



Trabalho de Graduação

**CURSO MEDIADO POR COMPUTADOR PARA O ENSINO DE
HEURÍSTICAS E META-HEURÍSTICAS PARA O PROBLEMA DE *BIN
PACKING***

ADRIANA FARIAS DE MEDEIROS

Curso de Ciência da Computação

SANTA MARIA, RS, BRASIL

JULHO , 2004

**CURSO MEDIADO POR COMPUTADOR PARA O ENSINO DE
HEURÍSTICAS E METAHEURÍSTICAS PARA O PROBLEMA DE
*BIN PACKING***

POR
ADRIANA FARIAS DE MEDEIROS

Trabalho de graduação apresentado ao curso de Ciência da
Computação – Bacharelado, da Universidade Federal de
Santa Maria (UFSM, RS), como requisito
parcial para obtenção do grau de
Bacharel em Ciência da Computação

Curso de Ciência da Computação

SANTA MARIA, RS, BRASIL

JULHO, 2004

Trabalho de Graduação nº 177

Santa Maria, RS, Brasil

Julho de 2004

Universidade Federal de Santa Maria

Centro de Tecnologia

Curso de Ciência da Computação

**A Comissão Examinadora, abaixo assinada, aprova o Trabalho de
Graduação**

**CURSO MEDIADO POR COMPUTADOR PARA O ENSINO DE
HEURÍSTICAS E METAHEURÍSTICAS PARA O PROBLEMA DE
*BIN PACKING***

elaborado por

Adriana Farias de Medeiros

Como requisito parcial para obtenção do grau em Bacharel em Ciência da
Computação

COMISSÃO EXAMINADORA

Profº Dr. Felipe Martins Müller (Orientador)

Profº Ms. Andre Zanki Cordenonsi

Profª Ms. Iria Brucker Roggia

**“Todos os dragões da nossa vida são talvez
princesas que esperam ver-nos um dia belos
e corajosos. Todas as coisas aterradoras não
são mais, talvez, do que coisas indefesas que
esperam que as socorramos.**

(Rilke)”

**“Nossas dúvidas são traidoras e nos fazem
perder o que, com frequência, poderíamos
ganhar, por simples medo de arriscar.
(William Shakespeare)”**

Agradecimentos

Primeiramente agradeço a minha família: meus pais, Ivan e Marina pelo esforço, carinho, dedicação e paciência que tiveram comigo nestes anos de graduação, meu irmão, Evandro que soube me “aturar” e ao Seu Pedro, que é como um pai para mim. Gostaria de fazer um agradecimento especial ao meu pai, que durante todo o período que eu estudei de noite, vinha me buscar na cidade todas as noites, não reclamando e muitas vezes abrindo mão de dormir ou de outros compromissos para me buscar, visto que não havia outro meio de locomoção até a minha casa. Muito obrigada por tudo.

Agradeço a todos os professores, especialmente ao Professor Felipe Müller, meu orientador para elaboração deste trabalho, que soube transmitir seus conhecimentos com sabedoria e atenção, sempre me incentivando a continuar, mesmo quando a vontade de desistir falava mais alto.

Não poderia deixar de agradecer ao orientando de doutorado Andre Zanki Cordenonsi, pelo acompanhamento e auxílio na elaboração das aulas e na estruturação do trabalho como um todo.

Finalizando os agradecimentos, afinal não posso agradecer nominalmente a todos, agradeço aos meus amigos e amigas, muitos dos quais foram de extrema importância para eu chegar aonde cheguei. Agradeço pelas bobagens ditas na hora do lanche, pelas caronas, pelas dúvidas tiradas, pelos trabalhos em grupo, pelos ombros amigos na hora do desabafo. Sem vocês, com certeza eu não teria conseguido.

E o último, porém o mais importante de todos, agradeço a Deus por estar aqui.

Sumário

Lista de figuras.....	vii
Lista de tabelas.....	viii
Resumo	ix
Introdução	10
1 Investigação – Ação Educacional	12
1.1 Potencialidades para a IAE.....	13
1.1.1 Três momentos pedagógicos.....	14
1.1.2 Matriz dialógica problematizadora	16
2 Ambiente Multimídia de Educação Mediado por Computador.....	17
2.1 O Ambiente AMEM	17
2.2 Funcionalidades e características do AMEM	18
3 Heurísticas e Metaheurísticas	23
3.1 Definição do problema de <i>bin packing</i>	25
3.1.1 Algoritmos genéticos e o HGGA.....	28
4 O ensino de heurísticas e metaheurísticas sob a perspectiva da investigação-ação educacional no ambiente AMEM utilizando o problema de <i>bin packing</i>	30
4.1 Ensino de heurísticas e Metaheurísticas	30
4.2 Proposta de metodologia.....	32
4.2.1 MDP proposta	33
4.2.2 Aulas	35
Conclusão.....	71
Bibliografia	72

Lista de figuras

Ilustração 1: Área pessoal	19
Ilustração 2: Módulo de comunicação	20
Ilustração 3: Módulo das disciplinas	21
Ilustração 4: Módulo de biblioteca	21
Ilustração 5: Módulo de ajuda	22
Ilustração 6: Módulo de saída	22
Ilustração 7: Peças para o problema com duas dimensões	36
Ilustração 8: Peças para uma dimensão	37
Ilustração 9: Distribuição utilizada pela empresa considerando uma dimensão feita utilizando o algoritmo NFD.	37
Ilustração 10: Distribuição utilizada pela empresa para o problema com duas dimensões feita utilizando o algoritmo FFD.	37
Ilustração 11: Peças ordenadas	38
Ilustração 12: Distribuição encontrada	38
Ilustração 13: Distribuição encontrada	38

Lista de tabelas

Tabela 1: MDP proposta	33
Tabela 2: Arranjos para o problema apresentado	40
Tabela 3: Lista de compras do supermercado	44
Tabela 4: Itens ordenados por peso.....	45
Tabela 5: Lista de compras a serem empacotadas.	47
Tabela 6: Provável solução utilizando o algoritmo de BFD.....	48

Resumo

Trabalho de graduação

Ciência da Computação

Universidade Federal de Santa Maria

Curso mediado por computador para o ensino de heurísticas e metaheurísticas para o problema de *bin packing*

Autora: Adriana Farias de Medeiros

Orientador: Profº Dr. Felipe Martins Müller

Local e Data de Defesa: Santa Maria, 15 de julho de 2004

O curso proposto foi desenvolvido sob a perspectiva da investigação-ação educacional, utilizando os conceitos da educação dialógica-problematizadora e seus operacionalizadores de aprendizagem: três momentos pedagógicos (De Bastos e Müller, 1999) e a matriz dialógica-problematizadora. O curso faz uso também da problematização do conhecimento e do Ambiente Multimídia de Educação Mediado por Computador (AMEM) como ambiente de ensino virtual, que implementa os conceitos da educação dialógica-problematizadora. O público alvo para o qual ele se destina são os alunos de graduação do Curso de Ciência da Computação da Universidade Federal de Santa Maria.

Introdução

O trabalho de graduação é uma oportunidade para o aluno colocar em prática parte dos conhecimentos adquiridos durante sua formação acadêmica associada ao curso de graduação, com conseqüente aprofundamento do seu conhecimento sobre o assunto em questão.

Através dos tempos, as tecnologias computacionais mostram-se mais presentes nas casas, nos escritórios, nas escolas, nos estabelecimentos comerciais, entre outros, evoluindo os processos rotineiros, que eram executados de maneira manual. Juntamente com essa evolução, percebe-se o processo de globalização que tem alcançado grandes proporções a partir da disseminação de conhecimentos.

Pode-se notar que, fazendo uma analogia entre o mundo dos negócios globalizados e área da educação (Lauermann *et al.*, 2003), pode-se interpretar estratégias de negócio como uma estratégia de ensino-aprendizagem cujo objetivo é potencializar os procedimentos didáticos-metodológicos com a introdução da informática na educação, pois o processo de mudança alcança todas as instituições, inclusive a educação e o ensino nos diversos níveis, principalmente nas universidades. Essas mudanças exigem da população uma aprendizagem constante.

As tecnologias computacionais aparecem como meios de possibilitar uma ação docente inovadora; entretanto essa inovação não deve estar somente restrita ao uso da tecnologia, mas, também, à maneira como o professor vai aproveitar as formas de elaborar planos metodológicos que superam a simples reprodução do conhecimento.

Por isso deve-se ressaltar que recursos existem e estão disponíveis, basta-se dispor de projetos educativos adequados para desencadear o processo de aprendizagem (Lauermann *et al.*, 2003).

Mediante uma realidade palpável e para uma efetiva melhoria na qualidade do ensino com a introdução da informática na educação, foi que surgiu o ambiente AMEM (Ambiente Multimídia de Educação Mediado por Computador) (Müller *et al.*, 2000). Seu uso está presente como uma ferramenta de apoio a professores do ensino médio, graduação e pós-graduação na Universidade Federal de Santa Maria (UFSM).

Logo, tendo-se em mãos os operacionalizadores presentes no AMEM e a metodologia de ensino da Investigação-Ação Educacional decidiu-se implementar um curso desenvolvido sob essa perspectiva.

No capítulo 1, tem-se a descrição da metodologia da Investigação-Ação Educacional e dos dois operacionalizadores presentes no AMEM, fundamentais para o desenvolvimento deste trabalho: três momentos pedagógicos e a Matriz Dialógica Problematizadora.

No capítulo 2, tem-se a descrição sobre o funcionamento das ferramentas presentes no AMEM.

No capítulo 3, tem-se a descrição do problema que será problematizado no curso: Heurísticas e Metaheurísticas para o problema de *bin packing*.

No capítulo 4, tem-se a descrição do curso, com metodologias alternativas para a exemplificação e as aulas propostas com a matriz que norteia as mesmas.

No capítulo 5, encontra-se a conclusão.

No capítulo 6, tem-se a bibliografia.

1 Investigação – Ação Educacional

Segundo Carr e Kemmis (1986) pode-se definir a Investigação-Ação como: “uma forma de **investigação auto-reflexiva** feita em situações sociais, pelos participantes, no sentido de melhorar a racionalidade e justiça de suas próprias **práticas**, seu entendimento sobre essas, e situações que essas acarretam” (p. 162, **grifos nossos**).

Assim sendo, refletindo-se sobre as práticas presentes nas escolas de ensino básico e superior e empregando-as em situações recorrentes do relacionamento entre professor-aluno, que Elliott (1978) descreveu: “*a investigação nas escolas investiga as ações humanas e situações sociais as quais são experienciadas por professores como: a) inaceitáveis em alguns aspectos (problemáticas); b) suscetíveis de mudanças (contingentes); c) requerendo uma resposta prática (prescritivas)*” (p.1).

Então, tendo-se em mãos essas duas definições e, segundo Miquelin e de Bastos (2002), pode-se compreender a Investigação-Ação Educacional (IAE) como: “*uma ação investigativo-escolar estratégica donde decorre a programação de práticas escolares, sua implementação, o monitoramento dessas através de registros, reflexões e auto-reflexões para uma sistemática transformação em forma de reprogramações*”.

Lewin *apud* Carr e Kemmis (1986), descrevem também um âmbito mais amplo para a IAE, caracterizado como um processo estrutural dividido em: conjunto de programação, implementação, observação e reflexão. Esse processo comporta-se como um ciclo da chamada “*espiral reflexiva lewiniana*”.

Esses ciclos, quando se utiliza a ação-investigação estratégica, garantem que todos os envolvidos no processo estejam constantemente verificando sua prática escolar, possibilitando assim um maior dinamismo no processo, ou seja, uma freqüente mudança nas nossas práticas escolares.

Por isso, quem trabalha com a IAE nunca encara o trabalho como um produto acabado, mas sim, está sempre refletindo sobre como melhorar o processo.

Sendo assim, se convém destacar a prática educacional norteada pelas teorias da IAE e a Educação Dialógica-Problematizadora (EDP) (Carr e 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 à 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 aos planejamentos das mesmas, face às evidências observadas, reconstruindo uma nova ação informada.

Já a EDP considera que a atividade pode ser elaborada metodologicamente através dos seguintes momentos (Angotti e 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 momento seguinte, que é a organização do conhecimento, o educador deve orientar a sistematização do mesmo, para a completa compreensão do tema e do problema apresentado. O aluno deverá fazer uso dos conhecimentos para estudar o problema inicial e conseguir generalizá-lo para outras situações que são explicadas pelo mesmo conhecimento. A essa capacidade de generalização é dado o nome de aplicação do conhecimento, caracterizando o terceiro momento da EDP.

1.1 Potencialidades para a IAE

O comportamento da IAE pode ser compreendido através de Bowen (1998), que discute a sistematização da IAE através de diagramas gráficos, os quais levou a elaborar mapas de todo o seu trabalho.

Assim sendo, podem-se compreender os passos da IAE através dos meios de investigação voltada à identificação, compreensão e estratégias de resolução de problemas voltados à prática escolar.

De tal maneira, Miquelin e De Bastos (2002), definem o âmbito da IAE como “*o processo de ensino/aprendizagem, pois aparece como uma relação dinâmica entre os envolvidos, tanto nas práticas escolares como na formação inicial e continuada de professores*”.

Então, mediante esses conceitos, pode-se acoplar meios de comunicação virtual como forma de monitoramento para as discussões dos alunos sobre a aula. Esse monitoramento é feito à distância.

O uso de meios de comunicação virtual possibilita ao aluno entrar em contato com o professor fora do horário “normal” de aula, para discutir e obter maiores informações sobre o que foi apresentado em aula.

Através do uso de ambientes multimídia-telemáticos como o AMEM (Müller *et al.*, 2000), que será explicado com maiores detalhes no próximo capítulo, torna-se possível disponibilizar aos alunos bibliografias e tarefas escolares. Esses materiais estão disponíveis nos planejamentos das aulas e podem ser acessados antes dos encontros presenciais, possibilitando a inclusão de professores e alunos nos ciclos de ensino/aprendizagem que ocorrem dentro e fora da sala de aula.

O uso dessas ferramentas não dispensa o encontro presencial entre os integrantes do processo educacional, mas fortalece a resolução de problemas.

Conforme o investigador vai levantando os dados e refletindo sobre eles, o processo de ensino/aprendizagem começa a ganhar uma dinâmica maior do que se fosse apenas baseada na reflexão da observação do ocorrido no encontro presencial.

Tem-se então considerado, nos termos de ensino-aprendizagem na IAE, os momentos em que os alunos refletem fora do âmbito escolar, à medida que disponibilizamos as atividades extraclasse, tarefas escolares e colaboração com as reprogramações no AMEM.

Sendo assim, os alunos têm a sua disposição mecanismos tecnológicos para uma comunicação eletrônica e colaboração no trabalho escolar, no âmbito à distância. Isso possibilita a troca de informações entre os integrantes do grupo, contribuindo para o aumento da espiral dos ciclos e agregando dados relativos ao ensino-aprendizagem fora da sala de aula.

1.1.1 Três momentos pedagógicos

Durante a concepção do ambiente AMEM, a teoria educacional Dialógica-Problematizadora foi reorganizada pelo grupo de pesquisa, e apresentada em (De Bastos e Müller, 1999), como três momentos: **desafio inicial, melhor solução educacional no momento e desafio mais amplo**. Inicialmente, 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. Esse problema pode inclusive ter ou não uma solução factível. O emprego desse procedimento metodológico faz com que os educandos trabalhem na resolução do desafio, obrigando-os a “pensar”, visto que eles têm que elaborar uma solução ou discutir com o educador ou com os colegas sobre prováveis dúvidas.

Através de múltiplas interações entre aluno/educador e aluno/aluno, deve-se consolidar o conhecimento científico através da codificação/decodificação da *melhor solução educacional no momento*, sistematizada pelo educador. O professor avalia os resultados encontrados pelos alunos, possibilitando um acompanhamento das atividades bem como um maior incentivo sobre o retorno dos resultados. Uma forma de realizar a avaliação dos resultados encontrados pelos alunos é através de dinâmicas de grupos.

