambiente. As opções disponíveis são: agendar aulas, atividades ou colaborações (explicados no próximo parágrafo), registrar aulas (onde o docente pode manter um registro das atividades/metodologias desenvolvidas em uma aula/encontro, subsídio este indispensável para a constante melhoria de sua própria atividade docente), avaliar atividades e analisar colaborações (explicados posteriormente), registrar MDP (onde o docente poderá responder as questões da Matriz Dialógica Problematizadora para a aula específica da turma onde ele está trabalhando), alocar um professor para a turma (a lista de professores disponíveis é representada por todos os professores alocados para esta disciplina), verificar matrículas (analisar e responder ao pedido de matrículas dos alunos) e relatórios da turma.

O agendamento de uma aula, atividade ou colaboração é realizado de forma semelhante. O docente deve escolher a aula/atividade/colaboração específica (letra e), informar a data e hora de início e finalização do operacionalizador (letra f) e realizar o cadastramento. É importante salientar dois aspectos:

- para uma determinada colaboração ser agendada, a aula posterior a que este operacionalizador se refere já deve estar agendada para a turma específica;
- a finalização de um operacionalizador (estabelecimento da data/hora finais) não impede o aluno de visualizar os conteúdos, bibliografias e/ou objetos de aprendizagem anexados ao mesmo. Tal informação é utilizada para o gerenciamento dos tempos de estudo e como forma de cobrar eventuais prazos finais para a entrega das atividades e/ou colaborações.

Além do agendamento dos operacionalizadores dentro de uma turma, há outra opção implementada no AMEM 2.0 que não existia na primeira versão. O *agendamento rápido* é realizado com o professor dentro de uma disciplina. Após cadastrar ou modificar uma aula/colaboração/atividade, o professor pode escolher o ícone do agendamento rápido (última ação disponível na longínqua figura 45) e a interface da figura 58 será apresentada. Como o docente ainda se encontra em uma disciplina e não na turma, e considerando também que ele poderá escolher agendar cada um dos operacionalizadores a partir de uma lista, as opções divergem ligeiramente. No *agendamento rápido*, o docente não precisa escolher a aula/colaboração/atividade que deseja agendar, pois a mesma é realizada anteriormente, ao selecionar o ícone da lista representada na figura 45. Outrossim, é necessário ele informar a *turma* para a qual ele deseja realizar o agendamento, informação esta desnecessária na situação anterior. Finalmente, basta ao docente informar os detalhes relativos ao início e finalização do agendamento, terminando sua operação.

Figura 58 - Agendamento rápido de um operacionalizador – AMEM 2.0

Algumas últimas considerações acerca dos operacionalizadores e seus agendamentos precisam ser feitas. A eliminação do encapsulamento *atividade* no AMEM 1.0 trouxe como benefício o alargamento das possibilidades em relação a metodologia pedagógica que o docente deseja empregar em suas atividades docente. No entanto, com a possibilidade de agendamento unitário de cada um dos operacionalizadores, o trabalho necessário para a formalização de um encontro que use todos os operacionalizadores é maior.

#### *Avaliar Atividades e Analisar Colaborações*

Em ambas as implementações do AMEM, o docente tem acesso aos textos/arquivos enviados pelos alunos em suas respostas às colaborações e atividades desenvolvidas pelos professores. No entanto, a primeira versão permitia somente a visualização da resposta do aluno, sem fornecer nenhum tipo de ferramenta que auxiliasse o professor na tarefa de organizar suas avaliações e análises dos textos recebidos. A segunda versão do ambiente fornece duas ferramentas, muito semelhantes entre si, que permitem ao professor responder de forma organizada as tarefas agendadas.

As ferramentas *avaliar atividades* e *analisar colaborações* funcionam praticamente da mesma forma. Depois de ingressar em uma turma, o docente escolhe a ferramenta apropriada. A interface principal do sistema apresentará uma lista com todas as atividades ou colaborações agendadas para aquela turma. Ao escolher uma das atividades/colaborações, é fornecida uma lista com o nome de todos os alunos (figura 59). Para cada aluno, o docente é capaz de verificar a *situação* da atividade/colaboração (*não respondida* ou *já respondida*), baixar o arquivo ou abrir o texto com a resposta da atividade/colaboração (caso o aluno já tenha logrado uma resposta), além de duas ações: editar comentários e remover comentários.



Figura 59 - Avaliação de Atividades – AMEM 2.0

A ação de edição de comentários permite ao docente inserir registros pessoais acerca da atividade/colaboração desenvolvida, notas estas somente disponíveis ao usuário que enviou o texto solicitado. Como é possível observar na figura 60, o docente tem a possibilidade de inserir ou editar diversos comentários acerca do trabalho desenvolvido, sendo que ele também tem acesso aos outros comentários fornecidos por os demais professores da turma (se for o caso). O aluno pode visualizar os comentários dos professores, tendo acesso ao texto, o nome do professor, a data e a hora em que o mesmo foi enviado. Com a utilização destas ferramentas, o docente é capaz de organizar seu trabalho de avaliação/análise das tarefas, tecendo comentários diretamente aos autores.

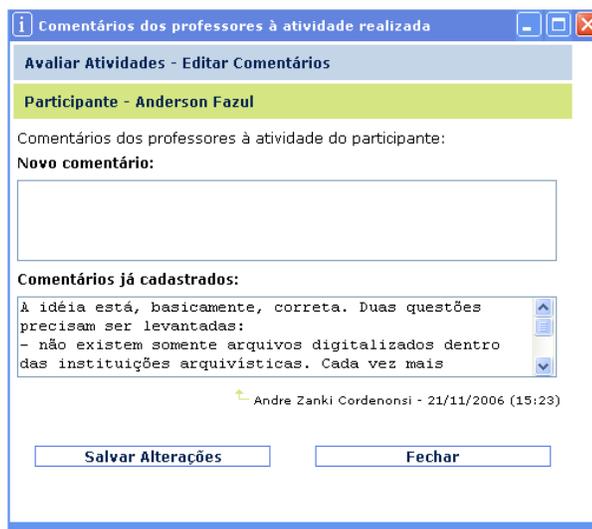


Figura 60 - Edição de comentários às atividades desenvolvidas – AMEM 2.0

### *Diferenciação entre Turmas e Disciplinas*

Outro elemento cognitivo de difícil entendimento por parte dos alunos estava relacionado à diferença entre turma e disciplina. Apesar da grande maioria dos centros de estudo universitário trabalharem fortemente utilizando disciplinas que podem manter uma ou mais turmas dentro de um mesmo semestre (situação bastante comum no ensino superior), para os alunos em geral era difícil perceber a diferenciação entre as duas entidades. Tal fato era reforçado pela própria cultura universitária em geral (instituição, professores e ambiente) que descarregam uma grande quantidade de informações acerca das disciplinas de um determinado curso, mas em quase nenhum lugar é explicado ao aluno que, na verdade, ele efetua sua matrícula em uma *determinada* turma daquela disciplina. Para que o ambiente representasse de forma mais fidedigna o interfaceamento aluno/universidade, a nova implementação do ambiente buscou minimizar a utilização da palavra *turma*, mostrando ao aluno, sempre que possível, informações da *disciplina*, mesmo quando estas se referiam a uma turma específica.

No entanto, para o professor tal distinção é crucial para seu envolvimento com o ambiente mas, neste caso, usualmente o mesmo já possui experiência no desenvolvimento de suas atividades docentes, transitando entre as turmas e disciplinas de forma mais satisfatória.

#### 4.2.2 Modelos navegacionais

As questões levantadas anteriormente e suas respectivas soluções foram desenvolvidas através dos novos modelos navegacionais para o ambiente, utilizando para tanto a técnica da *análise de tarefas* (HOELZEL, 2004) dos usuários, buscando uma melhor navegabilidade do mesmo. Serão apresentados dois modelos construídos, salientando as diferenças entre os dois ambientes. No primeiro caso, apresentado na figura 61, verifica-se a mudança da navegabilidade para um aluno que deseja responder uma atividade no ambiente.

A versão 2.0 do ambiente apresenta uma navegabilidade mais clara devido a exclusão da opção *sair* – que poderia ser traduzida como uma troca de sala de aula, conforme relatado anteriormente. A implementação dos atalhos que permitem a troca de uma disciplina por outra foi desenvolvida através de uma barra de navegação do tipo *migalha de pão* que se encontra na parte superior do sistema. Ela contém a estrutura de navegação do ambiente conforma o usuário se desloca por suas ferramentas, permitindo que o mesmo, ao selecionar uma de suas opções, se transfira automaticamente para níveis superiores na árvore de navegação.

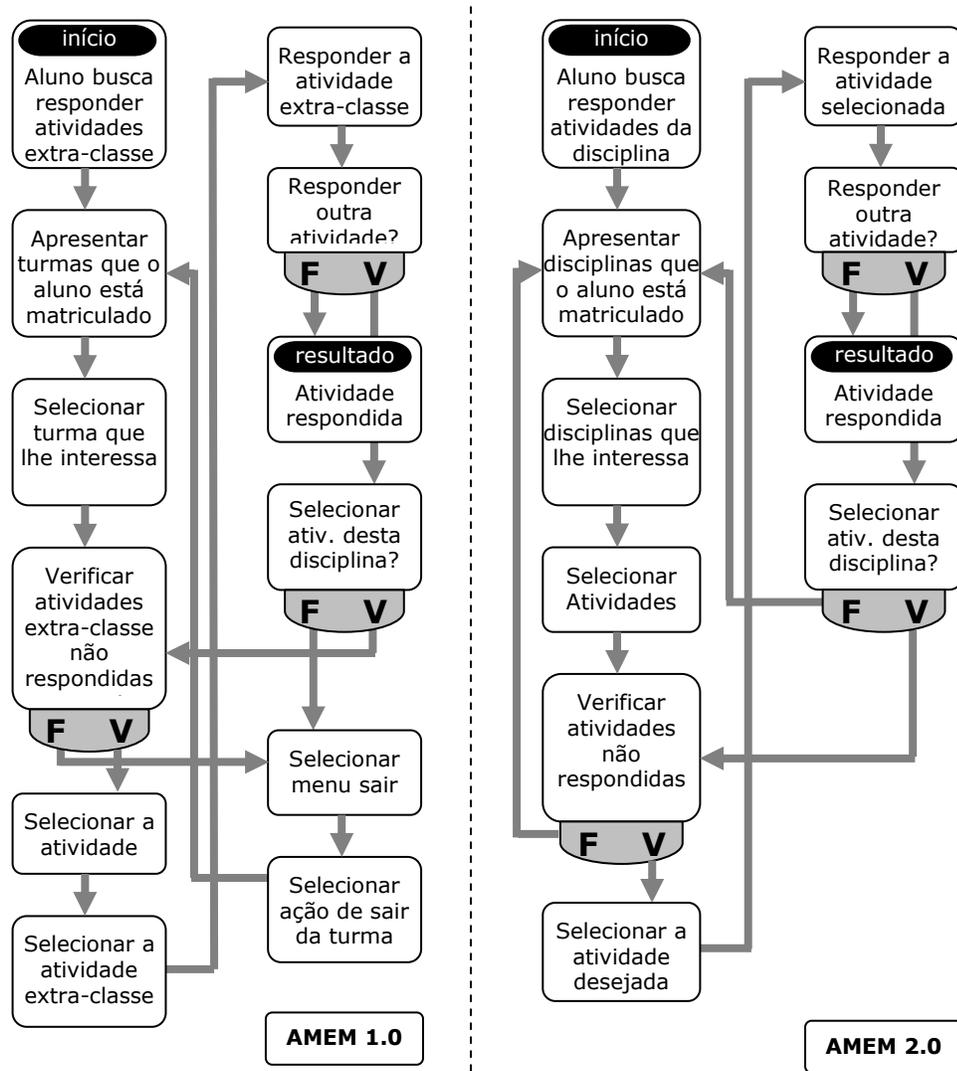


Figura 61 - Fluxo de tarefa Responder Atividades

No segundo caso, representado pelo fluxo da tarefa da figura 62, é apresentando a situação de o professor agendar uma determinada tarefa. Percebe-se, novamente, que a navegabilidade se torna mais fácil e menos onerosa para o professor se ele necessitar agendar diversas tarefas, principalmente em caso de disciplinas e turmas diferenciadas. Os últimos passos do fluxo na versão 1.0, em ambos os casos, foram eliminados pela retirada da opção *sair da turma* e *sair da disciplina* do sistema e a implementação da barra de navegação *migalha de pão*.

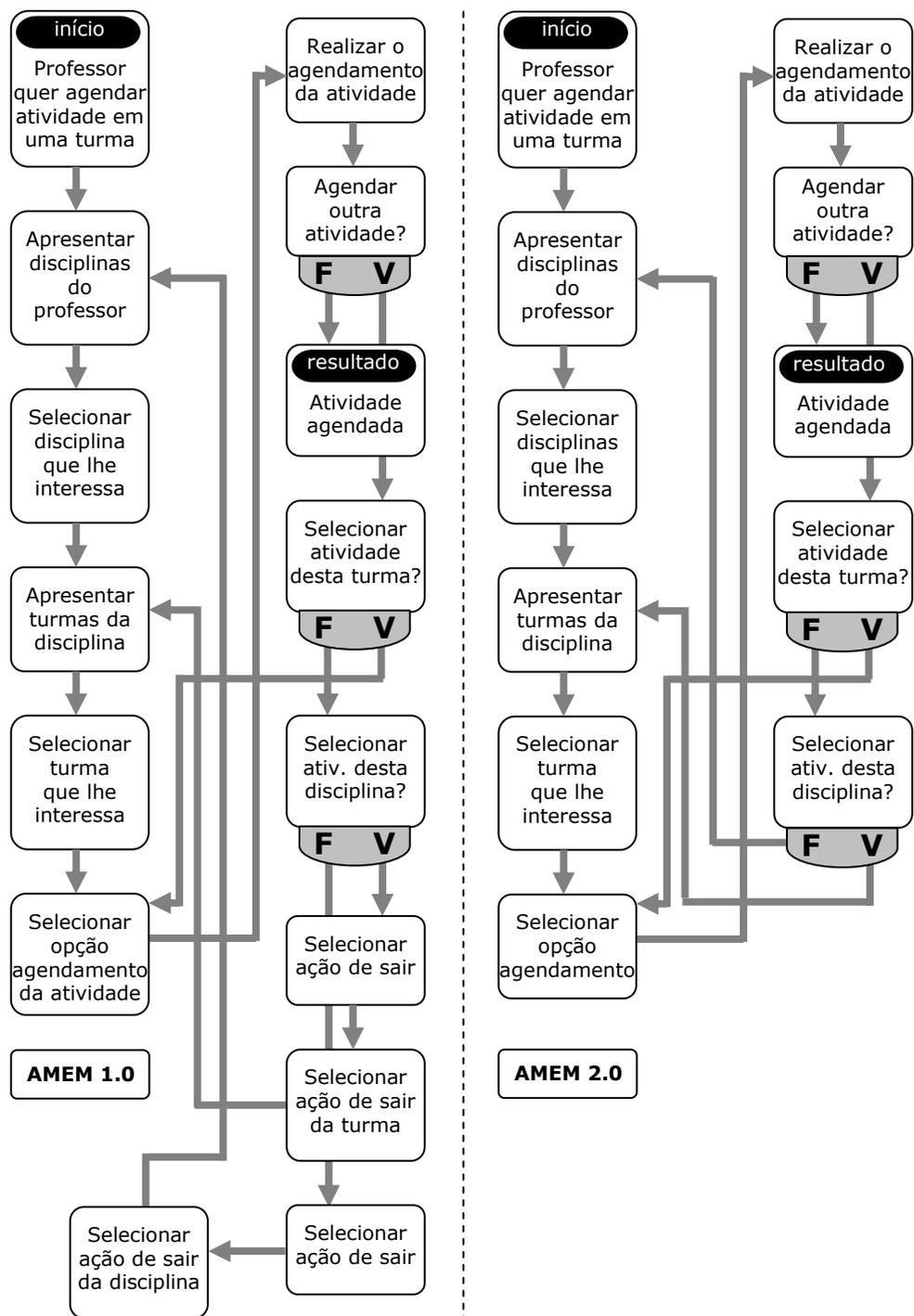


Figura 62 - Fluxo de tarefa Agendar atividade

#### 4.2.3 Aspectos Tecnológicos

Atualmente há centenas de linguagens de programação disponíveis no mercado, incluindo linguagens próprias para o desenvolvimento de aplicações definidas como

*standalone* (softwares que rodam ao serem instalados em uma máquina específica, tais como sistemas operacionais e processadores de texto) e aplicações para *web* (fornecem serviços ou funcionalidades através de uma página da Internet). Dentre esta miríade de possibilidades, a linguagem Java vem se destacando no cenário mundial.

Desenvolvida por um consórcio de empresas liderados pela Sun®, o Java oferece uma gama bastante abrangente de tecnologias, estando disponível para as mais diversas plataformas: computadores pessoais, servidores, *palm-tops*, celulares e *hand-helds*. Há diversas formas de desenvolvimento de aplicações em Java, sendo que as mais importantes são:

- aplicações *standalone*: utilizam o *Java Standard Development Kit* do Java ou o *Java Enterprise Development Kit*. Usualmente são aplicações que rodam em apenas um computador por vez, através da construção de interfaces gráficas ou em modo texto;
- *applets*: aplicações tipicamente para *web*, os *applets* são pequenos programas que podem rodar diretamente sobre uma página HTML<sup>1</sup>. No entanto, diferentemente das páginas tradicionais, os *applets* oferecem funcionalidades que permitem a implementação de páginas interativas, onde o comportamento do *software* pode sofrer modificação através das ações do usuário. Ao carregar uma página que possua uma *applet* embutida, o navegador descarrega o código da aplicação e o executa a partir da máquina local. Problemas de segurança e a dificuldade em implementar programas maiores levaram os desenvolvedores a migrarem as aplicações *web* para as duas novas tecnologias *web* oferecidas pela linguagem: os *servlets* e o *Java Web Start*;
- *servlets*: programas escritos em linguagem *script*, ou seja, que não são compilados previamente. Um programa desenvolvido neste tipo de linguagem é interpretado pela linguagem em tempo de execução. A grande vantagem no desenvolvimento de aplicativos através de *scripts* é a possibilidade de interação imediata com as linguagens de apresentação, tais como o HTML. Em uma mesma página, é possível ter marcadores HTML e comandos *script* interagindo entre si. O Java oferece este serviço através das *Java Servlets Pages* (JSP), páginas que ficam disponíveis em servidores *internet* e podem ser visualizadas como uma página *web* qualquer. Diferentemente das outras linguagens *script*, que realmente fazem um processo de interpretação linha a linha ao serem requisitadas pelo servidor, o JSP compila as páginas para a linguagem intermediária de *bytecodes* (detalhes a seguir) na primeira vez que uma página for requisitada. Quando outros usuários acessarem aquela página, o servidor realiza um teste rápido para verificar se página não foi alterada pelo desenvolvedor. Em caso positivo, a página é novamente recompilada. Caso contrário, a execução ocorre através da aplicação em *bytecodes*, diminuindo o tempo de acesso e aumentando a velocidade do sistema – em comparação com outras linguagem *script*;
- *Java Web Start* (JWS): esta nova tecnologia permite que sejam desenvolvidas aplicações para *web* da mesma forma que as aplicações *standalone*. Os sistemas para a Internet, em sua grande maioria, oferecem uma interface com o usuário

---

<sup>1</sup> HTML – acrônimo para a expressão *Hyper Text Markup Language*, é uma linguagem de marcação utilizada para produzir páginas para *internet* que podem ser interpretados por navegadores.

através da linguagem de marcas HTML, o que diminui as possibilidades de interfaceamento entre o usuário e o *software*. Os *applets*, apesar de oferecem possibilidades mais animadoras, foram desenvolvidos através de uma premissa fechada que atrapalhava a implementação de sistemas mais complexos. Para oferecer uma terceira via, que fornecesse uma gama maior de possibilidades para os desenvolvedores sem a complexidade das *applets*, o Java disponibiliza desde 2001 a tecnologia JWS através do *Java Network Launching Protocol & API Specification*, disponível em <http://java.sun.com/products/javawebstart/download-spec.html>. Aplicações implementadas e compiladas na linguagem Java podem ser disponibilizadas pela internet através do desenvolvimento de um *manifesto de software* e da compactação dos arquivos necessários em um arquivo comprimido no formato *jar*. Este arquivo único é descarregado automaticamente pelo navegador ao acessar um programa baseado nesta tecnologia e o mesmo é executado como se fosse um sistema realmente instalado no computador do usuário. Desta forma, não é necessário que o mesmo rode *dentro* de uma página HTML e qualquer interface implementada disponível para os *softwares standalone* podem ser utilizadas. A vantagem do JWS está no ganho de tempo que é dado aos desenvolvedores, pois uma mesma aplicação pode ser implementada tanto para a *web* como para os computadores pessoais. A desvantagem é que a *Máquina Virtual Java*, uma aplicação presente na grande maioria dos navegadores atuais e que permite rodar programas baseados nesta linguagem, não suporta sistemas mais complexos da tecnologia JWS. É necessário que o usuário tenha instalado em seu computador o JRE (*Java Run-time Environment*) para que as aplicações funcionem.

Outro aspecto muito importante na tecnologia Java está na sua interoperabilidade entre os mais diversos Sistemas Operacionais e plataformas de *hardware*. Usualmente, um programa desenvolvido sobre um determinado sistema operacional e sobre uma arquitetura de computador só pode rodar naquela família específica de computadores e naquele sistema operacional específico. Como exemplo, se um desenvolvedor implementa um sistema na linguagem C, utilizando como plataforma um computador com o processador Intel Pentium IV© e rodando o sistema operacional Microsoft Windows XP©, este *software* não poderá rodar no *mesmo computador*, mas com um sistema operacional diferente, como o Linux Kurumin. Há uma forte dependência entre o binômio sistema operacional e arquitetura de computador. Para compreender este problema, é preciso entender como um programa é desenvolvido. A figura 63 apresenta um esquema simplificado deste processo.

Usualmente, o programador desenvolve sua aplicação em uma *linguagem de programação* em um formato texto. Este *texto* é processado pelo *compilador*, um sistema especial que pode ler, interpretar e transforma os comandos em linguagem pseudo-natural para a linguagem das máquinas (código binário). No entanto, durante este processo, o compilador necessita carregar e processar bibliotecas do sistema operacional e informações provenientes da arquitetura da máquina onde está ocorrendo a compilação. Estas informações são inseridas no produto final da compilação, o código executável que, posteriormente, pode ser rodado na arquitetura de computador específica.

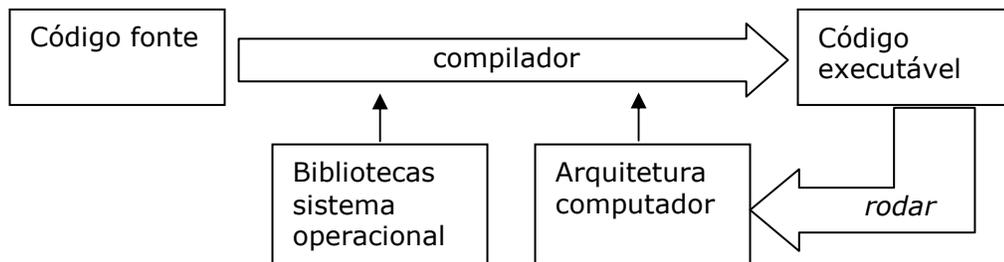


Figura 63 - Processo de compilação de um programa de computador

O processo de compilação dos programas em Java é um pouco diferente, como é possível observar na figura 64. Nesta tecnologia, não existe uma arquitetura de computador embutida no processo. Ela é substituída do que foi convencionalmente denominado *Máquina Virtual Java*, uma espécie de *hardware* virtual que fornece uma camada extra entre o código executável e a arquitetura do computador. Logo, informações específicas de uma arquitetura não precisam ser embutidas dentro do código final. Por outro lado, as bibliotecas Java fornecem o suporte transparente ao sistema operacional, ou seja, novamente não há a necessidade de obter informações específicas sobre qual *software* está controlando a máquina em tempo de compilação.

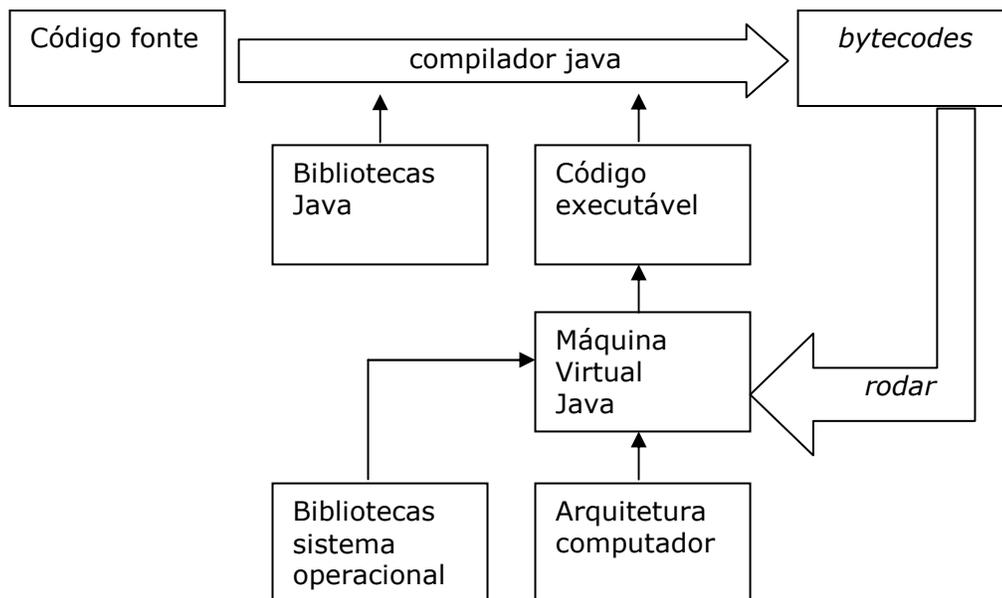


Figura 64 - Processo de compilação de um programa Java

Baseado nesta nova arquitetura, programas em Java não precisam ser reconstruídos ou recompilados para diferentes sistemas operacionais ou arquiteturas de computadores. Uma vez desenvolvido um programa em Java, ele rodará em qualquer binômio *máquina x sistema* que possua uma implementação da linguagem disponível. No entanto, este ganho na interoperabilidade traz como problema adicional a necessidade de interpretar o resultado final da compilação (*bytecodes*) de forma a adicionar, em tempo de execução, as

informações necessárias acerca da arquitetura do computador e do sistema operacional. Este passo a mais torna os programa escritos em Java sensivelmente mais lentos que os programas desenvolvidos em outras plataformas.

A versão 2.0 do AMEM foi desenvolvida com a tecnologia Java dos *scriptlets*, através de páginas JSP. Esta escolha foi realizada levando em conta os seguintes aspectos:

- maturidade da linguagem: a tecnologia Java está a mais de dez anos em desenvolvimento, tempo suficiente para que os problemas originários de sistemas novos tendam a desaparecer. Os aspectos de segurança e confiabilidade estão bem definidos e são bem aceitos pela comunidade em geral. A linguagem fornece subsídios para o desenvolvimento de quaisquer programas de computador, através de extensões e bibliotecas especializadas fornecidas pela própria Sun© ou por fornecedores independentes;
- interoperabilidade: a tecnologia de *script* foi escolhida objetivando alcançar a maior interoperabilidade entre sistemas e arquiteturas disponíveis no mercado. Ao desenvolver páginas *web* que são carregadas dinamicamente ao serem acessadas no servidor, é possível disponibilizar as mesmas para quaisquer sistemas operacionais e *hardware* que implementem um provedor de páginas Java. Atualmente, a grande maioria dos servidores e arquiteturas de computadores já possuem a *máquina virtual Java* em seus sistemas.

Também é importante salientar que a escolha do *servlet* em detrimento à tecnologia JWS pode ser explicada pelo desenvolvimento da aplicação. O AMEM 2.0 possui um grande número de funcionalidades e múltiplas formas de interação com os seus usuários. A versão final do ambiente possui mais de mil e quinhentos arquivos HTML e JSP. Esta enorme complexidade poderia ser implementada em uma aplicação *standalone* e depois fornecida via *web* através da tecnologia *Java Web Start*, mas possivelmente o tamanho do arquivo final, mesmo compactado, tornaria tal opção inviável. Além disso, os usuários não acessam *todas* as funcionalidades do ambiente quando ingressam nele, apesar de estarem aptos a tanto. Desta forma, grande parte do sistema que seria descarregado pela internet e rodado pelo usuário provavelmente não seria utilizado.

Por último, tanto o ambiente AMEM quanto o objeto de aprendizagem LOBO, que serão detalhados no próximo capítulo, foram desenvolvidos e são disponibilizados como sistemas *open source*, ou seja, de livre acesso ao código fonte, permitindo também a redistribuição dos mesmos, a inclusão do código fonte junto ao sistema, a permissão para modificações e trabalhos derivados, sendo que estes também devem ser distribuídas sob os mesmo termos, a não discriminação da distribuição em relação a quaisquer tipos ou agrupamentos e a distribuição conjunta com a licença (<http://www.inf.ufsm.br/~andrezc>). Estes aspectos representam as principais características da *Open Source Initiative* (OSI, 2006). A distribuição da produção tecnológica se encontra em mídia digital, no anexo A.

### 4.3 Conclusões do Capítulo

Como relatado anteriormente, alguns aspectos importantes foram considerados nestes últimos cinco anos de utilização do sistema que levaram a equipe multidisciplinar a uma nova rodada de discussão para a geração da segunda versão do ambiente onde as seguintes necessidades foram levantadas e desenvolvidas: (a) a necessidade de permitir a

inserção de novos operacionalizadores pedagógicos, incluindo também formas de utilização do ambiente que não estejam estritamente baseadas na concepção pedagógica original: a criação e manutenção de novos operacionalizadores pedagógicos através da implementação de objetos educacionais manipuláveis se torna um objetivo interessante por si mesmo e fornece subsídios e ferramental importante para os diversos aspectos das ações educacionais dos professores. Ao mesmo tempo propicia aos professores, cujo embasamento teórico-metodológico diverge do escolhido para o sistema, uma gama maior de práticas educacionais que não são completamente atreladas ao sistema original; (b) a necessidade da readequação da interface do ambiente: algumas questões relacionadas a interface, no que concerne a escolha de cores e a forma de manipulação dos usuários através dos perfis foi rediscutida. A maior liberdade dos ambientes atuais permite uma melhor harmonização no que concerte a parte estética, sem deixar de lado a necessidade de manter o mesmo visual consistente em todos os navegadores e sistemas, ao mesmo tempo em que a manipulação dos perfis dentro da metodologia do sistema foi alterada. Na concepção atual, o usuário pode trocar de *aula* – ou sala de aula, ou disciplina – sem necessitar sair explicitamente da mesma. O ato de *entrar* em uma disciplina automaticamente o exclui da anterior, poupando trabalho do usuário e tornando a interface mais clara e transparente para alunos e professores.

Apesar de todas as considerações a respeito dos AVEA e de sua importância na condução de disciplinas presenciais, semi-presenciais ou à distância, a utilização de um arcabouço tecnológico não é o único elemento necessário para que os objetivos educacionais de um determinado curso/disciplina sejam atingidos. A tecnologia e o ferramental disponível ao professor constituem um conjunto de saberes e instrumentos que devem ser utilizados com parcimônia e sabedoria pelo agente docente. Como foi definido anteriormente, os ambientes virtuais constituem um espaço para o *ensino* e a aprendizagem, constituindo eles próprios *objetos* deste processo. Deste modo, o papel do professor continua sendo de fundamental importância no processo do desenvolvimento do aluno. Contudo, suas ações são diferenciadas, pois o enfoque tradicional centradas na simples *informação* ou *transmissão de conteúdo* se transformam em práticas colaborativas, onde o educando constrói seu conhecimento a partir de suas ações. As práticas bancárias (FREIRE, 1981 *apud* DE BASTOS *et al.* 2005), definidas como situações onde a escola se torna um ato de depositar, os alunos são os depósitos e os professores os depositantes, devem ser descartadas pelas práticas colaborativas, onde professores e alunos comunicam-se e a responsabilidade pela aprendizagem parte tanto do educador como do educando.

*Portanto, a mediação pedagógica (cuidadosa e competente) tem um papel de destaque, não somente no sentido de procurar ampliar as interações (mantendo a existência do ambiente), como também, fazer intervenções para garantir conexões de qualidade (desconstrução/ construção/ reconstrução do conhecimento).*

(SANTOS, OKADA, 2003)

Como constata (SANTOS, OKADA, 2003), um AVEA, mesmo contendo uma forte base pedagógica em sua definição formal, não pode, por si mesmo, garantir práticas pedagógicas ou eficiência no desenvolvimento dos alunos. Além da estrutura, é necessário um *conteúdo* adequado ao processo ensino-aprendizagem requerido e uma *estratégia* que

reflita as escolhas epistemológicas definidas pelos docentes. A primeira parte desta questão será desenvolvida no próximo capítulo.

## 5 Objetos de Aprendizagem

Neste capítulo, serão caracterizados tecnologicamente os objetos de aprendizagem digitais.

### 5.1 O que são objetos de aprendizagem?

A utilização da tecnologia de objetos está se tornando cada vez mais difundida entre os envolvidos da área de informática. A Orientação a Objetos, surgida na década de 60, extrapolou os limites apenas da programação para invadir as áreas de análise e projeto de sistemas, modelagem de banco de dados, entre outras. A modelagem e implementação de sistemas baseados em objetos torna o desenvolvimento de sistemas o mais próximo da realidade observada. Para (SANTANCHÉ, 2002) várias são as vantagens do uso de objetos na construção de *softwares*, entre elas:

- objetos podem ser construídos de forma independente, podendo ser reaproveitados futuramente, evitando que novos projetos reconstruam partes de sistemas já implementados anteriormente;
- podem ser combinados com outros objetos, não limitando seu uso a um programa específico;
- são facilmente configuráveis e podem ser personalizados de acordo com determinada necessidade;
- a estrutura de construção dos objetos apresenta uma eficiente divisão entre sua lógica interna e sua interface de comunicação com o meio exterior, tornando mais fácil sua utilização.

Assim, a utilização de objetos na elaboração e desenvolvimento de projetos educacionais pode apresentar uma ampla gama de possibilidades, permitindo o intercâmbio entre os diversos conteúdos desenvolvidos, bem como o reaproveitamento na construção de novos projetos. A Orientação a Objetos vem possibilitar a utilização de padrões no desenvolvimento de projetos, de forma a permitir uma adequada comunicação entre objetos de diferentes lugares, desenvolvidos em um contexto de diferentes configurações.

Adotando esta filosofia surgem os chamados “Objetos de Aprendizagem” (*Learning Objects*), sendo utilizados de uma maneira até mesmo mais ampla do que a estabelecida pelo paradigma da orientação a objetos. Para (IP, MORRISON, 2001), “Objetos de Aprendizagem incluem mecanismos de *software* relacionados ao tratamento dispensado para o conteúdo, apoio as interfaces de usuário para a interatividade e capacidade de acesso aos mesmos”. O IEEE *Learning Technology Standards Committee (LTSC)* define Objetos de Aprendizagem como: “...qualquer entidade, digital ou não digital, que pode ser utilizada, reutilizada ou referenciada durante o aprendizado apoiado pela tecnologia. O aprendizado apoiado pela tecnologia inclui sistemas de treinamento baseados em computador, sistemas de aprendizado a distância, sistemas inteligentes de instrução, entre outros.” (IEEE, 2002). Já (TAROUÇO, FABRE e TAMUSIANAS, 2003) define um objeto de aprendizagem como a idéia de recursos suplementares ao processo de aprendizagem, que possam ser reutilizados para apoiar a aprendizagem.

Muitos autores têm usado a metáfora dos blocos de construção do jogo LEGO para descrever Objetos de Aprendizagem. A metáfora LEGO expressa a noção de “pequenas peças de instrução (blocos LEGO) que podem ser montadas (agrupadas) em uma grande estrutura instrucional.” Entretanto, uma nova corrente tem sugerido a transformação da metáfora do LEGO para a metáfora do átomo (WILEY, 2001). Para estes autores, a definição de objetos como uma peça que pode ser combinada com *qualquer* outra peça, da maneira que o usuário escolher e de forma tão simples que até mesmo uma criança é capaz de realizar, é simplista. Educacionalmente, nem todos os objetos podem ser combinados com outros objetos. Ainda, uma simples coleção de objetos não monta um curso, independente de sua natureza, e muito menos por pessoas não especializadas. A metáfora do átomo indica que:

- nem todo o objeto é combinável com quaisquer outros objetos (assim como os átomos não se combinam com *todos* os outros átomos existentes);
- objetos só podem ser combinados em determinadas estruturas que são definidas por sua própria estrutura interna (da mesma forma, átomos só se ligam em moléculas de maneiras muito bem definidas);
- é necessário treinamento para realizar a agregação de objetos (assim como para conduzir experiências químicas ou físicas que construam ou combinem moléculas).

Para (DOWNES, 2000), a utilização de Objetos de Aprendizagem percorre um caminho onde os *softwares* educacionais eram desenvolvidos de forma customizada para cada situação necessária, passando para uma produção em massa (acesso a todos) e, finalmente, chegando na customização em massa dos componentes produzidos e distribuídos. Em (WILEY, 2001) é proposta uma taxonomia para a classificação de objetos de aprendizagem, considerando cinco grandes grupos que são apresentados e exemplificados em relação a suas diferenças e similaridades. Para o autor, um objeto de aprendizagem pode ser:

- *fundamental*: um recurso individual e digital que não pode ser dividido. Usualmente, se apresenta como uma peça visual ou sonora que é utilizada para exibir um conceito ou exemplificar uma função;
- *combined-closed*: um pequeno número de recursos digitais combinados pelo criador do objeto em tempo de projeto. Cada recurso individual não é acessível para reutilização, somente o objeto inteiro. O exemplo mais característico deste tipo de objeto é o vídeo, cujas imagens e trilha sonora se tornam suas partes constituintes mas, usualmente, não podem ser acessadas individualmente. Os objetos *combined-closed* podem conter algum tipo de lógica limitada (realização de testes de múltipla escolha com auto-correção) mas não devem conter lógicas internas complexas. Usualmente, estes objetos têm um propósito único podendo prover, simultaneamente, conteúdo e práticas;
- *combined-open*: um grande número de recursos digitais combinados pelo computador em tempo real quando é realizada a requisição pelo objeto. Os objetos constituintes são diretamente acessíveis para reutilização. O exemplo mais característico é uma página *web*, cujos componentes (imagens, vídeos, textos, sons e/ou outras mídias) usualmente existem em formatos reutilizáveis e são combinados em um só objeto, a página em si, quando a mesma é requisitada. Usualmente,

objetos *combined-open* relacionam objetos *combined-closed* e/ou *fundamentals* para formar uma unidade instrucional completa;

- *generative-presentation*: define a lógica e estrutura para combinar ou gerar objetos de aprendizagem de níveis inferiores (*fundamental* e *combined-closed*). São utilizadas para criar apresentações para instrução, prática e teste de conteúdos. Usualmente, estes objetos têm a habilidade de passar mensagens para outros objetos tipicamente em situações de prática ou teste. Apesar dos *generative-presentation* terem um alto grau de reuso em contextos similares, a possibilidade de utilização em outros domínios não é acentuada;
- *generative-instructional*: define a lógica e estrutura para combinar objetos de aprendizagem do tipo *fundamental*, *combined-closed* e *generative-presentation* com o propósito de instrução e avaliação de estudantes, suportando a instanciamento de estratégias instrucionais abstratas. Objetos deste tipo possuem um alto grau de reusabilidade em contextos similares ou diferenciados.

Muitos projetos, nacionais e internacionais, têm sido estabelecidos para a disseminação dos objetos de aprendizagem, representando um verdadeiro esforço coletivo para o desenvolvimento de novos objetos com o intuito de formar uma biblioteca instrucional que possa ser utilizada nos mais diversos processos de ensino-aprendizagem (educação infantil, superior, treinamento). Entre as iniciativas mais conhecidas, pode-se destacar:

- RIVED: lançado pelo Ministério da Educação do Brasil no primeiro semestre de 2000 (NASCIMENTO, 2005), a Rede Internacional Virtual de Educação – RIVED (<http://rived.proinfo.mec.gov.br>) tinha como objetivos a melhoria no processo ensino/aprendizagem nas escolas de Ensino Médio. O projeto era desenvolvido colaborativamente entre o Brasil, Venezuela, Colômbia, Peru e Argentina mas, atualmente, não há mais atividades com os demais países. Envolvendo especialistas conteudistas nas áreas de Química, Biologia, Física e Matemática (Ciências da Natureza, Matemática e suas Tecnologias), uma *designer* instrucional, um pedagogo especialista em informática, um *web designer*, um programador e um ilustrador, a equipe piloto do projeto desenvolveu procedimentos operacionais, identificando estruturas e habilidades apropriadas para a aceleração da produção do material. Após esta fase inicial, o projeto Fábrica Virtual foi iniciado em 2004, transferindo para as instituições de ensino superior o processo de desenvolvimento e produção de recursos educacionais digitais na forma de objetos de aprendizagem. Em novembro de 2006, dezessete instituições nacionais<sup>1</sup> participavam do projeto, desenvolvendo objetos nas áreas de ciências, biologia, física, matemática, química, história, artes e geografia;

---

<sup>1</sup> Universidade Federal de Santa Maria (UFSM), Universidade Federal do Rio Grande do Sul (UFRGS), Universidade Federal do Rio de Janeiro (UFRJ), Universidade Estadual do Norte Fluminense (UENF), Universidade de São Paulo (USP), Universidade Estadual Paulista (UNESP), Centro Universitário Franciscano (UNIFRA), Universidade Federal de Uberlândia (UFU), Universidade Federal do Ceará (UFC), Universidade Federal da Paraíba (UFPB), Universidade Federal do Maranhão (UFMA), Universidade Estadual do Maranhão (UEMA), Universidade Federal de Alagoas (UFAL), Universidade de Brasília (UnB), Centro Federal de Educação Tecnológica/Goiás (CEFET/GO), Universidade Regional do Noroeste do Estado do Rio Grande do Sul (UNIJUI), Universidade Federal de Ouro Preto (UFOP)

- CESTA: o projeto CESTA (Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem) foi desenvolvido na Universidade Federal do Rio Grande do Sul, com vistas a “sistematizar e o organizar o registro dos objetos educacionais que vinham sendo desenvolvidos pela equipe para cursos de capacitação em Gerências de Redes, Videoconferência e no Pós-Graduação Lato Sensu Informática na Educação” (TAROUCO, FABRE e TAMUSIANAS, 2003). A partir dos cursos citados acima, organizados na modalidade à distância e formatados com intuito de utilizar objetos de aprendizagem, o projeto passou a contar com uma base catalogada que poderia ser pesquisada a qualquer momento. A especificação dos metadados utilizados foi elaborada de acordo com a norma IEEE 1484.12.1 (*Learning Objects Metadata*). Os objetos atualmente cadastrados no projeto CESTA (<http://www.cinted.ufrgs.br/CESTA>) abrangem uma grande gama de conhecimentos, trespassando áreas como história, física, ciência da computação, educação, saúde, tecnologia e ecologia;
- MERLOT: o projeto MERLOT (*Multimedia Educational Resource for Learning and Online Teaching*) é um repositório baseado na *web*, livre e de desenvolvimento cooperativo onde é possível buscar e catalogar objetos educacionais, além de guias para seu uso. O Merlot (<http://www.merlot.org>) também permite que outros usuários avaliem o objeto desenvolvido, atribuindo notas e comentários que podem ser visualizados pelos demais. É importante salientar que uma das estratégias principais do projeto MERLOT é a revisão dos materiais instrucionais *on-line*, com vistas a auxiliar o processo de manutenção de uma coleção que contribua com objetos de significado teórico, corretos contextualmente, pedagogicamente orientados e de fácil utilização (MALLOY, HANLEY, 2001). A revisão pode ser efetuada por qualquer membro do projeto MERLOT. Um membro é definido como qualquer pessoa que realiza o cadastro no projeto, estabelecendo suas áreas de atuação. Todos os membros podem avaliar os objetos de suas áreas de atuação, contribuindo em dois aspectos sobre a referida *qualidade* do objeto: (a) os modelos e conceitos são apresentados *corretamente*?; (b) os objetos apresentam conceitos, modelos e/ou exemplos significativos para a área?;
- CAREO: o *Campus Alberta Repository of Educational Objects* (Careo) é um projeto da Universidade de Calgary. Ele possui uma lista de objetos novos, apresentada em sua interface principal (<http://careo.ucalgary.ca>) além de permitir que usuários façam comentários acerca do material instrucional disponibilizado ali, através de textos no padrão *Wiki*. Para ter acesso aos objetos, é necessário um cadastro no sistema, cuja inscrição permanece gratuita (VERHAART, 2004);
- *Wisconsin Online Resource Center*: o projeto foi desenvolvido inicialmente pelos membros da *Wisconsin Technical College System*(WTCS), que produziam objetos utilizando técnicas de multimídia. Os professores da WTCS passaram a produzir objetos, que foram difundidos pela página do projeto(<http://www.wisc-online.com>). Dois aspectos importantes tornam este repositório diferenciado dos demais:(a) os objetos reportados no repositório podem ser utilizados gratuitamente, através de um termo de permissão; (b) existe uma definição de *tempo* para os objetos cadastrados no repositório. A definição de um tempo pré-determinado e arbitrário para que um estudante utilize ou trabalhe com um objeto de aprendizagem é um conceito bastante discutível. (POLSANI, 2003) afirma que o tempo necessário para explorar

um objeto varia de aprendiz para aprendiz, dependendo fortemente das características individuais do estudante. No entanto, os membros da *Wisconsin Online Resource Center* reafirmam a necessidade de um *tempo* arbitrário para o estudante na sua própria concepção de objeto de aprendizagem. Em sua página principal, a definição para um objeto de aprendizagem é formalizada como “uma nova maneira de pensar sobre conteúdos de aprendizagem. Tradicionalmente, os conteúdos são trabalhos em pedaços divididos por períodos de uma hora. Objetos de aprendizagem são unidades de aprendizagem muito menores, tipicamente variando de dois a quinze minutos”.

## 5.2 Catalogação e busca de objetos de aprendizagem: propostas e iniciativas

A popularização dos objetos de aprendizagem trouxe como preocupação investigativa formas de padronizar a especificação, construção e identificação dos objetos. Como em todas as áreas da computação, a adoção de modelos e padrões para o desenvolvimento de *software* se torna importante para a rápida prototipação e implementação dos mesmos. Aliada as modernas técnicas de desenvolvimento de *software*, é necessário também permitir a interoperabilidade dos objetos nos mais diversos tipos de AVEA – Ambientes Virtuais de Ensino e Aprendizagem. Ambientes desenvolvidos com padrões fechados ou que não possam *importar* diferentes tipos de objetos traçam, seguramente, seus caminhos para a obsolescência. O próprio esforço de desenvolverem novos e atrativos ambientes virtuais pode tornar os próprios sistemas inadequados em tempo inferior aos objetos de aprendizagem.

Desta forma, é necessário estabelecer formas de transferir objetos de aprendizagem de um AVEA para o outro sem perda de sua funcionalidade, até mesmo porque não há escolhas *certas* em termos de um ambiente virtual de ensino/aprendizagem, e sim, escolhas mais ou menos adequadas ao próprio *locus* de trabalho dos docentes e alunos, estratégias pedagógicas, financeiras e outros aspectos de caráter puramente tecnológico, como parque computacional (*hardware* e *software*) instalado na escola/instituição.

O principal objetivo das especificações é manter a reusabilidade, a portabilidade e interoperabilidade dos objetos de aprendizagem. Para alcançar estes objetivos, é necessário que os objetos construídos: (a) possam ser utilizados nas mais diversas plataformas (*portabilidade*); (b) sejam desenvolvidos com um objetivo claro e bem definido, possibilitando uma busca qualificada por este recurso (*reusabilidade*); e (c) possam *conversar* com outros objetos de aprendizagem, permitindo que diversos objetos sejam utilizados conjuntamente (*interoperabilidade*). Atualmente, é possível reconhecer dois esforços internacionais para o desenvolvimento de padrões possíveis para conjuntos de objetos de aprendizagem:

- definição de metadados: em sua concepção mais comum, metadados representam *dados sobre dados*. Um metadado é uma abstração do dado, usualmente contendo as informações necessárias para a administração dos recursos definidos pelos dados;
- desenvolvimento de objetos e conteúdos portáveis: também conhecidos como *empacotadores de objetos*, estes padrões tem por objetivo fornecer descrições mais detalhadas dos que os metadados, possibilitando a inserção de diversos objetos em um único curso.

Nas próximas seções, as principais iniciativas internacionais para cada um dos aspectos mencionados acima serão descritas em maiores detalhes.

### 5.2.1 Padrões de metadados

Há dois padrões de metadados que merecem destaque no que concerne à sua utilização e desenvolvimento.

