VII Jornada de Atualização em Informática da Unicentro

13, 14 e 15 de setembro de 2017

ANAIS DA VII JAI/UNICENTRO

ISSN: 2177-708X

Realização:

UNICENTRO
PARANÁ

Departamento de Ciência da Computação

Patrocínio:

CAIXA

TEOREMA SISTEMAS

reptec tecnologia

VII JAI-UNICENTRO

VII Jornada de Atualização em Informática da UNICENTRO

Departamento de Ciência da Computação

13 a 15 de setembro de 2017

Guarapuava – PR

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

J82a

Jornada de Atualização em Informática da Unicentro (7. : 13-15 set. 2017 : Guarapuava)
Anais da VII Jornada... / coordenado [por] Mauro Miazaki, Lucélia de Souza, Tony Alexander Hild – Guarapuava : Unicentro, 2017.
67 p.

ISSN: 2177-708X

Evento realizado entre 13 e 15 de setembro de 2017

1. Software. 2. Pesquisa. 3. Tecnologia. 4. Internet. 5. Dispositivos Móveis. I. Título.

CDD 004

VII Jornada de Atualização em Informática da Unicentro

VII JAI-UNICENTRO

Reitoria
Reitor: Prof. Aldo Nelson Bona
Vice-Reitor: Prof. Osmar Ambrósio de Souza

Pró-Reitorias
Ensino: Profª. Regina Célia Habib Wipieski Padilha
Pesquisa e Pós-Graduação: Prof. Marcos Ventura Faria
Extensão e Cultura: Profª. Elaine Maria dos Santos
Administração e Finanças: Profª. Eliane Horbus
Recursos Humanos: Robson Paulo Ribeiro Ferras
Planejamento: Prof. Gilberto Franco de Souza

Direção do Campus Cedeteg
Diretor: Prof. Fábio Hernandes
Vice-Diretora: Profª.Adriana Knob

Setor de Ciências Exatas e de Tecnologia
Diretor: Profª. Karina Worm Beckmann
Vice-Diretor: Prof. Rodrigo Oliveira Bastos

Departamento de Ciência da Computação
Chefe: Prof. Mauro Miazaki
Vice-Chefe: Profª. Lucélia de Souza

APRESENTAÇÃO

A VII Jornada de Atualização em Informática da UNICENTRO – VII JAI/UNICENTRO é promovida pelo Departamento de Ciência da Computação (DECOMP).

Como nas edições anteriores, a VII JAI/UNICENTRO tem por objetivo divulgar o curso de 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 de pesquisas e do mercado.

O objetivo geral deste evento é abordar temas atuais da Computação, visando atualizar e agregar conhecimento à comunidade acadêmica, aos alunos do Ensino Médio e demais profissionais da área. Dentre os objetivos específicos destacam-se: possibilitar a troca de experiências entre a academia e o mercado de trabalho, bem como promover divulgações e discussões sobre oportunidades profissionais na área.

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

Esperamos que os trabalhos apresentados possam contribuir para o enriquecimento técnico e científico 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: Comissão Organizadora, Execução, acadêmicos voluntários, 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
VII JAI-UNICENTRO

Análise do Impacto da Dimensionalidade de Objetos em Problemas de Agrupamento de Dados em Partições Nebulosas

Paulo Henrique Abdanur Gomes[1], Alexandre Szabo[1], Regiane Orlovski[1]

ph_abdanur@yahoo.com.br, alexandreszabo@gmail.com, regianeorlovski@hotmail.com

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *Agrupamento de dados é a análise de grupos utilizando métodos para organizar objetos de acordo com suas características. Na abordagem nebulosa todos os objetos pertencem simultaneamente a todos os grupos, variando o grau de pertinência [0,1]. Quando o número de atributos é alto, objetos podem ficar dispersos no espaço de busca. Dessa maneira, este resumo avalia a qualidade dos resultados produzidos pelo algoritmo Fuzzy c-Means (FCM) quando aplicado em problemas de alta dimensionalidade. Com este estudo foi possível concluir que o algoritmo não é sensível à dimensionalidade das bases de dados, não interferindo na qualidade das soluções.*

Palavras-chave: Mineração de Dados; Agrupamento Nebuloso de Dados; Multidimensionalidade de Objetos.

1. Introdução

Diante do aumento significativo de dados presentes em diversos segmentos, tornou-se morosa e complexa a análise e a organização dos mesmos, dificultando o uso desses, de maneira a auxiliar no processo de tomada de decisão. Portanto, tornou-se evidente a necessidade de uma técnica capaz de analisar dados de maneira automática e inteligente.

O trabalho descrito conta com testes utilizando o algoritmo FCM (Fuzzy c-Means) [1], um algoritmo clássico da literatura de agrupamento nebuloso, visando o impacto da alta dimensionalidade de objetos a serem agrupados.

O agrupamento de dados é responsável por agrupar objetos de maneira que objetos similares entre si pertençam ao mesmo grupo, enquanto objetos dissimilares entre si pertençam a grupos diferentes [2]. Como a organização dos objetos é desconhecida, *à priori*, existe o risco de comprometer a qualidade do agrupamento [3]. Quanto à redução de

dimensionalidade, é uma das mais relevantes formas de regressão, permitindo extinguir subconjuntos de atributos do conjunto original que descrevem os objetos do banco de dados [4].

Em contexto geral, o trabalho aqui descrito teve como objetivo a análise de desempenho do algoritmo *Fuzzy c-Means* (FCM), visando sua apropriação para determinadas dimensionalidades de objetos e quantidades de grupos.

2. Materiais e métodos

Para a confecção deste trabalho foi utilizado o algoritmo FCM (*Fuzzy c-Means*), proposto por Bezdek (1981), que é uma versão nebulosa do algoritmo *K-means*. No que diz respeito aos parâmetros de execução, o algoritmo foi executado utilizando o expoente de ponderação m = 2 e esse valor foi escolhido pelo fato de que não se sabe o quão sobrepostos os grupos estão [3]. O número de iterações utilizadas é de 100 para todas as bases, de forma a garantir que as mesmas terminem a execução. O algoritmo executou 10 vezes cada base e o número de protótipos foi igual ao número de classes presentes nas respectivas bases. Os resultados obtidos foram avaliados em função da medida que avalia o percentual de acerto do algoritmo em relação ao total de objetos avaliados, denominado Percentual de Classificação Correta (Pcc).

A Tabela 1 representa algumas das bases utilizadas nos testes realizados e suas respectivas características.

Tabela 1. Principais características das bases sintéticas *Fuzzy*.

Base de Dados	Número de Objetos	Número de Dimensões	Número de Classes
Glass	214	9	6
Iris	150	4	3
Liver	345	6	2
Libras	45	91	15
Ruspini	75	2	4

As bases de dados sintéticas já se encontram normalizadas com valores no intervalo [0,1] e o número de protótipos utilizado foi igual ao número de classes da respectiva base de dados.

3. Resultados e discussão

Na Tabela 2 são apresentados os resultados obtidos após os testes. As bases testadas tiveram resultados variados, sendo a base *Ruspini* a única com 100% de acerto. Vale destacar o desempenho do algoritmo nas bases *Iris* e *Libras*, com resultados satisfatórios acima de 88%. Entretanto, nas bases *Glass* e *Liver* o algoritmo apresentou um resultado inferior em comparação às demais.

Tabela 2. Resultados obtidos após os testes.

Bases de Dados	MPCC(%)	Número de Protótipos
Glass	53.74	6
Iris	89.33	3
Liver	57.97	2
Libras	88.89	15
Ruspini	100	4

4. Considerações finais

Neste trabalho foi abordado o problema da dimensionalidade de objetos no agrupamento de dados nebuloso, utilizando o algoritmo FCM. O trabalho permitiu concluir que os resultados não são influenciados pela dimensionalidade das bases. Em relação ao número de classes, foi possível aferir que o mesmo pouco influencia na solução, considerando que tanto bases com um número elevado de grupos quanto com números inferiores tiveram resultados adequados.

Referências

[1] BEZDEK, J. C. Pattern recognition with fuzzy objective function algorithms. Springer Science & Business Media, 1981.

[2] DI CARLANTONIO, L. M. Novas metodologias para clusterização de dados. PhD thesis, UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, 2001.

[3] SZABO, A. et al. Agrupamento nebuloso de dados baseado em enxame de partículas: seleção por métodos evolutivos e combinação via relação nebulosa do tipo-2, 2014.

[4] HAIR, J. F., BLACK, W. C., et al. Multivariate data analysis. Prentice hall Upper Saddle River, NJ, 1995.

Aplicativo de apoio ao manejo dietoterápico para Fenilcetonúria

Guilherme V. G. da Silva[1], Ana Elisa T. P. da Palma[1], Guilherme B. L. de Freitas[2],
{guilhermevgorski, aetpiekarski, prof.gbarroso}@gmail.com
[1]Dep. de Ciência da Computação – DECOMP, Univ. Estadual do Centro-Oeste – UNICENTRO
[2]Dep. de Farmácia – DEFAR, Univ. Estadual do Centro-Oeste – UNICENTRO

Resumo: *A fenilcetonúria atinge cerca de 8 mil brasileiros. A doença não tem cura, apenas um controle, que é constituído da redução de fenilalanina na dieta, aminoácido presente na proteína dos alimentos. Esse tipo de dieta é considerada uma das mais restritivas do mundo, pois a proteína é essencial ao desenvolvimento e à vida. O controle é realizado a partir de cálculos baseados na quantidade de proteínas presente nos alimentos, limitando a ingestão de fenilalanina. Este trabalho descreve o desenvolvimento de um aplicativo móvel para apoio ao manejo dietoterápico para fenilcetonúria.*
Palavras-chave: Fenilcetonúria; PKU; Phenylketonuria.

1. Introdução

A Fenilcetonúria (PKU) é uma doença congênita que gera o acúmulo do aminoácido fenilalanina (Phe) no corpo, devido à deficiência da fenilalanina hidroxilase, enzima que catalisa a conversão de fenilalanina em tirosina. A introdução de uma dieta com baixo teor de proteínas deve ter início assim que a doença é diagnosticada, por meio do "teste do pezinho", pois a Fenilcetonúria não tem cura, apenas um controle sobre o nível de fenilalanina [1].

Os resultados de pesquisas feitas em 12 estados brasileiros identificaram cerca de 8 mil casos de fenilcetonúria no país [3]. As pessoas que sofrem dessa doença possuem muita dificuldade no momento de calcular a dieta e a quantidade de cada produto que é possível consumir, sendo necessário, muitas vezes, fazer contas com papel e caneta e com bases de dados alimentares restritas [1].

Tendo em vista que 84% da população possui aparelho celular, sendo que 81,7% utiliza o sistema operacional Android, 17,9% o iOS e 0,3% Windows [2], um aplicativo para apoiar o manejo dietoterápico da doença pode ser uma ferramenta útil, tanto para os fenilcetonúricos, quanto para os profissionais de saúde. No entanto, até o momento não há aplicativos, em Língua Portuguesa, disponíveis para apoiar esses cálculos.

2. Dieta para fenilcetonúricos: como calcular a quantidade de fenilalanina diária

Para realizar os cálculos necessários e determinar a quantidade máxima de fenilalanina que pode ser consumida diariamente, é preciso definir a quantidade desse aminoácido presente

em cada grama de proteína. Considera-se que, em média, 5% do total de aminoácidos presentes nas proteínas seja de fenilalanina [6].

