

CORDENONSI, A. Z., MÜLLER, Felipe Martins, BASTOS, Fábio da Purificação de. Lobo: Um Objeto de Aprendizagem para o Ensino de Heurísticas e Metaheurísticas. In: Congreso de Educación Superior de Computación – Jornadas Chilenas de Computación, Punta Arenas, Chile. 2008.

Lobo: Um Objeto de Aprendizagem para o Ensino de Heurísticas e Metaheurísticas

Andre Zanki Cordenonsi – Universidade Federal de Santa Maria – andrezc@inf.ufsm.br

Felipe Martins Müller – Universidade Federal de Santa Maria – felipe@inf.ufsm.br

Fábio da Purificação de Bastos – Universidade Federal de Santa Maria – fbastos@ce.ufsm.br

Av. Roraima n. 1000, Cidade Universitária, Bairro Camobi, 97050.210 – Santa Maira – RS - Brasil

Resumo: *O ensino de heurísticas e metaheurísticas – algoritmos utilizados amplamente na resolução de problemas complexos nas mais diversas áreas do conhecimento – é disperso na estrutura educacional superior brasileira em disciplinas da graduação e pós-graduação. O objeto de aprendizagem LOBO – (Learning Object Based on Optimization), busca uma forma inovadora de discutir os algoritmos heurísticos e metaheurísticos em torno de um problema comum, através de práticas baseadas no diálogo e instigando os alunos a buscar as respostas para os problemas que são apresentados aos mesmos. Através de sua interface gráfica, que permite a manipulação de problemas de dimensões mais elevadas, o objeto se mostrou útil na demonstração dos algoritmos mais complexos, incentivando os alunos a buscar soluções mais refinadas através da visualização da sua performance.*

Palavras-chave: Objetos de Aprendizagem, Ensino de Heurísticas e Metaheurísticas, Investigação-Ação Educacional

The LOBO object learnign as a Tool in the dialogical and investigative teaching of heursticas and metaheuristics

Abstract: *The education of heuristics and metaheuristics - algorithms used widely in the resolution of complex problems in the most diverse areas of the knowledge - is dispersed in the Brazilian higher educational structure in disciplines of the graduation and posgraduation. The approach of the learning object LOBO – (Learning Object Based on Optimization) produce an innovative way to the heuristics and metaheuristics educational, based in a common problem, through a practical focused in the dialogue and instigating the students to seek their owner answers. Through the graphical interface to manipulate problems with big dimensions, the LOBO can be very useful in the most complex algorithms demonstrations, stimulating the students to understand the refined solutions through the visualization of its performance.*

Keywords: *Learning Objects, Heuristics and Metaheuristics Teaching, Educational Action-Research*

1. Introdução

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

Dentro deste contexto, no âmbito deste artigo, será destacada a prática educacional orientada pelas teorias da Investigação-Ação Educacional (IAE) e a Prática Educacional Dialógica-Problematizadora (PEDP) (CARR, KEMMIS, 1986). Esta metodologia pode ser aplicada em todas as áreas do conhecimento e em todos os graus de ensino, desde os primeiros anos da educação básica até o ensino universitário. Como ferramental, é apresentado neste artigo o LOBO (*Learning Object Based on Optimization*), que permite que sejam

utilizados exemplos de maiores dimensões aos alunos, em uma interface gráfica de simulação. Através da utilização do objeto em sala de aula e disponibilizando o mesmo para os alunos trabalharem fora do ambiente escolar, espera-se que a compreensão sobre as diferenças entre os algoritmos mais simples e mais complexos, bem com os pontos fortes e pontos fracos de cada um, se tornem mais consolidados.

2. Investigação-Ação e o Diálogo Problematicador

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

O professor se transforma em um investigador de seus próprios métodos e agora, além do conhecimento, que já não mais lhe pertence de forma sozinha e autocrática, o próprio fazer educacional também não é mais algo intocável. A partir do seu próprio escrutínio e das observações que podem ser coletadas por seus pares e alunos, o professor pode tomar consciência da metodologia que ele *realmente* pratica em aula, e não mais aquela que ele *imagina* estar realizando.

