

V JAI

JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA DA UNICENTRO
de 1 a 5 de outubro de 2013

**Palestras e minicursos sobre
temas atuais em informática**

ANAIS
ISSN: 2177-708X

apoio



realização

Departamento de
Ciência da Computação



V JAI-UNICENTRO

V Jornada de Atualização em Informática da UNICENTRO

Departamento de Ciência da Computação

01 a 05 de outubro de 2013

Guarapuava – PR

Catálogo na Publicação
Fabiano de Queiroz Jucá – CRB 9/1249
Biblioteca Central da UNICENTRO, Campus Guarapuava

Jornada de Atualização em Informática da Unicentro (5. : 1-5 out. 2013 : Guarapuava)
J82a Anais da V Jornada... / coordenado [por] Fábio Hernandes e Josiel Neumann
Kuk -- Guarapuava : Unicentro, 2013.
1 cd-rom

ISSN: 2177-708X

Evento realizado entre 01 e 05 de outubro de 2013

1. Informática - Educação. 2. Software. 3. Tecnologia. 4. Internet. 5. Educação.
I. Título.

CDD 004

“Esta obra foi editada a partir de originais entregues, já compostos pelos autores.”

V JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA DA UNICENTRO – V JAI-UNICENTRO

Coordenação Geral

Fábio Hernandes

Josiel Neumann Kuk

Comissão Editorial

Carolina Paula de Almeida

Inali Wisniewski Soares

Revisão Gramatical

Suellen de Fátima Egiert

Comissão Organizadora

Carolina Paula de Almeida

Evanise Araújo Caldas

Fábio Hernandes

Inali Wisniewski Soares

Josiel Neumann Kuk

Richard Aderbal Gonçalves

Tiago Gerke

Comissão Científica

Ana Elisa Tozetto Piekarski (DECOMP/UNICENTRO)

Andres Jessé Porfírio(UTFPR-Guarapuava)

Angelita Maria de Ré (DECOMP/UNICENTRO)

Carolina Paula de Almeida (DECOMP/UNICENTRO)

Diego Marczal(UTFPR-Guarapuava)

Evanise Araújo Caldas (DECOMP/UNICENTRO)

Fábio Hernandes (DECOMP/UNICENTRO)

Gisane Aparecida Michelon (DECOMP/UNICENTRO)

Inali Wisniewski Soares (DECOMP/UNICENTRO)

Josiane Michalak Hauagge Dall Agnol (DECOMP/UNICENTRO)

Josiel Neumann Kuk (DECOMP/UNICENTRO)

Lucélia de Souza (DECOMP/UNICENTRO)

Luciane Telinski Wiedermann Agner (DECOMP/UNICENTRO)

Marcos Alexandre Bronoski (DECOMP/UNICENTRO)

Marcos Antonio Quináia (DECOMP/UNICENTRO)

Mauro Henrique Mulati (DECOMP/UNICENTRO)

Mauro Miazaki (DECOMP/UNICENTRO)

Richard Aderbal Gonçalves (DECOMP/UNICENTRO)

Sandra Mara Guse Scós Venske (DECOMP/UNICENTRO)

Sandro Rautenberg(DECOMP/UNICENTRO)

Sediane Carmem Lunardi Hernandes(UTFPR-Guarapuava)

Sílvia Amélia Bim(UTFPR-Curitiba)

Sílvio Luiz Bragatto Boss(UTFPR-Pato Branco)

Tiago Gerke (DECOMP/UNICENTRO)

Reitoria

Reitor: Prof. Aldo Nelson Bona

Vice-Reitor: Prof. Osmar Ambrósio de Souza

Pró-Reitorias

Ensino: Prof^ª. Márcia Terezinha Tembil

Pesquisa e Pós-Graduação: Prof. Marcos Ventura Faria

Extensão e Cultura: Prof^ª. Marquiana de Freitas Vilas Boas Gomes

Administração e Finanças: Prof. Amarildo Hersen

Recursos Humanos: Prof^ª. Roseli Machado

Planejamento: Prof. Silvano Simões Rocha

Direção do Campus Cedeteg

Diretor: Prof. Juliano Tadeu Vilela Resende

Vice-Diretora: Prof^ª. Sonia Maria Kurchaidt

Setor de Ciências Exatas e de Tecnologia

Diretor: Prof. Fábio Hernandes

Vice-Diretora: Prof^ª. Karina Worm Beckmann

Departamento de Ciência da Computação

Chefe: Prof^ª. Luciane Telinski Wiedermann Agner

Vice-Chefe: Prof. Josiel Neumann Kuk

APRESENTAÇÃO

Da mesma forma que as edições anteriores, a V JAI-UNICENTRO tem por objetivo divulgar o Bacharelado em Ciência da Computação da UNICENTRO para a comunidade em geral. Desse modo, são apresentados aos participantes conteúdos para atualização e disseminação de técnicas e metodologias, conforme as tendências do mercado.

O evento é constituído por palestras e minicursos, sobre temas atuais e ministrados por profissionais experientes, que são convidados a compartilhar seus conhecimentos. Além disso, as sessões de apresentações de trabalhos são compostas pela seleção de trabalhos dos alunos que submeteram resumos de seus trabalhos de conclusão de curso, estágio supervisionado e iniciação científica, que depois de avaliados e revisados constituem o conteúdo destes Anais.

Esperamos que os assuntos aqui contidos possam contribuir para o enriquecimento técnico dos seus leitores e sirvam de referência para trabalhos futuros.

Agradecemos aos que, de forma voluntária, aceitaram o convite e possibilitaram a realização do evento, incluindo palestrantes e instrutores dos minicursos. Também agradecemos o empenho dos autores que submeteram seus resumos e da Comissão Científica que os avaliou, razão da edição deste material.

*Coordenação Geral e Comissão Editorial
V JAI-UNICENTRO*

Algoritmo Imuno-Genético para o Prognóstico do Câncer de Próstata	6
Alex Wargenhak, Angelita Maria de Ré	
Protótipo para Controle de Cardápio e Pedidos Utilizando Android com Servidor de Interatividade	10
Alexandre Antoniu Neto, Josiel Neumann Kuk, Andres Jessé Porfirio	
Aplicação de Abordagem Baseada em Otimização por Colônia de Formigas para o Problema de Cobertura de Conjuntos	14
Alexandre Silvestre Ferreira	
Algoritmo Minimax Aplicado ao Jogo de Damas	20
Guilherme L. de Mello, Mauro Miazaki, Andres Jessé Porfirio	
Prototipação de um <i>e-commerce</i> para Pizzarias com disponibilização de dados semânticos	25
Marcelo Araújo, Sandro Rautenberg	
Evolução Diferencial Adaptativa para o Problema do Despacho Econômico de Energia Elétrica	29
Lucas Prestes, Ricardo H. R. Lima, Josiel N. Kuk, Carolina P. Almeida, Sandra M. G. S. Venske, Richard A. Gonçalves	
Um Estudo da Aplicação de Redes Neurais Artificiais <i>Multilayer Perceptron</i> no Prognóstico de Câncer de Próstata	33
Victor Alexandre Padilha, Sandro Rautenberg	
Estudo da Variação de Parâmetros do Algoritmo Genético na Evolução de Criaturas	37
Willian Eiji Yassue, Mauro Miazaki, Andres Jessé Porfirio	
Projeto de um Aplicativo Android de Monitoramento de Veículos do Transporte Público	41
Geovane Fedrechski, Richard Aderbal Gonçalves, José Neumann Kuk	
Paralelização em GPU de algoritmo baseado em otimização por colônia de formigas aplicado ao problema de coloração de grafos	45
Paulo Roberto Urio, Mauro Henrique Mulati	
Algoritmo de Busca Gravitacional aplicado ao Problema do Despacho Econômico de Energia Elétrica	49
Eduardo Thomas Feliczaki, Richard Aderbal Gonçalves	

Algoritmo Imuno-Genético para o Prognóstico do Câncer de Próstata

Alex Wargenhak¹, Angelita Maria de Ré (orientadora)

Departamento de Ciência da Computação

Universidade Estadual do Centro-Oeste (UNICENTRO) – Guarapuava, PR – Brasil

¹alex_wargenhak@yahoo.com.br

Resumo. *Este trabalho trata da implementação de um algoritmo Imuno-Genético para auxílio no prognóstico do estágio do câncer de próstata. Para facilitar ajustes e validações do algoritmo proposto foi utilizada uma base de dados real, que contém informações dos estados pré e pós-cirúrgicos dos pacientes diagnosticados com a doença.*

Palavras-chave: *Algoritmos Genéticos; Algoritmos Híbridos; Câncer de Próstata; CLONALG Paralelo.*

1. Introdução

A computação evolucionária é um ramo da Ciência da Computação que tem por base os mecanismos evolutivos encontrados na natureza. Assim, nos Algoritmos Genéticos (AG) populações de indivíduos são criadas e submetidas aos operadores genéticos. Estes operadores caracterizam a qualidade de cada indivíduo como solução do problema, utilizando a avaliação em um processo de evolução natural (LINDEN, 2006). Também, considerando os algoritmos inspirados na natureza, pode-se destacar os Sistemas Imunológicos Artificiais (SIA), que são algoritmos inspirados no Sistema Imunológico Natural (SIN), sendo aplicados na solução de diversas classes de problemas como, por exemplo, reconhecimento de padrões (DE CASTRO, 2001).

Há, na literatura, diversos trabalhos que abordam pesquisas desenvolvidas com a utilização de técnicas de computação natural aplicadas ao câncer de próstata, de modo a contribuir para melhorar o prognóstico da doença. Neste contexto, pode-se destacar os trabalhos de Tosatti *et al.* (2008), entre outros. O enfoque no prognóstico do câncer de próstata se deve ao fato de que este é o 2º tipo de câncer mais comum entre os homens no Brasil e o 6º tipo no mundo (INCA, 2013).

Neste contexto, e com o intuito de investigar a possibilidade de um algoritmo Imuno-Genético reconhecer padrões existentes em bases de dados de câncer de próstata, foi implementado um AG para representar os anticorpos inatos existentes em um SIN, pois os anticorpos inatos são responsáveis por uma rápida resposta imunológica na identificação de alguns antígenos (DE CASTRO, 2001). Na sequência, este algoritmo foi integrado ao CLONALG implementado, gerando um sistema Imuno-Genético desenvolvido na linguagem Java. Além disso, como forma de facilitar ajustes e validações do algoritmo Imuno-Genético proposto, foi utilizada uma base de dados disponibilizada pela SEER (*Surveillance Epidemiology and End Results*) (SEER, 2003). Esta base possui informações dos estados pré e pós-cirúrgicos de pacientes com câncer de próstata.

2. Metodologia

Sendo um dos objetivos da pesquisa auxiliar no prognóstico do estágio do câncer de próstata, é importante destacar alguns conceitos. Segundo Tosatti *et al.* (2008), determinar

o grau de confinamento do câncer é um fator fundamental na análise do benefício da cirurgia para a remoção do tumor. A cirurgia justifica-se apenas se o tumor estiver confinado no órgão, caso contrário, outras terapias podem ser aplicadas. Por isso, auxiliar o médico na predição do estágio da doença torna-se relevante.

Para avaliar a extensão da doença, são comumente utilizados o exame digital retal (DRE – *Digital Rectal Examination*), a dosagem de antígeno prostático específico (PSA – *Prostate Specific Antigen*) e o escore de *Gleason* da biópsia. A extensão patológica baseia-se nos achados cirúrgicos que auxiliam a identificar a extensão da doença, sendo essa classificada em dois grupos: *Confinados*, em que o câncer está localizado no interior da cápsula prostática, e *Não-Confinados*, quando o tumor já se encontra em regiões além da cápsula prostática.

Nesta pesquisa, foram considerados os dados da base SEER relacionados à idade, raça, PSA, *gleason*, DRE e extensão patológica. Com base nesses dados, foi implementada uma versão do AG para otimizar um conjunto de soluções, que são utilizadas na primeira etapa do algoritmo Imuno-Genético proposto. Essa etapa corresponde à geração dos anticorpos aleatórios, que, neste caso, são inicializados pelas soluções obtidas no AG implementado.

No AG implementado foram utilizadas duas representações do indivíduo, uma para armazenar os dados da base SEER, chamada de *Indivíduo Original*, e outra para o processamento do AG, chamada de *Indivíduo Gerado*. A estrutura do Indivíduo Gerado é composta pelas características idade, raça, PSA, *gleason*, DRE e extensão patológica.

O cálculo do *fitness* reflete a habilidade de adaptação de um indivíduo, isto é, indica quão parecido o Indivíduo Gerado é dos Indivíduos Originais da base. Foram utilizados 5 modelos de cruzamento, de 1, 2 e 3 pontos, entre indivíduos escolhidos aleatoriamente na população, e os modelos de 1 e 2 pontos parecidos com os modelos anteriores, mas com a restrição de que o cruzamento ocorrer somente entre pares de indivíduos pertencentes ao mesmo grupo. Para a mutação foram utilizados 3 modelos de 1, 2 e 3 pontos. Tanto os cruzamentos quanto as mutações são aplicados sobre uma quantidade de indivíduos selecionados aleatoriamente na população por meio de taxas. Também foram utilizados 2 modelos de seleção, um baseado no Ranking e outro nos Melhores e Piores (LINDEN, 2006).

Após a implementação do AG, foi possível otimizar a população inicial para o algoritmo Imuno-Genético proposto. Assim, passou-se para a segunda etapa da pesquisa, que consiste na implementação do CLONALG.

O algoritmo CLONALG implementado baseou-se na versão sequencial desenvolvida por De Castro (2001). Porém, devido a problemas encontrados durante a execução de testes do algoritmo, foi desenvolvida uma versão paralela, que se baseia em trabalhos relacionados (WATKINS, 2005). Essa versão paralela divide inicialmente a população de antígenos de acordo com o número de processadores, sendo que cada processador executará uma instância do algoritmo CLONALG sequencial para a devida subpopulação de antígenos. Após as execuções dessas instâncias, as populações de anticorpos obtidas são armazenadas em uma única população que, posteriormente, é utilizada para calcular a taxa de acerto, que corresponde à capacidade de reconhecimento dos antígenos apresentados.

Cada antígeno armazena os dados de um registro da base SEER, sendo composto pelas características: idade, raça, PSA, *gleason*, DRE e patologia. Um anticorpo é composto pelas mesmas características da representação do antígeno, acrescido da

característica afinidade. As demais etapas do CLONALG implementado seguem as fases descritas no CLONALG proposto por De Castro (2001), utilizado para o reconhecimento de padrões. Após essa etapa, passou-se para a terceira etapa, que corresponde à utilização do AG para otimizar o CLONALG paralelo.

O algoritmo Imuno-Genético proposto inspira-se em um conceito relacionado ao SIN, que consiste no princípio de que cada indivíduo nasce com um repertório de anticorpos capazes de reconhecer um certo conjunto de antígenos. Para simular esse comportamento natural é utilizado o AG implementado, que produz uma população inicial de indivíduos utilizada no algoritmo CLONALG implementado. Considerando o conceito Imuno-Genético proposto, essa etapa corresponde à geração dos anticorpos aleatórios na fase inicial do algoritmo CLONALG (DE CASTRO, 2001).

Nesta etapa, a população gerada no AG é considerada como a população inicial de anticorpos no algoritmo Imuno-Genético proposto. Essa população de anticorpos é inicialmente dividida pelo número de processadores, assim como ocorre com a população de antígenos no algoritmo CLONALG implementado. Cada subpopulação resultante da divisão será utilizada para inicializar a população de anticorpos nas instâncias do algoritmo CLONALG mencionado.

3. Testes e Resultados

Com o intuito de observar os resultados obtidos foram realizadas três baterias de testes. Salienta-se que para a padronização dos testes realizados, utilizou-se a base de dados SEER. Como metodologia de realização de testes, primeiramente, foi executado o AG implementado. Na sequência, foram rodados os testes com o CLONALG implementado e, por último, testes com o algoritmo Imuno-Genético proposto.

Para o AG implementado foram realizadas aproximadamente 1.000 execuções do algoritmo, obtendo-se a melhor configuração com os seguintes parâmetros: população de 155.181 indivíduos; número de gerações 60; modelo de mutação de um ponto com taxa de 0,4%; modelo de cruzamento de um ponto entre indivíduos do mesmo grupo, com taxa de 1%; e modelo de seleção Ranking com taxa de 80%. Nesta configuração, foi obtida uma taxa de acerto aproximada de 88%, que corresponde ao número de Indivíduos Gerados iguais aos Indivíduos Originais da base. Vale ressaltar que o número de execuções do AG implementado foi superior ao número de execuções dos outros algoritmos implementados, pois o AG implementado possui um número maior de parâmetros do que os testados para encontrar a melhor configuração citada.

Para o CLONALG implementado foram executadas 500 execuções do algoritmo, que obteve uma taxa de acerto de 33,66%. A taxa mencionada diz respeito à quantidade de anticorpos produzidos e que possuem uma alta afinidade com os antígenos da base de dados SEER, isto é, a capacidade de reconhecimento de padrões do algoritmo. Nas colunas 2 e 3 da Tabela 1 são mostrados os parâmetros e acertos que obtiveram os melhores resultados nos testes realizados para o CLONALG implementado.