#### *Dublin Core*

O *Dublin Core Metadata Initiative* (DC) (<http://dublincore.org>) se caracteriza por um fórum aberto com o intuito de desenvolver padrões de metadados que mantenham as características da interoperabilidade e do acesso eletrônico, suportando uma grande variedade de propósitos e modelos de negócios. Apesar de não ter sido criado especificamente para fins educacionais, o DC pode ser utilizado em objetos de aprendizagem e elementos bibliográficos tradicionais, tais como artigos e livros, o que representa uma maior representatividade em termos de generalidade. O padrão possui dezoito (18) elementos (SOUZA *et al.*, 2000):

- título: nome dado para o recurso;
- autor: o principal responsável pela criação intelectual do recurso. Este campo pode representar um ou mais autores físicos ou organizações;
- palavras-chave: com a utilização de um vocabulário controlado (possível, mas não estritamente necessário), deve prover vocábulos ou uma frase que expresse o conteúdo/assunto do recurso;
- categoria: fornece a expressão do conteúdo através de um sistema formal de classificação. É encorajada a utilização de sistemas internacionais;
- descrição: texto livre descrevendo o conteúdo do recurso;
- publicador: entidade responsável por tornar o recurso disponível;
- colaborador: pessoas ou organizações não citadas no campo *autor* mas que tenham fornecido colaborações intelectuais relevantes e significativas para o desenvolvimento do recurso;
- data: data da criação ou disponibilização do recurso. É recomendável a utilização da norma ISO 8601 (ISO, 1997);
- tipo: categoria do recurso, tais como texto, som, imagem ou vídeo. O tipo deve ser selecionado da lista de tipos desenvolvida pela DC;
- formato: formato dado ao recurso, possivelmente indicando *software* ou *hardware* necessários à exibição/operação do recurso. O formato também deve ser selecionado da lista de formatos desenvolvida pela DC;

- acesso: identificador do recurso. A URL<sup>1</sup> e a ISBN<sup>2</sup> são normas de identificação que podem ser utilizadas de acordo com o *tipo* do recurso;
- identificador do recurso: em forma numérica ou alfanumérica, de tal forma a identificar inequivocamente o recurso;
- fonte: informações sobre outros recursos que tenham sido utilizados de forma significativa na construção deste. Pode ser textual ou na forma de identificação de um recurso;
- idioma: idioma do conteúdo do curso;
- relação: permite o estabelecimento de relações entre diversos recursos. Esta relação pode assumir várias formas, usualmente através de vocábulos pré-definidos como *baseado em*, *parte de* ou *versão de*.
- cobertura: características espaciais ou temporais do recurso. A cobertura espacial se refere a uma região física específica (cidade, estado ou qualquer outro elemento geopolítico). Ela pode ser definida através de coordenadas ou utilizar uma lista controlada de localizações. A cobertura temporal se refere a que período o recurso tem relação (se existir). Ela deve utilizar o mesmo padrão ISO 8601 que é fornecido para a data de criação;
- direito autoral: declaração de direitos sobre a propriedade intelectual do recurso. Pode ser textual ou um identificador que relacione ao recurso que contenha a declaração supra-citada;
- contato: indicação para contato através de nome, endereço eletrônico ou postal da pessoa física ou instituição responsável pelo recurso.

#### *IEEE 1484.12.1 – Learning Objects Metadata*

O *Learning Objects Metadata* (LOM) desenvolvido pelo *Learning Technology Standards Committee* da IEEE (<http://ltsc.ieee.org/wg12>) estabelece uma especificação com atributos para descrever especificamente objetos educacionais (TAROUCO, FABRE, TAMUSIANAS, 2003). Para este padrão, um objeto de aprendizagem é definido como qualquer entidade digital ou não digital que pode ser utilizada em ambientes de ensino/aprendizagem e treinamento (DUTRA, 2003).

Uma característica importante do LOM é que este não define nenhum sistema tecnológico prioritário para a implementação do padrão. Apesar do desenvolvimento dos metadados utilizarem, usualmente, a linguagem XML (*eXtended Markup Language*) (como pode ser observado em HANSEN e PINTO, 2003, SANTANCHÉ, 2002, RAMALHO *et al.*, 2004 e DUTRA, TAROUCO, KONRATH., 2005a), esta não é obrigatória pelo padrão LOM.

As propriedades do objeto educacional são especificadas através de esquemas. Um esquema é uma estrutura hierárquica que determina as propriedades essenciais associadas a

<sup>1</sup> URL – Acrônimo para *Uniform Resource Locator*. Especifica o endereço de um recurso disponível em uma rede de computadores, estabelecendo o protocolo de comunicação, a máquina, o caminho e o recurso.

<sup>2</sup> ISBN – Acrônimo para *International Standard Book Number*, estabelece um código internacional e único para um determinado livro e publicações não periódicas.

um objeto educacional, especificando o tipo, domínio e obrigatoriedade. O esquema básico do LOM (*BaseScheme*) define nove grupos de propriedades, que são apresentadas a seguir. No entanto, é importante observar que não há a necessidade de se utilizar os nove grupos para que um objeto de aprendizagem esteja *em conformidade* ao esquema LOM. Extensões ou a utilização de partes do *BaseScheme* são permitidas pela especificação, como pode ser observado em (TAROUCO, FABRE e TAMUSIANAS, 2003).

As nove categorias/grupos do *BaseScheme* do LOM são (IEEE, 2002):

- geral: agrupa as informações genéricas que descrevem o objeto de aprendizagem como um todo. Ela contém os identificadores:
  - identificador: identificação *global* única do objeto;
  - título: nome dado ao objeto;
  - catálogo: nome ou designação do esquema de catalogação utilizado para identificar formalmente o objeto;
  - entrada: valor do identificador ligado ao esquema de catalogação;
  - idioma: idioma utilizado no objeto;
  - descrição: descrição textual do objeto;
  - palavras-chave: vocábulos ou frases que descrevam os tópicos de interesse do objeto;
  - cobertura: temporal, cultural, geográfica ou regional;
  - estrutura: organização estrutural do objeto. Pode ser: (a) *atomic*: objeto indivisível; (b) *collection*: o objeto é formado por uma coleção de objetos sem uma relação específica entre eles; (c) *networked*: o objeto é formado por uma coleção de objetos com uma relação específica mas não informada; (d) *hierarchical*: o objeto é formado por uma coleção de objetos cujas relações podem ser representadas por uma estrutura em árvore; e (e) *linear*: o objeto é formado por uma coleção de objetos ordenados.
  - nível de agregação: representa a função granular do objeto. Pode ser de nível: (a) um, o menor nível, representando normalmente um objeto único dentro de um contexto; (b) dois, representando uma coleção de objetos do nível um, como uma lição ou capítulo; (c) três, representando uma coleção de objetos do nível dois, como uma disciplina; (d) quatro, o maior nível, representa um conjunto de agregados do nível três, por exemplo, as disciplinas de um curso completo.
- ciclo de vida: esta categoria descreve a história e o estado corrente do objeto e das entidades que atuaram diretamente no desenvolvimento do recurso. Este grupo contém as propriedades:
  - versão: a edição atual do objeto de aprendizagem;
  - estado: o estado atual do objeto, podendo possuir os valores de *draft* (rascunho), *final* (versão final), *revised* (finalizado e revisado) ou *unavailable* (ainda não disponível);
  - contribuições: informa as entidades (pessoas ou organizações) que contribuíram com o estado atual do objeto de aprendizagem durante seu ciclo de vida (criação, edição ou publicação). É dividida em:
    - papel: tipos de contribuição para cada entidade. Deve possuir, no mínimo, o papel *author* (autor do objeto). Além deste, os papéis possíveis são *publisher* (publicador), *unknown* (desconhecido),

- initiator* (entidade que iniciou ou fomentou o início do objeto), *terminator* (entidade responsável pela eliminação do objeto, ou seja, não disponibilizá-lo mais), *validator* (validação do objeto como um todo), *editor* (editor), *graphical designer* (designer gráfico), *technical implementer* (desenvolvedores técnicos), *content provider* (provedor dos conteúdos), *technical validator* (validação dos aspectos técnicos), *educational validator* (validador dos aspectos pedagógicos), *script writer* (desenvolvedor das seqüências do objeto), *instructional designer* (designer instrucional) ou *subject matter expert* (especialista na área do conteúdo do objeto);
- entidade: entidade a qual o papel é designado;
  - data: data da contribuição.
- meta metadata: esta categoria descreve os próprios registros dos metadados utilizados. As propriedades deste grupo são:
- identificador: identificador global único do metadado;
  - catálogo: nome ou designação da identificação do sistema de catálogo utilizado para a entrada;
  - entrada: valor da identificação utilizando o sistema de catálogo representando anteriormente;
  - contribuição: assim como anterior, representa os contribuidores para a definição dos metadados utilizados no objeto. É dividido em: (a) papel, possuindo apenas dois tipos de atributos (*creator* e *validator*); e (b) entidade, que utiliza como identificador o padrão *vCard* (RFC2426, 1998);
  - esquema: nome e versão das especificações utilizadas para a criação desta instância de metadados;
  - idioma: linguagem utilizada para a definição dos metadados.
- técnica: esta categoria descreve os elementos técnicos e os requerimentos para utilização do recurso. Os atributos possíveis são:
- formato: tipos de dados técnicos de todos os componentes do objeto de aprendizagem. Estes elementos devem ser utilizados para identificar o *software* necessário para acessar o recurso. A identificação deve utilizar o padrão *MIME types* (RFC2048, 1996);
  - tamanho: tamanho do objeto em *bytes*;
  - localização: *string* utilizada para acessar o objeto. Pode ser tanto uma localização virtual (URL) ou um método para achar a localização (URI<sup>1</sup>);
  - requerimentos: requerimentos técnicos necessários para a utilização do objeto de aprendizagem. Os requerimentos utilizam o conector lógico *and*, ou seja, o cadastramento de múltiplos requerimentos considera que todos sejam necessários para o funcionamento do objeto;
  - composição\_ou: requerimentos técnicos necessários para a utilização do recurso utilizando o conector lógico *or*, ou seja, é necessário apenas um dos requerimentos para o funcionamento do objeto;
  - tanto os requerimentos ou os requerimentos de composição\_ou são divididos em:

---

<sup>1</sup> URI – Acrônimo para *Uniform Resource Identifier*, identifica um recurso único em uma rede de computadores. Normalmente, é dividido em esquema de nomenclatura, nome da máquina e nome do recurso.

- tipo: tipo de requerimento necessário. Pode ser *operating system* (sistema operacional) ou *browser* (navegador);
  - nome: nome do requerimento tecnológico. Se o tipo for *operating system*, os nomes possíveis são *pc-dos*, *ms-windows*, *macos*, *unix*, *multi-os* ou *none*. Se o tipo for *browser*, os nomes possíveis são *any*(qualquer), *netscape communicator*, *ms-internet explorer*, *opera* ou *amaya*.
  - versão mínima: menor versão possível do requerimento para que o recurso seja utilizado;
  - versão máxima: maior versão possível do requerimento para que o recurso seja utilizado;
  - instalação: descrição de como instalar este objeto de aprendizagem;
  - outras plataformas requeridas: informações sobre outros *softwares* ou *hardwares* requeridos;
  - duração: tempo que o recurso utiliza quando é visualizado por um estudante (normalmente utilizado para sons, vídeos ou animações).
- educacional: esta categoria descreve as características pedagógicas do objeto de aprendizagem. É dividida em:
- tipo de interação: modo predominante da aprendizagem suportada por este recurso. Seus valores podem ser:
    - *active*: conteúdo que induz diretamente a ação do estudante (aprendizagem pela realização – *learning by doing*);
    - *expositive*: a principal tarefa do educando é compreender o conteúdo apresentado pelo recurso (aprendizagem passiva);
    - *mixed*: este valor deve ser escolhido se o recurso apresenta, em partes relativamente iguais, uma composição ativa e expositiva.
  - tipo de recurso: especifica a categoria do objeto de aprendizagem. Deve ser utilizado o tipo dominante do recurso. Seus valores possíveis são *exercise* (exercícios), *simulation* (simulação), *questionnaire* (exercícios ou avaliações na forma de questionários), *diagram* (imagem contendo um diagrama explicativo), *figure* (imagem fotográfica ou desenhada), *graph* (gráfico), *index* (índice), *slide* (transparências virtuais de alguma apresentação), *table* (tabela), *narrative text* (texto narrativo), *exam* (teste), *experiment* (experimento de algum tipo), *problem statement* (formulação de algum problema), *self assessment* (auto aprendizagem) e *lecture* (texto).
  - nível de interatividade: a graduação da interatividade que caracteriza o objeto. Esta graduação implica em como o estudante pode influenciar no comportamento do recurso. Os valores possíveis são *very low* (muito pouco), *low* (pouco), *medium* (interatividade média), *high* (alta) e *very high* (muito alta);
  - densidade semântica: a graduação da concisão do recurso. Ela pode ser estimada em termos do seu tamanho e duração. Os valores possíveis são os mesmos do nível de interatividade. É importante salientar que a densidade semântica é independente da dificuldade, mas esta pode ser utilizada como um parâmetro. Exemplo de um objeto de densidade semântica baixa seria a

- explicação do problema do caixeiro viajante apresentado no capítulo 2. Um objeto de densidade semântica alta seria sua formulação matemática;
- papel do usuário final: usuário principal para o qual o objeto de aprendizagem foi intencionalmente construído. Seus valores possíveis são *teacher* (professor), *author* (autor), *learner* (estudante) ou *manager* (gerente);
  - contexto: o principal ambiente(contexto) o qual o recurso foi desenvolvido primeiramente. Os valores possíveis são *school* (escola de ensino fundamental ou médio), *higher education* (graduação ou pós-graduação), *training* (treinamento) ou *other* (outros);
  - idade típica: valores (inicial e final) das idades típicas para as quais foi desenvolvido o objeto;
  - dificuldade: o quão complexo e difícil é compreender o material exposto no objeto de aprendizagem. Seus valores possíveis são *very easy* (muito fácil), *easy* (fácil), *medium* (média), *difficult* (difícil) e *very difficult* (muito difícil);
  - tempo de aprendizagem típica: aproximadamente o tempo típico que alunos na idade típica para o contexto apresentando trabalham com o recurso;
  - descrição: comentários sobre como este objeto deve ser utilizado;
  - linguagem: linguagem que deve ser utilizada pelos usuários típicos deste objeto.
- direitos: descreve as propriedades legais e intelectuais, assim como as condições de uso do objeto de aprendizagem. Divide-se em:
    - custo: utilizado quando o objeto requer algum tipo de pagamento;
    - informações de *copyright* e outras restrições: informações que devem ser aplicadas ao recurso;
    - descrição: comentários sobre as condições de uso dos objetos.
  - relações: esta categoria define as relações existentes entre este recurso e outros objetos de aprendizagem, caso estas existam. Esta categoria possui os seguintes atributos:
    - tipo: natureza da relação entre os objetos. Pode assumir os valores: *ispartof* (é parte de), *haspart* (possui parte de), *isversionof* (é versão de), *hasversion* (possui versão de), *isformatof* (é um formato de), *hasformat* (possui formato de), *references* (referencia), *isreferencedby* (é referenciado por), *isbasedon* (é baseado em), *isbasisfor* (é base para), *requires* (requer) e *isrequiredby* (é requerido por);
    - recurso: o objeto alvo do qual a relação faz parte. É dividido em:
      - identificador: identificação *global* única do objeto;
      - catálogo: nome ou designação do esquema de catalogação utilizado para identificar formalmente o objeto;
      - entrada: valor do identificador ligado ao esquema de catalogação;
      - descrição: descrição textual do objeto.
  - anotações: esta categoria provê comentários acerca dos usos educacionais do objeto de aprendizagem, além de prover informações sobre quem e quando tais comentários foram introduzidos. Divide-se em:
    - entidade: pessoa ou organização que criou a anotação;
    - data: data de criação da anotação;

- descrição: conteúdo da anotação.
- classificação: o último grupo descreve a classificação do objeto de aprendizagem em relação a um particular sistema de classificação. Ele se divide em:
  - propósito: sob qual aspecto é classificado este objeto de aprendizagem. Seus valores possíveis são: *discipline* (disciplina), *idea* (idéia), *prerequisite* (pré-requisitos), *educational objective* (objetivos educacionais), *accessibility* (acessibilidade), *restriction* (restrições), *educational level* (nível educacional), *skill level* (nível de habilidade requerido).
  - taxonomia: verbetes utilizados de acordo com alguma ontologia em particular.

### 5.2.2 Empacotadores de objetos

Os empacotadores de objetos provêm uma forma de agregar funcionalidades ao objeto, tais como sua descrição formal através de um padrão de metadados bem como aspectos relativos a utilização pedagógica e/ou técnica do mesmo. Nesta seção, apresentamos dois empacotadores de objetos representativos destas técnicas: o SCORM e o IMS.

#### *SCORM – Sharable Content Object Reference Model*

O SCORM especifica técnicas e guias para a modelagem de agregação de conteúdo (identificação e agregação de recursos em conteúdos estruturados de aprendizagem) e os ambientes de execução (provendo a comunicação e rastreamento de conteúdo em ambientes baseados na *Internet*). Para tanto, o padrão é desenvolvido como um conjunto de quatro especificações, representados na figura 65:

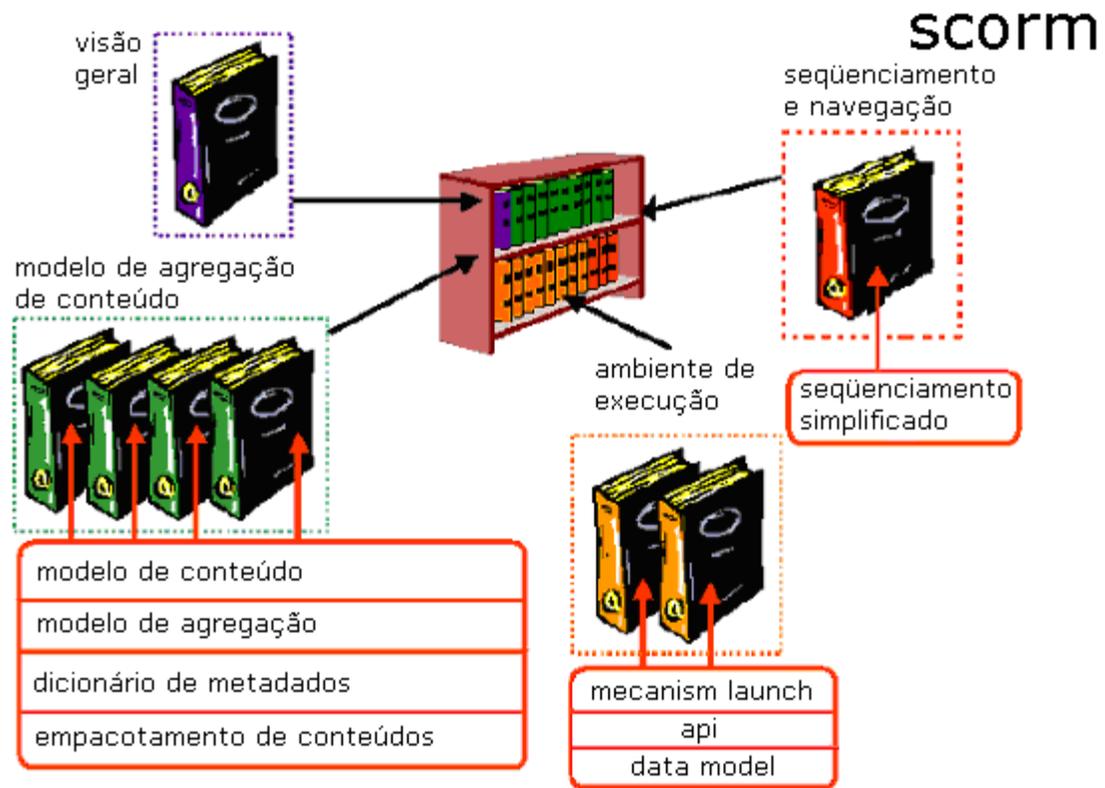


Figura 65 - Conjunto de especificações do SCORM (adaptado de SCORM, 2004)

### O Modelo de Agregação de Conteúdo

Este modelo define o *modelo de conteúdo*, o *modelo de agregação*, o *dicionário de metadados* e o *empacotamento de conteúdos*. O *modelo de conteúdo* contém as informações educacionais e didáticas e é dividido em:

- *assets*: forma mais básica de conteúdo de ensino (textos, imagens, sons e/ou outros em qualquer formato que pode ser enviado pela *Internet*);
- *sharable content objects* (SCO): representa um ou mais *assets* e deve incluir, obrigatoriamente, um mecanismo para a comunicação com os AVEA. Preferencialmente, devem ser pequenas unidades de ensino para posterior reutilização em diferentes contextos;
- *content aggregation model*: mapa utilizado para reunir os recursos de ensino catalogados anteriormente em uma unidade de instrução coesa e única, com uma estrutura bem definida. Esta estrutura permite definir a seqüência que será utilizada pelos educandos através da atribuição de pré-requisitos a cada conteúdo. Esta seqüência e lista de pré-requisitos devem ser interpretadas de forma correta pelo AVEA.

O *dicionário de metadados* descreve o conteúdo dos recursos de ensino do pacote. Esta informação é disponibilizada ao AVEA para que este organize os pacotes disponíveis e realize a pesquisa de forma mais eficiente. Em relação ao *empacotamento de conteúdos*,

este consiste em um conjunto de normas e regras para a agregação dos conteúdos de ensino em blocos (pacotes), permitindo a transferência dos mesmos entre diferentes AVEA. O *empacotador* é dividido em:

- manifesto: documento XML contendo informações sobre: (a) o bloco em si; (b) uma seção opcional (*organization*) descrevendo a estrutura e o comportamento das unidades de ensino; e (c) uma lista de apontadores para os recursos *empacotados*, sem nenhuma noção pré-definida de ordem de hierarquia;
- arquivos físicos: objetos propriamente ditos, geralmente guardados em arquivos PIF (*Package Interchange File*). Estes arquivos possuem diversos formatos, sendo os mais comuns os arquivos *zip*<sup>1</sup>, *jar*<sup>2</sup>, *tar*<sup>3</sup> e *cab*<sup>4</sup>.

O *empacotamento de conteúdos* e os *metadados* são utilizados para o agrupamento, organização e identificação de todos os objetos de aprendizagem necessários que pertencem a este recurso e são utilizados para disponibilizar uma ou unidades de aprendizagem em diferentes AVEA.

### O Ambiente de Execução

A especificação do *ambiente de execução* permite que os conteúdos de ensino possam ser visualizados em diferentes AVEA com o mesmo comportamento. Para tanto, é necessário especificar como os objetos SCORM são enviados para os navegadores *web*, definindo também o protocolo que os SCO e os AVEA utilizarão para a sua intercomunicação. Ele é dividido em três elementos:

- *mecanism launch*: envia o SCO para o navegador e realiza as tarefas iniciais de comunicação entre o objeto e o AVEA;
- *api (application program interface)*: cria um canal de comunicação entre o SCO e o AVEA;
- *data model*: define o protocolo e a linguagem de comunicação entre o objeto e os ambientes.

Este ambiente permite recolher informações sobre a situação do curso e do estudante, informando quais conteúdos e o progresso que um determinado educando está realizando em um curso. A padronização da linguagem de comunicação (*data model*) pode minimizar os problemas associados à troca de objetos entre os diversos AVEA (DUTRA, TAROUCO, KONRATH, 2006).

### Seqüenciamento e Navegação

Esta estrutura descreve como os conteúdos SCORM podem ser seqüenciados e como um AVEA deve interpretar as regras de seqüenciamento. Ele define também os requerimentos e limitações de cada objeto de aprendizagem e permite criar um encadeamento lógico entre os objetos. Em relação a navegação, este modelo permite que

---

<sup>1</sup> ZIP – Formato de compactação de arquivos.

<sup>2</sup> JAR – Arquivo compactado utilizado para distribuir um conjunto de classes em Java.

<sup>3</sup> TAR – Formato de arquivo que permite armazenar diversos arquivos em um só.

<sup>4</sup> CAB – Formato de compactação de arquivos para distribuição de *software*.

eventos de navegação sejam iniciados pelos próprios objetos com base no modelo de seqüenciamento.

### *IMS – Instructional Management Systems*

O *IMS* surgiu nos Estados Unidos em 1997, tendo como foco principal a educação superior. Seus objetivos incluem: (a) definição de técnicas padrões para a interoperabilidade de aplicações e serviços em aprendizagem distribuídos; (b) suportar a incorporação de especificações *IMS* em produtos e serviços; (c) adoção de especificações para que *AVEA* diferentes trabalhem com conteúdos de vários autores em conjunto; e (d) prover um conjunto de especificações não proprietárias (definição de metadados e especificação de questionários/avaliações para prover o estudo à distância e *on-line*) (KRATZ, 2006).

Para o *IMS*, um objeto de aprendizagem possui três divisões: seus objetivos, o conteúdo instrucional em si e as atividades de prática e *feedback*. O objetivo representa o que um determinado aluno pode compreender com aquele objeto. O conteúdo instrucional representa o material didático em si, contendo os arquivos físicos que fazem parte do mesmo. Finalmente, as atividades de prática e *feedback* devem fornecer ao aluno uma auto-verificação que permita ao mesmo verificar se os objetivos do objeto foram alcançados e, em caso contrário, a possibilidade de reutilizar partes ou todo o conteúdo instrucional novamente (IMS QTI, 2006).

Em (IMS, 2005) são apresentados os seis grupos de trabalho que tem como principais questões as especificações das normas *IMS*:

- *IMS Content Packagin Specification*: define as normas para o empacotamento e estruturação dos conteúdos educacionais, incluindo a formalização de *courses* como um conjunto de objetos educacionais em uma estrutura de árvore. Esta estrutura (*manifesto do curso*) define também as relações entre os módulos e seus componentes físicos (arquivos dos objetos);
- *IMS Enterprise Specification*: define os modelos de dados para a interoperabilidade entre os diversos *AVEA*;
- *IMS Question & Test Interoperability (QTI)*: define os modelos de dados básicos para representação de questões e/ou testes/avaliações, permitindo também a interoperabilidade de questões entre os diversos *AVEA*;
- *IMS Metadata Specification*: fornece uma estrutura organizada de elementos para descrever um conteúdo educacional, tornando a tarefa de busca e localização dos objetos mais segura e robusta;
- *IMS Learner Information*: define um modelo de dados normalizado para a troca de informações entre os utilizadores (aprendizes/alunos) além de prover um protocolo para a troca de dados e informações entre os diversos *AVEA*;
- *IMS Competency Definitions*: define um modelo descritivo das competências utilizadas em uma plataforma de ensino baseada por interações eletrônicas. Estas referências devem ser referidas por significados comuns a todos os participantes.

Para as considerações realizadas neste trabalho, nos deteremos mais especificamente na especificação do *IMS Content Packing*. Este padrão define como os objetos de aprendizagem são organizados e distribuídos em plataformas de *AVEA*

diferentes, fornecendo detalhes também sobre como será realizada a visualização do objeto de aprendizagem pelo estudante. A especificação define um pacote PIF (*Package Interchange File*) como centro da organização de um objeto de aprendizagem. Este pacote PIF possui duas áreas, representadas na figura 66 como *manifesto* e *arquivos físicos*.

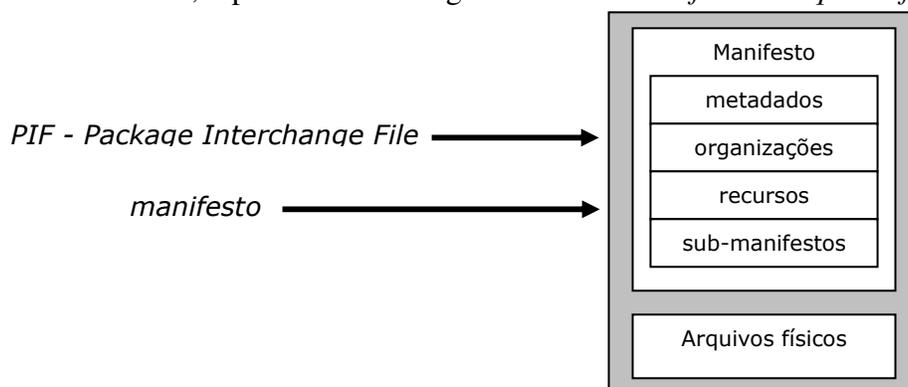


Figura 66 - *Package Interchange File*

O manifesto é estruturado através da linguagem *XML*, subdivido em quatro áreas para uma melhor organização. A primeira área, *metadados*, contém a descrição do pacote em si, explicitando de forma clara qual/quais conteúdo/conteúdos o pacote contém. A definição de quais metadados utilizar é de exclusividade do definidor do pacote. É possível utilizar o padrão LOM (seção 5.2.1) ou quaisquer outros padrões originados deste, tais como o UK Learning Object Metadata Core (UK LOM Core - CETIS, 2004) e o *RDN/Academy LOM Application Profile* (POWELL *et al.*, 2004).

A segunda área (organizações), contém uma ou mais estruturas ou organizações do pacote, referenciando principalmente a taxonomia organizacional dos dados referenciados. Os recursos representam a terceira área e eles estão organizados como uma coleção de referências aos recursos, sem nenhum tipo de hierarquia. Estes recursos farão parte dos *arquivos físicos*. Finalmente, os sub-manifestos contém as ligações com outros manifestos, incluindo nesta área a possibilidade da criação da árvore de conhecimento de um determinado curso.

Após o manifesto, é anexado os arquivos físicos, que contém todos os arquivos referenciados em *recursos*. Estes arquivos podem ser quaisquer tipos de mídias, tais como vídeos, animações, textos ou figuras.

Adicionalmente aos pacotes do IMS (estrutura dos objetos), é possível também representar *atividades de aprendizagem* (incorporação da reusabilidade de atividades e projetos de ensino), através do *IMS Learning Design*. Este modelo prevê o suporte a uma grande variedade de abordagens de ensino-aprendizagem através de uma linguagem genérica, utilizando *unidades de aprendizagem* que são definidas como unidades elementares que provêm eventos de aprendizagem para os educandos, com o objetivo de satisfazer um ou mais *objetivos* de aprendizagem. Detalhes da utilização desta nova metodologia podem ser encontrados em (DUTRA, TAROUCO, KONRATH, 2005b), (DUTRA, TAROUCO, KONRATH, 2006) e (KOPER, OLIVIER, 2004).

### 5.3 Conclusões do capítulo

O pacote SCORM tem como principal preocupação a organização do conteúdo, sua visualização e o rastreamento das ações do aluno. No entanto, outras operações, além das que ocorrem especificamente entre o aluno e os objetos de aprendizagem, não são descritas no pacote (DUTRA, TAROUCO, KONRATH, 2006). Em relação ao IMS, com a definição do *IMS Learnign Design*, este se concentra muito mais na relação ensino-aprendizagem do que propriamente nas inter-relações fixas entre o aluno e objeto, tentando descrever um *framework* que abarque todo o processo de aprendizagem e não somente do conteúdo do objeto em si. No entanto, é interessante observar que, em algum momento, ambos os padrões utilizam metadados para a descrição dos objetos sem, contudo, uma especificação formal ou obrigatório de qual conjunto de descritores deve ser utilizado.

Analisando os padrões *Dublin Core* e *LOM*, o último se apresenta de forma muito mais completa (e complexa) do que o primeiro. Apesar das dificuldades em catalogar um objeto seguindo o padrão LOM de forma completa (IMS, 2006), ele apresenta alguns pontos positivos que se destacam: sua utilização em empacotadores de objetos, como citado acima, principalmente seguindo o padrão *Instructional Management Systems (IMS) - IMS Learning Metadata*; a possibilidade de se utilizar um sub-conjunto dos grupos e atributos acima. Como definido em (IEEE, 2002), um conjunto de metadados que engloba alguns dos grupos e atributos da IEEE 1484.12.1 também é considerado compatível com o padrão estabelecido. Desta forma, os projetos de desenvolvimento de objetos educacionais podem estabelecer seus próprios sub-conjuntos de atributos e manter a compatibilidade entre os sistemas.

Desta forma, objetos que agregam um conjunto de metadados no padrão LOM podem ser utilizados em diversos empacotadores de objetos, bem como serem distribuídos na *internet*, desde que mantenham o seu conjunto de descritores. As opções tecnológicas atuais dos empacotadores agregam o objeto e seus metadados em arquivos compactados. No entanto, a opção por esta tecnologia exige que o ambiente e/ou o usuário que estiver interessado no objeto em si possua o *software* necessário para realizar a descompactação do objeto. Com o objetivo de eliminar esta necessidade e de prover tanto um ambiente virtual de ensino aprendizagem como um objeto que possa ser disseminado livremente pela *internet*, o AMEM e o LOBO (*Learning Object Based on Optimization* – objeto implementado para o ensino de heurísticas e metaheurísticas e que será apresentado no próximo capítulo) foram desenvolvidos de forma a manterem junto ao seu sistema um conjunto de metadados baseados na IEEE 1484.12.1 (LOM) para consulta e catálogo em bibliotecas virtuais de objetos educacionais. Estes descritores podem ser exportados ou importados pela grande maioria dos ambientes que utilizam objetos educacionais padronizados, assim como é possível importar objetos no padrão LOM ou IMS a partir do seu conjunto de descritores.

## 6 LOBO – Um Objeto de Aprendizagem para o Ensino de Heurísticas e Metaheurísticas

O objeto de aprendizagem LOBO (*Learning Object Based on Optimization*) foi desenvolvido com o objetivo de propiciar aos professores e alunos uma base comum para as experimentações necessárias à condução das atividades didáticas que envolvem o ensino na área de heurísticas e metaheurísticas. Como apresentando anteriormente (capítulo 2), a didática nesta área do saber é carente de ferramental e pesquisas que envolvam especificamente os aspectos tecnológicos e didáticos do fazer e aprender dos professores e alunos. O LOBO pode ser utilizado como um *objeto de aprendizagem* para a condução de investigações iniciais que explorem a área de heurísticas e metaheurísticas para problemas combinatórios, bem como um *framework* de desenvolvimento de algoritmos. Desta foram, foram definidos dois modos de utilização do objeto: o modo tutoriado e o modo *expert*.

O modo tutoriado apresenta uma série de telas informativas ao usuário, fornecendo dicas e informações acerca da implementação do LOBO, bem como explicações mais detalhadas dos algoritmos implementados e a teoria do desenvolvimento de heurísticas e metaheurísticas. O modo *expert* não apresenta estas informações, transformando o LOBO, deste modo, em uma ferramenta de experimentação pura, onde os resultados são interpretados pelos próprios usuários. Ao entrar no ambiente, o usuário receberá uma tela de boas-vindas, representada na figura 67. As informações contidas nesta apresentação informam a natureza do objeto e seu objetivo, o problema base do LOBO e os dois modos de apresentação, avisando também que, por padrão, são apresentadas as informações no modo tutoriado.

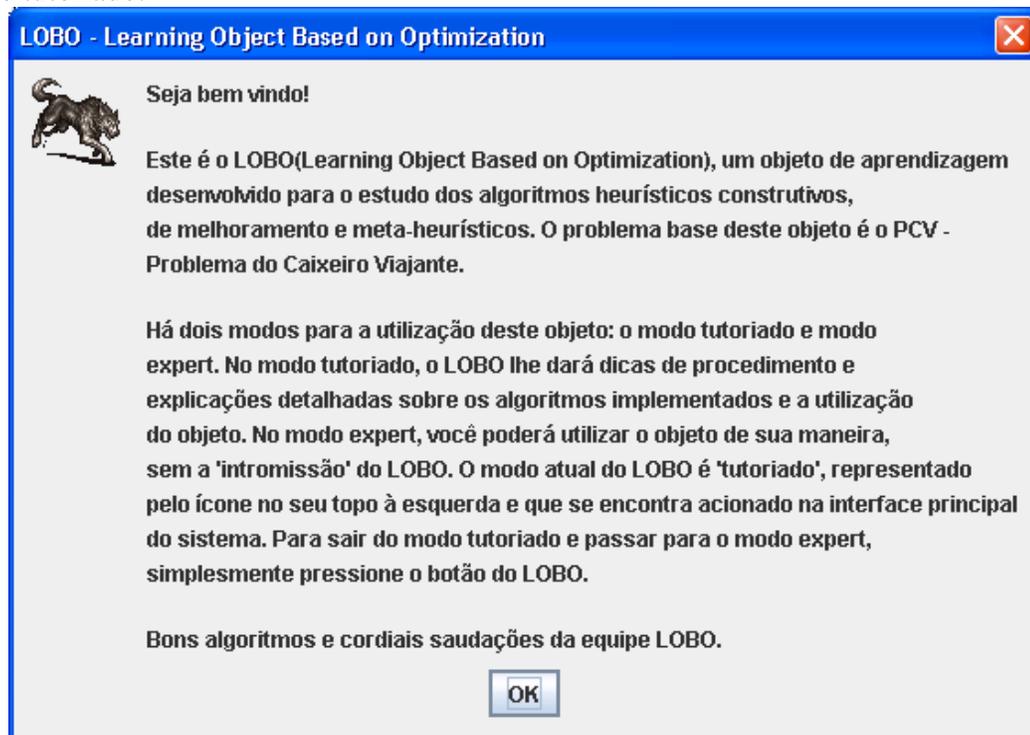


Figura 67 - Tela de boas vindas do objeto LOBO

Ao fechar a tela de boas vindas, é apresentado ao usuário a interface completa do objeto LOBO, formada por quatro áreas distintas, representadas na figura 68: (a)menu principal, (b)botões de ação, (c)histórico dos resultados e (d)área de simulação. Como é possível notar, o quarto botão se encontra pressionado. Este botão representa os modos do objeto. Na posição padrão, o modo tutoriado está ativo e o botão permanece pressionado. Ao clicar sobre o mesmo, o modo *expert* é ativado e o botão apresentará o estado desativado. Os demais botões representam, na ordem, as seguintes ações: execução da heurística, execução de um passo da heurística e exibição da rota ótima, caso ela existir. Estas ações serão explicadas detalhadamente nas próximas páginas. A primeira área do objeto apresenta o menu principal do LOBO. Este menu é dividido em quatro sub-menus: *O Problema*, *Algoritmos*, *Dados de teste* e *Sobre*.

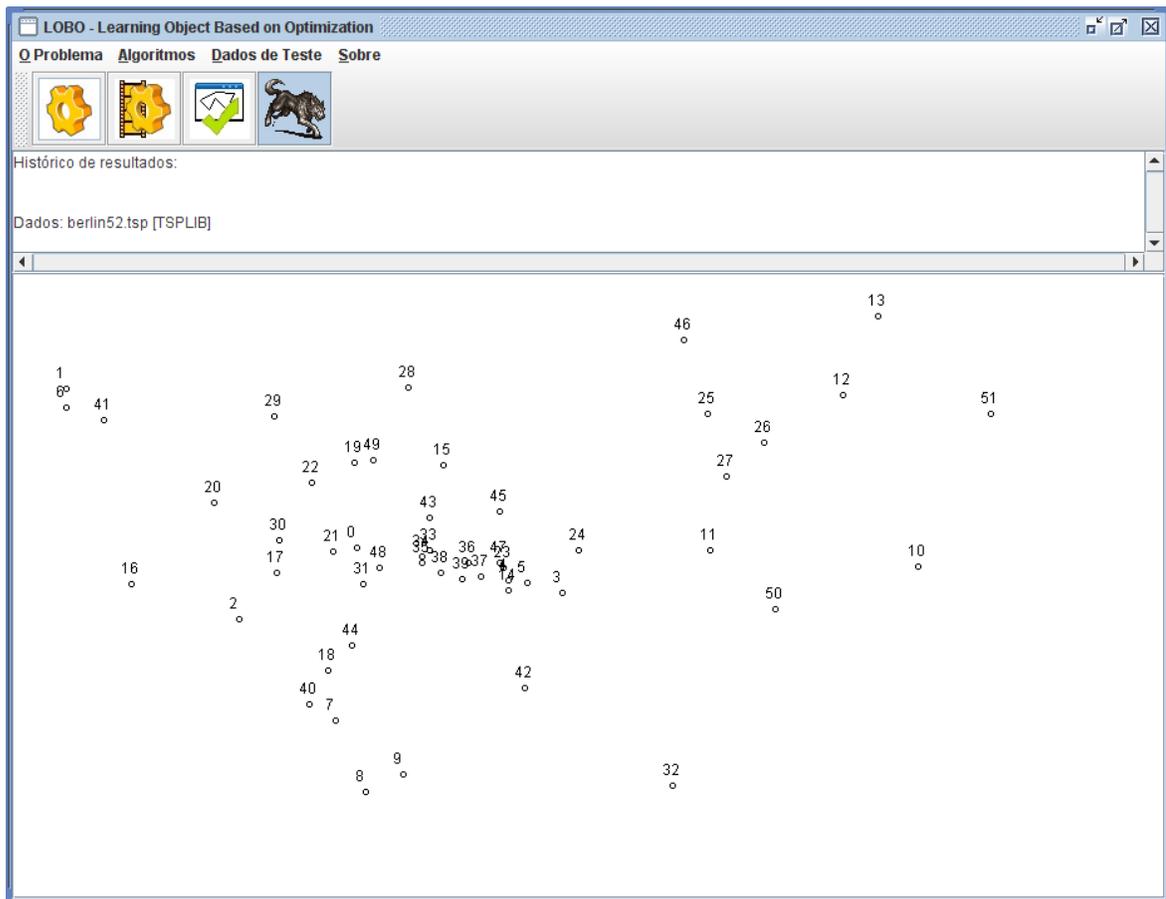


Figura 68 - Interface principal do LOBO

## 6.1 Menu Principal

### 6.1.1 Sub-Menu: O Problema

O sub-menu *O Problema* apresenta duas opções: história e definição do PCV. A opção *História* apresenta um breve apanhado sobre a história e desenvolvimento do Problema do Caixeiro Viajante, bem como dos problemas que o precederam no seu desenvolvimento, conhecidos como o Problema de Euler e o Problema de Hamilton.

Maiores detalhes sobre estes problemas clássicos no desenvolvimento da otimização combinatória podem ser encontrados em (REI, 1994) e (LAWLER *et al.*, 1985). A figura 69 apresenta a tela de abertura da apresentação da história do caixeiro viajante. Ela foi desenvolvida através de páginas HTML que estão embutidas dentro do próprio objeto.



Figura 69 - Apresentação da História do Problema do Caixeiro Viajante

A primeira seção da apresentação (motivação história) é apresentada na figura 70, que é representativa de todas as demais seções desenvolvidas. Na parte inferior da página, encontram-se três *links*: anterior (que retorna para a última seção visitada), índice (que retorna para a tela apresentada na figura 69) e próximo (que desloca a apresentação para a próxima seção). O texto das seções desenvolvidas foi desenvolvido baseado em (LAWLER *et al.*, 1985 e ARAÚJO, 2006).

O objetivo desta apresentação é trazer para os estudantes um pouco da história que permeia o desenvolvimento do problema escolhido para o objeto. Os principais objetivos aqui envolvem a discussão sobre os aspectos históricos reais do desenvolvimento da otimização combinatória, bem como demonstrar os passos na definição do problema que ocorreram pela análise de situações reais e sua teorização através do arcabouço matemático.

A segunda opção deste sub-menu apresenta a definição do Problema do Caixeiro Viajante, tal como discutido na seção 2.2. Esta definição, mais formal que a apresentação anterior, fornece ao estudante as fórmulas matemáticas necessárias para a compreensão dos algoritmos implementados no objeto de aprendizagem assim como permite que os mesmos possam desenvolver suas próprias soluções através do desenvolvimento de algoritmos, caso o professor solicite.

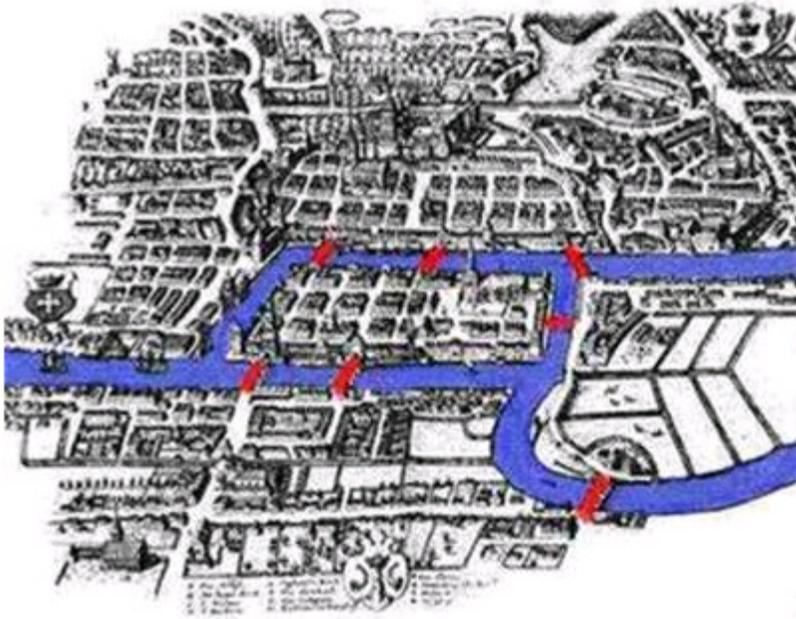
História do Problema do Caixeiro Viajante

### O Problema do Caixeiro Viajante

#### Motivação Histórica

- As Pontes de Königsberg

A cidade de Königsberg é banhada pelo rio Pregel que, ao atravessar a cidade, se ramifica, formando a ilha de Kneiphof. Esta ilha está ligada à restante da cidade por sete pontes. Dizia-se que os habitantes da cidade, nos dias de descanso, tinham por hábito tentar encontrar um percurso que os obrigasse a passar por todas as pontes, mas apenas uma vez em cada uma. Como as suas tentativas foram sempre falhas, muitos deles acreditavam que não era possível encontrar tal percurso. Será que tinham razão?



<< Voltar      Índice      Próximo >>

Fechar

Figura 70 - Primeira seção da História do Problema do Caixeiro Viajante

### 6.1.2 Sub-Menu: Algoritmos

O sub-menu algoritmos do objeto de aprendizagem apresenta a principal escolha disponível para o aluno no objeto. A opção por um determinado algoritmo heurístico é dada pela seguinte lista: (a) construtivo, aleatório; (b) construtivo, vizinho mais próximo; (c) construtivo, inserção mais distante; (d) melhoramento, 2-opt; (e) melhoramento, 3-opt; (f) metaheurística *simulated annealing*; e (g) metaheurística busca tabu. Ao selecionar um determinado algoritmo, o aluno se torna apto a executá-lo através dos botões de ação. A opção padrão do LOBO é a heurística construtiva aleatória.

No modo tutoriado, ao escolher um algoritmo, o aluno é apresentado a uma tela explicativa sobre o mesmo. A figura 71 apresenta a tela do algoritmo construtivo vizinho mais próximo. Como é possível observar, o aluno recebe uma mensagem explicativa sobre o algoritmo e tem três opções (disponíveis para todos os algoritmos): sair da explicação e rodar o algoritmo seguindo os passos descritos na apresentação (botão “Fechar”); selecionar explicações mais detalhadas, através do botão “Heurísticas” ou “Metaheurísticas” e visualizar o algoritmo em sua formulação matemática, através do botão “Algoritmo”. A segunda opção apresenta uma tela explicativa sobre o tipo de algoritmo que o aluno escolheu (heurística construtiva, heurística de melhoramento ou metaheurística). A figura 72 apresenta a tela explicativa dos algoritmos construtivos.

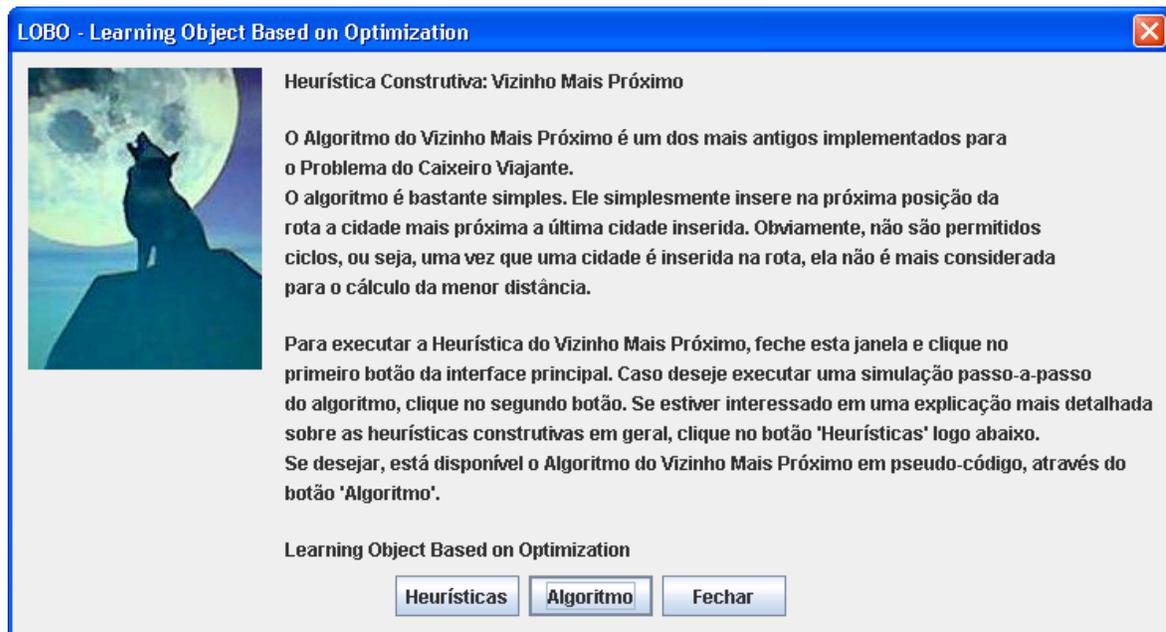


Figura 71 - Tela explicativa da Heurística Vizinho Mais Próximo

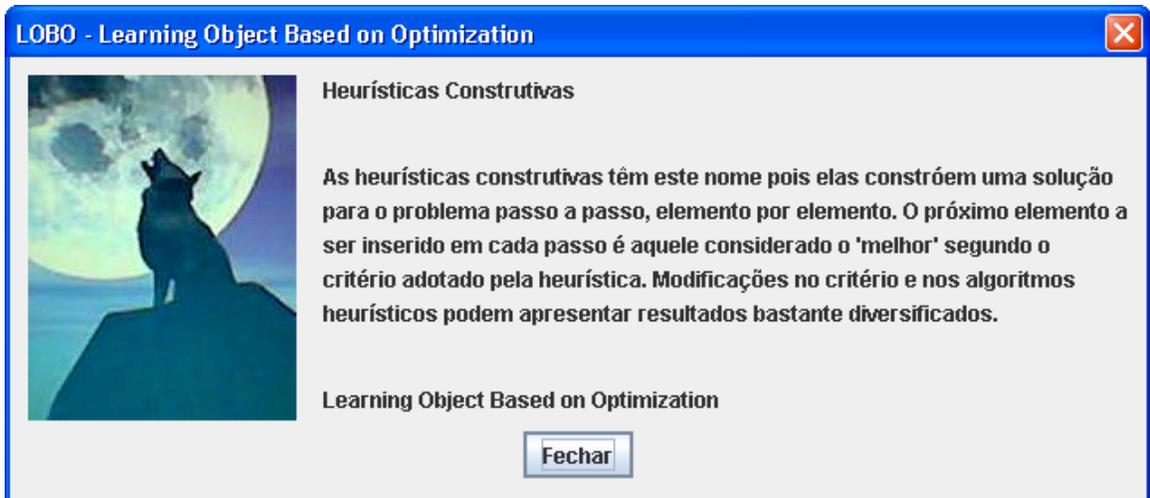


Figura 72 - Tela explicativa sobre os algoritmos construtivos

A figura 73 apresenta a tela explicativa da heurística construtiva Inserção Mais Distante com os botões “*Heurísticas*”, “*Algoritmo*” e “*Fechar*” na sua borda inferior. A figura 74 apresenta o algoritmo formal que é apresentado ao aluno ao clicar sobre o botão “*Algoritmo*”.