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

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

Em (ANGOTTI, DELIZOICOV, 1990), a espiral reflexiva da investigação-ação foi desenvolvida através do aporte metodológico dos três momentos pedagógicos, implementado pelos autores no contexto do ensino de ciências naturais. O primeiro momento é o *Problematização Inicial*, apresentado, normalmente, como uma questão e/ou situação que deve ser discutida e debatida pelos alunos. A problematização inicial deve problematizar, de forma dialógica, o tema a ser tratado durante a atividade educativa. Durante uma aula, este problema é colocado para os alunos na forma de uma questão, por exemplo, onde os mesmos podem expor seu conhecimento prévio sobre o assunto, discutindo suas implicações. Este momento é primordial para que ocorra o clima dialógico, comunicativo e participativo, onde as idéias podem ser livremente debatidas – considerando que o professor, mesmo assim, ainda é o foco dirigente do processo e não pode se abster desta responsabilidade.

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

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

Em (DE BASTOS e MULLER, 1999), os três momentos pedagógicos foram reorganizados em: *Desafio Inicial*, *Melhor Solução Educacional no Momento* e *Desafio Mais Amplo*. Para os autores, o *desafio inicial* deveria envolver uma atividade a realizar e não uma questão, transformando o momento em uma efetiva resolução de um problema, diferenciando, assim, da *questão* apresentada anteriormente. A *melhor*

solução educacional no momento está relacionada à resolução do problema e o *desafio mais amplo* amplia o horizonte da questão específica, trabalhando sob o conhecimento geral da disciplina.

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

3. O Objeto de Aprendizagem LOBO

O objeto de aprendizagem LOBO (*Learning Object Based on Optimization*) foi desenvolvido com o objetivo de propiciar aos professores e alunos uma base comum para as experimentações necessárias à condução das atividades didáticas que envolvem o ensino na área de heurísticas e metaheurísticas. O LOBO pode ser utilizado como um *objeto de aprendizagem* para a condução de investigações iniciais que explorem a área de heurísticas e metaheurísticas para problemas combinatórios, bem como um *framework* de desenvolvimento de algoritmos. Desta forma, foram definidos dois modos de utilização do objeto: o modo tutoriado e o modo *expert*.

O modo tutoriado apresenta uma série de telas informativas ao usuário, fornecendo dicas e informações acerca da implementação do LOBO, bem como explicações mais detalhadas dos algoritmos implementados e a teoria do desenvolvimento de heurísticas e metaheurísticas. O modo *expert* não apresenta estas informações, transformando o LOBO, deste modo, em uma ferramenta de experimentação pura, onde os resultados são interpretados pelos próprios usuários.

A interface completa do objeto LOBO é formada por quatro áreas distintas, representadas na figura 1: (a)menu principal, (b)botões de ação, (c)histórico dos resultados e (d)área de simulação. Como é possível notar, o quarto botão se encontra pressionado. Este botão representa os modos do objeto. Na posição padrão, o modo tutoriado está ativo e o botão permanece pressionado. Ao clicar sobre o mesmo, o modo *expert* é ativado e o botão apresentará o estado desativado. Os demais botões representam, na ordem, as seguintes ações: execução da heurística, execução de um passo da heurística e exibição da rota ótima, caso ela existir. Estas ações serão explicadas detalhadamente nas próximas páginas. A primeira área do objeto apresenta o menu principal do LOBO. Este menu é dividido em quatro sub-menus: *O Problema*, *Algoritmos*, *Dados de teste* e *Sobre*.

O sub-menu *O Problema* apresenta duas opções: história e definição do PCV. A opção *História* apresenta um breve apanhado sobre a história e desenvolvimento do Problema do Caixeiro Viajante, bem como dos problemas que o precederam no seu desenvolvimento, conhecidos como o Problema de Euler e o Problema de Hamilton. Maiores detalhes sobre estes problemas clássicos no desenvolvimento da otimização combinatória podem ser encontrados em (REI, 1994) e (LAWLER *et al.*, 1985).