Para o Imuno-Genético proposto foram realizadas aproximadamente 500 execuções, com a melhor taxa de acerto de 66,31%. A taxa de acerto do algoritmo Imuno-Genético é calculada da mesma forma que no CLONALG implementado. Nas colunas 4 e 5 da Tabela 1 são apresentados os parâmetros e acertos que obtiveram os melhores resultados nos testes realizados no Imuno-Genético.

Tabela 1. Taxas de acertos em cada parâmetro dos algoritmos AG e Imuno-Genético

	CLONALG	IMUNO-GENÉTICO
--	---------	----------------

Parâmetros	Melhores parâmetros	Taxas de acertos	Melhores parâmetros	Taxas de acertos
Gerações	35	23,57%	20	66,11%
Anticorpos	71616	22,64%	23874	47,09%
Beta (fator multiplicativo)	0,7%	33,4%	0,7%	66,31%
Melhores anticorpos para clonagem	40%	33,66%	40%	65,85%
Piores anticorpos a serem substituídos	1,5%	15,13%	1%	46,64%
Melhores clones a substituírem os piores anticorpos da população de memória	80%	22,58%	55%	46,73%
Tamanho da população restante ¹	30%	14,65%	30%	45,62%

4. Conclusão

Neste trabalho foi proposto o algoritmo Imuno-Genético para auxílio no prognóstico do estágio do câncer de próstata. Este algoritmo mostrou um desempenho não muito significativo, pois a taxa de acerto foi de 66,31%, isto é, a quantidade de indivíduos gerados pelo algoritmo que são reconhecidos na base de dados. Entretanto, o algoritmo Imuno-Genético implementado necessita de uma população de anticorpos e números de gerações inferiores para obter resultados superiores ao CLONALG implementado. Isso se deve à capacidade inicial de reconhecimento de alguns antígenos, devido à população de anticorpos gerada no AG implementado.

Em trabalhos futuros, pretende-se executar novos testes para ajustar a configuração dos parâmetros do algoritmo Imuno-Genético; e para aprimoramento do CLONALG implementado, especificamente no cálculo da afinidade dos anticorpos.

5. Referências Bibliográficas

- DE CASTRO, L. N. (2001), **Engenharia imunológica: desenvolvimento e aplicação de ferramentas computacionais inspiradas em sistemas imunológicos artificiais**. Tese (doutorado em Engenharia Elétrica) – FEEC/Unicamp, Campinas, SP.
- INCA – Instituto Nacional do Câncer, Ministério da Saúde (2013), **Próstata**, <http://www2.inca.gov.br/wps/wcm/connect/tiposdecancer/site/home/prostata>, Maio.
- LINDEN, R. (2006), **Algoritmos Genéticos: Uma importante ferramenta da inteligência computacional**, Brasport.
- SEER – Surveillance Epidemiology and End Results (2003), **Public-Use Data (1073-2000)**, www.seer.cancer.gov.
- TOSATTI, M. A. *et al.* (2008), Algoritmo híbrido genético-fuzzy aplicado em câncer de próstata. In: **HÍFEN**, v. 32, n. 62, p. 164-171.
- WATKINS, A.. **Exploiting immunological metaphors in the development of serial, parallel and distributed learning algorithms**. Tese (doctor of philosophy) – Computing Laboratory, University of Kent, Canterbury, UK, 2005.

¹ Taxa para determinar o tamanho da população restante baseada no repertório da população de anticorpos. Este parâmetro baseia-se no algoritmo CLONALG de WEKA *Classification Algorithms*, <http://downloads.sourceforge.net/weka/classalgos>.

Protótipo para Controle de Cardápio e Pedidos Utilizando Android com Servidor de Interatividade

Alexandre Antoniu Neto¹, Josiel Neumann Kuk², Andres Jessé Porfirio³

^{1,2}Departamento de Ciência da Computação – Universidade Estadual do Centro-Oeste (UNICENTRO) – Guarapuava – PR – Brasil

³Departamento de Sistemas para Internet - Universidade Tecnológica Federal do Paraná (UTFPR) – Guarapuava – PR – Brasil

{alexandreantoniun, josielkuk}@gmail.com, andresjesse@yahoo.com.br

Abstract. *This article depicts both the development and the technologies used to implement a prototype system consisting of an application and an interactivity server, in order to facilitate the running of restaurants and eateries, by automating the collection of orders taken and menu control. As a result of such, the objectives of the work were accurately carried out, being these learning with new technologies and prototype implementation.*

Key-words. *Application, Management, Food Industry, Mobility.*

Resumo. *Este artigo descreve o desenvolvimento e as tecnologias utilizadas na implementação de um protótipo de sistema, composto por um aplicativo e um servidor de interatividade, para facilitar o gerenciamento de restaurantes e lanchonetes, automatizando a coleta dos pedidos e controle do cardápio. O trabalho resultou na execução dos objetivos, que são o aprendizado das tecnologias e a implementação do protótipo.*

Palavras-chave. *Aplicativo; Gerenciamento; Ramo Alimentício; Mobilidade.*

1. Introdução

O Sistema Operacional Android está presente em centenas de milhões de dispositivos móveis em mais de 190 países no mundo. É a maior base instalada de qualquer plataforma móvel e vem crescendo rapidamente (DEVELOPERS, 2013). Segundo Lecheta (2009), o Android é o primeiro projeto de uma plataforma *open source* para dispositivos móveis criado pela OHA, um grupo formado em 2007 por gigantes do mercado de telefonia e de celulares, liderados pela Google.

Em restaurantes é ainda comum o uso de comandas e blocos de papel para fazer os pedidos, porém, nesses locais, a mobilidade é fundamental e o uso deste tipo de recurso pode causar erros de interpretação na cozinha, além de consumir mais tempo.

Segundo a Intel Corporation (2013), a mobilidade pode ser definida como a capacidade de acessar informações a partir de qualquer lugar, a qualquer hora. Pode-se então traduzir mobilidade como a necessidade de dispositivos móveis com conexão sem fio. Portanto, esses recursos seriam suficientes para automatizar a coleta de pedidos de estabelecimentos e manter a mobilidade exigida.

Considerando isso, este trabalho disponibiliza um protótipo de um aplicativo para atender à demanda de restaurantes, bares, lanchonetes e similares, através de uma

ferramenta que agilize o atendimento do garçom e um servidor local para melhorar o controle do cardápio do estabelecimento, buscando facilitar o gerenciamento do local e um trabalho que resulte em menos erros de interpretação na cozinha.

2. Materiais e Métodos

O protótipo divide-se em um aplicativo Android, criado para agilizar o atendimento do garçom, e um servidor de interatividade, que busca facilitar o gerenciamento do local, oferecendo controle sobre o cardápio do dia.

2.1. Cliente

Cada garçom terá um dispositivo móvel com o sistema, e, assim, quando um cliente efetuar o pedido, este será enviado para o servidor; então, o pedido poderá ser visualizado pela equipe responsável pela cozinha. Dessa maneira, o cliente não perderá a comodidade de ser atendido pelo garçom e o dono do estabelecimento terá facilidade em controlar o seu cardápio e pedidos.

A tela que mostra as mesas disponíveis no estabelecimento pode ser observada no *link*¹. Nessa tela, os atendentes poderão visualizar quais mesas estão abertas no momento ou podem abrir uma delas. Caso já esteja aberta, é mostrada a etiqueta “Aberta”.

Quando é selecionada uma das mesas, abre-se outra tela para que o atendente possa inserir o pedido do cliente. Esta tela é dividida em duas colunas: coluna esquerda e coluna direita. Como pode ser observado no *link*², a coluna esquerda corresponde às categorias dos produtos cadastrados no servidor e a coluna direita mostra uma prévia dos produtos consumidos pelo cliente.

Quando um novo produto é inserido, este é colocado na comanda, onde ficará disponível para edição até ser enviado (botão “Confirmar”) para o servidor. Após isso, o botão de remoção do produto (“-”) é desabilitado. O botão “Finalizar”, por sua vez, finaliza a mesa no aplicativo e limpa todos os dados dos produtos consumidos, deixando um histórico somente no servidor.

2.2. Servidor

O servidor é o responsável pela manutenção das informações. Nele é possível adicionar, excluir e modificar produtos e o cardápio do dia. Para armazenar os dados dos produtos, categorias, mesas, pedidos e o histórico, fez-se necessário um SGBD (Sistema Gerenciador de Banco de Dados).

Todas as informações dos dados são armazenadas no servidor com a utilização do SQLite. Através dessa biblioteca é criado o banco de dados e são realizadas as consultas, inserções e remoções.

A implementação do Servidor fez-se com a utilização da linguagem de programação Java. Para a interface gráfica, foram utilizados os componentes do *Swing Toolkit*, kit de ferramentas criado pela Sun Microsystems para o desenvolvimento da interface com o usuário em Java.

A interface gráfica do Servidor favorece a administração, pois permite a definição

¹Disponível em: <https://www.dropbox.com/s/cdlugo33ypnwdsp/mesas.png>

²Disponível em: <https://www.dropbox.com/s/o11roayccjno61y/layout.png>

das seguintes configurações: Cadastro, Edição e Exclusão de mesas; Cadastro, Edição e Exclusão de categorias; Cadastro, Edição e Exclusão de produtos; Edição dos Pratos do Dia; Exibição dos Pedidos Abertos; Manter um histórico dos pedidos.

2.3. Comunicação entre Cliente/Servidor

O modelo entre Cliente/Servidor implica em um processamento cooperativo das requisições feitas por um cliente para o servidor, o qual processará e retornará os resultados para o cliente. O processamento é iniciado e parcialmente controlado pelo cliente, mas tanto o cliente quanto o servidor cooperam para executar com sucesso a aplicação.

Para criar páginas dinâmicas com Java é necessário um *servlet*, cujo objetivo é receber chamadas HTTP (*request*) e processá-las, devolvendo uma resposta (*response*) ao cliente. Por isso, a tecnologia *Java Servlet* pode ser interpretada como uma implementação do protocolo HTTP. Os *servlets* devem ser implantados dentro de um *Container Servlet*, o qual auxiliará na manipulação do protocolo HTTP.

Quando o aplicativo faz uma solicitação para um *servlet*, o servidor não entrega a solicitação diretamente para ele, mas para o *container* que o contém. O *container* encontrará o *servlet* apropriado baseado na solicitação e, após isso, a solicitação de entrada é convertida em um objeto *ServletRequest*, que conterá as informações do pedido.

O *servlet* é responsável pelo tratamento da solicitação de entrada e por preencher o objeto de resposta que será entregue ao cliente. Após o processamento do *servlet* ser efetuado, o *container* leva a informação para o *ServletResponse* e utiliza-o para gerar uma resposta HTTP, a qual será enviada para o aplicativo do cliente.

3. Resultados e discussão

3.1. Testes

O planejamento dos testes ocorreu em três níveis e em paralelo ao desenvolvimento do protótipo, com base nos conhecimentos adquiridos de Dias Neto (2007). Os três níveis utilizados foram: Teste de Unidade, Teste de Integração e Teste de Sistema.

3.2. Resultados

Como resultado, tem-se um protótipo de um aplicativo que se comunica com um servidor multiusuários e que pode atender a uma possível demanda de restaurantes. Com uma interface gráfica intuitiva e simples, qualquer funcionário pode rapidamente aprender e dominar o aplicativo, garantindo agilidade e eficiência para as suas tarefas, e contribuindo para um melhor gerenciamento do local.

Outro fator importante, adquirido como resultado, é o conhecimento relacionado à plataforma Android e todas as tecnologias envolvidas. A implementação da comunicação incluiu uma carga de pesquisas extras, pois como as tecnologias de comunicação *web* (Serviços *Web*) possuem vantagens e desvantagens, foi necessário entender e escolher a que melhor se encaixava no problema.

4. Conclusões

Através desse trabalho foi possível idealizar e desenvolver um protótipo de uma

ferramenta para restaurantes e lanchonetes capaz de automatizar a coleta dos pedidos e controle do cardápio. Da mesma forma, foi possível desenvolver o servidor de interatividade que faz a comunicação com o aplicativo, atendendo as funcionalidades inicialmente previstas.

O protótipo desenvolvido é uma base para um sistema de administração de estabelecimentos comerciais focados no ramo alimentício, como bares e restaurantes. Possui, como vantagens, um Servidor multiplataforma, a comodidade para o cliente, que continua a efetuar os pedidos a um garçom, e a possibilidade do controle de gastos.

O enriquecimento dos conhecimentos adquiridos com a realização deste trabalho foi de grande importância, especialmente por se tratar do momento de grandes oportunidades no ramo alimentício. Muitas empresas procuram novos desenvolvedores de sistemas móveis a fim de criar ou adaptar serviços existentes e suprir a demanda do mercado, sendo, portanto, uma boa opção de atuação no mercado de trabalho.

Como trabalho futuro, pretende-se finalizar o protótipo, tornando-o um sistema completo e adequado para restaurantes. Para isso, podem ser incluídos módulos extras e/ou funções extras que melhorem o gerenciamento do estabelecimento.

Testar o aplicativo em um ambiente real será, também, um momento importantíssimo para se obter informações de possíveis novos módulos (com base na necessidade) e também de possíveis correções (se necessárias – pois, como o sistema é genérico, alguns estabelecimentos podem ter peculiaridades).

5. Referências Bibliográficas

DEVELOPERS, ANDROID. **Android, the world's most popular mobile platform.**

Disponível em: <<http://developer.android.com/about/index.html>>. Acesso em: 14 fev. 2013.

DIAS NETO, Arilo Cláudio. Introdução a Teste de Software: Níveis de teste de software.

Qualidade de Software: Entenda os principais conceitos sobre Testes e Inspeção de Software, n. , p.54-59, 2007.

INTEL CORPORATION. **O conceito de mobilidade.** Referência com licença de direitos autorais limitados, adquiridos no site: <http://lz1.intel.com/copyright/>. Disponível em: <<http://www.nextgenerationcenter.com/detalle-curso/Mobilidade.aspx>>. Acesso em: 27 maio 2013.

LECHETA, Ricardo R.. Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. São Paulo: Novatec, 2009.

Aplicação de Abordagem Baseada em Otimização por Colônia de Formigas para o Problema de Cobertura de Conjuntos

Alexandre Silvestre Ferreira

Departamento de Ciência da Computação – Universidade Estadual do Centro-Oeste
(UNICENTRO)

Guarapuava – PR – Brasil

alexandrefscan@gmail.com

Resumo. Neste artigo é apresentada uma solução para o Problema de Cobertura de conjuntos, utilizando-se a meta-heurística de Otimização por Colônia de Formigas, mais precisamente o algoritmo Ant-Line. As implementações foram feitas em C++, e as instâncias utilizadas para teste são instâncias de Benchmark. Os resultados obtidos foram em torno de 6% distantes do ótimo e verificou-se que o pré-processamento tem impacto significativo na qualidade das soluções obtidas.

Palavras-chave: Meta-heurística; Otimização por Colônia de Formigas; Problema de Cobertura de Conjuntos.

1. Introdução

O Problema de Cobertura de Conjuntos (PCC) é um problema de otimização combinatória NP- difícil (CORMEN, 2009), de grande importância para a Ciência da Computação, já que possui diversas aplicações no mundo real, como se pode destacar: geração de escalas de tripulação (ferroviária e aérea, por exemplo) e localização de facilidades (OLIVEIRA, 1999).

Basicamente, um algoritmo ACO (do inglês, *Ant Colony Optimization*) é iterativo. Dado um grafo que representa o ambiente que a formiga percorrerá, cada vértice representa um componente da solução, e, no caso do PCC, um componente representa uma coluna. Em cada passo da construção de seu caminho, as formigas selecionam o próximo vértice através de um mecanismo heurístico que é baseado no feromônio (uma substância química deixada pelas formigas no ambiente enquanto fazem seu caminho). Sendo assim, cada vértice é escolhido com base na quantidade de feromônio presente nele e no valor da função heurística aplicada à coluna. (DORIGO & STUTZLE, 2004).

Muitos algoritmos baseados em ACO foram propostos para o PCC. Hadji *et al.* (2000) apresentou uma abordagem cujos resultados ainda não eram comparáveis aos obtidos por outros métodos. Lessing *et al.* (2004) desenvolveram uma aplicação do *Max-Min Ant System* (MMAS) e obtiveram um bom resultado. E Ren *et al.* (2008), que propuseram uma aplicação ACO e a compararam com o MMAS, testando com e sem busca local, e apresentaram resultados em média 0,17% distantes da solução ótima.

O objetivo deste trabalho é apresentar um estudo sobre o desempenho de uma

implementação da meta-heurística ACO para o PCC, estudar o problema, realizar testes e comparar os resultados obtidos com alguns resultados apresentados pela literatura.

2. O Problema de Cobertura de Conjuntos

O PCC pode ser definido por uma matriz binária A de $m \times n$ dimensões, na qual cada elemento pode ser 0 ou 1, e cada coluna dessa matriz está associada a um custo c_j . Se a coluna j cobre a linha i , então temos que o elemento $a_{ij} = 1$. O objetivo é escolher um subconjunto de colunas minimizando a soma de seus custos, de forma que toda linha deve ser coberta por pelo menos uma coluna (MULATI *et al.*, 2013).