Desta forma, o educador pode confrontar as visões de mundo dos alunos, tensionando as visões científicas e cotidianas, rompendo estas, 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 Müller, 1999). Essa avaliação pode ser feita com o auxílio de certas ferramentas como fórum, debates ou atividades para casa, onde o aluno deverá aplicar os conhecimentos adquiridos durante a aula, porém sem a presença do professor.

Deste modo, o projeto AMEM procura consolidar a problematização dos conteúdos escolares, desafiando educadores e alunos a trabalhar na escola com a perspectiva da resolução de problemas científico-tecnológicos. De forma prática, o ambiente provê, entre outras, as seguintes facilidades (Lauer mann, 2002): (a) construção da *matriz dialógica-problematizadora*, que possibilita a criação de uma estrutura sistemática envolvendo educador, aluno, tema de estudo e contexto, favorecendo o exame e discussão da preocupação temática; (b) implementação das *atividades extraclasse*, que reúne as três partes previamente definidas que compõem o planejamento: programação, atividade extraclasse e atividade de colaboração. A programação apresenta a estrutura cronológica das atividades. A atividade extraclasse é apresentada na forma de um problema a ser resolvido e a atividade de colaboração é usada pelo educador para possibilitar ao aluno participação ativa do processo de programação da próxima aula (disponibilização de bibliografias, pré-programação, recebimentos de sugestões, etc.); e (c) anotação de *registros*, onde o educador poderá registrar suas reflexões sobre cada ação.

1.1.2 Matriz dialógica problematizadora

As primeiras referências à Matriz Dialógica-Problematizadora (MDP) encontram-se na Tábua Aristotélica de Invenção, e seu objetivo era “criar uma estrutura sistemática para examinar e discutir um tema” (Kemmis e McTaggart, 1988, p. 123). O uso da matriz não garante que surgirão questões novas, mas pode auxiliar os professores a definirem os seus problemas relacionados às orientações educativas (De Bastos e Müller, 2004). A matriz utiliza as quatro categorias básicas ou componentes básicos em qualquer situação educativa citadas por Kemmis e McTaggart (1988): professores, alunos (público alvo), tema de estudo e contexto.

Segundo De Bastos e Müller, 2004: *“O componente investigativo do trabalho é orientado pela Matriz Dialógica-Problematizadora (MDP), que contém questionamentos para orientar os planejamentos. A implementação compreende a fase de agendamento da ação e sua posterior aplicação, onde professores e alunos seguem os passos definidos na programação, envolvendo-se em discussões que geram e sustentam o módulo de comunicação, bem como, para momentos de cooperação onde se registram os dados obtidos na ação, orientados pela MDP e as sugestões para novas ações. Os registros asseguram que as contribuições e dados resultantes das atividades, tanto presenciais quanto a distância, não sejam perdidos. Assim, as contribuições dos alunos e/ou dos professores colaboradores alimentarão a fase de reprogramação”*.

O professor pode fazer uso dos recursos de planejamento onde poderá compor a MDP para uma disciplina de uma determinada turma. A finalidade é nortear a programação, as atividades extraclasse e colaboração.

Considerando esse referencial, passou-se a criar uma tabela com quatro lugares comuns colocados nos dois eixos, horizontal e vertical. Em seguida inicia-se a formulação da questão pelo elemento A1 da tabela, questionando, em cada quadro, que se pode dizer deste lugar com relação ao outro (De Bastos e Müller, 2004).

Após a formulação da matriz, deve-se relacionar um ou mais elementos da matriz [A1], [A2]... [D4] com o planejamento das aulas. Cada elemento contém um questionamento que o professor deve procurar respondê-lo durante o processo de ensino-investigação-aprendizagem. Maiores referências sobre os conceitos da MDP podem ser encontradas em De Bastos e Müller (2004).

A seguir será apresentada uma descrição sobre o funcionamento e implementação do AMEM.

2 Ambiente Multimídia de Educação Mediado por Computador

O projeto AMEM – Ambiente Multimídia de Educação Mediado por Computador na perspectiva da Investigação-Ação Educacional, projeto aprovado pela FAPERGS edital 06/2000 (Müller *et al.*, 2000), surgiu para viabilizar a realidade apresentada no capítulo anterior.

O AMEM está sendo utilizado como ferramenta de auxílio para professores do ensino médio, graduação e pós-graduação na Universidade Federal de Santa Maria (UFSM), visto que ele possui todos os operacionalizadores presentes no processo de Investigação-Ação Educacional, podendo assim, colocar em prática a metodologia empregada nesse processo.

Sendo assim, optou-se por desenvolver o curso proposto nos moldes do AMEM, fazendo-se necessário que haja uma descrição mais detalhada do seu funcionamento e da sua implementação.

2.1 O Ambiente AMEM

O AMEM foi desenvolvido e projetado por uma equipe multidisciplinar, composta por especialistas em informática, educação científica-tecnológica, pedagogia e desenho gráfico da UFSM, sendo que dois dos integrantes dessa equipe eram mestrandos em Engenharia de Produção na área de concentração de Tecnologia de Informação. As tecnologias empregadas no seu desenvolvimento, serão descritas no decorrer do capítulo.

Acredita-se que somente os recursos tecnológicos não garantem inovação nenhuma no processo de aprendizagem, mas sim dependem de um projeto bem feito, integrando professores e alunos (usuários), onde o computador seja uma ferramenta capaz de auxiliar na prática do “aprender a aprender” existente no processo de ensino-aprendizagem.

O AMEM foi desenvolvido com as seguintes características: Ambiente multimídia de educação mediada por computador em rede, para educação presencial, semipresencial e à distância, que tem como objetivo proporcionar uma ferramenta de auxílio a professores que adotam a investigação-ação (Carr e Kemmis, 1986; Ângulo, 1990) e a educação dialógica-problematizadora (Freire, 1983) como norteadores do trabalho escolar, conforme foi explicado no capítulo anterior.

Além disso, entre suas características pode-se citar a boa qualidade do sistema, a alta taxa de flexibilidade e não possuir custos para aquisição e manutenção, visto que foi

desenvolvido utilizando sistema operacional e aplicativos livres, de domínio público onde não existe a necessidade de pagamento.

O motivo para desenvolver um sistema livre foi baseado na descrição do projeto proposto por Müller *et al.*, 2000 para a construção de um ambiente virtual. Maiores detalhes relativos à modelagem e implementação do ambiente podem ser encontrados em (Fernández, 2003).

Pode-se resumir a modelagem do sistema da seguinte forma: sistema de processamento compartilhado para Web, modelado numa arquitetura cliente-servidor com três níveis - repositório de informação depositada num banco de dados, lógica do negócio disponível via WWW (*World Wide Web*) e interface de visualização, também via WWW.

Para a implementação das interfaces foi escolhida a linguagem de programação para Internet HTML 4.0 (*HyperText Markup Language*), com folhas de estilo (CSS – *Cascade Style Sheet*). Para viabilizar o dinamismo necessário que permitiria a coleta de informações e o acesso às outras ferramentas escolheu-se a linguagem PHP 4.0 (*Hypertext Preprocessor*). O servidor WWW escolhido foi o servidor Apache, devido a sua segurança e robustez. O banco de dados utilizado como gerenciador e repositório de dados foi o MySQL 3.23 (Yarger *et al.*, 2000), por ser multiplataforma e por ser livre. E o sistema operacional utilizado para o desenvolvimento do ambiente foi o Linux Mandrake 8.0.

Essas escolhas foram feitas em nível de desenvolvimento, mas deve-se destacar que o sistema independe de sistema operacional, banco de dados e servidor Internet.

Além das características de implementação, deve-se explicar também o funcionamento das ferramentas que estão presentes no AMEM, que são mostradas a seguir.

2.2 Funcionalidades e características do AMEM

Existem três diferentes tipos de perfis que podem ser assumidos pelos usuários: professor e aluno, integrantes do processo de ensino-aprendizagem e administrador, responsável pela manutenção do ambiente. O administrador é encarregado por atribuir o perfil de administrador para outros usuários, criar novas disciplinas e atribuir seus professores (Lauermann *et al.*, 2003).

Os professores são responsáveis por matricular alunos, criar turmas, atualizar o cadastro das suas respectivas disciplinas e seus conteúdos, além de propiciar meios de comunicação e cooperação dos alunos, tanto intraturmas quanto interturmas. A criação de

grupos de alunos (turmas) facilita o gerenciamento das atividades por parte do professor, que pode também analisar as interações feitas pelos participantes (Lauermann *et al.*, 2003).

Já como aluno, o usuário poderá ter acesso às atividades propostas pelo professor e seus materiais didáticos, bem como às ferramentas de comunicação disponibilizadas pelo professor. Para ter acesso como aluno e se ainda não estiver cadastrado no sistema, o usuário precisará cadastrar-se no ambiente, preenchendo os campos presentes no formulário de cadastro.

O processo de interação entre alunos e professores está baseado no conceito de educando-educador (Freire, 1983) e nas teorias-guias da investigação-ação e educação dialógico-problematizadora. Para que isso seja possível, notou-se a necessidade de criação de certos mecanismos de coordenação, comunicação e cooperação, agrupados conforme suas similaridades.

O modo como foram organizados os módulos das interfaces do AMEM foi baseado no estudo desenvolvido por Azolin (2002) onde se apresentou um trabalho sobre a ergonomia das interfaces homem-máquina no ambiente, buscando obter interfaces amigáveis.

Segundo Lauermann *et al.*, (2003), os módulos do AMEM são os seguintes:

- **Pessoal**: manutenção dos cadastros pessoais, de senhas, consulta de dados cadastrais dos usuários, acesso à agenda de compromissos pessoais e as mensagens (correio eletrônico), caracterizado por ser um mecanismo para troca de mensagens de texto entre os usuários do AMEM de forma assíncrona (figura 1).

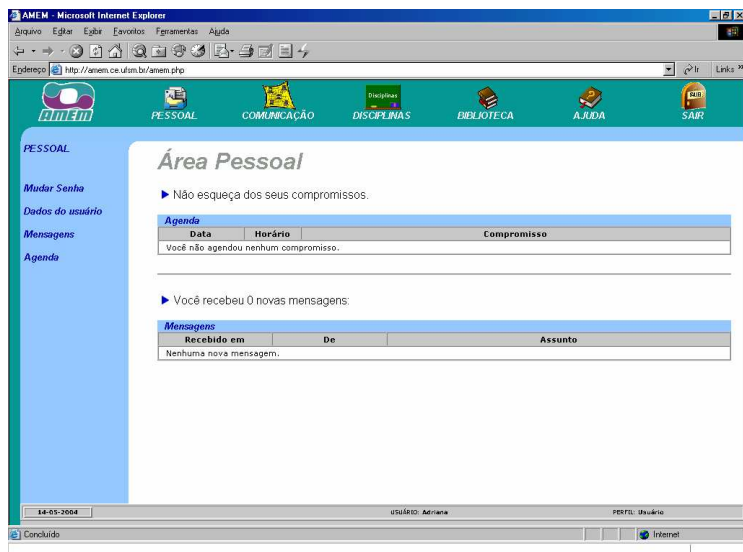


Ilustração 1: Área pessoal

- **Comunicação:** implementa ferramentas para comunicação síncrona, como salas de discussão (*chat*), e assíncrona como fóruns, mural e notícias. O fórum caracteriza-se por ser multinível, permitindo restringir certos tópicos dentro de um tópico maior. O mural e as notícias são os meios de divulgação do professor em relação a uma ou mais turmas, sendo o mural usado como um mecanismo de recados curtos relacionados às atividades desenvolvidas em aula, datas de encontros presenciais, formas de entrega das atividades, entre outros; já as notícias, referem-se à divulgação de manchetes pertinentes ao tema estudado e/ou eventos (figura 2).

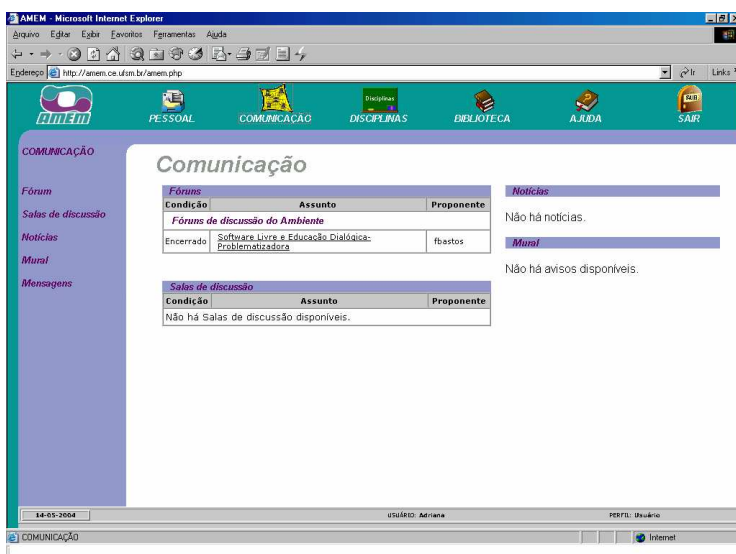


Ilustração 2: Módulo de comunicação

- **Disciplinas:** disponibiliza as disciplinas cadastradas no ambiente. A entrada nesse módulo só é permitida para quem já está cadastrado como aluno ou professor. São apresentadas opções de consulta das disciplinas, lista de disciplinas para as quais se é aluno e/ou professor. No momento em que se define o perfil, o usuário tem acesso às atividades propostas pelo professor, se aluno, ou tem acesso às ferramentas de gerenciamento da disciplina (gerenciamento de turmas, alunos, atividades e materiais didáticos), se professor (figura 3).

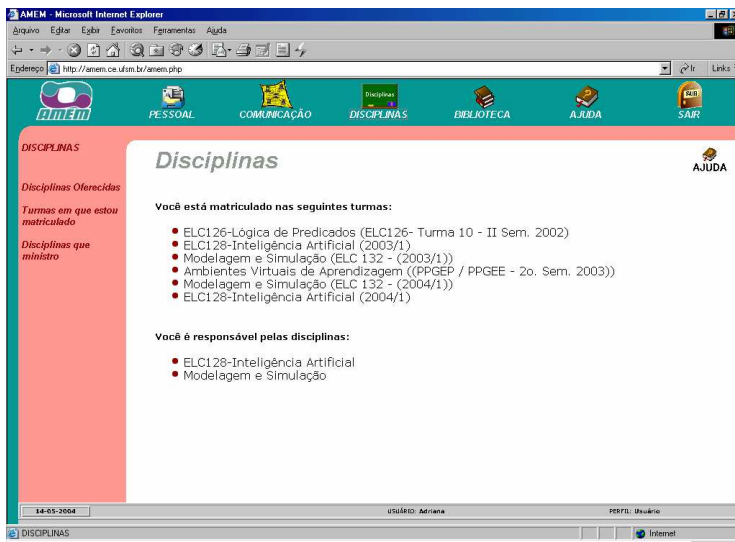


Ilustração 3: Módulo das disciplinas

- **Biblioteca:** é onde se encontram disponíveis os materiais didáticos e as referências bibliográficas para consulta e uso de todos em geral. Os materiais podem estar disponíveis para *download* ou apenas conterem o local onde podem ser encontrados (figura 4).