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

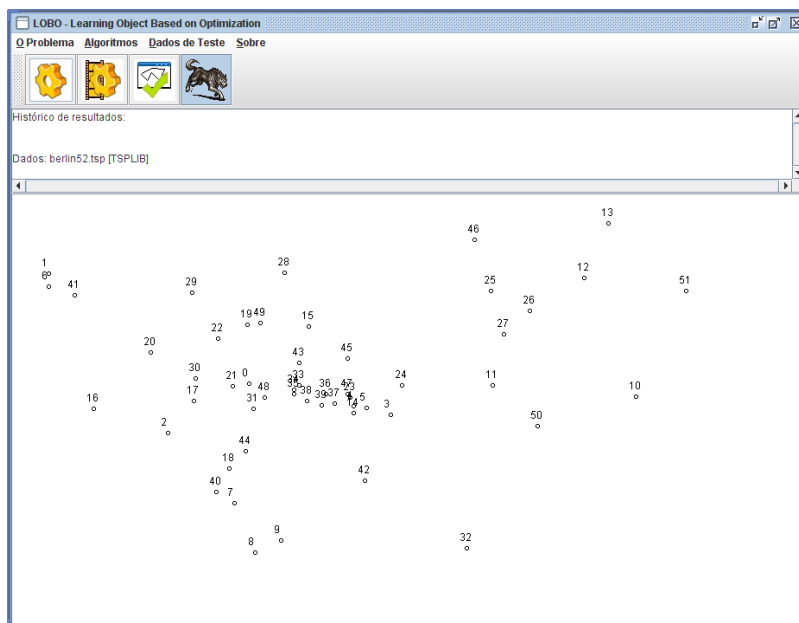


Figura 1 - Interface principal do LOBO

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

Finalmente, o sub-menu *Sobre* apresenta somente duas opções. A primeira, *Ajuda*, apresenta uma ajuda rápida ao sistema, implementada no formato de páginas HTML embutidas ao próprio objeto. A segunda opção, *Sobre...*, simplesmente apresenta uma tela contendo os principais dados do projeto LOBO e seus contribuidores.

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

O botão *executar a heurística* apresenta o resultado final da aplicação da heurística selecionada no sub-menu *Algoritmos* sobre o problema escolhido através do sub-menu *Dados de teste*. Como em todo o objeto, há dois modos de execução de uma heurística: o modo tutoriado e o *expert*. No modo tutoriado, ao pressionar o botão em questão, uma tela explicativa é apresentada ao usuário, avisando o mesmo que o LOBO está pronto para executar o algoritmo escolhido e a forma de apresentação dos dados após a finalização do mesmo. Assim que o usuário pressionar o botão *OK* para fechar a tela, o objeto de aprendizagem executa o algoritmo sobre os dados selecionados. Após a realização dos cálculos, uma janela de resultados é apresentada ao usuário (figura 2).

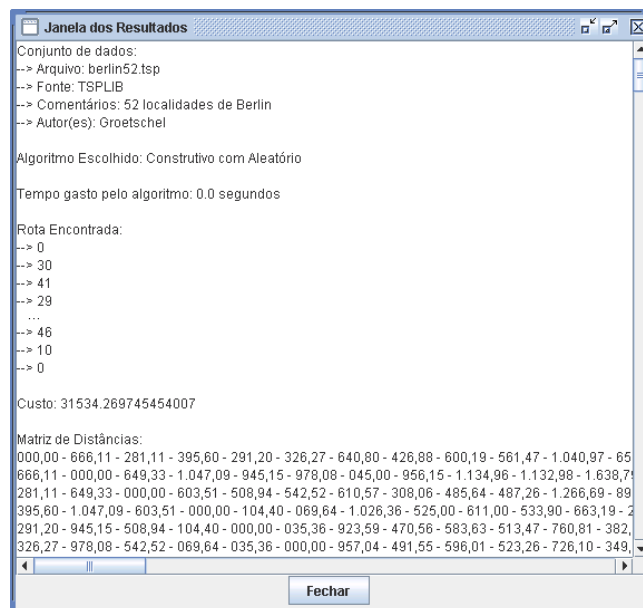


Figura 2 – Janela dos Resultados para o Algoritmo Construtivo Aleatório sobre o Problema *berlin52.tsp*