3. Otimização por Colônia de Formigas

A meta-heurística de Otimização por Colônia de Formigas baseou-se, principalmente, nos conceitos apresentados pelo experimento das duas pontes dirigido por Deneubourg *et al.* (1990). Seu principal foco está na comunicação e cooperação das formigas, que é feita através do ambiente, com o propósito de encontrar a melhor solução. No Algoritmo 1, temos a estrutura geral de um algoritmo ACO.

Algoritmos ACO podem ser utilizados para a resolução tanto problemas de otimização combinatória estáticos quanto dinâmicos. Um ACO é um algoritmo iterativo. Dado um grafo que representa o ambiente que a formiga percorrerá, cada vértice representa um componente da solução no PCC, em cada iteração, e um número de formigas artificiais constrói, cada uma, sua solução, através da escolha de vértices presentes no grafo. Em cada passo da construção de seu caminho, a formiga seleciona o próximo vértice a ser escolhido através de um mecanismo heurístico que é baseado no feromônio e na informação heurística. No fim de cada interação, com base na qualidade das soluções construídas, os valores do feromônio são atualizados, sendo que a

Meta-heurística ACO

```
while Nao alcançou condicao de parada do
    construaSolucoes()
    buscaLocal()
    atualizaFeromonio()
end while
```

▷ Opcional

Algoritmo 1. Meta-heurística ACO (MULATI *et al.*, 2013)

evaporação serve para fazer com que as formigas evitem as escolhas ruins, enquanto o depósito serve para intensificar a atração das formigas para a escolha daquele componente para a solução.

4. Ant-Line

O algoritmo *Ant-Line* (MULATI & CONSTANTINO, 2011) tem sua estrutura similar à apresentada na meta-heurística ACO, constante no Algoritmo 2.

Para a construção da solução, cada formiga k deve primeiro selecionar uma linha, aleatoriamente, que ainda não foi coberta por nenhuma coluna da solução s_k , e então escolher uma coluna para cobrir esta linha, utilizando uma regra de decisão determinística. A fórmula da escolha da linha é dada pela Equação 1.

$$p_e^k = \begin{cases} \frac{1}{|M-R(s_k)|} & \text{se } e \notin R(s_k) \\ 0 & \text{se } e \in R(s_k) \end{cases}, \forall e \in M \quad (1)$$

Em que $R(s_k)$ é o conjunto de todas as linhas cobertas pelas colunas presentes na solução s_k , e M é o conjunto que contém todas as linhas.

Através da escolha da linha, o algoritmo seleciona um conjunto de colunas candidatas, escolhidos através da Equação 2.

$$N(e, s_k) = \{j | (j \notin S_k) \wedge (e \notin R(s_k)) \wedge (a_{ej} = 1)\}, \forall j \in C \quad (2)$$

Na qual e é a linha escolhida pela Equação 1. Após isso, o algoritmo seleciona, deterministicamente, um componente que está em $N(e, s_k)$, pela Equação 3.

$$j = \operatorname{argmax}_{h \in N(e, s_k)} \{\tau_h^\alpha \cdot \eta_h^\beta\} \quad (3)$$

A informação heurística η_j é calculada pela Equação 4 e α e β representam, respectivamente, a relevância do feromônio e da informação heurística.

$$\eta = \frac{k_j}{c_j} \quad (4)$$

Em que k_j e c_j representam, respectivamente, a cardinalidade e o custo da coluna j . A evaporação é feita de acordo com a Equação 5.

$$\tau_j = (1 - \rho) \cdot \tau_j, \forall j \text{ in } C \quad (5)$$

Sendo que τ_j representa o feromônio associado à coluna j , ρ representa a taxa de evaporação e C é o conjunto com todas as colunas.

O depósito segue um procedimento em que s' é a melhor solução da iteração e s^* a melhor solução da tentativa. O depósito do feromônio é feito por apenas pela melhor formiga da iteração seguindo a Equação 16.

$$\tau_j = \tau_j + \left(\frac{f(s^*)}{f(s')} \right)^y, \forall j \in s \quad (6)$$

Na qual y é um parâmetro cujo valor regula a quantidade de feromônio a ser depositada nos componentes da solução e a função f representa o valor da solução.

No Algoritmo 2 é apresentado o pseudocódigo de como o algoritmo *Ant-Line* foi implementado neste trabalho.

Pseudocódigo Ant-Line

```
 $\tau_j \leftarrow 1, \forall j \in C$  e  $f(s^*) = \text{INFINITO}$   
while  $i < \text{iteracoes}$  do  
   $f(s') \leftarrow \text{INFINITO}$   
  for  $k = 1 \rightarrow \text{numeroDeFormigas}$  do  
     $s_k \leftarrow \{\}$   
    while  $!(s_k.\text{estaCompleta}())$  do  
       $l \leftarrow \text{escolha uma linha que nao e coberta}$   
       $\text{por nenhuma coluna em } s_k \text{ (Equacao 1)}$   
       $j \leftarrow \text{escolha a coluna pertencente ao conjunto de colunas candidatas}$   
       $\text{com maior valor heuristico Equacoes(2 e 3)}$   
       $s_k \leftarrow s_k \cup \{j\}$   
    end while  
     $\text{eliminarColunasRedundantes}(s_k)$   
    if  $f(s_k) < f(s')$  then  
       $s' \leftarrow s_k$   
    end if  
  end for  
  if  $f(s') < f(s^*)$  then  
     $s^* \leftarrow s'$   
  end if  
   $\text{atualizaFeromonio}() \text{ Equacoes(5 e 6)}$   
end while  
return  $s^*$ 
```

Algoritmo 2. Pseudocódigo Ant-Line (MULATI & CONSTANTINO, 2011)

5. Testes e Resultados

Os testes foram realizados em um computador Intel® Core™ i5-3210M CPU, com frequência de 2.50GHz e 4 núcleos de processamento, com 6GB de memória RAM e sistema operacional Linux Fedora 18. Serão apresentados os resultados e comparações dos testes executados com o *Ant-Line* implementado no presente trabalho, bem como a implementação proposta por Mulati & Constantino (2011).

Os testes em Mulati & Constantino (2011) foram executados em um servidor Intel Xeon E5620 de 2.40 GHz, com 8 GB de memória RAM, rodando o sistema operacional Rocks Cluster Linux.

5.1 Resultados

O Algoritmo implementado no trabalho é mais sensível à densidade da instância, obtendo melhores resultados à medida que a densidade aumenta. Já na implementação do *Ant-Line*, sem busca local, de Mulati & Constantino (2011), os resultados obtidos são mais homogêneos, independentemente da instância. A principal diferença do trabalho proposto em relação ao trabalho de Mulati & Constantino (2011) é que algumas instâncias sofrem um pré-processamento que elimina colunas redundantes antes da execução. O depósito de feromônio é feito de forma que, inicialmente, a melhor solução da iteração reforça mais resíduos que a melhor solução global e isso muda gradualmente

com base no número de iterações do algoritmo.

Em geral, os resultados ficaram, aproximadamente, 6% distantes em relação ao ótimo, enquanto que em Mulati & Constantino (2011) essa diferença não chega nem a 1%. O tempo de execução foi em média 10 vezes maior comparado ao de Mulati & Constantino (2011).

6. Conclusão

Com base nos resultados obtidos nos testes do algoritmo *Ant-Line* implementado neste trabalho, pode-se concluir que o pré-processamento das instâncias e a troca de valores entre s' e s^* , como feito em Mulati & Constantino (2011), apresentam impactos significativos em relação à qualidade da solução obtida. As soluções deste trabalho ficaram aproximadamente 6% distantes da solução ótima e no trabalho de Mulati & Constantino (2011), sem busca local, os resultados ficaram aproximadamente 1% distantes da solução ótima.

O número de iterações do algoritmo foi outro ponto interessante, já que, com 75 iterações, este trabalho, na maioria das vezes, não alcança a melhor solução de uma instância, mas em Mulati & Constantino (2011) a solução ótima é alcançada, na maioria das vezes, em menos de 75 iterações.

O algoritmo *Ant-Line* apresenta um bom desempenho para o PCC, justamente pela sua característica de escolha de linhas, o que evita que todas as colunas sejam examinadas a cada passo da seleção de coluna, diminuindo, portanto, o número de comparações necessárias. Como trabalhos futuros, sugere-se a implementação do *Ant-Line* com busca local e a exploração de novas vertentes da meta-heurística ACO.

7. Referências

DORIGO, M; STUTZLE, T. *Ant Colony Optimization*. Cambridge, Massachusetts: Mit Press, 2004.

LESSING, L; DUMITRESCU, I; STUTZLE, T. *A Comparison Between ACO Algorithms for the Set Covering Problem*. In: A BUCHMANN. *Lecture Notes in Computer Science*. Berlim: Springer, 2004.

DENEUBOURG *et al.* *The self-organizing exploratory pattern of the Argentine ant*. *Journal of Insect Behavior*, 3, 159–168, 1990.

MULATI, M. H. & CONSTANTINO, A. A. *Ant-Line: A Line-Oriented ACO Algorithm for the Set Covering Problem*. In: XXX International Conference of the Chilean Computer Science Society, 2011.

R. HADJI, M. RAHOUAL, E. TALBI, & V. BACHELET, *Ant Colonies for The Set Covering Problem*. In: *Abstract proceedings of ANTS2000–From Ant Colonies to Artificial Ants: A Series of International Workshops on Ant Algorithms*, 2000.

OLIVEIRA, N V de. *Problema De Cobertura De Conjuntos – Uma Comparação*

Numérica De Algoritmos Heurísticos. 1999. Dissertação (Mestre) - Universidade Federal De Santa Catarina, Florianópolis, 1999.

REN, Z et al. A Fast and Efficient Ant Colony Optimization Approach for The Set Covering Problem. In: *EVOLUTIONARY COMPUTATION CEC*, Hong Kong, China. IEEE World Congress on Computational Intelligence. 2008.

Algoritmo Minimax Aplicado ao Jogo de Damas

Guilherme L. de Mello¹, Mauro Miazaki¹, Andres Jessé Porfirio²

¹Universidade Estadual do Centro-Oeste (UNICENTRO) – Campus CEDETEG
Rua Simeão Camargo Varela de Sá, 3 – CEP 85.040-080 – Guarapuava – PR – Brasil

²Universidade Tecnológica Federal do Paraná
Rua Padre Salvatore Rena, 875 – Guarapuava – PR – Brasil

guilhermelmello@yahoo.com.br

Resumo: *A competitividade sempre existiu entre os homens e com relação aos computadores não é diferente, produzindo máquinas (programas) que pudessem vencer outras em uma tarefa. Neste contexto, surgiu a Inteligência Artificial, que procura examinar e definir o que é um comportamento inteligente. Este resumo apresenta o estudo do comportamento de uma máquina em um jogo de damas e um programa jogador baseado em um algoritmo conhecido como Minimax. O desempenho do programa jogador foi analisado durante a busca pelo próximo movimento a ser realizado.*

Palavras-chave: *Inteligência Artificial; busca; jogos; alfa-beta.*

1. Introdução

Há milhares de anos tentamos entender como pensamos, ou seja, como somos capazes de perceber, compreender e prever um mundo maior que nós. O campo da Inteligência Artificial (IA) vai muito além disso, pois tenta transportar o comportamento humano para uma máquina. Rich e Knight (1993) definem IA como “(...) o estudo de como fazer os computadores realizarem coisas que, no momento, as pessoas fazem melhor.”, porém, esta definição é um pouco específica, devido à sua abrangência.

No início, os trabalhos eram focados na resolução de tarefas formais. Dentre eles estavam os jogos, os quais sempre se mostraram efetivos para a experimentação de IA, devido às suas características, como regras bem definidas, permitindo uma fácil medição entre sucesso e fracasso, além dos aspectos presentes em um comportamento inteligente (estratégia, conhecimento, memória, etc.) (BITTENCOURT, 1998).

Problemas como jogos são comumente resolvidos por uma busca entre estados, considerando as jogadas de todos os jogadores. Porém, há casos muito extensos em que são utilizadas heurísticas para guiar a busca (RICH, KNIGHT; 1993). Com isso, muitos estudos têm se concentrado nas funções de avaliação, tornando a análise mais precisa.

Neste resumo, a aplicação do algoritmo Minimax é demonstrada no jogo de damas, sendo este um algoritmo de busca com profundidade limitada, que, baseada em uma heurística, procura maximizar o ganho de um jogador para uma jogada, utilizando uma árvore de jogo. Como complemento, foi utilizada a técnica de cortes Alfa-Beta, permitindo eliminar algumas ramificações da árvore e melhorando a eficiência da busca.

2. Jogo de Damas

O jogo de damas é muito popular no mundo inteiro, com mais de 150 variações

documentadas, mas apenas duas versões possuem comunidades internacionais (SHAEFFER, LAKE; 1996). Na computação, o jogo de damas teve início com Arthur Samuel (RICH, KNIGHT; 1993), e desde então tem sido muito estudado (RUSSEL, NORVIG; 2004). Para a implementação do programa jogador foram utilizadas as regras praticadas em território brasileiro, disponíveis no site da Confederação Brasileira do Jogo de Damas¹, as quais são uma mescla entre as duas versões das comunidades internacionais.

3. Estratégia Minimax e Podas Alfa-Beta

Em sistemas multiagentes, um agente precisa considerar o máximo de ações possíveis, pois estas afetam o seu bem-estar e, desta forma, maximizam seu desempenho. O jogo de damas é considerado um sistema multiagentes competitivo, no qual as ações são revezadas em um jogo de soma zero com informação perfeita (RUSSEL, NORVIG; 2004). Assim, são considerados dois jogadores (MAX e MIN), cujos movimentos são revezados e iniciados por MAX. Ao final do jogo, uma pontuação é atribuída para cada um deles, definindo um vencedor.

A configuração atual do jogo e as possíveis jogadas seguintes formam a árvore de jogo, e os níveis da árvore representam as jogadas para um jogador. Ao analisar os nós das folhas, atribui-se alguns valores (em função de MAX), definidos por uma função de avaliação, e cabe ao jogador MAX usar estes valores para definir sua jogada e estratégia. Considerando que ambos os jogadores são infalíveis, cada um irá escolher a jogada que tenha o melhor valor para ele.

O algoritmo Minimax é um procedimento de busca em profundidade limitada (LUGER, 2004). O processamento é iniciado com o estado atual do jogo e a árvore de busca é construída com base em um gerador de movimentos. Após gerar a árvore, os nós das folhas são avaliados com um valor definido pela função de avaliação, a qual é baseada em uma heurística. O valor do melhor estado é, então, retornado ao estado anterior, e este valor é baseado no jogador responsável pela jogada. Caso o jogador MAX possua a jogada, o maior valor é retornado, pois seu objetivo é maximizar sua medida de desempenho. Caso contrário, se o jogador MIN possuir a jogada, o menor valor é retornado, pois seu objetivo é minimizar a medida de desempenho de seu oponente.

Como o algoritmo Minimax faz a análise de todas as jogadas possíveis em uma árvore de jogo, este acaba se tornando muito custoso em casos mais extensos. Com isso, surge uma alternativa que procura diminuir o número de jogadas analisadas, conhecida como podas (ou cortes) alfa-beta.

Os cortes Alfa-Beta são uma variação do método Minimax, o qual permite eliminar algumas ramificações, assim que estas são descritas como ruins, e quando aquele caminho da árvore de jogo nunca será atingido em casos reais. O nome Alfa-Beta é atribuído a este método devido a dois índices acrescentados ao Minimax, estes índices são o alfa e o beta. O índice Alfa refere-se ao melhor (maior) valor encontrado pelo caminho analisado para uma jogada maximizante. O Beta refere-se ao melhor (menor) valor encontrado pelo caminho analisado para uma jogada minimizante.

¹ http://www.codexdamas.com.br/regras_oficiais.html.

4. Heurística

Para executar o algoritmo Minimax, é necessária a aplicação de uma heurística aos nós terminais da árvore de jogo. Neste trabalho, uma heurística simples é utilizada, assim como uma variação desta. Quando a análise do tabuleiro é realizada pela função de utilidade, são considerados alguns aspectos, como a disposição e o tipo das peças presentes no tabuleiro.

As peças e as damas possuem pesos diferentes (2 e 5 respectivamente), os quais são multiplicados pelo peso da casa em que estão. A Figura 1 descreve os pesos das casas utilizadas em cada heurística: posicional² e avanço. A heurística avanço é uma proposta descrita neste resumo. Ao final da análise, a utilidade de todas as peças são somadas. O mesmo é feito para as peças adversárias. Com o valor de utilidade de cada jogador calculado, um coeficiente final é gerado e atribuído ao tabuleiro. O valor final é baseado no valor de utilidade do jogador MAX subtraído do valor de utilidade do jogador MIN. Sendo assim, a vantagem do jogador MAX é representada por valores positivos, e a do jogador MIN por valores negativos.

	4		4		4		4
4		3		3		3	
	3		2		2		4
4		2		1		3	
	3		1		2		4
4		2		2		3	
	3		3		3		4
4		4		4		4	

(a)

	6		6		6		6
6		5		5		5	
	4		3		3		4
4		3		2		3	
	3		2		2		3
3		2		1		2	
	2		1		1		2
2		1		1		1	

(b)

Figura 1: Heurística posicional (a) e de avanço (b).