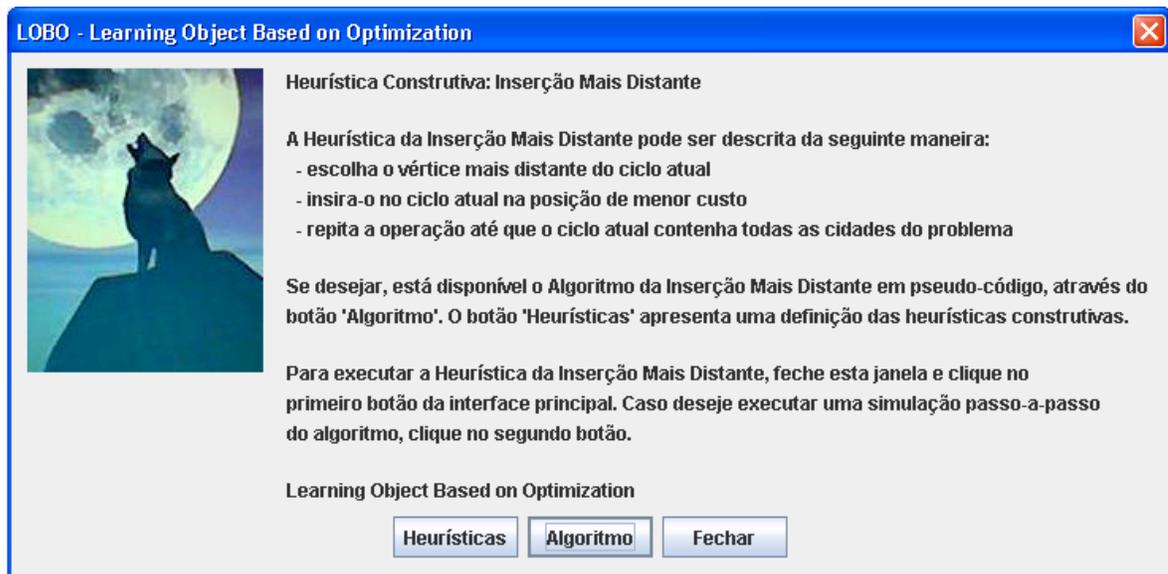


Figura 73 - Tela explicativa da Heurística Inserção Mais Distante

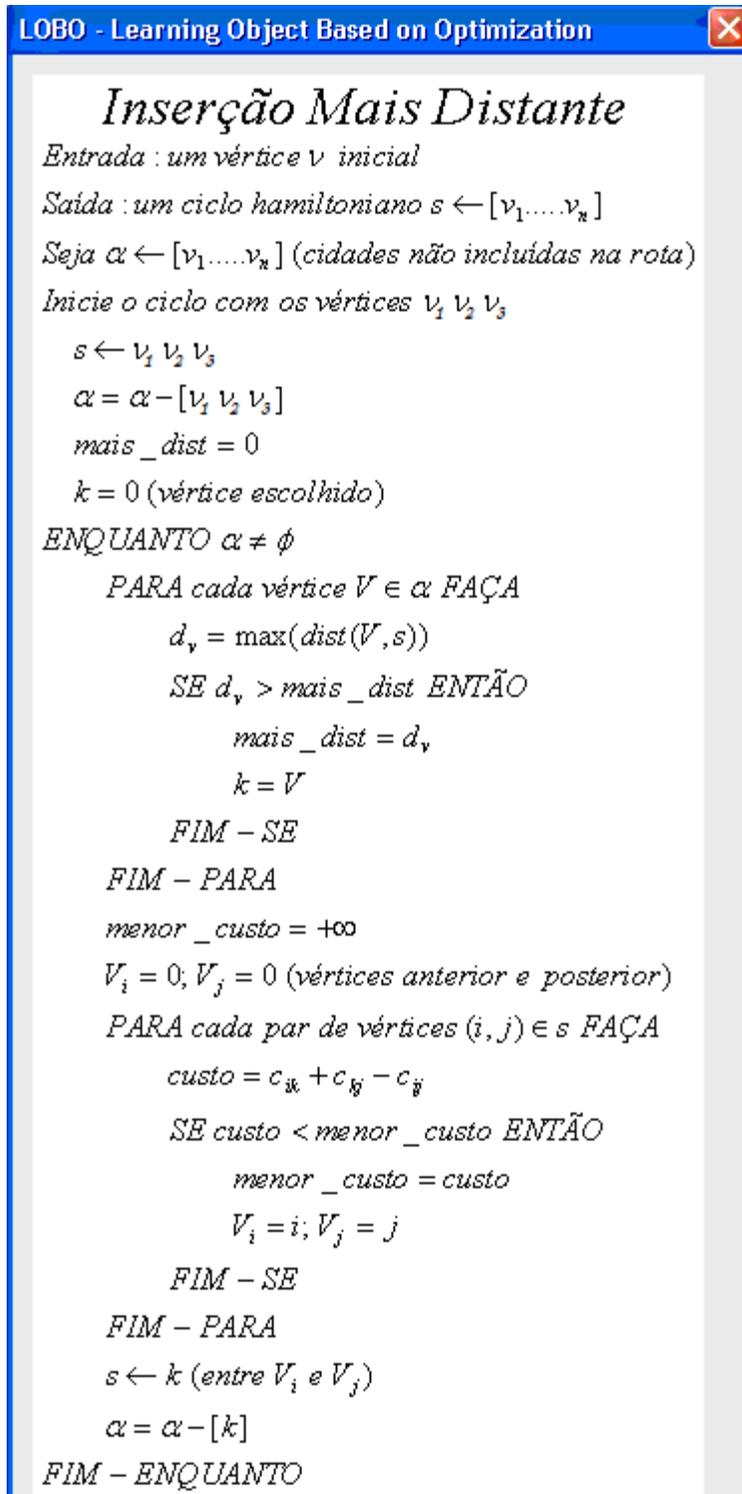


Figura 74 - Algoritmo formal da Heurística Inserção Mais Distante

Para as heurísticas de melhoramento (2-Opt e 3-Opt), também existem três botões disponíveis para o usuário. O primeiro, “Heurísticas”, o aluno tem acesso a uma explicação sobre as heurísticas de melhoramento na forma geral. O segundo botão, “Algoritmo”, apresenta um texto em formato HTML onde o aluno recebe informações

mais detalhadas sobre a heurística, uma animação da utilização do mesmo e a apresentação do algoritmo formal. A tela da figura 75 é a apresentação que o aluno recebe sobre a heurística de melhoramento 2-opt.

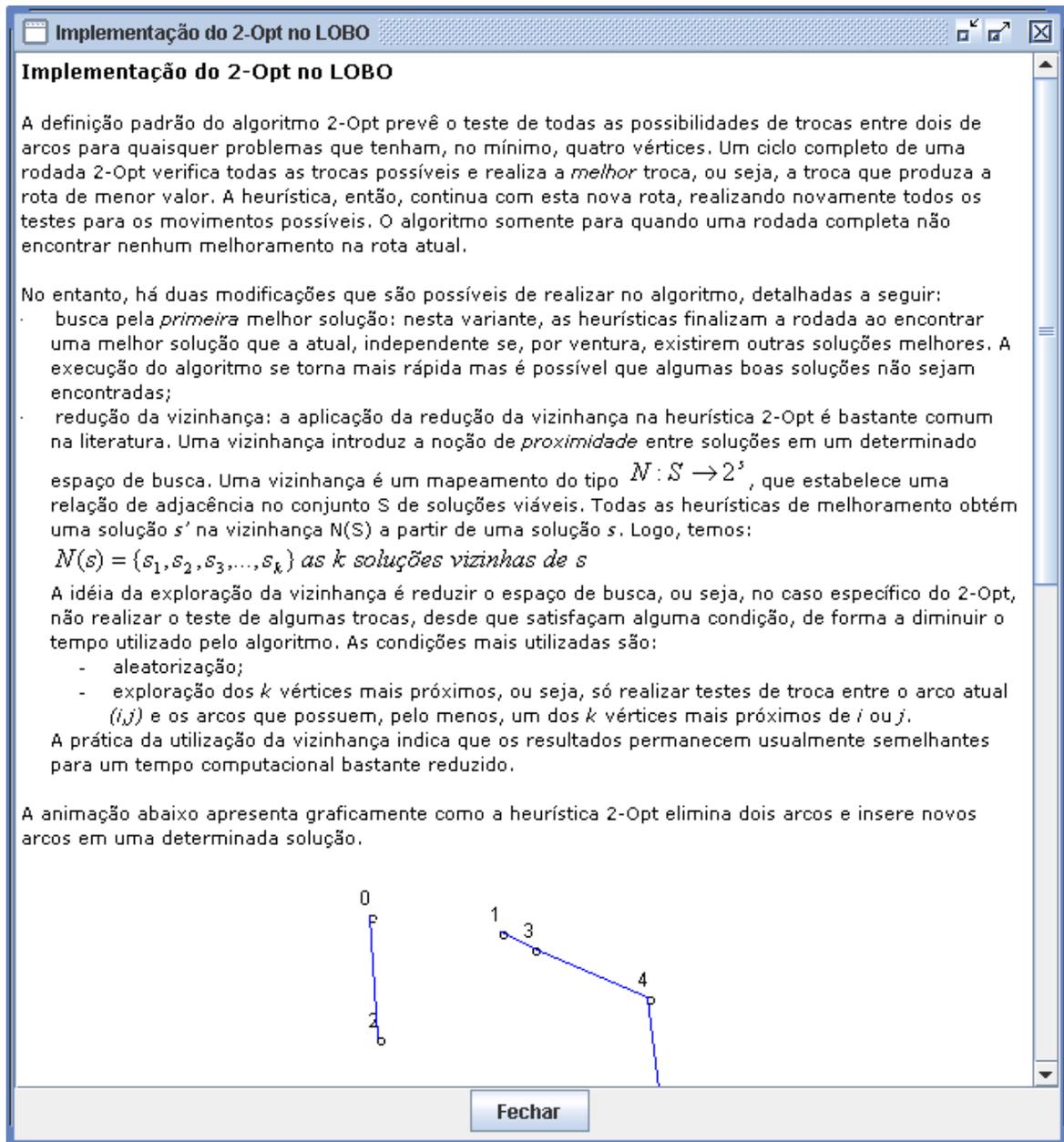
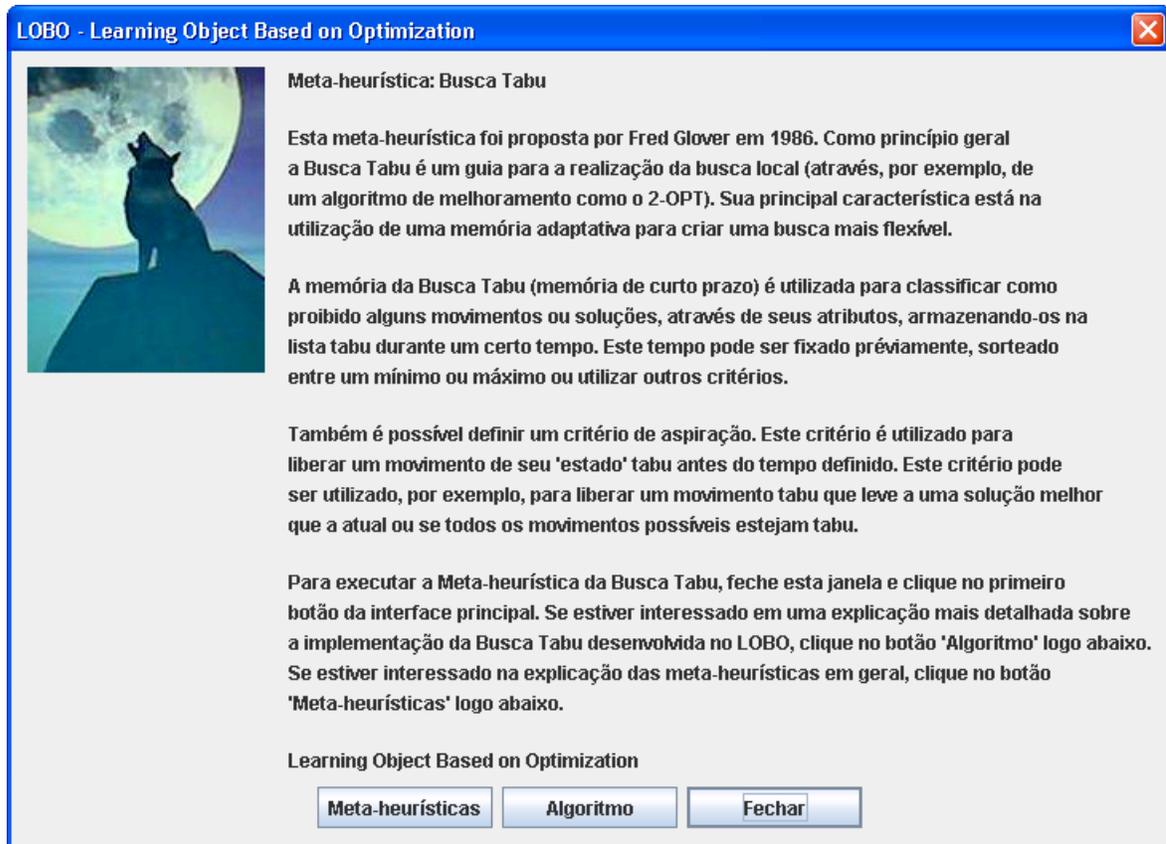


Figura 75 - Apresentação sobre a heurística de melhoramento 2-Opt

Em relação às duas metaheurísticas implementadas, o botão *Algoritmo* apresenta uma explicação acerca da *implementação* do algoritmo escolhido. Tal fato decorre da própria natureza destes algoritmos, que apresenta particularidades que devem ser definidas em tempo de construção do *software*. Desta forma, para que os alunos interpretem os resultados do LOBO de forma mais adequada, se torna necessário explicitar as decisões de

projeto para os algoritmos metaheurísticos implementados, o que não é necessário para as demais heurísticas, onde sua própria definição formal é estática.

A figura 76 apresenta a tela explicativa da metaheurística Busca Tabu. Ao clicar sobre o botão “*Metaheurística*”, o aluno é transportado para outra tela, implementada através de páginas HTML, tal como as apresentações do menu “*O Problema*”. Esta decisão foi tomada pois as explicações necessárias exigiam um espaço físico maior, como é possível perceber na figura 77. Esta figura apresenta formalmente a implementação da metaheurística busca tabu no LOBO, através de um texto explicativo e da apresentação do próprio algoritmo, logo abaixo.



**LOBO - Learning Object Based on Optimization**

**Meta-heurística: Busca Tabu**

Esta meta-heurística foi proposta por Fred Glover em 1986. Como princípio geral a Busca Tabu é um guia para a realização da busca local (através, por exemplo, de um algoritmo de melhoria como o 2-OPT). Sua principal característica está na utilização de uma memória adaptativa para criar uma busca mais flexível.

A memória da Busca Tabu (memória de curto prazo) é utilizada para classificar como proibido alguns movimentos ou soluções, através de seus atributos, armazenando-os na lista tabu durante um certo tempo. Este tempo pode ser fixado previamente, sorteado entre um mínimo ou máximo ou utilizar outros critérios.

Também é possível definir um critério de aspiração. Este critério é utilizado para liberar um movimento de seu 'estado' tabu antes do tempo definido. Este critério pode ser utilizado, por exemplo, para liberar um movimento tabu que leve a uma solução melhor que a atual ou se todos os movimentos possíveis estejam tabu.

Para executar a Meta-heurística da Busca Tabu, feche esta janela e clique no primeiro botão da interface principal. Se estiver interessado em uma explicação mais detalhada sobre a implementação da Busca Tabu desenvolvida no LOBO, clique no botão 'Algoritmo' logo abaixo. Se estiver interessado na explicação das meta-heurísticas em geral, clique no botão 'Meta-heurísticas' logo abaixo.

Learning Object Based on Optimization

Meta-heurísticas    Algoritmo    Fechar

Figura 76 - Tela explicativa da Metaheurística Busca Tabu

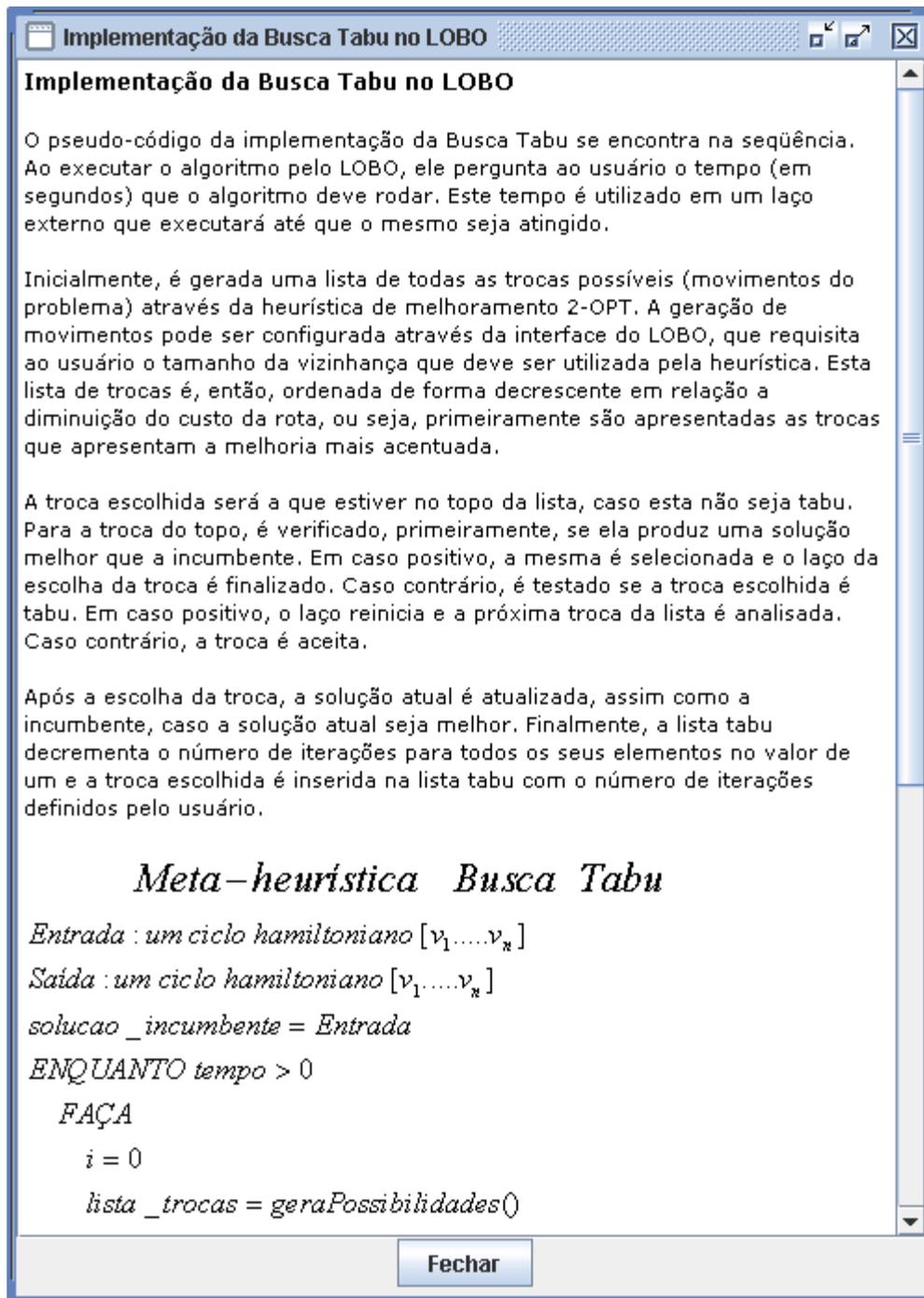


Figura 77 - Apresentação sobre a implementação da metaheurística Busca Tabu

No modo *expert* não são apresentadas nenhuma das telas explicativas definidas anteriormente. Ao escolher um algoritmo, ele simplesmente se tornará selecionado para execução. O funcionamento dos algoritmos citados aqui será mais detalhado na seção 6.2.

### 6.1.3 Sub-Menu: Dados de Teste

O menu dados de teste apresenta sete opções para o usuário escolher. As cinco primeiras opções representam dados colhidos na biblioteca de problemas e algoritmos para o Problema do Caixeiro Viajante conhecida como TSPLib (REINELT, 1991). A TSPLib apresenta um repertório de dezenas de problemas para as mais diversas variantes do PCV, contendo as instâncias e, quando existirem, os valores ótimos e suas respectivas rotas. Desta forma, esta base se tornou rapidamente uma referência mundial para testes na área da otimização combinatória. A biblioteca está disponível em <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>.

Os problemas escolhidos para integrarem o LOBO estão listados abaixo:

- berlin 52: problema padrão apresentando ao aluno quando este inicia o objeto de aprendizagem. Este problema foi proposto por Groetschel e representa 52 localidades na cidade de Berlin, Alemanha;
- pr 76: o segundo problema representa um conjunto de 76 cidades distribuídas de forma a gerar *clusters* nas quatro pontas de um quadrado imaginário que embarcaria todas as cidades. Este problema foi definido por Padberg e Rinaldi;
- eil 51: o terceiro problema representa um conjunto de 51 cidades distribuídas uniformemente em um retângulo imaginário. O problema foi proposto por Christofides e Eilon;
- kroA 100: o quarto problema também representa uma distribuição relativamente uniforme em um retângulo imaginário, mas o número centenário de cidades o torna de mais difícil resolução. O problema foi proposto por Krolak, Felts e Nelson;
- lin 105: finalmente, o quinto e último problema padrão apresenta 105 cidades distribuídas em diversos *clusters* esparsados em uma grande região. Este problema foi proposto por Lin e Kernighan.

A escolha dos problemas supra-citados para a implementação do LOBO levou em conta as seguintes características:

- formato de problema: coordenadas cartesianas. O objeto de aprendizagem foi implementado para abrir e processar problemas que estejam definidos através de suas distâncias euclidianas. Outros formatos definidos pela TSPLib não estão correntemente implementados;
- existência da rota ótima: para que os alunos percebessem a diferença entre os algoritmos implementados e sua performance em relação à rota ótima do problema, só foram levados em consideração instâncias que apresentavam o custo e a rota ótima associada a este;
- diversidade: para que os algoritmos possam ser testados em sua plenitude, é necessário que o mesmo seja confrontado com os mais diversos tipos de problemas. Desta forma, foram escolhidos problemas com dimensões e características diversas, auxiliando os alunos na compreensão da performance dos algoritmos frente a adversidades diferentes.

No modo tutoriado, a troca de dados do objeto resulta também na visualização de uma tela de apresentação do problema. Esta tela, visualizada para o conjunto de dados *berlin 52* na figura 78, apresenta o nome do problema, seu criador, a dimensão do mesmo e

seu custo ótimo. Uma pequena explicação sobre como visualizar a rota ótima é fornecida e duas opções são abertas para o usuário: (a) selecionar o botão *Saber mais...* para visualizar uma explicação mais detalhada sobre o TSPLib; ou (b) fechar a visualização para escolher outra ação. O formato TSPLib é apresentado nas próximas páginas.

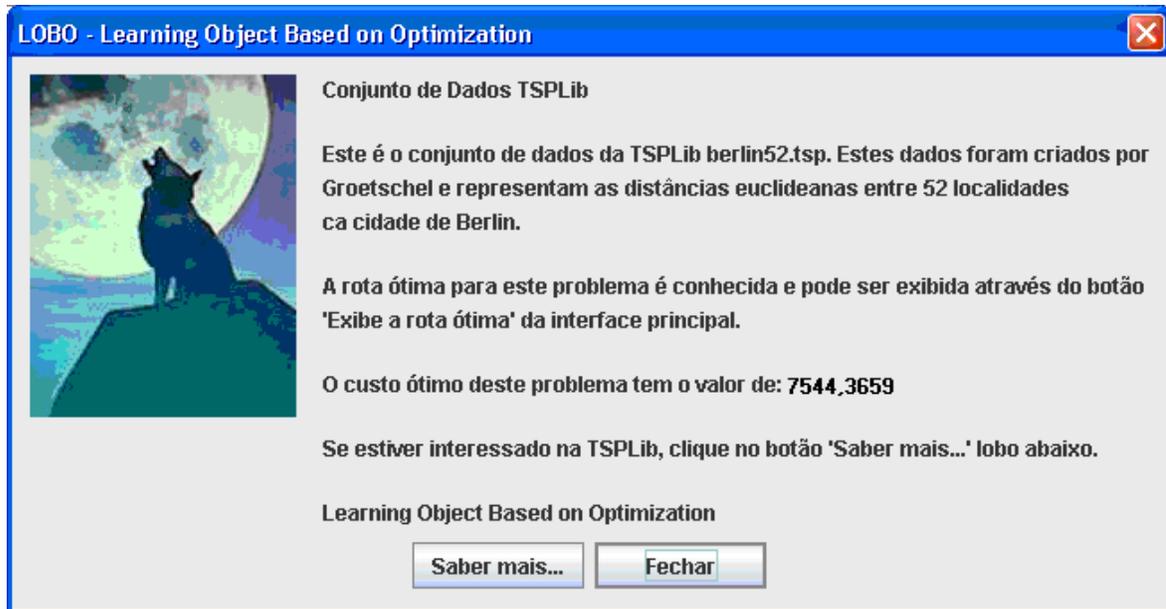


Figura 78 - Apresentação do conjunto de dados berlin52.tsp

Apesar da diversidade dos problemas escolhidos, é bastante claro que este pequeno conjunto de dados não representa todas as possibilidades ou os mais diversos problemas que podem ocorrer em situações reais. Para tanto, duas funcionalidades foram agregadas ao LOBO com o intuito de fornecer ao professor e aos alunos possibilidades de interação mais ricas. A primeira funcionalidade se refere a possibilidade do usuário inserir seus próprios dados. A criação manual de um novo problema permite que o professor ou o aluno possa inserir dados com suas necessidades específicas, apresentando para o usuário, desta forma, os resultados da aplicação dos algoritmos nas instâncias as quais o docente sente necessidade de explorar. O LOBO pergunta ao usuário quantas cidades o problema conterà, qual cidade será considerada o ponto de origem e abre uma janela para a digitação dos dados, como demonstra a figura 79. O usuário deverá digitar as coordenadas cartesianas de cada cidade para inserir os dados no objeto de aprendizagem.

Cidade	Posição X	Posição Y
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		

Ok  
Cancelar

Figura 79 - Funcionalidade Inserir dados

No entanto, o processo de digitação manual de dados só se torna interessante para problemas de pequena monta. Para problemas maiores, esta ação se torna enfadonha e pouco produtiva (os dados não são salvos, logo, ao deixar o sistema, o usuário deverá redigitar novamente todo o problema se quiser rodar novamente os algoritmos). Desta forma, se optou por inserir outra funcionalidade, através da abertura de arquivos. Esta funcionalidade abre arquivos no padrão TSPLib, definido por uma estrutura baseada em linhas de um arquivo texto. Usualmente, os arquivos deste padrão contêm cinco linhas iniciais:

- *name*: nome do problema;
- *comment*: comentários acerca do problema, contendo, normalmente, o(s) nome(s) do(s) autor(es);
- *type*: tipo do problema. Para o objeto LOBO, só serão aceitos problemas do tipo TSP (*Traveling Salesman Problem* – Problema do Caixeiro Viajante);
- *dimension*: número de nós do problema;
- *edge weight type*: tipo de entrada de dados para os pesos dos arcos. O objeto desenvolvido aceita somente entradas do tipo EUC\_2D, ou seja, euclidiana em duas dimensões;
- *node\_coord\_section*: inicia a seção das coordenadas. Cada coordenada (x,y) é precedida por um número, que varia entre 1 e o tamanho da dimensão.

Ao final do arquivo, após a última coordenada, deve se encontrar uma linha com a sentença EOF (*end of file* – fim de arquivo). Um extrato de um arquivo no formato TSPLib se encontra na figura 80.

```

NAME : a280
COMMENT : drilling problem (Ludwig)
TYPE : TSP
DIMENSION: 280
EDGE_WEIGHT_TYPE : EUC_2D
NODE_COORD_SECTION
  1 288 149
  2 288 129
  3 270 133
  4 256 141
  ...
279 260 129
280 280 133
EOF

```

Figura 80 - Extrato do arquivo de dados no formato TSPLib

A TSPLib também fornece um padrão para os dados que contém a rota ótima, ou seja, a configuração final do problema que apresenta o menor custo. Este arquivo contém quatro linhas iniciais:

- *name*: nome do arquivo;
- *type*: tipo do arquivo. Neste caso específico, o tipo sempre será *tour*;
- *dimension*: número de nós do problema;
- *tour\_section*: inicia a seção do *tour*, que é apresentado através de uma sucessão de cidades representadas numericamente. Desta forma, um *tour* que contenha os dados 3,4,2 e 1 representa que a rota ótima deve partir da cidade 3, se deslocar para a cidade 4, posteriormente para a 2, seguir até a 1 e retornar novamente para a três, fechando o *tour*.

Ao final do arquivo, após a última cidade listada, deve se encontrar uma linha com o número -1, que representa o final do *tour*. Um extrato de um arquivo de rota ótima no formato TSPLib se encontra na figura 81.

```

NAME : ./TSPLIB/a280.tsp.optbc.tour
TYPE : TOUR
DIMENSION : 280
TOUR_SECTION
1
2
242
243
...
3
280
-1

```

Figura 81 - Extrato do arquivo contendo a rota ótima no formato TSPLib

Como ambos os tipos de arquivos são muito fáceis de se construir com qualquer editor de texto no padrão ASCII<sup>1</sup>, o professor que utilizar o objeto de aprendizagem pode preparar seus próprios exemplos e disponibilizá-los para os alunos, bem como os próprios alunos podem carregar arquivos da TSPLib ou criar seus próprios modelos e compartilhar com os demais. A funcionalidade *Abrir* do LOBO permite que qualquer usuário carregue um arquivo contendo os dados nos formatos discutidos anteriormente. Se a rota ótima existir, ele poderá carregá-la também na mesma entrada. Caso contrário, esta opção pode ser deixada em branco. A interface para a abertura dos arquivos no formato TSPLib pode ser visualizada na figura 82.

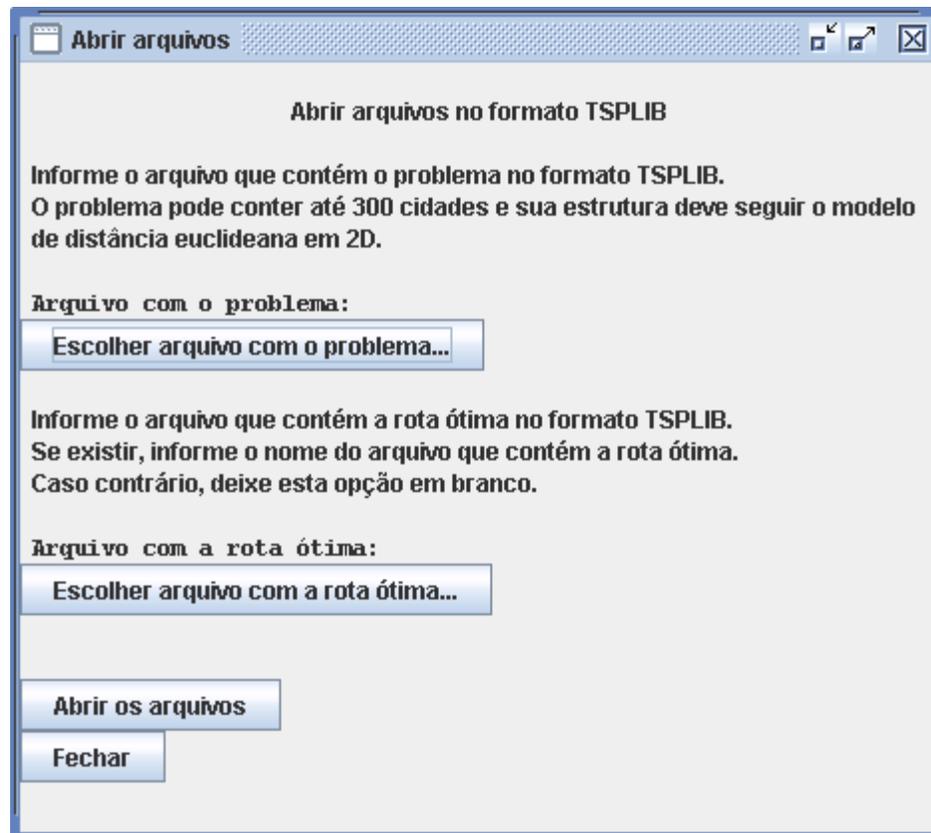


Figura 82 - Funcionalidade Abrir arquivos

#### 6.1.4 Sub-Menu: Sobre

O sub-menu *Sobre* apresenta somente duas opções. A primeira, *Ajuda*, apresenta uma ajuda rápida ao sistema, implementada no formato de páginas HTML embutidas ao próprio objeto.

A segunda opção, *Sobre...*, simplesmente apresenta uma tela contendo os principais dados do projeto LOBO e seus contribuidores. Esta tela se encontra representada na figura 83.

---

<sup>1</sup> ASCII – Acrônimo para *American Standard Code for Information Interchange*, é um conjunto de códigos para o computador representar números, letras e outros caracteres.

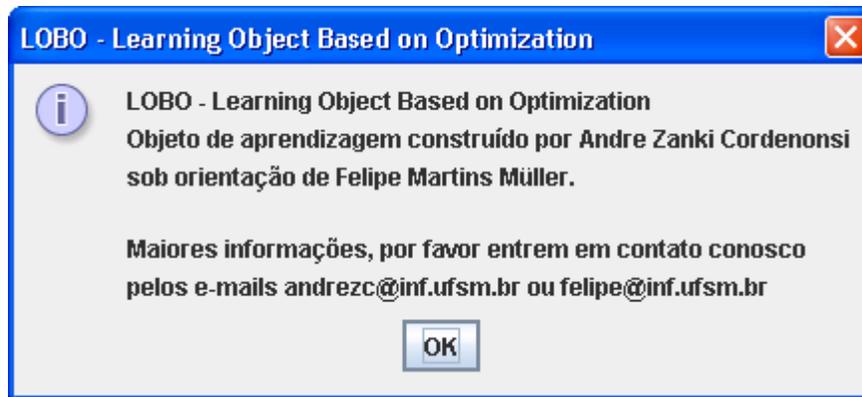


Figura 83 - Tela de visualização da opção Sobre

## 6.2 Algoritmos Implementados

Antes de uma explanação mais detalhada acerca dos objetos de ação, que apresentam as funcionalidades mais importantes do LOBO, é necessário conhecer os algoritmos que foram selecionados para implementação no objeto de aprendizagem. O seu conhecimento formal é necessário para a compreensão das técnicas de simulação utilizadas em cada caso, bem como as opções pedagógicas implementadas nas aulas que serão discutidas no próximo capítulo.

### 6.2.1 Heurística Construtiva: Algoritmo Aleatório

Este algoritmo, de simples definição e construção, foi implementado no objeto LOBO por se tratar da idéia mais simples possível para a resolução do Problema do Caixeiro Viajante. Tal escolha se justifica para exemplificar aos alunos que, em problemas de otimização combinatória, bons resultados só são conseguidos através de muito esforço. Soluções triviais obtêm resultados triviais; boas soluções são obtidas após uma análise cuidadosa e criteriosa do problema, estrutura e suas características.

A figura 84 apresenta a formulação desta heurística, que consiste, basicamente, na simples inserção de novos vértices à solução de forma aleatória, ou seja, não existe um critério para a inserção de um novo vértice. Todos os vértices concorrem para a inserção em uma determinada posição da solução com as mesmas chances.

*Entrada : um vértice  $v$  inicial*  
*Saída : um ciclo hamiltoniano  $s \leftarrow [v_1, \dots, v_n]$*   
*Seja  $\alpha \leftarrow [v_1, \dots, v_n]$  (cidades não incluídas na rota)*  
*Inicie o ciclo com apenas o vértice inicial  $v$*   
 $s \leftarrow [v]$   
 $\alpha = \alpha - [v]$   
*ENQUANTO  $\alpha \neq \phi$*   
     *Selecione aleatoriamente um vértice  $k \in \alpha$*   
     *Insira o vértice  $k$  em  $s$*   
          $s \leftarrow k$   
          $\alpha = \alpha - [k]$   
*FIM – ENQUANTO*

Figura 84 - Heurística Construtiva: Algoritmo Aleatório

### 6.2.2 Heurística construtiva: Vizinho Mais Próximo

A escolha da heurística do vizinho mais próximo para a implementação do LOBO foi decidida por três fatores fundamentais:

- simplicidade: o algoritmo é muito simples de entender e se baseia em um conceito genérico;
- produz resultados razoáveis: apesar de sua simplicidade e de apresentar dificuldades em problemas maiores, os resultados obtidos pela heurística são razoáveis como ponto de partida para outras soluções (algoritmos de melhoramento);
- a idéia é facilmente reconhecida pelos alunos: no processo das aulas, sem a apresentação da heurística, vários alunos tendem a construir, em formas gerais, uma heurística do vizinho mais próximo caso sejam instruídos a resolver o problema.

O algoritmo da figura 85 apresenta o formalismo da heurística do Vizinho Mais Próximo. Em resumo, esta heurística trabalha da seguinte forma: o ciclo inicia com uma cidade  $v_i$  atual e, a partir desta, encontra a próxima cidade  $v_k$  de forma que o custo (usualmente a distância) de ligação entre as duas cidades seja mínimo. Ao encontrar esta cidade  $v_k$ , o algoritmo insere a mesma no final do ciclo e repete a operação até que todas as cidades pertençam à solução. Finalmente, o ciclo é fechado ligando a cidade inicial à cidade final. Não é permitido visitar uma cidade duas vezes ou modificar uma escolha, ou seja, após uma cidade ser inserida em uma determinada posição da rota, ela não sofrerá nenhuma modificação de posicionamento.

*Entrada : um vértice  $v$  inicial*  
*Saída : um ciclo hamiltoniano  $s \leftarrow [v_1, \dots, v_n]$*   
*Seja  $\alpha \leftarrow [v_1, \dots, v_n]$  (cidades não incluídas na rota)*  
*Inicie o ciclo com apenas o vértice inicial  $v$*   
 $s \leftarrow v$   
 $\alpha = \alpha - [v]$   
 $ultima\_cidade = 1$   
**ENQUANTO**  $\alpha \neq \phi$   
 $v_i = s[ultima\_cidade]$   
 $v_j = \text{cidade de } \alpha \text{ de menor dist com } v_i$   
 $s \leftarrow v_j$   
 $\alpha = \alpha - [v_j]$   
 $ultima\_cidade = ultima\_cidade + 1$   
**FIM – ENQUANTO**  
*Feche o ciclo*  
 $s \leftarrow v$

Figura 85 - Heurística Construtiva: Vizinho Mais Próximo

A idéia que permeia esta heurística está no fato de buscar inserir na solução final os menores arcos, ou seja, as ligações entre duas cidades que apresentam o menor custo. Esta premissa é verdadeira para as primeiras ligações. No entanto, na medida em que o algoritmo prossegue, as escolhas diminuem e cresce a possibilidade de inserir arcos de custo elevado.

A figura 86 apresenta o resultado da aplicação desta heurística para o problema *pr76.tsp*. O valor da função objetivo para este problema foi de *153.461,92* em um tempo de 1.0 segundos. Vale lembrar que o valor ótimo para o problema em questão é de *108.159,43*.

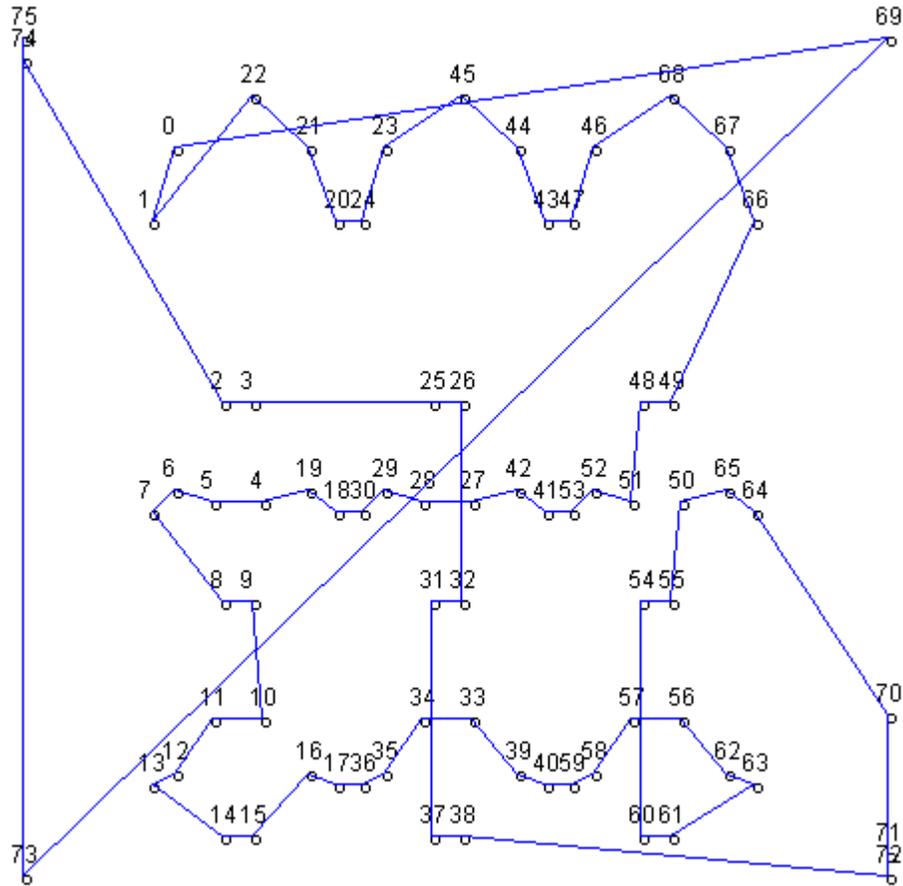


Figura 86 - Aplicação da Heurística Vizinho Mais Próximo sobre o problema pr76.tsp

### 6.2.3 Heurística construtiva: Inserção Mais Distante

A heurística da inserção mais distante contrapõe a do vizinho mais próximo por inserir cidades na solução final de forma a não gerar cruzamentos de rotas. O algoritmo inicia com uma sub-rotas que só contém a cidade de origem. Para cada cidade ( $l$ ) que não esteja na rota, é calculado a distância mínima entre  $l$  e todas as cidades que estejam na rota, e armazenado a *menor* distância, que chamaremos  $\min\_dist(l)$ . Depois, é escolhida a cidade ( $k$ ), tal que  $\min\_dist(k)$  seja máxima. Finalmente, o local de inserção de  $k$  na sub-rotas atual é dada pelo par de vértices  $i$  e  $j$ , ligando  $k$  na forma  $(i,k)$  e  $(k,j)$ , minimizando

$$c_{ik} + c_{kj} - c_{ij}$$

Desta forma, na sub-rotas atual são inseridos os arcos  $(i,k)$  e  $(k,j)$  e retirado o arco  $(i,j)$ . Finalmente, o ciclo reinicia, agora com o número de cidades fora da sub-rotas diminuído de um.

Existem várias variantes deste algoritmo, incluindo:

- inserção mais próxima: é escolhido a cidade( $k$ ), tal que  $\min\_dist(k)$  seja mínimo;
- inserção mais barata: para cada cidade( $k$ ) que não esteja na rota e o par de vértices  $(i,k)$  e  $(k,j)$  que ligam a cidade( $k$ ) a sub-rotas atual, encontrar o custo

dado pela fórmula anterior. É escolhido para inserção a cidade( $k$ ) cujo custo calculado anteriormente seja mínimo.

Estas duas variantes são apresentadas em (REI, 1994). O formalismo da heurística da inserção mais distante é apresentado na figura 87.

*Entrada* : um vértice  $v$  inicial  
*Saída* : um ciclo hamiltoniano  $s \leftarrow [v_1, \dots, v_n]$   
 Seja  $\alpha \leftarrow [v_1, \dots, v_n]$  (cidades não incluídas na rota)  
 Inicie o ciclo com os três primeiros vértices  $[v_1, v_2, v_3]$   
 $S \leftarrow [v_1, v_2, v_3]$   
 $\alpha \leftarrow \alpha - [v_1, v_2, v_3]$   
 $mais\_dist = 0$   
 $k = 0$  (vértice escolhido)  
 ENQUANTO  $\alpha \neq \phi$   
   *PARA* cada vértice  $V \in \alpha$  FAÇA  
      $d_v = \max(dist(V, s))$   
     SE  $d_v > mais\_dist$  ENTÃO  
        $mais\_dist = d_v$   
        $k = V$   
     FIM – SE  
 FIM – PARA  
 $menor\_custo = +\infty$   
 $V_i = 0; V_j = 0$  (vértices anterior e posterior)  
 PARA cada par de vértices  $(i, j) \in s$  FAÇA  
      $custo = c_{ik} + c_{kj} - c_{ij}$   
     SE  $custo < menor\_custo$  ENTÃO  
        $menor\_custo = custo$   
        $V_i = i; V_j = j$   
     FIM – SE  
 FIM – PARA  
 $s \leftarrow k$  (entre  $V_i$  e  $V_j$ )  
 $\alpha = \alpha - [k]$   
 FIM – ENQUANTO

Figura 87 - Heurística Construtiva: Inserção Mais Distante

A aplicação do algoritmo produz uma solução que é iniciada *pelos pontos*, como se fosse uma corda em círculo que é esticada primeiramente nas pontas e depois levada até os pontos interiores. Esta forma de atuação permite que o algoritmo não produza cruzamentos, se o problema respeitar a desigualdade do triângulo. Visualmente, é possível observar o desempenho no algoritmo na figura 88, que apresenta quatro estágios progressivos da aplicação da heurística para o problema *pr76.tsp*. O valor da função objetivo para este problema foi de 119.962,93 em um tempo de 1,0 segundo.

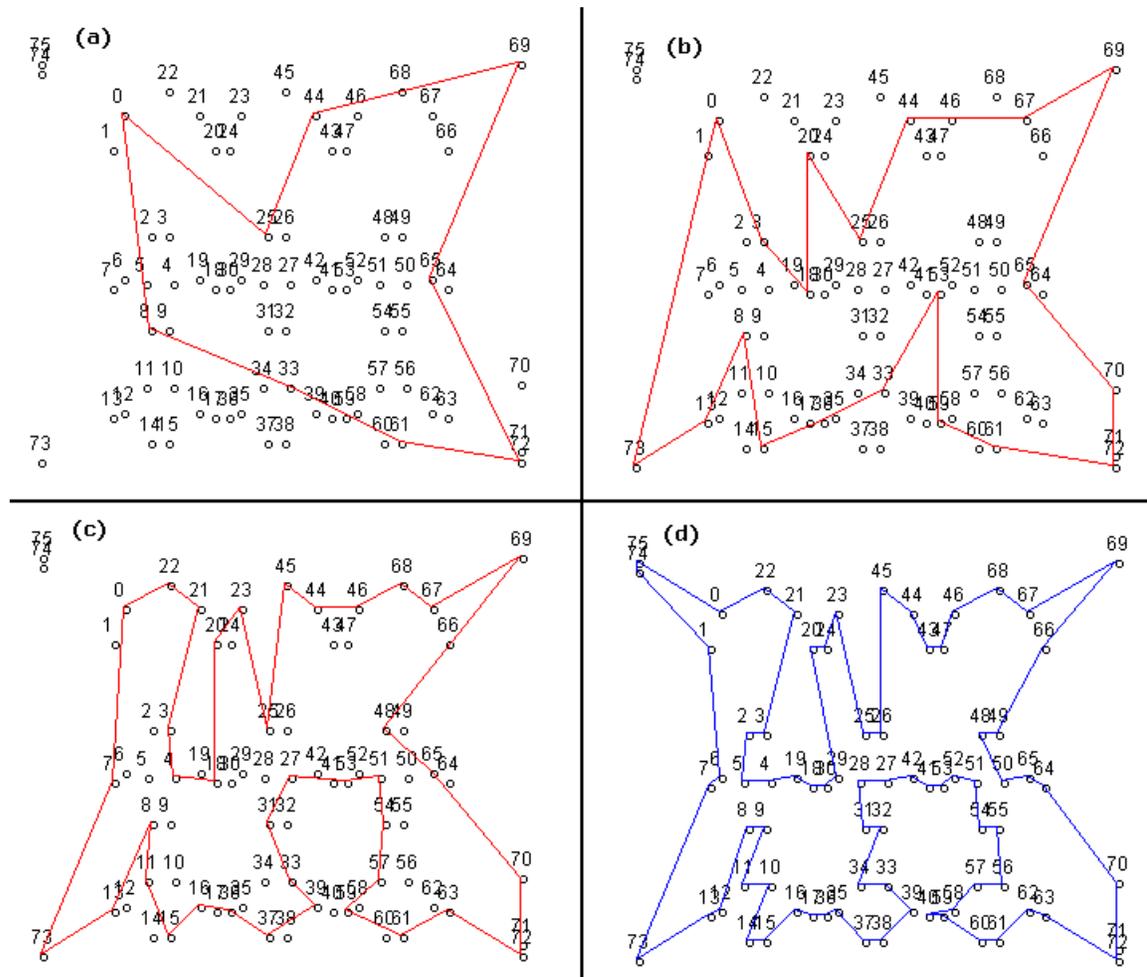


Figura 88 - Aplicação da Heurística Inserção Mais Distante sobre o problema *pr76.tsp*

#### 6.2.4 Heurísticas de melhoramento: 2-Opt e 3-Opt

As heurísticas de melhoramento 2-Opt e 3-Opt foram propostas por (CROES, 1958) e (LIN, 1965), respectivamente, e funcionam de maneira semelhante, sendo que a única diferença está no número de arcos envolvidos na análise (dois ou três). A idéia destas heurísticas de melhoramento é bastante simples e pode ser mais bem explicada pelo 2-Opt: elimine duas arestas da solução e insira novamente duas arestas de forma cruzada, ou seja, se as arestas removidas foram os pares que ligam as cidades  $(k_1, k_2)$  e  $(j_1, j_2)$ , as arestas

inseridas ligam as cidades na forma  $(k_1, j_2)$  e  $(j_1, k_2)$ . Se esta nova configuração for melhor que a anterior, ou seja, se a distância diminuir, mantenha a nova rota. Caso contrário, escolha novamente duas arestas para análise.