Nesta janela, que pode permanecer aberta indefinidamente, o usuário tem acesso às informações acerca de qual conjunto de dados foi escolhido, a heurística que foi executada, o tempo gasto pelo algoritmo, a rota encontrada, o custo da solução encontrada e a matriz de distâncias para o problema selecionado. Estes dados, em formato ASCII, podem ser copiados para qualquer tipo de processador de texto e compilados mais tarde na forma de tabelas e/ou gráficos. Para os algoritmos construtivos e os algoritmos de melhoramento, o funcionamento é idêntico. No entanto, a execução das metaheurísticas busca tabu e *simulated annealing* apresenta características próprias que podem ser visualizadas através de outras ferramentas de análise. Para o objeto LOBO, foi definido que estas metaheurísticas gerariam gráficos explicativos que apresentassem a performance do algoritmo no decorrer do tempo. Desta forma, além da janela de resultados, o objeto de aprendizagem apresenta uma segunda janela, contendo os gráficos da execução. Para a metaheurística *simulated annealing*, o gráfico apresentado representa o custo da solução \times temperatura atual, como apresentado na figura 3.

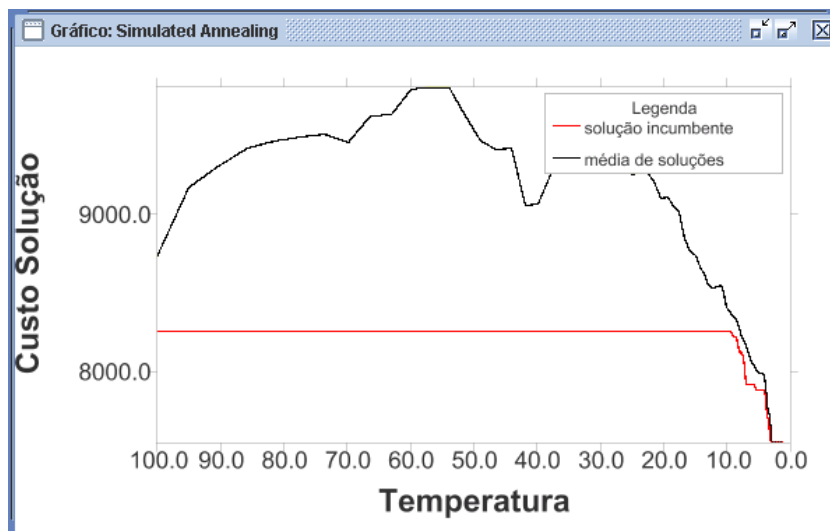


Figura 3 – Gráfico de uma Aplicação da Metaheurística *Simulated Annealing*

Para a metaheurística busca tabu, somente uma linha de plotagem é visualizada no gráfico, que representa o *custo da solução* em relação à iteração atual. O número de iterações é definido implicitamente pelo usuário através do parâmetro tempo e pela dimensão do problema. Quanto maior for o número de cidades, o algoritmo necessita um maior tempo para executar uma rodada da Busca Tabu. Quanto maior for o tempo, mais iterações podem ser executadas. Os parâmetros utilizados para o gráfico da figura 4, que representa o resultado da aplicação da busca tabu sobre o problema *pr76.tsp*, são: 40 segundos de processamento, 60 iterações na lista tabu, lista tabu com um tamanho máximo de 25 elementos e sem esquema de vizinhança para a busca local 2-Opt. O ponto onde a linha de plotagem atinge o seu menor valor é a solução incumbente que é apresentada ao usuário. É importante salientar, que na parte final da execução da metaheurística, ocorreu uma ciclagem da busca tabu, ou seja, as soluções estão em uma seqüência de maneira que a solução vizinha da última solução do ciclo seja a primeira solução do mesmo ciclo. Isso indica ao usuário que o tamanho da lista tabu é muito pequeno ou o número de iterações é muito baixo.