5. Resultados

Os testes foram divididos em duas etapas, uma em que se analisou a eficiência das heurísticas, e outra em que se analisou o tempo de execução gasto e o número de estados

²<http://www.cos.ufrj.br/~ines/courses/cos740/leila/cos740/damas.doc>

percorridos. Na primeira etapa, foram confrontadas as heurísticas em partidas completas, utilizando-se árvores de jogo com 3 e 5 níveis de profundidade e alterando-se qual é responsável pelo primeiro lance da partida. Como o algoritmo Minimax é determinístico, apenas uma execução foi necessária para cada configuração. Os resultados obtidos nesta primeira etapa mostraram que a heurística posicional (Figura 1 (a)) obteve melhor desempenho, vencendo 3 partidas e empatando 1 (uma) (para árvore com 5 níveis de profundidade).

Na segunda etapa, foram definidas 21 configurações de tabuleiros, variando-se o número de peças para cada jogador. Em cada configuração executou-se a busca para as profundidades de 3, 5 e 7 níveis na árvore de jogo, registrando-se a quantidade de estados percorridos e o tempo de execução para encontrar a próxima jogada. O algoritmo Alfa-Beta mostrou melhor desempenho em relação ao Minimax, pois este analisou um número consideravelmente menor de estados e, conseqüentemente, em menor tempo.

Para 3 níveis de profundidade, o Minimax obteve a média de 154,85 estados analisados em 0,401 segundos, enquanto o Alfa-Beta registrou a média de 80,85 estados analisados em 0,253 segundos. Para 5 níveis de profundidade, as médias atingidas pelo Minimax foram de 4.645,61 estados em 8,81 segundos, enquanto o Alfa-Beta registrou 924,90 estados em 2,07 segundos. Com uma busca em 7 níveis de profundidade, a diferença ficou ainda maior, em que o Minimax obteve a média de 212.954,66 estados em 277,36 segundos, e o algoritmo Alfa-Beta com 8.751,85 estados em 19,26 segundos.

Dentre as configurações de tabuleiros analisadas, duas podem mostrar a diferença entre os algoritmos em casos de ganho e perda. Em uma dessas configurações, para uma árvore de jogo com 7 níveis de profundidade, foram podados 3.128.520 estados, reduzindo-se o tempo de processamento em 3.142,38 segundos. Porém, nem sempre o ganho computacional é garantido pela poda Alfa-Beta, como no caso de outra configuração, para 3 níveis de profundidade, na qual nenhum dos 17 estados analisados foram podados. Sendo assim, o algoritmo Alfa-Beta (0,064 segundos) obteve tempo de processamento ainda maior que o Minimax (0,059 segundos). Mas casos como este ocorrem com pouca frequência, o que mostra o Alfa-Beta como uma boa prática.

6. Conclusões

O Minimax é um algoritmo de busca em profundidade de alto custo computacional, sendo este impraticável para casos extensos e com alto grau de ramificação. A heurística posicional (Figura 1(a)) obteve melhor rendimento em confrontos com a heurística de avanço (Figura 1(b)), pois esta se preocupa mais com a capacidade defensiva, impedindo que o adversário capture suas peças.

Ao comparar os desempenhos obtidos por cada algoritmo (Minimax e Alfa-Beta), o uso de cortes mostrou um ganho considerável, podendo reduzir um número alto de estados analisados. Porém, esse ganho não é garantido em raras exceções.

Referências

BITTENCOURT, G. **Inteligência Artificial: Ferramentas e Teorias**. Florianópolis: Ed. da Ufsc, 1998.

LUGER, George F. **Inteligência Artificial: Estruturas e estratégias para a solução de**

problemas complexos. 4. ed. Porto Alegre: Bookman, 2004.

RICH, Elaine; KNIGHT, Kevin. **Inteligência Artificial**. 2. ed. São Paulo: Makron Books, 1993.

RUSSEL, S.; NORVIG, P. **Inteligência Artificial**. Rio de Janeiro: Elsevier, 2004.

SAMUEL, A. L. **Some studies in machine learning using the game of checkers**, IBM J. Res. Devel. 3 (1959), 210–229.

SHAEFFER, Jonathan; LAKE, Robert. **Solving the game of checkers**. In: NOWAKOWSKI, Richard. Games of No Chance. Berkeley, Usa: Msri Books, 1996. p. 119-133.

Prototipação de um *e-commerce* para Pizzarias com disponibilização de dados semânticos

Marcelo Araújo, Sandro Rautenberg (Orientador)

Departamento de Ciência da Computação
Universidade Estadual do Centro-Oeste (UNICENTRO) – Guarapuava, PR – Brasil

marcelonucleo@yahoo.com.br

Resumo. *Este artigo descreve conceitos e ferramentas envolvidos na Web Semântica e aplicados a um protótipo de e-commerce para pizzarias. Dados semânticos são obtidos na DBpedia para enriquecer a interação de usuários no contexto de compra de um produto.*

Palavras-chave: *Web Semântica; Linked Open Data; Dpedia.*

1. Introdução

No início da *World Wide Web*, segundo (BREITMAN, 2005), “as páginas da Internet eram desenvolvidas por programadores de *software*. Como essas páginas ofereciam uma maneira simples de compartilhar informações, logo tornaram-se populares entre programadores e engenheiros de *software* do mundo todo”. Desde então, a chamada Web Sintática utiliza os computadores apenas para fazer a apresentação de informações, que são essencialmente direcionadas a seres humanos, cabendo a estes a tarefa de interpretação (BREITMAN, 2005).

Assim, os computadores não são capazes de interpretar, de forma eficiente, a informação disponível na Web Sintática. Para que essa incapacidade de interpretação possa ser contornada, é necessário realizar a estruturação dos dados disponibilizados. Dessa necessidade surge o conceito de Web Semântica, o qual visa transformar a Web Sintática ao adicionar semântica às suas páginas (BREITMAN, 2005).

A Web Semântica, segundo Berners-Lee (2001), “é uma Web de dados, e estes podem ser processados direta ou indiretamente por máquinas”. Esses dados devem ser enriquecidos de significado, proporcionando ao mesmo tempo condições para que os computadores possam interpretá-los de forma automatizada e que sejam de fácil acesso aos seres humanos (PASSIN, 2004).

Para que os dados existentes na Web possam ser interpretados e processados por computadores, o W3C¹ ajuda no desenvolvimento de tecnologias que dão suporte à Web Semântica. Nesse contexto, tem-se a definição de linguagens de representação (como o *Resource Definition Framework* – RDF), linguagens de consulta (como o *SPARQL Protocol and RDF Query Language* – SPARQL) e um conjunto de boas práticas, como segue (BERNERS-LEE, 2013):

- usar URIs² como nomes para as coisas;
- usar URIs e HTTP³ para que as pessoas possam procurar esses nomes;
- quando alguém procurar uma URI, prover informação útil, usando os padrões

¹<http://www.w3.org/>

²*Uniform Resource Identifier*

³*Hypertext Transfer Protocol*

(RDF, SPARQL);

- incluir *links* para outras URIs, de modo que se permita a descoberta de mais coisas.

Dentre as fontes de dados abertos no formato RDF, destaca-se a DBpedia, como “um esforço colaborativo para extrair informações estruturadas da Wikipédia, tornando estas informações disponíveis na Web” (DBPEDIA, 2013).

O objetivo deste trabalho é investigar a utilização de dados semânticos obtidos na DBpedia para enriquecer a interação de usuários em um *e-commerce*, no contexto de compra de um produto.

2. Materiais e métodos

O trabalho se dá na forma de um estudo de caso sobre um protótipo de *e-commerce* que foi desenvolvido com a utilização das seguintes tecnologias: as linguagens de programação Javascript e PHP⁴, em conjunto com a biblioteca jQuery, e o *framework* Zend Framework, respectivamente; a linguagem de consulta SPARQL e RDF, para manipulação de dados semânticos; as linguagens de marcação HTML5 e CSS3, esta última utilizando o *framework* Twitter Bootstrap, para construção do *front-end*. O Banco de Dados baseado em grafos Neo4j foi utilizado para persistir os dados, além da API Neo4jPHP, visando facilitar a utilização da linguagem de consulta CYPHER para manipular os dados através do PHP.

Os dados semânticos utilizados pelo protótipo encontram-se no formato RDF, o qual é um modelo de dados que possibilita armazenar dados estruturados em arquivos. Nesse modelo, informações são expressas em forma de triplas. Ou seja, em um formato dividido em três partes, sendo elas: sujeito, predicado e objeto. Cada tripla pode ser vista como a descrição de uma coisa (sujeito), que possui uma propriedade (predicado), e um valor para tal propriedade (objeto).

Também foi utilizada a linguagem de consulta SPARQL, a qual é capaz de extrair e manipular dados armazenados no formato RDF. Esta é considerada como uma das tecnologias chave da Web Semântica (SPARQL, 2013), sendo a melhor alternativa para obter dados semânticos para utilização nas mais diversas aplicações (DUCHARME, 2011).

O principal objetivo do *e-commerce* para pizzarias, além de possibilitar que seus usuários realizem o pedido da pizza *on-line*, é proporcionar o enriquecimento de contexto utilizando dados semânticos em algumas fases do processo.

Os dados semânticos são utilizados durante a fase de escolha do sabor da pizza. Nessa fase estão disponíveis ao usuário informações sobre cada ingrediente que compõe um sabor de pizza.

As informações atreladas a cada ingrediente consistem em:

- uma breve definição;
- as grafias do ingrediente em outros idiomas disponíveis;
- um arquivo de áudio que contém a pronuncia num idioma selecionado.

⁴PHP *Hypertext Preprocessor*

3. Resultados

Como resultado, este trabalho tem um protótipo de *e-commerce* que utiliza dados semânticos em determinadas fases do processo de compra.

Ao entrar no *site*, o usuário visualizará uma página inicial que oferece a ele a opção de realizar um pedido. Após selecionar tal opção, o usuário é redirecionado para uma página que exibe uma lista com todas as pizzarias disponíveis no sistema para que se possa escolher a de sua preferência. A página seguinte lista todos os produtos (pizzas) que aquela pizzaria oferece e permite ao usuário escolher entre os possíveis tamanhos e eventuais quantidades de sabores para cada um. Após a escolha do produto, são exibidos os sabores disponíveis para aquele produto, onde cada sabor terá uma lista com os ingredientes que o compõem atrelada.

Quando o usuário clicar sobre o nome de um ingrediente, será utilizada a técnica AJAX, em que a requisição feita é do tipo *XMLHttpRequest*. Quando a requisição é recebida pelo servidor, é feita uma consulta no Banco de Dados, e, em seguida, recupera-se o valor da propriedade *link* correspondente ao registro retornado. Tal valor é utilizado para realizar uma consulta na DBpedia via SPARQL. A resposta dessa consulta é retornada em formato JSON, a partir do qual são extraídas as informações a serem utilizadas. Por fim, o servidor retorna a resposta ao lado para o cliente, no formato JSON, com tais informações. Esta resposta será processada pelo Javascript, o qual manipulará o HTML da página e exibirá as informações ao usuário.

Em conjunto com as informações supracitadas, é oferecida ao usuário a opção de visualizar a grafia do ingrediente selecionado em outros idiomas. Escolhendo essa opção, a técnica AJAX será novamente utilizada, e o arquivo JSON retornado contém todas as grafias disponíveis na DBpedia para aquele ingrediente, sendo cada uma associada à sigla ISO referente ao seu idioma, e a um *link* que, quando acionado, faz novamente uma requisição, utilizando a técnica AJAX, em que o servidor do sistema fará uma requisição ao servidor do Google TTS⁵, o qual retorna um arquivo de áudio. O conteúdo deste arquivo é a pronúncia daquela grafia relativa ao código ISO vinculado.

4. Conclusão

Este artigo apresentou um protótipo de *e-commerce* para pizzarias com a disponibilização de dados semânticos. Para realização do trabalho foi necessário o estudo de conceitos de Engenharia de *Software*, de diferentes linguagens e paradigmas de programação, dos protocolos e técnicas de comunicação, do paradigma de Banco de Dados baseado em Grafos, e dos conceitos inerentes à Web Semântica. Na forma de estudo de caso, dados semânticos são obtidos a partir da DBpedia, enriquecendo a interação dos usuários no contexto de realização de um pedido de pizza *on-line*.

Com o estudo realizado, ficou evidenciado que utilizar dados obtidos na Web Semântica pode proporcionar contextos mais ricos aos usuários de *e-commerce*. Tal fato permite ao usuário descobrir novas informações, o que também poderia ser feito, inicialmente, sem a utilização da Web Semântica. Porém, isso demandaria desenvolvimento adicional para tratar previamente os dados, uma vez que esses precisariam ser armazenados pela própria aplicação, gerando mais uma desvantagem, a falta de dinamismo das informações.

⁵Google *Text-to-Speech*

Portanto, conclui-se que os fundamentos da Web Semântica são importantes para criar aplicações cada vez mais ricas e dinâmicas, oferecendo aos usuários, além de simples serviços, o conhecimento.

Para trabalhos futuros, recomenda-se a realização de um estudo com potenciais usuários do protótipo para avaliar as funcionalidades implementadas, a interação humano computador, a utilidade da disponibilização de dados semânticos e o levantamento de novos requisitos, com a finalidade do desenvolvimento comercial. Ainda é recomendada a utilização de dados semânticos oriundos de outras fontes do projeto LOD, para georreferenciamento das pizzarias disponíveis no sistema, visando oferecer a possibilidade de buscas baseadas em localização geográfica.

Referências Bibliográficas

- [BERNERS-LEE, 2001] BERNERS-LEE, Tim; HENDLER, James; LASSILA, Ora. **The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities**. *Scientific American Magazine*, [s. L.], n. , p.34-43, maio 2001.
- [BERNERS-LEE, 2013] BERNERS-LEE, Tim. **Linked Data - Design Issues**. Disponível em: <<http://www.w3.org/DesignIssues/LinkedData.html>>. Acesso em: 27 fev. 2013.
- [BREITMAN, 2005] BREITMAN, Karin Koogan. **Web semântica: a internet do futuro**. Rio de Janeiro: LTC, 2005. 190 p.
- [DBPEDIA, 2013] DBPEDIA. **DBpedia About**. Disponível em: <<http://dbpedia.org/>>. Acesso em: 15 mar. 2013.
- [DUCHARME, 2011] DUCHARME, Bob. **Learning SPARQL**. Sebastopol, Ca: O'reilly Media, 2011. 258 p.
- [PASSIN, 2004] PASSIN, Thomas B. **Explorer's Guide to the Semantic Web**. Greenwich : Manning , 2004. 304 p.
- [SPARQL, 2013] **SPARQL** Disponível em: <<http://en.wikipedia.org/wiki/SPARQL>>. Acesso em: 15 fev. 2013.

Evolução Diferencial Adaptativa para o Problema do Despacho Econômico de Energia Elétrica

Lucas Prestes¹, Ricardo H. R. Lima¹, Josiel N. Kuk¹, Carolina P. Almeida¹, Sandra M. G. S. Venske¹, Richard A. Gonçalves¹

¹Departamento de Ciência da Computação – Universidade Estadual do Centro-Oeste (UNICENTRO) Guarapuava – PR – Brasil

lucas.prestes.lp@gmail.com, ricardo_des@hotmail.com

Resumo. A produção eficiente de energia elétrica é importante para o dia-a-dia da sociedade moderna; portanto, o problema do despacho econômico de energia, o qual visa gerar a energia demandada com o menor custo possível, é de grande importância. Porém, esse problema não pode ser resolvido de maneira exata, necessitando de técnicas alternativas – tais como as meta-heurísticas. Uma das meta-heurísticas mais bem-sucedidas na atualidade é a Evolução Diferencial. A Evolução Diferencial pode ser aprimorada através do uso de técnicas de adaptação. Portanto, neste trabalho, propõe-se duas técnicas de Evolução Diferencial Adaptativa para a resolução do Problema do Despacho Econômico de Energia Elétrica. Os resultados obtidos foram satisfatórios e encorajam trabalhos futuros relacionados ao tema.

1. Introdução

Atualmente, a sociedade é extremamente dependente da energia elétrica no seu dia-a-dia. Portanto, a geração eficiente de energia tem impacto na vida de toda a população. Neste sentido, o despacho econômico de energia elétrica, que visa gerar a energia demandada pela sociedade ao menor custo possível, é um importante problema a ser resolvido no planejamento de um sistema de geração de energia elétrica (GONÇALVES, 2010). Contudo, esse problema não pode ser resolvido de maneira exata, o que faz com que o uso de meta-heurísticas torne-se necessário. Uma meta-heurística de bastante sucesso na atualidade é a Evolução Diferencial (ED). A ED é uma técnica de Computação Evolucionária cujo principal operador se baseia nas diferenças entre os valores das soluções da população. Apesar de eficiente, a ED pode ser melhorada através do uso de técnicas de adaptação de parâmetros. Esse trabalho propõe, portanto, o uso da Evolução Diferencial Adaptativa para a solução do despacho econômico de energia elétrica.