A soma da quantidade de fenilalanina de todos os produtos a serem consumidos deve se manter moderada, pois por ser um aminoácido essencial, o seu consumo é necessário para o desenvolvimento. Portanto, a quantidade mínima necessária que deve ser consumida diariamente também deve ser estipulada [6].

O método de cálculo adotado no aplicativo, segundo Santos [6], é o mais confiável dentre os métodos comumente adotados.

3. Desenvolvimento do aplicativo para o manejo dietoterápico

Desenvolver aplicativos para sistemas operacionais distintos é repetitivo, pois, apesar de a lógica ser semelhante, cada um possui uma arquitetura específica. A plataforma Xamarin visa amenizar a necessidade de desenvolver códigos separados utilizando apenas uma linguagem de programação [4]. Em relação à compatibilidade das interfaces, deve-se utilizar o modelo Xamarin.Forms e a linguagem XAML.

O Xamarin.Forms é um modelo de projeto criado para reduzir a necessidade da criação de uma interface para cada sistema operacional. A plataforma Xamarin compartilha a lógica de negócio do aplicativo e o Xamarin.Forms compartilha a interface [7].

O XAML (Extensible Application Markup Language) é uma linguagem de marcação declarativa, baseada em XML, criada pela Microsoft, usada para a formação de UI (Interfaces com Usuário). A vantagem do uso do XAML é a interface nativa gerada para Android, iOS e Windows. O Xamarin trata de mapear os componentes da interface de usuário para cada componente de UI específico em cada plataforma, sem precisar que o desenvolvedor defina isso manualmente [5].

A partir do estudo e análise das dificuldades encontradas por fenilcetonúricos ou seus cuidadores, os especialistas que apoiaram o desenvolvimento do aplicativo propuseram os requisitos do sistema. Com base nos requisitos, foi definida a estrutura do aplicativo, contendo quatro módulos: Atualizar Dados, Lista de Produtos, Meus Itens e Acompanhamento Diário.

Na primeira utilização do aplicativo, por meio do módulo *Atualizar Dados*, o usuário deverá inserir seu nome, peso e data de nascimento, as informações são armazenadas no banco de dados e utilizados para a realização dos cálculos no módulo *Acompanhamento Diário*, sempre atualizando os dados de acordo com o ganho ou perda de peso.

O módulo *Lista de Produtos* apresenta os itens comercializados no mercado. Ao selecionar um dos itens da lista, é realizado o cálculo de fenilcetonúria e apresentado o resultado na tela do aplicativo. O usuário também pode incluir novos itens na lista, a partir das informações da tabela nutricional dos alimentos, cadastrando o nome, a marca, o tamanho da porção e a quantidade de proteínas de cada produto.

Devido à alta restrição de proteínas, o fenilcetonúrico prefere criar suas receitas sem proteínas ou com quantidades mínimas, as quais utilizam vários ingredientes, sendo que a quantidade de proteína deve ser indicada por porção, como nos demais produtos que constam no sistema. O módulo *Meus Itens* permite que o usuário adicione novos itens, definindo o nome da receita, a quantidade de cada ingrediente e de proteína presente nas porções dos ingredientes. Quando um item é selecionado, o cálculo de fenilalanina é realizado de acordo com a quantidade de proteínas presente no item, e, então, são apresentadas as informações do produto e o usuário pode adicioná-lo ao seu consumo diário.

O módulo *Acompanhamento Diário* lista os produtos adicionados, somando a quantidade de proteína e fenilalanina total presentes em todos os alimentos. A partir dos dados do usuário, a quantidade máxima e mínima de fenilalanina que pode ser consumida é apresentada e uma barra referente à quantidade consumida é preenchida a medida que os novos itens são adicionados. Quando a barra atinge a quantidade máxima diária, o usuário é notificado sobre ter atingido o limite que pode ser consumido.

4. Considerações finais

A fenilcetonúria é grave e, se não tratada, gera vários riscos para a saúde. Por ser uma doença rara, não existem muitas soluções que auxiliem no seu tratamento. Portanto, o aplicativo foi desenvolvido com a intenção de disponibilizar uma ferramenta para auxiliar as pessoas que possuem dificuldades para realizar os cálculos necessários para o manejo dietoterápico. O aplicativo atende os objetivos propostos, realizando os cálculos de forma confiável.

Referências

[1] DE MIRA, N. V.; MARQUEZ U. M. L. Importância do diagnóstico e tratamento da fenilcetonúria. *Revista de Saúde Pública 34*, 1 (2000), 86-96.

[2] GARTNER. Gartner says worldwide sales of smartphones grew 7 percent in the fourth quarter of 2016. Disponível em: http://www.gartner.com/newsroom/id/3609817. Acesso em: 29 abril 2017.

[3] MONTEIRO, L. T. B.; CÂNDIDO, L. M. B. Fenilcetonúria no Brasil: evolução e casos. Revista de Nutrição (2006).

[4] PROCEDI, L. Avaliação do framework xamarin.forms para desenvolvimento de aplicativos móveis multiplataforma, criando uma aplicação real.

[5] REYNOLDS, M. Xamarin móbile application development for Android. Packt Publishing Ltd, 2014.

[6] SANTOS, M. Fenilcetonúria: diagnóstico e tratamento. Comunicação em Ciências da Saúde 23, 4 (2012), 263-70.

[7] XAMARIN INC. Sharing code options. Disponível em: https://developer.xamarin.com/guides/cross-plataform/application_fundamentals/code-sharing/offline.pdf Acesso em: 30 abril 2017.

Combinação de Agrupamentos com o CLONALG: Métodos Evolutivos para a Seleção de Partições

Juliano Carvalho, Alexandre Szabo

julianocarvalho128@gmail.com.br, alexandreszabo@gmail.com.br

Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *Avanços tecnológicos têm favorecido a geração e o armazenamento de grandes quantidades de dados, sendo necessária a aplicação de algoritmos capazes de analisá-los e extrair conhecimento. A tarefa de agrupamento de dados refere-se à organização de objetos em grupos, usando alguma medida de (dis)similaridade entre eles. Combinação de Agrupamentos é uma técnica que combina partições por meio de uma função de consenso. Este trabalho propõe utilizar o Algoritmo de Seleção Clonal para gerar as partições individuais (diferentes soluções geradas para um mesmo problema) e aplicá-lo como função de consenso.*

Palavras-chave: Agrupamento de Dados; Combinação de Agrupamentos; Função de Consenso; Métodos Evolutivos; Sistema Imunológico Artificial.

1. Introdução

A informatização de tarefas aumentou substancialmente a capacidade para gerar e coletar dados de diversas fontes. Isso levou à geração de uma área promissora na Ciência da Computação chamada Mineração de Dados.

Agrupamento de dados é uma tarefa caracterizada pela segmentação de objetos em grupos em função da similaridade entre esses, geralmente avaliada por alguma medida de distância. A parametrização do algoritmo a ser aplicado em determinado problema depende das características da base de dados a ser agrupada, as quais, a *priori,* não são facilmente identificadas. Para melhorar a estabilidade e a robustez das saídas de agrupamento, surgiu, recentemente, a combinação de agrupamento. Em uma combinação de agrupamento, vários algoritmos de agrupamento fornecem soluções para a tarefa de agrupamento de dados. O algoritmo investigado neste trabalho, para realizar a tarefa de agrupamento de dados e a função de consenso, é o Algoritmo de Seleção Clonal (CLONALG) [1], inspirado no

Princípio da Seleção Clonal [2]. O algoritmo possui um parâmetro que representa o limiar de afinidade antigênica entre célula e antígeno, fazendo analogia ao mecanismo de defesa do sistema imunológico presente nos animais vertebrados [1].

2. Materiais e métodos

A implementação do CLONALG foi feita na linguagem de programação Java, a qual é orientada a objetos e possui grande portabilidade, oferecendo uma diversidade de bibliotecas e ferramentas que auxiliam no desenvolvimento e produtividade. Para cada execução do algoritmo, uma única célula será gerada, inicialmente, pois a quantidade de grupos não é conhecida. As bases de dados (que constam na Tabela 1) para testes e análise foram escolhidas de acordo com a literatura [3].

Tabela 1. Principais características das bases de dados

Bases	Objetos	Atributos	Classes
Ionosphere	351	34	2
Iris	150	4	3
Soybean	47	35	4
(Statlog)Heart	270	13	2
Wine	178	13	3

Para o estudo da qualidade das partições geradas foi avaliado o percentual de Classificação Correta (PCC(%)), que é uma medida que avalia o número de objetos classificados corretamente em relação ao total de objetos que compõe a base de dados.

A partição consenso foi avaliada para diferentes métodos de seleção de partições a serem combinadas. Os métodos escolhidos foram o Elitismo, que consiste em preservar os melhores indivíduos, e o Roleta, em que indivíduos de uma geração são escolhidos aleatoriamente na roleta e proporcionalmente ao seu valor de *fitness*.

3. Resultados e discussão

Na Figura 1, são apresentados quais percentuais obtiveram os melhores resultados dos métodos Roleta e Elitismo. Assim, é possível afirmar que o método Roleta teve desempenho superior ao método Elitismo e os percentuais 75% e 100%, na maioria dos casos, teve desempenho superior aos demais percentuais. Com base na Figura 1, pode-se concluir que os resultados das partições de consenso são satisfatórios e que, para a maioria das bases, a função de consenso teve ganho em relação a partição da base.

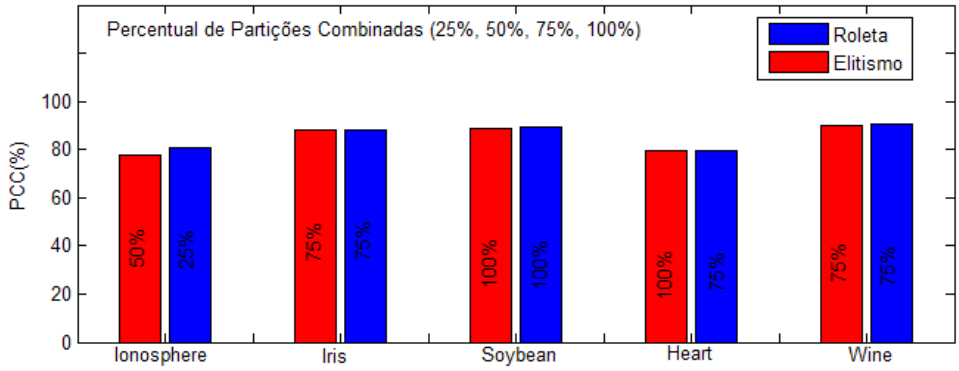

Figura 1. Melhores resultados para as bases da literatura

4. Considerações finais

É possível concluir que a aplicação do CLONALG como função de consenso gera resultados promissores, uma vez que a maioria das partições de consenso teve ganho em relação à partição da base. Também conclui-se que os métodos evolutivos Roleta e Elitismo, nos percentuais 25% e 50%, são os mais promissores para a base Ionosphere, porém, para as demais bases, os percentuais 75% e 100% tiveram resultados melhores para o CLONALG.

Referências

[1] De CASTRO, L. N. et al. The clonal selection algorithm with engineering applications. In Proceedings of GECCO, volume 2000, pages 36–39, 2000.