Para a heurística 3-Opt, o número de arcos envolvidos aumenta. São eliminados três arestas  $-(k_1, k_2), (j_1, j_2)$  e  $(i_1, i_2)$  – e são testados todas as combinações de ligações novas entre estas cidades, representadas na figura 89. As quatro novas rotas são testadas e, assim como no 2-Opt, caso alguma se mostre de custo inferior a rota atual, a troca é mantida.

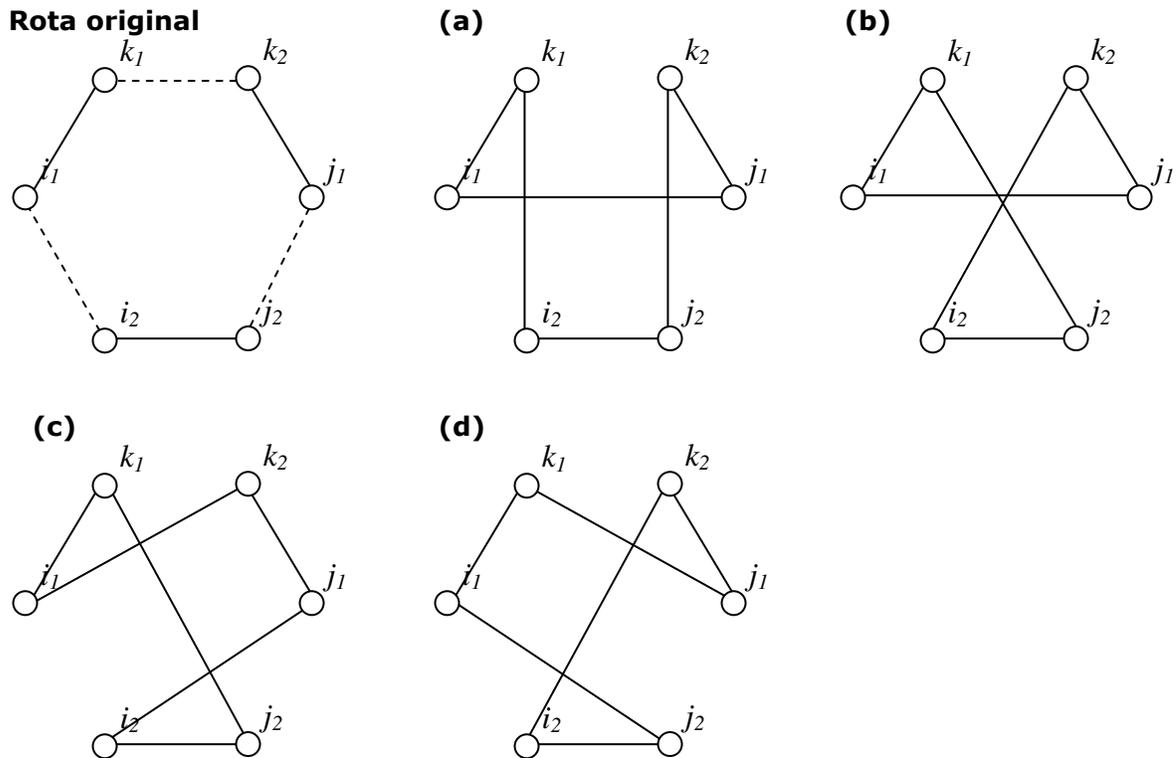


Figura 89 - Aplicação do algoritmo 3-Opt: rota original com os arcos que serão removidos e as novas rotas possíveis

A definição padrão dos algoritmos 2-Opt e 3-Opt utiliza o teste de todas as possibilidades de trocas entre dois ou três pares de arcos para quaisquer problemas que tenham, no mínimo, quatro ou seis vértices, respectivamente. Um ciclo completo de uma rodada 2-Opt ou 3-Opt verifica todas as trocas possíveis e realiza a *melhor* troca, ou seja, a troca que produza a rota de menor valor. A heurística, então, continua com esta nova rota, realizando novamente todos os testes para os movimentos possíveis. O algoritmo somente para quando uma rodada completa não encontrar nenhum melhoramento. O algoritmo da figura 90 apresenta a heurística 2-Opt no seu modo formal.

*Entrada* : um ciclo hamiltoniano  $s \leftarrow [v_1, \dots, v_n]$   
*Saída* : um ciclo hamiltoniano  $s^* \leftarrow [v_1, \dots, v_n]$   
*melhor\_troca\_a* = -1  
*melhor\_troca\_b* = -1  
*melhor\_troca\_custo* =  $-\infty$   
 PARA  $k = 1$  até  $\left\lceil \frac{\text{num\_cid}}{2} \right\rceil + \frac{\text{num\_cid}}{2}$  FAÇA  
     PARA  $i = 0$  até  $\text{num\_cid} - 1$  FAÇA  
          $a = i$   
          $b = \left\lceil \frac{i + k}{\text{num\_cid}} \right\rceil$   
         SE ( $\text{custo}(\text{anterior}(a), b) + \text{custo}(a, \text{proximo}(b))$ ) <  
             ( $\text{custo}(\text{anterior}(a), a) + (b, \text{proximo}(b))$ ) ENTÃO  
             SE ( $[\text{custo}(\text{anterior}(a), b) + \text{custo}(a, \text{proximo}(b))] > \text{melhor\_troca\_custo}$ ) ENTÃO  
                 *melhor\_troca\_a* =  $a$   
                 *melhor\_troca\_b* =  $b$   
                 *melhor\_troca\_custo* =  $\text{custo}(\text{anterior}(a), b) + \text{custo}(a, \text{proximo}(b))$   
             FIM - SE  
         FIM - SE  
     FIM - PARA  
 FIM - PARA  
 Insira arco( $\text{anterior}(\text{melhor\_troca\_a}), \text{melhor\_troca\_b}$ )  
 Insira arco( $\text{melhor\_troca\_a}, \text{proximo}(\text{melhor\_troca\_b})$ )  
 Remova arco( $\text{anterior}(\text{melhor\_troca\_a}), \text{melhor\_troca\_a}$ )  
 Remova arco( $\text{melhor\_troca\_b}, \text{proximo}(\text{melhor\_troca\_b})$ )

Figura 90 - Heurística de Melhoria: 2-Opt

No entanto, há duas modificações que são possíveis de realizar nos algoritmos, detalhadas a seguir:

- busca pela *primeira* melhor solução: nesta variante, as heurísticas finalizam a rodada ao encontrar uma melhor solução que a atual, independente se, por ventura, existirem outras soluções melhores. A execução do algoritmo se torna mais rápido mas é possível que algumas boas soluções não sejam encontradas;
- vizinhança: a aplicação da vizinhança nas heurísticas 2-Opt e 3-Opt é bastante comum na literatura. A idéia principal da utilização da vizinhança é reduzir o espaço de busca, diminuindo as trocas possíveis para um determinado número de arcos nas cercanias do vértice analisado, impedindo a análise de trocas entre arcos longes demais. A prática da utilização da vizinhança indica que os

resultados permanecem usualmente semelhantes para um tempo computacional bastante reduzido.

O LOBO implementa as duas variantes para as heurísticas 2-Opt e 3-Opt. Antes da execução do algoritmo, o usuário deve responder se deseja usar o esquema de vizinhança, fornecendo também o número de vértices vizinhos que deseja analisar; e se o algoritmo deve procurar pela melhor troca possível ou pela primeira troca aprimorante.

A figura 91 apresenta a aplicação da heurística 2-Opt para o problema *pr76.tsp* após a construção de uma rota inicial através da heurística do Vizinheiro Mais Próximo. A *rota original* apresenta a solução antes da aplicação da heurística de melhoramento. A *escolha dos arcos* representa graficamente qual foi o par de arcos escolhidos para a remoção (25,69)(49,66), pois sua substituição geraria a melhor solução resultante. A *remoção dos arcos* representa a rota sem os dois arcos escolhidos e a *nova rota* apresenta a nova solução atual, com a inserção dos dois novos arcos (25,49)(66,69).

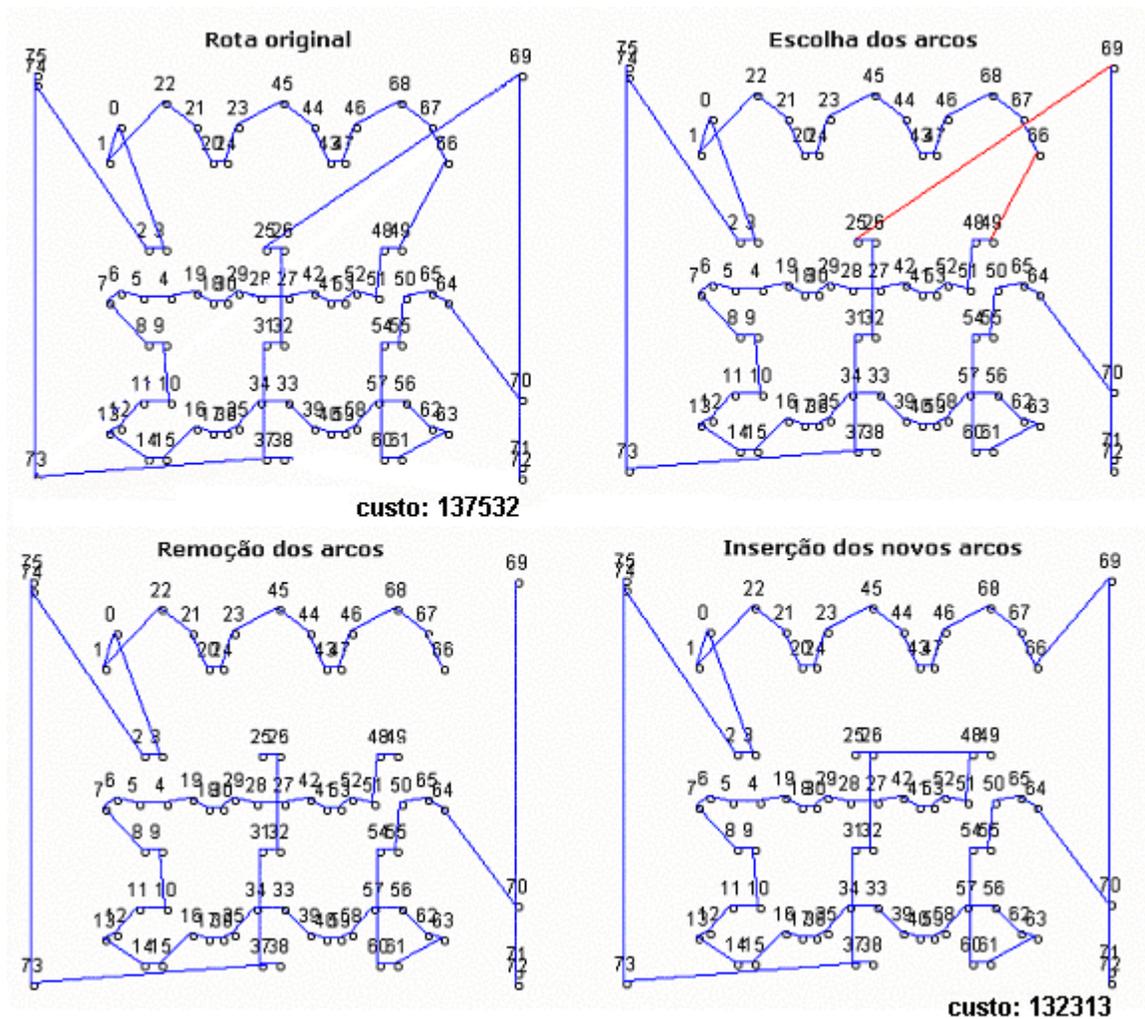


Figura 91 - Aplicação de uma rodada da Heurística 2-Opt sobre o problema *pr76.tsp*

### 6.2.5 Metaheurística: Busca Tabu

A metaheurística busca tabu (*tabu search*) é um procedimento utilizado para guiar um algoritmo heurístico de busca local (melhoramento) para explorar o espaço de soluções além dos ótimos locais (DIAZ *et al.*, 1996). A principal premissa para a busca tabu está na incorporação de uma memória adaptativa e da exploração reativa do espaço de soluções.

Esta metaheurística foi proposta por (GLOVER, 1986). Como princípio geral, a Busca Tabu é um guia para a realização da busca local (através, por exemplo, de um algoritmo de melhoramento como o 2-Opt). Sua principal característica está na utilização de uma memória adaptativa para criar uma busca mais flexível. Existem dois tipos principais de memória em uma busca tabu: memória de curto prazo e de longo prazo.

A memória de curto prazo é utilizada para classificar como proibido alguns movimentos ou soluções, através de seus atributos, armazenando-os na lista tabu durante um certo tempo. Este tempo pode ser fixado previamente, sorteado entre um mínimo ou máximo ou utilizar outros critérios. Também é possível definir um critério de aspiração. Este critério é utilizado para liberar um movimento de seu *estado* tabu antes do tempo definido. Este critério pode ser utilizado, por exemplo, para liberar um movimento tabu que leve a uma solução melhor que a atual ou se todos os movimentos possíveis estejam tabu.

A memória de longo prazo é utilizada em aplicações cujos componentes da memória de curto prazo não se mostraram suficientes para produzir soluções de alta qualidade. Dentre as práticas de construção de memórias de longo prazo, é possível citar as estratégias de intensificação (modificação reativa das regras de aceitação de um movimento para favorecer trocas cujas características de solução tem se mostrado historicamente melhores) e diversificação (modificação reativa das regras de aceitação para explorar regiões do espaço de busca que não tenham sido utilizadas frequentemente) (DIAZ *et al.*, 1996).

Existem várias formas de implementar uma busca tabu. A discussão que se segue demonstra como foi desenvolvida a metaheurística para a implementação do LOBO. O formalismo é dado pela figura 92. Ao executar o algoritmo pelo objeto de aprendizagem, o sistema pergunta ao usuário o tempo (em segundos) que o algoritmo deve rodar. Este tempo é utilizado em um laço externo até que o mesmo seja atingido. A função *atualiza* realiza o sincronismo do tempo real (dado pelo sistema computacional) e o tempo, em segundos, dado pelo usuário.

Inicialmente, é gerada uma lista de todas as trocas possíveis (movimentos do problema) através da heurística de melhoramento 2-opt. A geração de movimentos pode ser configurada através da interface do LOBO, que requisita ao usuário o tamanho da vizinhança que deve ser utilizada pela heurística. Esta lista de trocas é, então, ordenada de forma decrescente em relação a diminuição do custo da rota, ou seja, primeiramente são apresentadas as trocas que apresentam a melhoria mais acentuada.

A troca escolhida será a que estiver no topo da lista, caso esta não seja tabu. Para a troca do topo, é verificado, primeiramente, se ela produz uma solução melhor que a incumbente. Em caso positivo, a mesma é selecionada e o laço da escolha da troca é finalizado. Caso contrário, é realizado um teste para verificar se a troca escolhida é tabu. Em caso positivo, o laço reinicia e a próxima troca da lista é analisada. Caso contrário, a troca é aceita.

Após a escolha da troca, a solução atual é atualizada, assim como a incumbente, caso a solução atual seja melhor. Finalmente, a lista tabu decrementa o número de iterações para todos os seus elementos no valor de um e a troca escolhida é inserida na lista tabu com o número de iterações definidos pelo usuário. Nesta estratégia, só foi utilizada a memória de curto prazo.

```

Entrada : um ciclo hamiltoniano [v1.....vn]
Entrada : tempo _ execucao(s)
Saída : um ciclo hamiltoniano [v1.....vn]
solucao _ incumbente = Entrada
atualiza(tempo _ execucao,tempo _ real)
ENQUANTO tempo _ execucao > 0
  FAÇA
    i = 0
    lista _ trocas = geraPossibilidades2Opt()
    ordena(lista _ trocas)
    troca _ escolhida = lista _ trocas(0)
    SE custo(troca _ escolhida) < custo(solucão _ incumbente)
      SAIA DO FAÇA
    FIM – SE
    i = i + 1
  ENQUANTO lista _ tabu(troca _ escolhida) = VERDADEIRO
    solucao _ atual = realiza _ troca(troca _ escolhida)
    SE custo(solucão _ atual) < custo(solucão _ incumbente)
      solucao _ incumbente = solucao _ atual
    FIM – SE
    lista _ tabu.decrementa()
    lista _ tabu.adiciona(troca _ escolhida,numero _ iteracoes)
    atualiza(tempo _ execucao,tempo _ real)
  FIM – ENQUANTO

```

Figura 92 - Metaheurística: Busca Tabu

### 6.2.6 Metaheurística: *Simulated Annealing*

A metaheurística *Simulated Annealing* foi proposta pela primeira vez em (KIRKPATRICK *et al.*, 1983). Ela se baseia no processo físico conhecido como *annealing*, que se caracteriza por ser um processo térmico de resfriamento de um material que se encontra a uma alta temperatura através de uma lenta e gradual diminuição de sua

temperatura. A cada configuração atingida, o material apresenta um nível de energia que, ao final do processo, deve apresentar uma configuração de energia mínima. A maneira pela qual a temperatura irá decrescer é a parte mais importante do processo. Um decréscimo rápido demais poderá levar a imperfeições significativas no material, ou seja, este não alcançara a configuração de energia mínima.

Para os autores (KIRKPATRICK *et al.*, 1983), é possível realizar uma adaptação do processo físico para a busca de soluções factíveis em um espaço de busca com o objetivo de encontrar a melhor solução possível. Para tanto é necessário realizar as seguintes analogias: configuração = solução; configuração de energia mínima = solução ótima; nível de energia = função objetivo; e temperatura = parâmetro de controle.

Basicamente, no decorrer da aplicação da metaheurística *simulated annealing*, são permitidos movimentos que aumentem o valor da função objetivo, ou seja, que piorem a solução atual. No entanto, a frequência com que isso pode ocorrer é definida por uma função probabilística que se altera no decorrer do funcionamento do algoritmo. Existem várias funções que podem ser utilizadas para a estratégia *simulated annealing*, mas a mais comum é a probabilística baseada no processo físico, conhecida como distribuição de Boltzmann-Gibbs.

$$p = Ke^{\frac{-\Delta E}{T}}$$

onde  $\Delta E$  representa a diferença entre o custo da solução atual e a da que está sendo analisada,  $T$  é a temperatura atual e  $K$  uma constante. A estratégia da utilização desta função está na idéia de que, no início da aplicação do algoritmo, este se encontra em uma *alta temperatura*, permitindo um grande número de alterações ruins. Posteriormente, a temperatura sofrerá um decréscimo e a possibilidade de alterações ruins vai diminuindo até estabilizar no ótimo local.

Os parâmetros para o *simulated annealing* são:

- $T_0$ : temperatura inicial;
- $T_f$ : temperatura final;
- $L$ : número de iterações necessárias para atingir um equilíbrio em uma determinada temperatura;
- $\alpha$ : proporção da redução da temperatura.

Quanto mais alta for a temperatura inicial ( $T_0$ ), maiores as chances de movimentos ruins serem aceitos por um maior tempo, como é possível observar no gráfico representado pela figura 93, que apresenta a função de probabilidade para temperaturas iniciais variando de 0.5 a 20 e diferenças entre as duas soluções analisadas entre 1 e 9.

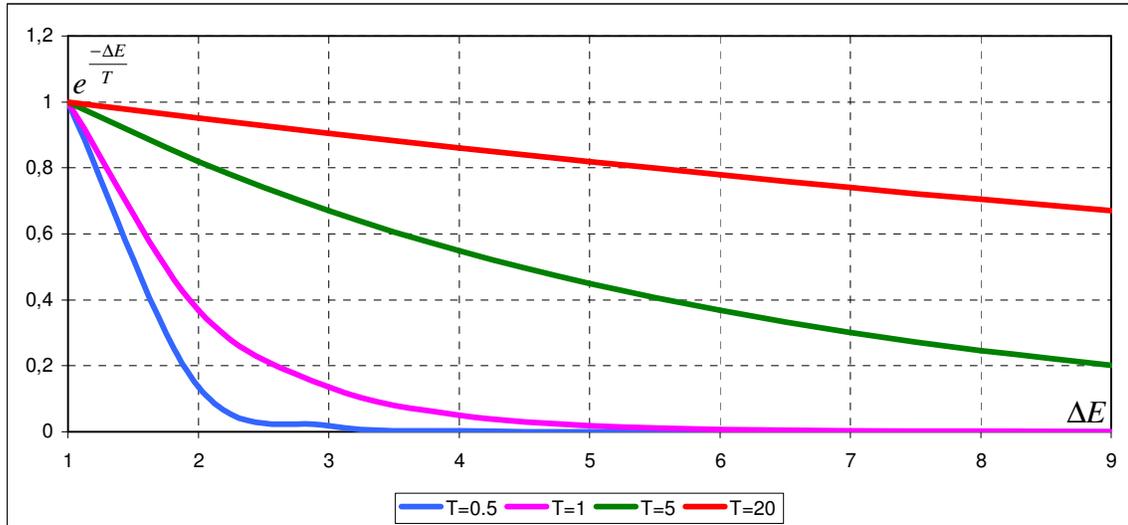


Figura 93 - Comportamento da função probabilística da metaheurística *simulated annealing* para diferentes graus de temperatura inicial e diferença entre soluções analisadas

A metaheurística inicia com uma temperatura inicial  $T_0$  fornecida pelo usuário. Até que ocorra o *equilíbrio energético*, ou seja, até um determinado número de passos fornecido pelo usuário através do parâmetro  $L$ , o algoritmo deve gerar uma nova solução a partir da corrente, calculando o  $\Delta E$ . Se o custo for negativo, ou seja, houve uma melhoria na solução, então a solução atual deve ser atualizada. Caso contrário, é utilizada a função probabilística para verificar se, mesmo sendo um movimento não aprimorante, ele deve ser aceito. Em caso positivo, a solução atual é atualizada. Caso contrário, o algoritmo segue até atingir o número de passos  $L$ .

Neste ponto, a metaheurística diminui a temperatura através do parâmetro  $\alpha$  e o processo reinicia até que a temperatura atual seja igual ou menor a temperatura final definida pelo usuário. A definição formal da metaheurística *simulated annealing* pode ser observada no algoritmo descrito na figura 94.

*Entrada* : um ciclo hamiltoniano  $[v_1, \dots, v_n]$   
*Saída* : um ciclo hamiltoniano  $[v_1, \dots, v_n]$   
*solucao\_incumbente* = *Entrada*  
*solucao\_atual* = *Entrada*  
 $T_o, T_f, L, \alpha$  = fornecidos pelo usuário  
 $T = T_o$   
**ENQUANTO**  $T > T_f$  **FAÇA**  
     **PARA**  $cont = 1$  **ATÉ**  $L$  **FAÇA**  
         *solucao* = *geraMovimento2Opt(solucão\_atual)*  
          $\Delta E = custo(solucão) - custo(solucão\_atual)$   
         **SE**  $\Delta E < 0$  **ENTÃO**  
             *solucao\_atual* = *solucao*  
         **FIM** – **SE**  
         **SENÃO**  
             **SE**  $U[0,1] < \exp^{\frac{-\Delta E}{T}}$   
                 *solucao\_atual* = *solucao*  
             **FIM** – **SE**  
         **SE**  $custo(solucão\_atual) < custo(solucão\_incumbente)$  **ENTÃO**  
             *solucao\_incumbente* = *solucao\_atual*  
         **FIM** – **SE**  
     **FIM** – **PARA**  
      $T = T * \alpha$   
     **FIM** – **ENQUANTO**  
     *Saída* = *solucao\_incumbente*

Figura 94 - Metaheurística: *simulated annealing*

### 6.3 Histórico dos Resultados

O histórico dos resultados é uma região do LOBO onde é apresentado ao usuário um resumo da utilização do mesmo. A cada nova troca dos dados do problema, seja pela simples escolha de um novo conjunto ou pela inserção de dados, ou ainda pela abertura de um arquivo no formato TSPLib, o histórico é atualizado. Mas sua principal contribuição está na manutenção dos resultados provenientes da aplicação dos algoritmos sucessivamente sobre um problema. Desta forma, é possível para os alunos e professores perceberem se existem melhorias e como as diferentes heurísticas podem trazer resultados diferentes através da simples troca na ordem dos algoritmos. Pedagogicamente, o professor pode sugerir seqüências de algoritmos ou ainda pedir que os alunos encontrem a melhor seqüência ou os melhores parâmetros para as heurísticas e metaheurísticas implementadas.

A figura 95 apresenta um exemplo de histórico de resultados para uma seqüência aplicada sobre o problema *kroa100.tsp*.

```
Histórico de resultados:

Dados: berlin52.tsp [TSPLIB]

Troca de dados: kroA100.tsp

==>Construtivo: Vizinho mais próximo
Custo: 26856.388591241608
Tempo Decorrido: 0.0 milisegundos.

==>Melhoramento: Rodada completa 2OPT
Parâmetros:
Execução até encontrar a melhor troca.
Execução com todas as possibilidades de troca.
Custo: 21921.4318352778
Número de rodadas 2-opt: 21
Tempo Decorrido: 1102.0 milisegundos.

==>Metaheurística: Simulated Annealing
Parâmetros:
Temperatura inicial: 100.
Temperatura final: 0.1.
Grau de resfriamento: 0.95.
Número de iterações para estabilização: 20.
Custo: 21905.496447763864
Tempo Decorrido: 6559.0 milisegundos.
```

Figura 95 - Histórico de resultados para uma seqüência de aplicação de heurísticas sobre o problema *kroA100.tsp*

Com a manutenção deste histórico, é possível perceber a interação entre os diversos algoritmos. Um exemplo de utilização do histórico de resultados pedagogicamente está na observação do custo e do tempo computacional exigido por cada algoritmo sob um determinado problema. Considere, por exemplo, que o professor peça para os alunos executar uma seqüência de aplicações de heurísticas, onde o aluno deverá observar o resultado das heurísticas de melhoramento 2-Opt e 3-Opt sob diferentes heurísticas construtivas. A figura 96 apresenta a seqüência de aplicações das duas heurísticas de melhoramento, consecutivamente, até que nenhuma melhoria seja possível. No caso específico, o algoritmo construtivo escolhido foi o aleatório. Convém lembrar que as informações em negrito foram sublinhadas posteriormente e o tempo total não consta da ferramenta.

```

Dados: berlin52.tsp [TSPLIB]

==>Construtivo: Aleatório
Custo: 27711.21684024698
Tempo Decorrido: 0.0 milisegundos.

==>Melhoramento: Rodada completa 2OPT
Parâmetros:
Execução até encontrar a melhor troca.
Execução com todas as possibilidades de troca.
Custo: 8206.596492994753
Número de rodadas 2-opt: 45
Tempo Decorrido: 641.0 milisegundos.

==>Melhoramento: Rodada completa 3OPT
Parâmetros:
Execução até encontrar a melhor troca.
Execução com todas as possibilidades de troca.
Custo: 7805.202030569165
Número de rodadas 3-opt: 15
Tempo Decorrido: 120.0 milisegundos.

==>Melhoramento: Rodada completa 2OPT
Parâmetros:
Execução até encontrar a melhor troca.
Execução com todas as possibilidades de troca.
Custo: 7598.442340904543
Número de rodadas 2-opt: 5
Tempo Decorrido: 100.0 milisegundos.

==>Melhoramento: Rodada completa 3OPT
Parâmetros:
Execução até encontrar a melhor troca.
Execução com todas as possibilidades de troca.
Custo: 7598.442340904543
Número de rodadas 3-opt: 0
Tempo Decorrido: 20.0 milisegundos.

TEMPO TOTAL: 881 milisegundos

```

Figura 96 - Aplicação das heurísticas de melhoramento 2-Opt e 3-Opt sob um problema cuja solução inicial é aleatória

Para que seja possível realizar as comparações necessárias, o professor também deve pedir que o aluno realize os mesmo testes, mas para algoritmos construtivos diferentes. A figura 97 apresenta a mesma seqüência manual para o algoritmo construtivo do Vizinho Mais Próximo e a figura 98 apresenta a seqüência realizada sob o algoritmo construtivo da Inserção Mais Distante.

```

Dados: berlin52.tsp [TSPLIB]

==>Construtivo: Vizinho mais próximo
Custo: 8980.918279329191
Tempo Decorrido: 10.0 milisegundos.

==>Melhoramento: Rodada completa 2OPT

```

```

Parâmetros:
Execução até encontrar a melhor troca.
Execução com todas as possibilidades de troca.
Custo: 7841.387787835102
Número de rodadas 2-opt: 11
Tempo Decorrido: 231.0 milisegundos.

==>Melhoramento: Rodada completa 3OPT
Parâmetros:
Execução até encontrar a melhor troca.
Execução com todas as possibilidades de troca.
Custo: 7799.51014565324
Número de rodadas 3-opt: 1
Tempo Decorrido: 30.0 milisegundos.

==>Melhoramento: Rodada completa 2OPT
Parâmetros:
Execução até encontrar a melhor troca.
Execução com todas as possibilidades de troca.
Custo: 7799.51014565324
Número de rodadas 2-opt: 0
Tempo Decorrido: 20.0 milisegundos.

TEMPO TOTAL: 291 milisegundos

```

Figura 97 - Aplicação das heurísticas de melhoramento 2-Opt e 3-Opt sob um problema cuja solução inicial é dada pelo Vizinho Mais Próximo

```

Dados: berlin52.tsp [TSPLIB]

==>Construtivo: Inserção Mais Distante
Custo: 8308.602406210206
Tempo Decorrido: 0.0 milisegundos.

==>Melhoramento: Rodada completa 2OPT
Parâmetros:
Execução até encontrar a melhor troca.
Execução com todas as possibilidades de troca.
Custo: 8196.158304695628
Número de rodadas 2-opt: 2
Tempo Decorrido: 60.0 milisegundos.

==>Melhoramento: Rodada completa 3OPT
Parâmetros:
Execução até encontrar a melhor troca.
Execução com todas as possibilidades de troca.
Custo: 8022.391808199605
Número de rodadas 3-opt: 3
Tempo Decorrido: 50.0 milisegundos.

==>Melhoramento: Rodada completa 2OPT
Parâmetros:
Execução até encontrar a melhor troca.
Execução com todas as possibilidades de troca.
Custo: 8022.391808199605
Número de rodadas 2-opt: 0
Tempo Decorrido: 20.0 milisegundos.

TEMPO TOTAL: 130 milisegundos

```

Figura 98 - Aplicação das heurísticas de melhoramento 2-Opt e 3-Opt sob um problema cuja solução inicial é dada pela Inserção Mais Distante

Várias considerações podem ser realizadas sob o experimento anterior. A primeira, e mais óbvia, é que a aplicação de um algoritmo construtivo aleatório não inviabiliza a busca por uma boa solução. No caso específico, o custo final da solução iniciada por este

algoritmo foi o *melhor* (7598,44). Mas este é apenas uma das facetas do problema. Outro aspecto que deve ser considerado para a implementação de algoritmos heurísticos está centrado no seu tempo. O compromisso entre a qualidade da solução e o esforço computacional necessário para encontrá-lo é uma questão, pedagógica e cientificamente, muito relevante. A seqüência iniciada pelo algoritmo construtivo aleatório gastou 881 milissegundos para encontrar uma solução, um tempo muito superior aos 291 milissegundos da seqüência do vizinho mais próximo e dos 130 milissegundos da seqüência da inserção mais distante.

Um aspecto importante do algoritmo construtivo aleatório é que o mesmo tem características probabilísticas, sendo que, neste caso, todas as cidades ainda não inseridas na solução têm o mesmo grau de probabilidade de se serem inseridas na posição  $i$ . Uma seqüência iniciada de forma aleatória pode nunca mais se repetir. Logo, é possível encontrar soluções muito boas ou muito ruins com este tipo de algoritmo. Para o problema apresentado nesta análise, o *berlin52.tsp*, uma seqüência iniciada aleatoriamente encontrou, inclusive, o resultado ótimo para o problema. Novamente, não há nenhuma garantia que, se repetidos o experimento mais  $n$  vezes, alguma vez o resultado ótimo seja encontrado novamente, não importa o tamanho de  $n$ . Isso pode ser explicado pelo comportamento do espaço de soluções. As duas soluções construtivas (inserção mais distante e vizinho mais próximo) sempre produzem os mesmos resultados para os mesmos dados de entrada. No caso do algoritmo aleatório, é possível que o mesmo percorra todo o espaço de busca, apresentando uma configuração de cada vez. Através de um *golpe de sorte*, o algoritmo aleatório apresentou uma solução que, apesar de ser muito pior que as demais soluções construtivas (aleatória  $\approx 27771$ , vizinho mais próximo  $\approx 8980$ , inserção mais distante  $\approx 8308$ ), a aplicação das heurísticas de melhoramento 2-Opt e 3-Opt alcançaram a melhor solução possível (ótimo global). Ainda, mesmo que a solução inicial da inserção mais distante seja melhor, o vizinho mais próximo em seqüência com as heurísticas de melhoramento 2-Opt e 3-Opt apresentou um resultado melhor.

A discussão sobre o desempenho dos algoritmos e seu tempo computacional pode ser implementada pelo professor com seus alunos através da construção de uma grade, como a apresentada na tabela 3. A tabela apresenta seis problemas, suas respectivas cidades, o algoritmo utilizado para chegar ao resultado (VMP: Vizinho Mais Próximo com posterior execução dos algoritmos 2-Opt e 3-Opt até não houver mais melhorias – IMD: Inserção Mais Distante e ALE: Algoritmo Construtivo Aleatório, ambos com a execução dos algoritmos de melhoramento 2-Opt e 3-Opt), custo final (custo da solução depois da execução dos algoritmos de melhoramento), tempo total (tempo somado entre a execução do algoritmo construtivo e todas as execuções dos algoritmos de melhoramento), custo ótimo do problema e uma percentagem do custo final em relação ao custo ótimo do problema. Para o algoritmo construtivo aleatório, é apresentado tanto no custo final quanto no tempo total a média de três execuções. Se, por acaso, em alguma das três execuções foi encontrado um valor final menor que os algoritmos anteriores (VMP e IMD), este custo também é apresentado, diferenciado por um asterisco.

Problema	n. cid.	Algoritmo	Custo Final	Tempo Total	Custo Ótimo	% Ótimo
eil51	51	VMP	431.91	270.00	426	98.61%
eil51	51	IMD	453.75	140.00	426	93.45%
eil51	51	ALE	434.03	644.33	426	98.11%

pr76	76	VMP	110887.08	1041.00	108159	97.47%
pr76	76	IMD	109666.07	831.00	108159	98,60%
pr76	76	ALE	111097.77 *108879.74	2242.66	108159	97.28%
kroA100	100	VMP	21710.49	1773.00	21282	97.98%
kroA100	100	IMD	21601.19	1310.00	21282	98.50%
kroA100	100	ALE	22326.26	5504.66	21282	95.09%
lin105	105	VMP	14620.97	1852.00	14379	98.31%
lin105	105	IMD	14576.45	1294.00	14379	98.62%
lin105	105	ALE	15027.05	6940.00	14379	95.49%
kroA200	200	VMP	29575.32	10977.00	29368	99.30%
kroA200	200	IMD	30574.67	7680.00	29368	95.89%
kroA200	200	ALE	30626.06	49991.33	29368	95.71%
pr299	299	VMP	50558.86	34620.00	48191	95.08%
pr299	299	IMD	50652.43	31546.00	48191	94.89%
pr299	299	ALE	50962.36 *50500.77	237458.66	48191	94.25%

Tabela 3 - Qualidade das soluções x Tempo computacional

Montando esta tabela, os alunos são levados a pensar sobre diversas questões. Primeiramente, é bastante óbvio perceber que o custo computacional em relação ao tempo cresce de forma expressiva em relação direta ao tamanho do problema. Além disso, a diferença entre o menor tempo (IMD, para todos os casos) e o pior tempo (ALE, para todos os casos), também cresce:

eil51	460%
pr76	269%
kroA100	420%
lin105	536%
kroA200	650%
pr299	752%

Porque isso ocorre? Analisando o histórico dos resultados, é possível perceber que o algoritmo construtivo aleatório gera um grande número de cruzamentos que precisam ser desfeitos pelo algoritmo 2-Opt. Para o algoritmo VMP, o número de cruzamentos é bem menor e, no algoritmo IMD, não há cruzamentos para serem desfeitos. Como forma de comparação, para o problema pr299, foram realizadas 51 trocas 2-Opt utilizando o algoritmo VMP, 16 trocas utilizando o IMD e 358 trocas com o algoritmo ALE. A necessidade de explorar mais amplamente o espaço de soluções acaba gerando problemas em relação ao custo computacional.

Outro ponto importante que pode ser obtido na tabela é que a qualidade da solução, para as três seqüências, decai em relação ao tamanho do problema. Este ponto também deve ser discutido com os alunos, lembrando-lhes que problemas maiores são mais difíceis de resolver e, desta forma, exigem uma aplicação de algoritmos mais sofisticados, como as metaheurísticas. Desta forma, os alunos conseguem perceber na prática os problemas que só podiam ser relatados pelos professores de forma oral.

#### 6.4 Botões de ação

Existem quatro botões de ação no LOBO, que concentram grande parte da funcionalidade da simulação do objeto de aprendizagem. O usuário realmente executa suas decisões através dos botões de ação, enquanto que o menu principal e os seus sub-menus atuam como uma forma do usuário informar os dados necessários para a simulação. Os quatro botões de ação desenvolvidos no objeto são: *executar a heurística*, *simulação passo-a-passo*, *exibir a rota ótima* e *tutorial do LOBO*.

#### 6.4.1 Botão: Executar a Heurística



O botão *executar a heurística* apresenta o resultado final da aplicação da heurística selecionada no sub-menu *Algoritmos* sobre o problema escolhido através do sub-menu *Dados de teste*. Quando o usuário executa o LOBO, o problema padrão apresentado é o *berlin52.tsp* e a heurística selecionada é a construtiva aleatória.

Como em todo o objeto, há dois modos de execução de uma heurística: o modo tutoriado e o *expert*. No modo tutoriado, ao pressionar o botão em questão, uma tela explicativa é apresentada ao usuário, avisando o mesmo que o LOBO está pronto para executar o algoritmo escolhido e a forma de apresentação dos dados após a finalização do mesmo. Esta tela, para a heurística construtiva aleatória, é apresentada na figura 99.

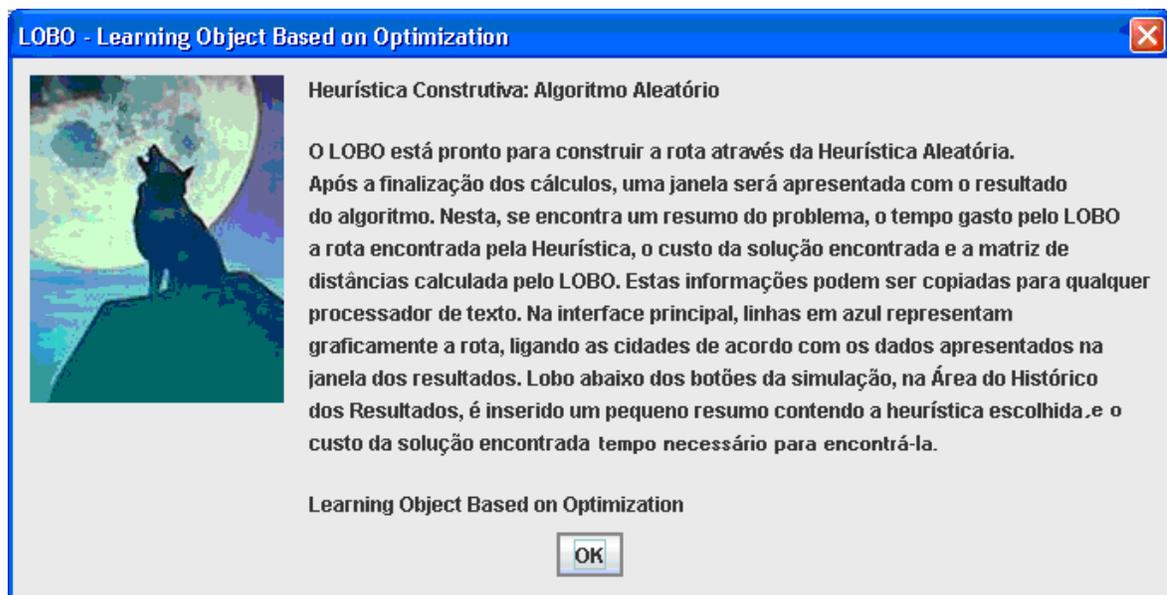


Figura 99 - Tela de apresentação para execução do algoritmo construtivo aleatório

Assim que o usuário pressionar o botão *OK* para fechar a tela, o objeto de aprendizagem executa o algoritmo sobre os dados selecionados. Após a realização dos cálculos, uma janela de resultados é apresentada ao usuário. Nesta janela, que pode permanecer aberta indefinidamente, o usuário tem acesso as informações acerca de qual conjunto de dados foi escolhido, a heurística que foi executada, o tempo gasto pelo algoritmo, a rota encontrada, o custo da solução encontrada e a matriz de distâncias para o problema selecionado. Estes dados, em formato ASCII, podem ser copiados para qualquer tipo de processador de texto e compilados mais tarde na forma de tabelas e/ou gráficos. A

figura 100 apresenta partes da interface gráfica da janela dos resultados para o algoritmo construtivo aleatório aplicado sobre o problema *berlin52.tsp*.

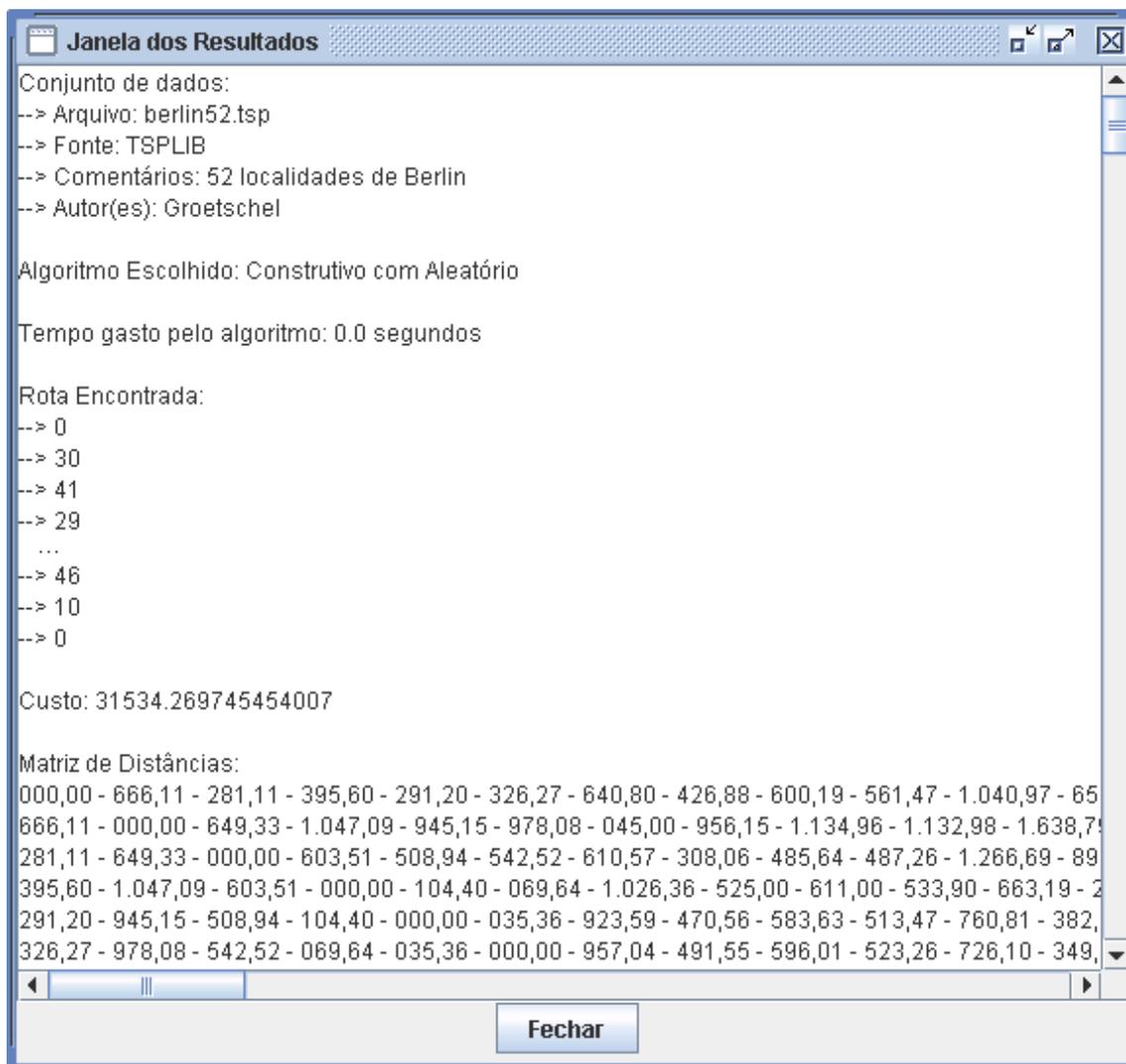


Figura 100 - Janela dos resultados para o algoritmo construtivo aleatório sobre o problema *berlin52.tsp*

Além da janela de resultados, a ação do botão *executar a heurística* também insere o resumo no *histórico de resultados*, conforme explicado na seção 6.3. Para os algoritmos construtivos e os algoritmos de melhoramento, o funcionamento é idêntico. No entanto, a execução das metaheurísticas busca tabu e *simulated annealing* apresenta características próprias que podem ser visualizadas através de outras ferramentas de análise. Para o objeto LOBO, foi definido que estas metaheurísticas gerariam gráficos explicativos que apresentassem a performance do algoritmo no decorrer do tempo. Desta forma, além da janela de resultados, o objeto de aprendizagem apresenta uma segunda janela, contendo os gráficos da execução.

Para a metaheurística *simulated annealing*, o gráfico apresentado representa o custo da solução  $x$  temperatura atual, como apresentado na figura 101. Este gráfico contém duas

linhas de plotagem. A primeira representa a solução incumbente e seu valor só é modificado quando é encontrado uma solução melhor que a melhor solução possível até o momento (incumbente). Pelo gráfico da figura 101, é possível observar que só houve melhoria quando a temperatura estava próxima do menor valor, ou seja, somente para movimentos muito bons (que foram aceitos quando a temperatura já estava baixa, conforme o algoritmo apresentado na seção 6.2.6). A segunda linha de plotagem representa uma média das soluções encontradas para todos os movimentos testados em uma determinada temperatura, parâmetro este definido pelo variável L. Como era de se esperar, a média das soluções encontradas tende a ser muito ruim durante grande parte da execução, principalmente porque em temperaturas mais altas, a probabilidade gerar movimentos não aprimorantes é muito alta. Conforme a temperatura decresce, forma-se uma convergência entre a média das soluções e a solução incumbente, o que também era esperado.

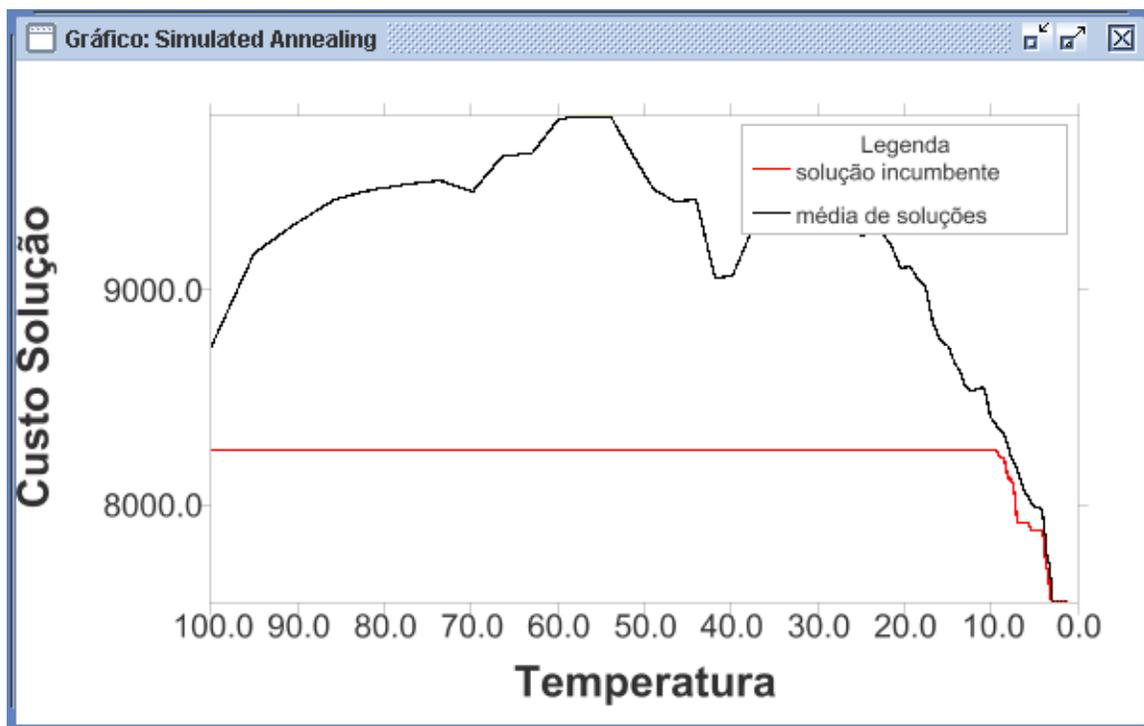


Figura 101 - Gráfico da metaheurística *simulated annealing* aplicado sobre o problema *berlin52.tsp*, após a execução construtiva aleatória e da aplicação da heurística de melhoramento 2-Opt

Para a metaheurística busca tabu, somente uma linha de plotagem é visualizada no gráfico, que representa o *custo da solução* em relação à iteração atual. O número de iterações é definido implicitamente pelo usuário através do parâmetro tempo e pela dimensão do problema. Quanto maior for o número de cidades, o algoritmo necessita um maior tempo para executar uma rodada da Busca Tabu. Quanto maior for o tempo, mais iterações podem ser executadas. Os parâmetros utilizados para o gráfico da figura 102, que representa o resultado da aplicação da busca tabu sobre o problema *pr76.tsp*, são: 40 segundos de processamento, 60 iterações na lista tabu, lista tabu com um tamanho máximo de 25 elementos e sem esquema de vizinhança para a busca local 2-Opt. O ponto onde a linha de plotagem atinge o seu menor valor é a solução incumbente que é apresentada ao

usuário. É importante salientar, que na parte final da execução da metaheurística, ocorreu uma ciclagem da busca tabu, ou seja, as soluções estão em uma seqüência de maneira que a solução vizinha da última solução do ciclo seja a primeira solução do mesmo ciclo. Isso indica ao usuário que o tamanho da lista tabu é muito pequeno ou o número de iterações é muito baixo.