2. Despacho Econômico de Energia Elétrica

O Problema do Despacho Econômico de Energia Elétrica (PDEEE) pode ser formulado como um problema de minimização de custos. Formalmente, tem-se:

$$\text{Minimizar } F = \sum_{j=1}^n F_j(P_j) \quad \text{sujeito a } PD = \sum_{j=1}^n P_j \quad \text{e } p_j^{\min} \leq P_j \leq P_j^{\max}, j=1, \dots, n,$$

em que: $F_j(P_j)$ é a função que representa o custo do j -ésimo gerador; P_j é a potência de saída do j -ésimo gerador; n é a quantidade de geradores; PD é a demanda que precisa

ser atingida pelo sistema; enquanto P_j^{min} e P_j^{max} são os limitantes mínimos e máximos do j-ésimo gerador, respectivamente.

O custo de cada gerador é obtido a partir da fórmula a seguir, sem a consideração dos efeitos de ponto de válvula, o que resulta em uma aproximação quadrática:

$$F_j(P_j) = a_j * P_j^2 + b_j * P_j + c_j,$$

na qual: P_j é a saída do j-ésimo gerador; a_j , b_j e c_j são coeficientes que determinam o custo de combustível para o j-ésimo gerador. Caso sejam considerados os efeitos de ponto de válvula, os resultados então mostram um certo grau de não-linearidade maior, e a fórmula que representa isso é dada como:

$$F_j(P_j) = a_j * P_j^2 + b_j * P_j + c_j + |e_j * \sin(f_j * (P_j^{min} - P_j))|,$$

em que: P_j é a saída do j-ésimo gerador; a_j , b_j , c_j , e_j e f_j são os coeficientes da função de custo do j-ésimo gerador, quando se considera os efeitos do ponto de válvula (GONÇALVES, 2010).

3. Evolução Diferencial

A Evolução Diferencial é uma técnica de Computação Evolucionária e, como tal, é composta de uma população de soluções e de um conjunto de operadores, que são utilizados para modificar as soluções. A população inicial é gerada de maneira aleatória. As populações seguintes são geradas pela aplicação dos operadores de mutação, cruzamento e seleção (STORN; PRICE, 1995). Na mutação é calculada a diferença entre dois vetores aleatórios da população, adicionada a um terceiro vetor e multiplicado por um fator de mutação (M), ou seja, M controla a intensidade da mutação (quanto maior o valor de M mais intensa é a mutação). O cruzamento escolhe partes do vetor resultante da mutação e partes de um quarto indivíduo da população para formar um novo indivíduo, de acordo com o fator cruzamento (CR), ou seja, CR é a probabilidade de escolher um componente do vetor resultante da mutação e $1 - CR$ é a probabilidade de escolher componentes do quarto indivíduo. Na seleção, esse indivíduo resultante do cruzamento é comparado ao indivíduo original da população e, caso seja melhor, ele passa a fazer parte da população em substituição ao original.

O ajuste dos parâmetros M e CR é uma tarefa custosa e é difícil garantir que os valores adotados são os valores ótimos para esses parâmetros. Nesse sentido, nos últimos anos tem-se observado na literatura da área um crescimento do uso de algoritmos de Evolução Diferencial Adaptativos, ou seja, algoritmos que ajustam automaticamente os parâmetros da ED. Neste trabalho são adotadas duas formas de adaptação: SaDE e CoDE.

No SaDE os fatores de mutação (M) e cruzamento (CR) são incluídos na estrutura de cada indivíduo. Esses valores são, inicialmente, aleatórios. Ao gerar um novo indivíduo, os fatores M e CR podem ser modificados. Essa modificação também é aleatória (BREST et al., 2013).

O algoritmo CoDE (WANG; CAI; ZHANG, 2010) é uma adaptação da implementação da DE normal, com a diferença de que são usadas três estratégias diferentes para a mutação dos cromossomos e que os parâmetros CR e M variam ao

longo do processo evolutivo. Para o problema do despacho econômico de energia elétrica, as estratégias escolhidas foram: “rand/1/bin”, que usa de seleção aleatória de 1 par de vetores para mutação e cruzamento binário; “rand/2/bin”, que usa dois pares de vetores aleatórios para mutação e o cruzamento também binário; e “current-to-rand/1”, que se assemelha ao “rand/1/bin”, mas não realiza cruzamento. E os valores para os parâmetros CR e M escolhidos foram os seguintes pares: [M=1.0, CR=0.1], [M=1.0, CR=0.9], e [M=0.8, CR=0.2]. Outros pares de valores poderiam ser adotadas, mas esses valores foram escolhidos por serem adequados para resolver problemas com diferentes características (WANG; CAI; ZHANG, 2010). No CoDE, a cada geração três estratégias são usadas na mutação dos indivíduos. Os parâmetros M e CR são escolhidos aleatoriamente para cada mutação. A seleção também é alterada: escolhe-se o melhor indivíduo entre aqueles resultantes das estratégias de mutação e esse indivíduo é comparado com o indivíduo original da população; e caso o indivíduo da mutação seja melhor, ele substitui o indivíduo original na população (WANG; CAI; ZHANG, 2010).

4. Resultados

Foram realizados 50 testes com as instâncias de 13 geradores e demanda de 1800 MW, 13 geradores e demanda de 2520 MW e 40 geradores com demanda de 10500 MW, com população (NP) de tamanho igual ao número de geradores multiplicado por 10; e 1000 gerações para os algoritmos adaptativos; 1500 para o algoritmo sem adaptação. Os resultados foram comparados aos valores obtidos por testes da técnica de Sistema Imunológico (SI) feitas por Gonçalves (2010). As máquinas usadas para os testes foram: uma Intel Dual Core, com 2.30GHz e 3GB de RAM, e uma Intel Core i5, com 2.30GHz e 4GB de RAM. Nas Tabelas 1, 2 e 3 temos que Melhor, Média e Pior correspondem, respectivamente, ao melhor valor, ao valor médio e ao pior valor encontrado durante as 50 execuções dos algoritmos.

Tabela 1. Teste para 13 geradores com demanda de 1800

	Melhor	Média	Pior
Não-adaptativo	18112,15	18155,63	18194,17
CoDE	18036,59	18077,31	18122,08
SaDE	18124,69	18213,95	18213,95
SI	17960,36	17969,37	17976,5

Na tabela 1, para os testes dos 13 geradores com demanda de 1800, os melhores resultados – aqueles mais próximos dos resultados encontrados por Gonçalves (2010) – vem da mesma técnica SI, sendo o segundo melhor resultado obtido com o CoDE. Para a média e o pior caso a técnica SI se mantém a melhor.

Tabela 2. Teste para 13 geradores com demanda de 2520

	Melhor	Média	Pior
Não-adaptativo	24161,49	24167,62	24174,78

CoDE	24163,15	24166,95	24175,56
SaDE	24164,11	24164,69	24166,46
SI	24164,8	24233,22	24427,29

Na tabela 2, o melhor resultado foi obtido com a versão não adaptativa da ED em comparação com a técnica adaptativa SI, enquanto a melhor média e o pior caso são obtidos com a técnica do SaDE.

Tabela 3. Teste para 13 geradores com demanda de 10500

	Melhor	Média	Pior
Não-adaptativo	122217,21	122362,43	122468,96
CoDE	122099,31	122245,13	122343,67
SaDE	121802,69	121942,72	122111,16
SI	121500,43	121880,13	122283,55

Na tabela 3, a técnica de SI mostrou melhor desempenho para o melhor caso e a média, enquanto a técnica SaDE mostrou-se melhor no pior caso.

5. Conclusões

O uso da DE para a resolução do problema do despacho econômico de energia elétrica mostrou bons resultados, principalmente quando utilizada em conjunto com técnicas adaptativas. Os métodos adaptativos mostraram-se mais eficientes que o método não adaptativo e, apesar de bem diferentes, apresentaram resultados semelhantes e satisfatórios, se comparados ao algoritmo com a técnica SI.

Trabalhos futuros pretendem investigar o impacto do uso de outras técnicas adaptativas.

Referências

- STORN, Rainer; PRICE, Kenneth. **Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces**. 1995. 12 p.
- GONÇALVES, Richard Aderbal. **Algoritmos Culturais para o Problema do Despacho de Energia Elétrica**. 2010. 205 f. Tese (Doutorado) - Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade de Tecnologia Federal do Paraná, Curitiba, 2010.
- BREST, Janez et al. **Real Parameter Single Objective Optimization using Self-Adaptive Differential Evolution Algorithm with more Strategies**. In: IEEE CONGRESS ON EVOLUTIONARY COMPUTATION, Cancún, 2013.
- WANG, Yong; CAI, Zixing; ZHANG, Qingfu. **Differential Evolution with Composite Trial Vectors Generation Strategies and Control Parameters**. *IEEE Transactions on Evolutionary Computation*, 2010. 11 p.

Um Estudo da Aplicação de Redes Neurais Artificiais *Multilayer Perceptron* no Prognóstico de Câncer de Próstata

Victor Alexandre Padilha, Sandro Rautenberg

Departamento de Ciência da Computação – Universidade Estadual do Centro-Oeste
(UNICENTRO) – Guarapuava – PR – Brasil

padilha@linuxmail.org, srautenberg@unicentro.br

Resumo. *Este trabalho apresenta uma aplicação das Redes Neurais Artificiais Multilayer Perceptron no prognóstico de câncer de próstata. Como estratégia de treinamento utilizou-se o algoritmo backpropagation com momentum e a técnica k-fold cross-validation. Para a definição da melhor configuração, realizou-se testes em dados artificialmente produzidos, oriundos de um Sistema Genético-Fuzzy. No teste da melhor configuração de Rede Neural encontrada, duas bases com dados reais e distintas foram utilizadas. Comparando-se os resultados alcançados com um benchmark da literatura, observou-se que não houve melhoras.*

1. Introdução

O câncer é uma multiplicação desregulada de células com a consequência de um aumento no número de células anormais em órgãos específicos. Este tipo de anomalia pode ocorrer em qualquer parte do corpo humano, porém, é mais comum na pele, próstata e mama (Schwab, 2008). O câncer de próstata pode ser definido como uma neoplasia maligna que surge na glândula prostática (Theodorescu; Ehdaie, 2008).

Segundo o Instituto Nacional do Câncer, no Brasil o câncer de próstata é o segundo tipo de câncer mais comum entre homens. Em valores absolutos, é o sexto tipo mais comum no mundo e o mais prevalente na população masculina, representando cerca de 10% do total de cânceres (INCA, 2013). A determinação do grau de confinamento do câncer de próstata é um fator determinante na análise do benefício da cirurgia para a remoção do tumor, pois os riscos da cirurgia somente se justificam no caso do confinamento do tumor ao órgão (Tosatti *et al*, 2008).

O objetivo deste trabalho é investigar a aplicação das Redes Neurais Artificiais (RNAs) *Multilayer Perceptron* (MLPs) no auxílio do prognóstico do câncer de próstata, visando melhorar resultados alcançados ao se aplicar Sistemas Genético-*Fuzzy* (Castanho *et al*, 2013).

2. Materiais e Métodos

Foram utilizadas duas bases de dados reais de pacientes submetidos à prostatectomia radical no Hospital das Clínicas da UNICAMP-SP para as atividades de verificação e de validação das RNAs. As bases são compostas por: 287 casos confinados e 35 não confinados para verificação; e 34 casos confinados e 14 não confinados para validação. Ao estadiamento são associadas três variáveis cirúrgicas: exame digital retal (DRE) (Srougi, 1999); dosagem de PSA sérico (Srougi, 1999); e escore de Gleason (Furihata; Tamotsu, 2008).

No treinamento das RNAs foram utilizados dados artificiais com cerca de 38.000 prognósticos válidos (19.000 casos com câncer confinado e 19.000 casos com câncer não confinado), produzidos por um Sistema Genético-*Fuzzy* (Castanho *et al*, 2013). Estes dados são denominados como base de treinamento e permitiram a utilização da técnica *k-fold cross-validation* (Kohavi, 1995).

Para medir o desempenho das RNAs, considera-se uma medida de *fitness*, obtida a partir da soma de outras duas medidas: sensibilidade (S), que mede a probabilidade de uma RNA classificar corretamente um paciente com tumor não confinado; e especificidade (E), que indica a probabilidade de uma RNA classificar corretamente um paciente com câncer confinado ao órgão.

$$\text{fitness} = \frac{TP}{TP+FN} + \frac{TN}{TN+FP} \quad (1)$$

As variáveis supracitadas são evidenciadas na Equação (1), na qual: TP – *True Positive* – é o número de pacientes com câncer não-confinado corretamente classificados; FN – *False Negative* – expressa o número de pacientes com câncer não-confinado classificados como confinado; TN – *True Negative* – contabiliza o número de pacientes com câncer confinado corretamente classificados; e FP – *False Positive* – define o número de pacientes com câncer confinado classificados como não-confinado.

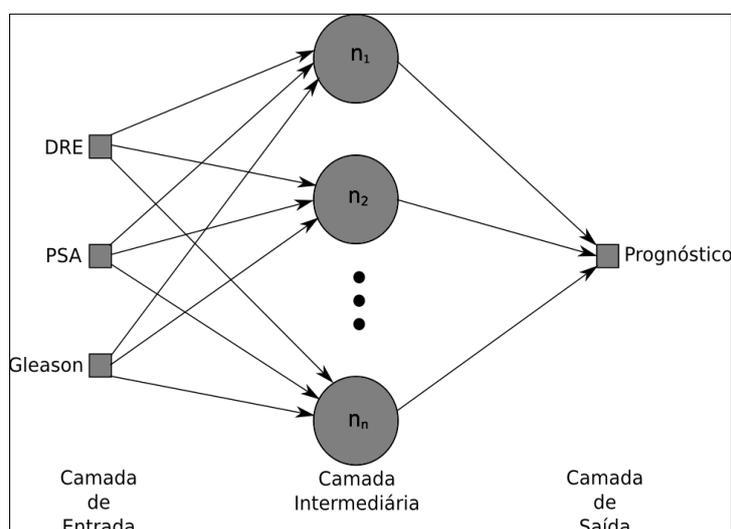


Figura 1. Ilustração gráfica da MLP.

Considerando a abordagem empregada, para a classificação dos dados foram utilizadas as MLPs, que possuem como característica principal o fato de conterem uma ou mais camadas intermediárias de neurônios, que não fazem parte da entrada ou saída. Desse modo, esses neurônios capacitam a RNA para adquirir o conhecimento, progressivamente, baseando-se nos padrões de entrada (Haykin, 1999). A Figura 1 ilustra um modelo genérico de uma MLP com uma camada intermediária para o prognóstico de câncer de próstata. Como algoritmo de treinamento, optou-se pelo *backpropagation* com *momentum*. A implementação do algoritmo é evidenciada em Fausett (1994).

Para efeitos de comparação, utiliza-se como *benchmark* de referência um Sistema Genético-*Fuzzy* para o mesmo problema (Castanho *et al*, 2013). Nesse estudo, alcançou-se o *fitness* médio de 1,6668 na verificação e 1,6386 para a validação.

3. Resultados e Discussão

Considerando-se a técnica de treinamento *k-fold cross-validation*, a base de treinamento foi dividida em dez partições, com 3.800 casos cada, perfazendo dez baterias de treinamento e teste. Assim, para cada configuração da arquitetura de RNA investigada puderam ser aferidas as médias do *fitness*, da sensibilidade e da especificidade. O critério utilizado na determinação de qual configuração obteve o melhor desempenho foi referente àquela com o maior valor de *fitness*.

Durante essa fase, foram utilizados os seguintes parâmetros: 3 neurônios na camada de entrada; 1 camada intermediária; de 2 a 100 neurônios na camada intermediária; 1 neurônio na camada de saída; função de transferência logarítmica sigmoide para todas as camadas; taxa de aprendizado de 0,3; coeficiente de *momentum* de 0,3; e 2.500 épocas de treinamento.

Ao serem executados todos os *k-folds* para todas as Redes Neurais Artificiais, foram aferidos os valores médios de *fitness* para cada configuração. A Figura 2 apresenta o gráfico com as médias de *fitness* dos dez *k-folds* para cada configuração de RNA.

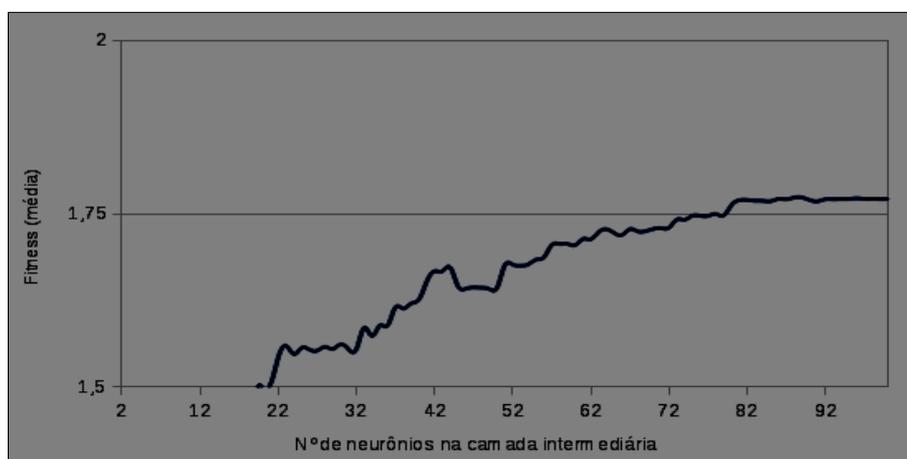


Figura 2. Gráfico dos dez *k-folds* realizados com os Dados de Treinamento.