[2] BURNET, S. F. M. et al. **The clonal selection theory of acquired immunity**, volume 3.Vanderbilt University Press Nashville, 1959.

[3] De OLIVEIRA, J. V. et al. Particle swarm clustering in clustering ensembles: Exploiting pruning and alignment free consensus. Applied Soft Computing, 55:141–153, 2017.

Definição de regras de *Object Constraint Language* em modelos para auxiliar o desenvolvimento de aplicações Android

Jean Carlos Hrycyk[1], Inali Wisniewski Soares[1]
jean_carlos97@yahoo.com.br, inaliw@gmail.com

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *A Linguagem de Restrição de Objetos (Object Constraint Language – OCL) permite melhorar o desenvolvimento de modelos na abordagem Arquitetura Dirigida a Modelos (Model Driven Architecture – MDA), tornando-os mais precisos. Devido ao crescente uso de dispositivos móveis e a necessidade de novos aplicativos, a MDA, aliada ao uso de restrições OCL, permite desenvolver aplicativos de qualidade em tempo de desenvolvimento reduzido. Este trabalho visa o desenvolvimento e a aplicação de restrições OCL a modelos Android no contexto da MDA.*

Palavras-chave: Arquitetura Dirigida a Modelos; Linguagem de Restrição de Objetos; Aplicações Android.

1. Introdução

Atualmente, buscam-se novas abordagens para agilizar o desenvolvimento de aplicações móveis, a Arquitetura Dirigida a Modelos (*Model Driven Architecture* – MDA) representa sistemas a partir de combinações de desenho e texto, chamados modelos. Estes modelos possibilitam à MDA maior flexibilidade e rapidez no desenvolvimento de aplicações, sendo uma alternativa para a utilização em aplicações móveis [3].

Uma das maiores plataformas *mobile* atualmente é o Android, este sistema é gerenciado pela Google e a *Open Handset Alliance* (OHA), que oferecem suporte ao projeto do sistema. Sua principal característica é ser de código livre e gratuito, possibilitando ao desenvolvedor maior personalização e possibilidades de desenvolvimento de aplicações [2].

A definição de modelos pela MDA é feita com a linguagem de modelagem *Unified Modeling Language* (UML) [5], porém, esta não consegue expressar todos os detalhes de um modelo, pois, para isso, é preciso adicionar mais informações aos modelos, como: restrições, regras, entre outros. Nesse sentido, a fim de deixar os modelos mais robustos e bem definidos, utiliza-se a Linguagem de Restrição de Objetos (*Object Constraint Language – OCL),* que é

uma linguagem de expressão de restrições especificada pelo Grupo de Gerenciamento de Objetos (*Object Management Group* – OMG) [4].

2. Materiais e métodos

Neste trabalho foi desenvolvido um caso de estudo para apresentar as regras OCL definidas para um aplicativo Android, denominado AppCalagem, que foi escolhido por ser simples e apresentar características da plataforma. A função do aplicativo é efetuar o cálculo de calagem de solo a partir de dados da análise de solo que o usuário insere no aplicativo.

Para obter maior conhecimento sobre a plataforma Android, o aplicativo AppCalagem foi desenvolvido na forma tradicional, baseado em código, e, para isso, utilizou-se a linguagem JAVA em conjunto com a IDE oficial de desenvolvimento Android Studio.

O aplicativo AppCalagem também foi desenvolvido, parcialmente, utilizando modelos da abordagem MDA. Primeiramente, foi desenvolvido o Modelo Independente de Plataforma (*Plataform Independent Model* – PIM), que representa a especificação formal da estrutura e a função da aplicação, e nele detalhes técnicos de implementação e plataforma são abstraídos [1]. É nesse modelo que serão aplicadas as restrições OCL futuramente definidas.

Em seguida, após obter conhecimento técnico sobre desenvolvimento Android, baseado em código, foi possível definir o Modelo de Plataforma (*Plataform Model – PM)* e o Modelo Específico de Plataforma (*Plataform Specific Model* – PSM) da aplicação. O modelo de plataforma contém todos os detalhes de utilização e funcionalidade da plataforma, que neste caso é o Android. Por fim, combinando o modelo PIM e o modelo de plataforma, foi definido o PSM do AppCalagem e nesse modelo está toda a definição da função do sistema e também todas as funcionalidades da plataforma Android utilizadas no aplicativo [3].

Após a definição do modelo PIM foram criadas e aplicadas algumas regras OCL para fazer a verificação sintática e semântica do modelo. A verificação sintática avalia se as construções estruturais dos modelos estão corretas, já a verificação semântica realiza a definição de restrições estruturais nos modelos e avalia se estas estão definidas corretamente.

A validação do modelo ocorre em duas etapas. Na primeira é analisado se o modelo está em conformidade com as especificações da linguagem UML que define o modelo. Na segunda etapa, são definidas restrições OCL para o modelo PIM e avaliado se essas restrições foram definidas corretamente, se isso ocorrer e estiverem em conformidade com o modelo, então o modelo está bem formado e é válido.

3. Resultados e discussão

Utilizando-se da linguagem UML foi definido o modelo PIM do aplicativo AppCalagem. Em seguida foram definidas algumas regras OCL para fazer a validação desse modelo. Com isso as restrições e a semântica de operações do modelo ficaram bem definidas e livres de ambiguidade. A verificação garante que o sistema foi construído corretamente e a validação permite que o sistema realize os requisitos do usuário.

Esse processo colabora com um dos principais objetivos da MDA que é a transformação de modelos, pois tendo modelos bem definidos e estruturados diminui-se a ocorrência de erros e falhas nessas transformações.

4. Conclusão

O desenvolvimento deste trabalho permitiu verificar na prática a importância dos modelos no desenvolvimento de software. O desenvolvimento tradicional foi importante para entender os conceitos técnicos da plataforma Android. Além disso, esse tipo de desenvolvimento auxiliou na construção parcial do aplicativo na abordagem MDA.

Com a definição e aplicação de regras OCL no modelo PIM estrutural do aplicativo, foi possível validar o modelo e torná-lo mais preciso em suas informações, e, assim, observar que estas regras deixam os modelos mais robustos e completos e fazem com que se comportem conforme as especificações do sistema.

Referências

[1] KENT, S. Model Driven Engineering. *In International Conference on Integrated Formal Methods*, pages 286–298. Springer, 2002.

[2] KUMAR, S. Development and Research Implementation of Remote Object Monitoring Through Video Streaming Based on Android Mobile. International Journal of Internet Computing, v. 2, p. 61-65, 2011.

[3] OMG (2003). Technical Guide to Model Driven Architecture: The MDA guide v1.0.1. Disponível em: http://www.omg.org/docs/omg/03-06-01/PDF. Object Management Group, 2003.

[4] OMG. Object Constraint Language Specification (OCL). 2014. Disponível em: <http://www.omg.org/spec/OCL/2.4>.

[5] OMG (2015). Unified Modeling Language (UML). Disponível em: http://www.omg.org/spec/UML/2.5/PDF. Object Management Group, 2015.

Descoberta de Conhecimento em Base de Dados e Dados Abertos Conectados: um Estudo Dirigido a Dados Cientométricos

Sandro Kaue Motyl

sandro.motyl@gmail.com

Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *Este trabalho utiliza os conceitos de Descoberta de Conhecimento em Bases de Dados (Knowledge Discovery in Databases) e Dados Abertos Conectados (Linked Open Data) no contexto da Cientometria. No desenvolvimento, seguiu-se o processo metodológico de Descoberta de Conhecimento em Base de Dados em uma base de dados cientométricos no âmbito de uma universidade. Como ferramenta foi utilizado o software RapidMiner. Esse processo metodológico foi aplicado a um cenário de uso, mais especificamente a análise do histórico de produções científicas da universidade. Como resultado, foi possível identificar o departamento/setor mais produtivo e com maior qualidade das produções científicas.*

Palavras-chave: Descoberta de Conhecimento em Base de Dados; Dados Abertos Conectados; Cientometria.

1. Introdução

A Cientometria é a "disciplina que tem por objetivo medir as atividades de pesquisa científica e tecnológica mediante insumos (mão-de-obra, investimentos) e produtos (equipamentos, produtos, publicações)" [1].

Recentemente, alguns dados cientométricos e bibliométricos foram disponibilizados conforme os princípios *Linked Open Data* (LOD) [2]. O termo LOD é usado para definir um conjunto de melhores práticas para organizar, publicar e conectar dados na Web de Dados [3].

Nesse sentido, este trabalho tem como objetivo estabelecer um entendimento acerca da evolução da pesquisa científica da UNICENTRO, mediante dados cientométricos. Para isso, usou-se os processos de Descoberta de Conhecimento em uma Base de Dados (*Knowledge Discovery in Database* – KDD), que é um campo interdisciplinar criado a partir do

desenvolvimento de métodos e técnicas para dar sentido aos dados, ou seja, faz a extração de informações úteis de uma base de dados [4].

Neste estudo, o processo de KDD tem suas etapas de seleção de dados, pré-processamento, transformação, mineração e avaliação conduzidas com o apoio do software RapidMiner, com uma base de dados composta por dados cientométricos da UNICENTRO.

2. Materiais e métodos

Este trabalho possui duas fontes de dados distintas. Uma das fontes é o índice Qualis, que é atualizado anualmente e avalia os meios de divulgação de produções científicas, como revistas, anais e periódicos. A outra fonte de dados é proveniente da Plataforma Lattes, um sistema de informação integrado mantido pelo Ministério da Ciência, Tecnologia e Inovação do Brasil. Essas fontes são detalhadas a seguir:

1. **QualisBrasil** – conjunto de dados abertos conectados e provenientes do histórico do índice Qualis [5], período 2005-2015, acessível a partir do *endpoint* http://lod.unicentro.br/sparql, no grafo http://lod.unicentro.br/QualisBrasil/.
2. **LattesProduction** – conjunto de dados abertos conectados de registros dos artigos científicos publicados em revistas pelos professores da Unicentro, período 2005-2015 [6].

3. Resultados e discussão

O processo de KDD foi aplicado na base de dados obtida pela junção das duas bases de dados previamente listadas e seus resultados são divididos em três cenários de uso:

1. Relação entre quantidade e qualidade de produções científicas;
2. Análise de publicações de um departamento;
3. Ranqueamento de docentes em uma determinada área de publicação.

Dentre os cenários anteriormente citados, na Figura 1 está ilustrado o acompanhamento de produções científicas da universidade durante os anos de 2005 a 2015 (um dos resultados obtidos no cenário 1).