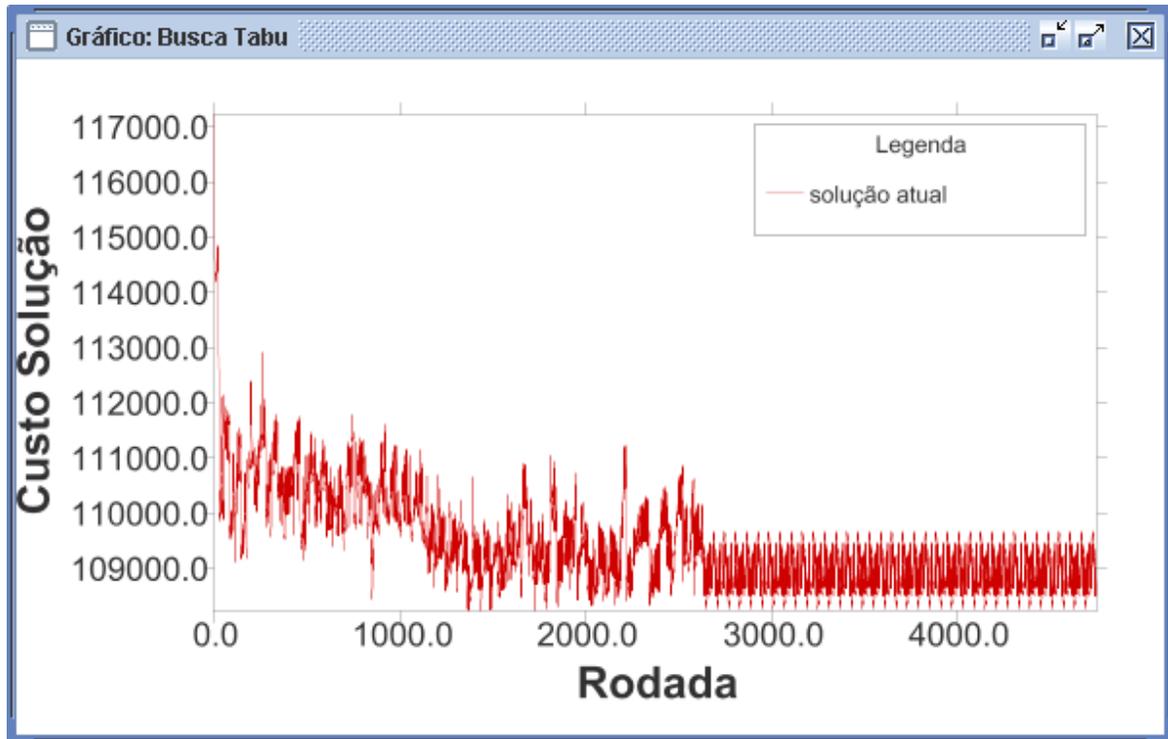


Figura 102 - Gráfico da metaheurística busca tabu aplicado sobre o problema *berlin52.tsp*, após a execução construtiva aleatória e da aplicação da heurística de melhoramento 2-Opt

No modo *expert*, o usuário recebe como resposta do LOBO a janela de resultados e os gráficos, mas as telas de apresentação dos resultados da execução das heurísticas não estão disponíveis.

#### 6.4.2 Botão: Simulação Passo-a-Passo



O botão de simulação passo-a-passo está disponível para as seguintes heurísticas: vizinho mais próximo, inserção mais distante, 2-Opt e 3-Opt. A heurística construtiva aleatória, por seu caráter não previsível e sua simplicidade, não foi incluída na ação de simulação passo-a-passo. Em relação as metaheurísticas, sua complexidade impede que um passo isolado seja visualizado sem a perda do contexto. Como é possível observar nas figuras anteriores, 101 e 102, usualmente as metaheurísticas realizam um número excessivamente longo de passos, o que também tornaria monótono a visualização pausada de sua execução.

O objetivo da simulação passo-a-passo é estritamente pedagógico. Para auxiliar o entendimento dos algoritmos implementados, é interessante que o usuário possa observar o

comportamento das heurísticas através da visualização de seu funcionamento. Desta forma, gargalos e decisões equivocadas, ou muito boas, são observadas *in loco*, ao contrário da simples visualização do resultado final, resultado do botão de ação *executar a heurística*. Ao acompanhar a aplicação da heurística escolhida sobre um determinado conjunto de dados, o usuário pode verificar, a qualquer momento, como o algoritmo se comporta passo-a-passo.

No modo tutoriado do objeto de aprendizagem, ao selecionar uma heurística que possa ser simulada passo-a-passo e apertar o botão de ação específico, uma tela explicativa é apresentada ao usuário. Esta tela contém os detalhes do funcionamento da simulação, que dependem do tipo de algoritmo que está sendo simulado. Após fechar esta tela, o usuário receberá como entrada a *Janela de Resultados* (seção 6.4.1). Contudo, a mesma só exibirá as informações acerca do problema, a heurística escolhida e a matriz de distâncias. Esta janela pode se manter ativa durante toda a simulação. Para executar um passo de simulação, basta o usuário clicar novamente sobre o botão *simulação passo-a-passo*. A cada novo clique, uma nova rodada do algoritmo é executada e a *janela de resultados* e a *área de simulação* são atualizadas de acordo com a heurística escolhida. Ao realizar o último passo, o novo custo encontrado é atualizado e exibido na *Janela de Resultados* e no *Histórico dos Resultados*.

A figura 103 apresenta a tela de apresentação para a simulação passo-a-passo da heurística do vizinho mais próximo.

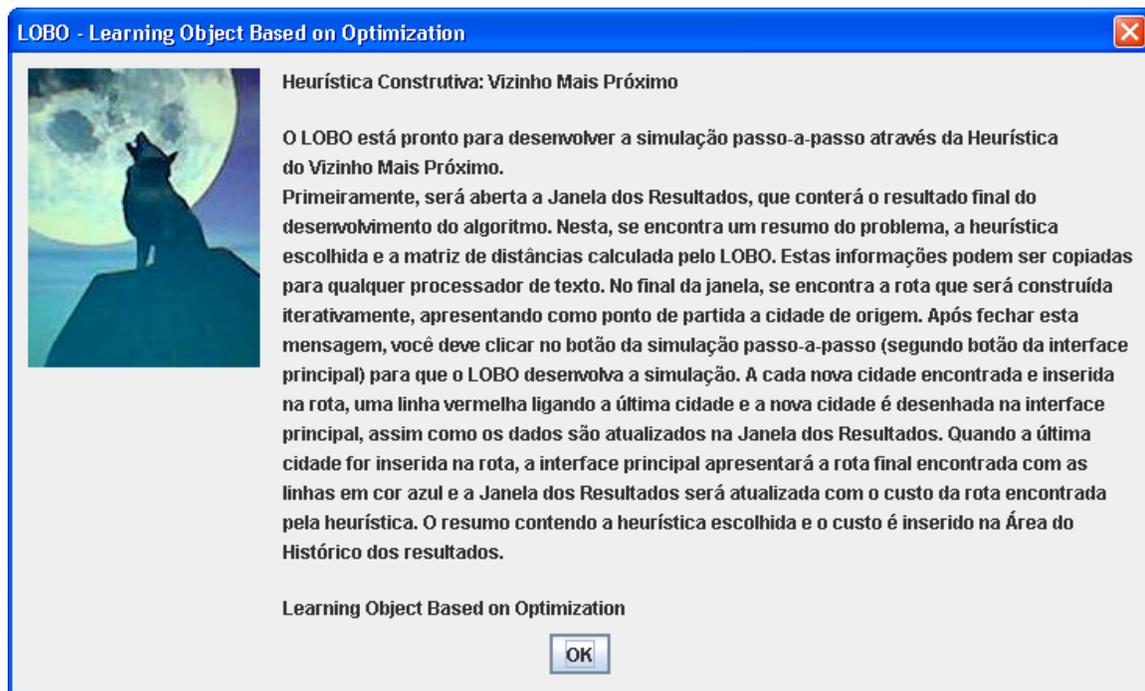


Figura 103 - Apresentação da simulação passo-a-passo da heurística do vizinho mais próximo

Como relatado anteriormente, cada heurística apresenta particularidades na sua exibição passo-a-passo. O algoritmo do vizinho mais próximo é simulado através da inserção de arcos ligando as cidades a cada novo passo. A observação da execução deste algoritmo é extremamente importante para que o aluno perceba como o mesmo tende a piorar a solução final ao diminuir o número de cidades disponíveis para a inserção,

conforme já discutido na seção 6.2.2. Graficamente, é muito mais fácil perceber o problema desta heurística, pois os arcos que são inseridos ao final da execução se tornam maiores e os cruzamentos ocorrem com mais frequência.

A figura 104 apresenta seis momentos da aplicação da heurística do vizinho mais próximo sobre o problema *berlin52.tsp*. É fácil perceber que, no início, o algoritmo se comporta bem (*a,b*), mas rapidamente os arcos se tornam cada vez maiores (*c,d*) até que os cruzamentos se tornam impossíveis de evitar (*e*). No final(*f*), o resultado da solução encontrada apresenta arcos muito ruins: (10,28) e (6,1). Enquanto o algoritmo está construindo a solução, os arcos inseridos são representados pela cor vermelha, para dar destaque à simulação passo-a-passo. Assim que o último arco for inserido no problema, este passa à cor azul, demonstrando o final da simulação (figura 104, f).

A heurística da inserção mais distante apresenta desempenho semelhante. A única diferença para a simulação do vizinho mais próximo está na apresentação da rota sendo construída como uma sub-rota, onde novas cidades são inseridas a cada passo do algoritmo. Uma seqüência desta simulação já foi apresentada na seção 6.2.3, na figura 88.



forma de busca pela melhor rota, ambas já discutidas na seção 6.2.4. Após responder de forma adequada aos parâmetros necessários, a simulação é realizada. O algoritmo 2-Opt escolhe os dois arcos que devem ser retirados e estes são apresentados graficamente através do desenho de linhas em cor vermelha(figura 105,a). A seguir, estes arcos são eliminados da rota(figura 105,b). Finalmente, os dois novos arcos são inseridos(figura 105,c). A cada novo clicar no botão, uma nova rodada do algoritmo é realizada e dois novos arcos são retirados e inseridos, assim como o *histórico dos resultados* é alterado. Quando não existir mais trocas 2-Opt possíveis, a *Janela de Resultados* é exibida com a rota encontrada.

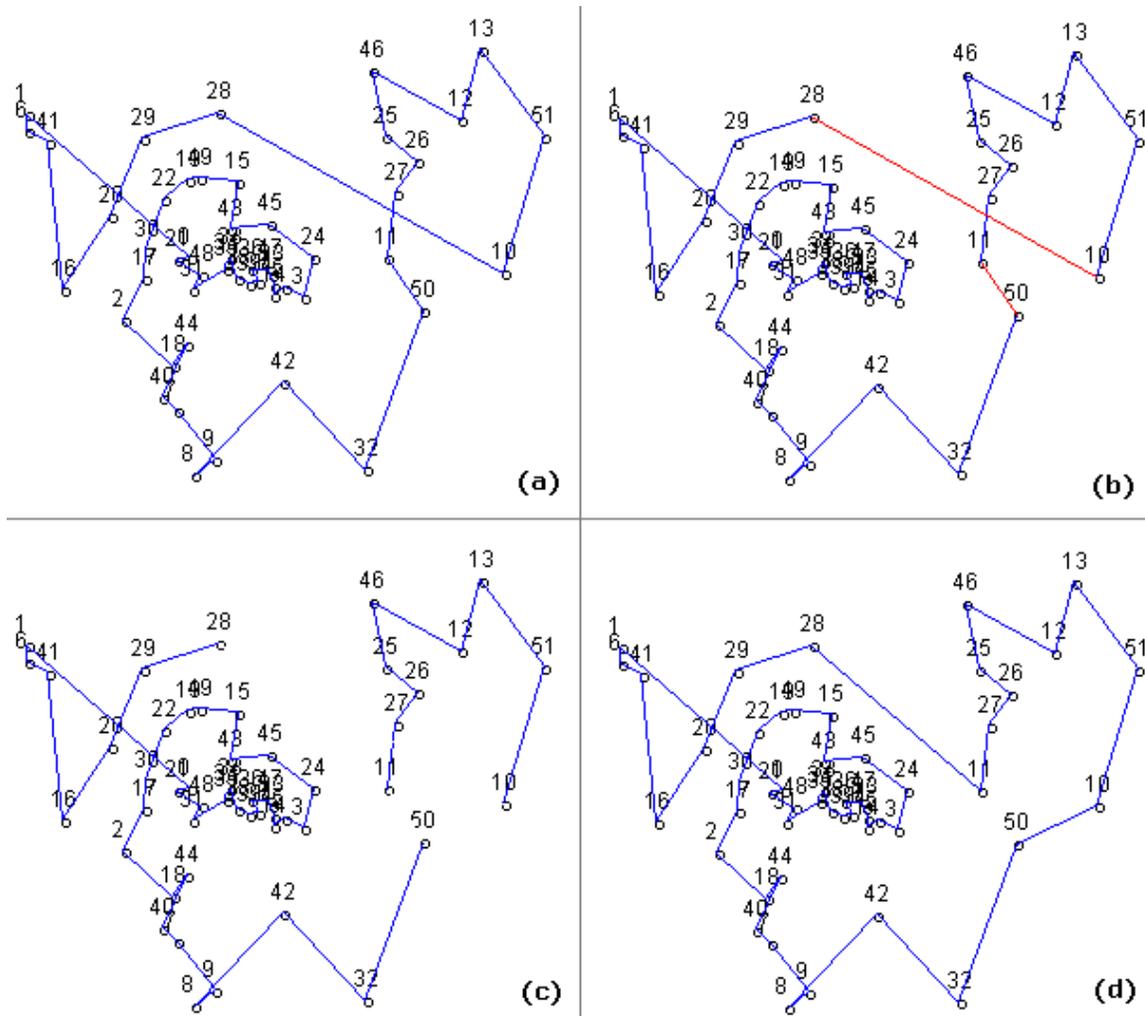


Figura 105 - Simulação de uma rodada 2-Opt passo-a-passo sobre o problema berlin52.tsp

### 6.4.3 Exibir a Rota Ótima



O botão de ação *Exibir a Rota Ótima* tem como única função a visualização da rota de custo ótima, se ela existir. Para os problemas cadastrados no LOBO, as rotas ótimas existem e estão disponíveis. Para problemas carregados pelo usuário, este tem a possibilidade de inserir a rota ótima também, conforme explicado na seção 6.1.3. Conjuntos de dados que não possuem a informação da rota ótima não apresentam funcionalidade para este botão.

A rota ótima, assim como o custo ótimo calculado para a mesma, são apresentados para o usuário em destaque, em cor vermelha e com as linhas mais grossas. A qualquer momento, os alunos e professores podem clicar sobre este botão para exibir a rota e o custo ótimo e compará-los com a rota atual, através da sobreposição dos gráficos, conforme pode ser visualizado na figura 106; os arcos em azul representam a rota que o usuário encontrou através da aplicação do algoritmo de inserção mais distante. Para visualizar novamente a solução atual, basta clicar novamente sobre o botão de ação.

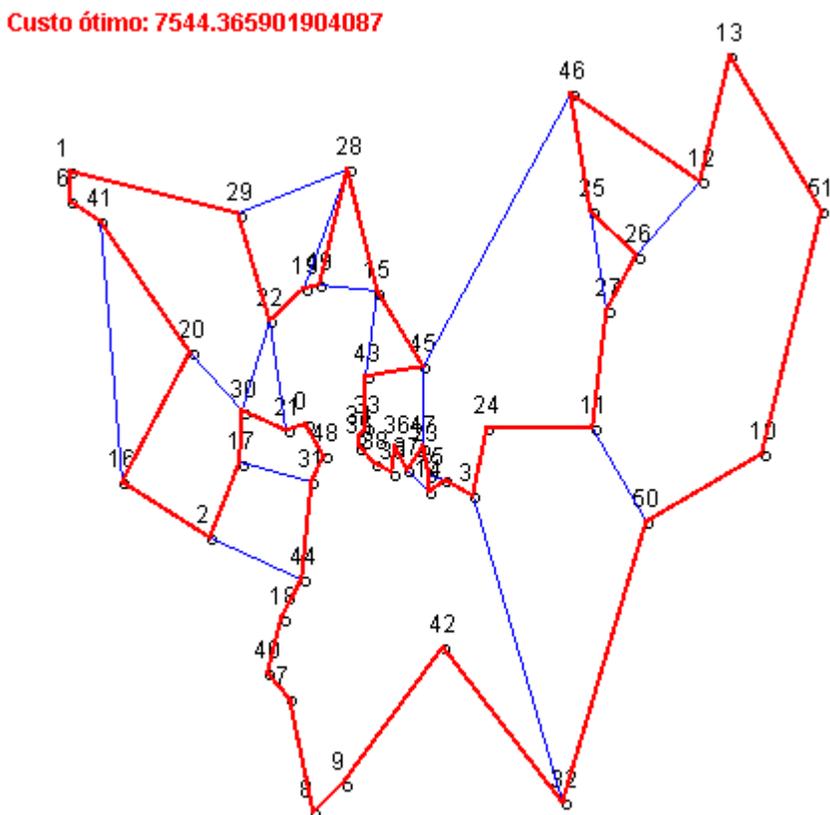


Figura 106 - Exibição da rota ótima para o problema berlin52.tsp

#### 6.4.4 Tutorial do LOBO



O último botão de ação do LOBO é o que comanda o modo tutoriado e o modo *expert*. Quando este botão está pressionado, o usuário se encontra em modo tutoriado e as telas de apresentação e dicas são apresentadas para o mesmo. Se o botão não estiver pressionado, o modo *expert* é ativado e o objeto se torna uma ferramenta de simulação.

#### 6.5 Área de simulação

A área de simulação é a principal interface do objeto LOBO. Nesta área, o usuário pode observar os dados do problema em sua forma gráfica, ou seja, como um conjunto de pontos representando as cidades e um conjunto de linhas representando as ligações entre estas cidades. Cada cidade é representada por um número que se encontra logo acima de um pequeno círculo. A área de simulação foi implementada através de uma câmera sintética (FOLEY, 1997), que pode ser definida como um dispositivo de *software* que permite representar objetos em um plano de projeção. Desta forma, se o plano de projeção for alterado, a representação do objeto também é alterada.

A implicação prática deste conceito transforma o LOBO em um visualizador de problemas de otimização que se adapta ao tamanho da janela que o usuário definir, ou seja, ao redimensionar a janela, os *espaços* entre as cidades aumentam ou diminuem e o problema se adapta visualmente ao novo tamanho. É importante observar que os valores do problema não se alteram, ou seja, a distância calculada entre duas cidades nunca muda de valor. O que muda é sua posição cartográfica na interface visualizada. A figura 107 apresenta o mesmo problema em duas visões de tamanhos diferentes.

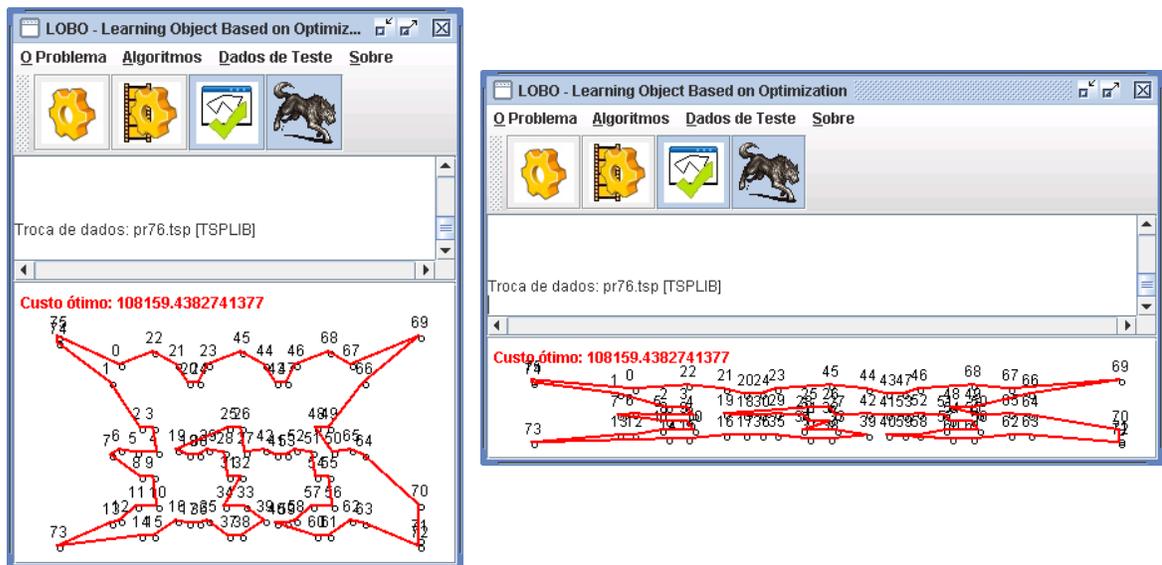


Figura 107 - Diferentes visões para o mesmo problema

## 6.6 Aspectos tecnológicos

O LOBO foi desenvolvido através da tecnologia *Java Web Start*, que permite o desenvolvimento de um aplicativo gráfico e de interface amigável, mantendo a interoperabilidade entre diversos sistemas operacionais e arquiteturas de computadores (atualmente, a *JWS* suporta os sistemas *Windows 98/NT/2000/ME/XP/Vista*, *Solaris*, *Linux*). A discussão acerca das funcionalidades do Java foi referenciada na seção 4.3.3. A complexidade do objeto de aprendizagem não permitia o seu correto desenvolvimento em uma *applet*. Além disso, as linguagens de *script* baseados em servidores *web*, tais como as *servlets*, apresentam dificuldades imensas no desenvolvimento de interfaces gráficas complexas, como é o caso do LOBO, principalmente na simulação das heurísticas (como exemplos, o PHP, para dar suporte a parte gráfica, necessita de um *plug-in* adicional, o GTK, e a linguagem ASP.NET não dá suporte nativo a interfaces gráficas que permitam realizar simulações).

O desenvolvimento dos gráficos utilizados pelas duas metaheurísticas (busca tabu e *simulated annealing*) utilizou como base o *Java Chart Construction Kit* (JCCKit), disponível na Internet em <http://sourceforge.net/projects/jcckit>, e implementada pelo professor Franz-Josef Elmer, da *University of Basel*, Suíça. O JCCKit é uma biblioteca que provê um *framework* para o desenvolvimento de gráficos científicos para a linguagem Java, tanto em aplicações usuais quanto em *applets* para Internet.

Além das facilidades implementadas nesta ferramenta (altamente configurável, programação simples e direta, geração automática de legendas), o JCCKit é distribuído na forma de um arquivo java compactado (tecnologia Jar), o que permite ser descarregado pela *web* de maneira mais rápida. É importante observar também que a biblioteca é distribuída através da licença *Gnu Lesser General Public License*, versão 2.1, de fevereiro de 1999, o que garante a utilização dos seus módulos na implementação de um *software open source*, como o LOBO.

## 6.7 Conclusões do capítulo

Os modelos pedagógicos normalmente utilizados para o desenvolvimento de objetos de aprendizagem são centralizados no aluno, onde este é considerado o principal sujeito da construção do seu próprio conhecimento. O LOBO é plenamente condizente com este aspecto, pois o sistema fornece ao usuário uma gama rica de opções e possibilidades de interatividade, trazendo informações relevantes sobre cada aspecto das simulações que podem ser desenvolvidas. O professor pode fornecer instruções sobre quais problemas resolver ou direcionar a utilização do objeto através de conjuntos de dados específicos mas, em última análise, o aluno está sob seu próprio julgamento ao trabalhar com o simulador podendo, inclusive, desligar o modo *tutoriado* e utilizar o sistema sem qualquer tipo de auxílio.

A experimentação e a interação social são os principais instrumentos para o funcionamento adequado de um modelo pedagógico que utiliza um sistema computacional como mediador. A primeira característica pode ser definida como o próprio cerne do desenvolvimento do LOBO, cujo objetivo central está na condução de experimentos com os algoritmos de otimização combinatória. A própria liberdade que é fornecida ao aluno, como comentado anteriormente, está coerente com esta expectativa. Muito mais que os

acertos, os erros cometidos pelos alunos – aqui traduzidos como simulações com resultados ruins – trazem valiosas experiências e maturidade para o pensamento científico acerca do que está sendo observado, principalmente no que concerne à construção de heurísticas novas ou modificações nos algoritmos já consagrados. O professor pode auxiliar neste processo, trazendo problemas instigantes e que conduzam a resultados ora satisfatórios e ora frustrantes, com o intuito de apresentar aos alunos os pontos positivos e negativos de cada heurística ou metaheurística estudada. Dificilmente, nesta área do conhecimento, há a definição de métodos heurísticos que conduzam a um desempenho satisfatório para a imensa gama de problemas e sub-problemas definidos na literatura. Usualmente, algoritmos que trazem excelentes resultados para uma classe de problemas podem ter desempenho pífio em outra ligeiramente diferente. Desta forma, é muito mais importante que os alunos reconheçam as limitações e os aspectos positivos das heurísticas e metaheurísticas como um todo do que simplesmente decorar o funcionamento de um ou mais algoritmos. É neste ponto específico que um simulador pode atuar de forma mais intensiva, fornecendo um ambiente único e controlado que pode ser experimentado pelos usuários e testado a exaustão, fornecendo detalhes e dados que poderão ser analisados de forma criativa pelos professores e alunos.

Em relação a interação social entre os alunos e seus colegas ou mesmo entre os alunos e o(s) professor(ers), o LOBO não possui nenhuma ferramenta ou funcionalidade específica para o mesmo. No entanto, sua utilização pode ser fundamentada através de um ambiente que o cerque, utilizando o objeto como ponto de apoio para as discussões. Professores podem fornecer dados e/ou algoritmos que devam ser implementados no LOBO e listas para discussão sobre os detalhes técnicos ou dos resultados atingidos podem ser abertas, gerando uma gama de conhecimento que pode extrapolar até mesmo os objetivos iniciais. Alunos usuários do LOBO podem compartilhar suas experimentações e até mesmo estabelecer *campeonatos* de algoritmos, tentando encontrar a combinação ou a implementação de heurísticas e metaheurísticas que gere o melhor resultado para um ou mais problemas escolhidos. Neste caso, o professor pode sugerir novos problemas, particularmente difíceis, disponíveis na literatura (REINELT, 1991).

Outro aspecto importante versa sobre a necessidade dos objetos de aprendizagem serem desafiadores, promovendo a motivação necessária para a investigação. O LOBO apresenta cinco problemas e uma coleção de algoritmos que podem ser utilizados para sua solução. Além da aplicação direta das heurísticas sobre os problemas selecionados, o objeto é fornecido para os professores e alunos na sua forma não compilada, ou seja, seu código fonte. De posse deste, é relativamente fácil para alunos familiarizados com a linguagem Java produzirem extensões do LOBO, criando ou implementando heurísticas clássicas da literatura. A vantagem da utilização do código fonte do objeto está na utilização de uma estrutura pronta e bem definida, assim como a disponibilidade de processos de simulação e visualização. Desta forma, os alunos não precisam se deter em aspectos acessórios, como a leitura dos dados do sistema ou a forma de visualizar a construção de uma solução. Todo este processo já está implementado no LOBO, poupando tempo e deixando a cargo dos alunos a possibilidade de se concentrar no objetivo central do trabalho a ser desenvolvido. Como mencionado anteriormente, é possível criar verdadeiros *campeonatos* de algoritmos, estimulando grupos de alunos a competir entre si através dos resultados alcançados pelos seus algoritmos. Com a facilidade de visualizar graficamente e a qualquer momento a solução que está sendo construída, assim como seu

custo, é possível motivar os alunos a explorarem as possibilidades e interagir com o ambiente de forma mais produtiva.

Ainda em relação a este aspecto, é importante salientar que o material introdutório ao Problema do Caixeiro Viajante, contido no próprio objeto, traz aspectos importantes acerca do problema que podem ser utilizados como um fator motivacional adicional. Além disso, através da implementação de aplicações práticas do problema teórico, condizentes com seu universo, os alunos tendem a considerar com especial interesse o assunto em questão.

É importante salientar que os problemas apresentados para os alunos dentro do LOBO podem ser trabalhados pelo professor na forma de desafios, que instigam os alunos para a resolução dos mesmos. O cerne da concepção pedagógica que será trabalhada no próximo capítulo é o ensino através da resolução de problemas. O objeto de aprendizagem aqui desenvolvido, deste modo, não deve ser enxergado como um facilitador, mas sim como uma ferramenta para apresentar situações-problema na forma de desafios provocativos e interessantes, que possam prender a atenção dos alunos. Estes, por sua vez, devem prover soluções para os problemas apresentados através de mediações dialógicas com professor.

Finalmente, em relação a disponibilidade do objeto de aprendizagem, ou seja, a possibilidade da interação com o mesmo em qualquer lugar e de qualquer modo, o LOBO atinge parte destes propósitos. Desenvolvido com a tecnologia Java, ele pode ser simulado em qualquer computador que provenha esta tecnologia. No entanto, a tecnologia baseada no JWS ainda não permite sua migração para dispositivos portáteis como celulares ou computadores de mão.

Ao final deste capítulo, a metodologia de ensino está definida, assim como um ambiente que suporte sua real utilização. Para o problema específico que foi definido como objeto deste trabalho, um sistema de simulação e desenvolvimento de algoritmos heurísticos e metaheurísticos também foi implementado. Através do objeto de aprendizagem LOBO, o professor e os alunos possuem uma ferramenta capaz de interagir diretamente nos aspectos técnicos e pedagógicos da disciplina em questão durante todo o seu desenvolvimento. No entanto, somente a construção do ferramental não é o suficiente. Mesmo tecnologias modernas e embasadas didaticamente não garantem que o processo pedagógico não siga as práticas bancárias (FREIRE, 1987).

Logo, se há um AVEA que privilegie a concepção dialógica-problematizadora para o ensino de heurísticas e metaheurísticas e também há um objeto de aprendizagem que auxilie o professor na condução de suas aulas, através de experimentações práticas dos algoritmos matemáticos desenvolvidos, estes são suficientes para que a compreensão dos alunos sobre o tema se intensifique? A resposta para esta pergunta está no centro da implementação da prática educacional em heurísticas e metaheurísticas mediadas pelo AMEM e o LOBO nas aulas.

## 7 As Mediações Tecnológicas AMEM-LOBO na Prática

Durante os capítulos predecessores, são estabelecidos alguns conceitos e metodologias que abarcam uma nova filosofia centrada no diálogo entre professores e alunos como peça fundamental para a construção do saber em bases firmes, reconhecendo como importância ímpar a transformação da passividade em proatividade através do desenvolvimento da iniciativa pessoal e o reconhecimento da realidade de cada aluno como um elemento rico e profícuo de idéias, e assumindo como inequívoco o papel do professor como guia deste processo. Desta forma, é necessário, para que o ciclo educacional esteja completo e faça sentido, que além da metodologia escolhida, depois do ambiente educacional mediador estar implementado e testado e, finalmente, após o desenvolvimento do objeto educacional citado anteriormente, o desenvolver das atividades docentes deve ser relatado, detalhado e analisado.

Neste capítulo, serão apresentados algumas aulas e as primeiras conclusões que podem ser obtidas ao aplicar todo o arcabouço mediático e metodológico já explanados, descrevendo a experiência desta articulação na disciplina Inteligência Artificial, pertencente ao currículo pleno do Curso de Ciência da Computação da Universidade Federal de Santa Maria, no primeiro semestre do ano de dois mil e sete.

A disciplina tem sessenta horas e foi dividida em trinta encontros de duas horas, na modalidade presencial. Os objetivos definidos para ela são: *Aplicar os conceitos e técnicas de inteligência artificial, dando ênfase ao projeto e à construção de sistemas de resolução de problemas*. As unidades a serem cumpridas são *A Inteligência Artificial, Problemas, Espaços e Busca, Representação do Conhecimento e Técnicas Heurísticas*.

A Matriz Dialógica-Problematizadora da disciplina Inteligência Artificial já foi apresentada anteriormente no capítulo 10, na seção 3.3. No desenvolver da programação das atividades docentes, estabelece-se as ligações entre os conteúdos desenvolvidos e as perguntas de pesquisa que podem ser analisadas durante a aplicação das aulas.

De uma forma mais genérica, a disciplina Inteligência Artificial foi dividida em quatro seções, que possuem coesão entre si e que, juntas, fornecem uma explanação metodológica de como o conhecimento nesta área é desenvolvido. O LOBO é utilizado nas quatro seções, que são:

- **Concepção e metodologia:** neste primeiro momento, que representa cronologicamente as primeiras aulas da disciplina, são estabelecidas algumas idéias básicas acerca da área do saber e também sobre a metodologia de trabalho. O problema que será tratado durante todo o decorrer da disciplina (Problema do Caixeiro Viajante), também é desenvolvido nesta seção;
- **Desenvolvimento de Heurísticas Construtivas:** a segunda seção já inicia com a resolução do problema proposto anteriormente. Agora, já cientes da natureza do problema, é necessário buscar soluções para o mesmo e, de forma mais natural, a aplicação de heurísticas construtivas, usualmente mais simples e intuitivas, são apresentadas primeiramente;
- **Desenvolvimento de Heurísticas de Melhoramento:** nesta fase, é esperado que os alunos já tenham desenvolvido, tanto intelectualmente quanto na prática, através da construção de algoritmos construtivos, um entendimento muito mais apurado do problema e das dificuldades em resolvê-lo de forma eficiente. Nesta

seção, são apresentadas as heurísticas de melhoramento que necessitam, como sua base primária, de uma solução inicial, que é dada pelas heurísticas construtivas;

- Desenvolvimento de Metaheurísticas: o desenvolvimento de metaheurísticas se deve, basicamente, aos problemas encontrados nas heurísticas de melhoramento que, sozinhas, só podem atingir até um determinado patamar de eficiência. Através de experiências práticas, os alunos, nesta última fase da disciplina, já possuem elementos teóricos e experimentais para analisar as deficiências das heurísticas de melhoramento, que são a principal razão de ser das metaheurísticas.

As próximas páginas estabelecem as aulas iniciais da disciplina Inteligência Artificial, notadamente centradas na primeira seção. Para cada aula, são apresentados o conteúdo programático de cada ponto, o tempo em que será desenvolvido este tópico e as questões metodológicas de um ponto específico. Os elementos da matriz dialógica-problematizadora envolvidos em cada encontro são discutidos no final de cada aula.

É importante frisar que as colaborações, além de serem respondidas ao professor, eram também postadas em *fóruns* abertos para cada atividade, para que as mesmas pudessem ser debatidas entre os alunos durante a semana que antecedia a aula presencial. Estabelece-se, para questões de nomenclatura, as abreviaturas *DI* (desafio inicial), *MSEM* (melhor solução educacional no momento) e *DA* (desafio mais amplo) e *MDP* (matriz dialógica-problematizadora).

## 7.1 Concepção e Metodologia

### 7.1.1 Apresentação da Disciplina

Esta aula está dividida em quatro pontos:

<b>Ponto 1</b>	Apresentação da disciplina. Discussão sobre o que é Inteligência e do que trata a Inteligência Artificial.
<b>Tempo</b>	30 minutos
<b>Metodologia</b>	<b>DI:</b> o que é inteligência? <b>MSEM:</b> conceituação de inteligência. <b>DA:</b> algum dia, teremos, realmente, uma inteligência de máquina?
<b>Observações</b>	A conceituação da inteligência pode ser realizada facilmente através das respostas fornecidas pelos alunos, principalmente em relação a inteligência de máquina, que é tratada pela disciplina. Nas experiências realizadas, os estudantes tendem a conceituar inteligência baseada no comportamento, o que condiz fortemente com a disciplina. Durante a MSEM, são explorados também os conceitos de IA forte e IA fraca e o Teste de Turing (PENROSE, 1997), que embasam os alunos para o desafio mais amplo. O DA será tratado, de forma mais específica, através de uma atividade.
<b>Ponto 2</b>	Apresentação do ambiente de apoio à disciplina: AMEM - Ambiente Multimídia para Educação Mediada por Computador

<b>Tempo</b>	30 minutos
<b>Metodologia</b>	<b>DI:</b> o que é um AVEA? <b>MSEM:</b> conceituação de AVEA e suas metodologias pedagógicas. <b>DA:</b> é possível desenvolver um AVEA que contemple uma metodologia pedagógica?
<b>Observações</b>	A discussão sobre o que é um AVEA e suas vantagens em relação a mera exposição do material didático na página de um professor são tópicos importantes na consolidação da metodologia de trabalho que está sendo desenvolvida. Considerando também que, no caso específico da turma, os estudantes são alunos de graduação de um Curso de Ciência da Computação, os aspectos da concepção tecnológica e pedagógica de um AVEA são elementos que podem ser discutidos com propriedade.
<b>Ponto 3</b>	Cadastro no Ambiente AMEM
<b>Tempo</b>	20 minutos
<b>Observações</b>	Foi reservado um espaço de vinte minutos para o cadastro no ambiente, mas como a grande maioria dos alunos já utilizava o mesmo devido a disciplinas anteriores, o tempo desenvolvido foi muito menor que o esperado, e o espaço cronológico foi preenchido pelo próximo ponto.
<b>Ponto 4</b>	Definição da Metodologia de Trabalho
<b>Tempo</b>	20 minutos
<b>Metodologia</b>	<b>DI:</b> o que é uma metodologia pedagógica? <b>MSEM:</b> discussão sobre as diversas metodologias educacionais. <b>DA:</b> a metodologia dialógica-problematizadora, sob o ponto de vista dos alunos, é adequada para o ensino de heurísticas e metaheurísticas dentro da disciplina Inteligência Artificial?
<b>Observações</b>	Muitas vezes, os alunos, principalmente das áreas tecnológicas, traçam paralelos entre os professores e, mesmo sem uma consistência teórica, verbalizam suas opiniões sobre os docentes que, de uma maneira geral, são divididos em <i>bons</i> ou <i>ruins</i> e/ou <i>fáceis</i> ou <i>difíceis</i> . É também comum entre ouvir conversas sobre a <i>falta de didática</i> de um determinado professor. No entanto, raros são os que se preocupam de maneira mais aprofundada sobre o tema. Como a disciplina foi construída em torno de uma apresentação metodológica diferenciada, muito diferente da construção bancária e positivista que grande parte dos alunos está acostumada, é interessante trabalhar com os mesmos alguns conceitos introdutórios sobre as questões pedagógicas e estabelecer as bases da metodologia que será empregada durante o resto do semestre. Em relação ao DA, o mesmo será discutido em todo o semestre, através de um fórum permanente criado no AMEM. Ao final da disciplina, os alunos receberão um questionário, onde serão propostas algumas questões relativas a metodologia como um todo.

Em relação à MDP, foi definido a pergunta B3 para ser respondida durante esta aula: *Os estudantes têm assumido que os problemas vistos em aula realmente contribuem no processo de aprendizagem de heurísticas e metaheurísticas?* Esta pergunta, que

realmente não pode ser respondida somente na primeira aula, deve estar em mente do professor durante a mesma para que ele possa perceber como foi estabelecida a receptividade dos alunos em relação a explicação da metodologia proposta para a disciplina. A aula aplicada demonstrou que, neste primeiro momento, os alunos sentem-se atraídos pela idéia de desenvolver a disciplina através de uma experimentação baseada na dialogicidade e na construção conjunta do conhecimento, o que demonstra que, alunos com este grau de amadurecimento (a disciplina ocorre no último semestre do curso), estão preocupados também com questões relativas ao aprendizado fora dos bancos escolares tradicionais e institucionais.

### 7.1.2 Inteligência Artificial e suas Inter-relações com a Sociedade da Informação

Esta aula está dividida em dois pontos:

<b>Ponto 1</b>	Exibição da entrevista de Minsky sobre a Inteligência Artificial, seus problemas, perspectivas e o futuro da área.
<b>Tempo</b>	40 minutos
<b>Observações</b>	Marvin Minsky é um dos "pais" da Inteligência Artificial, co-fundador do <i>Artificial Intelligence Laboratory</i> do <i>Massachusetts Institute of Technology</i> . A entrevista, distribuída na série IstoÉ Grandes Temas: Inteligência Artificial, Minsky trata da sua própria experiência dentro da área e discute alguns temas relacionados ao futuro da área e, principalmente, sobre como a IA pode resolver problemas.
<b>Ponto 2</b>	Análise da entrevista e debate.
<b>Tempo</b>	60 minutos
<b>Metodologia</b>	<b>DI:</b> a IA pode resolver problemas difíceis por ser <i>inteligente</i> ou porque o programador era <i>inteligente</i> ? <b>MSEM:</b> conceituação de "problema" em IA e solução. <b>DA:</b> algum dia, teremos, realmente, uma inteligência de máquina?
<b>Observações</b>	Através do debate após a entrevista do Minsky, é possível estabelecer as diferenças entre a <i>resolução inteligente de problemas</i> e a própria inteligência. Nas experiências realizadas, os alunos têm se mostrado céticos em relação a um <i>ser</i> inteligente que possa provir da computação atual, opinião que é compartilhada por Minsky, que apregoa que um novo tipo de <i>hardware</i> e <i>software</i> são necessários para o desenvolvimento real de uma inteligência de máquina. É importante ressaltar que o DA deste ponto é igual ao DA do primeiro ponto da aula anterior. Neste momento, o professor deve dirigir os alunos para o debate sobre o futuro da IA, em termos das perspectivas levantadas por Minsky e os debates em aula.

Em relação à MDP, foram definidas as pergunta C3 e A2 para serem respondidas durante esta aula: (C3) *Será que o Problema do Caixeiro Viajante é adequado para o ensino de heurísticas e metaheurísticas?*; e (A2) *Como os professores podem instigar os alunos, utilizando a problematização e o AMEM?* A primeira questão, que também permeia outras aulas e é discutida no final da disciplina com os alunos, envolve as expectativas dos alunos em relação ao tema. Através da discussão dos alunos e da proposta

da primeira atividade, que é apresentada a seguir, são estabelecidos alguns parâmetros que servem como guia para responder o questionamento.

A grande maioria dos alunos acredita que a resolução inteligente de problemas é o objetivo da Inteligência Artificial ou, mais precisamente, o objetivo maior ou mais *palpável*. Desta forma, é crível trabalhar com um problema gerador, no caso o Problema do Caixeiro Viajante, que pode ser formulado rapidamente e que tem uma grande aplicação prática como ponto de apoio para a disciplina. A percepção das colocações que são discutidas nesta aula e na colaboração desenvolvida fora do *locus* escolar deve ser levada em conta quando da definição do PCV como problema a ser trabalhado, principalmente em relação a importância do mesmo, o que serve como parâmetro para analisar a segunda pergunta.

<b>Atividade 1</b>	A Inteligência Artificial e a Inteligência de Máquina
<b>Tempo</b>	02 semanas
<b>Questão</b>	Com base no debate em aula sobre a visão de Minsky sobre a Inteligência Artificial e através das leituras das bibliografias indicadas abaixo, responda: sistemas baseados em Inteligência Artificial podem ser realmente considerados inteligentes?
<b>Bibliografia</b>	MAROLDI, Marcelo Masson. Computação e cognição. In: Ciências & Cognição, v. 07. pg. 122-127. 2006. Disponível por www em <a href="http://www.cienciasecognicao.org/pdf/v07/m31676.pdf">http://www.cienciasecognicao.org/pdf/v07/m31676.pdf</a> SICHMAN, Jaime. Mais próximos de HAL 9000: caminhos e desafios da Inteligência Artificial. Entrevista com Jaime Simão Sichman, Revista Eletrônica ComCiência, 2005. Disponível por www em <a href="http://www.comciencia.br/entrevistas/2005/10/entrevista2.htm">http://www.comciencia.br/entrevistas/2005/10/entrevista2.htm</a> Inteligência Artificial. Verbete, Wikipédia, disponível por www em <a href="http://pt.wikipedia.org/wiki/Inteligência_Artificial">http://pt.wikipedia.org/wiki/Inteligência_Artificial</a>

*Análise da Atividade 1* – apresentada e discutida durante a aula *O Problema do Caixeiro Viajante*

A atividade 1 da disciplina foi desenvolvida com o intuito de fazer os alunos pensarem sobre o que é inteligência e como ela pode ser implementada dentro de um computador. Analisando as respostas dos alunos, podemos verificar que existem quatro grupos bem distintos:

- não há como gerar uma verdadeira inteligência de máquina, ou seja, a inteligência é inerente aos organismos vivos: os alunos que defendem este ponto de vista não estão sozinhos, pois diversos pesquisadores, que foram *catalogados* como desenvolvedores da IA Fraca por Searle, acreditam que uma verdadeira inteligência de máquina não pode ser obtida através de algoritmos. É importante notar que, mesmo sendo alunos do último semestre do curso, a grande maioria dos mesmos nunca havia desenvolvido um tipo de programa que envolvesse variáveis de controle que podem alterar de forma dramática o comportamento do mesmo (típico das heurísticas). Desta forma, frases como “...*por mais não determinístico que seja um sistema, ele não passa de uma seqüência de código a ser seguida...*”, “*Os sistemas que utilizam Inteligência Artificial, na verdade são representações da inteligência de seus criadores. Feitos para auxiliar em*

*tarefas específicas, os programas que usam inteligência artificial, apenas geram resultados a problemas previstos (normalmente) pelos seus desenvolvedores”, e “...os sistemas não são capazes de compreender o problema em questão...”* são comuns e bem recebidas por estes alunos;

- é possível resolver problemas de forma inteligente e é possível *simular* uma inteligência real: aqui, a discussão permeia outros caminhos. A questão sobre a *simulação* de comportamentos é bastante emblemática para estes alunos, envolvendo, principalmente, relações sobre o *objetivo* do sistema desenvolvido. Se o objetivo é que um sistema possa *imitar* com perfeição um ser humano, tanto fisicamente quando em relação as suas ações, estes alunos acreditam que isso é possível. Mas se o objetivo é imitar o pensamento humano, então há problemas extremamente complexos na área que este grupo de alunos acredita serem insolúveis. Algumas frases representativas destes alunos exemplificam de forma clara estes pensamentos: “...eles (os computadores) não conseguem ter consciência da problemática em questão...”, “...as máquinas podem reagir, dialogar e agir da mesma forma que os seres humanos, o que vem a ser a reprodução do comportamento. No entanto, elas não fazem isso por conta própria, de livre e espontânea vontade...” e “...sistemas baseados em inteligência artificial ainda não são capazes de simular com perfeição o pensamento...”;
- a resposta depende do conceito de inteligência: muitos alunos acreditam que, para reconhecermos se um computador é realmente inteligente, a questão sobre o que é inteligência está em primeiro lugar. Baseados fortemente na dicotomia de Searle, este grupo apresentou respostas coerentes com as diferenças existentes entre a solução inteligente de problemas e a verdadeira inteligência, como é possível observar nos seguintes pensamentos: “...depende o que for considerado “inteligência”. Se ao abordarmos o conceito de inteligência como um conceito relativo cognição do ser humano, responsável pela formação da razão, não considero que uma máquina possa realmente ser inteligente. Por outro lado, se considerarmos inteligência como instrumento de resolução de problemas, com capacidade de adaptação à certas circunstâncias, pode-se dizer que a máquina é inteligente...” e “...se considerarmos que a capacidade de reconhecer padrões e encontrar soluções adequadas é uma forma de inteligência, então podemos afirmar que a IA é inteligente...”;
- por último, há um grupo (reduzido) de alunos, que acredita que é possível a criação de uma verdadeira inteligência de máquina. Novamente, cabe salientar que os alunos estão apenas no início da disciplina e a *desconfiança* acerca de como ou por que um algoritmo pode ser considerado inteligente ainda está plenamente enraizada. É interessante observar que, na bibliografia apresentada para a construção da atividade, todos os pontos de vista apresentados reafirmam a convicção de que, algum dia, uma inteligência verdadeiramente computacional vai emergir. Mesmo assim, a maioria dos alunos não optou por este caminho. Para os que acreditam nesse caminho, as frases mais contundentes foram: “...talvez encontremos a forma mais adequada de realizar “sinapse” em silício...” e “...(a inteligência) é a capacidade deste em interagir e aprender para a realização de um objetivo. E isso, alguns sistemas atuais

*baseados em inteligência artificial já são capazes de fazer, ou seja, esses sistemas podem ser realmente considerados inteligentes...”*

O objetivo desta tarefa, muito mais do que simplesmente medir o conhecimento do aluno ou verificar se ele entendeu ou adquiriu algum preceito básico, era realmente instigá-los a gastar parte do seu tempo para pensar sobre o futuro da computação como um todo, pois a área de Inteligência Artificial está rapidamente abarcando preceitos e interferindo no desenvolvimento de inúmeros outros setores da informática. Ao pensar sobre o que é inteligência e como ela pode ser processada em uma máquina, alguns conceitos começam a aparecer com mais frequência no vocabulário nos alunos, mesmo que de forma incipiente, principalmente relativos a: o que é um *problema* e o que é um *objetivo*. Este conceitos, pincelados de forma vaga nas respostas, foram trabalhados com os alunos em dois momentos:

- de forma direta, através da inserção de comentários nas respostas dos alunos;
- de forma global, através de um debate presencial sobre o tema, após a finalização da tarefa.

### 7.1.3 Complexidade Computacional

Antes de observar com mais detalhamento esta aula, é importante ressaltar que os alunos de Inteligência Artificial devem realizar a disciplina de *Projeto e Análise de Algoritmos*, que tem como objetivos analisar e projetar algoritmos, levando em consideração a complexidade computacional envolvida, com o objetivo de encontrar soluções computacionais ideais para os problemas. Sua ementa envolve: (a) Introdução e fundamentos matemáticos; (b) Técnicas de projeto e análise; e (c) Algoritmos selecionados.