Ao analisar os dados apresentados na Figura 2, verificou-se que a MLP com o resultado mais significativo possuía 89 neurônios na camada intermediária e atingiu o *fitness* de 1,7735 ($S = 0,9276$ e $E = 0,8459$). Posteriormente, a melhor MLP foi submetida aos testes de verificação e de validação.

Ao realizar a verificação da MLP, obteve-se o *fitness* médio de 1,4481 ($S = 0,7714$ e $E = 0,6767$). Estas medidas estão aquém do *benchmark* de referência, o qual alcançou o *fitness* médio de 1,6668 ($S = 0,7097$ e $E = 0,9571$) para essa fase.

Já no teste de validação, atingiu-se o *fitness* médio de 1,3488 ($S = 0,6429$ e $E = 0,7059$). Estes resultados também não demonstram melhoras em relação ao *benchmark*, que obteve o *fitness* médio de 1,6386 ($S = 0,7857$ e $E = 0,8529$).

4. Conclusão

Este trabalho apresentou o desenvolvimento de MLPs para o prognóstico de câncer de próstata. Para o treinamento das RNAs, utilizou-se uma base de dados artificial gerada a

partir de um Sistema Genético-Fuzzy (Castanho *et al*, 2013). Na fase de testes, por sua vez, utilizou-se duas bases com dados reais, cada uma com, respectivamente, 322 e 48 casos de pacientes submetidos à prostatectomia radical no Hospital das Clínicas da UNICAMP-SP. No treinamento foi empregada a técnica *k-fold cross-validation* para encontrar os melhores parâmetros de configuração da arquitetura da RNA. De posse dos parâmetros de treinamento e testes, a melhor configuração de RNA pôde ser verificada e validada, obtendo um *fitness* de 1,4481 e de 1,3488, respectivamente.

Ao comparar as medidas supracitadas aos resultados alcançados no *benchmark* de referência (*fitness* de verificação de 1,6668 e *fitness* de validação de 1,6386), pode-se concluir que as configurações de MLPs estudadas não são capazes de obter um desempenho superior.

Como trabalhos futuros, pretende-se aplicar outras arquiteturas de RNAs baseadas em clusterização, considerando as mesmas bases de dados e os mesmos *k-folds*, a fim de realizar estudos comparativos.

Referências

- Castanho, M. J. P. *et al*. Fuzzy expert system for predicting pathological stage of prostate cancer. **Expert Systems With Applications**, v. 40, n. 2, p.466-470, 01 fev. 2013.
- Fausett, L. V.. **Fundamentals of Neural Networks: Architectures, Algorithms and Applications**. New Jersey: Prentice Hall, 1994.
- Furihata, M.; Tamotsu, T.. Gleason Grading. In: Schwab, M. (Ed.). **Encyclopedia of Cancer**. Heidelberg: Springer, 2008. p. 1253-1255.
- Haykin, S.. **Neural Networks: A Comprehensive Foundation**. 2. ed. Hamilton: Prentice Hall, 1999.
- INCA – Instituto Nacional do Câncer, Ministério da Saúde, disponível em: <<http://www2.inca.gov.br/wps/wcm/connect/tiposdecancer/site/home/prostata>>. Acesso em: 25 maio 2013.
- Kohavi, R.. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 14., 1995, Montreal. **Proceedings...** . San Francisco: Morgan Kaufmann Publishers Inc., 1995. p. 1137 – 1143.
- Schwab, M.. Cancer. In: Schwab, M. (Ed.). **Encyclopedia of Cancer**. Heidelberg: Springer, 2008. p. 461-463.
- Srougi, M.. Adenocarcinoma da Próstata. In: Barata, H. S.; Carvalho, G. F. (Org.). **Urologia: princípios e prática**. Porto Alegre: Artmed, 1999. p. 356-372.
- Theodorescu, D.; Ehdaie, B.. Prostate Cancer, Clinical Oncology. In: Schwab, M. (Ed.). **Encyclopedia of Cancer**. 2. ed. Heidelberg: Springer, 2008. p. 2445-2452.
- Tosatti, M. A. *et al*. Algoritmo Híbrido Genético-Fuzzy aplicado em Câncer de Próstata. **Hifen**, Uruguaiana, v. 32, n. 62, p.164-171, 2008.

Estudo da Variação de Parâmetros do Algoritmo Genético na Evolução de Criaturas

Willian Eiji Yassue¹, Mauro Miazaki¹, Andres Jessé Porfirio²

¹Universidade Estadual do Centro-Oeste (UNICENTRO) – *Campus* CEDETEG
Rua Simeão Camargo Varela de Sá, 3 – CEP 85.040-080 – Guarapuava – PR – Brasil

²Universidade Tecnológica Federal do Paraná (UTFPR)
Rua Presidente Zacarias de Góes, 875 – CEP 85015-430 – Guarapuava – PR – Brasil

{eiji.yassue, maurom}@gmail.com, andresjesse@yahoo.com.br

Resumo. *As descobertas de Charles Darwin em 1859 revolucionaram várias áreas. Entre elas, a computação foi beneficiada por vários conceitos-chaves para solucionar problemas com a utilização de algoritmos inspirados na teoria da evolução. Neste trabalho, foi aplicado um Algoritmo Genético na evolução do comportamento de criaturas em um ambiente artificial. As criaturas implementadas simulam o comportamento de herbívoros, carnívoros e onívoros. Experimentos foram realizados variando-se os valores dos parâmetros de cruzamento e mutação e os resultados serão apresentados.*

Palavras-chave: *Algoritmo Genético; seleção natural.*

1. Introdução

A Teoria da Seleção Natural proposta pelo biólogo e naturalista Charles Darwin, em 1859, mostra a evolução e concepção da luta pela sobrevivência com dois princípios: a *evolução biológica*, conceito de que todas as formas de vida estão em constante evolução, e a *seleção natural*, que promove a variabilidade entre seus descendentes, permitindo a sobrevivência dos melhores e/ou mais adaptados, promovendo o melhoramento e aprimoramento genético, e a modificação entre as espécies. Os *Algoritmos Evolutivos*, nos quais o *Algoritmo Genético* (AG) enquadra-se, são exemplos de modelos computacionais inspirados no Evolucionismo.

Os Algoritmos Evolutivos são modelos computacionais com processos adaptativos e que propõem um modelo com populações que evoluem para melhorar qualitativamente uma solução em um determinado problema (TANOMARU, 1995). Com grande quantidade de métodos e paradigmas, o Algoritmo Genético foi apresentado por John Holland em seu trabalho *Adaptation in natural and artificial systems*, em 1975, com o objetivo de formalizar matematicamente e explicar os conceitos neo-darwinistas da teoria da evolução, formando um alicerce para pesquisas em diversas aplicações (REEVES, 2013).

Este trabalho tem como objetivo investigar o conceito de AG aplicado na evolução de criaturas. Foram realizados e analisados vários experimentos envolvendo a variação dos parâmetros de cruzamento e mutação para os três tipos de população (herbívoro, carnívoro e onívoro).

2. Métodos

2.1. Algoritmo Genético

Para resolver um determinado problema, um Algoritmo Genético trabalha com uma população, que evolui sob condições dos operadores probabilísticos, representando as possíveis soluções. Na metáfora biológica, há uma tendência dos indivíduos representarem soluções gradativamente melhores à medida que o processo evolutivo progride (TANOMARU, 1995).

Neste trabalho, a cada período de 150 dias (iterações no ambiente simulado), é feita a reprodução para todas as populações. A cada par de criaturas é gerada uma nova criatura, que é adicionada à população (não sobrescreve os pais). Desta forma, serão introduzidas novas criaturas no ambiente. O melhoramento é feito pelo algoritmo genético, que realiza o processo de seleção e a formação de um novo indivíduo com métodos de cruzamento e mutação.

Cada criatura é associada a um tipo (carnívoro, herbívoro ou onívoro), que define o seu comportamento específico relacionado à variedade de alimento que pode consumir. A estrutura do cromossomo é dividida em três partições: controle de navegação, atributos e variação anatômica (Figura 1). A parte do cromossomo responsável pelo controle de navegação representa os movimentos em todos os estados possíveis da criatura. Os atributos representam as características da criatura, como: ataque, defesa, visão e consumo. A partição no cromossomo que define a variação anatômica representa as variações que as criaturas podem adquirir, a partir de operações de mutação e cruzamento, para obter vantagem no ambiente.

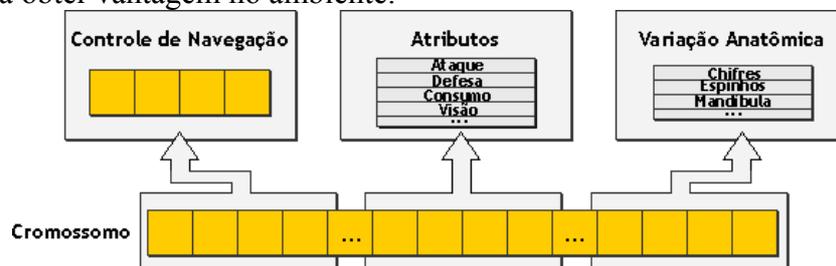


Figura 1: Representação da estrutura do cromossomo da criatura

A operação de cruzamento utilizada neste trabalho é o *crossover* uniforme. Para cada posição do cromossomo é escolhido, aleatoriamente, um *bit* de qualquer um dos pais. Isto significa que cada *bit* é herdado de forma independente de qualquer outro *bit*. A sua configuração ocorre através da especificação de um valor de probabilidade chamado taxa de cruzamento (WHITLEY, 1994).

O operador de mutação é responsável por alterar o material genético do indivíduo, melhorando a diversidade do cromossomo. Neste trabalho é utilizada uma taxa de mutação para configurar seu funcionamento, que é um valor entre 0 e 1, e que define a probabilidade de ocorrer a mutação em um *bit* (GONÇALVES, 2008).

A função de avaliação ou *fitness* é um valor que denomina a qualidade de um indivíduo (solução) no problema. Esse valor determina a seleção dos pais, sendo a forma que diferencia as soluções boas das ruins (LINDEN, 2006). Para este trabalho foi considerada a soma dos atributos de ataque e defesa.

O método utilizado para a seleção dos parceiros foi o método da roleta simples de seleção (em inglês: *roulette wheel*). Essa abordagem utiliza uma roleta fictícia em

que o tamanho da roleta é a soma de todos os *fitness* dos indivíduos e cada indivíduo recebe uma área proporcional ao seu *fitness*. Então, quanto maior é o *fitness*, maior a chance de ser escolhido (HOLLAND, 2013).

3. Resultados

Foram realizados diversos experimentos com o objetivo de buscar a melhor combinação entre os parâmetros do Algoritmo Genético. Para todos os resultados, a seguir, o programa foi executado com os mesmos parâmetros, 100 vezes, para três tipos de população (herbívoro, carnívoro e onívoro). Portanto, os dados são resultantes da média de 100 execuções. Inicialmente, foram atribuídos para o ataque e a defesa o valor de 1 para todas as criaturas. Esta inicialização foi utilizada para melhor visualizar a evolução dos atributos e a influência das taxas de mutação e de cruzamento ao longo das gerações. Os gráficos dos resultados dos melhores indivíduos em cada geração, para cada tipo de criatura, são apresentados na Figura 2.

3.1. Operador de mutação

Para este experimento foi definida uma taxa de cruzamento de 50%. Nos resultados obtidos, observa-se que os herbívoros conseguiram alcançar sua melhor pontuação com a taxa de 10%. O melhor herbívoro de cada geração atingiu um *fitness* superior a 14, aproximadamente, a partir da geração 22. Já a média da pontuação da população atingiu valores de *fitness* superiores a 9, aproximadamente, a partir da geração 30. A partir de taxas maiores que 10% há uma queda substancial de desempenho. Pode-se também ressaltar que para as taxas mais altas ocorre uma evolução mais rápida nas primeiras gerações, que, porém, não chegam a gerar as pontuações mais altas a longo prazo.

Os carnívoros alcançaram valores bem mais baixos, obtendo o melhor desempenho (pico) com a taxa de 30%. Nas primeiras gerações pode-se observar um aumento do *fitness*, porém, após poucas gerações ocorre uma rápida queda. Um dos principais motivos para essa queda é a dependência dos carnívoros em relação à sobrevivência de outras populações, ou seja, disponibilidade de alimento. Além disso, a configuração inicial do cromossomo ocorre com valores aleatórios. Portanto, as primeiras gerações encontram-se, a princípio, ineficientes em relação à caça e ao ambiente, ocasionando alta taxa de mortalidade.

Em relação aos onívoros, pode-se notar uma correlação entre o aumento da taxa de mutação e um melhor desempenho nas curvas correspondentes, apresentando valores cada vez maiores e crescimento cada vez mais rápido nas primeiras gerações. Porém, a maioria das curvas apresentou um desempenho bastante próximo. Ainda assim, pode-se destacar que a curva de melhor desempenho foi a da taxa de 30%.

3.2. Operador de cruzamento

O objetivo deste experimento foi compreender o comportamento do operador genético de cruzamento. Para isso, a mutação e os demais parâmetros foram estabelecidos com valores fixos. Foram testados os valores de cruzamento de 0% a 100%, com intervalos de 5%. A partir dos resultados obtidos anteriormente foram fixados os valores de taxa de mutação para cada tipo: 10% para herbívoros, 30% para carnívoros e 30% para onívoros. Nos resultados obtidos não houve diferença significativa entre as curvas definidas pelos vários valores de taxa de cruzamento para os herbívoros e os onívoros. De forma similar aos resultados na mutação, os carnívoros, nestes experimentos, também apresentaram curvas com um pico e uma queda acentuada, sendo que o pico alcançou

baixa pontuação. Os motivos para isso são os mesmos apresentados anteriormente para a taxa de mutação. As curvas apresentaram valores bem próximos nas primeiras gerações e valores um pouco mais variados depois do pico.

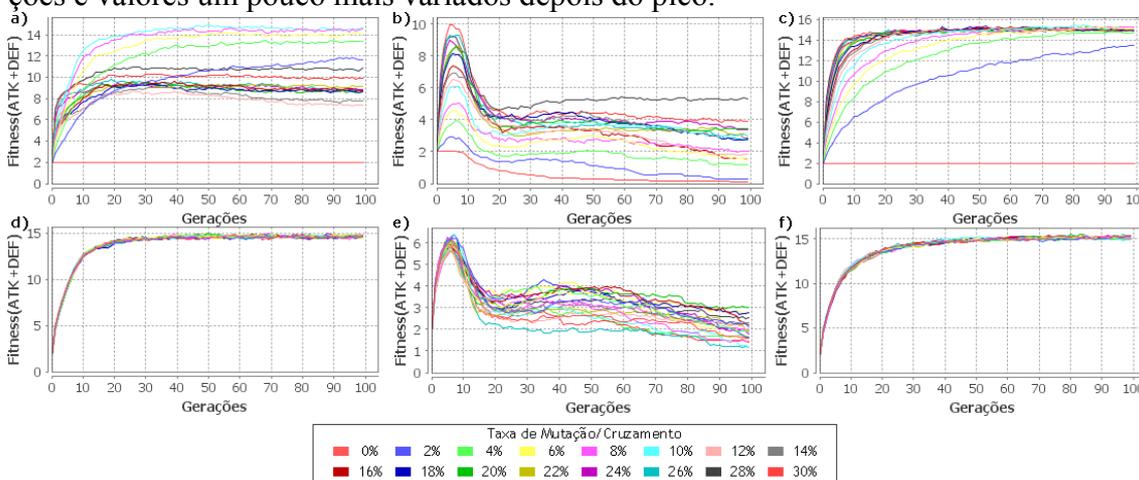


Figura 2: Curvas dos melhores valores de *Fitness* (ataque+defesa) ao longo das gerações, considerando várias taxas de mutação (a, b, c) e cruzamento (d, e, f) para herbívoros (a, d), carnívoros (b, e) e onívoros (c, f).

4. Conclusão

Os resultados do AG foram satisfatórios para os herbívoros e onívoros. Os valores de taxa de mutação que se mostraram mais adequados foram de 10% para os herbívoros e de 30% tanto para carnívoros quanto para onívoros. Pelos experimentos executados, pôde-se notar que o cruzamento influenciou pouco na evolução das criaturas.

Para os carnívoros, o resultado não atendeu as expectativas. Isso se deve a diversos fatores, como a dependência dos carnívoros em função da sobrevivência de outras populações. No começo da execução, a configuração inicial do cromossomo é com valores aleatórios. Desta forma, as primeiras populações naturalmente se apresentam imaturas (inadequadas) ao ambiente e à caça. Por isso, os carnívoros apresentam-se ineficientes no início e acabam apresentando uma alta taxa de mortalidade.