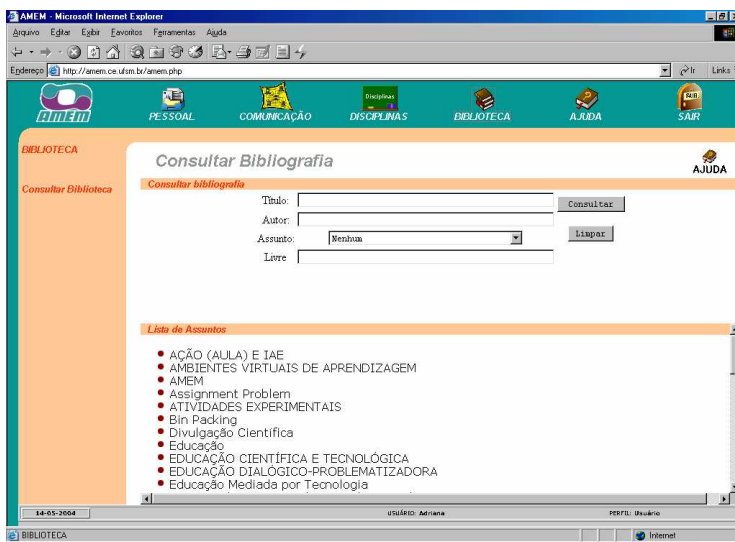


Ilustração 4: Módulo de biblioteca

- **Ajuda:** Disponibiliza um tutorial *on-line* sobre o funcionamento das ferramentas do AMEM, onde se pode encontrar um espaço para o envio de sugestões e/ou críticas e uma sessão de dúvidas organizadas como uma FAQ (Perguntas Mais Freqüentes). Sua interface está representada na figura 5. Maiores informações sobre esse módulo estão disponíveis em Lauermann, (2002).

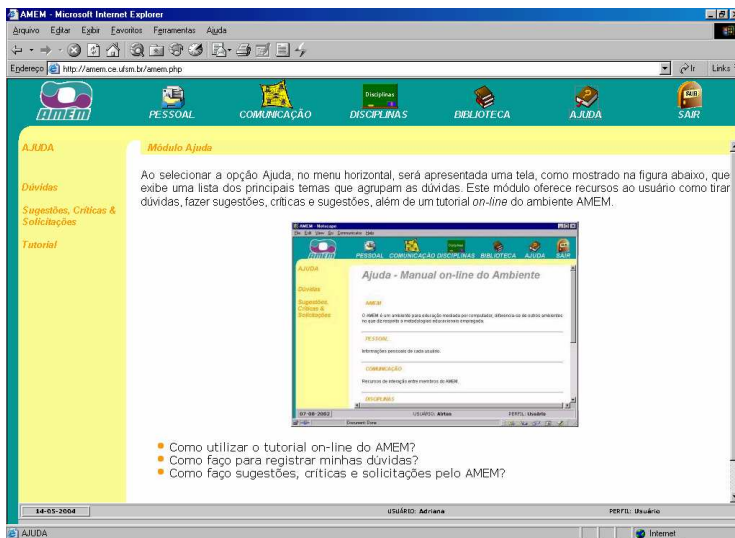


Ilustração 5: Módulo de ajuda

- **Sair:** Este módulo dá ao usuário meios de sair do ambiente, caso possua perfil de aluno ou professor ou então sair de uma determinada turma ou disciplina (figura 6).

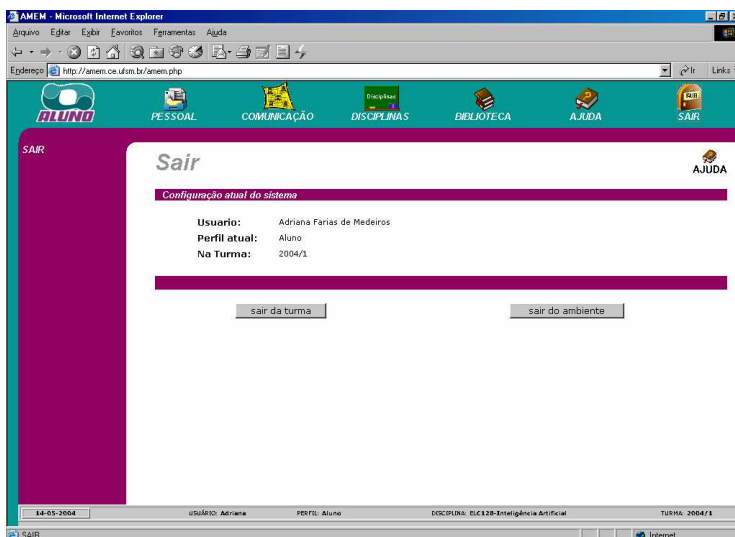


Ilustração 6: Módulo de saída

O ambiente AMEM está disponível no seguinte endereço eletrônico: <http://amem.ce.ufsm.br>.

Após a descrição do funcionamento do ambiente onde estará situado o curso, será explicado o problema pelo qual optou-se problematizar para elaborar o curso proposto.

3 Heurísticas e Metaheurísticas

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. Para resolver problemas com tempos de resolução impraticáveis utilizando algoritmos exatos (testar todas as combinações possíveis), usualmente são aplicadas heurísticas e metaheurísticas. Uma heurística pode ser definida com um algoritmo que encontra uma solução, nem sempre factível, não necessariamente a melhor solução, para um determinado problema com uma determinada função objetiva, num tempo computacional razoável (Diaz, 1996). Sendo que uma solução factível é aquela cujo resultado obtido viola alguma das restrições do problema.

O objetivo de uma heurística é tentar encontrar uma solução “boa” de maneira simples e rápida. É interessante a utilização de heurísticas quando o algoritmo exato conhecido consome alto tempo computacional, ou como passo intermediário para aplicação de outro algoritmo (um algoritmo exato, por exemplo), ou ainda quando temos limitação de tempo ou espaço. Uma outra vantagem é uma maior flexibilidade para manipular diferentes características do problema (Müller, 2002).

A maioria dos problemas NP-Difíceis exige a avaliação de um imenso número de alternativas para determinar a sua solução exata. Heurísticas aumentam a eficiência do processo de busca, geralmente por não explorar todas as alternativas, ou seja, limitar o espaço de busca de melhor solução (Mendes *et al.*, 2002).

Uma heurística, para ser considerada boa precisa apresentar um bom desempenho, encontrando soluções o mais próximo do ótimo em um tempo computacional razoável. Além disso, ela necessita de robustez, rapidez, simplicidade e ter suporte para aceitar múltiplas soluções de partida. Geralmente são baseadas em método de busca, que é um procedimento que determina a ordem em que a vizinhança da solução é explorada, aceita ou gerada. A busca guiada usa informações sobre a natureza do problema e sobre a natureza

do objetivo no sentido de direcionar a busca através de regiões que se mostrem mais promissoras (Mendes *et al.*, 2002).

Segundo Müller (1997), podem-se classificar as heurísticas em 7 grupos demonstrados a seguir:

- a) **Heurísticas de Construção:** constroem uma solução passo a passo, onde a cada passo se adicionam componentes individuais (nós, arcos, variáveis), onde uma solução factível será encontrada ao final do método;
- b) **Heurísticas de Melhoramento:** iniciam com uma solução factível para o problema e buscam através de trocas e inserções melhorar esta solução. Nem sempre ocorre melhora da solução;
- c) **Heurísticas de Programação Matemática:** utiliza modelos de otimização e uma versão adaptada de um procedimento de solução exata para obter um algoritmo heurístico eficiente para o problema;
- d) **Heurísticas com Restrição do Espaço de Soluções:** neste método o espaço de soluções é restrito, portanto o problema se torna mais fácil de resolver através de um algoritmo eficiente. De certa maneira, todas as heurísticas são métodos de restrição, em particular os métodos iterativos de melhoria, que permitem somente soluções factíveis durante seu processo de busca;
- e) **Heurísticas com Relaxação do Espaço de Soluções:** o espaço de soluções factíveis é expandido para se obter um problema relaxado mais simples e mais fácil de se resolver;
- f) **Heurísticas Compostas:** são baseadas na idéia de que duas heurísticas podem ser combinadas de modo a ter um desempenho total significativamente melhor;
- g) **Metaheurísticas:** são métodos computacionais para resolver, de maneira aproximada, problemas de otimização combinatória difíceis, grandes e/ou complexos para os quais não é possível obter soluções em um tempo computacional razoável, independente do computador utilizado. Tais algoritmos organizam e direcionam a busca atuando sobre outros métodos, ditos subordinados, tais como os de busca local e aplicam diferentes estratégias operacionais e organizacionais no sentido de aumentar seu desempenho, possuindo como um de seus objetivos superar duas falhas: o término em um ótimo local muito cedo e a ciclagem, ou seja, fazer e refazer o mesmo

movimento.

Estas mesmas 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 complexos. Em relação às metaheurísticas, podem-se citar como principais métodos: *simulated annealing*, baseada numa busca aleatória e estratégias de aceitação, originada da analogia entre o processo de resfriamento lento e controlado de sólidos e a resolução de problemas de otimização combinatória; *busca tabu* baseada em um processo de restringir e liberar a busca, utilizando informação da estrutura do problema e da maneira como a busca avança na procura por soluções melhores; *algoritmos genéticos* que são algoritmos baseados nos mecanismos de seleção natural e genética dos sistemas biológicos (Smith *et al*, 1996); *algoritmos meméticos* (Coelho, 2001); *busca local genética* (Tanev *et al*, 2001) e *redes neurais*, baseadas em idéias oriundas das técnicas de inteligência artificial.

A avaliação do desempenho das heurísticas conta com três diferentes métodos: análise do pior-caso, análise probabilística (ou caso-médio) e análise empírica. O primeiro estabelece o desvio máximo da otimalidade que pode ocorrer quando uma dada heurística é aplicada a instâncias de um problema e também procura estabelecer limitantes superiores do número de passos que um algoritmo pode realizar para resolver uma instância do problema. O segundo inicia pela caracterização de uma instância de desempenho médio do problema, em termos de uma função de distribuição de probabilidade em relação a todas as classes de instâncias. E o último método é baseado na experiência computacional obtida quando um algoritmo é executado para um conjunto de problemas testes supostamente representativos (Müller, 2002).

Devido à existência de inúmeros problemas de otimização combinatória amplamente divulgados na literatura, optou-se pela escolha de apenas um problema típico para a elaboração da metodologia proposta, conhecido como Problema de *bin packing* ou Problema de Empacotamento que terá sua descrição apresentada na seqüência.

3.1 Definição do problema de *bin packing*

Segundo Cogo *et al* (2001), problemas de *bin packing* são extremamente úteis no estabelecimento das metas de produção de vários segmentos industriais, que possuem como

objetivo a minimização dos efeitos negativos – custos, desperdícios – e igualmente importantes no planejamento de operações que visam à minimização dos espaços ociosos.

Em indústrias de papel, por exemplo, nota-se o interesse pela otimização no processo de cortes de bobina de papel de maneira a diminuir as perdas evidenciando a existência da descrição do problema de *bin packing* em que se considera apenas uma dimensão. Já em indústrias de móveis onde se deseja cortar placas retangulares de madeira para obtenção de peças também retangulares, é necessário considerar duas dimensões, visto que o problema é determinar a quantidade e os tipos de placas a serem compradas, pois cada tipo de placa possui um determinado preço. O problema de *bin packing* também está presente na construção civil, no corte das barras de aço utilizadas para as obras (Cogo *et al.*, 2001).

Para determinarmos o tipo de um problema de corte e *bin packing* aspectos básicos devem ser focados com respeito à estrutura lógica, segundo Arenales (1993):

- Dimensionalidade;
- Tipo de seleção dos objetos/itens;
- Características dos objetos/itens;
- Restrições associadas aos padrões;
- Objetivos;
- Status da informação e variabilidade dos dados;
- Métodos de solução.

Os três primeiros aspectos da estrutura são os mais importantes para o relacionamento dos problemas de corte e *bin packing* (Frazzon *et al.*, 2003).

De uma forma geral, os problemas de *bin packing* consistem em empacotar uma lista de itens em recipientes. Dependendo da natureza dos itens (barra, placas, caixas, dentre outros) a serem cortados se diz que os problemas são unidimensionais, bidimensionais ou tridimensionais (Frazzon *et al.*, 2003).

Podemos descrever o problema de *bin packing* como:

Considerando **n itens**, com **pesos w_i** e **capacidade C** , todos números **inteiros, positivos, indivisíveis e menores ou iguais ao valor da capacidade**, o problema é encontrar **o menor número necessário de bins**, para armazenar **todos os itens existentes**, sempre respeitando a restrição da capacidade, ou seja, **o somatório dos pesos w_i dos itens não deve ultrapassar a capacidade do bin** (Martello e Toth, 1990) (grifos nossos).

Em analogia com o mundo real, *bin* é uma caixa que armazenará os itens após sua alocação. Considerando os itens como livros, as caixas onde eles serão guardados serão os *bins*.

Para diminuir a complexidade do problema, e assim encontrar a solução num tempo mais eficiente, devemos utilizar recursos que reduzam o “espaço de busca”, como os limitantes (*bounds*) inferiores e superiores e procedimentos que diminuam o tamanho da instância analisada (redução).

O limitante inferior indica que a solução ótima do problema, ou a melhor solução possível terá um valor maior ou igual a esse limitante. Então, caso o limitante inferior de um determinado problema for quatro, significa que o menor número de *bins* possível será quatro ou mais, nunca menos, excluindo as tentativas de buscas de soluções com menos *bins*, evitando trabalhos sem nenhum resultado factível, e ainda diminuindo o tempo de execução do algoritmo (Martello e Toth, 1990). A determinação desse limitante é feita através de cálculos matemáticos.

Um limitante superior, por sua vez, indica que a melhor solução para o problema será sempre menor ou igual ao valor encontrado, como no exemplo citado, com limitante inferior igual a quatro, se o limitante superior for sete, sabe-se que a solução ótima, ou a melhor encontrada está no intervalo de quatro a sete (Frazzon *et al*, 2003). Esse limitante pode ser calculado por algoritmos aproximativos, que podem ser caracterizados por um conjunto de técnicas e algoritmos computacionalmente muito eficientes mas que não garantem a solução ótima do problema (Luna, 2000).

Eles podem ser divididos em duas classes:

On-line: Os dados não são previamente conhecidos, podendo assim, serem determinados dinamicamente;

Off-line: Os dados são previamente conhecidos e ordenados em ordem não-crescente;

Ainda tentando minimizar o tempo e a dificuldade de resolução do problema dispõe-se dos chamados métodos de redução. Esses métodos visam diminuir o tamanho da instância a ser analisada, fixando itens a *bins*, através de uma pré-análise numérica do tamanho do item, diminuindo assim os itens que serão distribuídos e por consequência diminuindo a dificuldade do algoritmo (Martello e Toth, 1990).

Partindo dessa formulação, pode-se encontrar uma segunda versão para o problema, o problema de *bin packing* dual, onde o que se busca é a minimização da capacidade de um número fixo de *bins*, de forma que o somatório dos itens que estão alocados não ultrapasse a atual capacidade. Tanto o Problema de *bin packing* unidimensional quanto o Problema de *bin packing* dual guardam estreita relação com o problema de seqüenciamento em máquinas paralelas idênticas com o objetivo de minimizar o *makespan*, este sendo o somatório dos tempos de processamento das tarefas no processador mais carregado (Scholl *et al*, 1997).

O problema de *bin packing* também pode ser resolvido utilizando-se metaheurísticas. Um exemplo disso é o algoritmo genético desenvolvido por Emmanuel Falkenauer (1996) que ficou conhecido como *Hybrid Grouping Genetic Algorithm* (HGGA). Os conceitos de algoritmos genéticos e o funcionamento do HGGA será explicado na seqüência.

3.1.1 Algoritmos genéticos e o HGGA

Algoritmos genéticos são baseados na analogia entre o processo de evolução natural, onde populações evoluem de acordo com os princípios de seleção natural e a sobrevivência dos mais aptos (evolução das propriedades genéticas da população) e os indivíduos mais bem sucedidos em adaptar-se a seu ambiente terão maior chance de sobreviver e de reproduzir. Genes dos indivíduos mais bem adaptados vão espalhar-se para um maior número de indivíduos em sucessivas gerações.

A população é representada pelos seus cromossomos, sendo um cromossomo igual a um indivíduo. Novos cromossomos gerados a partir da população corrente são incluídos na população, enquanto outros são excluídos. A geração de novos cromossomos é através de mecanismos de reprodução e de mutação.