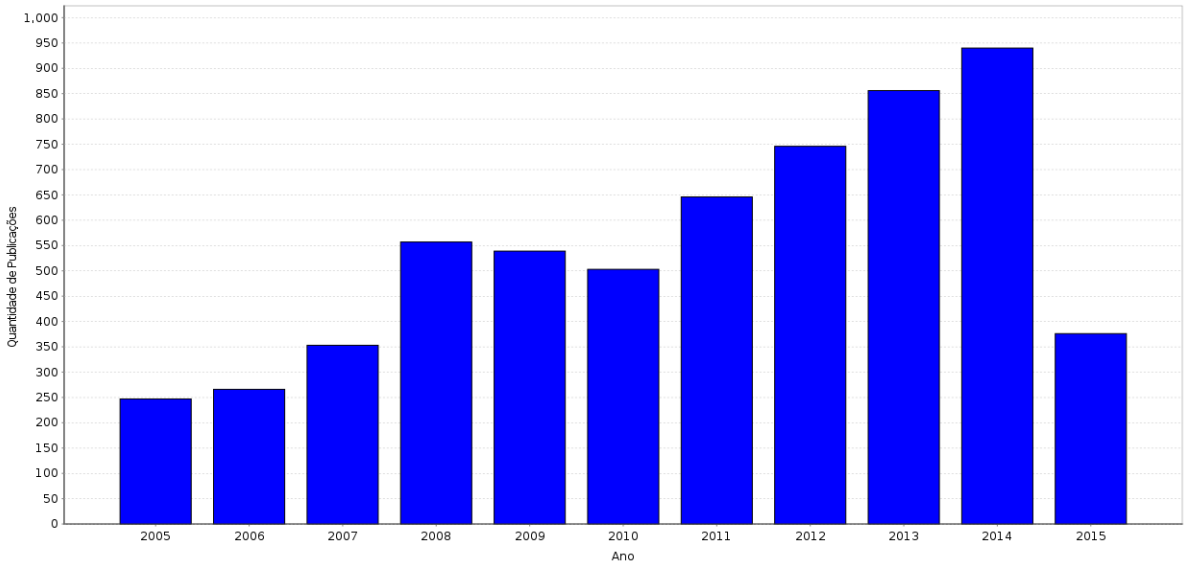

Figura 1. Quantidade de publicações/ano referentes à Unicentro.

4. Considerações finais

Este trabalho alcançou o objetivo, pois foi possível traçar e analisar o histórico de produções científicas da universidade. Portanto, confirma-se a aderência do processo KDD à prospecção de cenários com dados cientométricos.

Referente à ferramenta RapidMiner, algumas limitações foram identificadas, porém essas limitações não comprometeram a capacidade de obter resultados confiáveis.

Referências

[1] CUNHA, M. B. D., CAVALCANTI, C. R. D. O. **Dicionário de biblioteconomia e arquivologia. Briquet de Lemos/Livros**. 2008.

[2] RAUTENBERG, S., MARX, E., ERMILOV, I., AUER, S. Linked data workflow project ontology: uma ontologia de domínio para publicação e preservação de dados conectados. *Tendências da Pesquisa Brasileira em Ciência da Informação*. 2014.

[3] DATA, L. Linked data – connect distributed data across the web. Disponível em: http://linkeddata.org. Acesso em: 28 de Maio de 2017 16:30.

[4] FAYYAD, U., PIATETSKY-SHAPIRO, G., SMYTH, P. From data mining to knowledge discovery in databases. *AI magazine*, 1996.

[5] RAUTENBERG, S.; BURDA, A. Linked Open Data para Cientometria: Compartilhando e Mantendo o índice Qualis na Web de Dados. In: Encontro Brasileiro de Bibliometria e Cientometria, 5., 2016, São Paulo. *Anais...* São Paulo: USP, 2016. p. A34

[6] RAUTENBERG, S.; MOTYL, S. LINKED OPEN DATA E CIENTOMETRIA: Um Ensaio de Consumo de dados na Web de Dados In: Encontro Brasileiro de Bibliometria e Cientometria, 5., 2016, São Paulo. *Anais...* São Paulo: USP, 2016.

Desenvolvimento de um Protótipo de Aplicativo Móvel Multiplataforma para Agentes Comunitários de Saúde utilizando Xamarin

Gustavo Henrique Jacomel[1], Tony Alexander Hild[1]
gustavojacomel@hotmail.com, thild@unicentro.com.br

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *Atualmente, no âmbito da Saúde Pública, há regiões monitoradas por Agentes Comunitários de Saúde (ACS). Os ACS têm como trabalho visitar famílias da região e monitorar seus membros. Tal monitoramento é feito por meio de dados coletados em entrevistas e preenchidos nas Fichas A. O passo seguinte, realizado nas Unidades Básicas de Saúde, consiste em digitalizar no sistema os mesmos dados preenchidos na ficha física. Este trabalho propõe a prototipação de um aplicativo móvel multiplataforma, utilizando Xamarin, um framework desenvolvido pela Microsoft. Esse aplicativo pretende melhorar a velocidade e a qualidade do trabalho dos ACS.*

Palavras-chave: Agente Comunitário de Saúde; Aplicativo Móvel; Multiplataforma; Xamarin; Unidade Básica de Saúde.

1. Introdução

É dever do Estado, garantido pela Constituição, oferecer tratamento de saúde pública para todos os cidadãos [1]. Atualmente, no Brasil, o Sistema Único de Saúde (SUS) deve oferecer estruturas e funcionários dedicados a desempenhar funções referentes à saúde pública.

Dentre esses funcionários estão os Agentes Comunitários de Saúde (ACS), que têm como papel realizar visitas em domicílios atendidos pelas Unidades Básicas de Saúde (UBS). Nessas visitas são levados livros, contendo fichas, sendo a principal a “Ficha A”, utilizada para armazenar dados de endereço, membros da família, idade, escolaridade, doenças, condição especial, dentre outros. A Ficha A é carregada de maneira física e preenchida a cada visita. Finalmente, após coletar os dados em fichas físicas nas visitas domiciliares, os ACS têm que digitalizá-los novamente nas UBS, o que gera uma considerável perda de tempo e produtividade.

Tendo em vista esse problema, este trabalho tem como objetivo desenvolver um

aplicativo móvel multiplataforma, a partir da plataforma Xamarin. Tal aplicativo móvel deve conter os mesmos campos contidos nas fichas de registro utilizadas pelos ACS. Os dados coletados serão salvos em um banco de dados local e, assim que uma rede de internet estiver disponível, essas informações serão enviadas a um servidor para a persistência dos dados.

2. Materiais e métodos

2.1. Microsoft.NET

É um *framework* desenvolvido pela Microsoft, com foco para o sistema operacional Windows. Possui uma grande biblioteca de classes, chamada de *Framework Class Library* e permite a interoperabilidade entre diferentes linguagens de programação. O servidor, desenvolvido em um trabalho a parte, utiliza o .NET.

2.2. Xamarin

Xamarin é uma plataforma que permite o desenvolvimento de aplicativos móveis multiplataforma, para os sistemas operacionais Android, IOS e Windows Phone. O desenvolvimento é feito através de um único projeto, que utiliza a linguagem C# e bibliotecas fornecidas pelo próprio Xamarin. Também oferece ferramentas para o teste e *debug* dos aplicativos. Pode-se desenvolver aplicativos a partir do Xamarin Studio ou de uma extensão para Visual Studio.

2.3. SQLite

SQLite é uma biblioteca que implementa um motor transacional SQL embarcado, sem servidor, e que não necessita de configuração. Segundo o seu site oficial, é a base de dados mais utilizada no mundo, sendo parte de grandes projetos [2].

2.4. Newtonsoft JSON.NET

Newtonsoft JSON.NET é um componente para aplicações .NET que permite a serialização e desserialização de qualquer objeto para o formato JSON. Este, por sua vez, é um formato de texto independente de linguagem, tornando-o, assim, ideal para a troca de dados [3]. O formato JSON é utilizado na comunicação do cliente com o servidor.

3. Resultados e discussão

Com o desenvolvimento do trabalho obteve-se uma versão funcional do aplicativo que cumpre todos os objetivos propostos. A Ficha A contém campos para o cadastro de famílias, pessoas ligadas a essa família e visitas. Todos esses registros podem ser realizados digitalmente no aplicativo. Os ACS também podem consultar no mapa onde a casa da família

pesquisada está localizada para se dirigir ao local com maior facilidade.

A comunicação com o servidor para a persistência dos dados também acontece durante operações de salvamento quando há comunicação com a rede, porém, se esta não estiver disponível, os dados são armazenados no banco de dados interno e quando novos dados forem salvos com acesso à rede, todos serão enviados de uma só vez ao servidor.

Foram realizados dois testes informais sobre o aplicativo, com usuários que não são ACS. Em ambos os testes os usuários navegaram entre as páginas preenchendo os campos pertinentes à Ficha A e depois visualizaram no mapa. A comunicação com o servidor foi totalmente transparente, contendo apenas mensagens de sucesso se as informações fossem salvas e, erro, caso o servidor não pudesse ser alcançado durante aquela operação. Em ambos os casos não houve nenhuma grande dificuldade na utilização do aplicativo. Pode-se afirmar, portanto, que a Interface do Usuário desenvolvida atende aos objetivos do protótipo.

4. Considerações finais

A informatização da Ficha A é um bom começo para melhorar o serviço prestado pelo ACS e por todo o Sistema de Saúde Brasileiro. Com uma interface de fácil utilização, comunicação transparente com o servidor e rapidez nas operações, o protótipo do aplicativo foi desenvolvido sem apresentar problemas.

Como trabalhos futuros, propõe-se a integração de todas as fichas utilizadas pelos ACS no aplicativo, como acompanhamento de gestantes, hipertensos e crianças menores de 2 anos. Ao observar outros aplicativos no mercado, é interessante notar que os mais bem avaliados tem comunicação com o sistema eSUS AB das UBS, permitem o login do ACS pelo seu Conselho Nacional da Saúde (CNS) e cadastram pessoas vinculadas a um cartão do SUS. Porém, além de olhar as funcionalidades dos concorrentes, é preciso também fazer mais, portanto, a comunicação constante com a Equipe de Saúde da Família e os testes presenciais são obrigatórios.

Referências

[1] Brasil (1988). Constituição da República Federativa do Brasil. Senado, Brasília, DF. Constituição (1988).
[2]SQLite (2017). About sqlite. Disponível em https://sqlite.org/about.html. Acesso em 2/06/2017.
[3] Newtonsoft (2017). JSON.NET. Disponível em http://www.newtonsoft.com/ json. Acesso em 9/5/2017.
[4] Xamarin (2017). Documentação do Xamarin. Disponível em https://www. xamarin.com/. Acesso em 2 jun. 2017.

Jornada de Atualização em Informática da Unicentro
13, 14 e 15 setembro 2017
UNICENTRO PARANÁ DECOMP

Estimativa de dormência em frutíferas de clima temperado

Juliano Carvalho, Gabriel Fedacz, João Felipe Pizzolotto Bini, Marcos Antonio Quináia
julianocarvalho128@gmail.com, gabrielfedacz@gmail.com, joaofelipex85@gmail.com, quinaia@unicentro.br

Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *Este trabalho apresenta, resumidamente, um software chamado SleepTree, que tem como ideia principal calcular as unidades de dormência de árvores frutíferas. O software recebe algumas temperaturas diárias e, por intermédio de interpolação, calcula as temperaturas do restante do dia. Após calcular as temperaturas, o software utiliza fórmulas matemáticas para somar a quantidade de unidades de frio obtidas no dia. No processo de desenvolvimento do software foram usados os seguintes recursos: linguagem de modelagem UML, padrão de arquitetura de software MVC, banco de dados SQL e linguagem JAVA. O software foi implantado e sua validação foi realizada em um trabalho de mestrado em agronomia.*

Palavras-chave: Dormência de Gemas; Horas de Frio; Fruticultura; Software.