Desta forma, a aula sobre Complexidade Computacional serve para recapitular alguns conceitos e adequar algumas definições, exemplificando as métricas de complexidade em relação aos problemas que a inteligência artificial usualmente abarca. Esta aula está dividida em dois pontos.

<b>Ponto 1</b>	Análise de Algoritmos e Tempo Polinomial
<b>Tempo</b>	50 minutos
<b>Metodologia</b>	<p><b>DI:</b> como saber quanto tempo um computador leva para resolver um problema?</p> <p><b>MSEM:</b> definição de problema, algoritmo, instância e solução.</p> <p><b>DA:</b> há soluções melhores do que outras?</p> <p><b>DI:</b> o que é um algoritmo <i>bom</i>?</p> <p><b>MSEM:</b> definição de tempo polinomial.</p> <p><b>DA:</b> todos os problemas são resolvidos em tempo polinomial?</p>
<b>Observações</b>	O primeiro ponto desta aula é dividido em duas partes em termos de estratégia metodológica. Primeiramente, o aluno é instigado a pensar, ou mais apropriadamente, a recordar, os conceitos da disciplina <i>Projeto e Análise de Algoritmos</i> . Na experiência realizada, era fácil constatar que muitos tinham noções razoáveis sobre complexidade computacional e que a construção das definições foi realizada

satisfatoriamente. O DA desta parte é, em última análise, o DI da segunda parte, pois para que um algoritmo seja *melhor* do que outro, é necessário saber o quão *bom* é o sistema que está sendo avaliado. Em relação à segunda parte, a intenção era repensar a complexidade em termos mais próximos à Inteligência Artificial e, mais precisamente, a área de Heurísticas e Metaheurísticas. A definição do que é um tempo polinomial e a forma como podemos caracterizar um *bom* algoritmo são essenciais para o entendimento dos algoritmos que são apresentados posteriormente. Os alunos, usualmente, conceituaram um algoritmo como *bom* ligando a expressões como *rápido* e *solução eficiente*. Os dois conceitos são essenciais na definição do que é o PCV e como ele pode ser resolvido.

<b>Ponto 2</b>	Redução e Classes de Problemas
<b>Tempo</b>	50 minutos
<b>Metodologia</b>	<p><b>DI:</b> existe diferença entre resolver o problema e avaliar o custo de um problema?</p> <p><b>MSEM:</b> definição de redução e classes de problemas.</p> <p><b>DA:</b> quais são os problemas que não podem ser resolvidos em tempo polinomial que existem na literatura?</p>
<b>Observações</b>	<p>A maioria dos alunos não percebe a diferença entre a resolução e a avaliação do custo de resolver um problema e, mesmo depois de todas as discussões, tais pendências permaneceram por um bom tempo. Esta questão só se tornou consolidada nos alunos no momento em que os mesmos construíram seus próprios algoritmos e testaram, na prática, os diferentes métodos heurísticas e metaheurísticos. Com os resultados computacionais na mão, as definições apresentadas nesta aula se tornaram mais claras. A DA deste ponto foi utilizada como ponto de partida para uma atividade de Colaboração.</p>

Por se tratar de uma aula que permeia o objetivo principal da disciplina, foi escolhida a pergunta C1 da MDP: *A abordagem dialógica-problematizadora potencializa o ensino de heurísticas e metaheurísticas?* Problematizar o conhecimento prévio dos alunos, de forma a obter uma construção mais rica dos conceitos e seus inter-relacionamentos, constituiu uma prática interessante para abordar um assunto intrincado como a Complexidade Computacional. Através da construção de soluções para o desafio (DI), os alunos foram conduzidos a pensarem sobre suas próprias asserções mentais acerca de um tema já visto em outro momento, cuja relação aos objetivos educacionais atuais é crucial. Novamente aqui, mostra-se mais uma vez a importância da gerência do processo educativo pelo professor. Mesmo mantendo um canal aberto de diálogo e de comunicação com seus alunos, ele deve conduzir os mesmos através das ligações entre os assuntos, caso estas não sejam corretamente interpretadas pelos mesmos. Na experiência realizada, foi observado que havia problemas em relação a conceituação formal, pois noções acerca de tempo computacional eram facilmente confundidas com tempo de máquina. Ao pensar sobre os problemas apresentados, alguns alunos saíram do conforto do *saber* anterior e sentiram-se, momentaneamente, desconfortáveis com a dicotomia entre o que eles *imaginavam* ser verdade e como esta mesma *verdade* já não era mais aplicável em outras situações. O desafio inicial, neste caso, atuou fundamentalmente na *desestabilização* dos conceitos para que novas idéias fossem discutidas.

Ao final desta aula, foi apresentada aos alunos para que, trabalhando no AMEM, realizassem a seguinte atividade de colaboração.

<b>Colaboração 1</b>	Quais são os problemas que não podem ser resolvidos em tempo polinomial que existem na literatura?
<b>Tempo</b>	Até o próximo encontro presencial
<b>Descrição</b>	Considerando as questões relativas à Complexidade Computacional desenvolvidas na última aula, traga os tipos e a conceituação sobre os problemas que não podem ser resolvidos em tempo polinomial.

#### *Análise da Colaboração 1 – apresentada e discutida na aula O Problema do Caixeiro Viajante*

Ao final da terceira aula, foi apresentado aos alunos a colaboração que poderia ser respondida até a próxima aula, sem compromisso em relação a atribuição de notas. Um percentual muito menor de alunos respondeu a mesma, comparativamente à *atividade 1*, que tinha um caráter obrigatório. De qualquer forma, vários alunos buscaram informações, notadamente na *internet* (em grande parte, utilizando-se da *wikipedia*<sup>1</sup>), sobre os tipos de problemas pedindo, trazendo definições sobre o próprio Problema do Caixeiro Viajante, o Problema da Mochila, o Problema da Coloração de Grafos e o Problema do Particionamento. As informações contidas nas respostas dos alunos foram utilizadas na próxima aula, a última desta seção, ação que representa o verdadeiro objetivo desta colaboração, tentando mostrar ao aluno que sua participação influi no desenvolvimento da aula e tem resposta imediata do professor à suas colocações.

#### 7.1.4 O Problema do Caixeiro Viajante

Esta aula apresenta o Problema do Caixeiro Viajante, suas possíveis variações e sua formulação matemática. Ela é dividida em quatro pontos.

<b>Ponto 1</b>	Discussão e Entrega da Atividade 1
<b>Tempo</b>	20 minutos
<b>Ponto 2</b>	Discussão da Atividade de Colaboração
<b>Tempo</b>	20 minutos
<b>Observações</b>	Neste ponto, foram discutidos os problemas apresentados pelos alunos, levantando as possíveis aplicações práticas dos mesmos, tendo o cuidado de deixar o PCV como última opção.
<b>Ponto 3</b>	Problemas de Otimização Combinatória
<b>Tempo</b>	40 minutos
<b>Metodologia</b>	<b>DI:</b> o que é um Problema Combinatório? <b>MSEM:</b> formulação dos Problemas de Otimização Combinatória e

<sup>1</sup> <http://pt.wikipedia.org/wiki> - enciclopédia colaborativa

	estratégias de resolução.
<b>Observações</b>	O DI é baseado na generalização dos problemas encontrados pelos alunos, discutidos na colaboração. O importante deste ponto é discutir com os alunos as questões relativas ao número de soluções possíveis para um problema, o que se traduz na impossibilidade prática de alcançar a melhor solução possível para uma determinada instância através da enumeração pura e simples de todas as possibilidades. Conceitos formais relativos a matemática combinatória são explorados. Adicionalmente, são explorados conceitos relativos a formalização de algoritmos para a <i>construção</i> de soluções para problemas combinatórios.
<b>Ponto 4</b>	O Problema do Caixeiro Viajante
<b>Tempo</b>	30 minutos
<b>Metodologia</b>	<b>DI:</b> o que é o Problema do Caixeiro Viajante? <b>MSEM:</b> formulação matemática e variantes do PCV. <b>DA:</b> definir um algoritmo que <i>construa</i> uma <i>boa</i> solução para instâncias do PCV.
<b>Observações</b>	O ponto de partida dos alunos para o DI foi a pesquisa realizada na colaboração e a própria definição do que é um <i>caixeiro viajante</i> . Através das suas próprias palavras e da análise realizada no ponto anterior, foi possível construir uma definição <i>informal</i> do problema, materializada posteriormente nas fórmulas e restrições do mesmo. A partir da formulação, as variantes do problema (normalmente impostas por restrições adicionais), foram discutidas. Neste ponto, é apresentado o LOBO para os alunos, onde os mesmos têm o primeiro contato com o objeto de aprendizagem. As noções de distâncias euclidianas e da construção de uma rota que unisse todas as cidades, passando somente uma vez em cada ponto e retornando à cidade de origem, foram apresentadas através da interface gráfica do objeto.

Para esta aula, foram escolhidas as mesmas perguntas da aula 2: (C3) *Será que o Problema do Caixeiro Viajante é adequado para o ensino de heurísticas e metaheurísticas?*; e (A2) *Como os professores podem instigar os alunos, utilizando a problematização e o AMEM?* Durante a explanação do PCV e de sua formalização, foram observados a reação dos alunos, em forma de perguntas e outras manifestações, sobre como eles compreendiam o problema em relação à IA e a computação em geral. Buscou-se, neste momento, incentivar o desenvolvimento de bons algoritmos para o problema com exemplos reais da aplicação do PCV, demonstrando que um tema tão ligado à pesquisa também possui um viés prático interessante e que pode e deve ser explorado mais adequadamente pela indústria e pela área de gestão administrativa. No caso específico do PCV, o viés prático apresentado foi o relato de casos de utilização de algoritmos heurísticos e metaheurísticas em situações reais, conforme relatado em (TOSO, MORABITO, 2005), (ALVES, CARVALHO, 2001), (FARKUH NETO, LIMA, 2006) e (ARAÚJO, ARENALES, 2003).

Analisando a experiência realizada, percebeu-se que alguns alunos buscavam mais informações sobre as aplicações práticas, mas os demais encararam com uma relativa indiferença a escolha do problema em si. Talvez, neste ponto, fosse necessário um trabalho

colaborativo ou uma atividade que os levasse a pesquisar, por si só, as possibilidades de aplicação do problema alvo na sociedade produtiva em geral.

Novamente, ao final desta aula, foi apresentada aos alunos para que, trabalhando no AMEM, realizassem a seguinte atividade de colaboração.

<b>Colaboração 2</b>	É possível definir um algoritmo único que <i>construa</i> uma <i>boa</i> solução para diversas instâncias do PCV?
<b>Tempo</b>	Até o próximo encontro presencial
<b>Descrição</b>	Considerando a formulação do PCV, desenvolva um algoritmo que gere uma solução de custo razoável para as diversas instâncias apresentadas.
<b>Bibliografia</b>	As cinco instâncias que são utilizadas no LOBO ( <i>berlin52, pr76, eil51, kroA100 e lin105</i> ), retiradas da TSPLib.

### *Análise da Colaboração 2 – apresentada e discutida na aula Heurísticas Construtivas: Introdução e Exemplos*

Apenas quatro alunos responderam a colaboração 2, o que levou a um replanejamento da próxima aula, onde um debate entre alunos e o professor foi realizado no início do encontro presencial, com o intuito de discutir com os alunos as causas do baixo interesse em realização das tarefas que, a princípio, não seriam avaliadas em relação a uma nota que influenciasse a média final. A análise deste debate será apresentada na próxima seção.

Em relação aos resultados apresentados, os alunos, confirmando a experiência prática dos professores, apresentaram, diferindo em forma e acuidade, versões mais ou menos elaboradas da heurística do vizinho mais próximo, já apresentada neste trabalho na seção 6.2.2. A explicação para este fato está na própria noção ingênua com que a heurística foi construída: inserir sempre o arco de menor custo a partir da cidade atual gera uma rota de custo baixo. Foram inseridas observações sobre este fato nas respostas dos alunos, mas tais questionamentos não tiveram réplica pela parte dos mesmos.

## **7.2 Desenvolvimento de Heurísticas Construtivas**

### **7.2.1 Heurísticas Construtivas: Introdução e Exemplos**

Esta aula tinha como objetivo apresentar o que são as heurísticas construtivas, como elas podem ser concebidas, além de formalizar o algoritmo do Vizinho Mais Próximo.

<b>Ponto 1</b>	Discussão sobre o que são Atividades de Colaboração
<b>Tempo</b>	20 minutos
<b>Observações</b>	Como comentado na seção anterior, o baixo número de respostas frente à Atividade de Colaboração 2 levou a um replanejamento desta aula, onde este ponto específico foi inserido. Nos primeiros vinte minutos da aula presencial, foi discutido com os alunos o que é uma atividade de colaboração, bem como sua importância na condução da disciplina. Os motivos alegados pelos alunos para não responderem à

	<p>atividade foram (total de 28 alunos, sendo que quatro haviam respondido):</p> <ul style="list-style-type: none"> <li>- 04 alunos: não valia nota;</li> <li>- 10 alunos: não se lembraram;</li> <li>- 07 alunos: tinham outros compromissos;</li> <li>- 03 alunos: não compareceram.</li> </ul> <p>Para os alunos que achavam que não era importante responder, pois não valia nota, o argumento utilizado pelo professor foi que a atividade de colaboração com a aula que será ministrada, antes que ela ocorra, faz parte do processo de ensino e, como tal, deveria ser levada tão a sério quanto o comportamento esperado pelos alunos dentro do ambiente universitário, como uma sala de aula. Este mesmo argumento foi utilizado para os alunos que alegavam ter outros compromissos, pois como o semestre estava no início, provas e trabalhos finais não eram empecilhos para a realização da atividade que, conforme todos acabaram concordando, não precisava de um tempo excessivamente longo para ser respondida. Finalmente, em relação aos alunos que se esqueceram da atividade, os mesmos foram lembrados que a colaboração havia sido proposta há apenas uma semana e que estava no ambiente AMEM. Aqui, porém, cabe salientar uma proposta que foi colocada por dois alunos: na tela inicial do ambiente, quando o aluno se loga (ou ao entrar em uma disciplina), é apresentada uma tela com um resumo do que ele precisa realizar (número de mensagens recebidas, número de atividades não respondidas em cada disciplina). No entanto, as atividades de colaboração não são listadas. A possibilidade de visualizar este tipo de atividade ao entrar em uma disciplina ou no sistema é uma facilidade que pode auxiliar na condução das aulas.</p>
--	--

<b>Ponto 2</b>	Discussão sobre a Atividade de Colaboração 2
<b>Tempo</b>	20 minutos
<b>Metodologia</b>	Os quatro algoritmos apresentados pelos alunos foram discutidos em sala de aula, onde suas similaridades e diferenças foram ressaltadas.
<b>Observações</b>	A apresentação dos algoritmos, muito mais do que servir como base para a formalização da heurística do vizinho mais próximo, tinha como objetivo discutir os aspectos cruciais de um algoritmo construtivo: ponto de partida, vizinhança, critério de escolha, forma de inserção e critério de parada.
<b>Ponto 3</b>	Heurísticas Construtivas
<b>Tempo</b>	40 minutos
<b>Metodologia</b>	<p><b>DI:</b> quais são as características necessárias para definir um algoritmo construtivo?</p> <p><b>MSEM:</b> definição de ponto de partida, vizinhança, critério de escolha, forma de inserção e critério de parada.</p> <p><b>DA:</b> como é possível formalizar os algoritmos apresentados utilizando-se das definições discutidas?</p>
<b>Observações</b>	Durante este ponto, foi discutida com os alunos a formalização de algoritmos para o Problema do Caixeiro Viajante. Questões relativas à vizinhança também foram abordadas, utilizando-se do LOBO, pois o

objeto de aprendizagem permitia visualizar graficamente a definição de vizinhos através da distância euclidiana entre dois pontos.

Para esta aula, também era previsto a apresentação da Heurística Construtiva do Vizinho Mais Próximo, mas com a inserção do primeiro ponto, a discussão foi levada para a próxima aula. Como forma de corroborar a importância da atividade de colaboração e estabelecer a ligação entre as aulas, foi proposta uma nova atividade de colaboração, que não estava prevista.

<b>Colaboração 3</b>	É possível formalizar os algoritmos apresentados pelos alunos como uma heurística construtiva?
<b>Tempo</b>	Até o próximo encontro
<b>Descrição</b>	Considerando a formulação do PCV e os quatro algoritmos apresentados pelos colegas, tente formalizar os mesmos utilizando-se das definições discutidas em sala de aula.

*Análise da Colaboração 3 – apresentada e discutida na aula Heurísticas Construtivas: Vizinho Mais Próximo e Algoritmos de Inserção*

Depois do debate sobre a importância das atividades de colaboração para o andamento da disciplina, todos os alunos responderam a questão apresentada. As formalizações foram levadas para a próxima de aula e serviram de ponto de partida para a construção das heurísticas que seriam discutidas.

Em relação à MDP, as perguntas escolhidas para serem respondidas nesta aula foram a C1 (*A abordagem dialógica-problematizadora potencializa o ensino de heurísticas e metaheurísticas?*) e a B3 (*Os estudantes têm assumido que os problemas vistos em aula realmente contribuem no processo de aprendizagem de heurísticas e metaheurísticas?*). Durante as aulas presenciais, a abordagem dialógica-problematizadora se mostrou, até então, bastante adequada e desafiadora aos alunos, que participavam ativamente das atividades. No entanto, o costume de continuar estudando fora do horário escolar, que já deveria estar estabelecido em alunos que estão completando um curso de graduação, se mostrou estranho para os mesmos. Conforme relatado no primeiro ponto da aula, as questões metodológicas tiveram que ser trazidas novamente à tona, para que parte do processo de dialogicidade, que deveria ocorrer fora do horário escolar, não ficasse esquecido ou subestimado durante o decorrer da disciplina. Desta forma, a resposta para a primeira questão colocada é que a abordagem *pode* ser utilizada nesta disciplina mas, talvez, os alunos não estejam preparadas para este tipo de metodologia. Neste caso, cabe o professor, dirigente do processo, estabelecer as normas e a conduta dos alunos.

### 7.2.2 Heurísticas Construtivas: Vizinho Mais Próximo e Algoritmos de Inserção

Esta aula apresenta as heurísticas do Vizinho Mais Próximo e Inserção Mais Distante.

**Ponto 1** Discussão sobre a Atividade de Colaboração 3

<b>Tempo</b>	20 minutos
<b>Metodologia</b>	Apresentação das formalizações realizadas pelos alunos, ressaltando suas semelhanças e diferenças.

**Ponto 2** A Heurística do Vizinho Mais Próximo

**Tempo** 30 minutos

**Metodologia** **DI:** como pode ser definido o *vizinho mais próximo* em um PCV?  
**MSEM:** formulação algorítmica da heurística do Vizinho Mais Próximo.  
**DA:** a heurística do Vizinho Mais Próximo é adequada para todas as instâncias de problemas?

**Observações** Através das discussões da atividade de colaboração 3, cujos algoritmos se aproximavam da heurística apresentada, foi possível definir formalmente o vizinho mais próximo. Em relação ao DI, os alunos utilizaram o LOBO para tentar definir o que seria o *vizinho mais próximo*. Além da noção de distância euclidiana, colocada graficamente pelo objeto de aprendizagem, alguns alunos levantaram questões relativas ao custo de um determinado arco. Neste ponto, questões mais aprofundadas sobre a implementação do próprio objeto tiveram destaque. A apresentação gráfica no LOBO (pontos representando cidades e arcos representando caminho entre as cidades) trouxe algumas dúvidas em relação à matriz de custos. Para alguns alunos, o LOBO distorcia a visão espacial das distâncias, pois o tamanho do arco que une duas cidades quaisquer estava diretamente relacionado ao custo e não a uma distância física. No entanto, como foi argumentado pelo professor e por outros alunos, a utilização de uma matriz de custos permitia a inserção de outras variáveis, além da distância física, para quantificar o custo envolvido em uma viagem. Além disso, a utilização de uma representação onde os pontos permanecessem fixos (como em um mapa) e o custo entre diversas cidades fosse diferenciado, traria elementos a mais na apresentação gráfica que dificultaria a leitura visual do objeto. Por fim, foi ressaltado que o resultado final da aplicação de uma heurística em um PCV é a seqüência de cidades a serem percorridas, bem como a distância total. A visualização gráfica é um acessório pedagógico, mas não é estritamente necessário.

Em relação ao DA, o LOBO foi utilizado pelos alunos para executarem diversas instâncias de problemas (já previamente cadastrados no ambiente), como outras instâncias da literatura, provenientes da biblioteca da TSPLib. Alguns alunos testaram a heurística em relação a outros algoritmos já cadastrados no LOBO, como a inserção mais distante, e perceberam que o resultado do vizinho mais próximo era pior. No entanto, as questões relativas aos caminhos auto-intersectantes não foram levantadas por nenhum dos alunos.

**Ponto 3** A Heurística da Inserção Mais Distante

**Tempo** 50 minutos

**Metodologia** **DI:** caminhos auto-intersectantes são mais caros em relação aos custos de uma instância de PCV?  
**MSEM:** desigualdade do triângulo  
**DI:** como formalizar um algoritmo que não permita a criação de

	<p>caminhos auto-intersectantes?  <b>MSEM:</b> formulação algorítmica da heurística da Inserção Mais Distante.  <b>DA:</b> existem outras modificações que podem ser utilizadas neste algoritmo?</p>
<p><b>Observações</b></p>	<p>Para o primeiro DI, os alunos que ainda não haviam testado as heurísticas do Vizinho Mais Próximo e da Inserção Mais Distante no LOBO, o fizeram para verificar as diferenças entre as duas abordagens. Somente de forma visual, o que os alunos relataram era uma sensível melhoria, em termos de custos, na utilização da segunda heurística. No entanto, o desafio era mais do que apenas perceber que um algoritmo que não permita caminhos auto-intersectantes apresente resultados melhores. Muitos alunos tentaram explicar o <i>porquê</i> deste fenômeno, utilizando-se de trigonometria. Estas tentativas foram utilizadas no MSEM, onde as propriedades da desigualdade do triângulo foram apresentadas (POOLE, 2004). Após, os alunos foram apresentados a outro desafio: formalizar um algoritmo que não permitisse a inserção de arcos que cruzassem outros arcos já inseridos. Os algoritmos formalizados pelos alunos utilizavam, na sua maioria, de listas de arcos inseridos ou ângulos entre os arcos; posicionamento global; divisão da área da simulação em uma matriz e armazenamento dos pontos construídos pelo algoritmo de desenho de linhas de Bresenham, entre outros. A segunda MSEM apresentada foi a formalização da heurística da inserção mais distante e sua simulação passo-a-passo no LOBO. Finalmente, como DA, que foi deixado para que os alunos realizassem até a próxima aula por falta de tempo, eles deveriam pesquisar e verificar que outras variações os algoritmos de inserção permitiam, pesquisando na literatura.</p>

As questões da MDP relacionadas para este ponto foram a B3 (*Os estudantes têm assumido que os problemas vistos em aula realmente contribuem no processo de aprendizagem de heurísticas e metaheurísticas?*) e B4 (*Os alunos têm compreendido as questões de heurísticas e metaheurísticas com a prática de investigação-ação educacional, o LOBO e o AMEM?*). A primeira questão foi inserida neste ponto após a aula anterior, onde houve um debate sobre as atividades de colaboração. Após a discussão sobre a importância necessária das atividades fora do âmbito escolar ter sido realizada, era interessante perceber se a mesma influenciara no desempenho da próxima atividade. Como relatado anteriormente, a atividade de colaboração proposta foi prontamente realizada pelos alunos, demonstrando que os estudantes, mesmo que provisoriamente, novamente haviam assumido o papel de darem sua contribuição para o processo de condução da disciplina. Em relação à segunda questão orientadora, depois de algumas aulas utilizando o LOBO, era necessário verificar se os alunos realmente estavam aproveitando o objeto dentro da sala de aula e se o mesmo contribuía para o entendimento das questões discutidas. Ao final da segunda MSEM do terceiro ponto, foi apresentada aos alunos a seguinte questão:

*O LOBO auxilia no entendimento das heurísticas e metaheurísticas?*

Neste primeiro momento, dezoito dos vinte e um alunos presentes na sala de aula disseram que se tornava mais *fácil* entender as heurísticas apresentadas através das simulações realizadas e três acreditavam que o objeto não representava um diferencial em relação à explicação do professor. Inquiridos mais profundamente sobre o assunto, os alunos argumentaram que a discussão em sala de aula com o professor era muito mais importante do que a utilização do LOBO para o entendimento dos algoritmos. Neste ponto, o contra-argumento utilizado pelo professor foi que a resposta não era condizente com a pergunta, pois não estava sendo avaliado se o professor deveria ou não ministrar uma aula presencial sobre o assunto, mas sim se o LOBO auxiliaria neste processo. Mesmo assim, estes alunos acreditavam que a não existência do objeto não influenciaria na sua compreensão dos algoritmos. É importante salientar que, mesmo que a grande maioria tenha aprovado a utilização do LOBO nas aulas de heurísticas, os alunos ainda estavam tendo os primeiros contatos com o *software* e, desta forma, ainda há um peso relativo ao impacto inicial do sistema que deve ser considerado.

### 7.2.3 Heurísticas Construtivas: Algoritmos de Inserção e Heurística das Economias

Esta aula apresenta as heurísticas da Inserção Mais Barata e Mais Próxima e a Heurística das Economias.

<b>Ponto 1</b>	Heurísticas de Inserção Mais Barata e Mais Próxima
<b>Tempo</b>	50 minutos
<b>Metodologia</b>	<b>DI:</b> existem outras modificações que podem ser utilizadas na heurística da Inserção Mais Distante? <b>MSEM:</b> formalização das heurísticas da Inserção Mais Barata e Mais Próxima.
<b>Observações</b>	O DI deste ponto é o mesmo DA do ponto da última aula, que foi deixado como uma atividade de colaboração para os alunos. Através da bibliografia da disciplina, os alunos trouxeram formulações bastante semelhantes à MSEM desenvolvida logo após. Questões relativas a como pequenas modificações no algoritmo podem trazer profundas variações no resultado final foram abordadas, o que serviu como alerta para que os alunos realizassem testes exaustivos nos seus algoritmos, quando da implementação dos mesmos.

<b>Ponto 2</b>	A Heurística das Economias
<b>Tempo</b>	50 minutos
<b>Metodologia</b>	<b>DI:</b> é possível adaptar uma heurística construtiva desenvolvida para outro problema para o PCV? <b>MSEM:</b> formalização da Heurística das Economias <b>DA:</b> que critérios devem ser utilizados para comparar as diversas heurísticas construtivas?
<b>Observações</b>	Para o DI, foi apresentado um exerto do texto retirado de (CAMPELLO, MACULAN, 1994) contendo a descrição da Heurística das Economias, desenvolvida por Clarke e Wright, para o Problema do Roteamento dos Veículos. Baseados neste texto, os alunos deveriam formalizar uma adaptação do mesmo para o PCV, o que foi realizado durante a MSEM. Já mais acostumados com o tipo de formulação necessária, a

transposição da heurística para as necessidades do problema proposto foi realizada de forma bastante tranqüila pelos alunos. O DA, por sua vez, já previa que, como esta seria a última aula relativa às heurísticas construtivas, alunos e o professor deveriam formar uma base comum para avaliação dos algoritmos que deveriam ser implementados pelos alunos, na Atividade 2, descrita a seguir. Os parâmetros de comparação escolhidos foram: tempo computacional (com descrição da máquina utilizada) e qualidade da solução (custo total da rota encontrada).

<b>Colaboração 4</b>	Quais são as limitações da busca de soluções com as heurísticas construtivas?
<b>Tempo</b>	Até o próximo encontro
<b>Descrição</b>	Na próxima aula serão apresentadas as heurísticas de melhoria. Elas existem porque existem limitações na utilização das heurísticas construtivas, que devem ser tratadas à custa de um maior esforço computacional.
<b>Bibliografia</b>	(LAWLER <i>et al.</i> , 1985)

*Análise da Colaboração 4 – apresentada e discutida na aula Heurísticas de Melhoria: Introdução e Exemplos*

Ao finalizar as aulas sobre as heurísticas construtivas, era necessário verificar quais eram suas limitações e onde estavam os problemas que poderiam ser corrigidos com outras técnicas, tais como já observado na seção 2.3.2.

<b>Atividade 2</b>	Desenvolvimento de uma Heurística Construtiva
<b>Tempo</b>	03 semanas
<b>Questão</b>	Através do <i>framework</i> do LOBO, desenvolva uma heurística construtiva e realize testes com todas as instâncias disponíveis no objeto.

*Análise da Atividade 2 – apresentada e discutida durante a aula Metaheurísticas: Introdução*

A segunda atividade da disciplina requeria que os alunos, divididos em grupos, implementassem uma heurística construtiva baseada no *framework* do LOBO. As heurísticas foram sorteadas entre os grupos e os alunos tiveram acesso ao código do LOBO, porém sem os métodos das heurísticas já desenvolvidas. Nesta mesma aula, o código-fonte do objeto foi discutido em detalhes para que os alunos pudessem iniciar imediatamente a implementação das heurísticas.

O objetivo desta atividade era verificar se os alunos tinham condições de implementar uma heurística construtiva para o PCV a partir das teorias discutidas em sala de aula. A definição prévia dos critérios de comparação entre os algoritmos desenvolvidos trouxe para aulas elementos de pesquisa, onde questões acerca de metodologia científica e realização de testes confiáveis foram levantadas à tona. Em relação aos critérios de avaliação do professor, foi deixado claro que o desempenho dos algoritmos não seria o

elemento crucial, pois as heurísticas, por si só, apresentam performance diferentes. Os critérios utilizados foram:

- cumprimento dos prazos;
- utilização correta da metodologia de desenvolvimento (orientação à objetos);
- correte do método desenvolvido;
- análise dos resultados.

Em relação ao primeiro critério, todos os grupos cumpriram fielmente os prazos, entregando os algoritmos no tempo pré-definido. Da mesma forma, todos os grupos construíram seus códigos dentro do *framework* do LOBO utilizando a metodologia de orientação à objetos, o que permitiu importar diretamente o código desenvolvido para o ambiente de teste do professor. Em relação à correte dos métodos desenvolvidos, alguns grupos apresentaram problemas em relação a implementação, cujos algoritmos não realizavam exatamente o que deveriam fazer, face à formalização definida nas aulas anteriores. Correções foram apontadas para os alunos, que modificaram prontamente seus algoritmos. Por fim, todos os grupos também apresentaram um relatório com a análise dos resultados.

Estes relatórios foram discutidos em sala de aula, após a entrega dos trabalhos e análise pelo professor. Como era possível imaginar, houve uma discrepância muito grande em relação ao tempo computacional, pois o parque de máquinas utilizado pelos alunos era diferenciado. Tal situação já era esperada pelo professor e serviu para balizar comentários acerca da correta escolha de parâmetros de comparação entre pesquisadores.

Questionados pelo professor, o principal ponto que os alunos consideravam positivo na metodologia do desenvolvimento do trabalho foi a disponibilização do *framework*. A possibilidade de utilizar um ambiente gráfico para testar o algoritmo que estavam desenvolvendo, além dos resultados textuais apresentados na Janela de Resultados, auxiliou em todo o processo de desenvolvimento da atividade. Ao retirar os aspectos auxiliares do desenvolvimento de heurísticas (abertura de arquivos, impressão de resultados, visualização da rota) dos alunos, eles podem se concentrar no desenvolvimento da atividade em si, diminuindo a redundância de códigos (vários grupos construindo rotinas semelhantes) e melhorando o esforço coletivo em prol dos algoritmos.

Outra questão importante relativa à esta atividade é que ela servia como base para a implementação e o desenvolvimento de testes nas heurísticas de melhoramento e metaheurísticas. Desta forma, os alunos deveriam realizar todas as modificações requeridas pelo professor, quando fosse o caso, e disponibilizar o código para os demais alunos. Este banco de códigos seria utilizado, posteriormente, nas atividades 3 e 4.

Em relação à MDP, foram escolhidas as perguntas C2 (*Quais são as dificuldades e os avanços do ensino de heurísticas e metaheurísticas utilizando o Problema do Caixeiro Viajante?*) e C3 (*Será que o Problema do Caixeiro Viajante é adequado para o ensino de heurísticas e metaheurísticas?*), pois este era o primeiro marco divisor da disciplina, onde os alunos passariam da exploração do LOBO e da teoria para a implementação prática. No decorrer da explicação da atividade 4, ambas as questões, exatamente como foram formuladas na MDP, foram colocadas para os alunos. Para a questão C2, as opiniões foram divergentes, em relação aos alunos. Cerca de metade deles acreditava que trabalhar com um problema único era interessante e motivador, enquanto os demais tinham a impressão

(palavra utilizada pelos alunos) que um maior número de problemas poderia ser enriquecedor à disciplina. É importante salientar que, ao discutir estas questões, foi pontuado a preocupação de alguns alunos referente ao POSCOMP – Exame Nacional para Ingresso na Pós-Graduação em Computação, uma prova, teórica e prática, que objetiva testar os conhecimentos na área de computação e avaliar os candidatos a programas de pós-graduação em computação oferecidos no Brasil (fonte: Portal de Informações da SBC – <http://www.sbc.org.br>). Nesta prova, a área de Inteligência Artificial é abrangida pelo exame de Tecnologia da Computação. Não há uma ementa pré-definida por área, nem uma bibliografia indicada para a realização da prova.

A discussão, neste caso específico, foi centrada nas relações entre os diversos problemas que envolvem heurísticas e metaheurísticas. É necessário, para que o aluno possa implementar corretamente uma solução para um problema, que ele possua um conhecimento aprofundado sobre o mesmo, pois as heurísticas, como discutidas no capítulo 2, advém do pensamento criativo sobre soluções algorítmicas. A utilização de vários problemas em uma disciplina poderia acarretar na implementação superficial de soluções, pois não haveria tempo hábil para o aprofundamento adequado e necessário para que as questões intrínsecas às diversas heurísticas e metaheurísticas pudessem ser discutidas. Por outro lado, a teoria sobre o desenvolvimento dos diversos algoritmos não sofre modificações conceituais de um problema para outro. Como foi exemplificado para os alunos, é possível generalizar um algoritmo construtivo, de melhoramento ou uma metaheurística.

Em relação à questão C3, os alunos foram unânimes em perceber a importância do problema apresentado (PCV) e todos acreditavam que ele poderia ser utilizado como objeto de aprendizagem durante a disciplina.

Analisando o relato acima, reafirmamos a convicção de que o PCV pode ser utilizado como um problema tipo para o ensino de heurísticas e metaheurísticas. Como avanço interessante, neste estágio da disciplina, percebia-se a apropriação do jargão da área, onde as discussões já ocorriam relacionando arcos, nós, vizinhança e formas de inserção em algoritmos construtivos.

### 7.3 Desenvolvimento de Heurísticas de Melhoramento

#### 7.3.1 Heurísticas de Melhoramento: Introdução e Exemplos

Esta aula tinha como objetivo apresentar o que são as heurísticas de melhoramento, para que servem e como elas podem ser concebidas.

<b>Ponto 1</b>	Discussão sobre a Atividade de Colaboração 4
<b>Tempo</b>	20 minutos
<b>Metodologia</b>	Apresentação dos textos produzidos pelos alunos e formalização dos conceitos de platô, escarpa, mínimo local e mínimo global.
<b>Ponto 2</b>	Heurísticas de Melhoramento
<b>Tempo</b>	50 minutos
<b>Metodologia</b>	<b>DI:</b> como melhorar uma rota encontrada por uma heurística construtiva?

	<b>MSEM:</b> apresentação de um algoritmo de melhoramento genérico
<b>Observações</b>	Neste ponto, o LOBO é utilizado para demonstrar que as heurísticas de melhoramento, usualmente, realmente modificam a rota inicial encontrada pelas heurísticas construtivas, diminuindo o valor final da mesma (encontrando um novo mínimo local e, potencialmente, global). Aqui, são explorados os conceitos de <i>rota inicial</i> , troca de movimentos, busca local em vizinhança e critérios de parada.

A primeira aula das heurísticas de melhoramento utilizou intensamente o LOBO como forma de visualizar o ganho de desempenho ao inserir um algoritmo que modificasse a solução inicial construída anteriormente. Para esta aula, a questão da MDP escolhida foi a B1 (*Os alunos efetivamente estão aproveitando os operacionalizadores pedagógicos e o LOBO para o entendimento das heurísticas e metaheurísticas?*). Como a aula teve um caráter bastante prático, observou-se a atenção dispensada nos alunos para a realização das simulações que eram proposta pelo professor. As observações realizadas apresentaram uma situação bastante diversificada em relação aos estudantes: enquanto alguns se dedicavam exclusivamente à utilização do simulador, outros estavam com diversas janelas abertas, executando muitas tarefas ao mesmo tempo e, ainda, alguns poucos *não* estavam com o simulador aberto, se dedicando exclusivamente a outras tarefas. Esta situação, mesmo repreendida pelo professor durante a aula (principalmente aos que não estavam executando as simulações) ocorreu de foram semelhante em outras aulas práticas. Em um primeiro momento, seria possível imaginar que os alunos não estivessem utilizando o objeto e, desta forma, seu desenvolvimento não atingiria os objetivos propostos. No entanto, também deve ser analisado que, quando as tarefas exigidas deveriam ser feitas de uma aula para outra, a grande maioria dos alunos realizava as simulações e discutia os resultados. A questão, deste modo, transforma-se do *será que os alunos estão utilizando realmente o LOBO para compreender as heurísticas e metaheurísticas para os alunos querem e/ou necessitam utilizar o LOBO em sala de aula?*

Questionados sobre o assunto, os alunos, quase na sua totalidade, se mostraram interessados em continuar com as aulas práticas, utilizando o objeto em sala de aula. Alguns argumentaram que poderiam *fazer mais de uma coisa ao mesmo tempo* (expressão deles). Como era de se esperar, os alunos que não estavam realizando a tarefa requerida não fizeram manifestações sobre o assunto, mesmo quando inquiridos diretamente.

Desta forma, a resposta para esta questão, neste momento, seria que os alunos estão utilizando, pelo menos, de forma parcial o LOBO como uma ferramenta capaz de auxiliar a compreensão das heurísticas e metaheurísticas.

### 7.3.2 Heurísticas de Melhoramento: 2-Opt, 3-Opt e Or-Opt

Esta aula tinha como objetivo apresentar as heurísticas de melhoramento 2-Opt, 3-Opt e Or-Opt.

<b>Ponto 1</b>	Heurística de Melhoramento 2-Opt
<b>Tempo</b>	40 minutos
<b>Metodologia</b>	<b>DI:</b> como formalizar um algoritmo que realize a troca entre dois arcos quaisquer de uma rota? <b>MSEM:</b> formalização da heurística de melhoramento 2-Opt

	<b>DA:</b> é possível realizar trocas com mais de dois arcos?
<b>Observações</b>	A heurística de melhoramento 2-Opt (apresentada na seção 6.2.4) é discutida com os alunos no DI, onde as questões relativas à vizinhança e os arcos possíveis de serem trocados é levantada (não é possível trocar dois arcos consecutivos, pois o ciclo hamiltoniano seria desfeito). Neste ponto, o LOBO é utilizado para demonstrar, passo-a-passo, como a heurística 2-Opt funciona, assim como suas variações.
<b>Ponto 2</b>	Heurística de Melhoramento 3-Opt
<b>Tempo</b>	20 minutos
<b>Metodologia</b>	<b>DI:</b> é possível realizar trocas com mais de dois arcos? <b>MSEM:</b> formalização da heurística de melhoramento 3-Opt
<b>Observações</b>	A DI deste ponto é a mesma DA do ponto anterior, realizando a ligação entre as duas heurísticas de melhoramento, que possuem muitos elementos semelhantes entre si. A heurística de melhoramento 3-Opt (apresentada na seção 6.2.4) é discutida com os alunos, bem como a impossibilidade prática da realização de trocas do tipo 4-Opt ou superior (LAWLER <i>et al.</i> , 1985). Assim como no ponto anterior, o LOBO aqui também é utilizado para demonstrar como funciona a heurística 3-Opt, tanto passo-a-passo como suas variações e a diferença no desempenho.
<b>Ponto 3</b>	Heurística de Melhoramento Or-Opt
<b>Tempo</b>	30 minutos
<b>Metodologia</b>	<b>DI:</b> trocar um <i>pedaço</i> de uma rota de uma posição para outra, em uma solução, pode trazer ganhos significativos de melhoria de custo? <b>MSEM:</b> formalização da heurística de melhoramento Or-Opt <b>DA:</b> que critérios devem ser utilizados para avaliar uma heurística de melhoramento?
<b>Observações</b>	A heurística de melhoramento Or-Opt (OR, 1976) é apresentada como uma modificação na heurística 3-Opt, cujo desempenho é semelhante, apesar do número reduzido de cálculos (e, conseqüentemente, do tempo envolvido). Assim como na última aula das heurísticas construtivas, o DA deste ponto leva a pensar sobre o desenvolvimento da atividade 3.
<b>Colaboração 5</b>	Quais são as limitações da busca de soluções com as heurísticas de melhoramento?
<b>Tempo</b>	Até o próximo encontro
<b>Descrição</b>	Na próxima aula serão apresentadas as metaheurísticas. Elas existem porque existem limitações na utilização das heurísticas de melhoramento, que devem ser tratadas à custa de um maior esforço computacional.
<b>Bibliografia</b>	(DIAZ <i>et al.</i> , 1996)

*Análise da Colaboração 5 – apresentada e discutida na aula Metaheurísticas: Introdução*

Ao finalizar as aulas sobre as heurísticas de melhoramento, são retomadas as discussões sobre as escarpas, platôs e mínimos locais, pois as limitações dos algoritmos de troca servem como base para justificar a existência das metaheurísticas.

<b>Atividade 3</b>	Desenvolvimento de uma Heurística de Melhoramento
<b>Tempo</b>	04 semanas
<b>Questão</b>	Através do <i>framework</i> do LOBO, desenvolva uma heurística de melhoramento e realize testes com todas as instâncias e todos os algoritmos construtivos disponíveis no objeto.

*Análise da Atividade 3 – apresentada e discutida durante a aula Metaheurísticas: Operadores de Recombinação e Mutação para o PCV*

A terceira atividade da disciplina requeria que os alunos, divididos em grupos, implementassem uma heurística de melhoramento baseada no *framework* do LOBO. Da mesma forma que a atividade 2, as heurísticas foram sorteadas entre os grupos e os alunos tiveram acesso ao código do LOBO, porém sem os métodos das heurísticas já desenvolvidas. Implementar uma heurística de melhoramento traz desafios diferenciados e serve como balizador para verificar se as teorias discutidas em sala de aula poderiam ser eficientemente desenvolvidas em um trabalho prático.

Em relação aos critérios de comparação entre os algoritmos desenvolvidos e aos critérios de avaliação do professor, ambos seguiram a mesma metodologia apresentada na atividade 2.

Não houve atrasos em relação a entrega dos trabalhos, mas houve problemas um pouco maiores na correção dos métodos desenvolvidos. Dos seis trabalhos entregues, metade continha erros no desenvolvimento da implementação. Da mesma forma que os anteriores, os equívocos foram apontados e as soluções discutidas com os alunos, que se comprometeram em sanar os problemas.

A análise dos resultados foi realizada pelos alunos, que deveriam realizar testes do algoritmo de melhoramento que estava sendo implementado sob o resultado de todos os algoritmos construtivos desenvolvidos pelos grupos. Desta forma, cada grupo deveria realizar trinta simulações (01 algoritmo de melhoramento \* 06 algoritmos construtivos \* 05 instâncias de problemas).

Ao trazer e comparar os resultados em sala de aula, foi possível estabelecer algumas medidas de performance entre os diversos algoritmos, notadamente centrado na questão da vizinhança para os algoritmos 2-Opt e 3-Opt. Como encontrado na literatura (LAWLER *et al.*, 1985), o desenvolvimento de esquemas de vizinhança pode diminuir o tempo de processamento sem perda de qualidade. Como os resultados computacionais eram diversos, devido as máquinas diferenciadas, o professor optou por realizar testes *in loco*, simulando os algoritmos entregues em sala de aula, na mesma máquina, para demonstrar os fatos relacionados acima.

Em relação a esta aula, foi escolhida a questão C2 (*Quais são as dificuldades e os avanços do ensino de heurísticas e metaheurísticas utilizando o Problema do Caixeiro Viajante e o LOBO?*), pois neste momento, o segundo tipo de heurísticas haviam sido

apresentados para os alunos e, na próxima aula, os primeiros resultados práticos seriam apresentados. As constatações, até então, eram:

- houve um certo deslumbramento, por parte dos alunos, com o LOBO no início da disciplina, pelo potencial das simulações apresentadas. Gradativamente, o objeto deixou de ser uma novidade para se transformar em uma ferramenta. Quando do estabelecimento da primeira atividade, que exigia utilizar o LOBO como um *framework*, este novamente *retomou* a sua importância inicial. Este processo é natural com uma nova tecnologia (MORAN, 1997) e deve ser encarado dentro das próprias limitações do mesmo, assim como seus objetivos;
- a apresentação de resultados com custo menor pelas heurísticas de melhoramento trouxe um novo ânimo em relação à motivação dos alunos pelo Problema do Caixeiro Viajante. A inserção de somente dois algoritmos construtivos no LOBO fez com que as aulas sobre as heurísticas de inserção não tivessem a mesma dinâmica das primeiras aulas, onde o objeto foi utilizado com mais ênfase;
- de forma um tanto quanto habitual, os alunos haviam deixado a implementação da atividade 2 para os últimos dias, o que acabou atrapalhando a condução das duas últimas aulas, pois muitos queriam tirar dúvidas durante e após o período de encontro. Uma metodologia de acompanhamento mais diretiva em relação aos trabalhos talvez devesse ser adotada para diminuir esta dificuldade.

## 7.4 Desenvolvimento de Metaheurísticas

### 7.4.1 Metaheurísticas: Introdução

A primeira aula de metaheurísticas trabalha com a quinta atividade de colaboração, onde os alunos deveriam discutir por que os platôs, escarpas e mínimos locais ainda poderiam ocorrer durante a utilização de uma heurística de melhoramento. A impossibilidade de realizar um passo de retrocesso (algoritmo guloso) foi a característica mais lembrada e discutida com os alunos.

<b>Ponto 1</b>	Discussão e Entrega da Atividade 2
<b>Tempo</b>	20 minutos
<b>Ponto 2</b>	Discussão sobre a Atividade de Colaboração 5
<b>Tempo</b>	20 minutos
<b>Metodologia</b>	Discussão sobre as colaborações elaboradas pelos alunos, assim como suas dúvidas sobre o que é uma metaheurística.
<b>Ponto 3</b>	Introdução às Metaheurísticas
<b>Tempo</b>	50 minutos
<b>Metodologia</b>	<p><b>DI:</b> é possível evitar as armadilhas que usualmente ocorrem quando da utilização de um algoritmo de busca local?</p> <p><b>MSEM:</b> definição teórica de uma metaheurística.</p> <p><b>DA DI:</b> como sair de um mínimo local?</p> <p><b>MSEM:</b> definição formal de uma metaheurística genérica.</p> <p><b>DA:</b> quais são as metaheurísticas que podem ser aplicadas ao PCV?</p>

<b>Observações</b>	Este ponto é dividido em dois momentos. No primeiro momento, são explorados com os alunos as questões relativas as características de um algoritmo de busca local e o espaço de busca de soluções, onde o conceito de vizinhança novamente é reforçado. A partir da primeira MSEM, propomos aos alunos que pensem em como sair de um mínimo local, ou seja, como “escapar” de uma solução cujas soluções vizinhas são, todas, de custo mais elevado. Relembrando que estes alunos estão cursando o sétimo semestre do Curso de Ciência da Computação e que no segundo e terceiro semestres eles abordam disciplinas que trabalham com grafos e pesquisa e ordenação de dados, foi fácil verificar que as propostas dos alunos para realizar a saída de um mínimo local eram baseadas nas teorias de caminhamento em árvores ( <i>backtracking</i> ). De forma mais ou menos completa, a maioria dos alunos abordou o assunto por este prisma. Uma dupla de alunos apresentou um algoritmo já formalizado, baseado em uma estrutura de pilha que, no seu arcabouço teórico, remetia à busca em largura. Considerando as proposições dos alunos, um debate foi realizado para definir como uma metaheurística genérica poderia ser formalizada, conceitualizando novamente as estruturas de soluções, vizinhança, solução inicial, procedimentos de busca local, solução incumbente e solução atual. Desta forma, este processo foi a segunda MSEM. O DA, que será utilizado como atividade de colaboração, instigava os alunos a pesquisarem mais sobre as metaheurísticas que seriam abordadas no decorrer do semestre.
<b>Colaboração 6</b>	Quais são as metaheurísticas que podem ser aplicadas ao PCV?
<b>Tempo</b>	Até o próximo encontro
<b>Descrição</b>	Nas próximas aulas serão apresentadas algumas metaheurísticas que podem ser utilizadas para resolver o PCV. Pesquise na bibliografia indicada como estas metaheurísticas tentam resolver o problema do mínimo local.
<b>Bibliografia</b>	(DIAZ <i>et al.</i> , 1996)(BURIOL, 2000)(KÖHLER, 2004)

*Análise da Colaboração 6 – apresentada e discutida durante a aula Metaheurísticas: Simulated Annealing*

Alguns alunos já haviam manifestado interesse sobre a área de Algoritmos Genéticos e, desta forma, não foi surpresa que as respostas ligadas a esta abordagem tenham sido desenvolvidas de forma mais ampla. Como o LOBO tem implementado as metaheurísticas Busca Tabu e *Simulated Annealing*, estes dois métodos também foram bastante discutidos, tanto utilizando o material do próprio objeto como da bibliografia apresentada. No entanto, percebeu-se que, neste ponto, a reprodução das informações bibliográficas foi a regra, e não a exceção, como ocorrera nas atividades colaborativas relacionadas aos algoritmos construtivos e de melhoramento. Tal fato pode ser explicado, primeiramente, pela complexidade do tema. As metaheurísticas, tanto conceitualmente como na prática, trazem dificuldades adicionais para seu entendimento, pois as teorias que a cercam requerem que os conceitos anteriores estejam bem definidos e entendidos pelos

alunos. Além disso, alguns alunos se mostraram um tanto céticos quanto ao bom desempenho das metaheurísticas, pois tanto o *Simulated Annealing* quanto os Algoritmos Genéticos contém uma boa dose de variáveis probabilísticas. Para os alunos que criticaram o *Simulated Annealing*, foi pedido para que testassem o mesmo utilizando o LOBO. Em relação ao segundo grupo, a abordagem utilizada foi refutar as declarações através do grupo que estava estudando os algoritmos genéticos há mais tempo. A discussão entre os dois grupos levantou algumas questões que foram utilizadas, mais tarde, quando da apresentação dos Algoritmos Genéticos.