- Reprodução: selecionar os cromossomos pais e executar uma operação de *crossover*, que é uma combinação simples das representações de cada cromossomo.
- Mutação: modificação arbitrária de uma parte (pequena) do cromossomo

Mediante esses conceitos, o HGGA é um algoritmo genético modificado que incorpora algoritmo de busca local com estratégias de *First Fit Decreasing* (FFD). Sendo que o FFD trabalha com os itens previamente conhecidos e ordenados em ordem não-crescente de peso. A sua execução consiste em alocar os itens em um *bin* enquanto o somatório dos pesos não ultrapassar a capacidade do bin, caso haja violação da capacidade

do *bin* corrente, o FFD pesquisa nos *bins* já abertos ou então cria um novo *bin* para o item a ser alocado. O HGGA utiliza critérios de dominância e redução do problema. Atualmente é considerado o algoritmo de melhor desempenho conhecido (Bernardi, 2001).

Nos testes realizados por Falkenauer (1996) relata-se que o HGGA apresenta um melhor desempenho em termos de qualidade da solução (número de sublistas utilizadas) e tempo de execução quando comparado com o algoritmo exato de Martello e Toth (1990). Segundo Falkenauer (1996), a eficácia do HGGA é acentuada quando o número de itens aumenta. Seus resultados mostram que o HGGA utiliza um menor número de sublistas e foi capaz de produzir uma solução em um tempo menor que um método exato (Bernardi, 2001).

Tendo posse dos resultados dos testes, Falkenauer realizou a comparação entre o desempenho de seu algoritmo com um procedimento *branch and bound* (cálculo de limitantes) em termos de qualidade da solução (número de sublistas utilizadas) e tempo de execução. O propósito desta comparação foi mostrar que o HGGA apresenta-se como um dos melhores algoritmos para o problema de *bin packing* unidimensional (Bernardi, 2001).

Tendo em vista que heurísticas e metaheurísticas fazem parte da maioria das disciplinas de inteligência artificial ministradas nos cursos de ciência da computação, o seu ensino é pedagógica e tecnologicamente apresentado de forma desinteressante e dependente da fórmula *giz + quadro-negro*. O ensino de heurísticas e metaheurísticas, que usualmente apresenta um embasamento matemático forte e necessita de um grande número de passos para que o processo faça sentido, pode tornar-se desestimulante se realizado através de um processo didático tradicional. Também é importante salientar que a literatura é pobre no que concerne ao estudo didático de tais conteúdos, concentrando-se normalmente na definição e comparação de algoritmos e suas complexidades.

Mediante essa realidade está sendo proposto um curso que promova uma maior interação entre os envolvidos (educadores – educandos) no processo da aprendizagem. Esse curso será descrito no capítulo a seguir.

4 O ensino de heurísticas e metaheurísticas sob a perspectiva da investigação-ação educacional no ambiente AMEM utilizando o problema de *bin packing*

Quando na literatura há referências ao ensino de heurísticas e metaheurísticas têm-se as atenções voltadas, principalmente, para a estrutura, 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).

Porém existe um grande vácuo tecnológico e pedagógico que precisa ser preenchido para que o ensino das técnicas de heurísticas e metaheurísticas se tornem mais acessíveis 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 prove 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 incumbentes, assim como os parâmetros alteram as mesmas. No entanto, 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.

Devido à realidade demonstrada anteriormente, optou-se por fazer a descrição de exemplos de metodologias alternativas para o ensino de heurísticas e metaheurísticas. Esses exemplos estão descritos na seção a seguir.

4.1 Ensino de heurísticas e Metaheurísticas

Dentre as metodologias 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 é definida por (Hearn *et al*, 1997) 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 educados. 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 (Garcia *et al*, 1996), (Martins *et al*, 2003) e (Zachary, 2004).

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 absorção do funcionamento das heurísticas seja ainda mais intensa.

Outro exemplo de metodologia alternativa para o ensino de heurísticas e metaheurísticas, e fazendo uma analogia com pesquisa operacional (PO), pode ser encontrado no artigo escrito por Ricardo Villaroel Dávalos (Dávalos, 2002), que descreve a metodologia usada na Universidade do Sul de Santa Catarina (UniSul) para o ensino deste tema, bem como os recursos tecnológicos que podem ser utilizados em conjunto com as aulas.

Segundo Dávalos (2002) os planos de ensino propostos nos cursos em que é ministrado a PO apresentam uma metodologia que está definida por aulas expositivas, seminários, trabalhos de pesquisa, exercícios teóricos e práticos, assim como o atendimento paralelo aos alunos. Também utilizam recursos computacionais como planilhas eletrônicas, linguagens de programação, pacotes específicos e pesquisas na Internet. Além disso, são utilizados aplicativos gráficos como Solver (baseado em planilhas eletrônicas) e Lindo (baseado em linguagens de modelagem algébricas). Como trabalho final da disciplina os alunos apresentam um projeto de implementação de um sistema com características reais, em um dos aplicativos comentados anteriormente, em que o objetivo é aplicar os conhecimentos adquiridos na disciplina e propor algumas alternativas para uma melhoria do desempenho do sistema a ser aplicado.

Analisando as metodologias citadas anteriormente e dispondo dos operacionalizadores citados no capítulo um, propõe-se aqui uma metodologia alternativa que será explicada na próxima seção.

4.2 Proposta de metodologia

Desta forma, este trabalho motiva-se na melhoria dos processos pedagógicos e instrucionais da área de heurísticas e metaheurísticas baseadas também nos paralelos que podem ser realizados entre os três momentos pedagógicos propostos por (De Bastos e Müller, 1999) e a construção de uma solução heurística para um problema complexo: *desafio inicial X algoritmo construtivo, melhor solução educacional no momento X heurísticas e desafio mais amplo X metaheurísticas.*

Considerando que: (a) a Otimização Combinatória é uma tecnologia chave nas organizações empresariais e educacionais, na medida em que seus processos inerentes aumentam em complexidade; (b) os cursos de graduação e pós-graduação nas áreas de ciências e engenharias tem aumentado o número de disciplinas que utilizam a Otimização Combinatória; (c) há uma complexidade inerente nos métodos e algoritmos utilizados nesta área, que exigem um grande número de passos e repetições intensivas; deseja-se investigar uma nova concepção didático-pedagógica para o ensino de heurísticas e metaheurísticas. Além disso, pretende-se avaliar a sistemática a ser desenvolvida em um ambiente real. Como uma das principais contribuições para a área de Informática na Educação, pode-se citar o estudo da perspectiva dialógica-problematizadora em um tema essencialmente

problematizador; onde a experimentação é a base para a compreensão e que, contudo, não possui ferramentas adequadas para o mesmo.

Considerando então a perspectiva da educação dialógica-problematizadora, o curso tem os seguintes componentes básicos de educação, que formam a base da matriz dialógica-problematizadora.

Professores responsáveis: Professor responsável pela disciplina de Inteligência Artificial e Alunos em docência orientada.

O público alvo: Alunos de graduação em Ciência da Computação na Universidade Federal de Santa Maria.

Tema do curso: Heurística e Metaheurísticas para o problema *bin packing*.

Contexto em que será aplicado o curso: Aulas de heurísticas e metaheurísticas do curso de Ciência da Computação utilizando o AMEM.

O curso está dividido em quinze encontros de três horas cada, feitos de forma presencial ou à distância via ambiente AMEM. Em cada encontro será abordado um tópico diferente do tema estudado, além de fazer uso de trabalhos práticos de implementação dos algoritmos heurísticos e metaheurísticos vistos em aula.

A metodologia proposta reúne os conceitos de IAE, explicados anteriormente, com a problematização do conhecimento e com os operacionalizadores presentes no AMEM.

4.2.1 MDP proposta

A MDP que norteia o curso proposto (tabela 1):

Tabela 1: MDP proposta

	A-Professores	B-Alunos	C-Tema	D-Contexto
I – Professores	[A1] Os professores possuem conhecimentos básicos sobre heurísticas e metaheurísticas para o <i>bin packing</i> e sobre a investigação	[B1] Os alunos efetivamente estão aproveitando os operacionalizadores pedagógicos para o entendimento das heurísticas e metaheurísticas?	[C1] A abordagem dialógica-problematizadora potencializa o ensino de Heurísticas e Metaheurísticas?	[D1] As aulas de heurísticas e metaheurísticas utilizando a investigação ação-educacional e o AMEM favorecem no processo de

	ação-educacional que forma a base do AMEM?			ensino-aprendizagem dos professores?
2 – Alunos	[A2] Como os professores podem instigar os alunos, utilizando a problematização e o AMEM?	[B2] Os alunos efetivamente estão compreendendo o que são heurísticas e metaheurísticas através da abordagem baseada em problemas?	[C2] Quais são as dificuldades e os avanços do ensino de Heurísticas e Metaheurísticas utilizando <i>bin packing</i> ?	[D2] Como verificar se os alunos estão compreendendo o conteúdo de Heurísticas e Metaheurísticas que está sendo desenvolvido através da investigação ação-educacional, presente no AMEM?
3 – Tema	[A3] Como os professores têm construído a organização didática de suas aulas nas práticas de investigação-ação educacional de Heurísticas e Metaheurísticas?	[B3] Os estudantes têm assumido que os problemas vistos em aula realmente contribuem no processo de aprendizagem de Heurísticas e Metaheurísticas?	[C3] Será que o <i>bin packing</i> é adequado para o ensino de Heurísticas e Metaheurísticas?	[D3] As aulas de heurísticas e metaheurísticas contribuem para a evolução do conhecimento científico dos problemas propostos?
4 – Contexto	[A4] Os professores conseguem adotar a prática de investigação ação-educacional no ensino de heurísticas e metaheurísticas?	[B4] Os alunos têm compreendido as questões de heurísticas e metaheurísticas com a prática de investigação-ação educacional e o AMEM?	[C4] De que forma o problema de <i>bin packing</i> e o AMEM contribuem para o ensino de heurísticas e metaheurísticas?	[D4] Quais são as contribuições do ensino de heurísticas e metaheurísticas sob a perspectiva da investigação-ação educacional no ambiente AMEM?

4.2.2 Aulas

O curso foi dividido em 15 aulas, sendo 13 aulas presenciais e 2 aulas utilizadas para discussão on-line acerca dos temas propostos.

A seguir tem-se a descrição das aulas estruturadas em desafio inicial com a problematização do conhecimento (problemas que possam ser resolvidos no mundo real), melhor solução educacional de momento, com explicação do tema do desafio inicial e uma provável solução para o problema proposto; e o desafio mais amplo, com a proposta de um novo problema relacionado com a próxima aula.

Aula 1: Aula de apresentação da disciplina

Objetivo: apresentar a disciplina, sua metodologia e o ambiente AMEM.

Elementos da MDP: A1, A2 e C4. Esses elementos se justificam pela criação do pacto entre professor e aluno que obrigatoriamente deve ser realizado na primeira aula para uma boa condução dos trabalhos.

Programação:

a) Atividade Um:

Duração: 60 minutos

Discussão da metodologia de ensino empregada na disciplina, explicando os três momentos pedagógicos que poderão fazer parte das aulas posteriores. Neste encontro, deve-se deixar bem claro aos estudantes que a condução das aulas é realizada colaborativamente, mas que as tarefas agendadas deverão ser realizadas dentro dos prazos. Os alunos precisam assumir que o AMEM é um instrumento valioso para a condução da disciplina e que devem utilizá-lo de forma completa.

Atividade de Colaboração: leitura do texto sobre o problema de *bin packing*, que será abordado durante toda a disciplina. Os alunos poderão colaborar enviando sugestões e comentários através do AMEM. Bibliografia: Algoritmo Interativo para o Problema de Empacotamento (Frazzon *et al*, 2003).

Aula 2: Noções sobre bin packing utilizando Lego®

Objetivo: apresentar os tipos de problemas de *bin packing* e sua relação com a pesquisa operacional

Elementos da MDP: B1, B2. A escolha desses elementos baseia-se na observação da condução do processo de ensino-aprendizagem dos alunos, elemento crucial dessa aula.

Programação:

a)Atividade Um:

Duração: 40 minutos

Apresentação inicial sobre o que é o problema de *bin packing*, como ele pode ser subdividido (unidimensional, bidimensional e tridimensional) e porque a solução deste tipo de problema é considerada difícil e complexa.

b)Atividade Dois:

Duração: 50 minutos

Desafio Inicial: Você trabalha no setor de estoque de uma grande empresa. O seu chefe mandou você empacotar diversos objetos de tamanhos diferentes que serão enviados a um cliente. A empresa utiliza uma distribuição satisfatória, porém seu chefe disse que caso você consiga encontrar uma distribuição para os objetos que ocupe menos caixas do que o normal, você ganhará uma bonificação salarial. Encontre uma distribuição melhor, sistematizando o processo, primeiro considerando o problema com uma dimensão, depois duas. Ao acondicionar as peças nas caixas, não pode haver empilhamento das mesmas.

Dimensão das peças para duas dimensões: cada retângulo mede 1cm de base, por 1 de largura e 1 de altura.

Dimensão das caixas: 8 cm de comprimento por 3 de largura e 1 de altura.

Dimensão das peças para uma dimensão: cada retângulo: 1cm X 1cm X 1cm

Dimensão das caixas: 8 cm de comprimento X 1cm de altura X 1cm de largura

As peças e as distribuições utilizadas estão demonstradas nas figuras a seguir:

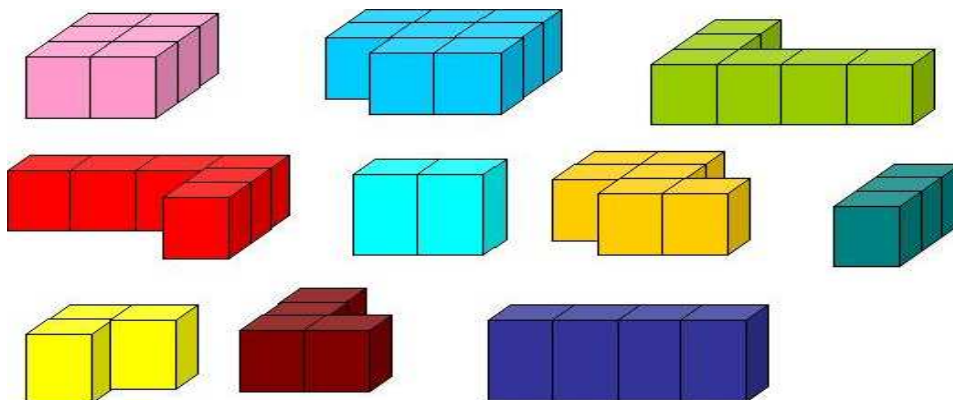


Ilustração 7: Peças para o problema com duas dimensões

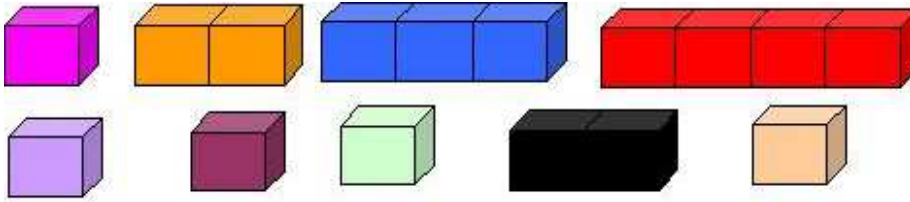


Ilustração 8: Peças para uma dimensão

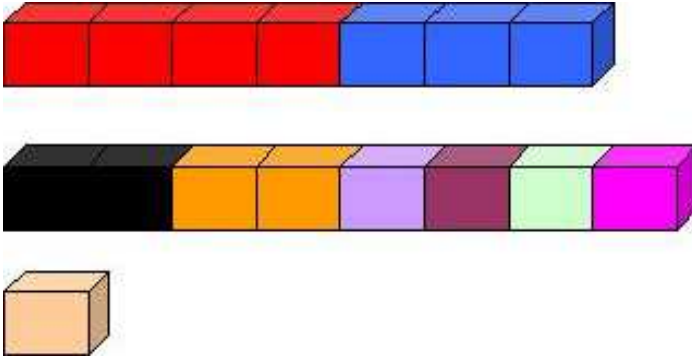


Ilustração 9: Distribuição utilizada pela empresa considerando uma dimensão feita utilizando o algoritmo NFD.