1. Introdução

As plantas, para sobreviverem a períodos de estresse, como ocorre quando há baixas temperaturas hibernais, desenvolveram um mecanismo adaptativo caracterizado pela aquisição da resistência ao frio e do controle do crescimento, denominado dormência. Este trabalho apresenta, resumidamente, um software chamado SleepTree, que tem como ideia principal calcular as unidades de dormência de árvores frutíferas. O software recebe algumas temperaturas diárias e, por intermédio de interpolação, calcula as temperaturas do restando do dia. Após interpoladas as temperaturas, o software calcula as unidades de frio, seguindo os seguintes métodos: Horas de Frio Ponderadas, Utah e Carolina do Norte. O software foi desenvolvido para atender uma demanda oriunda do Departamento de Agronomia da Universidade Estadual do Centro-Oeste, UNICENTRO, e sua validação foi realizada em um trabalho de mestrado em agronomia.

2. Materiais e métodos

No processo de desenvolvimento do software foram usados os seguintes recursos: linguagem de modelagem UML, padrão de arquitetura de software MVC, banco de dados SQL e linguagem JAVA. Foram realizados o levantamento de requisitos do software, o cronograma de desenvolvimento, os casos de uso, os protótipos das janelas, os diagramas de sequência e de classes, bem como o estudo e projeto sobre: arquitetura do software, manutenibilidade, evolução, banco de dados, casos de testes e confiança e proteção do software. Devido às limitações do modelo de submissão, são mostrados, resumidamente, neste trabalho, apenas algumas partes do projeto desenvolvido.

O projeto envolveu a utilização de várias ferramentas que oferecem recursos para as atividades do desenvolvimento de software, como a linguagem de modelagem, linguagem de programação, ferramentas de trabalho colaborativo, IDE e outras, citadas a seguir: desenvolvimento do software e geração de protótipos de janelas, Netbeans [1]; desenvolvimento dos diagramas ER de banco de dados, BrModelo [2] e Microsoft Visio [3]; criação de tabelas e campos do banco de dados, MySQL Workbench [4] e Postgree [5]; criação de cronograma, Microsoft Project Manager [6]; criação de algoritmos de teste, CodeBlocks [8].

Tabela 1 - Métodos de cálculos de horas de frio

Modelo HF Ponderadas		Modelo de Utah		Modelo Carolina do Norte	
Temperatura (°C)	Unidade deFrio	Temperatura (°C)	Unidade de Frio	Temperatura (°C)	Unidade de Frio
3	0,9	≤ 1,4	0,0	<-1,1	0,0
6	1,0	1,5 a 2,4	0,5	1,6	0,5
8	0,9	2,5 a 9,1	1,0	7,2	1,0
10	0,5	9,2 a 12,4	0,5	13,0	0,5
		12,5 a 15,9	0,0	16,5	0,0
		16,0 a 18,0	-0,5	19,0	-0,5
		>18,0	-1,0	20,7	-1,0
				22,1	-1,5
				>23,3	-2,0

3. Resultados e discussão

Após a etapa de projeto, o software foi desenvolvido. A Figura 1 mostra a interface principal do software, com fácil acesso aos menus que são utilizados para a criação de áreas, manipulação das temperaturas, geração de relatórios, importação e exportação de dados, além da ajuda ao usuário.

O software foi implantado no ambiente do Departamento de Agronomia da UNICENTRO e validado por meio da utilização em um trabalho de mestrado. Foram feitos diversos cálculos de interpolação de temperaturas em áreas de cultivares da região.

Figura 1 - Janela Principal

Foi possível obter com boa aproximação da quantidade de frio hibernal, utilizando temperaturas estimadas através de temperaturas (diárias) mínimas, máximas e das 21 horas.

4. Considerações finais

O programa computacional SleepTree representa uma nova e eficiente ferramenta de estimativa de frio hibernal, facilitando os cálculos e popularizando o uso dessa tecnologia, tanto para produtores rurais quanto para pesquisadores, permitindo a quantificação, de forma simples, do frio nas diferentes condições climáticas sul brasileiras, possibilitando aos usuários a obtenção de informações para o subsídio na tomada de decisões.

Referências

[1] Netbeans, Disponível em <https://netbeans.org/>. Acesso em 29 de agosto 2014.

[2] BrModelo, Disponível em <http://sis4.com/brModelo/>. Acesso em 29 de agosto 2014.

[3] Microsoft Visio, Disponível em <http://office.microsoft.com/pt-br/visio/>. Acesso em 29 de agosto 2014.

[4] MySQL Workbench, Disponível em <http://www.mysql.com/products/workbench/>. Acesso em 29 de agosto 2014.

[5] Postgree, Disponível em <http://www.postgresql.org/>. Acesso em 29 de agosto 2014.

[6] Microsoft Project Manager, Disponível em <http://office.microsoft.com/pt-br/project/>. Acesso em 29 de agosto 2014.

[8] CodeBlocks, Disponível em <http://www.codeblocks.org/>. Acesso em 29 de agosto 2014.

Estudo de Paralelização da Evolução Diferencial Aplicada em Problemas *Benchmarks*

Daniel Lucas de Paula[1], Sandra M. Venske[1]
dl1808@hotmail.com, ssvenske@unicentro.br

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *A computação paralela utiliza-se de múltiplos processadores simultâneos para a resolução de problemas. Nesse contexto, este trabalho propõe uma investigação da aplicação da programação paralela, com o uso da técnica evolucionária Evolução Diferencial, aplicada em problemas benchmarks. Para a realização da paralelização do código foi utilizado o modelo de programação OpenMP. Os resultados foram favoráveis à versão paralelizada, associados a um bom speedup.*

Palavras-chave: OpenMP; Evolução Diferencial; Otimização.

1. Introdução

A cada ano, observa-se uma tendência de crescimento na demanda computacional, a fim de resolver grandes problemas. A resolução dos problemas, utilizando programas computacionais sequenciais, desenvolve um algoritmo com um fluxo serial de instruções. Num programa sequencial, uma unidade central de processamento de um computador executa instruções uma a uma. Nesse tipo de programa é executada uma instrução por vez, sendo que depois do seu término a próxima instrução é executada [2, 6]. Muitas aplicações têm mostrado que algoritmos evolucionários paralelos podem acelerar a computação e encontrar melhores soluções, quando comparados com algoritmos evolucionários sequenciais. Algumas aplicações foram propostas utilizando-se o método de Evolução Diferencial [1, 4].

Nesse contexto, este trabalho propôs a aplicação de programação paralela em um código desenvolvido na Linguagem C, aplicação que se refere ao método evolucionário da Evolução Diferencial, aplicado em problemas *benchmarks*. Para a realização da paralelização do código foi utilizado o modelo de programação OpenMP.

2. Materiais e métodos

OpenMP é um modelo de programação *multithreading* de memória compartilhada. *Thread* é uma forma de um processo dividir a si mesmo em duas ou em mais tarefas que

podem ser executadas concorrencialmente. No OpenMP, as *threads* se comunicam utilizando variáveis compartilhadas. Esse modelo trata de um conjunto de diretivas do compilador e bibliotecas de rotinas para aplicações paralelas [6, 7].

Inicialmente, no procedimento do algoritmo paralelizado proposto, ocorre a inicialização dos números aleatórios paralelos, dos parâmetros da Evolução Diferencial (ED), inicialização e avaliação da população inicial. Foi utilizado o modelo LCG (*Linear Congruential Generator*) para gerar os números paralelos [5]. Em seguida, o algoritmo entra no laço de comunicação e sincronização e, então, na região paralela. Após isso, faz a divisão da população inicial no número total de *threads*. No laço principal, são realizados os processos mutação, cruzamento e seleção. O laço que controla a comunicação entre as *threads* é executado NTC vezes, sendo que NTC é o número total de comunicações. Em cada comunicação é realizada a ordenação da população de cada *thread* e, então, é feita a troca de informações entre as *threads*. Neste passo, os três melhores indivíduos de uma *thread* são copiados no lugar dos piores indivíduos da outra *thread*, seguindo o modelo de anel. O critério de parada do algoritmo é atingir o número máximo de avaliações e de comunicações.

3. Resultados e discussão

O algoritmo evolucionário paralelo proposto foi testado com 30 execuções independentes, considerando uma instância *benchmark* do CEC-2015, chamada de *Schwefel* [3]. A dimensão do espaço de busca foi definida como 30 e os intervalos considerados foram [-100, 100]. Os parâmetros usados foram: número da população 2.000; taxa de cruzamento 1,0; fator de mutação 0,5; número máximo de avaliações 1.000.000; número de *threads* 1, 2 e 4; número total de comunicações 20; número de indivíduos trocados entre as *threads* 3.

A Tabela 1 mostra os resultados obtidos pelo algoritmo proposto, considerando-se a instância *Schwefel*. A qualidade das soluções se mantém, quando comparadas à versão sequencial (uma *thread*) e às versões paralelas (2 e 4 *threads*).

Tabela 1: Resultados obtidos pelo algoritmo proposto, considerando-se a instância Schwefel.

Número de *Threads*	***Fitness* Médio (em minutos)**	**Desvio Padrão**	**Menor *Fitness***	***p-value***	***Speedup***
1	4689,81	493,28	3420,25		
2	4496,79	305,41	3939,63	0,00142	1,91
4	4389,23	238,81	3903,43	0,00010	2,07

O valor de *speedup* (métrica de desempenho para algoritmos paralelos) do tempo médio de execução foi de 1,91 em relação ao algoritmo sequencial e ao paralelo com duas *threads*. Considerando quatro *threads*, o *speedup* alcançado foi de 2,07. Para analisar os

resultados estatisticamente, o teste *Mann-Whitney-Wilcox* foi utilizado, sendo considerado um nível de confiança de 95% (*p-value* abaixo de 0,05), o qual mostra que os algoritmos com 2 e 4 *threads* são estatisticamente melhores que o algoritmo sequencial.

4. Considerações finais

Neste trabalho foi realizado o desenvolvimento de uma aplicação paralela em um código com abordagem de Evolução Diferencial escrito em Linguagem C.

Os resultados obtidos mostraram que o *speedup* adquirido no algoritmo paralelo teve um ganho em relação ao algoritmo sequencial. Outra contribuição do trabalho é que o algoritmo proposto pode ser estendido e utilizado em outros algoritmos evolucionários.

Como trabalhos futuros, pretende-se utilizar e testar outros métodos e modelos de paralelização, a fim de otimizar esse tipo de aplicação evolucionária.

Referências

[1] ALBA, E. (2005). Parallel metaheuristics: a new class of algorithms, volume 47. JohnWiley & Sons.

[2] BARNEY, B. (2010). Introduction to parallel computing. livermore computing. *Online*: http://www. llnl. gov/computing/tutorials/parallelcomp.

[3] CHEN, Q., LIU, B., ZHANG, Q., LIANG, J., SUGANTHAN, P., AND QU, B. (2014). Problem definitions and evaluation criteria for cec 2015 special session on bound constrained single-objective computationally expensive numerical optimization. Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University.

[4] ENGELBRECHT, A. P. (2007). Computational intelligence: an introduction. John Wiley & Sons.

[5] FONTAINE, C. (2005). Linear congruential generator. In Encyclopedia of Cryptography and Security, pages 350–350. Springer.

[6] MATTSON, T. AND MEADOWS, L. (2009). A “hands-on” introduction to openmp. Intel Corporation.