Esta aula também foi marcada pela entrega e apresentação da primeira atividade prática (atividade 2), onde os grupos de alunos deveriam desenvolver heurísticas construtivas. Para a MDP, foi escolhida a pergunta D2 (*Como verificar se os alunos estão compreendendo o conteúdo de heurísticas e metaheurísticas que está sendo desenvolvido através da investigação ação-educacional, do LOBO e do AMEM?*) e B4 (*Os alunos têm compreendido as questões de heurísticas e metaheurísticas com a prática de investigação-ação educacional, o LOBO e o AMEM?*). Em relação a primeira pergunta, foram definidos com os alunos, desde o início das aulas, que haveriam três pontos de verificação principais (atividades 2, 3 e 4) e que outros processos poderiam ser utilizados no decorrer da disciplina. A apresentação da primeira atividade representa a oportunidade do professor verificar o resultado da soma das habilidades computacionais predecessoras (programação) com a teoria desenvolvida em sala de aula. Como relatado anteriormente, os grupos apresentaram resultados consistentes na sua implementação e a discussão dos resultados foi bastante esclarecedora. A comparação dos algoritmos em sala de aula, além de enriquecer o debate sobre o tempo computacional, permitiu que os alunos percebessem, de forma clara, as diferenças entre os diversos métodos desenvolvidos. Mesmo considerando que a apresentação de um trabalho não é a forma única e inequívoca de avaliar o desempenho estudantil, a boa condução desta primeira atividade salienta que as teorias sobre heurísticas e metaheurísticas foram assimiladas, no mínimo, em relação a uma correta implementação algorítmica.

Em relação à pergunta B4, os alunos foram questionados se o AMEM e a prática de desenvolvimento das aulas através de desafios motivadores influenciava positivamente, negativamente ou não tinha influência sobre o conteúdo e o desenvolvimento da atividade. Sobre o ambiente, houve unanimidade em acreditar que o mesmo auxiliava na comunicação e na disponibilização dos materiais didáticos, principalmente ressaltando a forma *organizada* (grifo deles) como as aulas estavam armazenadas, o que facilitaria a consulta às aulas predecessoras e as atividades que deveriam ser desenvolvidas. Dos seis grupos em que foram divididos os alunos, quatro acreditavam que o AMEM influenciou positivamente no desenvolvimento da atividade e dois relataram que não houve influência, negativa ou positiva. Para estes últimos, se houvesse uma página *html* simples com o conteúdo ou mesmo uma pasta em uma loja de fotocópias, a atividade seria desenvolvida da mesma maneira.

Este ponto de vista é importante de ser ressaltado em consideração a forma como foi planejada e conduzida a disciplina. Como se tratava de um curso presencial, não houve, com exceção das atividades de colaboração, nenhum momento não-presencial na disciplina e, deste modo, o AMEM não foi utilizado em suas outras potencialidades, tais como a utilização de salas de discussão. É claro que, em relação à organização da disciplina, da

manutenção do histórico das discussões, tanto pelo correio interno quanto pelos fóruns abertos, o ganho para o professor na realização das suas atividades de pesquisador-docente não são apresentadas para o aluno, que não percebe, de forma direta, como isso pode influenciar na disciplina.

Já em relação à metodologia aplicada diretamente em aula, através de desafios e construção da solução de forma dialógica, três alunos se mostraram reticentes em relação às mesmas. Os mesmos, contudo, eram os que participavam de forma menos colaborativa durante as aulas, falando somente o necessário e quando eram perguntados. Questionados mais fortemente sobre o assunto, eles assumiram que acreditavam que este tipo de aula *era de um professor que não queria dar aula*. Antes de o professor intervir, houve uma forte reação de alguns alunos, que argumentaram que era melhor trabalhar deste modo do que só ouvir um monte de *conversa fiada em cima de transparências*. Em pouco tempo, o debate foi desvirtuado sobre a metodologia de trabalho de diversos professores, até que houvesse a intervenção diretiva do docente. Mais uma vez, as questões relativas aos objetivos da disciplina e da metodologia foram explanadas e, para ilustrar a origem destas questões, o professor se comprometeu em deixar disponível para os alunos alguns textos para quem estivesse interessado em se aprofundar sobre o assunto (os textos escolhidos foram (FREIRE, 1987), (MÜLLER, DE BASTOS, 2004) e (DE BASTOS, MÜLLER, 1999)).

Como é possível observar, alguns alunos não acreditavam na forma de condução das aulas e, para os mesmos, a resposta à pergunta B4 seria negativa em relação à metodologia. No entanto, as observações realizadas em sala de aula admitem argumentar que a prática da investigação-ação e a condução dos momentos presenciais através dos três momentos dialógico-problematizadores permitiram que os assuntos fossem discutidos e aprofundados de forma mais objetiva. Em vários momentos, o professor teve que recorrer aos artigos científicos para exemplificar e responder aos questionamentos dos alunos, usualmente trazidos à tona na aula subsequente. Por outro lado, a prática de investigar a própria ação docente permitiu que redirecionamentos fossem realizados, como a inserção de um ponto na aula *Heurísticas Construtivas: Introdução e Exemplos* e os questionamentos sobre a utilização do LOBO nas aulas práticas.

#### 7.4.2 Metaheurísticas: *Simulated Annealing*

Esta aula tinha como objetivo discursar brevemente sobre as metaheurísticas em geral e, em particular, trabalhar com o algoritmo do *Simulated Annealing*.

<b>Ponto 1</b>	Discussão sobre a Atividade de Colaboração 6
<b>Tempo</b>	30 minutos
<b>Metodologia</b>	Debate sobre as metaheurísticas comentadas pelos alunos na atividade, com ênfase nas três que serão exploradas na disciplina.
<b>Ponto 2</b>	Metaheurística <i>Simulated Annealing</i>
<b>Tempo</b>	60 minutos
<b>Metodologia</b>	<b>DI:</b> como definir uma metaheurística que não utilize um procedimento de busca local? <b>MSEM:</b> formulação da metaheurística <i>Simulated Annealing</i> <b>DA:</b> a parametrização de uma metaheurística pode ser controlada por

	outra heurística?
<b>Observações</b>	<p>No DI, os alunos discutiram e apresentaram propostas para métodos que escolhessem, de forma algorítmica, trilhar caminhos que não fossem sempre o de <i>melhor solução atual</i>. Ficou claro, rapidamente, que alguma forma probabilística deveria ser usada. Alguns apresentaram métodos de decréscimo na função probabilística de aceitação de passos <i>ruins</i>, claramente inspirados no próprio <i>Simulated Annealing</i> que já havia sido discutida durante a atividade de colaboração e no primeiro momento da aula. Outros alunos propuseram a inserção de movimentos já realizados em uma lista, um movimento também inspirado em outra metaheurística conhecida, a Busca Tabu. A MSEM foi construída a partir destes argumentos e, quando as idéias não pertenciam ao método que estava sendo discutido, apontamentos para as próximas aulas foram deixados para os alunos, deixando claro que é possível utilizar de tais estratégias e que elas seriam discutidas oportunamente. Durante a construção da MSEM, o LOBO foi utilizado para a execução do <i>Simulated Annealing</i> em sala de aula, simulando diversas situações, com diferentes algoritmos construtivos, de melhoramento e em diferentes instâncias para o problema. No final, foi exposto um Desafio Mais Amplo, onde as questões relativas à parametrização do algoritmo deveriam ser discutidas com mais afinco. Como relatado em (AARTS <i>et al.</i>, 2006), um dos principais problemas do <i>Simulated Annealing</i> era a correta definição dos parâmetros, pois uma pouca variação nestes valores poderia acarretar em diferenças de desempenho. A maioria dos alunos respondeu ao desafio de forma teórica, especulando sobre a possibilidade de construir um <i>Simulated Annealing</i> para encontrar parâmetros para outro <i>Simulated Annealing</i>, o que fatalmente geraria um problema de recursividade eterna. Outros alunos trabalharam sobre o texto apresentado na bibliografia, onde experimentos são relatados e a parametrização é discutida sobre vários problemas.</p>

Considerando que esta aula apresentava a primeira metaheurística desenvolvida no LOBO, foram escolhidas as perguntas B1(*Os alunos efetivamente estão aproveitando os operacionalizadores pedagógicos e o LOBO para o entendimento das heurísticas e metaheurísticas?*) e B4(*Os alunos têm compreendido as questões de heurísticas e metaheurísticas com a prática de investigação-ação educacional, o LOBO e o AMEM?*). Conforme o relato da aula, o desenvolvimento da apresentação da metaheurística *Simulated Annealing* iniciou ainda com a Colaboração 6, onde os alunos tiveram um semana para ler os textos recomendados e discutir sobre o assunto no fórum apropriado. Grande parte dos conceitos do algoritmo foram trazidos pelos próprios alunos, ainda no DI do ponto 2. Durante a MSEM, o LOBO foi utilizado para demonstrar como a metaheurística funcionava. Como explicado anteriormente na seção 6.4.2, não era possível realizar uma simulação passo a passo de uma metaheurística, devido ao grande número de iterações necessárias. Desta forma, a utilização do objeto se restringia a comprovar que as metaheurísticas poderiam melhorar a solução encontrada até então. Nesta aula específica, ficou bastante claro que, para atingir os objetivos educacionais definidos, a estratégia e o ferramental tecnológico são igualmente importantes. Logo, para ambas as questões apresentadas, a resposta, neste ponto específico, estaria centrada na utilização dos operacionalizadores pedagógicos e a prática da investigação-ação educacional como balizadores principais do desenvolvimento realizado.

### 7.4.3 Metaheurísticas: Busca Tabu

Esta aula tinha como objetivo trabalhar com a metaheurística Busca Tabu.

<b>Ponto 1</b>	Metaheurística Busca Tabu
<b>Tempo</b>	90 minutos
<b>Metodologia</b>	<b>DI:</b> como definir uma lista com movimentos já utilizados e como esta poderia ser inserida em uma metaheurística? <b>MSEM:</b> formulação da metaheurística Busca Tabu <b>DA:</b> assim como a metaheurística <i>Simulated Annealing</i> , a Busca Tabu também é parametrizável. Como definir o número de iterações que cada movimento deve permanecer na Lista Tabu, para que não ocorra ciclos e, ao mesmo tempo, o algoritmo não consuma um tempo excessivo de máquina?
<b>Observações</b>	A DI deste ponto foi reformulada, levando em conta os argumentos levantados no DI da aula anterior. Com base nas idéias propostas anteriormente, os alunos foram desafiados a pensar focando a utilização de uma lista de movimentos, para evitar os mínimos locais. Durante a realização do MSEM foram utilizados conceitos descritos pelos próprios alunos, principalmente nas questões relativas ao problema da ciclagem (ver seção 6.2.5) e os parâmetros da Busca Tabu. Nesta fase, o LOBO foi utilizado para demonstrar o funcionamento da metaheurística e a exploração de seus parâmetros. O DA tinha como objetivo retomar as questões relativas à parametrização, mas neste caso, sob o prisma da Busca Tabu. Uma das idéias apontadas pelos alunos, que o professor discutiu com bastante atenção, foi a utilização de variáveis auxiliares para o armazenamento e impressão de resultados parciais da aplicação das metaheurísticas. Como será explorado mais adiante, durante a análise da última tarefa, a implementação de metaheurísticas traz desafios adicionais no que concerne a verificação dos alunos em relação à correteza dos algoritmos implementados, devido aos elementos probabilísticos inerentes às mesmas. A utilização de variáveis auxiliares que apresentassem passos intermediários da solução e do estado interno do problema é uma proposta interessante para perceber o comportamento do algoritmo e, possivelmente, encontrar erros de implementação. Tal fato, comentado pelos próprios alunos, foi exaustivamente recomendado durante a aula.

Novamente, o desenvolvimento desta aula foi balizado fortemente nas discussões realizadas para a colaboração 6. O LOBO, mais uma vez, foi utilizado como um acessório importante, mas a construção do algoritmo através das definições apresentadas ainda na aula anterior foram o ponto forte sob qual a aula foi desenvolvida.

### 7.4.4 Metaheurísticas: Algoritmos Genéticos

Os Algoritmos Genéticos representam uma classe de metaheurísticas de estrutura diferenciada em relação aos demais, pois são algoritmos populacionais, ou seja, apresentam o conceito de uma *população* de soluções, diferente das demais

metaheurísticas apresentadas, onde a solução normalmente é única. Desta forma, a discussão sobre este algoritmo foi realizada em duas aulas.

<b>Ponto 1</b>	Representação de Soluções
<b>Tempo</b>	40 minutos
<b>Metodologia</b>	<p><b>DI:</b> quais são as possíveis representações de uma solução para o PCV?</p> <p><b>MSEM:</b> estruturas de dados que configurem uma solução para um PCV</p> <p><b>DA:</b> quais destas estruturas se adequam melhor a manutenção de diversas soluções em um algoritmo computacional?</p>
<b>Observações</b>	<p>Neste ponto, os alunos deveriam raciocinar sobre outras formas de representação de uma solução, diferentes do vetor unidimensional que estava sendo utilizado até então. A discussão anterior à MSEM centrou-se em estruturas baseadas em listas, grafos e árvores. Durante a construção da MSEM se utilizou da bibliografia contida em (BURIOL, 2000) (KOZA, 1998) e (KÖHLER, 2004). Em relação ao DA, a maioria dos alunos, devido a sua experiência em disciplinas anteriores, se mostrou interessada em utilizar as estruturas baseadas em listas para armazenar diversas instâncias de soluções.</p>
<b>Ponto 2</b>	Algoritmos Populacionais
<b>Tempo</b>	60 minutos
<b>Metodologia</b>	<p><b>DI:</b> é possível aplicar os conceitos de Evolução Natural (Darwinismo) em um algoritmo?</p> <p><b>MSEM:</b> formalização de um algoritmo populacional e evolutivo genérico</p> <p><b>DA:</b> que operadores de recombinação e mutação poderiam ser utilizados para o PCV?</p>
<b>Observações</b>	<p>No início do DI, uma pequena recapitulação sobre a Evolução Natural e o Darwinismo foi necessária, pois nem todos os alunos tinham conhecimento sobre o tema. Utilizando de um debate aberto, as proposições principais de Darwin e sua Teoria da Evolução (DARWIN, 1987) foram discutidas com todos os presentes. Após esta explanação, as propostas apresentadas pelos alunos para o DI se centraram nas concepções iniciais de um Algoritmo Evolutivo, principalmente as relacionadas à população de soluções e a escolha dos <i>melhores pais</i> através do custo. No entanto, como combinar duas ou mais soluções foi um problema que não foi solucionado por nenhum grupo, com exceção dos alunos que já estavam estudando algoritmos genéticos há mais tempo. Para este grupo específico, o professor deixou proposadamente que suas idéias fossem apresentadas por último e, baseadas nestas e nos conceitos iniciais dos demais, a formulação de um algoritmo evolutivo genérico foi realizada, concentrando-se nas questões da representação, na escolha dos pais, na realização das operações de recombinação e mutação e nos critérios de parada. Ao final da aula, o DA foi formalizado aos alunos para que, até a próxima aula, eles trabalhassem com operadores específicos para o PCV, baseados na literatura recomendada.</p>

**Bibliografia** (BURIOL, 2000) (KOZA, 1998) (DIAZ, 1996) e (KÖHLER, 2004)

Esta primeira aula sobre os algoritmos genéticos(AG) teve a estratégia diferenciada em relação à condução da mesma. Como um grupo já estava estudando os AG a algum tempo, eles foram utilizados como uma espécie de tutores durante desafio inicial, nos dois momentos definidos. Em relação à MDP, foi escolhida a pergunta A3(*Como os professores têm construído a organização didática de suas aulas nas práticas de investigação-ação educacional de heurísticas e metaheurísticas?*). Com a estratégia de utilizar tutores espalhados durante a sala de aula para auxiliar os alunos na condução dos desafios iniciais, esta pergunta ganha importância na medida em que os alunos poderiam deixar de lado o trabalho e se concentrar somente na *explicação pronta* que os tutores poderiam trazer. No entanto, como a atividade de colaboração 6 debateu, mesmo que de forma superficial, as metaheurísticas que seriam apresentadas posteriormente, o que foi observado é que os alunos conseguiram debater com os tutores como, realmente, um Algoritmo Genético poderia funcionar, mesclando perguntas teóricas com questões de implementação. Tal observação baliza e demonstra a importância das atividades de colaboração e a organização prévia da disciplina, principalmente na condução de leituras e debate sobre o tema que será tratado posteriormente.

#### 7.4.5 Metaheurísticas: Operadores de Recombinação e Mutação para o PCV

Após o estabelecimento das proposições básicas sobre o que é um algoritmo populacional e sua estrutura básica, esta aula tinha como objetivos explorar os algoritmos genéticos e os operadores especialmente construídos para o PCV.

<b>Ponto 1</b>	Discussão e Entrega da Atividade 3
<b>Tempo</b>	20 minutos
<b>Ponto 2</b>	Operadores de Recombinação e Mutação para o PCV
<b>Tempo</b>	40 minutos
<b>Metodologia</b>	<b>DI:</b> que operadores de recombinação e mutação poderiam ser utilizados para o PCV? <b>MSEM:</b> apresentação dos operadores de recombinação e mutação trabalhados na bibliografia <b>DA:</b> os algoritmos genéticos trabalham com estruturas de dados contendo diversas instâncias de solução. Qual é a performance esperada, em relação ao tempo computacional, para esta estratégia?
<b>Observações</b>	O DI deste ponto é o mesmo DA da aula anterior e, durante a semana que antecedeu o encontro, alguns alunos procuraram o professor para tirar dúvidas sobre os operadores que estavam sendo estudados. Durante a aula presencial foi realizado um debate prévio, onde os grupos apresentaram a visão que eles tinham sobre os operadores, que foi posteriormente consolidada durante a MSEM, onde as últimas dúvidas foram sanadas sobre cada operador. O DA proposto se baseou no capítulo 8 do livro de Koza (1998)( <i>Amount of Processing Required to Solve a Problem</i> ), onde as questões relativas ao tempo de processamento e número de gerações necessárias para alcançar um

resultado com um determinado grau de probabilidade de sucesso foram discutidas.

Conforme já relatado anteriormente, esta aula foi utilizada como ponto de entrega da atividade 3 (desenvolvimento das heurísticas de melhoramento). Com a apresentação da mesma no início da atividade presencial, houve certa dispersão dos alunos após o debate, principalmente dos grupos onde foram verificados equívocos na implementação. Como era de se esperar, muitos ficaram discutindo internamente onde estariam os problemas, o que perturbou a aula em alguns momentos.

Em relação à MDP, foram escolhidos as perguntas D2(*Como verificar se os alunos estão compreendendo o conteúdo de heurísticas e metaheurísticas que está sendo desenvolvido através da investigação ação-educacional, do LOBO e do AMEM?*) e B2(*Os alunos efetivamente estão compreendendo o que são heurísticas e metaheurísticas através da abordagem baseada em problemas?*). Como já relatado, este era o terceiro ponto de verificação definido previamente com os alunos. Dos três grupos que apresentaram algoritmos que continham problemas, em dois deles o equívoco era puramente tecnológico. Após debater sobre o funcionamento dos algoritmos, percebeu-se que eles estavam corretos em relação ao *como* o algoritmo deveria funcionar (teoria das heurísticas de melhoramento), mas a codificação apresentava alguma falha. No entanto, para o último grupo, a questão acerca da vizinhança não ficara clara durante as aulas e, desta forma, o algoritmo estava desenvolvido de forma *certa* sobre um conceito *errado*. Como isso ocorreria? Novamente, o acompanhamento do desenvolvimento das atividades pelo professor foi realizado de forma indireta, através de perguntas em sala de aula e do oferecimento do docente para tirar dúvidas. Contudo, não houve a marcação de pontos intermediários de apresentação de resultados parciais, o que talvez tivesse sanado este tipo de problema conceitual em tempo anterior.

Este ponto de verificação e o debate foram importantes para comparar as diversas estratégias de desenvolvimento de heurísticas de melhoramento. Após a implementação real das heurísticas de melhoramento, foi possível perceber que os conceitos sobre espaço de busca haviam se tornado mais claros. Neste ponto, foi possível fazer a ligação entre o trabalho implementado e as estratégias que deveriam ser implementadas para as metaheurísticas, que usualmente expandiam o espaço de busca.

As metaheurísticas representaram o último grupo de algoritmos trabalhados com os alunos e, neste ponto, foi proposto a quarta e última atividade, o desenvolvimento computacional de uma metaheurística.

<b>Atividade 4</b>	Desenvolvimento de uma Metaheurística
<b>Tempo</b>	04 semanas
<b>Questão</b>	Através do <i>framework</i> do LOBO, desenvolva uma metaheurística e realize testes com todas as instâncias e todos os algoritmos construtivos disponíveis no objeto.

#### *Análise da Atividade 4 – apresentada e discutida durante a terceira aula prática*

Finalmente, a última atividade da disciplina, que também utilizou as duas últimas aulas para esclarecimentos sobre os algoritmos e sobre a implementação, realiza o fechamento dos objetivos do curso, apresentando como resultado o desenvolvimento de um sistema completo de otimização, contendo um ou mais algoritmos construtivos, um algoritmo de melhoramento e uma metaheurística. Novamente, foi realizado um sorteio entre as metaheurísticas que cada grupo deveria implementar, com a exceção do grupo dos Algoritmos Genéticos, que já a um bom tempo estava realizando experimentos na área e, deste modo, se optou por permitir que os alunos continuassem nesta linha de implementação. Em relação ao grupo que foi sorteado para implementar a metaheurística *simulated annealing*, foi requisitado que eles realizassem testes com todos os algoritmos construtivos desenvolvidos, pois esta metaheurística exigia um grau de desenvolvimento computacional inferior às demais. Para os grupos que trabalhariam com os algoritmos genéticos e a busca tabu, onde a diferença existente seria a implementação dos operadores de recombinação/mutação e de busca local, respectivamente, foi sugerido que os mesmos trabalhassem em grupos, trocando idéias para o desenvolvimento do sistema.

Mesmo já de posse dos algoritmos construtivos e de melhoramento, esta atividade foi a que necessitou mais tempo e, ainda assim, houve atrasos na entrega dos trabalhos. O desenvolvimento de uma metaheurística, principalmente a busca tabu e os algoritmos genéticos, que necessitam de estruturas de dados adicionais, é mais complexo. Além disso, a percepção dos equívocos realizados no código também é perturbada pelo caráter probabilístico de todas as metaheurísticas trabalhadas, ao contrário das heurísticas construtivas e de melhoramento, cujo comportamento é determinístico. No caso destas últimas, é possível realizar testes em papel para verificar onde estão escondidos os *bugs* de programação. No entanto, para as metaheurísticas, cada aplicação em uma instância gera, possivelmente, um resultado diferente, o que torna complexa a previsão do comportamento do algoritmo sem o auxílio de exaustivos testes de mesa. Mesmo assim, para algoritmos que necessitam centenas de execuções para mostrar um resultado, como o caso das metaheurísticas evolutivas, tal proposição se torna impossível.

De todo modo, apesar dos atrasos para a entrega desta última atividade, todos os grupos foram capazes de apresentar os resultados e a implementação dos algoritmos com razoável grau de sucesso. Contudo, a análise dos resultados ficou aquém do esperado. Provavelmente devido à questão do tempo, as comparações realizadas pelos grupos não abarcaram a profundidade que estivera presente nas outras atividades. Outra questão preocupante foi a utilização de parâmetros aleatórios para a realização das simulações, completamente diversos dos apresentados em sala de aula. Um exemplo, trazido aqui a título de ilustração, ocorreu em um dos grupos que implementou a metaheurística de Algoritmos Genéticos.

*...pois a população inicial considerada foi totalmente aleatória, o que não permitiu ao algoritmo genético construir populações com fitness bons antes que o critério de parada estabelecido fosse atingido. Outros fatores como a reprodução aleatória dos indivíduos da população e o tamanho da população (apenas 20 rotas por geração) podem ter piorado o desempenho*

*do algoritmo drasticamente. A taxa de mutação foi considerada razoável, pois permitia que as rotas criadas sofressem uma mutação entre 10 e 20% dos seus elementos.*

Para cada instância do problema, o grupo realizou somente seis gerações, contribuindo para encontrar resultados muito aquém dos esperados. No caso específico deste grupo, o mesmo foi considerado em *exame* até que as simulações fossem realizadas novamente, com os parâmetros discutidos em sala de aula.

#### 7.4.6 Aulas Práticas

<b>Ponto 1</b>	Discussão da Atividade 4
<b>Tempo</b>	100 minutos

<b>Ponto 1</b>	Discussão da Atividade 4
<b>Tempo</b>	100 minutos

<b>Ponto 1</b>	Discussão e Entrega da Atividade 4
<b>Tempo</b>	100 minutos

Após a última aula sobre os algoritmos genéticos, foram reservadas as três últimas semanas para o desenvolvimento das metaheurísticas. Após verificar, nas atividades anteriores, que muitos alunos estavam deixando os trabalhos práticos para serem realizados na última hora, o professor exigiu ver os progressos intermediários de uma semana para a outra, discutindo com os alunos as questões conceituais e de implementação.

Para estas aulas, a questão da MDP escolhida foi a D2(*Como verificar se os alunos estão compreendendo o conteúdo de heurísticas e metaheurísticas que está sendo desenvolvido através da investigação ação-educacional, do LOBO e do AMEM?*). Ao acompanhar de forma mais intensa o desenvolvimento da última atividade, se tornou mais claro os diferentes graus de amadurecimento dos alunos, tanto em relação à disciplina quanto em relação à forma metodológica aplicada. O grupo que ficou em exame (seção anterior) simplesmente não apareceu nas duas primeiras aulas, comparecendo somente para a entrega da atividade. O motivo alegado foi o acúmulo de trabalhos que deveriam ser realizados e a *certeza* (grifo deles) de que estavam no caminho certo no desenvolvimento do algoritmo.

Apesar do acompanhamento mais intensivo do professor, ainda houve atrasos na entrega do trabalho (inicialmente, só haveria uma aula para discussão e a entrega ocorreria na próxima semana). No entanto, foi fácil perceber que o tempo era muito exíguo.

#### 7.5 Respondendo à Matriz Dialógica-Problematizadora

Com base nos apontamentos realizados na sala de aula e nas observações realizadas pelo docente, buscou-se responder algumas das questões da MDP que permeiam as principais dificuldades e avanços da disciplina.

*A.1 Os professores possuem conhecimentos básicos sobre heurísticas e metaheurísticas para o Problema do Caixeiro Viajante, sobre a investigação ação-educacional que forma a base do AMEM e sobre o objeto de aprendizagem LOBO?*

A resposta para se os professores realmente tinham conhecimento sobre a área científica, sobre a metodologia, sobre o AMEM e o LOBO deveria ser respondida durante toda a disciplina, e não somente em alguma aula específica. Para que as potencialidades dos objetos tecnológicos fossem alcançadas, seria interessante que os professores tivessem uma larga experiência nos mesmos. A favor desta perspectiva conta que o docente responsável foi o mesmo que modelou e desenvolveu as ferramentas, permitindo que as conhecesse profundamente. Como ponto negativo, deve ser considerado que ambas foram utilizadas pela primeira vez e, como qualquer *software* que está sendo operacionalizado, problemas poderia ocorrer.

Em relação aos demais aspectos, o docente, ao observar a si mesmo na sala de aula, aprofundou a relação dialógica no decorrer do semestre. Aqui cabe ressaltar, mais uma vez, a importância do planejamento da disciplina. Mesmo que os momentos dialógicos problematizadores, no início, se apresentassem mais escassos, tanto em relação ao tempo quanto aos assuntos trabalhados, eles serviram como um primeiro contato dos alunos com a metodologia, bem como do professor. O amadurecimento do docente e dos discentes no decorrer da disciplina transformou os DI mais profícuos, com discussões com melhores argumentos sobre os temas correntes.

Quanto aos conhecimentos básicos do docente em relação à disciplina, não só na preparação das MSEM como no DI eles foram testados à exaustão. Para realmente construir a teoria/prática que deveria ser demonstrada, utilizando o conhecimento dos alunos como ponto de partida ou exemplificação, exige uma capacidade de reconhecer as interligações dos diversos temas da disciplina científica, para ser capaz de, quando for o caso:

- Apontar direcionamentos para outras áreas;
- Apropriar-se das idéias diretamente relacionadas ao contexto e utilizá-las na construção da MSEM;
- Corrigir os equívocos cometidos, sem desmerecer o trabalho, apontando as falhas no raciocínio;
- Mostrar outras facetas do tema, baseados nos exemplos desenvolvidos pelos alunos.

*A.2 Como os professores podem instigar os alunos, utilizando a problematização, o LOBO e o AMEM?*

Esta questão foi parcialmente respondida em duas aulas. A utilização da metodologia dialógica-problematizadora chegou a ser questionada por alguns alunos mas, na sua maioria, os discentes mostraram interesse em desenvolver uma aula mais participativa. A questão central aqui está na boa definição dos DI e das atividades de colaboração, pois estas é que realmente irão instigar o aluno em relação as aulas, motivando-os a responderem as perguntas e resolverem os desafios. A busca dos alunos em materiais bibliográficos além dos sugeridos é uma indicação do interesse dos discentes.

Em relação ao LOBO, ele se mostrou mediação tecnológica fundamental no entendimento do problema que estava sendo trabalhado, tanto no seu modo de ferramenta

como na sua concepção de *framework* de desenvolvimento. A possibilidade de desenvolver algoritmos que poderiam ser simulados computacionalmente se mostrou viável-possível para os alunos.

No entanto, ainda há muito que fazer, pois os alunos, quando instigados, podem trabalhar como cientistas investigativos. Para tanto, o diálogo estabelecido entre professores e alunos deve conduzir a resolução de problemas inerente à disciplina e à metodologia através dos preceitos científicos da observação, análise e comparação dos dados. Para (MION, ANGOTTI, 2006), instigar o aluno a pesquisar, além de refletir, pode ser o ponto de partida para esta transformação.

*Então, como formar o investigador ativo? Como produzir conhecimento científico? Obviamente não é só reflexão que ele necessita. Ele necessita ser instrumentalizado para sistematizar o processo reflexivo de sua própria prática educacional e, com isso, produzir conhecimento científico-educacional. Isto implica, em nosso caso da Física, ter incorporado conhecimento da Física; investigar para aprender como transformar esse conhecimento da Física em conhecimento educacional em Física e, principalmente, implica produzir o conhecimento crítico. Este, por ser libertador, nos mune de poder (empowerment). Mas é preciso instigar o educando a buscar isso.*

(MION, ANGOTTI, 2006)

Finalmente, em relação ao AMEM, ele poderia ter sido mais explorado, principalmente nas suas ferramentas de comunicação. Essa, talvez, tenha sido a falha mais importante na condução da disciplina.

### *A.3. Como os professores têm construído a organização didática de suas aulas nas práticas de investigação-ação educacional de heurísticas e metaheurísticas?*

A organização didática foi estabelecida e apresentada nas páginas anteriores, dividindo as aulas em pontos, que poderiam conter um ou mais momentos dialógico-problematizadores. Além disso, em relação às colaborações, foram organizados fóruns que acompanhassem os debates que ocorriam em momentos não-presenciais. Em todos os momentos presenciais, buscou-se sempre, seguir a seguinte metodologia:

- Apresentar e discutir as atividades que deveriam ser entregues no dia, se houvessem: optou-se por iniciar com esta prerrogativa para evitar que os alunos perdessem a concentração no resto da aula por causa das atividades que ainda deveriam ser entregues;
- Apresentar e discutir as colaborações, se houvessem: primeiramente, antes de iniciar o primeiro ponto específico da aula, era feita uma releitura da discussão que antecedeu a aula, esclarecendo os pontos comuns e explorando os primeiros aspectos que seriam discutidos posteriormente;
- Desenvolvimento da aula, através dos três momentos dialógico-problematizadores: apresentando sempre um DI, construindo a MSEM e

propondo, quando fosse o caso, um DA, a aula transcorria entre os pontos através do diálogo;

- Definição de atividades ou colaborações para os próximos encontros: ao final da aula, quando fosse o caso, seriam discutidas as colaborações para o próximo encontro ou as atividades que deveriam ser desenvolvidas, bem como o prazo para as mesmas.

No entanto, novas perguntas podem ser formuladas para os próximos semestres desta disciplina: *a organização didática das aulas de heurísticas e metaheurísticas nas práticas de investigação-ação educacional são adequadas? Como elas podem ser melhoradas em relação as experiências anteriores?*

*A.4. Os professores conseguem adotar a prática de investigação ação-educacional no ensino de heurísticas e metaheurísticas?*

Como descrito na pergunta A1, a adoção da prática de investigação ação-educacional ocorreu desde o início da disciplina, mesmo que de forma mais humilde no princípio do curso, tanto pelas respostas mais curtas dos alunos quanto pela falta de uma iniciativa mais diretiva do docente. À medida que ambas as partes foram se acostumando com a metodologia, a prática se tornou mais corriqueira e os resultados, em relação à construção da MSEM, foram melhores.

*B.1. Os alunos efetivamente estão aproveitando os operacionalizadores pedagógicos e o LOBO para o entendimento das heurísticas e metaheurísticas? e D.2. Como verificar se os alunos estão compreendendo o conteúdo de heurísticas e metaheurísticas que está sendo desenvolvido através da investigação ação-educacional, do LOBO e do AMEM?*

Já foram relatadas anteriormente as questões relativas ao *deslumbramento* com a novidade, no caso, o objeto LOBO. Os alunos utilizaram o objeto durante todo o semestre letivo, tanto em sala de aula como em atividades extra-classe, onde eles deveriam realizar simulações e desenvolver algoritmos baseados no *framework* definido pelo objeto. Quando inquiridos sobre o assunto, os discentes comentaram dois aspectos interessantes do *software*:

- A possibilidade de implementação e simulação gráfica de heurísticas, eliminando o processo moroso de definição e desenvolvimento de rotinas auxiliares;
- A possibilidade de *enxergar os algoritmos* (expressão deles) funcionando na prática.

Quando havia a utilização intensiva do LOBO, a dinâmica em sala de aula era melhor. Os alunos perguntavam mais sobre os algoritmos desenvolvidos que, invariavelmente, acabavam entrando, também, nas questões técnicas da implementação dos algoritmos. Desta forma, e considerando também que todos os trabalhos práticos utilizaram o objeto como ponto de partida, é crível concluir que o LOBO teve uma influência efetiva no entendimento das heurísticas e metaheurísticas.

Outros desmembramentos destas questões, contudo, podem ser destacadas para trabalhos futuros: a avaliação através do desenvolvimento de *software* é suficiente para a disciplina?; quais aspectos dos operacionalizadores pedagógicos devem ser melhorados?

B.2. *Os alunos efetivamente estão compreendendo o que são heurísticas e metaheurísticas através da abordagem baseada em problemas?*, C.1. *A abordagem dialógica-problematizadora potencializa o ensino de heurísticas e metaheurísticas?* e C.4. *De que forma o Problema do Caixeiro Viajante, o LOBO e o AMEM contribuem para o ensino de heurísticas e metaheurísticas?*

As questões apresentadas envolvem o cerne de todo o trabalho: será que realmente se está *ensinando melhor* com esta mediação tecnológica? O pressuposto que a abordagem dialógica-problematizador potencializa o ensino de heurísticas e metaheurísticas mediados por tecnologia encontra ressonância em vários trabalhos:

*... se for planejada, vivida, auto-refletida e refletida - colaborativamente-, a investigação-ação educacional, como concepção de investigação científica, pode potencializar os seres humanos a interpretar a realidade a partir de suas próprias práticas, concepções e valores, projetando novas ações.*

(GRABAUSKA, DE BASTOS, 2001).

*... podemos potencializar novas ações, práticas, vivências, valores e concepção, que nos encaminharão ao processo de transformação. Essa transformação pode ser pensada num plano micro: as ações educativas em sala de aula; e/ou num plano macro: a própria vida social, extrapolando os muros escolares.*

(SEGAT, 2005)

*... a colaboração entre sujeitos com conhecimentos diferentes potencializa a aprendizagem e o desenvolvimento.*

(ESTEBAN et al., 2003)

*... o importante é que o aluno tem a oportunidade de fazer-se oportunidade, tornando-se mais apto a aproveitar a potencialidade disruptiva do conhecimento. Nasce o rebelde, capaz de confrontar-se com a realidade e a sociedade.*

(DEMO, 2004 apud JAQUÊS, 2005)

As mediações tecnológicas apresentadas durante este capítulo tentam sustentar os trabalhos já apresentado sobre o tema. Para (FREIRE, 1989), a educação dialogal, ativa e voltada para a responsabilidade social e política, caracteriza-se pela profundidade na interpretação dos problemas. Deste modo, durante a construção da MSEM, a mediação pedagógica lança “o homem ao debate, ao exame de seus problemas e dos problemas comuns”(FREIRE, 1989). Considerando o diálogo e a dinâmica da problematização como pontos principais dos atos educativos que foram realizados, potencializa-se o ensino de

heurísticas e metaheurísticas através da adoção dos saberes iniciais dos alunos como ponto de partida para a construção efetiva do conhecimento. Contudo, é importante salientar que mesmo Paulo Freire (2004) considera que o professor, mesmo indispensável, é somente em parte responsável pelas mediações pedagógicas. O aluno e o contexto em que o ato educativo é realizado também devem ser levados em conta, o que retoma as considerações feitas na pergunta B1.

Em relação as aulas desenvolvidas durante a disciplina, há duas formas de tentar responder esta questão: a prática das aulas e os pontos de verificação. Neste estudo, não foram utilizados critérios relativos à comparação de notas entre várias turmas, porque com a diversidade de fatores que influenciam a amostra (alunos diferentes, momentos diferentes, metodologias diferentes) seria muito difícil isolar a influência pura de uma possível melhoria (ou decréscimo) de notas que pudesse ser atribuída as mediações tecnológicas construídas (AMEM e LOBO), implementadas e avaliadas.

Durante a prática das aulas, observou-se que os alunos que participavam mais do DI tinham maior facilidade de compreender o MSEM, pelo menos aparentemente. Quando questionados sobre algum aspecto, durante o segundo momento, estes alunos usualmente conseguiram responder de forma mais adequada do que os que participavam de forma menos efetiva durante o DI. Isso se mostrou facilmente verificável desde o início da disciplina, o que levou a uma postura do professor para exigir que os alunos trabalhassem de forma mais profícua no primeiro momento. Outra observação importante estava no desenvolvimento das colaborações. Como era de se esperar, os que discutiam com mais afinco durante a semana, tinham colaborações mais pertinentes na sala de aula.

Em relação aos pontos de verificação, se optou em analisar as questões relativas à completude das tarefas, em detrimento da quantificação média das notas atribuídas. O que se constatou, durante a experimentação no semestre indicado, é que todos os grupos, sem exceção, acabaram por produzir, ao final da disciplina, um sistema completo de pesquisa operacional, composto de uma metaheurística, uma heurística de melhoramento e, possivelmente, várias heurísticas construtivas. Durante disciplinas anteriores, tal fato nunca havia acontecido. Usualmente, as metaheurísticas acabavam sendo implementadas somente em parte e, alguns grupos, conseguiam desenvolvê-las por completo. No entanto, cabe salientar que, além da metodologia, outro fator que pode ter influenciado nesta questão está no próprio LOBO, que também tem como objetivo diminuir o tempo de implementação, apesar do mesmo já ter sido utilizado na penúltima vez em que a disciplina foi ministrada.

### *B.3. Os estudantes têm assumido que os problemas vistos em aula realmente contribuem no processo de aprendizagem de heurísticas e metaheurísticas?*

Quando se estabelece o diálogo como mediador das práticas educativas, o confronto de idéias é inevitável tanto quanto é desejável. Alicerce da ação, principalmente no desenvolvimento da MSEM, o diálogo percorre todo o ato educativo e, desta forma, não causa estranheza que o próprio método seja objeto do mesmo.

*...no processo de investigação-ação é preciso preparar-se para perceber as transformações de trajetória realizadas pelo grupo com que se interage, como perspectiva de avaliar a viabilidade da proposta formulada e os impasses por ela gerados.*

(NARDI, RODRIGUES, 2005)

Desta forma, o canal de diálogo estabelecido durante as aulas de heurísticas e metaheurísticas também foi utilizado pelos alunos para levantar questionamentos sobre a condução da disciplina. Esta questão se tornou mais clara após dois momentos: quando da discussão em sala de aula sobre as colaborações e após o argumento utilizado por três alunos, que não acreditavam na metodologia. Por meio de comentários ou através de ações, já relatados, depois das primeiras semanas de aula a grande maioria deles se comprometeu com a sua parte no processo educativo e, desta forma, assumiram que os DI e a construção das MSEM eram mais produtivas que as aulas *tradicionais*, baseadas em transparências.

*C.2. Quais são as dificuldades e os avanços do ensino de heurísticas e metaheurísticas utilizando o Problema do Caixeiro Viajante e o LOBO?*

A principal dificuldade na utilização de um problema-tipo (PCV) para o desenvolvimento de toda a disciplina está no alto grau de associação que existe entre as três atividades práticas que foram solicitadas. Como não era possível iniciar a próxima atividade, sem a correta implementação da anterior, a última, e mais complexa, tarefa teve que ser desenvolvida em um tempo que foi considerado exíguo. A solução óbvia de adiantar as mesmas para a próxima turma da disciplina não é tão simples, pois o encadeamento inicial é necessário para que os temas apresentados façam sentido. Se, de um lado, é importante amadurecer e discutir um ponto antes de passar para o próximo, por outro, este *atraso* acaba gerando um complicador no desenvolvimento da última atividade.

Em relação aos avanços, além da questão da completude de todas as tarefas, relatada anteriormente, é possível citar que as aulas, desenvolvidas de forma dialógica, se tornaram menos monótonas e mais participativas e, os alunos, se mostravam interessados no que estava sendo debatido, pois eles *também faziam a aula* (expressão deles).

*C.3. Será que o Problema do Caixeiro Viajante é adequado para o ensino de heurísticas e metaheurísticas?*

Houve uma pequena discussão sobre este assunto, no início da disciplina, mas logo o problema foi sanado, em relação aos alunos, principalmente depois da apresentação da bibliografia que continha exemplos práticos do PCV. Em relação à metodologia, o PCV se mostrou um problema bastante interessante de trabalhar, pois ele permite que algoritmos com dificuldades incrementais possam resolvê-lo. Esta característica permite estabelecer uma metodologia de desenvolvimento prático adequada as necessidades tecnológicas. No entanto, como discutido na questão C3, o encadeamento das atividades pode gerar outros problemas.

Novamente, novas perguntas podem ser formuladas após o semestre concluído: *que tipos de problemas podem ser adequados para esta disciplina?; é possível trabalhar com mais de um problema utilizando esta metodologia?; que adaptações seriam necessárias para tanto?*

D.4. *Quais são as contribuições do ensino de heurísticas e metaheurísticas sob a perspectiva da investigação-ação educacional, o LOBO e o ambiente AMEM?*

Esta questão está no cerne das conclusões, que são tratadas no próximo capítulo.

## 8 Conclusões

As heurísticas e metaheurísticas são algoritmos e, como tal, foram desenvolvidas para a resolução de problemas. Ensinar um algoritmo é, em última análise, ensinar a resolver um problema. Os conteúdos desenvolvidos na pesquisa operacional podem ser problematizados através de exemplos próximos ao aluno, pois o cerne da área está no desenvolvimento de soluções para dificuldades do mundo real.

Recapitulando, a necessidade do professor em avaliar continuamente sua forma de ensinar está em sintonia com a própria mudança dos paradigmas que permeiam as áreas científicas, onde o conhecimento já não é mais estático, mas contínuo e em processo permanente de transformação. A busca dos professores pelo saber científico continua sendo crucial, mas a necessidade de readequar sua prática de ensino a uma sociedade calcada na *informação*, a bem da verdade nem sempre transformada em conhecimento, se torna premente na busca da formação de profissionais cujas habilidades de pesquisa e aprendizagem se tornam cada vez mais cruciais.

Para a área de heurísticas e metaheurísticas, estas preocupações alcançam novos patamares, pois o campo é tão vasto e rico em problemas que é necessário que o aluno reconheça os pontos principais de uma determinada situação, sugerindo e testando soluções criativas. Novos problemas, novas soluções. Desta forma, não há, a princípio, uma forma única e irrestrita que possa resolver *todos* os problemas da área de pesquisa operacional, bem como não há, em um tempo curto como o de uma disciplina de graduação, ou mesmo em um curso inteiro, como explorar todas as miríades possibilidades de problemas e soluções possíveis envolvendo heurísticas e metaheurísticas.

Buscando apoio nas características da área, principalmente no que concerne à resolução de problemas, foi estudado e definido como norteador do processo de ensino aprendizagem a metodologia dialógica-problematizadora, onde se estabelece como princípio geral a construção do conhecimento através da discussão das teorias embasadas nas idéias dos próprios alunos. No entanto, convém mais uma vez lembrar, que o professor é o dirigente do processo, ou seja, a sistematização do conhecimento coletivo deve ser conduzida pelo docente.

Este estudo formou a base que foi utilizada para analisar e desenvolver o Ambiente Multimídia para Educação Mediada por Computador (AMEM) – 2.0. Através das práticas pedagógicas e de sistema ambientadas na primeira versão do AVEA, a sua interface foi readequada e a possibilidade de inserir novos operacionalizadores pedagógicos, na forma de objetos de aprendizagem, foram incluídos. No entanto, como foi constatada anteriormente, a simples utilização de um arcabouço tecnológico não é elemento primordial para que todos os objetivos educacionais sejam atingidos. A tecnologia pode ser uma ferramenta importante e, até mesmo, indispensável, mas o conteúdo e a forma de apresentar e interagir com os alunos é o que realmente faz a diferença entre as práticas bancárias que Paulo Freire condena e o professor investigador e auto-avaliador.

Em relação ao conteúdo, foi apresentado o LOBO, um objeto de aprendizagem que está em sintonia com as principais iniciativas da área e que fornece ao professor e ao aluno possibilidades de interagir, experimentar e desenvolver trabalhos na área de heurísticas e metaheurísticas. Os dois modos do LOBO, *tutoriado* e *expert*, permitem que diferentes tipos de alunos trabalhem com o ambiente de diversos modos. É importante relembrar que

a área de heurísticas e metaheurísticas trabalha com o conceito de *boas* soluções para um problema. Desta forma, muito mais importante do que *acertar* um determinado algoritmo é verificar o processo de experimentação dos alunos, pois é somente através dos diversos erros e fracassos que os estudantes atingem o êxito e, seguramente, conseguem compreender de forma mais profunda o problema com que estão interagindo.

A realização de processos colaborativos, anteriores às aulas presenciais, onde o assunto era debatido entre os alunos e onde o professor poderia inserir os direcionamentos necessários, se mostrou de grande valia para o desenvolvimento da prática docente, principalmente balizando a construção da MSEM. Através dos exemplos retirados das colaborações dos alunos, estes se sentiram mais próximos da realidade escolar e do tema específico. Na experiência realizada, é possível observar que, quanto mais aprofundado fosse o debate anterior, mais rica se tornava a aula presencial. Os alunos, durante os desafios iniciais e o desenvolvimento da melhor solução educacional no momento, quando já haviam trabalhado previamente no problema exposto, conseguiam, de uma forma um tanto quanto óbvia, desenvolver o raciocínio e apresentá-lo de forma mais consistente. A operacionalização da aula em seus momentos era facilitada quando os alunos assumiam seu papel no processo educativo como um todo.

A realização de diversas experiências e o estudo de um problema único, onde todos estão envolvidos, desenvolveu uma massa crítica de conhecimento que se desaguava nas aulas presenciais. Comparativamente as experiências de educação tradicional – baseado no quadro e giz – os educandos se tornaram muito mais participativos, inquisitivos e propositivos. Mesmo que nem todas as idéias apresentadas fossem coerentes ou produtivas, a simples menção das mesmas em aula comprova que os educandos estavam trabalhando no problema proposto, ao contrário de simplesmente estarem recebendo um conhecimento pronto. Através do uso do objeto de aprendizagem com uma metodologia adequada, é possível melhorar o entendimento dos métodos matemáticos envolvidos na área de pesquisa operacional sem perder a profundidade necessária.