5. Referências

- Gonçalves, A. R. (2008). **Aplicações de técnicas de aprendizagem de máquina**. Londrina – PR, Universidade Estadual de Londrina. Relatório de Estágio Curricular.
- Holland, J. (2013). **Genetic Algorithms**, <http://www2.econ.iastate.edu/tesfatsi/holland.gaintro.htm>, Maio.
- Linden, R. (2006). **Algoritmos genéticos – Uma importante ferramenta da Inteligência Computacional**. Rio de Janeiro - RJ, BRASPORT .
- Reeves, C. (2013). **Genetic Algorithms**, <http://homes.ieu.edu.tr/~agokce/Courses/Genetic%20algorithms%20Colin%20reeves%20chapter%203.pdf>, Abril.
- Tanomaru, J. (1995). **Motivação, fundamentos e aplicações de algoritmos genéticos**. In: Anais do II Congresso Brasileiro de Redes Neurais.
- Whitley, D. (1994). **A Genetic Algorithm Tutorial**. In: Statistics and Computing, páginas 65–85.

Projeto de um Aplicativo Android de Monitoramento de Veículos do Transporte Público

Geovane Fedrechski¹
Richard Aderbal Gonçalves (Orientador)¹
Josiel Neumann Kuk (Co-orientador)¹

¹Departamento de Ciência da Computação (DECOMP)
Universidade Estadual do Centro-Oeste (UNICENTRO)
Rua Camargo Varela de Sá, 3 – CEP 85040-080
Guarapuava – PR – Brasil – Telefone: (42) 3629-8100

geonnave@gmail.com

Resumo. *O advento da computação móvel mostra o quanto ela é importante para o nosso mundo atual. Em paralelo, o transporte público urbano tem ocupado um importantíssimo papel para uma larga escala da população. Neste trabalho, apresenta-se um projeto com uma solução, baseada em Android, aplicada ao rastreamento em tempo real de veículos do transporte público. Para fornecer um projeto bem definido, foram utilizadas técnicas de Engenharia de Software e Interação Humano-Computador (como, por exemplo, Especificação de Requisitos e Design de Interface). Também foi desenvolvido um protótipo das principais telas do sistema na plataforma Android.*

Palavras-chave. *Android; Transporte Público Urbano; Monitoramento de Veículos em Tempo Real.*

1. Introdução

Em muitas áreas do conhecimento é evidente a busca do ser humano por ferramentas que possam melhorar a sua qualidade de vida. Tal afirmação se aplica fortemente à Computação: máquinas automatizadas foram desenvolvidas para, a princípio, ajudar em problemas matemáticos complexos, repetitivos e trabalhosos [Polloni e Fedeli 2009]. O objetivo deste trabalho está diretamente relacionado a esse princípio, pois propõe a utilização da Computação Móvel para resolver um problema de dificuldade de acesso a informações do transporte público urbano, por parte dos usuários.

Geralmente, o surgimento de novas áreas de pesquisa costuma trazer à tona uma ampla gama de novas possibilidades, bem como novos impactos na sociedade. Com a Computação Móvel não é diferente, pois já é notável a presença e o impacto dos dispositivos móveis (como *tablets*, *smartphones* e PDA's) na vida contemporânea [Ipea 2012] [Figueiredo e Nakamura 2003]. Temos como grandes exemplos disso os *smartphones*, que podem ser definidos como “celulares que oferecem recursos avançados, frequentemente com funcionalidades iguais às de um computador ou a capacidade de fazer *download* de aplicativos” [Google 2012].

Os *smartphones* estão entre os principais responsáveis pelo advento da Computação Móvel e hoje em dia já protagonizam um grande impacto social [Google 2012]. Segundo dados de uma pesquisa realizada em 27 países e liderada pelo Google, 14% dos brasileiros (cerca de 27 milhões de pessoas) possuem um *smartphone*. Destes, 73% não saem de casa sem seus aparelhos; 42% os utilizam para acessar a Internet todos os dias e 60% fazem uso deles enquanto utilizam algum meio de transporte público [Google 2012].

Assim, tendo uma visão do quão inserida no cotidiano das pessoas está a Computação Móvel, nada mais natural do que vê-la e utilizá-la cada vez mais como um instrumento de melhoria da qualidade de vida dos seus usuários. O problema que será abordado neste trabalho consiste na dificuldade – por parte dos usuários – do acesso à informação concernente ao transporte público nas grandes cidades.

Apesar do importantíssimo papel ocupado pelo transporte público nas grandes cidades, muitas vezes ele só é utilizado devido à inexistência de outras alternativas (como, por exemplo, um veículo próprio). Há uma certa aversão pelo uso do transporte público, a qual é muitas vezes consequência da má qualidade deste serviço [Ipea 2012].

A insatisfação dos usuários tomou forma num estudo realizado pelo IPEA (Instituto de Pesquisa Econômica Aplicada), no ano de 2011. Neste estudo, constatou-se que nas cidades com mais de 100 mil habitantes, 61% da população não consegue ser atendida pelo transporte público sempre que precisa [Ipea 2012]. Além disso, esse mesmo estudo aponta que 37% da população avalia a quantidade de informação sobre o transporte público como “muito ruim/ruim” (contra 34% dos que a julgam como “muito boa/boa”).

Sendo assim, constata-se que a falta de informação sobre o transporte público é um fator que desagrada significativamente os usuários. Dessa forma, seria interessante encontrar uma forma de diversificar e facilitar o acesso a esse tipo de informação. Os horários de ônibus e itinerários precisam, por exemplo, ser amplamente acessíveis. A partir disso, o objetivo deste trabalho é projetar um aplicativo para *smartphones* equipado com Sistema Operacional Android, destinado ao usuário final, cuja principal funcionalidade será a de exibir dados de geolocalização dos ônibus recebidos de um servidor em tempo real. Neste projeto, denominado “Cadê meu ônibus?”, abordou-se apenas o módulo do usuário para resolver o problema.

2. Materiais e Métodos

O projeto consiste na utilização de duas técnicas principais: Engenharia de Software (ES) e Interação Humano-Computador (IHC). A Engenharia de Software é uma área do conhecimento que coexiste com a Ciência da Computação. Nela encontramos teorias, métodos e ferramentas para projeto e desenvolvimento de *software* – desde os mais simples até os mais complexos e abstratos. Na maioria dos casos, os sistemas são relativamente grandes e complexos [Sommerville 1995]. A Interação Humano-Computador é uma disciplina interessada no projeto, implementação e avaliação de sistemas computacionais interativos para uso humano, juntamente com os fenômenos relacionados a esse uso [Barbosa e da Silva 2010].

A implementação do protótipo fez-se com a utilização da tecnologia Android, a qual está entre as mais avançadas em termos de mercado, funcionalidade e interface com o usuário na área de computação móvel [Google 2012]. O Android é baseado no *kernel* do Linux e permite que desenvolvedores criem aplicativos gratuitamente, inclusive com o auxílio ferramentas interessantes – como uma API (*Application Programming Interface*) pronta para integração com o Google Maps.

3. Resultados

Os resultados deste trabalho consistem nos projetos de ES e IHC desenvolvidos de uma forma prática. Foram desenvolvidos artefatos de ES e armazenados num documento de Especificação de Requisitos de *software* (ERSw), bem como projetados os Mockups e protótipos de interface do sistema.

No ERSw, uma das seções mais importantes é a seção Escopo, cujo objetivo é descrever o produto de *software* a partir do seu nome, e explicando as suas características. Além disso, é interessante também que se descreva os benefícios que se espera com o produto e o valor destes para o cliente [Paula Filho 2001]. Para um melhor entendimento da seção Escopo, tem-se a seguir o escopo do projeto desenvolvido.

O *software Cadê meu ônibus?* será um aplicativo para a plataforma móvel Android. Seu objetivo principal será exibir informações ao usuário sobre o geoposicionamento de veículos do transporte público em tempo real. Essas informações poderão ser visualizadas em um mapa, após a seleção de uma linha de ônibus. Além da visualização da localização dos ônibus em tempo real, também será possível ao *software* emitir alertas ao usuário, conforme a ocorrência de eventos pré-configurados. Para tanto, será necessária a conexão com a Internet.

Não obstante, o *software* também apresentará algumas funcionalidades que estarão disponíveis tanto em modo *online*, quanto em modo *offline* – com restrições nesta última. As principais funções disponíveis também no modo *offline* são: consulta, listagem e exibição de linhas, pontos e horários.

Ao facilitar o acesso às informações tanto em tempo real quanto estáticas aos usuários do transporte público urbano, o *Cadê meu ônibus?* visa agregar valor à qualidade deste serviço – cujos usuários estão frequentemente insatisfeitos [Ipea 2012] –, incentivando, assim, a sua utilização. Em se tratando dos resultados de IHC, os principais são os Mockups, os quais são representações de interface de baixa fidelidade, e foram desenvolvidos com o auxílio do *software* FluidUI [FluidSoftware 2013].

Durante o *design* da tela inicial, utilizamos uma convenção do Android que sugere a utilização do estilo *dashboard* (ou seja, um *layout* de grade com ícones e uma possível descrição adjacente para cada função [Google 2010]) para a tela inicial do aplicativo.

A elaboração dos Mockups foi capaz de abranger o sistema como um todo, isto é, todos os Requisitos Funcionais. Caso seja de interesse do leitor, todos os Mockups estão disponíveis para consulta no Material Suplementar B [Fedrechski 2013].

3.1. Protótipo

O escopo da prototipagem abordada neste trabalho reside na interação do usuário com o sistema. Portanto, o protótipo desenvolvido está focado somente no *design* da interface, deixando de lado – pelo menos por enquanto – as questões de funcionalidade. Os protótipos foram baseadas nos Mockups e as telas prototipadas encontram-se disponíveis no Material Suplementar B [Fedrechski 2013].

Os protótipos desenvolvidos tiveram uma abrangência significativamente menor do que os Mockups em termos de completude do sistema. Isso se deve à maior complexidade e atenção para cada detalhe que se faz necessária nesta fase. Mesmo que o termo “protótipo” sugira uma implementação mais propensa a falhas e com menos qualidade que um produto final (portanto, com menos riscos), considera-se ainda o protótipo desenvolvido em um contexto evolutivo, ou seja, trata-o como já sendo parte de um produto final.

4. Conclusão e Trabalhos Futuros

Neste trabalho, apresentou-se o projeto de um aplicativo para celulares Android cujo intuito é auxiliar os usuários do transporte público urbano a obterem informações sobre esse tipo de serviço. O grande diferencial do projeto é a funcionalidade de rastreamento dos veículos em tempo real. Além disso, informações estáticas sobre linhas, pontos e horários também são fornecidas através do aplicativo proposto.

Através de um estudo preliminar de algumas técnicas de Engenharia de Software e Interação Humano-Computador, e um posterior desenvolvimento destas, foi possível delimitar um escopo bem definido para a proposta de solução inicialmente apresentada. Além disso, a aplicação dos processos de ES e IHC, neste trabalho contribuiu para um melhor entendimento dos conteúdos abordados por cada uma das técnicas.

A utilização dessas técnicas serviu de base para a elaboração de um modelo de interação de baixa fidelidade (Mockups), o que culminou no desenvolvimento de um Protótipo voltado ao *design* da interface da aplicação proposta.

As principais dificuldades enfrentadas no desenvolvimento deste trabalho consistiram na prototipagem do aplicativo Android, bem como na definição do modelo do documento de ERSw a ser utilizado e na elaboração das tabelas de fluxo dos Caso de Uso.

Como continuação deste trabalho, espera-se que o legado deixado por este projeto possa ser utilizado para futuras melhorias da solução apresentada. Além disso, também fica em aberto o estudo e a implementação do módulo servidor e da infraestrutura de rastreamento dos veículos em tempo real.

Referências

- [Barbosa e da Silva 2010] Barbosa, S. D. J. e da Silva, B. S. (2010). *Interação Humano-Computador*. Campus.
- [Fedrecheski 2013] Fedrecheski, G. (2013). Material suplementar b. <http://goo.gl/z9ghdM>.
- [Figueiredo e Nakamura 2003] Figueiredo, C. M. S. e Nakamura, E. (2003). Computação móvel: Novas oportunidades e novos desafios.
- [FluidSoftware 2013] FluidSoftware (2013). Fluid ui. <https://www.fluidui.com/>.
- [Google 2010] Google (2010). Android ui design patterns. <https://dl.google.com/googleio/2010/android-android-ui-design-patterns.pdf>.
- [Google 2012] Google (2012). Nosso planeta mobile: Brasil. como entender o usuário de celular. <http://www.thinkwithgoogle.com/mobileplanet/pt-br/>.
- [Ipea 2012] Ipea (2012). Mobilidade urbana 2a edição análise preliminar dos dados coletados em 2011.
- [Paula Filho 2001] Paula Filho, W. P. (2001). *Engenharia de Software*. LTC, 1 edição.
- [Polloni e Fedeli 2009] Polloni, E. e Fedeli, R. (2009). *Introdução à Ciência da Computação*. Thomson Pioneira.
- [Sommerville 1995] Sommerville, I. (1995). *Software Engineering*. Addison-Wesley, 5 edição.

Paralelização em GPU de algoritmo baseado em otimização por colônia de formigas aplicado ao problema de coloração de grafos

Paulo Roberto Urio¹, Mauro Henrique Mulati (Orientador)¹

¹Departamento de Ciência da Computação (DECOMP)
Universidade Estadual do Centro-Oeste (UNICENTRO)
Rua Camargo Varela de Sá, 3 – CEP 85.040-080
Guarapuava – PR – Brasil

paulo@bk.ru, mhmulati@gmail.com

Resumo. Neste trabalho é investigada a implementação paralela de um algoritmo da meta-heurística de otimização por colônia de formigas, aplicado ao problema de coloração de grafos. Ao longo do trabalho, foram implementados dois algoritmos, um guloso e um de otimização por colônia de formigas sem busca local. Este, implementado em versão para CPU e versão GPU. Os parâmetros utilizados foram calibrados em um trabalho anterior, em uma implementação com busca local. O speedup adquirido foi de até oito vezes, porém a qualidade da solução degradou consideravelmente em relação às implementações com busca local.

Palavras-chave. Otimização por colônia de formigas; Coloração de grafos; Cuda.

1 Introdução

A otimização por colônia de formigas (ACO, do inglês *Ant Colony Optimization*) é uma meta-heurística inspirada no comportamento real de formigas. Esta técnica é baseada na simulação de vários indivíduos trabalhando coletivamente para encontrar uma solução satisfatória. Conceitualmente, as colônias de formigas formam um tipo de sistema distribuído, que sugere uma grande quantidade de paralelismo implícito (WEISS, 2011).

A unidade de processamento gráfico (GPU, do inglês *Graphics Processing Unit*) oferece um grande potencial de melhoria de desempenho para problemas maciçamente paralelos (GARLAND et al., 2008). Existem estudos que investigam a paralelização da ACO, como Stützle (1998).

Uma das aplicações da ACO é em coloração de grafos. O problema de coloração de vértices de grafos consiste em encontrar o menor número k de cores que possam ser atribuídas aos vértices de um grafo, tal que dois vértices ligados por uma aresta não possuam a mesma cor. O problema do número cromático de um grafo é um problema \mathcal{NP} -difícil e muita atenção e esforço tem sido voltados para o desenvolvimento de novas abordagens heurísticas e meta-heurísticas para obtenção de soluções satisfatórias. Estes métodos não garantem a solução ótima, porém são voltados para retornar boas colorações em tempo razoável.

Neste trabalho, são apresentadas implementações do algoritmo para a coloração de vértices de grafos proposto por Lintzmayer, Mulati e Silva (2011), sem a utilização de busca local, em uma versão sequencial e uma versão paralelizada em GPU.

2 Otimização por colônia de formigas

A técnica de otimização por colônia de formigas (ACO) é baseada na comunicação indireta de agentes simples, chamados de formigas artificiais, mediados por trilhas de feromônio artificiais (DORIGO; STÜTZLE, 2002). Esta técnica probabilística explora um mecanismo similar à estigmergia do comportamento de forrageamento. Inicialmente, as formigas artificiais fazem escolhas aleatórias e, ao longo das construções, as escolhas que são consideradas boas são reforçadas através do depósito de feromônio artificial.

Lintzmayer, Mulati e Silva (2011) apresentaram o algoritmo ColorAnt₃-RT. Este algoritmo utiliza como método construtivo um algoritmo com comportamento similar ao AntCol, proposto por Costa e Hertz (1997) e chamado AntFixed- k , o qual busca realizar uma k -coloração.

Na versão paralela, ao inicializar, enquanto uma k -coloração não for encontrada, um grupo de formigas constrói soluções paralelamente. Além disso, utilizando abordagem de decomposição de dados, cada formiga constrói uma solução paralelamente.

3 Metodologia

Duas versões implementadas foram testadas, uma sequencial, chamada versão *host*, e uma versão paralelizada em GPU, chamada versão *device*. Além disso, foi implementado o algoritmo guloso de Welsh e Powell (1967) para encontrar um limitante superior para cada instância DIMACS (JOHNSON; TRICK, 1996). A partir do limitante superior k , foi executado o algoritmo ColorAnt₃-RT com valor de k . Se em dez execuções, pelo menos uma conseguiu uma k -coloração, o valor de k foi decrementado em uma unidade para tentar melhorar a qualidade da solução.

Os parâmetros utilizados nos testes realizados foram encontrados em Lintzmayer, Mulati e Silva (2011), em que os algoritmos foram implementados utilizando-se busca local.