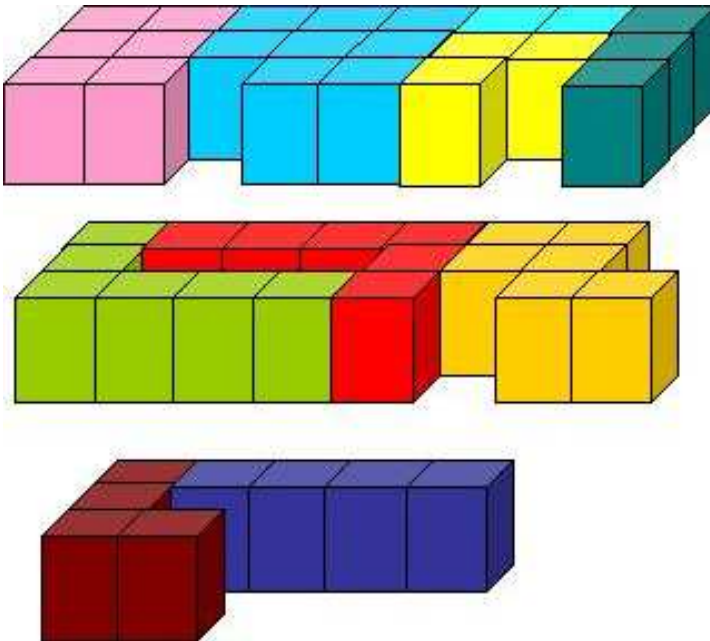


Ilustração 10: Distribuição utilizada pela empresa para o problema com duas dimensões feita utilizando o algoritmo FFD.

c)Atividade Três:

Duração: 60 minutos

Melhor solução educacional do momento: Sistematização do problema proposto aplicando uma heurística construtiva, por exemplo, FFD:

- 1 dimensão:

- Ordenar as peças por tamanho em ordem não-crescente. Resultado representado na ilustração a seguir.

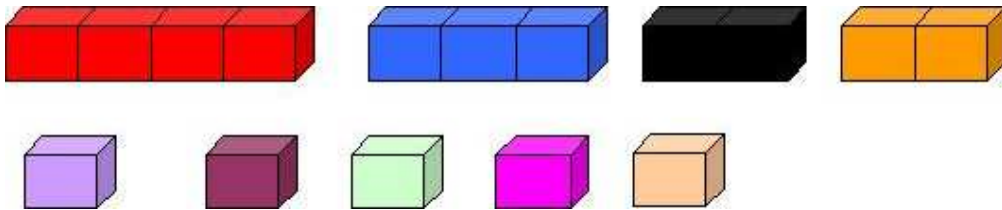


Ilustração 11: Peças ordenadas

- colocar os itens nos *bins* enquanto couberem, se não couberem, criar um novo e assim sucessivamente, pesquisa-se nos *bins* anteriores. Resultado representado na ilustração a seguir.

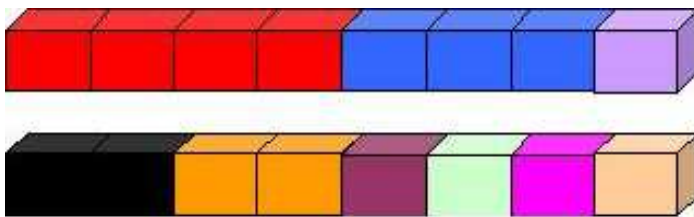
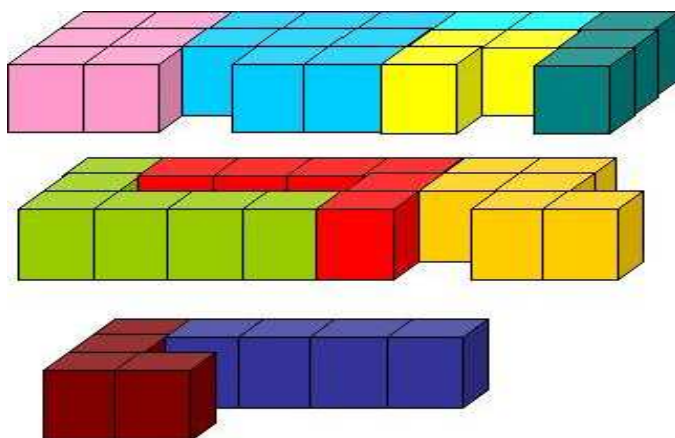


Ilustração 12: Distribuição encontrada

- 2 dimensões:

Considerar os *bins* na ordem apresentada.



Ir alocando as peças enquanto couberem, quando não couber mais, criar um novo bin. O resultado está representado na ilustração a seguir:

Ilustração 13: Distribuição encontrada

Atividade extraclasse:

Desafio mais amplo: Dado um problema de *bin packing* com 20 itens, sendo a capacidade de cada *bin* de 100, construa uma heurística que encontre a melhor solução possível para o problema.

Itens: 99, 43, 19, 7, 37, 94, 6, 79, 64, 32, 18, 50, 46, 3, 77, 25, 2, 5, 33, 67.

Solução usando BFD:

- Ordenar os itens: 99, 94, 79, 77, 67, 64, 50, 46, 43, 37, 33, 32, 25, 19, 18, 7, 6, 5, 3, 2.
- Vá pegando os itens na seqüência e alocando nos *bins* e avaliando a capacidade;
- Se um dos itens ultrapassar a capacidade, verifique a possibilidade de inclusão nos *bins* já existentes de maneira que a capacidade residual seja a menor possível ou abra um novo bin.
- Repita esse processo enquanto houverem itens não alocados.

Solução:

Bin 1: 99

Bin 2: 94, 6

Bin 3: 79, 19

Bin 4: 77, 18, 5

Bin 5: 67, 33

Bin 6: 64, 32, 3

Bin 7: 50, 46, 2

Bin 8: 43, 37, 7

Bin 9: 25

Atividade de Colaboração: leitura do texto sobre algoritmos exatos. Os alunos poderão colaborar enviando sugestões e comentários através do AMEM. Bibliografia: *Knapsack Problems: Algorithms and Computer Implementations* (Martello e Toth, 1990).

Aula 3: Algoritmos Exatos

Objetivo: compreender questões acerca do tamanho, complexidade e tempo computacional envolvidos na construção de algoritmos exatos para um problema de programação em pesquisa operacional.

Elementos da MDP: B1, B2. A escolha desses elementos baseia-se na observação da condução do processo de ensino-aprendizagem dos alunos, elemento crucial dessa aula.

Programação:

a)Atividade Um:

Duração: 50 minutos

Desafio Inicial: Você precisa fazer uma distribuição de cinco itens em *bins*, de tal forma que encontre a melhor distribuição para minimizar o número de *bins* necessários. Considerando os dados abaixo, teste todas as soluções possíveis, de forma sistemática, garantindo assim, que a melhor solução seja encontrada.

Capacidade do bin: 100

Itens: 99, 60, 40, 37 e 21

b)Atividade Dois:

Duração: 40 minutos

Melhor solução educacional do momento: Considere a enumeração somente das possíveis soluções, ou seja, que não extrapolem a capacidade. Para tanto, é realizada uma enumeração, considerando todas as possibilidades de arranjo entre os cinco elementos, desde que os pesos dos itens não extrapolem a capacidade.

- Caso o item de maior peso somado ao item de menor peso seja superior a capacidade, aloque um *bin* para o item de maior peso e tire-o do arranjo.
- Enumere todas as possibilidades de solução, combinando os *bins* de dois a dois até n a n , sendo n o número de *bins* restantes.

Os arranjos encontrados para esse exemplo estão mostrados na tabela a seguir:

Tabela 2: Arranjos para o problema apresentado

<i>Bin 1:</i> 99	<i>Bin 1:</i> 99	<i>Bin 1:</i> 99	<i>Bin 1:</i> 99
<i>Bin 2:</i> 60	<i>Bin 2:</i> 60, 40	<i>Bin 2:</i> 60, 37	<i>Bin 2:</i> 60, 21
<i>Bin 3:</i> 40	<i>Bin 3:</i> 37	<i>Bin 3:</i> 40	<i>Bin 3:</i> 37
<i>Bin 4:</i> 37	<i>Bin 4:</i> 21	<i>Bin 4:</i> 21	<i>Bin 4:</i> 40
<i>Bin 5:</i> 21			
Total de <i>Bins:</i> 5	Total de <i>Bins:</i> 4	Total de <i>Bins:</i> 4	Total de <i>Bins:</i> 4

<i>Bin 1: 99</i>	<i>Bin 1: 99</i>	<i>Bin 1: 99</i>	<i>Bin 1: 99</i>
<i>Bin 2: 60</i>	<i>Bin 2: 60</i>	<i>Bin 2: 60</i>	<i>Bin 2: 60</i>
<i>Bin 3: 40, 37</i>	<i>Bin 3: 40, 21</i>	<i>Bin 3: 40</i>	<i>Bin 3: 40, 37, 21</i>
<i>Bin 4: 21</i>	<i>Bin 4: 37</i>	<i>Bin 4: 37, 21</i>	
Total de <i>Bins</i> : 4	Total de <i>Bins</i> : 4	Total de <i>Bins</i> : 4	Total de <i>Bins</i> : 3
<i>Bin 1: 99</i>	<i>Bin 1: 99</i>	<i>Bin 1: 99</i>	
<i>Bin 2: 60, 40</i>	<i>Bin 2: 60, 37</i>	<i>Bin 2: 60, 21</i>	
<i>Bin 3: 37, 21</i>	<i>Bin 3: 40, 21</i>	<i>Bin 3: 40, 37</i>	
Total de <i>Bins</i> : 3	Total de <i>Bins</i> : 3	Total de <i>Bins</i> : 3	

c)Atividade Três:

Duração: 60 minutos

Apresentação e consolidação dos conceitos relativos a algoritmos exatos.

Atividade extraclasse:

Desafio mais amplo: Como é possível garantir que a solução encontrada por vocês ou em qualquer outro caso, seja considerada ótima, ou seja, que não possua uma solução melhor?

Atividade de Colaboração: leitura do texto sobre métodos de redução. Os alunos poderão colaborar enviando sugestões e comentários através do AMEM. Bibliografia: *Knapsack Problems: Algorithms and Computer Implementations* (Martello e Toth, 1990).

Aula 4: Métodos de Redução

Elementos da MDP: B1, B2. A escolha desses elementos baseia-se na observação da condução do processo de ensino-aprendizagem dos alunos, elemento crucial dessa aula.

Objetivo: compreender o que são métodos de redução e sua influência na construção de soluções para problemas de *bin packing*.

Programação:

a)Atividade Um:

Duração: 50 minutos

Desafio Inicial: A grande maioria dos problemas reais de *bin packing* trabalha com um grande número de itens. Neste caso, torna-se interessante um método que diminua o

problema, reduzindo os itens que podem ser alocados previamente, sem perda de informação do problema. Considerando o problema abaixo, desenvolva um método sistemático e único que possa reduzir o problema, de tal forma que possa ser aplicado em outras instâncias do problema.

Itens: 99, 43, 19, 7, 37, 94, 6, 79, 64, 32, 18, 50, 46, 3, 77, 25, 2, 5, 33, 67.

Capacidade: 99

b)Atividade Dois:

Duração: 60 minutos

Melhor Solução educacional do momento: será aplicado o algoritmo de redução de Martello e Toth(1990).

Passo 1. Ordene os itens em ordem não-crescente;

Passo 2. Verifique se os itens mais carregados (primeiros da lista), não cabem exatamente na capacidade (sem folga). Neste caso, aloque um *bin* para cada um dos itens e retire-os da lista.

Passo 3. Faça IM (Item mais carregado) = primeiro item da lista e IC (Item menos carregado) = último item da lista

Passo 3.1 Se $IM+IC = \text{capacidade}$, então aloque um *bin* para os dois itens e retire-os da lista. Vá para o passo 3.

Passo 3.2 Se $IM+IC > \text{capacidade}$, então aloque um *bin* para o IM e retire-o da lista. Vá para o passo 3.

Passo 3.3 Se $IM+IC < \text{capacidade}$, então faça $IC=IC-1$. Vá para o passo 3.1.

Passo 3.4 Se $IM=IC$, vá para o passo 4.

Passo 4. Inicie o processo de alocação dos itens restantes.

Solução:

1. 99, 94, 79, 77, 67, 64, 50, 46, 43, 37, 33, 32, 25, 19, 18, 7, 6, 5, 3, 2.

2.O primeiro item tem a mesma capacidade do *bin*, logo, ele é alocado e eliminado da lista.

3.Lista: 94, 79, 77, 67, 64, 50, 46, 43, 37, 33, 32, 25, 19, 18, 7, 6, 5, 3, 2

IM=94, IC=2

3.1 $94+2$ não é igual a 99

3.2 $94+2$ não é maior que 99

3.3 $94+2$ é menor que 99, logo $IC=3$

3.1 $94+3$ não é igual a 99

3.2 $94+3$ não é maior que 99

3.3 $94+3$ é menor que 99, logo $IC=5$

3.1 $94+5$ é igual a 99, logo aloque 94 e 5 num *bin* e retire-os da lista

3. Lista: 79, 77, 67, 64, 50, 46, 43, 37, 33, 32, 25, 19, 18, 7, 6, 3, 2

$IM=79$, $IC=2$

(continua)

c)Atividade Três:

Duração: 50 minutos

Apresentação formal dos conceitos sobre redução, limitantes superiores e limitantes inferiores.

Atividade extraclasse:

Desafio mais amplo: Há alguns casos, que a soma de dois itens não é suficiente para reduzir o problema. Neste caso, é possível pensar na redução para três itens em cada bin. Desenvolva um procedimento sistemático baseado no exposto anteriormente para que o algoritmo teste também a soma de três itens por *bin* na redução.

Atividade de Colaboração: leitura do texto sobre algoritmos construtivos. Os alunos poderão colaborar enviando sugestões e comentários através do AMEM. Bibliografia: Heurísticas e Metaheurísticas (Müller, 2002).

Aula 5: Algoritmos construtivos

Objetivo: compreender como uma solução pode ser construída de forma genérica, sistemática e possivelmente gerando boas respostas para o problema de *bin packing*.

Elementos da MDP: B1, B2. A escolha desses elementos baseia-se na observação da condução do processo de ensino-aprendizagem dos alunos, elemento crucial dessa aula.

Programação:

a)Atividade Um:

Duração: 40 minutos

Desafio Inicial: A rede de supermercados *3 Primos* quer adquirir o *Robô Packer* de sua empresa, que é capaz de empacotar automaticamente as compras dos clientes. No entanto a compra só será concretizada se o robô provar que pode diminuir os custos com o número de sacolas. Sabendo que o robô possui sensores que detectam o peso e o volume de cada item da compra e que as sacolas são todas iguais, sistematize o processo para o robô decidir como ele deve fazer o empacotamento com o menor número de sacolas possível. Um exemplo de lista de compras que o robô deverá empacotar está representado na tabela a seguir:

Sacola: máximo de cinco quilos

Tabela 3: Lista de compras do supermercado

Itens:	Peso
Açúcar	5,0 kg
Arroz	3,0 kg
Feijão	3,0 kg
Farinha de mandioca	2,0 kg
Farinha de rosca	1,0 kg
Farinha de milho	1,0 kg
Farinha de trigo	2,0 kg
Fubá	0,5 kg
Sal	1,0 kg
Café	3,0 kg
Leite Condensado	0,5 kg
Creme de leite	0,7 kg
Achocolatados	0,8 kg
Maisena	0,5 kg
Azeite	1,0 kg
Óleo	1,0 kg
Maionese	2,4 kg
Extrato de tomate	0,3 kg
Molho pronto	0,4 kg
Ketchup	0,5 kg

Molho inglês	0,2 kg
Mostarda	0,4 kg
Ervas aromáticas	0,1 kg
Temperos	0,2 kg
Molhos de pimenta	0,3 kg
Gelatinas	0,2 kg
Leite de coco	0,5 kg
Coco ralado	1,0 kg

b)Atividade Dois:

Duração: 50 minutos

Melhor solução educacional do momento: Sistematização do problema utilizando algoritmos construtivos. O algoritmo implementado no exemplo foi o *Best-Fit Decreasing*.