[7] TORELLI, J. C. AND BRUNO, O. M. (2004). Programação paralela em smps com openmp e posix *threads*: Um estudo comparativo. In Anais do IV Congresso Brasileiro de Computação (CBComp), volume 1, pages 486–491.

Estudo sobre Automatização de Equivalência de Funções

Lucas Fernando Frighetto[1], Fábio Hernandes[2]

lucas.frighetto@gmail.com, hernandes@unicentro.br

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *Neste trabalho é proposto um algoritmo para demonstrar a equivalência entre funções, sendo utilizadas transformações por meio da indução matemática. O algoritmo apresentado foi desenvolvido manipulando strings e utilizando as operações matemáticas de números naturais, que são: distributiva, associativa, comutativa e mínimo múltiplo comum. Como resultado, foram comprovadas as equivalências de um conjunto de funções constantes em um banco de dados.*

Palavras-chave: Números Naturais; Operações Matemáticas; Indução Matemática.

1. Introdução

Um dos principais problemas da Computação está na otimização automática de códigos, sendo que os compiladores atuais possuem diversas técnicas para otimizá-los, porém, com algumas dificuldades. A principal dificuldade está em descobrir como transformá-los, preservando seus significados e reduzindo o tempo computacional [1]. Algumas transformações são triviais, como código morto e recálculo de funções, porém outras são mais difíceis, como, por exemplo, transformar um *bubblesort* em *quicksort*. Existem várias formas pelas quais um compilador pode efetuar melhorias num programa preservando a sua função, os casos mais comuns de melhorias que preservam o significado do código podem ser encontradas em Aho *et al*. (2008), entre outros exemplos na literatura, destaca-se a otimização de laços [2].

Seguindo a temática de otimização de códigos, este trabalho tem por objetivo o desenvolvimento de uma técnica capaz de comprovar a equivalência entre funções, envolvendo números naturais, a qual poderá ser aplicada à otimização de código, pois, se provada a equivalência entre duas funções, o compilador poderá substituir uma função

recorrente de alto custo por outra função fechada de custo computacional menor. A fim de validar a técnica proposta, foi desenvolvido um algoritmo, descrito na Seção 2.

2. Algoritmo proposto

O algoritmo proposto procura a igualdade entre uma *P(n),* associado às funções recorrentes, e uma função *f(n)*, sendo composto dos passos: (i) dados de entrada – *f(n)* e o termo necessário da *P(n)* que é *P(n+1)-P(n)*; (ii) determinação de *f(n+1)* a partir da função *f(n)*, criada com a substituição de *n* por *n+1*; (iii) aplicação da operação distributiva na função e realização da soma da função *f(n)* inicial com o termo da *P(n)*, sendo executada uma operação de mínimo múltiplo comum, seguida da distributiva e da soma dos termos; (iv) se o resultado do passo (iii) for igual ao obtido por meio da substituição de *n* por *n+1*, seguido da distributiva, então conclui-se que a equivalência é verdadeira. Os passos descritos são destacados na Figura 1.

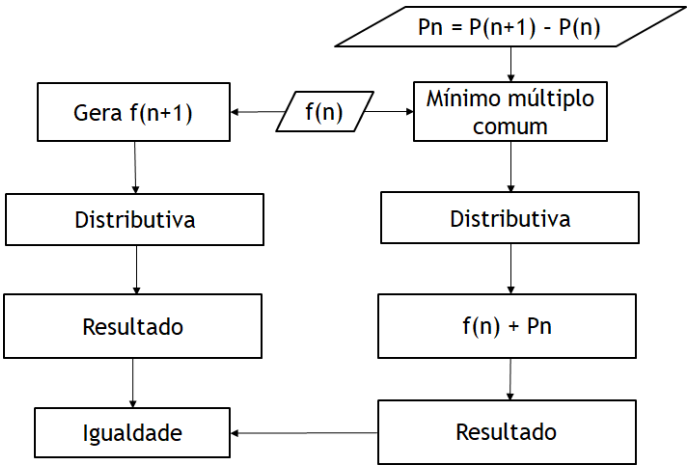

Figura 1. Passos do algoritmo proposto.

3. Resultados e discussão

O algoritmo proposto foi implementado utilizando a linguagem JAVA, pois esta possui uma biblioteca de *strings* que oferece muitos recursos para facilitar a implementação de operações de propriedades algébricas.

Para efeitos de validação dos resultados, neste trabalho, o algoritmo foi testado para a soma dos *(n+1)* números ímpares. Sendo $P(n): 1+3+...+(2n-1)=n^2$, somando *(2n+1)* em ambos os lados, tem-se: $P(n+1): 1+3+...+(2n-1)+(2n+1)=n^2+2n+1$.

O Quadro 1 apresenta os resultados obtidos pelo algoritmo para a função destacada.

f(n) = ((1n)(1n))/(1)
f(n+1) = (((1)(1n+1))((1)(1n+1)))/(1)
pn = (((2)(1n+1)-1)/(1)
Após o mínimo múltiplo comum:
pn = (((1)(2)(1n+1)-1)/(1)
f(n) = ((1)(1n)(1n))/(1)
f(n) + p(n+1) = f(n+1):
((1)(1n)(1n))/(1)+(((1)(2)(1n+1)-1)/(1) = (((1)(1n+1))((1)(1n+1)))/(1)
distributiva: (((1)(1n+1))((1)(1n+1)))/(1)
((1n+1)(1n+1))/(1)
(1nn+2n+1)/(1)
f(n+1) = (1nn+2n+1)/(1)
distributiva: ((1)(1n)(1n))/(1)
((1n)(1n))/(1)
(1nn)/(1)
distributiva: (((1)(2)(1n+1)-1)/(1)
(((2)(1n+1)-1)/(1)
((2n+2-1)/(1)
f(n) = (1nn)/(1)
pn = ((2n+2-1)/(1)
f(n) + pn = (1nn+2n+1)/(1)

Quadro 1. Resultados do algoritmo considerando a soma dos (n+1) números ímpares.

4. Considerações finais

Neste trabalho, estudou-se como automatizar a equivalência entre funções com números naturais. Como contribuição, foi proposto um algoritmo com a possibilidade de otimizar funções recorrentes, fazendo uso das operações básicas da matemática elementar e da indução matemática para provar a equivalência entre funções.

A dificuldade encontrada foi a análise e manipulação de expoentes e somatórios complexos, devido à complexidade da análise léxica e semântica, por isso foram implementados métodos limitados às funções relativamente simples, sendo um primeiro passo.

Referências

[1] Aho, A. V. et al. **Compiladores – Princípios, Técnicas e Ferramentas**, Massachusets: Pearson Addison – Wesley, 2ª edição, 2008.

[2] Allen, F. O tratado fundamental sobre técnicas para otimização de loops – 1969. Disponível em: <http://www.francesallen.com/ - Acesso em: 07/07/2017.

Evolução Diferencial e Busca Local para a Predição da Estrutura de Proteínas

Lucas Cordeiro Siqueira[1], Sandra M. Venske[1]

llucas_csiqueira@hotmail.com, ssvenske@unicentro.br

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *A predição da estrutura de proteínas (Protein Structure Prediction - PSP) é um problema da Bioinformática cujo objetivo é encontrar a estrutura terciária de uma proteína com energia mínima. A estratégia de busca aproximada é adequada para esta investigação, já que o espaço conformacional para o referido problema é muito extenso. Essa característica incentiva a pesquisa utilizando abordagens de algoritmos evolucionários. Neste trabalho, propõe-se a investigação da abordagem evolucionária chamada Evolução Diferencial (ED) para tratamento do problema PSP, testando-se a técnica de busca local quasi-simplex e uma variação da mesma no processo de otimização.*

Palavras-chave: Predição da Estrutura de Proteínas; Evolução Diferencial; Bioinformática; Busca Local.

1. Introdução

A predição da estrutura de proteínas é um problema que, na abordagem *ab initio*, prediz a estrutura terciária de uma proteína baseada em sua estrutura primária, sem qualquer outra informação, *à priori.* O objetivo do PSP é encontrar a estrutura terciária com energia mínima [5]. Este problema faz parte do grupo de problemas NP-completo [1].

Com a utilização de um algoritmo de busca local, espera-se que a convergência e a diversidade das soluções possam melhorar. Neste trabalho, propõe-se a aplicação do método de busca local *quasi-simplex* [2] e uma variação dele [6], aliados ao algoritmo de evolução diferencial para o problema da predição da estrutura de proteínas.

2. Materiais e métodos

O PSP é um problema de minimização, com o objetivo de encontrar a estrutura

terciária com a energia mínima dentre um conjunto de conformações de uma proteína. Neste trabalho foi utilizada a abordagem *ab initio*. Esta representação oferece uma perspectiva mais próxima da realidade do processo de dobramento da estrutura das proteínas, pois essas podem se movimentar livremente no espaço conformacional. Para comparar as conformações encontradas pelo algoritmo com a estrutura original da proteína, comumente é utilizado o cálculo de *Root-Mean-Square Deviation* (RMSD), que, de maneira geral, mede o grau de semelhança entre duas estruturas [5].

Dentre os algoritmos evolucionários, destaca-se o algoritmo de Evolução Diferencial - ED [4], visando a busca por melhores resultados com uma abordagem um pouco diferente da utilizada por outros algoritmos evolucionários e nas estratégias de evolução. Inicialmente, a ED cria uma população inicial aleatória, composta por *NP* indivíduos espalhados pelo espaço de busca e avalia esses indivíduos de acordo com uma função de *fitness*. Em seguida, a população é modificada a cada iteração, passando por três operadores da ED, sendo: mutação, cruzamento e seleção para chegar na próxima geração [3].

No método *Quasi-Simplex,* inicialmente, os *n* melhores indivíduos da população da ED são escolhidos para serem utilizados no cálculo do centroide. Na sequência, são aplicadas transformações ao conjunto de indivíduos, quais sejam: Reflexão, Expansão e Compressão; com duas operações para cada transformação. Todo esse processo é repetido até que um critério de parada seja atendido, sendo que, neste trabalho, esse critério corresponde a 50 repetições. Ao final, as seis melhores soluções obtidas são retomadas pelas buscas locais e introduzidas na população da ED para o PSP, substituindo as piores soluções.

A Variação do *Quasi-Simplex,* usada neste trabalho, foi proposta por Zapotecas-Martinez e Coello [6]. As transformações são: Reflexão, Expansão, Contração Externa e Contração Interna.

3. Resultados e discussão

A análise dos resultados foi feita baseada em 30 execuções independentes dos algoritmos. Os resultados obtidos foram comparados em termos de energia potencial livre. São comparados os valores de energias potenciais encontradas, sendo que tanto a Variação, quanto o *Quasi-Simplex*, foram melhores que a ED básica, mas tiveram o mesmo valor

mínimo. Como a Variação do *Quasi-Simplex* é menos custosa, pois utiliza menos equações, esse foi o método escolhido para ser comparado com a literatura. A Variação do Método *Quasi-Simplex* mostrou-se um bom otimizador e obteve o melhor resultado em termos de minimização nos valores de energia potencial em relação à literatura. Os valores de energia foram associados a bons valores de RMSD.