A versão paralela foi implementada utilizando a plataforma de desenvolvimento CUDA™ e executada em uma NVIDIA® GeForce® GTX 680.

4 Resultados

Em um teste preliminar, fez-se testes com cinco iterações fixas, sem levar em conta a qualidade de solução e a verificação de uma k -coloração como critério de parada. A versão *device* ficou apenas duas vezes mais rápida para grafos esparsos comparada à aceleração de até oito vezes para grafos densos.

Mesmo com *speedup* positivo apresentado pelo teste preliminar, foram obtidos vários tempos de execução mais lentos durante o teste das instâncias. Isto se deve pela divergência de soluções ou porque a versão sequencial encontrou uma k -coloração logo nas primeiras formigas, enquanto que a versão paralela sempre se constrói com um grupo de formigas, simultaneamente, a cada iteração. A versão paralela apresentou bons tempos de execução em instâncias de grafos densos.

De modo geral, a qualidade de todas as instâncias degradou em relação aos resultados obtidos por Lintzmayer, Mulati e Silva (2011). A exceção é a instância dsjr500.5 que

obteve k -coloração 122, a melhor coloração conhecida para a instância. Em comparação com o limitante superior obtido com o algoritmo guloso Welsh–Powell, os algoritmos implementados melhoraram a qualidade da solução, em média, com oito cores a menos.

A Tabela 1 apresenta os resultados para a execução das instâncias. A primeira coluna indica o nome da instância testada com o número cromático e o melhor limitante superior já encontrado, respectivamente, entre parênteses. A segunda coluna o valor de k utilizado para a instância. As três colunas seguintes representam os dados dos testes realizados com a versão *host* e as três últimas os testes realizados com a versão *device*. A terceira coluna mostra a quantidade de execuções bem sucedidas (S) e o total (T) de execuções. A quarta coluna a média do tempo de execução e a quinta coluna a média do número de arestas conflitantes (Cfs) de cada execução. Os resultados apresentados estão limitados para o menor valor k tal que ambas versões *host* e *device* tivessem pelo menos uma execução bem sucedida.

Comparados com o trabalho anterior de Lintzmayer, Mulati e Silva (2011), os resultados com busca local foram melhores. Acredita-se que a qualidade dos resultados obtidos possa ser melhorada com a implementação, sem busca local, com a realização da calibração dos parâmetros utilizados.

Em várias instâncias, como *dsjc500.5*, a versão *device* apresentou aceleração gradativa no tempo de execução, de até 5,9 para k igual a 64. Para k igual 63 a versão *device* apresentou tempo maior de execução. Nota-se que, para este caso, enquanto na versão *host* todas as dez tentativas foram bem sucedidas, na versão *device* apenas quatro tentativas das dez foram bem sucedidas. Assim, o tempo médio de execução é maior sempre que existe uma ou mais tentativas mal sucedidas.

Tabela 1. Resultados para o algoritmo ColorAnt₃-RT sem busca local, na versão implementada em *host* e versão implementada em *device*.

Instância (χ/k^*)	k	Host (CPU)			Device (GPU)		
		S/T	Tempo (s)	Cfs.	S/T	Tempo (s)	Cfs.
<i>dsjc500.1</i> (?/12)	15	10/10	0,62	0	10/10	0,74	0
<i>dsjc500.5</i> (?/48)	63	10/10	237,82	0	2/10	260,79	4,9
<i>dsjc500.9</i> (?/126)	159	10/10	869,15	0	4/10	424,55	4,3
<i>dsjc1000.1</i> (?/20)	25	10/10	190,40	0	9/10	316,14	0,2
<i>dsjc1000.5</i> (?/83)	116	10/10	831,92	0	10/10	299,22	0
<i>dsjr500.1c</i> (?/84)	92	2/10	2.902,19	3,5	2/10	346,47	5,4
<i>dsjr500.5</i> (?/122)	122	10/10	107,99	0	6/10	180,25	1
<i>flat300_28_0</i> (28/28)	39	10/10	49,60	0	1/10	5,85	5,1
<i>flat1000_76_0</i> (76/82)	114	9/10	1.834,69	0,2	4/10	1.858,45	3,9

5 Conclusão e trabalhos futuros

Do ponto de vista de ganho de desempenho por tempo de execução, ao comparar o tempo de teste em *device* e *host*, foi observado um *speedup* médio de 2,9 vezes para grafos com densidade de até 0,5 e *speedup* médio de 8,6 para grafos com densidade superior ou igual a 0,9. Mesmo assim, algumas instâncias na versão *device* foram mais lentas. Enquanto na versão *host* uma formiga constrói por vez, na versão *device* um grupo inteiro de formigas constrói em paralelo, tendo que esperar o grupo inteiro terminar para verificar se uma ou mais formigas conseguiram uma k -coloração.

Ambas as versões implementadas perderam tanto qualidade como tempo de execução em relação à implementação do algoritmo ColorAnt₃-RT com busca local. Os resultados das versões *host* e *device* foram diferentes devido à diferença da geração de números aleatórios. A diferença da quantidade de operações executadas nas versões *host* e *device* dificulta a comparação direta de tempo para cada instância.

Como trabalho futuro, a aplicação da busca local paralelizada pode ser estudada, para melhorar o tempo de execução em GPU.

Referências

COSTA, D.; HERTZ, A. Ants can colour graphs. *The Journal of the Operational Research Society*, Palgrave Macmillan Journals on behalf of the Operational Research Society, v. 48, n. 3, p. 295–305, 1997.

DORIGO, M.; STÜTZLE, T. The ant colony optimization metaheuristic: Algorithms, applications, and advances. In: *Handbook of Metaheuristics*. Dordrecht, Holanda: Kluwer Academic Publishers, 2002. p. 251–285.

GARLAND, M. et al. Parallel computing experiences with cuda. *Micro, IEEE*, v. 28, n. 4, p. 13–27, 2008. ISSN 0272-1732.

JOHNSON, D. J.; TRICK, M. A. (Ed.). *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, Workshop, October 11-13, 1993*. Boston, EUA: American Mathematical Society, 1996. ISBN 0821866095.

LINTZMAYER, C.; MULATI, M.; SILVA, A. da. Toward better performance of colorant aco algorithm. In: *Computer Science Society (SCCC), 2011 30th International Conference of the Chilean*. Curicó, Chile: IEEE, 2011. p. 256–264. ISSN 1522-4902.

STÜTZLE, T. Parallelization strategies for ant colony optimization. In: *Proceedings of PPSN-V, Fifth International Conference on Parallel Problem Solving from Nature*. Amsterdã, Holanda: Springer-Verlag, 1998. p. 722–731.

WEISS, R. M. Chapter 22 – gpu-accelerated ant colony optimization. In: HWU, W. mei W. (Ed.). *GPU Computing Gems Emerald Edition*. Boston, EUA: Morgan Kaufmann, 2011. p. 325–340. ISBN 978-0-12-384988-5.

WELSH, D. J. A.; POWELL, M. B. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, v. 10, n. 1, p. 85–86, 1967.

Algoritmo de Busca Gravitacional aplicado ao Problema do Despacho Econômico de Energia Elétrica

Eduardo Thomas Feliczaki¹
Richard Aderbal Gonçalves (Orientador)¹

¹Departamento de Ciência da Computação (DECOMP)
Universidade Estadual do Centro-Oeste (UNICENTRO)
Rua Camargo Varela de Sá, 3 – CEP 85.040-080
Guarapuava – PR – Brasil

www.edutf@gmail.com, richard@unicentro.br

Resumo. Neste trabalho, estudou-se o Algoritmo de Busca Gravitacional, que utiliza como metáfora as leis da gravitação e do movimento de Newton para a resolução de problemas não lineares. O algoritmo foi implementado e aplicado ao Problema de Despacho Econômico de Energia. A eficiência dos resultados desta implementação foi comparada com os resultados de outros algoritmos encontrados na literatura. Os resultados deste trabalho não foram melhores, mas podem ser aperfeiçoados em trabalhos futuros.

Palavras-chave: algoritmo gravitacional; despacho econômico de energia; computação natural.

1 Introdução

O Problema do Despacho Econômico de Energia Elétrica (EDP)¹ é um problema de otimização não linear em sistemas de energia que busca dividir a demanda de energia entre geradores elétricos, de forma a minimizar os custos de produção, enquanto obedece a determinadas restrições (FRAGA; YANG; PAPAGEORGIOU, 2012). Para solucionar esse problema, além de técnicas de programação dinâmica (LIANG; GLOVER, 1992), são utilizadas técnicas de computação natural, como o Algoritmo de Otimização com Enxame de Partículas (PSO)² (WANG et al., 2011) e o Algoritmo de Busca Gravitacional (GSA)³ (GÜVENÇ et al., 2012). Desta forma, neste trabalho será abordado o algoritmo de busca gravitacional, desenvolvido por Rashedi, Nezamabadi-pour e Saryazdi (2009) que utiliza como metáfora as leis de Newton da gravitação e do movimento. Para essa implementação, foi utilizada a versão mais clássica do Algoritmo, ou seja, sem operadores customizados.

¹EDP - *Economic Load Dispatch Problem*

²PSO - *Particle Swarm Optimization*

³GSA - *Gravitational Search Algorithm*

2 Materiais e Métodos

O problema de despacho de energia consiste em dividir a demanda de energia elétrica em um sistema composto por um conjunto de geradores, tendo como objetivo a minimização dos custos de operação, ao mesmo tempo que as restrições operacionais de cada gerador são satisfeitas (FRAGA; YANG; PAPAGEORGIOU, 2012), bem como a demanda de energia do sistema. O efeito de perda causado pelo estrangulamento de uma válvula parcialmente aberta de um gerador de energia elétrica é chamado de efeito de ponto de válvula (FRAGA; YANG; PAPAGEORGIOU, 2012). Para calcular o custo da produção de energia para um conjunto de geradores, levando em conta o efeito de ponto de válvula, é utilizada a equação 1,

$$f(P_j) = \left[a_j + b_j P_j + c_j P_j^2 + |e_j \times \sin(f_j \times (P_{jmin} - P_j))| \right], \quad (1)$$

sendo $f(P_j)$ o custo da produção de energia P_j pelo gerador j . Cada gerador possui uma função de custo diferente, o que é definido pelos coeficientes a_j, b_j, c_j, e_j e f_j . P_{jmin} é o limite mínimo de produção do gerador j .

O GSA é uma técnica recente de otimização heurística desenvolvida por Rashedi, Nezamabadi-pour e Saryazdi (2009). Nesse algoritmo, os agentes são considerados objetos e seus respectivos desempenhos são medidos por suas massas (SWAIN; SAHU; HOTA, 2012). Todos os objetos atraem-se pela força gravitacional e essa força faz com que os objetos se movam em direção aos objetos com maior massa. Assim, existe uma forma de comunicação entre as massas, a força gravitacional.

No GSA, os agentes/objetos sofrem alterações pelos operadores, tais operadores são implementações das equações das leis de gravitação e movimento, e com isso os objetos são deslocados dentro do espaço de busca. Os agentes com maior massa exercem uma maior força nos agentes com menor massa, assim os atraindo. Para o GSA aplicado ao Problema de Despacho de Energia, a posição de cada agente é definida pela configuração atual dos geradores e pela posição desses geradores é calculado o custo de cada indivíduo, utilizando a equação 1.

3 Resultados

O algoritmo foi testado em três configurações de geradores, que correspondem a instâncias existentes no mundo real e já usadas em outros estudos. As configurações e resultados de outros algoritmos para 3 e 13 geradores foram obtidas de Swain, Sahu e Hota (2012) e a instância e resultados de outros algoritmos com 40 geradores foi obtida de GÜvenÇ et al. (2012).

Foram computados intervalos de confiança (com 95% de confiança) e a mediana para os resultados deste trabalho, utilizando-se *bootstrap* com o método BCA

⁴ (CONOVER, 1999), com 10.000 reamostragens para todos os resultados, ou seja, se os experimentos forem replicados, em 95% das execuções, a mediana encontrada estará dentro do intervalo apresentado.

Cada resultado consiste do custo necessário em dólares para gerar a demanda de energia de cada instância.

3.1 Resultados para 3 e 13 geradores

Neste teste, o algoritmo foi executado 50 vezes para 3 geradores e 30 vezes para 13 geradores. Para fins de comparação de resultados, os parâmetros são iguais aos de Swain, Sahu e Hota (2012).

As configurações são as seguintes: demanda: 850 MW ⁵ (3 geradores) e 1800 MW (13 geradores); população: 100 agentes; constante gravitacional inicial G_0 : 100; valor de a : 8; iterações: 100 (3 geradores) e 3000 (13 geradores). Os resultados para 3 geradores podem ser observados na Tabela 1, e os resultados para 13 geradores podem ser observados na Tabela 2. O melhor resultado para 3 geradores foi US\$ 8.234,07, e para o GSA proposto foi US\$ 8.288,87, com os intervalos de confiança de (8425, 8604).

Tabela 1. Resultados para 3 geradores.

Método Evolutivo	CEP	FEP	GSA	GSA Proposto
Custo Mínimo	8234,07	8234,07	8234,07	8288,87
Custo Máximo	8241,83	8241,78	8241,95	9176,90
Custo Médio	8235,97	8234,24	8234,11	8534,53

Tabela 2. Resultados para 13 geradores.

Método Evolutivo	CEP	FEP	MFEP	IFEP	GSA	GSA Prop.
Mínimo	18048,21	18018,00	18028,09	17994,07	17963,84	18453,81
Máximo	18404,04	18453,82	18416,89	18267,42	18910,31	19533,10
Médio	18190,32	18200,79	18192,00	18127,06	18041,21	18905,24

O melhor resultado para 13 geradores foi US\$ 17.963,84, e para o GSA proposto neste trabalho, o menor custo foi de US\$ 18.453,81. Os intervalos de confiança para o GSA proposto são: (18.809, 18.972).

⁴BCA = *Bias-Corrected and Accelerated bootstrap*

⁵*Megawatt*

3.2 Resultado para 40 geradores

Neste teste, o algoritmo foi rodado 30 vezes e, para fins de comparação de resultados, os parâmetros são iguais aos de GÜvenÇ et al. (2012). As configurações são as seguintes: demanda: 10500 MW; população: 100 agentes; constante gravitacional inicial G_0 : 100; valor de a : 20; e iterações: 1000. Os resultados dessa execução podem ser observados na Tabela 3.

Tabela 3. Resultados para 40 geradores,

Método Evolutivo	MODE	PDE	NSGA-II	SPEA-2	GSA	GSA Prop.
Custo Mínimo	1,2579	1,2573	1,2583	1,2581	1,2578	1,2820

Nesta instância, o melhor resultado foi US\$ 125.730,00, para o algoritmo PDE ⁶, e o GSA proposto neste trabalho teve um custo de US\$128.200,00, com os intervalos de confiança de (129.158, 129.728).

4 Conclusão

O GSA foi estudado para a solução do problema de despacho de energia. Fez-se uma comparação dos resultados obtidos pela implementação proposta do GSA com aqueles da literatura e os resultados mostraram-se piores. Dada a natureza desse trabalho, isso era esperado e pode-se concluir que o uso de operadores não clássicos do GSA é benéfico para a resolução do Problema do Despacho Econômico de Energia. Cabe salientar que apesar das implementações do GSA com operadores customizados serem superiores aos demais algoritmos da literatura, a implementação clássica utilizada neste trabalho mostrou-se inferior a esses também. Isso apenas destaca ainda mais a importância de operadores mais eficientes no GSA.

Referências

CONOVER, W. *Practical nonparametric statistics*. [S.l.]: Wiley, 1999. (Wiley series in probability and statistics: Applied probability and statistics). ISBN 9780471160687.

FRAGA, E. S.; YANG, L.; PAPAGEORGIOU, L. G. On the modelling of valve point loadings for power electricity dispatch. *Applied Energy*, v. 91, n. 1, p. 301 – 303, 2012. ISSN 0306-2619.

GÜVENÇ, U. et al. Combined economic and emission dispatch solution using gravitational search algorithm. *Scientia Iranica*, v. 19, n. 6, p. 1754 – 1762, 2012. ISSN 1026-3098.

⁶PDE: *Pareto Differential Algorithm*

LIANG, Z.-X.; GLOVER, J. A zoom feature for a dynamic programming solution to economic dispatch including transmission losses. *Power Systems, IEEE Transactions on*, v. 7, n. 2, p. 544–550, 1992. ISSN 0885-8950.

RASHEDI, E.; NEZAMABADI-POUR, H.; SARYAZDI, S. Gsa: A gravitational search algorithm. *Inf. Sci.*, Elsevier Science Inc., New York, NY, USA, v. 179, n. 13, p. 2232–2248, jun. 2009. ISSN 0020-0255.

SWAIN, R.; SAHU, N.; HOTA, P. Gravitational search algorithm for optimal economic dispatch. *Procedia Technology*, v. 6, n. 0, p. 411 – 419, 2012. ISSN 2212-0173. 2nd International Conference on Communication, Computing ; Security [ICCCS-2012].

WANG, Y. et al. Chaotic self-adaptive particle swarm optimization algorithm for dynamic economic dispatch problem with valve-point effects. *Expert Systems with Applications*, v. 38, n. 11, p. 14231 – 14237, 2011. ISSN 0957-4174.