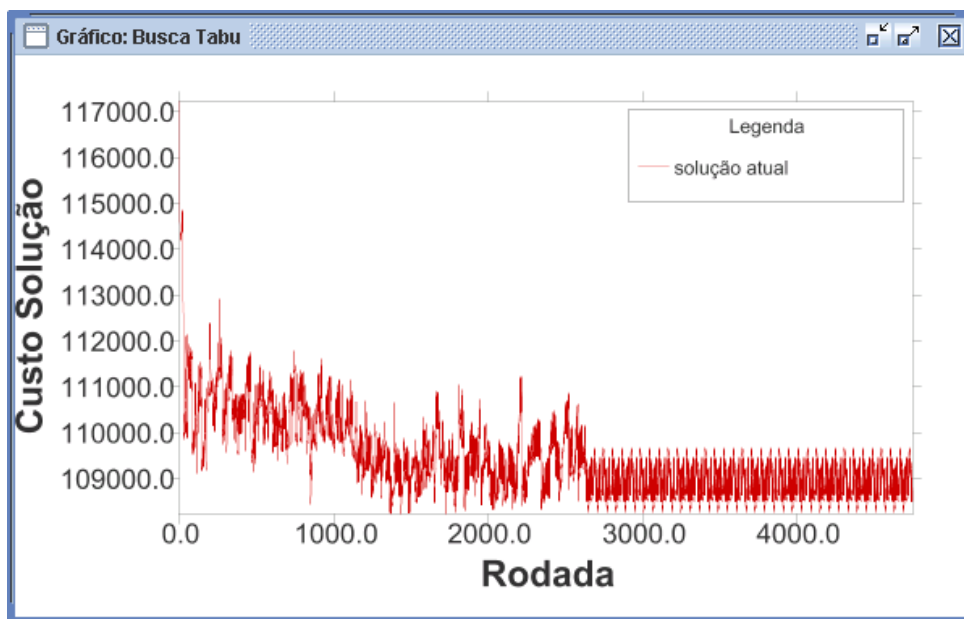


Figura 4 - Gráfico da metaheurística busca tabu aplicado sobre o problema *berlin52.tsp*, após a execução construtiva aleatória e da aplicação da heurística de melhoramento 2-Opt

O botão de simulação passo-a-passo está disponível para as seguintes heurísticas: vizinho mais próximo, inserção mais distante, 2-Opt e 3-Opt. A heurística construtiva aleatória, por seu caráter não previsível e sua simplicidade, não foi incluída na ação de simulação passo-a-passo. Em relação as metaheurísticas, sua complexidade impede que um passo isolado seja visualizado sem a perda do contexto.

Cada heurística apresenta particularidades na sua exibição passo-a-passo. O algoritmo do vizinho mais próximo é simulado através da inserção de arcos ligando as cidades a cada novo passo. A observação da execução deste algoritmo é extremamente importante para que o aluno perceba como o mesmo tende a piorar a solução final ao diminuir o número de cidades disponíveis para a inserção. Graficamente, é muito mais fácil perceber o problema desta heurística, pois os arcos que são inseridos ao final da execução se tornam maiores e os cruzamentos ocorrem com mais freqüência. A figura 5 apresenta seis momentos da aplicação da heurística do vizinho mais próximo sobre o problema *berlin52.tsp*. É fácil perceber que, no início, o algoritmo se comporta bem (a,b), mas rapidamente os arcos se tornam cada vez maiores (c,d) até que os cruzamentos se tornam impossíveis de evitar (e). No final(f), o resultado da solução encontrada apresenta arcos muito ruins: (10,28) e (6,1). Enquanto o algoritmo está construindo a solução, os arcos inseridos são representados pela cor vermelha, para dar destaque à simulação passo-a-passo. Assim que o último arco for inserido no problema, este passa à cor azul, demonstrando o final da simulação (figura 5, f).