Ordenar os itens:

Tabela 4: Itens ordenados por peso

Itens:	Peso
Açúcar	5,0 kg
Arroz	3,0 kg
Feijão	3,0 kg
Café	3,0 kg
Maionese	2,4 kg
Farinha de mandioca	2,0 kg
Farinha de trigo	2,0 kg
Farinha de rosca	1,0 kg
Farinha de milho	1,0 kg
Sal	1,0 kg
Azeite	1,0 kg
Óleo	1,0 kg
Coco ralado	1,0 kg
Achocolatados	0,8 kg

Creme de leite	0,7 kg
Fubá	0,5 kg
Leite Condensado	0,5 kg
Maisena	0,5 kg
Ketchup	0,5 kg
Leite de coco	0,5 kg
Molho pronto	0,4 kg
Mostarda	0,4 kg
Extrato de tomate	0,3 kg
Molhos de pimenta	0,3 kg
Molho inglês	0,2 kg
Temperos	0,2 kg
Gelatinas	0,2 kg
Ervas aromáticas	0,1 kg

Passo 1. Vá pegando os itens na seqüência e alocando nas sacolas e avaliando o seu peso;

Passo 2. Se um dos itens ultrapassar o peso da sacola, verifique a possibilidade de inclusão nas sacolas já existentes de maneira que a capacidade residual seja a menor possível ou abra uma nova sacola;

Passo 3. Repita esse processo enquanto houverem itens não alocados.

Distribuição:

Sacola 1: Açúcar

Sacola 2: Arroz, farinha de mandioca

Sacola 3: Feijão, farinha de trigo

Sacola 4: Café, farinha de rosca, farinha de milho

Sacola 5: Maionese, sal, azeite, fubá, ervas aromáticas

Sacola 6: Óleo, coco ralado, achocolatados, creme de leite, leite condensado, maisena, ketchup

Sacola 7: Molho pronto, mostarda, extrato de tomate, molhos de pimenta, molho inglês, temperos, gelatinas

c)Atividade Três:**Duração:** 60 minutos

Apresentação dos conceitos acerca dos algoritmos construtivos, formas de sistematizar uma solução e problemas comuns encontrados neste tipo de algoritmo.

Atividade extraclasse:

Desafio mais amplo: Considere a mesma lista de produtos do desafio inicial, porém agora você deve encontrar uma distribuição de sacolas levando em consideração o peso e o volume de cada item. Sendo o volume máximo suportado pela sacola de 40 cm^3 e o peso de 5,0 kg. Sistematize o processo de maneira a encontrar uma distribuição que seja satisfatória para a realidade apresentada no desafio inicial. A lista de compras que deve ser empacotada está representada na tabela a seguir:

Tabela 5: Lista de compras a serem empacotadas.

Itens:	Peso	Volume
Açúcar	5,0 kg	40 cm^3
Arroz	3,0 kg	25 cm^3
Feijão	3,0 kg	20 cm^3
Farinha de mandioca	2,0 kg	10 cm^3
Farinha de rosca	1,0 kg	10 cm^3
Farinha de milho	1,0 kg	10 cm^3
Farinha de trigo	2,0 kg	15 cm^3
Fubá	0,5 kg	07 cm^3
Sal	1,0 kg	10 cm^3
Café	3,0 kg	40 cm^3
Leite Condensado	0,5 kg	20 cm^3
Creme de leite	0,7 kg	30 cm^3
Achocolatados	0,8 kg	13 cm^3
Maisena	0,5 kg	10 cm^3
Azeite	1,0 kg	20 cm^3
Óleo	1,0 kg	17 cm^3
Maionese	2,4 kg	29 cm^3

Extrato de tomate	0,3 kg	08 cm ³
Molho pronto	0,4 kg	05 cm ³
Ketchup	0,5 kg	15 cm ³
Molho inglês	0,2 kg	12 cm ³
Mostarda	0,4 kg	17 cm ³
Ervas aromáticas	0,1 kg	05 cm ³
Temperos	0,2 kg	06 cm ³
Molhos de pimenta	0,3 kg	09 cm ³
Gelatinas	0,2 kg	10 cm ³
Leite de coco	0,5 kg	12 cm ³
Coco ralado	1,0 kg	07 cm ³

Provável Solução utilizando o Best-Fit Decreasing:

Passo 1. Vá pegando os itens na seqüência e alocando nas sacolas e avaliando o seu peso e volume;

Passo 2. Se um dos itens ultrapassar o peso, o volume da sacola, ou ambos verifique a possibilidade de inclusão, de maneira que a capacidade residual seja a menor possível, nas sacolas já existentes ou abra uma nova sacola

Passo 3. Repita esse processo enquanto houverem itens não alocados.

A solução proposta para esse problema está representada a seguir:

Tabela 6: Provável solução utilizando o algoritmo de BFD

Sacola	Itens	Peso	Volume
1	Açúcar	5	40
2	Café	3	40
3	Feijão, farinha de rosca e farinha de milho	5	40
4	Arroz e farinha de mandioca	5	35
5	Creme de leite e gelatina	0,9	40

6	Leite condensado e azeite	1,5	40
7	Maisena, óleo, extrato de tomate e molho pronto	1,7	40
8	Ervas aromáticas. Temperos e maionese	2,7	40
9	Ketchup, molho inglês e leite de coco	1,2	39
10	Farinha de trigo, fubá e sal	4,3	38,7
11	Mostarda, molho de pimenta e coco ralado	1,7	33

Atividade de Colaboração: leitura do texto sobre algoritmos construtivos para *bin packing* com demanda *on-line*. Os alunos poderão colaborar enviando sugestões e comentários através do AMEM. Bibliografia: *Modern Heuristic Search Methods* (Smith, 1996).

Aula 6: Algoritmos construtivos *on-line* para o *bin packing*

Objetivo: compreender que as características do problema podem influenciar no desenvolvimento de um bom algoritmo. Verificar porque o não conhecimento prévio de todos os itens pode alterar o desempenho e o desenvolvimento de algoritmos para *bin packing*.

Elementos da MDP: B1, B2. A escolha desses elementos baseia-se na observação da condução do processo de ensino-aprendizagem dos alunos, elemento crucial dessa aula.

Programação:

a)Atividade Um:

Duração: 40 minutos

Desafio Inicial: O sistema de Encomendas dos Correios de Jungleland possui uma esteira que divide automaticamente as encomendas em três grandes regiões da cidade. No entanto, há diversos carteiros para cada região, onde cada carteiro possui um limite máximo de peso que pode carregar. Como administrador dos Correios, desenvolva um algoritmo construtivo

para a distribuição das encomendas pelos carteiros, sabendo que a esteira informa o peso de cada encomenda e que as mesmas chegam ao sistema uma após a outra.

Peso máximo que o carteiro pode carregar: 30kg

Encomendas: variando entre 0,01kg a 5 kg.

b)Atividade Dois:

Duração: 110 minutos

Melhor solução educacional do momento: Sistematizar o processo com noções de algoritmos construtivos *on-line*: FF, BF, WF, NF.

Sistematizando com FF:

- Na medida que os itens vão chegando, deve-se entregá-los aos carteiros enquanto não ultrapassar a capacidade que eles podem carregar;
- Se ultrapassar a capacidade do carteiro que está sendo testado, testar com os outros carteiros já convocados;
- Entregue o item para o primeiro carteiro que puder carregá-lo;
- Se não houver nenhum carteiro apto para o serviço, convocar novo carteiro;
- Repetir esse processo enquanto houver itens chegando na esteira;

Sistematizando para BF:

- Na medida que os itens vão chegando, deve-se entregá-los aos carteiros enquanto não ultrapassar a capacidade que eles podem carregar;
- Se ultrapassar a capacidade do carteiro que está sendo testado, testar com os outros carteiros já convocados;
- Entregue a encomenda para o carteiro que estiver mais carregado, mas que não ultrapasse a capacidade dele;
- Se não houver nenhum carteiro apto para o serviço, convocar novo carteiro;
- Repetir esse processo enquanto houver itens chegando na esteira;

Sistematizando para NF:

- Na medida que os itens vão chegando, deve-se entregá-los aos carteiros enquanto não ultrapassar a capacidade que eles podem carregar;
- Se ultrapassar a capacidade do carteiro que está sendo testado, convoque novo carteiro para entregar a encomenda;
- Repetir esse processo enquanto houver itens chegando na esteira;

Sistematizando para WF:

- Na medida que os itens vão chegando, deve-se entregá-los aos carteiros enquanto não ultrapassar a capacidade que eles podem carregar;
- Se ultrapassar a capacidade do carteiro que está sendo testado, testar com os outros carteiros já convocados;
- Entregue a encomenda para o carteiro que estiver menos carregado, mas que não ultrapasse a capacidade dele;
- Se não houver nenhum carteiro apto para o serviço, convocar novo carteiro;
- Repetir esse processo enquanto houver itens chegando na esteira

Atividade extraclasse:

Desafio mais amplo: É possível melhorar as heurísticas apresentadas sabendo-se de antemão os itens que devem ser distribuídos? Caso haja melhora, quais são as alterações que devem ser feitas nos algoritmos para que trabalhe com todos os dados disponíveis?

Atividade de Colaboração: leitura do texto sobre algoritmos construtivos para *bin packing* sem demanda *on-line*. Os alunos poderão colaborar enviando sugestões e comentários através do AMEM. Bibliografia: Algoritmo Interativo para o Problema de Empacotamento Unidimensional (Frazzon, 2003).

Aula 7: Algoritmos construtivos para o bin packing

Objetivo: compreender que as características do problema podem influenciar no desenvolvimento de um bom algoritmo. Verificar porque o prévio conhecimento de todos os itens pode alterar o desempenho e o desenvolvimento de algoritmos para *bin packing*.

Elementos da MDP: B1, B2. A escolha desses elementos baseia-se na observação da condução do processo de ensino-aprendizagem dos alunos, elemento crucial dessa aula.

Programação:

a)Atividade Um:

Duração: 40 minutos

Desafio Inicial: Considerando o seguinte problema de *bin packing*, construa um algoritmo genérico que possa gerar boas soluções. Neste caso, considere que todos os dados do problema já são sabidos de antemão, logo esta informação pode ser crucial para a implementação do algoritmo.

Itens: 99, 43, 19, 7, 37, 94, 6, 79, 64, 32, 18, 50, 46, 3, 77, 25, 2, 5, 33, 67.

Capacidade: 99

b)Atividade Dois:

Duração: 120 minutos

Melhor solução educacional do momento: Apresentação dos algoritmos construtivos off-line BFD, NFD, FFD e WFD.

Itens ordenados: 99, 94, 79, 77, 67, 64, 50, 46, 43, 37, 33, 32, 25, 19, 18, 7, 6, 5, 3, 2.

Sistematizando com FFD:

- Ordene os itens em ordem não-crescente;
- Aloque itens no *bin* enquanto não ultrapassar a capacidade;
- Se ultrapassar a capacidade do *bin* que está sendo testado, testar nos outros *bins* já alocados;
- Aloque o item no primeiro *bin* que couber;
- Se não houver nenhum *bin* apto, alocar novo *bin*;
- Repetir esse processo enquanto houver itens para serem alocados;

Solução:

Bin 1: 99

Bin 2: 94, 6

Bin 3: 79, 19, 2

Bin 4: 77, 18, 5

Bin 5: 67, 33

Bin 6: 64, 32, 3

Bin 7: 50, 46

Bin 8: 43, 37, 7

Bin 9: 25

Sistematizando para BFD:

- Ordene os itens em ordem não-crescente;
- Aloque itens no *bin* enquanto não ultrapassar a capacidade;
- Se ultrapassar a capacidade do *bin* que está sendo testado, testar nos outros *bins* já alocados;
- Aloque o item no primeiro *bin* que couber de maneira que a capacidade residual do *bin* seja a menor possível;
- Se não houver nenhum *bin* apto, alocar novo *bin*;
- Repetir esse processo enquanto houver itens para serem alocados;

Solução:

Bin 1: 99

Bin 2: 94, 6

Bin 3: 79, 19

Bin 4: 77, 18, 5

Bin 5: 67, 33

Bin 6: 64, 32, 3

Bin 7: 50, 46, 2

Bin 8: 43, 37, 7

Bin 9: 25

Sistematizando para NFD:

- Ordene os itens em ordem não-crescente;
- Aloque itens no *bin* enquanto não ultrapassar a capacidade;
- Se ultrapassar a capacidade do *bin* que está sendo testado, alocar novo *bin*;
- Repetir esse processo enquanto houver itens para serem alocados;

Solução:

Bin 1: 99

Bin 2: 94

Bin 3: 79

Bin 4: 77

Bin 5: 67

Bin 6: 64

Bin 7: 50, 46

Bin 8: 43, 37

Bin 9: 33, 32, 25

Bin 10: 19, 18, 7, 6, 5, 3, 2

Sistematizando para WFD:

- Ordene os itens em ordem não-crescente;
- Aloque itens no *bin* enquanto não ultrapassar a capacidade;
- Se ultrapassar a capacidade do *bin* que está sendo testado, testar nos outros *bins* já alocados;
- Aloque o item no primeiro *bin* que couber de maneira que a capacidade residual seja a maior possível;
- Se não houver nenhum *bin* apto, alocar novo *bin*;
- Repetir esse processo enquanto houver itens para serem alocados;

Solução:

Bin 1: 99

Bin 2: 94

Bin 3: 79, 2

Bin 4: 77, 3

Bin 5: 67, 32

Bin 6: 64, 33

Bin 7: 50, 46

Bin 8: 43, 37

Bin 9: 25, 19, 18, 7, 6, 5

Atividade extraclasse:

Desafio mais Amplo: Implementação dos algoritmos construtivos propostos para o problema de *bin packing* unidimensional, com o objetivo de colocar em prática os

conhecimentos construídos em sala de aula e verificar o comportamento dos algoritmos frente as mais diferentes instâncias de um problema de *bin packing*.

Atividade de colaboração: criação de um fórum no ambiente AMEM para a discussão de dúvidas acerca das implementações dos diversos algoritmos construtivos para o *bin packing*. As atividades do fórum serão monitoradas e servirão como base para a discussão on-line que será realizada na próxima aula.

Aula 8: Discussão sobre os algoritmos implementados

Objetivo: através da sala de discussão do ambiente AMEM, realizar uma discussão técnica acerca dos problemas, soluções, métodos e resultados das implementações realizadas pelos educandos, de tal forma a perceber o grau de comprometimento dos alunos em relação à disciplina.

Elementos da MDP: B2, D2, D3. Esses elementos foram escolhidos com o intuito de observar a eficácia do processo de ensino-aprendizagem através da observação e da fomentação das discussões acerca do tema proposto.

Programação:

a) Atividade Um:

Duração: 150 minutos

Realização do encontro virtual.

Atividade de colaboração: leitura do texto sobre algoritmos de melhoramento para *bin packing*. Os alunos poderão colaborar enviando sugestões e comentários através do AMEM. Bibliografia: *New Ideas in Optimization* (Corne *et al*, 1999)

Aula 9: Algoritmos de melhoramento

Objetivo: compreender que a simples construção de uma solução não garante a qualidade da mesma. Testes e trocas podem ser feitos após a solução estar pronta, para que a mesma possivelmente melhore.

Elementos da MDP: B1, B2. A escolha desses elementos baseia-se na observação da condução do processo de ensino-aprendizagem dos alunos, elemento crucial dessa aula.