4. Considerações finais

O *Quasi-Simplex* original teve um desempenho um pouco melhor que sua variação em termos de média dos valores de energia encontrados, mas o melhor resultado dos dois algoritmos foi igual. Como a Variação do *Quasi-Simplex* é menos custosa que a original, pode-se concluir que os resultados foram satisfatórios em relação à literatura, principalmente em termos de energia potencial.

Como trabalhos futuros, pretende-se aplicar outras estratégias de Evolução Diferencial, como, por exemplo, *ED/rand/2*, *ED/current-to-rand/1*, entre outras, com a Variação do *Quasi-Simplex*. O algoritmo também pode ser testado utilizando-se outras proteínas.

Referências

[1] Crescenzi, P., Goldman, D., Papadimitriou, C. H., Piccolboni, A., and Yannakakis, M. (1998). On the complexity of protein folding (abstract). In *RECOMB*, pages 61–62.

[2] Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E. (1998). Convergence properties of the nelder–mead simplex method in low dimensions. S*IAM Journal on optimization*, 9(1):112–147.

[3] Rocha, N. C. and Saramago, S. d. F. P. (2011). Estudo de algumas estratégias de evolução diferencial. In *Congresso de Matematica Aplicada e Computacional*, CMAC Sudeste.

[4] Storn, R. and Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. 11(4):341–359.

[5] Tramontano, A. (2006). *Protein Structure Prediction: Concepts and Applications*. John Wiley and Sons.

[6] Zapotecas-Martinez, S. and Coello, C. A. C. (2016). Monss: A multi-objective nonlinear simplex search approach. *Engineering Optimization*, 48(1):16–38.

Evolução Gramatical e Evolução Diferencial Aplicadas à Solução do Despacho de Energia Elétrica

Tiago Remes Nunes[1], Ricardo Henrique Remes de Lima[2], Richard Aderbal Gonçalves[1]

{nunestiagorn, ricardo.hrlima}@gmail.com, richard@unicentro.br

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

[2]Universidade Federal do Paraná – UFPR

Resumo: *O Problema do Despacho Econômico de Energia Elétrica (PDEE) visa minimizar o custo de produção de energia de uma usina termoelétrica. Neste trabalho é utilizada a técnica de Evolução Gramatical para evoluir algoritmos de Evolução Diferencial (ED) para a solução do PDEE. A Evolução Gramatical se baseia numa gramática livre de contexto para gerar algoritmos (programas). Já a ED é uma técnica estocástica de otimização, baseada em população, desenvolvida para a otimização de valores reais. A Evolução Diferencial tem sido aplicada com sucesso na resolução do PDEE; contudo, o ajuste de seus parâmetros não é uma tarefa trivial, necessitando, tanto de conhecimento do problema, quanto da técnica em si. O uso da Evolução Gramatical deve permitir o ajuste automático dos parâmetros da ED. Os resultados numéricos obtidos foram analisados utilizando testes estatísticos. Os melhores algoritmos produzidos pela Evolução Gramatical foram interpretados de maneira a extrair conhecimento destes.*

Palavras-chave: Problema do Despacho Econômico de Energia Elétrica; Evolução Diferencial; Evolução Gramatical.

1. Introdução

O consumo de energia elétrica é vital para a sociedade, sendo necessário em quase todas as áreas de trabalho ou lazer. Com isso, o custo de produção de energia elétrica é muito importante, visto que impacta diretamente no custo de vida da população. Uma produção mais eficiente de energia pode colaborar para a obtenção de produtos e serviços menos onerosos, tendo impacto direto na competitividade da cadeia produtiva de uma nação.

O despacho econômico de energia elétrica tem por objetivo minimizar o custo operacional da produção de energia elétrica, atendendo a demanda gerada e levando em consideração restrições, como as capacidades de cada gerador [1]. Dadas as características do problema, o despacho econômico de energia elétrica não pode ser resolvido de maneira exata,

sendo necessário o uso de algoritmos aproximativos, dentre os quais se destacam os Algoritmos Evolucionários [2].

Portanto, o objetivo principal deste trabalho é a utilização da Evolução Gramatical para gerar algoritmos de Evolução Diferencial para solucionar o Problema do Despacho Econômico de Energia Elétrica.

2. Materiais e métodos

Inicialmente foi estudada a literatura relacionada ao Problema do Despacho Econômico de Energia Elétrica, visando a compreensão das principais características do problema, bem como quais são as melhores técnicas para a resolução desse. Na sequência, foram estudadas a Evolução Gramatical e a Evolução Diferencial. O estudo teve foco na compreensão dos aspectos teóricos das técnicas, visando utilizá-las de maneira eficiente na prática. Quanto à Evolução Gramatical, foi necessário o estudo de gramáticas livres de contexto.

O próximo passo do trabalho deu-se na proposta de uma gramática para a representação de algoritmos de Evolução Diferencial. Essa etapa foi de suma importância, visto que a proposta da gramática tem impacto direto, tanto na qualidade das soluções encontradas, quanto na interpretabilidade dos algoritmos gerados.

A análise dos resultados foi dividida em duas etapas: 1. Análise estatística dos resultados obtidos para o despacho econômico de energia elétrica; 2. Interpretação dos algoritmos gerados.

3. Resultados e discussão

O Problema do Despacho Econômico de Energia Elétrica tem sido muito estudado devido a sua importância no projeto e operação de sistemas de geração de energia elétrica. Diversos métodos têm sido usados para a resolução desse problema, dentre eles, vários algoritmos evolutivos foram empregados, como, por exemplo, a Evolução Diferencial.

Este trabalho propôs a utilização da Evolução Gramatical para gerar os parâmetros da Evolução Diferencial para a resolução do Problema do Despacho Econômico de Energia Elétrica. Foram propostas duas diferentes gramáticas para a geração dos parâmetros. A etapa de experimentos foi dividida em duas: uma para definir qual a melhor gramática e outra para comparar a melhor gramática com os resultados reportados na literatura.

A partir dos resultados obtidos com os experimentos foi possível concluir que a Evolução Gramatical, juntamente à Evolução Diferencial, conseguem gerar algoritmos que

possuem resultados satisfatórios para resolverem o Problema do Despacho de Energia Elétrica, inclusive com resultados estatisticamente melhores que os reportados na literatura.

4. Considerações finais

Este trabalho propôs a utilização da Evolução Gramatical para gerar os parâmetros da Evolução Diferencial para a resolução do Problema do Despacho Econômico de Energia Elétrica. Para isso, foi necessário definir gramáticas com diferentes e possíveis parametrizações da Evolução Diferencial. A criação e otimização desses algoritmos deram-se por meio da utilização do pacote R, chamado GramEvol, que realiza a implementação da abordagem da Evolução Gramatical no ambiente da linguagem R, bem como os componentes necessários para facilitar sua aplicação, requerendo apenas a definição de poucos componentes, tais como a gramática que descreve o espaço de busca e a função que avalia as soluções (neste trabalho o retorno da Evolução Diferencial).

Dos resultados obtidos com os experimentos, foi possível concluir que a Evolução Gramatical, juntamente à Evolução Diferencial, conseguem gerar algoritmos que possuem resultados satisfatórios para resolverem o Problema do Despacho Econômico de Energia Elétrica, inclusive com resultados estatisticamente melhores que os reportados na literatura.

Referências

[1] Park, J. B. and Jeong, Y. W. An improved particle swarm optimization for economic dispatch with valve-point effect, 2006.

[2] Gaspar-Cunha, A., Takahashi, R., and Antunes, C. H. Manual de computação evolutiva e metaheurística. Imprensa da Universidade de Coimbra/Coimbra University Press, 2012.

Geração Parcial de Aplicações Android Usando a Arquitetura Dirigida a Modelos

Bruno Cecatto[1], Inali Wisniewski Soares[1]
bruno-ceccatto@hotmail.com, inali@unicentro.br

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *O uso crescente de dispositivos móveis exige um software de qualidade e que apresente baixo custo de desenvolvimento. Neste sentido, a abordagem Model Driven Architecture (Arquitetura Dirigida a Modelos – MDA) contribui oferecendo desenvolvimento de software baseado em modelos que propiciam essas características. Este resumo descreve uma transformação de modelos, no contexto da abordagem MDA, para gerar aplicações Android usando a MDA.*

Palavras-chave: *Model Driven Architecture; Android; Acceleo.*

1. Introdução

O uso intensivo de dispositivos móveis, como *smartphones* e *tablets,* exige uma crescente disponibilidade de aplicações para estes dispositivos [3]. Assim, é importante o uso de abordagens na Engenharia de Software, como a Arquitetura Dirigida a Modelos (*Model Driven Architecture* - MDA) [6], para reduzir o tempo de desenvolvimento de aplicações.

Um grande número de dispositivos utiliza o Sistema Operacional (SO) Android, que é livre e de código aberto [2]. O Android foi construído com o objetivo de permitir aos desenvolvedores criar aplicações móveis para explorar ao máximo os recursos disponibilizados por um dispositivo móvel [5].

Nesta pesquisa foi desenvolvido um aplicativo móvel Android utilizando-se duas abordagens de desenvolvimento: a tradicional; e a MDA, que gera código-fonte parcial por meio de uma transformação de modelos *Model-to-Text* (M2T).

2. Materiais e métodos

Nesta seção são apresentadas as tecnologias e metodologias empregadas durante o desenvolvimento deste trabalho, que segue as diretrizes da MDA [6], na qual as construções dos modelos são realizadas em diferentes níveis de abstração.

A linguagem de modelagem utilizada para construir os modelos foi a *Unified Modeling Language* (UML) [7]. Para o desenvolvimento da transformação de modelos foi utilizada a ferramenta Acceleo [1], que permite gerar automaticamente códigos em diferentes linguagens (JAVA, PHP e C#), utilizando modelos definidos, conforme os metamodelos UML [10], MOF [11] ou EMF [10].

Algumas etapas definidas no caso em estudo foram: a) definição dos requisitos do aplicativo; b) definição do diagrama de casos de uso; c) definição do Modelo Independente de Plataforma (*Platform Independent Model* – PIM), que é um modelo livre de detalhes de plataforma; d) desenvolvimento do aplicativo na forma tradicional de desenvolvimento; e) definição do Modelo de Plataforma (*Platform Model* – PM), cuja plataforma é o Android; f) definição do Modelo Específico de Plataforma (*Platform Specific Model* – PSM); g) definição da transformação de Modelos *Model-to-Text* (M2T).

3. Resultados e discussão

Estudou-se, neste trabalho, o uso da abordagem MDA em aplicativos móveis. Definiu-se uma transformação de modelos M2T, que envolveu a transformação de um PSM para gerar o código para as características estruturais do aplicativo móvel. Foram definidos *templates* em Acceleo e os modelos definidos em UML foram utilizados para gerar códigos das classes do domínio e de acesso a dados.

Para apresentar o resultado do estudo foi definido um caso de estudo envolvendo aspectos práticos de desenvolvimento de aplicativos móveis e utilizando a plataforma Android.

O aplicativo selecionado para ser desenvolvido nas duas abordagens fornece ao usuário maior agilidade e comodidade para gerenciar as lojas que comercializam as cartas do jogo Yu Gi Oh, que é um jogo de cartas produzido pela empresa Konami [8]. Além disso, o aplicativo apresenta a localização destas lojas em um mapa.