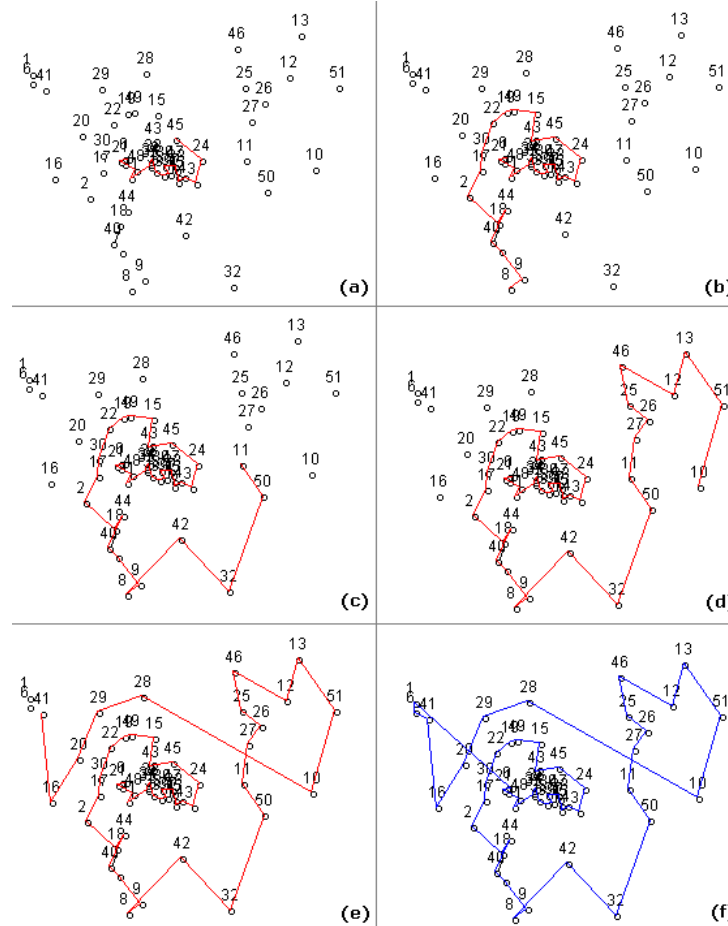


Figura 5 - Simulação passo-a-passo da heurística vizinho mais próximo sobre o problema berlin52.tsp

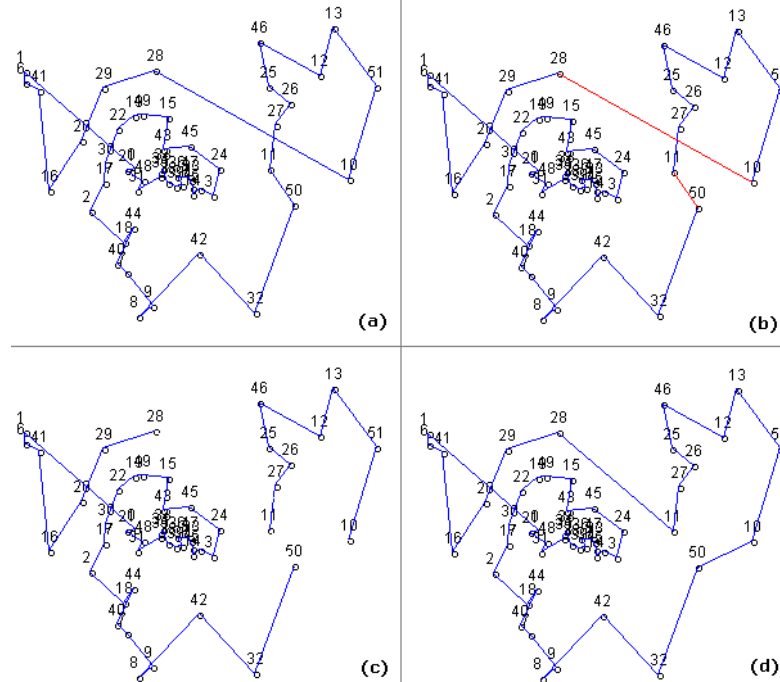


Figura 6 – Simulação de uma rodada 2-opt passo-a-passo

Em relação as heurísticas de melhoramento 2-Opt e 3-Opt, se optou por realizar uma animação dinâmica com a função de exibir para o usuário de forma didática como o algoritmo realiza as trocas de arcos. Ao clicar pela primeira vez no botão *Simulação passo-a-passo*, o LOBO faz duas perguntas ao usuário, relativas ao tamanho da vizinhança e a forma de busca pela melhor rota. Após responder de forma adequada aos parâmetros necessários, a simulação é realizada. O algoritmo 2-Opt escolhe os dois arcos que devem ser retirados e estes são apresentados graficamente através do desenho de linhas em cor vermelha(figura 6,a). A seguir, estes arcos são eliminados da rota(figura 6,b). Finalmente, os dois novos arcos são inseridos(figura 6,c). A cada novo clicar no botão, uma nova rodada do algoritmo é realizada e dois novos arcos são retirados e inseridos, assim como o *histórico dos resultados* é alterado. Quando não existir mais trocas 2-Opt possíveis, a *Janela de Resultados* é exibida com a rota encontrada.