Programação:

a)Atividade Um:

Duração: 60 minutos

Desafio Inicial: Considerando a distribuição apresentada, que foi obtida através da aplicação do algoritmo NFD no problema abaixo. Os algoritmos de melhoramento tentam, a partir de uma solução inicial, modificar a mesma para que novas soluções sejam encontradas, potencialmente melhores. Construa um procedimento sistemático que modifique a solução abaixo, tentando diminuir o número de *bins*.

Itens: 99, 43, 19, 7, 37, 94, 6, 79, 64, 32, 18, 50, 46, 3, 77, 25, 2, 5, 33, 67.

Solução inicial usando NFD:

Capacidade: 100

Bin 1: 99

Bin 2: 94

Bin 3: 79

Bin 4: 77

Bin 5: 67

Bin 6: 64

Bin 7: 50, 46

Bin 8: 43, 37

Bin 9: 33, 32, 25

Bin 10: 19, 18, 7, 6, 5, 3, 2

b)Atividade Dois:

Duração: 60 minutos

Melhor solução educacional do momento:

-Método de reconstrução: encontre o *bin* menos carregado e tente distribuir seus itens na capacidade residual dos outros *bins* alocados, seguindo a lista de *bins*. Faça isso enquanto for possível:

-Ex: Menor *Bin* (10) 19, 18, 7, 6, 5, 3, 2

Bin 1: 99 – capacidade residual = 1

Bin 2: 94 – capacidade residual = 6 – aloque o item 6

Bin 3: 79 – capacidade residual = 21 – aloque o item 19

Bin 4: 77 – capacidade residual = 23 – aloque o item 18

Bin 5: 67 – capacidade residual = 33 – aloque o item 7, 5, 3 e 2

Bin 6: 64 – capacidade residual = 36

Bin 7: 50, 46 – capacidade residual = 4

Bin 8: 43, 37 – capacidade residual = 20

Bin 9: 33, 32, 25 – capacidade residual = 10

-Ex: Menor *Bin* (6) 64

Bin 1: 99 – capacidade residual = 1

Bin 2: 94, 6 – capacidade residual = 0

Bin 3: 79, 19 – capacidade residual = 2

Bin 4: 77, 18 – capacidade residual = 5

Bin 5: 67, 7, 5, 3, 2 – capacidade residual = 16

Bin 7: 50, 46 – capacidade residual = 4

Bin 8: 43, 37 – capacidade residual = 20

Bin 9: 33, 32, 25 – capacidade residual = 10

c)Atividade Três:

Duração: 50 minutos

Apresentação formal dos conceitos sobre algoritmos de melhoramento.

Atividade extraclasse:

Desafio mais Amplo: Considerando o algoritmo de melhoramento proposto, sistematize uma maneira de melhorar a busca pelos *bins* que deverão receber os itens do *bin* eliminado no processo.

Atividade de colaboração: leitura do texto sobre algoritmos de melhoramento para *bin packing*. Os alunos poderão colaborar enviando sugestões e comentários através do AMEM. Bibliografia: *New Ideas in Optimization* (Corne et al, 1999)

Aula 10: Algoritmos de melhoramento para bin packing

Objetivo: aprofundar o estudo sobre os algoritmos de melhoramento para *bin packing*.

Elementos da MDP: B1, B2. A escolha desses elementos baseia-se na observação da condução do processo de ensino-aprendizagem dos alunos, elemento crucial dessa aula.

Programação:

a)Atividade Um:

Duração: 40 minutos

A vizinhança de um espaço de soluções é caracterizada como as soluções que estão próximas da solução corrente no espaço de busca, considerando um operador específico. Um operador é caracterizado como uma forma de migrar de uma solução para outra. No exemplo da aula anterior, o operador se caracteriza pela troca zero entre os *bins*. A vizinhança se altera bruscamente se o operador for modificado. Uma forma de tentar encontrar novas soluções é realizar trocas de itens entre os *bins*, gerando *bins* mais balanceados ou eliminando completamente um *bin* (conhecido como troca zero, ou seja, um *bin* recebe itens sem devolver nenhum).

b)Atividade Dois:

Duração: 40 minutos

Desafio Inicial: O algoritmo de melhoramento apresentado na aula anterior não realiza uma busca em todo o espaço de soluções. Elabore um procedimento sistemático que possa realizar trocas entre *bins*, com o intuito de gerar novas soluções potencialmente melhores para o problema abaixo.

Itens: 9, 8, 7, 6, 5, 5, 4, 4, 3, 3, 2, 2, 2

Solução inicial usando NFD:

Capacidade: 16

Bin 1: 9

Bin 2: 8, 7

Bin 3: 6, 5, 5

Bin 4: 4, 4, 4, 3

Bin 5: 3, 2, 2, 2

c)Atividade Três:

Duração: 70 minutos

Melhor solução educacional do momento:

Algoritmo de Dupla Troca:

Passo 1. Ordene os *bins* do mais carregado para o menos carregado

Passo 2. Faça $BM = bin$ mais carregado que contenha capacidade residual maior que zero e $BC = bin$ menos carregado

Passo 3. Ordene os itens em BC , do maior para o menor. Se algum item i de BC couber na capacidade residual de BM , faça uma troca zero entre i , que deve sair de BC e ir para BM

Passo 4. Faça $BC=BC-1$. Se $BM = BC$, faça $BM=BM+1$ e vá para o passo 2. Caso contrário vá para o passo 3. Se $BM >$ número de *bins* vá para o passo 5.

Passo 5. Teste todas as trocas possíveis entre itens do *bin* mais carregado e com capacidade residual diferente de zero com o menos carregado. Se houver alguma troca possível (que não estoure a capacidade do *bin*), realize a troca e volte para o passo 1. Caso contrário, encerre. Se houver mais de uma troca possível, escolha uma aleatoriamente.

1. *Bins* ordenados

Bin 3: 6, 5, 5 – capacidade residual = 0

Bin 2: 8, 7 – capacidade residual = 1

Bin 4: 4, 4, 4, 3 – capacidade residual = 1

Bin 1: 9 – capacidade residual = 7

Bin 5: 3, 2, 2, 2 – capacidade residual = 7

2. $BM = 8,7$ e $BC = 3,2,2,2 \Rightarrow$ não há elementos em BC que podem entrar em BM

2. $BM = 4,4,4,3$ e $BC = 3,2,2,2 \Rightarrow$ não há elementos em BC que podem entrar em BM

2. $BM = 9$ e $BC = 3,2,2,2$

Bin 1: 9, 3, 2, 2

Nova configuração:

Bin 3: 6, 5, 5 – capacidade residual = 0

Bin 1: 9, 3, 2, 2 – capacidade residual = 0

Bin 2: 8, 7 – capacidade residual = 1

Bin 4: 4, 4, 4, 3 – capacidade residual = 1

Bin 5: 2 – capacidade residual = 14

4. Não há mais trocas possíveis entre BC e BM

5. BM=8,7 ou BM= 4,4,4,3

BC=2

Trocas possíveis: *bin(8,2) e bin(7) ou bin(7,2) e bin(8)*

Trocas possíveis: *bin(2,4,4,3) e bin(4) ou bin(4,4,4,2) e bin(3)*

Escolha aleatória: *bin(8,2) e bin(7)*

Nova configuração

Bin 3: 6, 5, 5 – capacidade residual = 0

Bin 1: 9, 3, 2, 2 – capacidade residual = 0

Bin 2: 8, 2 – capacidade residual = 6

Bin 4: 4, 4, 4, 3 – capacidade residual = 1

Bin 5: 7 – capacidade residual = 8

5. BM=4,4,4,3

BC=7

Não há trocas possíveis entre BM e BC que gerem soluções factíveis.

Atividade extraclasse:

Desafio mais Amplo: Implementação dos algoritmos de melhoramento propostos para o problema de *bin packing* unidimensional com o objetivo de verificar a modificação da solução frente a estes algoritmos, considerando como principais norteadores à qualidade da solução e o tempo gasto na obtenção desta.

Atividade de colaboração: criação de um fórum no ambiente AMEM para a discussão de dúvidas acerca das implementações dos diversos algoritmos de melhoramento para o *bin packing*. As atividades do fórum serão monitoradas e servirão como base para a discussão on-line que será realizada na próxima aula.

Aula 11: *Discussão sobre os algoritmos implementados*

Objetivo: através da sala de discussão do ambiente AMEM, realizar uma discussão técnica acerca dos problemas, soluções, métodos e resultados das implementações realizadas pelos

educandos, de tal forma a perceber o grau de comprometimento dos alunos em relação à disciplina.

Elementos da MDP: B2, D2, D3. Esses elementos foram escolhidos com o intuito de observar a eficácia do processo de ensino-aprendizagem através da observação e da fomentação das discussões acerca do tema proposto.

Programação:

a) Atividade Um:

Duração: 150 minutos

Realização do encontro virtual.

Atividade de colaboração: leitura do texto sobre algoritmos de melhoramento para *bin packing*. Os alunos poderão colaborar enviando sugestões e comentários através do AMEM. Bibliografia: Heurísticas e Metaheurísticas (Müller, 2002).

Aula 12: Metaheurísticas

Objetivo: compreender o que são metaheurísticas e mostrar como elas podem guiar outras heurísticas já exploradas, tentando sanar os problemas encontrados pelas heurísticas construtivas e de melhoramento.

Elementos da MDP: B1, B2. A escolha desses elementos baseia-se na observação da condução do processo de ensino-aprendizagem dos alunos, elemento crucial dessa aula.

Programação:

a) Atividade Um:

Duração: 50 minutos

Os algoritmos de melhoramento podem encontrar boas soluções, mas enfrentam alguns problemas quando encontram um ótimo local. Um ótimo local é definido como uma solução dentro do espaço de soluções, cuja vizinhança, considerando o operador definido, é formada por soluções piores que a solução do ótimo local. Neste caso, o algoritmo de melhoramento finaliza e não consegue explorar outros espaços de solução. Um outro problema pode ocorrer quando o algoritmo atinge um platô, que é definido como um espaço de soluções onde todas elas possuem o mesmo valor. Da mesma maneira, o algoritmo de melhoramento pode finalizar para esta solução.

b)Atividade Dois:

Duração: 50 minutos

Desafio Inicial: Elabore um procedimento sistemático que possa ser utilizado no problema dado para que o algoritmo não finalize em platôs ou ótimos locais, mantendo, contudo, a boa qualidade da solução.

Itens: 99, 43, 19, 7, 37, 94, 6, 79, 64, 32, 18, 50, 46, 3, 77, 25, 2, 5, 33, 67.

Capacidade: 100

c)Atividade Três

Duração: 50 minutos

Melhor solução educacional do momento:

Múltiplo Start:

Executar um procedimento de construção qualquer (NFD, por exemplo). Depois, aplicar uma heurística de melhoramento (reconstrução).

Após a heurística de melhoramento, modificar os dados iniciais do NFD e executar novamente o processo.

Ex:

NFD com os itens ordenados

Bin 1: 99

Bin 2: 94

Bin 3: 79

Bin 4: 77

Bin 5: 67

Bin 6: 64

Bin 7: 50, 46

Bin 8: 43, 37

Bin 9: 33, 32, 25

Bin 10: 19, 18, 7, 6, 5, 3, 2

Algoritmo de Melhoramento

Bin 1: 99

Bin 2: 94, 6

Bin 3: 79, 19

Bin 4: 77, 18

Bin 5: 67, 7, 5, 3, 2

Bin 6: 64

Bin 7: 50, 46

Bin 8: 43, 37

Bin 9: 33, 32, 25

NFD com os itens ordenados na ordem inversa

Bin 1: 2, 3, 5, 6, 7, 18, 19, 25

Bin 2: 32, 33

Bin 3: 37, 43

Bin 4: 46, 50

Bin 5: 64

Bin 6: 67

Bin 7: 77

Bin 8: 79

Bin 9: 94

Bin 10: 99

Algoritmo de Melhoramento

Bin 1: 2, 3, 5, 6, 7, 18, 19, 25 – capacidade residual = 15

Bin 2: 32, 33 = capacidade residual = 35

Bin 3: 37, 43 = capacidade residual = 20

Bin 4: 46, 50 = capacidade residual = 4

Bin 5: 64 = capacidade residual = 36

Bin 6: 67 = capacidade residual = 33

Bin 7: 77 = capacidade residual = 23

Bin 8: 79 = capacidade residual = 21

Bin 9: 94 = capacidade residual = 6

Bin 10: 99 = capacidade residual = 1

1 Maior capacidade residual (36)= *Bin 5*, mas não é possível redistribuir
2 Maior capacidade residual (35)= *Bin 2*, (32,33), onde é possível alocar o item 32 no *bin 5* e o 33 no *bin 6*.

Bin 1: 2, 3, 5, 6, 7, 18, 19, 25 – capacidade residual = 15

Bin 2: 37, 43 = capacidade residual = 20

Bin 3: 46, 50 = capacidade residual = 4

Bin 4: 64, 32 = capacidade residual = 4

Bin 5: 67, 33 = capacidade residual = 0

Bin 6: 77 = capacidade residual = 23

Bin 7: 79 = capacidade residual = 21

Bin 8: 94 = capacidade residual = 6

Bin 9: 99 = capacidade residual = 1

Atividade extraclasse

Desafio mais Amplo: Desenvolva um procedimento sistemático que altere o algoritmo de melhoramento de tal forma que ele possa sair dos ótimos locais, buscando novas vizinhanças potencialmente melhores.

Atividade de colaboração: leitura do texto sobre algoritmos genéticos para *bin packing*. Os alunos poderão colaborar enviando sugestões e comentários através do AMEM. Bibliografia: *A Hybrid Grouping Genetic Algorithm for bin packing* (Falkeneauer, 1996).

Aula 13: Algoritmos Genéticos

Objetivo: compreender os conceitos e métodos dos algoritmos genéticos, desde sua fundamentação teórica na biologia às questões de representação, operadores e parâmetros.

Elementos da MDP: B1, B2. A escolha desses elementos baseia-se na observação da condução do processo de ensino-aprendizagem dos alunos, elemento crucial dessa aula.

Programação:

a)Atividade Um:

Duração: 50 minutos

Algumas metaheurísticas são conhecidas como algoritmos populacionais, ou seja, trabalham com populações de indivíduos, cujas características podem ser avaliadas de tal forma a gerar uma solução melhor. O principal problema neste tipo de algoritmo está na forma de representação dos indivíduos, que influencia diretamente em como as características destes podem ser combinadas para gerar um indivíduo melhor. As características interessantes de cada solução são fortemente dependentes do problema, que deve ser analisado para que o máximo possível de informação seja representado num indivíduo de forma precisa e eficiente.

b)Atividade Dois:

Duração: 50 minutos

Desafio Inicial: Considerando o problema abaixo e as cinco soluções encontradas, elabore um procedimento sistemático que busque e combine as melhores características das soluções, criando indivíduos (soluções) potencialmente melhores que os anteriores. Desenvolva também uma forma de representação para cada indivíduo.

Itens: 99, 43, 19, 7, 37, 94, 6, 79, 64, 32, 18, 50, 46, 3, 77, 25, 2, 5, 33, 67.