Outro aspecto importante do estudo realizado está centrado na *investigação-ação* educacional, ou seja, na transformação do ato educativo em objeto de pesquisa. Ao desenvolver uma análise sistemática de sua forma de ensino, através de anotações realizadas durante e após o encontro presencial e, ainda, respondendo as questões de pesquisa definidas pela matriz dialógica-problematizadora, o docente pode avaliar seu planejamento, sua metodologia e sua prática, ou seja, o seu próprio fazer profissional como professor. Deste modo, redirecionamentos podem ser realizados na disciplina que está ocorrendo no tempo atual ou uma revisão do planejamento pode ser desenvolvida antes que a disciplina ocorra novamente. Na experiência prática realizada, isso não seria diferente e, das observações realizadas, destaca-se:

- quando do início da apresentação das metaheurísticas (seção 7.4.1), os alunos apresentaram idéias desenvolvidas a partir do seu arcabouço teórico já consolidado, principalmente referentes às estruturas de dados e análise de grafos. Considerando que estes elementos já haviam sido trabalhados com alunos em disciplinas predecessoras e, ainda, que os mesmos foram trazidos à tona pelos alunos quando da análise das técnicas heurísticas, seria interessante que heurísticas de melhoramento inspiradas nestas técnicas fossem exploradas em uma versão futura da disciplina. Algoritmos baseados nos métodos de descida/subida, subida com primeira melhoria e aleatório de descida/subida

(RICH, KNIGHT, 1993) podem ser aplicados ao PCV e fariam uma ligação mais direta com as disciplinas já vistas anteriormente;

- o acompanhamento das atividades que devem ser desenvolvidas, com a inclusão de pontos de verificação, se mostrou uma alternativa interessante durante a realização da quarta e última atividade. Ao exigir que os alunos apresentassem o andamento do trabalho, dúvidas surgiram em tempo hábil de serem corrigidas e direcionamentos puderam ser realizados. Esta prática deverá ser adotada para as demais atividades práticas;
- romper a barreira entre a acomodação das aulas positivistas e o trabalho dialógico em sala de aula (e fora dela) é tarefa que deve ser realizada o quanto antes. Mesmo com as explicações iniciais sobre a metodologia, realizada durante o primeiro encontro, muitos alunos se mostraram reticentes em adotá-la na prática, apesar de sua concordância verbal. As maiores dificuldades surgiram no desenvolvimento das atividades fora do âmbito escolar, quando eles deveriam ler e colaborar com o próximo encontro. Optou-se por não avaliar estas colaborações em termos de aumento ou decréscimo da nota final, realizando, no seu lugar, um trabalho de convencimento, em sala de aula, sobre a importância destas atividades. Tal proposta foi parcialmente bem sucedida, pois gradativamente houve um aumento nas colaborações realizadas e nas discussões antes dos encontros presenciais. Ações que poderiam melhorar este aspecto incluem: a) utilização do AMEM como ferramenta motivacional, através do uso de mensagens, salas de discussão e mural; b) direcionamento individual *on-line*, através da percepção de quais são os alunos que não estão participando e cobrança dos mesmos. Na experiência realizada, ambas as ações acima foram realizadas na forma presencial. Também é importante lembrar que o tempo dedicado à cada disciplina pelo docente pode ser exíguo e, além de planejar as aulas, realizá-las no modo presencial, participar das atividades de colaboração, lendo, respondendo e instigando as discussões, o trabalho de motivar os alunos individualmente pode se tornar excessivo. Neste caso, a presença de um tutor e/ou monitor pode ser um diferencial significativo;
- o tempo curto para a realização da última tarefa prática já foi comentado anteriormente. Dependendo do contexto em que o professor esteja trabalhando com esta disciplina, algumas medidas podem ser tomadas, pois não acreditamos que exista uma solução única, que pode ser utilizada de forma irrestrita em quaisquer situações: a) condensar e/ou eliminar alguma das aulas iniciais; b) exigir o mesmo trabalho para todos os alunos, notadamente de uma das heurísticas ou metaheurísticas que seja trabalhada primeiramente – neste caso, corre-se o risco de os alunos perderem a motivação para as demais aulas; c) distribuir as heurísticas e metaheurísticas entre os alunos, conforme elas são apresentadas ao longo do tempo – neste caso, dois problemas podem ocorrer: trabalhos com os mesmos critérios e tempos diferentes para resolução e a desmotivação comentada no item *b*.

Em relação ao processo diretivo do professor, convém salientar que muitos alunos, principalmente ligados à tecnologia, têm um viés prático bastante desenvolvido e podem ser motivados por este prisma. No entanto, cabe ao professor direcionar tais características

de acordo com a proposição da disciplina e, desta, no contexto do curso em que está sendo inserido. No caso específico da disciplina Inteligência Artificial, a apresentação de trabalhos publicados na área onde o PCV foi utilizado para resolver problemas reais serviu para chamar a atenção dos alunos para a área, motivando-os a compreender como as teorias apresentadas poderiam ser utilizadas *lá fora, no mundo real* (expressão deles). Sob outro aspecto, este viés pode servir como um complicador, como ocorreu quando da discussão sobre o POSCOMP, onde alguns alunos se mostraram mais interessados em *estudar* o que poderia ser cobrado naquela prova específica do que realmente *entender* os conceitos da Inteligência Artificial que estavam sendo debatidos.

Concluindo, as principais contribuições deste trabalho podem ser resumidos em:

- propor, desenvolver e aplicar um objeto de aprendizagem (LOBO) para o ensino de heurísticas e metaheurística em uma disciplina de graduação. O LOBO, desenvolvido por dois prismas, como uma ferramenta e como um *framework* de implementação, pode ser utilizado em diversos outros cursos, inclusive para alunos que não sejam desenvolvedores de *software*;
- analisar, modelar e desenvolver a segunda versão do AMEM, dentro de uma perspectiva que auxiliasse o professor em suas tarefas metodológicas, tendo como ponto de partida a investigação ação-educacional e a prática dialógica-problematizadora. Apesar do AMEM 2.0 não exigir que o professor adote tais metodologias, o ambiente dá suporte a elas, permitindo que a investigação da prática educativa seja conduzida através do sistema;
- planejar, aplicar e analisar o fazer acadêmico do professor, em sala de aula, para um semestre letivo completo, utilizando a prática dialógica-problematizadora e o ferramental tecnológico abordado anteriormente. Através desta análise, é possível verificar onde o objeto e o ambiente podem ser utilizados, em que momentos ele foi subutilizado, bem como apontar as necessidades de melhorias, situação comum a todo *software*.

Em relação a trabalhos futuros e encaminhamentos, também convém citar:

- implementação, no LOBO, de todos os algoritmos heurísticos e metaheurísticos desenvolvidos em sala de aula, de forma a utilizar o objeto em todas os DI e as MSEM que trabalham com estes temas;
- desenvolver outras formas de visualização dos resultados das metaheurísticas no objeto. O grande número de passos necessários para que um algoritmo dessa classe faça sentido inviabiliza uma simulação passo-a-passo. Outras estratégias devem ser buscadas para que o aluno consiga visualizar, de forma mais pertinente, como as metaheurísticas se comportam em relação aos seus estados internos;
- desenvolvimento de um prática docente com o auxílio de um tutor e/ou monitor, a fim de verificar as diferenças que ocorrem nos problemas apresentados, cujo teor também estava centrado na falta de tempo hábil de um docente único lidar com todas as colaborações que, por ventura, possam vir a ocorrer;
- refinar a matriz dialógica-problematizadora a partir de outras ocorrências da disciplina, de forma a balizar mudanças nas aulas ou nas práticas docentes;
- modificar as mediações tecnológicas para que a mesma possa ser ministrada em cursos à distância;

- associar a mediação tecnológica LOBO ferramentas de atividades colaborativas do ambiente Moodle para que a mesma possa mediar atividades de estudo em cursos a distância no ensino superior na uab;
- utilizar a experiência do desenvolvimento do LOBO para investigar os potenciais pedagógicos e a metodologia e desenvolvimento de objetos de aprendizagem complexos, como os laboratórios virtuais e experimentação remota.

## 9 Referências Bibliográficas

- AARTS, Emile, KORST, Jan, MICHIELS, Wil. *Simulated Annealing. Search Methodologies*. Ed. Edmund K. Burke, Graham Kendall, Springer US, p.187-210, 2006. Disponível por www em <http://www.springerlink.com/content/x7317j4282441101/>. Acesso em jan 2007.
- ABREU, Ana Sílvia Couto de. *Reflexões sobre um projeto interdisciplinar*. Conhecimento Interativo, v.2., n2., p. 14-21, jul/dez, 2006.
- ALVES, Cláudio M., CARVALHO, Valério. *Planeamento de Rotas num sistema de Recolha de Desperdícios de Madeira*. Investigação Operacional, 2001.
- ANGOTTI, J. A. P., DELIZOICOV, D. N. *Metodologia do Ensino de Ciências*. São Paulo: Cortez, 1990.
- ARAÚJO, Adérito. *As Pontes de Königsberg*. Departamento de Matemática da Universidade de Coimbra. Disponível por www em <http://www.mat.uc.pt/~alma/escolas/pontes/>. Acesso em mar/2006.
- ARAÚJO, Silvio Alexandre de, ARENALES, Marcos Nereu. *Dimensionamento de Lotes e Programação do Forno numa Fundição Automatizada de Porte Médio*. Pesquisa Operacional, v.23, n.3, p. 403-420, set-dez, 2003.
- ASSIS, Wayne Santos. *Utilização de recursos multimídia no ensino de concreto armado e protendido*. Dissertação de mestrado. Escola Politécnica. Universidade de São Paulo, 2002.
- BALAS, E., GUIGNARD, M. *Branch and bound/implicit enumeration*. Ann. Discrete Math. n. 5, p.185-191, 1979.
- BASTOS, Eduardo Nunes; COSTA, Antônio Carlos da Rocha; DIMURO, Graçaliz Pereira. ENSINET: uma Solução Software Livre para Apoio ao Ensino Semi-Presencial Utilizando a Internet. In: IV WORKSHOP DE SOFTWARE LIVRE, 2003, Porto Alegre. *Anais do IV WSL*. Porto Alegre: Evangraf, 2003. p. 55-58.
- BEGNAMI, João Batista. *A Formação dos Professores em Alternância: Um Estudo da Práxis e dos Processos Formativos de Monitores em Escolas Famílias do Brasil*. Relatório Técnico. Universidade François Rabelais de Tours, França, 2003. Disponível por www em <http://www.univ-tours.fr/formadev/intranet/Joao.doc>. Acesso em ago/2006.
- BEHAR, Patricia Alejandra, KIST, Silvia de Oliveira, BITTENCOURT, Juliano de Vargas. *ROODA – Rede cOoperativa de Aprendizagem: Uma plataforma de suporte para aprendizagem à distância*. In: Informática na Educação – Teoria e Prática.

- Programa de Pós-Graduação em Informática na Educação - Vol. 4 N. 2, p.87-96. Porto Alegre: UFRGS, 2001a.
- BEHAR, Patricia Alejandra, KIST, Silvia de Oliveira, BITTENCOURT, Juliano de Vargas. *A caminho de um ambiente para a educação à distância ROODA - rede cooperativa de aprendizagem*. In: XII Simpósio Brasileiro de Informática na Educação. Vitória – ES, Brasil, 2001b.
- BEHRENS, Marilda Aparecida. *O Paradigma Emergente e a Prática Pedagógica*. Curitiba: Champagnat, 1999.
- BOHM, David. *Sobre el diálogo*. Barcelona: Kairós, 1996.
- BURIOL, Luciana Salete. *Algoritmo Memético para o Problema do Caixeiro Viajante Assimétrico como Parte dum Framework para Algoritmos Evolutivos*. Dissertação de Mestrado. Faculdade de Engenharia Elétrica e de Computação, UNICAMP, 2000.
- CAMPELLO, Ruy Eduardo, MACULAN, Nelson. *Algoritmos e Heurísticas: desenvolvimento e avaliação de performance*. Editora da Universidade Federal Fluminense – EDUFF, Niterói, 1994.
- CARR, W., KEMMIS, S. *Becoming Critical: Education, Knowledge and Action Research*. Brighton, Uk: Falmer Press, 1986.
- CETIS – Centre for Educational Technology Interoperability Standards. *UK LOM Core v. 02*. 2004. Disponível por www em [http://www.cetis.ac.uk/profiles/uklomcore/uklomcore\\_v0p2\\_may04.doc](http://www.cetis.ac.uk/profiles/uklomcore/uklomcore_v0p2_may04.doc). Acesso em fev/2006.
- CHRISTOFIDES, N. *The shortest Hamiltonian chain of a graph*. Applied Math, n. 19, p. 689-696, 1970.
- COLLIS, B., CARLEER, G. *The effects of technology-enriched school intervention: a multiple case-study analysis*. Computer and Education, Oxford, v. 21, n.1 e 2. p.151-162, 1993.
- CORDENONSI, Andre Zanki MÜLLER, Felipe Martins, ARAÚJO, Fabrício Viero, BASTOS, Fábio da Purificação de, ALONSO, Cleuza Maria Maximino Carvalho. *Ambiente Multimídia para Educação Mediada por Computador - AMEM*. In: VIII Congresso Ibero-Americano de Extensão Universitária, 2005, Rio de Janeiro. Anais do VIII Congresso Ibero-Americano de Extensão Universitária, 2005.
- CORNE, D., DORIGO, M., GLOVER, F. *New ideas in Optimization*. McGraw-Hill, 1999.
- CNE – Conselho Nacional de Educação, Câmara de Educação Superior. *Resolução n. 4 de 13 de julho de 2005*. Disponível por www em [http://portal.mec.gov.br/cne/arquivos/pdf/rces004\\_05.pdf](http://portal.mec.gov.br/cne/arquivos/pdf/rces004_05.pdf). Acesso em nov/2005.

- CNE – Conselho Nacional de Educação, Câmara de Educação Superior. *Resolução n. 2 de 11 de março de 2002*. Disponível por www em <http://portal.mec.gov.br/cne/arquivos/pdf/CES112002.pdf>. Acesso em nov/2005.
- CROES, G. A. *A method for solving traveling salesman problems*. In: *Operations Research* n. 6, p. 791-812, 1958.
- CUNHA, Claudio Barbieri da, BONASSER, Ulisses de Oliveira, ABRAHÃO, Fernando Teixeira Mendes. *Experimentos Computacionais com Heurísticas de Melhoria para o Problema do Caixeiro Viajante*. XVI Congresso da Associação Nacional de Pesquisa e Ensino em Transporte. Anais. Natal, 2002.
- DAMÁSIO, António R. *O Mistério da Consciência*. São Paulo : Companhia das Letras, 2000.
- DAMÁSIO, António R. *O Erro de Descartes: emoção, razão e o cérebro humano*. São Paulo : Companhia das Letras, 1996.
- DARWIN, Charles. *A Origem das Espécies*. Rio de Janeiro: Tecnoprint, 1987.
- DÁVALOS, R.V. *Uma Abordagem do Ensino de Pesquisa Operacional Baseada no Uso de Recursos Computacionais*. In: *International Conference On Engineering And Technology Education*. 6. Proceedings... Santos. Brasil, 1999.
- DE BASTOS, Fábio da Purificação, ALBERTI, Taís Fim, MAZZARDO, Mara Denize. *Ambientes Virtuais de Ensino-Aprendizagem: Os Desafios dos Novos Espaços de Ensinar e Aprender e suas Implicações no Contexto Escolar*. *Revista Novas Tecnologias na Educação*. V Ciclo de Palestras Novas Tecnologias na Educação, v. 3. n.1. mai. 2005.
- DE BASTOS, Fábio da Purificação.; MÜLLER, Felipe Martins. *Criando Desafios em Informática*. In: *Atas da IV Escola de Verão sobre Investigação-Ação Educacional*. UFSM, Santa Maria, RS, 1999.
- DEMO, Pedro. *Ser professor é cuidar que o aluno aprenda*. Porto Alegre : Mediação, 2004.
- DIAZ, Adenso Fernández, VELARDE, José Luiz González, LAGUNA, Manuel, MOSCATO, Pablo, TSENG, Fan T., GLOVER, Fred, GHAZIRI, Hassan M. *Optimización Heurística y Redes Neuronales*. Editorial Paraninfo, Madrid, 1996.
- DORIGO, Marco, STÜZLE, Thomas. *Ant Colony Optimization*. Cambridge : Mit Press, 2004.

- DOWNES, S. *Learning Objects*. 2000. Disponível por www em: [http://isit.arts.ubc.ca/cdf/L\\_Objects.pdf](http://isit.arts.ubc.ca/cdf/L_Objects.pdf). Acesso em jun/2005.
- DUTRA, Débora. *Uma Arquitetura de Biblioteca Digital de Aulas Baseada no Padrão IEEE LOM*. Dissertação de mestrado. Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, 2003.
- DUTRA, Renato Luís de Souza, TAROUCO, Liane Margarida Rockenbach, KONRATH, Mary Lucia Pedroso Konrath. *Objetos de Aprendizagem: Uma comparação entre SCORM e IMS Learning Design*. Revista Novas Tecnologias na Educação. Porto Alegre: Centro Interdisciplinar de Novas Tecnologias na Educação (UFRGS), v.4, n. 1, Julho, 2006.
- DUTRA, Renato Luís de Souza, TAROUCO, Liane Margarida Rockenbach, KONRATH, Mary Lucia Pedroso Konrath. *SACCA – Sistema Automático de Catalogação de Conteúdo Audiovisual*. Revista Novas Tecnologias na Educação. Porto Alegre: Centro Interdisciplinar de Novas Tecnologias na Educação (UFRGS), v.3, n. 2, Novembro, 2005a.
- DUTRA, Renato Luís de Souza, TAROUCO, Liane Margarida Rockenbach, KONRATH, Mary Lucia Pedroso Konrath. *IMS Learning Design, evoluindo de Objetos de Aprendizagem para Atividades de Aprendizagem*. Revista Novas Tecnologias na Educação. Porto Alegre: Centro Interdisciplinar de Novas Tecnologias na Educação (UFRGS), v.3, n. 1, Maio, 2005b.
- EICHLER, Marcelo Leandro, DEL PINO, José Claudio. *Ambientes Virtuais de Aprendizagem: Desenvolvimento e Avaliação de um Projeto em Educação Ambiental*. Editora da UFRGS, 2006.
- ELLIOT, J. *What is Action-Research in Schools?* In: Journal of Curriculum Studies, v.10, n.4, p. 355-357, 1978.
- ESTEBAN, Maria Theresa, HOFFMAN, Jussara, SILVA, Janssen Felipe. *Práticas avaliativas e aprendizagens significativas em diferentes áreas do currículo*. Porto Alegre : Mediação, 2003.
- FARKUH NETO, Alberto, LIMA, Renato da Silva. *Roteirização de Veículos de uma Rede Atacadista com o Auxílio de Sistemas de Informação Geográficas (SIG)*. Revista Pesquisa e Desenvolvimento em Engenharia de Produção, n.5, p.18-39, jun, 2006.
- FEHÉR, Marta. *Lo Natural y lo Artificial (un ensayo de clarificación conceptual)*. Teorema - Revista Internacional de Filosofia, v. XVII/3, 1998. Disponível por www em <http://www.oei.es/salactsi/teorema04b.htm>. Acesso em ago/2007.
- FEO, T., REZENDE, M. *A Greed Randomized Adaptative Search procedure for maximum independent set*. Operations Resarch n. 42, 860-879. 1994.

- FERNÁNDEZ, E. G. *Ambiente Multimídia para Educação Mediada por Computador na perspectiva da Investigação-Ação Educacional: Modelagem e Implementação*. 234f. Dissertação de Mestrado em Engenharia de Produção – Universidade Federal de Santa Maria, Santa Maria, 2003.
- FLEMMING, D.M., LUZ, E.F. *A educação à distância nas engenharias: relatos de uma experiência*. In: Congresso Brasileiro de Ensino de Engenharia, 28. Ouro Preto, 2000.
- FOLEY, James D. *Computer Graphics: Principles and Practice*. Ed. Reading: Addison-Wesley, 1997.
- FREIRE, Paulo. *Educação como prática da liberdade*. Rio de Janeiro: Paz e Terra, 19 edição, 1989.
- FREIRE, Paulo. *Pedagogia do Oprimido*. Rio de Janeiro: Paz e Terra, 17 edição, 1987.
- FREIRE, Paulo. *Pedagogia da autonomia: saberes necessários à prática educativa*. Rio de Janeiro: Paz e Terra, 2004.
- FREITAS, Maria Teresa de Assunção. *Vygotsky e Bakhtin - Psicologia e Educação: um intertexto*. São Paulo: Ática, 1996.
- FUKS, Hugo, PIMENTEL, M., GEROSA, M.A., FERNANDES, M.C.P. e LUCENA, J.P. *Novas Estratégias de Avaliação Online: aplicações e implicações em um curso totalmente a distância através do ambiente AulaNet*. Avaliação da Aprendizagem em Educação Online, orgs. Marco Silva e Edméa Santos. São Paulo: Loyola, 2006. p 369-385. Disponível em <http://groupware.les.inf.puc-rio.br>. Acesso em jul/2006.
- GLOVER, Fred, LAGUNA, M. *Tabu Search*. Springer, 1997.
- GLOVER, Fred. *Future Paths for Integer Programming and Links to Artificial Intelligence*. Computers and Operations Research, n.5. p. 533-549, 1986.
- GLOVER, Fred. *Heuristics for Integer Programming Using Surrogate Constraints*. Decision Sciences n. 8 (1), p. 156-166. 1977.
- GOLDBARG, Marco Cesar. LUNA, Henrique Pacca L. *Otimização combinatória e programação linear: modelos e algoritmos*. Rio de Janeiro: Campus, 2000.
- GOLDBERG, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
- GIMONET, J.C. A alternância na formação - Método pedagógico ou novo sistema educativo? A experiência das MFRs. – in: *Alternance, Développement Personnel et Local*, Demol Jean-Noël et Pilon Jean-Marc, coordinateurs, l'Harmattan, Paris. 1998.

- GOODRICH, Roberto Tamassia Michael. *Estrutura de Dados e Algoritmos em Java*. Bookman Companhia Editora, 2002.
- GRABAUSKA, C.J., DE BASTOS, F.P. *Investigação-ação educacional: possibilidade crítica e emancipatórias na prática educativa*. In: MION, R.A. *Investigação-ação: mudando o trabalho de formar professores*. Ponta Grossa: Gráfica Planeta, p. 9-20. 2001.
- GRINGS, Eliane Schlemmer, MALLMANN, Marly, DAUDT, Sônia Isabel Dondonis. *Ambiente virtual de aprendizagem: uma experiência interdisciplinar no ensino superior*. In: *Actas do V Congresso Iberoamericano de Informática Educativa*. Viña Del Mar, Chile, 2000. Disponível por www em <http://lsm.dei.uc.pt/ribie/docfiles/txt200372911557Ambientevirtualdeaprendizagem.pdf>. Acesso em fev/2006.
- GUEDES, Gilberto Gomes. *O profissional docente de administração como organizador e gestor do trabalho pedagógico no ensino: um estudo de caso*. Dissertação de mestrado. Curso de Mestrado Profissionalizante em Engenharia. Universidade Federal do Rio Grande do Sul, 2004.
- HADDAD, Sérgio; GRACIANO, Mariângela. *Educação: direito universal ou mercado em expansão*. São Paulo Perspec., São Paulo, v. 18, n. 3, 2004. Disponível por www em [http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0102-88392004000300008&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-88392004000300008&lng=en&nrm=iso). Acesso em fev/2006.
- HANSEN, Roseli P., PINTO, Sérgio Crespo. *Construindo Ambientes de Educação Baseada na Web Através de Web Services Educacionais*. In: *Anais do XIV Simpósio Brasileiro de Informática na Educação*, Rio de Janeiro, 2003.
- HELD, M., KARP, R.M. *The traveling salesman problem and minimum spanning trees*. *Operational Research*, n. 18. p. 1138-1162, 1970.
- HOELZEL, Carlos Gustavo. *Design Ergonômico de Interfaces Gráficas Humano-Computador: Um Modelo de Processo*. Tese de Doutorado, Programa de Pós-Graduação em Engenharia de Produção, UFSC, 2004.
- HOROWITZ, E., SAHNI, S., RAJASEKARAN, S. *Computers Algorithms C++*, Computer Science Press, 1998.
- IBGE – Instituto Brasileiro de Geografia e Estatística. *Censo Demográfico - 2003 : Educação: Resultados da amostra*. 2003. Disponível por www em <http://www.ibge.gov.br/home/presidencia/noticias/02122003censoeduchtml.shtm>. 2003. Acesso em jan/2007.
- IEEE Learning Technology Standards Committee (LTSC). *Proposed base document for a Draft Standard for Competency Definition Data Objects*. 2002. Disponível por www em [http://ltsc.ieee.org/doc/wg12/LOM\\_WD4.PDF](http://ltsc.ieee.org/doc/wg12/LOM_WD4.PDF). Acesso em nov/2006.

- IP, Albert; MORRISON, Iain. *Learning Objects in Different Pedagogical Paradigms*. Proceedings of the 18th Annual Conference of the Australian Society for Computers in Learning in Tertiary Education. Melbourne, 2001. Disponível por www em: <http://www.medfac.unimelb.edu.au/ascilite2001/pdf/papers/ipa.pdf>. Acesso em fev/2005.
- IMS. *IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 - Standard for Learning Object Metadata*, 2006. Disponível por www em [http://www.msglobal.org/metadata/mdv1p3/imsmd\\_bestv1p3.html](http://www.msglobal.org/metadata/mdv1p3/imsmd_bestv1p3.html). Acesso em dez/2006.
- IMS. *IMS Learner Information Package Specification*, 2005. Disponível por www em <http://www.msglobal.org/>. Acesso em dez/2006.
- IMS QTI. *IMS Question and Test Interoperability Overview*. Disponível por www em [http://www.imsproject.org/question/qti\\_v2p0/imsqti\\_oviewv2p0.html](http://www.imsproject.org/question/qti_v2p0/imsqti_oviewv2p0.html). 2006. Acesso em jul/2007.
- ISO, *Date and Time Formats*. 1997. Disponível por www em <http://www.w3.org/TR/NOTE-datetime>. Acesso em dez/2006.
- JAQUÊS, Jussara da Rocha. *Avaliação Mediadora: Uma Proposta para a Educação Superior*. Dissertação de Mestrado, Curso de Mestrado em Ciências e Matemática, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2005.
- KEMMIS, S.; MCTAGGART, R. *Cómo Planificar La Investigación-acción*. Barcelona: Editorial Laerts, 1988.
- KIRKPATRICK, S., GELATTI, C.D. Jr., VECCHI, M.P. *Optimization by Simulated Annealing*. In: Science, v. 220. n. 4598. p.671-680. May. 1983.
- KOCHE, Jose Carlos. *Fundamentos de metodologia científica: teoria da ciência e iniciação à pesquisa*. Vozes: Petrópolis, 2006.
- KÖHLER, Viviane Cátia. *Estratégias Evolutivas para o Problema de Sequenciamento de Tarefas em Máquinas Paralelas Idênticas com Tempo de Preparação das Tarefas*. Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia da Produção, Universidade Federal de Santa Maria, 2004.
- KONDER, Leandro. *O que é Dialética*. 17. ed. São Paulo: Brasiliense, 1987.
- KOPER, R., OLIVIER, B.. *Representing the Learning Design of Units of Learning*. In: Educational Technology & Society, n.7. v.3, p.97-111. 2004.
- KOZA, John R. *Genetic Programming*. Ed. Massachusetts Institute of Technology, 1998.

- KRATZ, Ricardo de Andrade. *Fábrica de adequação de conteúdo de ensino para Objetos de Aprendizagem Reutilizáveis (RLOs) respeitando a norma SCORM*. Dissertação de mestrado. Programa de Pós-Graduação em Computação Aplicada, Universidade do Vale do Rio dos Sinos, 2006.
- LAUERMANN, Rosiclei Aparecida Cavichioli. *Ambiente Multimídia para Educação Mediada por Computador na Perspectiva da Investigação-Ação: Avaliação e Tutorial*. Dissertação de Mestrado, PPGE/CT/UFSM, Santa Maria, 2002.
- LAWLER, E.L., LENSTRA, J.K., RINNOOY KAN, A.H., SHMOYS, D.B. *The traveling salesman problem*. John Wiley & Sons, 1985.
- LEÃO, Lúcia. *Ciberespaço: análises e reflexões para a construção de modelos descritivos de sistemas hipermediáticos*. In: O chip e o Caleidoscópio: reflexões sobre as novas mídias. Lucia Leão (org). Editora SENAC. 2003.
- LÉVY, Pierre. *Cibercultura*. Rio de Janeiro: Editora 34. 1999.
- LIN, S. *Computer solutions of the traveling salesman problem*. In: Bell System. Technical. Journal. n. 44, p. 2245-2269, 1965.
- LINSINGEN, I. *et al*. Formação do Engenheiro: Desafios da atuação docente, tendências curriculares e questões contemporâneas da educação tecnológica. Florianópolis: Editora da UFSC, 1999.
- LITTLE, J.D.C, MURTY, K.G., SWENEY, D.W., KAREL, C. *An algorithm for the traveling salesman problem*. Operational Research, n.11, p.972-989, 1963.
- LU, June, YU, Chun-Sheng, LIU, Chang. *Learning style, learning patterns, and learning performance in a WebCT-based MIS course*. Information & Management, 40, Edt. Elsevier. p. 497-507. 2003. Disponível por www em <http://140.128.151.3/Mispaper/infor&manag/03042811515606918.pdf>. Acesso em dez/2004.
- MALLOY, Thomas E., HANLEY, Gerard, E. *MERLOT: A faculty-focused Web site of educational resources*. Behavior Research Methods, Instruments, & Computers. v. 33. n.2 p. 274-276. 2001.
- MARIOTTI, Humberto. *Diálogo: um método de reflexão conjunta e observação compartilhada da experiência*. 2001. Disponível por www em <http://www.geocities.com/pluriversu/dialogo.html>. Acesso em jan/2007.
- MARTINS, António José, FIOLEAIS, Carlos, PAIVA, João. *Simulações On-Line no Ensino de Física e Química*. Revista Brasileira de Informática na Educação. Numero 7. SBC. p.111-117. v. 11, n. 2. Julho/Dezembro, 2003.

- MCTEAR, Michael F. *Spoken Dialogue Technology: towards the conversational user interface*. Springer, 2004.
- MERIEU, P. *Aprender...Sim, mas como?* Porto Alegre: Artmed, 1998.
- MION, Rejane Aurora, ANGOTTI, José André Peres, DE BASTOS, Fábio da Purificação. *Construindo a formação problematizadora e dialógica do educador*. Revista Olhar de Professor, Ponta Grossa, v.2, n.2, p. 195-205, nov. 1999.
- MION, Rejane Aurora, ANGOTTI, José André Peres. *Investigação-ação e a Formação de Professores em Física: O Papel da Intenção na Produção do Conhecimento Científico*. In: XI Encontro Nacional de Pesquisa em Ensino de Física, Londrina – PR, 2006.
- MORAN, José Manuel. *Como utilizar a Internet na Educação*. Ciência da Informação, v. 26, n.2, Brasília, mai/ago, 1997.
- MORONI, Janaína Luisa da Silva. *Ensino a Distância UlbraOrbe da Universidade Luterana do Brasil: A relação do uso de novas tecnologias e o custo benefício para a Universidade*. Revista Novas Tecnologias na Educação - III Ciclo de Palestras Novas Tecnologias na Educação, v.2. n.1, mar. 2004.
- MOSCATO, P. *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*. Technical Report, Caltech Concurrent Computation Program, C3P Report 826. 1989.
- MÜLLER, Felipe Martins, DE BASTOS, Fábio da Purificação. *Matriz Dialógico-Problematizadora como Ferramenta Organizadora do Trabalho Escolar no AMEM*. In: Congresso Nacional de Ambientes Hiperídia para Aprendizagem. Florianópolis: 2004.
- NASCIMENTO, Lisandra, LEIFHEIT, Marcelo. *Análise de um curso a distância que utilizou uma nova ferramenta de Courseware chamada Moodle*. Revista Novas Tecnologias na Educação. Maio, v.3, n1. V Ciclo de Palestras Novas Tecnologias na Educação. 2005.
- NASCIMENTO, Anna Christina de Azevedo. *Construindo Comunidades de Elaboradores de Objetos de Aprendizagem através de Conteúdo, Tutoria e Interação dos Pares*. Secretaria de Educação a Distância, Ministério da Educação, dezembro 2005. Disponível por www em: <http://rived.proinfo.mec.gov.br/artigos/Anna20063.pdf>. Acesso em jun/2006.
- NARAHARI, Y. *Data Structures and Algorithms*. Indian Institute of Science, Bangalore. 1997. Disponível por www em <http://lcm.csa.iisc.ernet.in/dsa/>. Acesso em jan/2007.

- NARDI, Ivanete, Rodrigues, Thaís Fernanda Castro. *Uma Experiência de Formação de Educador@s na Perspectiva Intercultural: Problematização Acerca da Investigação-Ação*. In: Reunião da Associação Nacional de Pós-Graduação e Pesquisa em Educação, XXVIII, Caxambu – MG, 2005.
- NUÑO, Andrés Sampedro, NISTAL, Angel Martinez, RUIZ, Beatriz Rodríguez, GONZÁLEZ, Raquel-Amaya Martínez. *Experiencia Didáctica con Objetos de Aprendizaje Multimedia Reutilizables en el LMS Claroline*. In: II Simposio Pluridisciplinar sobre Diseño, Evaluación y Descripción de Contenidos Educativos Reutilizables. Barcelona, 2005. Disponível por www em <http://www.uoc.edu/symposia/spdece05/pdf/ID21.pdf>. Acesso em fev/2006.
- OR, I. *Traveling Salesman-Type Combinatorial Problems and Their Relation to the Logistics of Regional Blood Banking*. Ph.D. Thesis, Northwestern University, Evanston, 1976.
- OSI, *The Open Source Definition*. Open Source Initiative, 2006. Disponível por www em <http://www.opensource.org/docs/osd>. Acesso em jul/2008.
- OSF – Open Software Foundation. *Guia de Estilos para Interface com Padrão OSF/MOTIF*. Cambridge, USA, 1990.
- PARIZOTTO, Rosamelia. *Elaboração de um Guia de Estilos para Serviços de Informação em Ciência e Tecnologia*. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia de Produção, UFSC, 1997.
- PENROSE, Roger. *A Mente Nova do Rei: computadores, mentes e as leis da física*. Rio de Janeiro: Campus, 1997.
- PETTRO, M.L. *Políticas públicas educacionais: dos materiais didáticos aos multimídias*. In: Reunião Anual da Associação de Pós-Graduação em Educação, 1999.
- PIRES, José A., ALVES, Paulo. *A Usabilidade em Software Educativo: Princípios e Técnicas*. Actas do VI Congresso Iberoamericano de Informática Educativa. 2002. Disponível por www em <http://lsm.dei.uc.pt/ribie/docfiles/txt2003729175845paper-198.pdf>. Acesso em maio/2005.
- PIRES, Armando. *O Ensino de Engenharia na Europa*. In: Congresso Brasileiro de Ensino de Engenharia – COBENGE, Campina Grande, PB, 2005. Disponível por www em [http://www.abenge.producao.ufrgs.br/portal/arquivo/iframe/startDownload.asp?File=../upload/arquivos/armandopiresauditorio.pps&Name=O Ensino da Engenharia na Europa&id=6&nro\\_download=19](http://www.abenge.producao.ufrgs.br/portal/arquivo/iframe/startDownload.asp?File=../upload/arquivos/armandopiresauditorio.pps&Name=O%20Ensino%20da%20Engenharia%20na%20Europa&id=6&nro_download=19). Acesso em jan/2007.
- PLATÃO. *A República*. Coleção Os Pensadores. São Paulo: Nova Cultura, 1997.

- POLSANI, Pithamber R. *Use and Abuse of Reusable Learning Objects*. In: Journal of Digital Information, v. 3, n. 4. article n. 164, 2003.
- POOLE, Martha David. *Álgebra Linear*. Ed. Thomson Learning Ibero, 2004.
- POPPER, K. *Verdade, racionalidade e a Expansão do Conhecimento Científico*. In: Conjecturas e Refutações (Popper, K. (org.)). Coleção Pensamento Científico, Ed. Universidade de Brasília, 1983.
- PORTUGAL, Cristina. *Educação à distância: o design como agente do “diálogo” mediado pelas interfaces computacionais*. Revista Brasileira de de Aprendizagem Aberta e a Distância. v. 1. n. 2. dez. 2002.
- POWELL, Andy, HOLLANDS, Paul, HIM, Debra, BARKER, Phil, JEWELL, Nik, REED, Anne. *RDN/Academy LOM Application Profile Compliance Guidelines*. Ed. Paul Hollands. Maio, 2004. Disponível por www em [http://www.ltsn-01.ac.uk/interoperability/compliance\\_guidelines](http://www.ltsn-01.ac.uk/interoperability/compliance_guidelines). Acesso em jun/2006.
- RAMALHO, José Carlos, HENRIQUES, Pedro Rangel, LIBRELOTTO, Giovani Rubert. *ADRIAN : sistema de suporte à produção de conteúdos*. In: Conferência eLearning no Ensino Superior. Aveiro - Portugal, 2004.
- REI, Gerhard. *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer-Verlag, 1994.
- REINELT, G. *TSP-LIB – A Traveling Salesman Library*. ORSA Journal of Computing, n. 3, p. 376-384, 1991.
- RFC 2048. *Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures: Request For Comments: 2048*. 1988. Disponível por www em <http://www.faqs.org/ftp/rfc/rfc2048.txt>. Acesso em jun/2006.
- RFC 2426. *vCard MIME Directory Profile: Request For Comments: 2426*. 1998. Disponível por www em <http://www.ietf.org/rfc/rfc2426.txt>. Acesso em jun/2006.
- RIBEIRO, Cora H.F. Pinto, CHIARAMONTE, Marilda Spindola. *Recomendações Básicas para o Projeto Gráfico e Navegacional de Interfaces de Aplicações Educacionais baseadas em Validação Experimental*. Revista Novas Tecnologias na Educação, v.1, n. 2, set. 2003.
- RICH, Elaine. KNIGHT, Kevin. *Inteligência Artificial*. São Paulo : Makron Books, 1993.
- RODRIGUES, Iracema Stancati, AZEVEDO, Hilton de, LESZCZYNKI, Sonia Ana C. A *Mudança da Prática Pedagógica do Modelo Presencial para o Modelo de Educação à Distância sob as Óticas da Teoria da Atividade e da Metodologia Inovadora*. In: I Seminário Nacional ABED de Educação à Distância, Belo Horizonte, 2003.

Disponível por www em <http://www.abed.org.br/seminario2003/texto12.doc>. Acesso em ago/2006.

ROMANI, Luciana Alvim Santos, ROCHA, Heloisa Viera da, SILVA, Celmar Guimarães da. *Ambientes para educação a distância baseados na Web: Onde estão as pessoas?*, em Anais do III Workshop sobre Fatores Humanos em Sistemas Computacionais, 18 a 20 de outubro de 2000, Gramado -RS.

SANTANCHÈ, André. *Anima: Representando e Integrando Objetos Educacionais na Web*. Dissertação de Mestrado. Salvador: UNIFACS/Programa de Pós-Graduação em Redes de Computadores, 2002.

SANTOS, Edméa Oliveira dos, OKADA, Alexandra Lilavati Pereira. *A Construção de Ambientes Virtuais de Aprendizagem: por Autorias Plurais e Gratuitas no Ciberespaço*. In: XXVI Reunião Anual da Associação Nacional de Pós-Graduação e Pesquisa em Educação, Poços de Caldas, 2003.

SANTOS, Wilson Nascimento. *Comunicação não-verbal e educação: um olhar filosófico sobre a espontaneidade da comunicação corporal como práxis pedagógica*. Dissertação de mestrado, Programa de Pós-Graduação em Educação, Universidade Federal da Bahia, 2003.

SAVIANI, Dermeval. *Escola e Democracia*, São Paulo: Cortez, 1991.

SCORM. *Sharable Content Object Reference Model 2004 – 3<sup>rd</sup> Edition*. Advanced Distributed Learning. 2004. Disponível por www em <http://www.adlnet.gov/downloads/files/311.cfm>. Acesso em junho/2005.

SEGAT, Taciana Camera. *Aprendendo e Ensinando no Exercício Docente*. In: Revista Eletrônica “Fórum Paulo Freire”, ano 1, n.1, 2005.

SIMON, H.A. *The Sciences of The Artificial*, M.I.T Press, Cambridge, 1969.

SÓCRATES. Coleção Os Pensadores. São Paulo: Nova Cultura, 1996.

SOLARTE, Mario Fernando, URBANO, Franco Arturo. *Experiencias en la utilización de dotLRN en educación presencial y la distancia en la Universidad del Cauca*. In: dotLRN Conferente, Madrid, 2005.

SOUZA, Analucia C.P., PEREIRA, Mariângela. *Tendências em Educação Matemática em um Curso de Extensão Universitária*. In: XXVIII Congresso Nacional de Matemática Aplicada e Computacional. São Paulo, 2005.

SOUZA, Marcia Izabel Fugisawa; VENDRUSCOLO, Laurimar Gonçalves; MELO, Geane Cristina. *Metadados para a descrição de recursos de informação eletrônica: utilização do padrão Dublin Core*, Revista Ciência da Informação. v.29 n.1. Brasília jan./abr. 2000.

- STRECK, Danilo R, REDIN, Euclides, ZITKOSKI, José (org). *Dicionário Paulo Freire*. Belo Horizonte : Autêntica, 2008.
- TAROUCO, Liane Margarida Rockenbach; FABRE, Marie-Christine Julie Mascarenhas; TAMUSIUNAS, Fabrício Raupp *Reusabilidade de objetos educacionais*. RENOTE - Revista Novas Tecnologias na Educação. Porto Alegre: Centro Interdisciplinar de Novas Tecnologias na Educação (UFRGS), v. 1, n. 1, 2003.
- TORRES, Germano Labert (org). *Advances in Intelligent Systems and Robotics: Laptec 2003*. IOS Press, 2003.
- TOSO, Eli Ângela Vitor, MORABITO, Reinaldo. *Otimização no Dimensionamento e Seqüenciamento de Lotes de Produção: Estudo de Caso numa Fábrica de Rações*. Gestão & Produção, v. 12, n.2, p. 203-217, mai-ago, 2005.
- TOZONI-REIS, Marília Freitas de Campos. *Temas ambientais como “temas geradores”*: contribuições para uma metodologia educativa ambiental crítica, transformadora e emancipatória. In: Educar, n. 27, p. 93-110, Curitiba – PR, 2006.
- TULLIO, A.A. *A prática pedagógica do professor de Engenharia Agrônômica*. In: Ciencia agrícola, n. 52(3). p.594-603. Piracicaba, set/dez. 1995.
- VALENTE, José Armando; ALMEIDA, Fernando José de. *Visão analítica da informática na educação no Brasil: a questão da formação do professor*. Disponível em <http://www.inf.ufsc.br/sbc-ie/revista/nr1/valente.htm>. 2001. Acesso em jul/2006.
- VERHAART, Michael. *Learning Object Repositories: How useful are they?* Proceedings of the 17th NACCQ 2004. Samuel Mann e Tony Clear (eds). Nova Zelândia, p. 465-470, 2004.
- VOLGENANT, T., JONKER, R. *A branch-and-bound algorithm for the symmetric traveling salesman problem based on the 1-tree relaxation*. European Journal of Operational Research. n.9, p.83-89. 1982.
- VRASIDAS, Charalambos. *Issues of Pedagogy and Design in e-learning Systems*. ACM Symposium on Applied Computing, 2004.
- WILEY, DAVID A. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In: WILEY, D. A. (Ed.). *The Instructional Use of Learning Objects*. p. 1–35. 2001. Disponível por www em: <http://reusability.org/read/chapters/wiley.doc>. Acesso em fev/2007.
- ZABALA, Antoni. A Avaliação. In: ZABALA, Antoni. *A Prática educativa: como ensinar*. Porto Alegre: ArtMed, 1998.

ZACHARY, J.L. *Introduction To Scientific Programming Computational Problem Solving Using: Maple And C*". University Of Utah. Animações disponíveis em <http://www.cs.utah.edu/~zachary/isp/applets/cannon/cannon.html>, 1999.

ZANAKIS, S.H., EVANS, J.R. *Heuristic Optimization: Why, when and how to use it*. Interfaces, v.11, n.5, out. 1981.

**APÊNDICE A – O Ambiente Amem 2.0 e o Objeto de Aprendizagem  
LOBO – Distribuição *Open Source***

## **APÊNDICE B – Lista de Publicações do autor geradas durante o desenvolvimento da Tese**

### **Artigos Completos - Periódicos**

CORDENONSI, A. Z., MÜLLER, Felipe Martins, BASTOS, Fábio da Purificação de. *O Objeto de Aprendizagem LOBO como Ferramenta no Ensino Dialógico e Investigativo de Heurísticas e Metaheurísticas*. RENOTE: Revista Novas Tecnologias na Educação, Porto Alegre, v. 6, n. 1, 2008.

CORDENONSI, A. Z., MÜLLER, Felipe Martins, BASTOS, Fábio da Purificação de O Fluxo de Tarefas de um AVEA: o (Re)Desenvolvimento do AMEM. RENOTE: Revista Novas Tecnologias na Educação. , v.5, n.2. p.1 - 8, 2007.

CORDENONSI, A. Z. ; MÜLLER, Felipe Martins ; BASTOS, Fábio da Purificação de . O Ensino de Heurísticas e Metaheurísticas na área de Pesquisa Operacional sob a ótica da Educação Dialógica Problematizadora. RENOTE: Revista Novas Tecnologias na Educação, Porto Alegre, v. 3, n. 1, 2005.

### **Trabalhos Completos - Eventos**

CORDENONSI, A. Z., MÜLLER, Felipe Martins, BASTOS, Fábio da Purificação de. *A Prática da Educação-Dialógica Problematizadora através do LOBO*. In: V Congresso Brasileiro de Ensino Superior à Distância, 2008, Gramado - RS. Anais do V Congresso Brasileiro de Ensino Superior à Distância. Porto Alegre: ABED, 2008.

CORDENONSI, A. Z., MÜLLER, Felipe Martins, BASTOS, Fábio da Purificação de. *Desenvolvendo Competências em Pesquisa Operacional: O Objeto de Aprendizagem LOBO*. In: X Conferência Internacional de Educação em Engenharia e Tecnologia, 2008, Santos. Anais da X Conferência Internacional de Educação em Engenharia e Tecnologia. , 2008.

CORDENONSI, A. Z., ARAÚJO, Fabrício Viero, MÜLLER, Felipe Martins, Elenusa, Tatiani. *Desenvolvimento de um Sistema de Importação e Exportação de Dados para Um*

*Ambiente Virtual de Ensino Aprendizagem Usando XML*. In: X Conferência Internacional de Educação em Engenharia e Tecnologia, 2008, Santos. Anais da X Conferência Internacional de Educação em Engenharia e Tecnologia. , 2008.

CORDENONSI, A. Z., MÜLLER, Felipe Martins, BASTOS, Fábio da Purificação de. *Investigação-Ação no Ambiente AMEM: Relato de uma Experiência*. In: III Congresso Nacional de Ambientes Hipermídia para Aprendizagem, 2008, São Paulo. Anais do III Congresso Nacional de Ambientes Hipermídia para Aprendizagem. , 2008.

CORDENONSI, A. Z., MÜLLER, Felipe Martins, BASTOS, Fábio da Purificação de. *LOBO: Learning Object Based on Optimization - Um Objeto de Aprendizagem para o Ensino de Heurísticas e Metaheurísticas*. In: XXXIX Simpósio Brasileiro de Pesquisa Operacional, 2007, Fortaleza - CE. Anais do XXXIX Simpósio Brasileiro de Pesquisa Operacional. , 2007.

CORDENONSI, A. Z., MÜLLER, Felipe Martins, BASTOS, Fábio da Purificação de, ARAÚJO, Fabrício Viero, Elenusa, Tatiani. *O LOBO e o Caixeiro Viajante*. In: XVIII Simpósio Brasileiro de Informática na Educação, 2007, São Paulo. Anais do XVIII Simpósio Brasileiro de Informática na Educação. , 2007.

CORDENONSI, A. Z., MÜLLER, Felipe Martins, BASTOS, Fábio da Purificação de. *O Professor como Agente Observador: a Investigação-Ação como Prática do Ensino de Graduação na Computação*. In: I Workshop de Educação em Informática - WEI Tche, 2007, Torres - RS. Anais do I Workshop de Educação em Informática - WEI Tche. , 2007.

CORDENONSI, A. Z., BERNARDI, Giliane, BASTOS, Fábio da Purificação de, ARAÚJO, Fabrício Viero, MÜLLER, Felipe Martins. *Objeto de Aprendizagem para o Ensino de Heurísticas e Metaheurísticas em Cursos Superiores de Pesquisa Operacional*. In: XX Encuentro Nacional de Docentes en Investigación Operativa e XVIII Escuela de Perfeccionamiento en Investigación Operativa, 2007, Mar del Plata - Argentina. Anais do XX Encuentro Nacional de Docentes en Investigación Operativa e XVIII Escuela de Perfeccionamiento en Investigación Operativa. Mar del Plata: Univeridad Nacional de Mar del Plata, 2007.

CORDENONSI, A. Z. ; MÜLLER, Felipe Martins ; BASTOS, Fábio da Purificação de ; ARAÚJO, Fabrício Viero ; ALONSO, Cleuza Maria Maximino Carvalho ; MEDINA, Roseclea Duarte . *Ambiente Multimídia para Educação Mediada por Computador Utilizando Software Livre - AMEM*. In: Congresso Nacional de Ambientes Hiperídia para Aprendizagem, 2006, Florianópolis. Anais do Congresso Nacional de Ambientes Hiperídia para Aprendizagem, 2006.

CORDENONSI, A. Z. ; MÜLLER, Felipe Martins ; BASTOS, Fábio da Purificação de ; ARAÚJO, Fabrício Viero . *Objetos de Aprendizagem para o Ensino de Heurísticas e Meta-Heurísticas: Uma Abordagem Baseada em um Ambiente de Aprendizagem e na Educação Dialógica Problematizadora*. In: XXXVIII Simpósio Brasileiro de Pesquisa Operacional, 2006, Goiânia - GO. Anais do XXXVI Simpósio Brasileiro de Pesquisa Operacional, 2006.

CORDENONSI, A. Z. ; MÜLLER, Felipe Martins ; ARAÚJO, Fabrício Viero ; BASTOS, Fábio da Purificação de ; ALONSO, Cleuza Maria Maximino Carvalho . *Ambiente Multimídia para Educação Mediada por Computador - AMEM*. In: VIII Congresso Ibero-Americano de Extensão Universitária, 2005, Rio de Janeiro. Anais do VIII Congresso Ibero-Americano de Extensão Universitária, 2005.

CORDENONSI, A. Z. ; MÜLLER, Felipe Martins ; BASTOS, Fábio da Purificação de . *A Educação de Heurísticas e Metaheurísticas Utilizando um Ambiente Virtual de Ensino-Aprendizagem e uma Metodologia Problematizadora*. In: XXXVII Simpósio Brasileiro de Pesquisa Operacional, 2005, Gramado - RS. Anais do XXXVII Simpósio Brasileiro de Pesquisa Operacional, 2005.

### **Resumos - Eventos**

CORDENONSI, A. Z., ARAÚJO, Fabrício Viero, BASTOS, Fábio da Purificação de, Elenusa, Tatiani, MÜLLER, Felipe Martins. *Desenvolvimento de um Sistema de Importação e Exportação de Dados para Um Ambiente Virtual de Ensino Aprendizagem Usando XML*. In: XVIII Simpósio Brasileiro de Informática na Educação, 2007, São Paulo - SP. Anais do XVIII Simpósio Brasileiro de Informática na Educação. SBC, 2007.