O botão de ação *Exibir a Rota Ótima* tem como única função a visualização da rota de custo ótimo (figura 7) se ela existir. Para os problemas cadastrados no LOBO, as rotas ótimas existem e estão disponíveis. Para problemas carregados pelo usuário, este tem a possibilidade de inserir a rota ótima também. O último botão de ação do LOBO é o que comanda o modo tutoriado e o modo *expert*. Quando este botão está pressionado, o usuário se encontra em modo tutoriado e as telas de apresentação e dicas são apresentadas para o mesmo. Se o botão não estiver pressionado, o modo *expert* é ativado e o objeto se torna uma ferramenta de simulação.

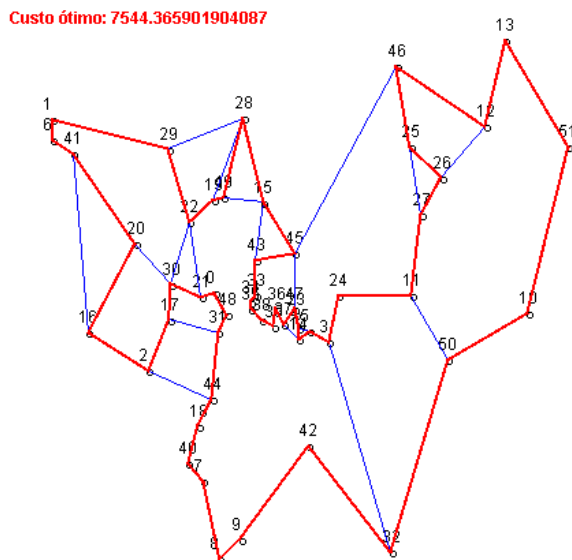


Figura 7 – Exibição da rota ótima para o problema berlin52.tsp

A área de simulação é a principal interface do objeto LOBO. Nesta área, o usuário pode observar os dados do problema em sua forma gráfica, ou seja, como um conjunto de pontos representando as cidades e um conjunto de linhas representando as ligações entre estas cidades. Cada cidade é representada por um número que se encontra logo acima de um pequeno círculo. A área de simulação foi implementada através de uma câmera sintética (FOLEY, 1997), que pode ser definida como um dispositivo de *software* que permite representar objetos em um plano de projeção. Desta forma, se o plano de projeção for alterado, a representação do objeto também é alterada. A implicação prática deste conceito transforma o LOBO em um visualizador de problemas de otimização que se adapta ao tamanho da janela que o usuário definir, ou seja, ao redimensionar a janela, os *espaços* entre as cidades aumentam ou diminuem e o problema se adapta visualmente ao novo tamanho. É importante observar que os valores do problema não se alteram, ou seja, a distância calculada entre duas cidades nunca muda de valor. O que muda é sua posição cartográfica na interface visualizada.