Capacidade: 100

Primeiro Indivíduo: NFD

Bin 1: 99

Bin 2: 94

Bin 3: 79

Bin 4: 77

Bin 5: 67

Bin 6: 64

Bin 7: 50, 46

Bin 8: 43, 37

Bin 9: 33, 32, 25

Bin 10: 19, 18, 7, 6, 5, 3, 2

Segundo Indivíduo: Algoritmo de Melhoramento sobre o NFD

Bin 1: 99

Bin 2: 94, 6
Bin 3: 79, 19
Bin 4: 77, 18
Bin 5: 67, 7, 5, 3, 2
Bin 6: 64
Bin 7: 50, 46
Bin 8: 43, 37
Bin 9: 33, 32, 25

Terceiro Indivíduo: BFD

Bin 1: 99
Bin 2: 94, 6
Bin 3: 79, 19
Bin 4: 77, 18, 5
Bin 5: 67, 33
Bin 6: 64, 32, 3
Bin 7: 50, 46, 2
Bin 8: 43, 37, 7
Bin 9: 25

Quarto Indivíduo: FFD

Bin 1: 99
Bin 2: 94, 6
Bin 3: 79, 19, 2
Bin 4: 77, 18, 5
Bin 5: 67, 33
Bin 6: 64, 32, 3
Bin 7: 50, 46
Bin 8: 43, 37, 7
Bin 9: 25

Quinto Indivíduo: WFD

Bin 1: 99
Bin 2: 94
Bin 3: 79, 2
Bin 4: 77, 3
Bin 5: 67, 32
Bin 6: 64, 33
Bin 7: 50, 46
Bin 8: 43, 37
Bin 9: 25, 19, 18, 7, 6, 5

c)Atividade Três:

Duração: 50 minutos

Melhor Solução educacional do momento:

Possível Solução Utilizando Algoritmos Genéticos:

Método de Representação: seqüência de números, onde cada número representa o *bin* onde está alocado numa lista previamente ordenada.

99, 94, 79, 77, 67, 64, 50, 46, 43, 37, 33, 32, 25, 19, 18, 7, 6, 5, 3, 2

Cromossomo do Indivíduo 1 - 1, 2, 3, 4, 5, 6, 7, 7, 8, 8, 9, 9, 9, 10, 10, 10, 10, 10, 10, 10

Cromossomo do Indivíduo 2 – 1, 2, 3, 4, 5, 6, 7, 7, 8, 8, 9, 9, 9, 3, 4, 5, 2, 5, 5, 5

Cromossomo do Indivíduo 3 – 1, 2, 3, 4, 5, 6, 7, 7, 8, 8, 5, 6, 9, 3, 4, 8, 2, 4, 6, 7

Cromossomo do Indivíduo 4 – 1, 2, 3, 4, 5, 6, 7, 7, 8, 8, 5, 6, 9, 3, 4, 8, 2, 4, 6, 3

Cromossomo do Indivíduo 5 – 1, 2, 3, 4, 5, 6, 7, 7, 8, 8, 6, 5, 9, 9, 9, 9, 9, 9, 4, 3

Método de Seleção: indivíduos com menor número de *bins*. Critério de desempate, maior soma de capacidade residual.

I1 = 10 bins

I2 = 9 bins, soma de capacidade residual = 94

I3 = 9 bins, soma de capacidade residual = 94

I4 = 9 bins, soma de capacidade residual = 194

IS = 9 bins, soma de capacidade residual = 94

Escolhidos: I4 e I5 (sorteio)

Recombinação de Um ponto. Sorteio aleatório: posição 13

Cromossomo do Indivíduo 4 – 1, 2, 3, 4, 5, 6, 7, 7, 8, 8, 5, 6, 9, **3, 4, 8, 2, 4, 6, 3**

Cromossomo do Indivíduo 5 – **1, 2, 3, 4, 5, 6, 7, 7, 8, 8, 6, 5, 9, 9, 9, 9, 9, 4, 3**

Filho 1 - 1, 2, 3, 4, 5, 6, 7, 7, 8, 8, 5, 6, 9, 9, 9, 9, 9, 4, 3

Filho 2 - 1, 2, 3, 4, 5, 6, 7, 7, 8, 8, 6, 5, 9, 3, 4, 8, 2, 4, 6, 3

Avaliação do Filho 1 -

Bin 1 – 99

Bin 2 – 94

Bin 3 – 79, 2

Bin 4 – 77, 3

Bin 5 – 67, 33

Bin 6 – 64, 32

Bin 7 – 50, 46

Bin 8 – 43, 37

Bin 9 – 25, 19, 18, 7, 6, 5

Solução possível! (não viola a capacidade)

Avaliação do Filho 2 -

Bin 1 – 99

Bin 2 – 94, 6

Bin 3 – 79, 19, 2

Bin 4 – 77, 18, 5

Bin 5 – 67, 32

Bin 6 – 64, 33, 3

Bin 7 – 50, 46

Bin 8 – 43, 37, 7

Bin 9 – 25

Solução possível! (não viola a capacidade)

Mutação: troca aleatória de um item para outro bin

Filho 1 - 1, 2, 3, 4, 5, 6, 7, 7, 8, 8, 5, 6, 9, 9, 9, 9, 9, 9, 4, 3

Filho 1' - 1, 2, 3, 4, 5, 6, 7, 7, 8, 8, 5, 6, 9, 9, 9, 9, 9, 8, 4, 3

Solução possível! (não viola a capacidade)

Atividade extraclasse:

Desafio mais Amplo: os algoritmos genéticos nem sempre geram soluções possíveis. Elabore um procedimento sistemático que reconstrua uma solução infactível (que, no caso é considerada infactíveis as soluções que estourem o limite da capacidade) a partir de um filho gerado pelo algoritmo genético. Utilize como bibliografia o algoritmo descrito em Falkenauer(1996).

Atividade de colaboração: leitura do texto sobre algoritmos genéticos para *bin packing*. Os alunos poderão colaborar enviando sugestões e comentários através do AMEM. Bibliografia: *A Hybrid Grouping Genetic Algorithm for bin packing* (Falkeneauer, 1996).

Aula 14: Metaheurísticas para bin packing

Objetivo: consolidação do conhecimento sobre algoritmos genéticos.

Elementos da MDP: B1, B2. A escolha desses elementos baseia-se na observação da condução do processo de ensino-aprendizagem dos alunos, elemento crucial dessa aula.

Programação:

a) Atividade Um:

Duração: 150 minutos

Discussão do desafio mais amplo apresentado na aula anterior e discussão acerca da implementação dos algoritmos genéticos para o *bin packing*, com o objetivo dos alunos implementarem uma metaheurística que possa encontrar soluções factíveis e melhores para o problema.

Aula 15: Fecho da Disciplina

Objetivo: discutir o andamento da disciplina, a forma metodológica utilizada no ensino e o aproveitamento dos conceitos discutidos durante todo o semestre.

Elementos da MDP: B3. Esse elemento foi escolhido com o objetivo de discutir com os alunos as vantagens e desvantagens da metodologia proposta, fomentando discussões que contribuam para os próximos semestres onde a disciplina será ministrada.

Programação:

a)Atividade Um:

Duração: 150 minutos

Discussão presencial sobre os temas definidos no objetivo da aula

Reunião final dos professores

Objetivos: responder e analisar as questões da MDP que só podem ser discutidas após o término da disciplina e realizar a avaliação, que é definida como a entrega dos trabalhos propostos em aula e a participação dos alunos nas atividades desenvolvidas, considerando tanto as atividades extra-classe como as de colaboração.

Elementos da MDP: C2, A4, D4. Esses elementos foram escolhidos para a realização de uma análise crítica do desempenho docente e discente durante a realização da disciplina.

Conclusão

Conforme foi descrito no decorrer deste trabalho, propôs-se uma metodologia para o ensino de heurísticas e metaheurísticas para o problema de *bin packing*.

Essa metodologia utiliza a problematização do conhecimento, técnica presente na Educação Dialógica-Problematizadora, e os conceitos divulgados no processo de Investigação-Ação Educacional.

Ela ainda faz uso dos três momentos pedagógicos, descritos no capítulo 1 e a matriz dialógico-problematizadora como norteadora para a elaboração das questões propostas.

O público alvo para o qual foi desenvolvida, são os alunos de graduação em Ciência da computação, visto que o ensino de heurísticas e metaheurísticas, por envolver muitos conceitos matemáticos pode tornar-se enfadonho aos alunos, e um dos objetivos é tentar, através da problematização do conhecimento, manter a atenção do aluno e também conseguir acompanhar com mais facilidade o seu processo de aprendizagem.

Acredita-se que a nova metodologia educacional utilizada para a preparação deste curso traz inúmeros benefícios para o educando e o educador. Visto que se por um lado, o professor estabelece de forma mais clara seus objetivos e metas, tendendo assim a uma melhoria na condução da turma, os alunos podem tirar proveito da prévia definição das aulas para uma melhor interação com o professor e, principalmente, com o conteúdo proposto. A problematização traz inúmeros ganhos a sala de aula, onde se pode enfatizar um melhor acompanhamento dos alunos, que se sentem mais motivados e interessados em conteúdos que podem ser aplicados em problemas reais do cotidiano. Além disso, o fazer se torna muito mais intuitivo, levando a reflexões acerca do que está sendo discutido de forma mais aprofundada, pois os alunos adquirem “experiência” com o conteúdo proposto. Desta forma, a disciplina se torna mais interessante e os alunos aparentam uma maior motivação pela metodologia educacional proposta.

Bibliografia

ANGOTTI, J. A. P. e DELIZOICOV, D. N. **Metodologia do Ensino de Ciências**. São Paulo: Cortez, 1990.

ANGULO, J.F. **Investigación-acción y curriculum: una nueva perspectiva em la investigación educativa**. *Investigación em la Escuel*. Sevilla, nº 11, p. 39-49. 1990.

ARENALES, M.N. (1993). **Uma teoria para o problema de corte**. Tese de Livre Docência, ICMSC-USP, São Carlos, S.P., Brasil.

AZOLIN, B.R. **Estudo e Implementação das Interfaces Homem-Máquina de um ambiente multimídia de educação mediada por computador**. 74 p. Monografia de final do curso de Ciência de Computação – Universidade Federal de Santa Maria, Santa Maria 2002.

BERNARDI, R.D. **Aplicando a técnica de times assíncronos na otimização de problemas de empacotamento unidimensional**. Dissertação (Mestrado em Engenharia), Escola Politécnica da Universidade de São Paulo, 2001.

BOWEN, R. **Graphic Approaches to Describing Action Research Methodology**. *Educational Action Research*, Volume 6, Nº 3, 1998.

CARR, W. e KEMMIS, S. **Becoming Critical: Education, Knowledge and Action Research**. The Falmer Press, London, 1986.

COELHO, L. S. **Nova Abordagem de Algoritmo Memético Aplicado a Problemas de Otimização Não-Linear**. In: V SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, Canela, Rio Grande do Sul, 2001.

COGO, A.P. e FURTADO, J.C., **Otimização Do Problema De Corte Unidimensional Na Indústria Usando Algoritmos Genéticos**. Trabalho Final de Graduação do Curso Superior de Sistemas de Informação do Centro Universitário Franciscano, Santa Maria, 2001.

CORNE, D., DORIGO M. e GLOVER, F. **New Ideas in Optimization**. McGraw Hill, Berkshire, 1999.

DÁVALOS, R. V. **Uma Abordagem do Ensino de Pesquisa Operacional Baseada no Uso de Recursos Computacionais**. Universidade do Sul de Santa Catarina, Palhoça, SC.

DE BASTOS, F.P. e MÜLLER, F.M. **Criando Desafios em Informática**. In: Atas da IV Escola de Verão sobre Investigação-Ação Educacional. UFSM, Santa Maria, RS, 1999.

DE BASTOS, F.P. e MÜLLER, F.M. **Matriz Dialógico-Problematizadora como Ferramenta Organizadora do Trabalho Escolar no AMEM.** Florianópolis, SC, 2004.

DÍAZ, A.; GLOVER, F.; GHAZIRI, H.M.; GONZALEZ, J.L.; LAGUNA, M.; MOSCATO, P.; TSENG, F.T. **Optimización Heurística y Redes Neuronales: en Dirección de Operaciones e Ingeniería.** Madrid : Editorial Paraninfo, 1996.

ELLIOTT, J. **What is Action-Research in Schools?** Journal of Curriculum Studies, v. 10, n.4: 355-57, 1978.

FALKENAUER, E. **A Hybrid Grouping Genetic Algorithm for bin packing,** Journal of Heuristics v. 2, p. 5-30, 1996.

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.

FRAZZON, T. A. B., MÜLLER, F. M. e MEDEIROS, A. F. **Algoritmo interativo para o problema de empacotamento.** XXIII Enegep, Ouro Preto, 2003.

FRAZZON, T. A. B. **Algoritmo Interativo para o Problema de Empacotamento Unidimensional.** 91f. Dissertação de Mestrado em Engenharia de Produção – Universidade Federal de Santa Maria, Santa Maria, 2003.

FREIRE, P. **Pedagogia do Oprimido.** Rio de Janeiro: Paz e Terra, 1983.

GARCIA, I.C., REZENDE, P.J. e CALHEIROS, F.C. **Astral: Um Ambiente para Ensino de Estruturas de Dados através de Animações de Algoritmos.** Congresso Iberoamericano de Educação Superior em Computação, Anais. México, 1996.

HEARN, D. e BAKER, M. P. **Computer Graphics – C version.** New Jersey. Prentice Hall. 1997.

KEMMIS, S. e MCTAGGART, R. **Cómo Planificar La Investigación-acción.** Barcelona: Editorial Laerts,1988

LAUERMANN, R. A. C. **Ambiente Multimídia para Educação Mediada por Computador na Perspectiva da Investigação-Ação: Avaliação e Tutorial.** 186f. Dissertação de Mestrado em Engenharia de Produção – Universidade Federal de Santa Maria, Santa Maria, 2002.

- LAUERMANN, R. A. C., MÜLLER, F. M., DE BASTOS, F. P. e FERNÁNDEZ, E. G., **Uma experiência de ensino em Engenharia de Produção com o apoio do AMEM**, Enegep, Ouro Preto, 2003.
- LUNA, H. P. L. e GOLDBARG, M. C. **Otimização combinatória e programação linear**, Campus, Rio de Janeiro: 2000.
- MARTELLO, S. e TOTH, P. **Knapsack Problems: Algorithms and Computer Implementations**, Wiley, Chinchester, 1990.
- MARTINS, A.J. FIOLHAIS, C e PAIVA, J. **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 de 2003.
- MENDES, A.; MÜLLER, F.M.; FRANÇA, P. e MOSCATO, P. **Comparing meta-heuristic approaches for parallel machine scheduling problems**. Production Planning & Control, v.13, n.2, p.143-154, 2002.
- MIQUELIN, A.F. e DE BASTOS, F.P. **Investigação-Ação Escolar e Meios Tecnológico-Comunicativos: Possíveis Potencialidades**. Dissertação em Mestrado em Educação – Universidade Federal de Santa Maria, Santa Maria, 2002.
- MÜLLER, F. M. et al. **AMEM - Ambiente Multimídia para Educação Mediada por Computador na Perspectiva da Investigação-Ação Educacional**. Projeto aprovado edital 06/2000 - FAPERGS.Santa Maria, UFSM, 2000.
- MÜLLER, F.M. **Problemas de Sequenciamento em Processadores Paralelos: Modelos e Algoritmos**. XXV Congresso Nacional de Matemática Aplicada e Computacional, Nova Friburgo, 2002.
- SCHOLL, A.; KLEIN, R. e JÜGENS, C. **BISON: A Fast Hybrid Procedure For Exactly Solving the One-Dimensional Bin Packing Problem**, Computers Ops Res. n. 7, v. 24, p. 627-645, 1997.
- SMITH, V.J.R., OSMAN, I.H., REEVES e C.R., SMITH, G.D. **Modern Heuristic Search Methods**. John Wiley & Sons. New York , 1996.
- TANEV, I., UOZUMI, T. e ONO, K.. **Parallel Genetic Programming: Component Object-based Distributed Collaborative Approach**. In: 15th International Conference on Information Networking, Beppu City, 2001.
- YARGER, R. J. et al. **MySQL & mSQL**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2000.

ZACHARY, J.L. **Introduction to Scientific Programming
Computational Problem Solving Using: Maple and C.** University of Utah. Animations
available from [http://www.cs.utah.edu/~zachary/isp/applets/Cannon.html](http://www.cs.utah.edu/~zachary/isp/applets/Cannon/Cannon.html) e
<http://www.cs.utah.edu/~zachary/isp/applets/SlidingBlock/SlidingBlock.html>, 2004.