4. Conclusões: O LOBO e suas possibilidades no ensino de Heurísticas e Metaheurísticas

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

A experimentação e a interação social são os principais instrumentos para o funcionamento adequado de um modelo pedagógico que utiliza um sistema computacional como mediador. A primeira característica pode ser definida como o próprio cerne do desenvolvimento do LOBO, cujo objetivo central está na condução de experimentos com os algoritmos de otimização combinatória. A própria liberdade que é fornecida ao aluno, como comentado anteriormente, está coerente com esta expectativa. Muito mais que os acertos, os erros cometidos pelos alunos – aqui traduzidos como simulações com resultados ruins – trazem valiosas experiências e maturidade para o pensamento científico acerca do que está sendo observado, principalmente no que concerne à construção de heurísticas novas ou modificações nos algoritmos já consagrados. O professor pode auxiliar neste processo, trazendo problemas instigantes e que conduzam a resultados ora satisfatórios e ora frustrantes, com o intuito de apresentar aos alunos os pontos positivos e negativos de cada heurística ou metaheurística estudada. Dificilmente, nesta área do conhecimento, há a definição de métodos heurísticos que conduzam a um desempenho satisfatório para a imensa gama de problemas e sub-problemas definidos na literatura. Usualmente, algoritmos que trazem excelentes resultados para uma classe de problemas podem ter desempenho pífio em outra ligeiramente diferente. Desta forma, é muito mais importante que os alunos reconheçam as limitações e os aspectos positivos das heurísticas e metaheurísticas como um todo do que simplesmente decorar o funcionamento de um ou mais algoritmos. É neste ponto específico que um simulador pode atuar de forma mais intensiva, fornecendo um ambiente único e controlado que pode ser experimentado pelos usuários e testado a exaustão, fornecendo detalhes e dados que poderão ser analisados de forma criativa pelos professores e alunos.

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

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

alunos a competir entre si através dos resultados alcançados pelos seus algoritmos. Com a facilidade de visualizar graficamente e a qualquer momento a solução que está sendo construída, assim como seu custo, é possível motivar os alunos a explorarem as possibilidades e interagir com o ambiente de forma mais produtiva.

Ainda em relação a este aspecto, é importante salientar que o material introdutório ao Problema do Caixeiro Viajante, contido no próprio objeto, traz aspectos importantes acerca do problema que podem ser utilizados como um fator motivacional adicional. Além disso, através da implementação de aplicações práticas do problema teórico, condizentes com seu universo, os alunos tendem a considerar com especial interesse o assunto em questão. Assim, a dissociação entre a teoria acadêmica (algoritmos heurísticos) e a aplicação (problemas e soluções) inibe a possibilidade da construção de conhecimentos significativos, pois o vácuo entre as duas fronteiras, por um lado, não fornece de modo adequado as possibilidades de o aluno pensar cientificamente, gerindo e expandindo seu conhecimento e, por outro, não o capacita para atuar profissionalmente. Finalmente, em relação a disponibilidade do objeto de aprendizagem, ou seja, a possibilidade da interação com o mesmo em qualquer lugar e de qualquer modo, o LOBO atinge parte destes propósitos. Desenvolvido com a tecnologia Java, ele pode ser simulado em qualquer computador que provenha esta tecnologia. No entanto, a tecnologia baseada no JWS ainda não permite sua migração para dispositivos portáteis como celulares ou computadores de mão.

Referências

- ANGOTTI, J. A. P., DELIZOICOV, D. N. *Metodologia do Ensino de Ciências*. São Paulo: Cortez, 1990.
- CARR, W., KEMMIS, S. *Becoming Critical: Education, Knowledge and Action Research*. Brighton, Uk: Falmer Press, 1986.
- CORNE, D., DORIGO, M., GLOVER, F. *New ideas in Optimization*. McGraw-Hill, 1999.
- DE BASTOS, Fábio da Purificação.; MÜLLER, Felipe Martins. *Criando Desafios em Informática*. In: Atas da IV Escola de Verão sobre Investigação-Ação Educacional. UFSM, Santa Maria, RS, 1999.
- FOLEY, James D. *Computer Graphics: Principles and Practice*. Ed. Reading: Addison-Wesley, 1997.
- LAWLER, E.L., LENSTRA, J.K., RINNOOY KAN, A.H., SHMOYS, D.B. *The traveling salesman problem*. John Wiley & Sons, 1985.
- MÜLLER, Felipe Martins, DE BASTOS, Fábio da Purificação. *Matriz Dialógico-Problematizadora como Ferramenta Organizadora do Trabalho Escolar no AMEM*. In: Congresso Nacional de Ambientes Hipermedia para Aprendizagem. Florianópolis: 2004.
- REI, Gerhard. *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer-Verlag, 1994.
- REINELT, G. *TSP-LIB – A Traveling Salesman Library*. ORSA Journal of Computing, n. 3, p. 376-384, 1991.