Assim, foram desenvolvidos os modelos necessários e definidas as transformações M2T, com o uso da ferramenta Acceleo, obtendo-se, desse modo, um desenvolvimento parcial do aplicativo móvel.

4. Considerações finais

O desenvolvimento de aplicativos móveis apresenta vários desafios. A introdução e a evolução de tecnologias exigem requisitos e restrições adicionais. O desenvolvimento e tratamento da Interface Gráfica do Usuário e de recursos limitados exige constante aprimoramento das abordagens de desenvolvimento desses aplicativos.

Neste trabalho, verificou-se, na prática, que os Modelos PM e PSM, definidos para o aplicativo atual, podem ser utilizados na construção de outros aplicativos da mesma plataforma, o que resulta, portanto, em aumento de produtividade e qualidade com a utilização da abordagem MDA.

Para trabalhos futuros há novas transformações de modelos em aplicativos móveis que podem ser investigadas e definidas, como transformações M2M, utilizando-se transformações entre modelos PSMs, refinando-os e contribuindo para o processo de automação deste tipo de software.

Referências

[1] ACCELEO. Obeo. Disponível em: http://www.eclipse.org/acceleo/. Acesso em: 03/04/ 2017.

[2] ANDROID (2017). Disponível em: http://www.android.com/. Acesso em: 03/06/2017.

[3] BENOUDA, H., ESSBAI, R., AZIZI, M., MOUSSAOUI, M. Modeling and code generation of android applications using acceleo. International Journal of Software Engineering and Its Applications, v. 10, n. 3, p. 83–94, 2016.

[4] EMT (2017). Eclipse modeling tool. http://www.eclipse.org/downloads/ packages/eclipse-modeling-tools. [Online. Acessado em 18 de junho de 2017].

[5] LECHETA, R. Google Android – Aprenda a criar aplicações para dispositivos móveis com o Android SDK. Novatec, 2015.

[6] OMG, (2003). Technical Guide to Model Driven Architecture: The MDA guide v1.0.1. Disponível em http://www.omg.org/docs/omg/03-06-01/PDF. Object Management Group, 2003. Acesso em 04/02/2017.

[7] OMG (2015). Unified Modeling Language (UML). Disponível em http://www.omg.org/spec/UML/2.5/PDF. Object Management Group, 2015. Acesso em 06/03/2017.

[8] KONAMI (2017). https://www.konami.com. [Online. Acessado em 18 de junho de 2017].

[9] GLENDAY, C. (2011). Guinness World Records 2011. Guinness World Records. Random House Publishing Group.

[10] OMG (2015). Unified Modeling Language (UML). Disponível em http://www.omg.org/spec/UML/2.5/PDF. Object Management Group, 2015. Acesso em 06/03/2017.

[11] OMG (2016). Meta Object Facility (MOF. Mof Core Specification, version 2.5.1. Disponível em http: http//www.omg.org/spec/MOF/2.5.1/. Acesso em 05/05/2017.

Jornada de Atualização em Informática da Unicentro
13, 14 e 15 setembro 2017
UNICENTRO PARANÁ DECOMP

Levantamento Bibliográfico de Metodologias para Classificação de Folhas de Tabaco Utilizando Processamento Digital de Imagens

Charlie Felipe Liberati da Silva[1], Mauro Miazaki[1], Evanise Araujo Caldas[2]
{charlie.liberati, maurom}@gmail.com, EvaniseCaldas@ufgd.edu.br

[1]Universidade Estadual do Centro-Oeste – UNICENTRO
[2]Universidade Federal da Grande Dourados – UFGD

Resumo: *Existem diversos métodos de Processamento Digital de Imagens (PDI) aplicados em várias áreas da agricultura. Porém, aplicações de métodos direcionados à cultura do tabaco são pouco populares. A aplicação de métodos computadorizados de classificação de qualidade nessa cultura pode trazer melhorias na produção, somando ganhos de precisão e economia de recursos aos agricultores. Nessa perspectiva, este trabalho realiza um levantamento bibliográfico na literatura científica de métodos para classificação de folhas de tabaco utilizando PDI. Os métodos encontrados foram organizados em uma tabela comparativa e o melhor método foi indicado. Assim, este trabalho serve como ponto de partida para projetos futuros.*

Palavras-chave: Agricultura Familiar; Classificadores de Imagens; Folhas de Tabaco.

1. Introdução

A indústria do tabaco no Brasil, somente na Região Sul do país, envolve mais de 187 mil famílias, cerca de 742 mil pessoas no campo e 30 mil empregos diretos nas indústrias de beneficiamento[1], configurando uma indústria de grande importância econômica na Região Sul. No entanto, a classificação de folhas de tabaco geralmente é feita de forma manual, devido à diversidade de características a serem avaliadas e à complexidade de combinações de características, exigindo muito tempo e esforço humano. Essa tarefa manual, além de desgastante, limita a capacidade de produção e, por consequência, o desenvolvimento econômico dos produtores de tabaco [1]. Nesse contexto, aplicações com Processamento Digital de Imagens (PDI) podem auxiliar na realização automática da tarefa de análise e classificação das folhas, proporcionando aumento de produtividade, precisão e rentabilidade para o produtor [2].

[1] http://sinditabaco.com.br/

2. Materiais e métodos

Os artigos utilizados e descritos neste trabalho foram obtidos a partir de pesquisas executadas no portal de trabalhos científicos Scielo e nas ferramentas de busca Google Acadêmico e *ScienceDirect*. As palavras-chave em português utilizadas nas buscas foram: tabaco; processamento digital de imagens; classificação; automatização; e redes neurais. Já em inglês foram utilizadas: *digital image processing*; *tobacco grading*; *tobacco*; *classification*; *neural network classification*; e *automation*. Os artigos encontrados foram sumarizados em uma tabela, que apresenta características de cada método para facilitar a comparação e discussão dos resultados na Seção 3.

3. Resultados e discussão

Observando os métodos, resultados obtidos e considerações finais dos autores, nota-se a variedade de características avaliadas, classificadores empregados e medidas de avaliação, impactando no resultado final, ou seja, na acurácia da classificação. Os métodos e informações relevantes sobre as questões citadas estão dispostos na Tabela 1, ordenados da menor para a maior acurácia, calculada considerando as imagens para teste. A acurácia é o percentual de quantas amostras foram classificadas corretamente.

A qualidade das folhas de tabaco pode ser classificada em grupos, subgrupos e classes. Dos trabalhos encontrados, quatro foram realizados com classificação de qualidade. O método de Liu *et al*. [3] se destacou, apresentando acurácia de 93,5%. O trabalho de Guru *et al*. [4] obteve acurácia um pouco maior, 93,54%, mas investiga outro assunto, detecção de doenças. Esse trabalho foi incluído nesta pesquisa por utilizar técnicas de PDI em folhas de tabaco, o que pode trazer novas ideias para a implementação de um novo algoritmo.

A maioria dos algoritmos utiliza MLP. O trabalho de Liu *et al*. [3] é o único que utiliza técnicas de redução de dimensionalidade das características, o que pode ter sido importante no bom desempenho dos resultados.

4. Considerações finais

Os métodos encontrados trouxeram visão mais ampla e concreta da utilização de PDI na classificação de folhas de tabaco. Pretende-se, em um trabalho futuro, implementar um método de classificação capaz de classificar grupos e subgrupos de folhas utilizando características dos métodos descritos e sumarizados.

Tabela 1. Métodos encontrados e informações relevantes. Siglas: *Generalized Regression Neural Network* (GRNN); *Mean Impact Value* (MIV); Rede Neural *Artificial Multilayer Perceptron* (MLP); *Fuzzy Comprehensive Evaluation* (FCE).

Métodos e Características	**[1] Zhang *et al.* (2011)**	**[5] Zhang *et al.* (1997)**	**[2] Dachi *et al.* (2014)**	**[3] Liu *et al.* (2012)**	**[4] Guru *et al.* (2011)**
Número total de Imagens	150	135	60	108	100
Número de imagens para treino	100	25	20 imagens reais e 9 imagens geradas	81	Não informado
Número de imagens para teste	50	110	40	27	100
Classes	Apenas três classes: X1L, B4L e S1	Grupos, subgrupos e classes de folhas de tabaco	Grupos de folhas de tabaco	Grupos, subgrupos e classes de folhas de tabaco	Folhas sadias, com olho-de-sapo ou com antracnose
Aquisição de imagens	Câmera fotográfica digital em ambiente com iluminação controlada	Câmera fotográfica digital em ambiente com iluminação controlada	Câmera fotográfica digital em ambiente com iluminação controlada	Câmera fotográfica digital em ambiente com iluminação controlada	Câmera fotográfica digital em ambiente com iluminação controlada
Redução de dimensionalidade das características	Não usou	Não usou	Não usou	Características independentes identificadas via GRNN e selecionadas as que mais impactam na classificação usando MIV	Não usou
Método de classificação	MLP e FCE	Algoritmo próprio, baseado em cálculo de distância ponderada entre grupos	MLP	MLP	MLP
Acurácia	72%	83%	87%	93,50%	93,54%

Referências

[1] ZHANG, F. et al. Classification and quality evaluation of tobacco leaves based on image processing and fuzzy comprehensive evaluation. *Sensors*, v. 11, p. 2369-2384, 2011.

[2] DACHI, E.P. et al. Automatic classification of tobacco leaves using image processing techniques and neural networks. In: The 18th World Multi-Conference on Systemics, Cybernetics and Informatics. *Proceedings of the WMSCI 2014*. Florida: IIIS, 2014.

[3] Liu, J. et al. Grading tobacco leaves based on image processing and generalized regression neural network. In: Intelligent Control, Automatic Detection and High-End Equipment (ICADE). *Proceeding of the IEEE International Conference on…* New Jersey: IEEE, 2012.

[4] Guru, D.S. et al. Segmentation and Classification of Tobacco Seedling Diseases. In: 4th Bangalore Annual Compute Conference (COMPUTE 2011), Bangalore, India. *Proceedings of the...* New York: ACM, 2011.

[5] ZHANG, J. et al. A trainable grading system for tobacco leaves. Computers and Electronics in Agriculture, v. 16, p. 231-244, 1997.

Materialização de consultas SPARQL

João Dutra Cristoforu[1], Josiane Michalak Hauagge Dall'Agnol[1]

jd_cristof@hotmail.com, jhauagge@unicentro.br

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *Atualmente, a utilização de padrões e modelos de publicação de dados RDF permitem conexões entre bases de dados diferentes e a navegação por esses dados distintos, como se fossem de uma única fonte. Como essas interligações de grafos RDF (Resource Description Framework) geram grafos ainda maiores, as consultas SPARQL (SPARQL Protocol and RDF Query Language) tornam-se demoradas. Entretanto, muitos usuários não estão interessados no conjunto de dados completo, mas sim em partes específicas desse conjunto. Nesse sentido, esta pesquisa busca realizar a materialização de consultas SPARQL como forma eficiente de extração de subconjuntos de dados úteis, que correspondam às informações que os usuários realmente necessitam. O principal resultado obtido é o ganho de custo relacionado à diminuição do tempo de resposta.*

Palavras-chave: Dados Abertos Conectados; Consulta SPARQL; Materialização de Consultas.

1. Introdução

De acordo com Isotani e Bittencourt (2015), a *web* disponibiliza uma massiva quantidade de dados, sendo que grande parte desses dados não pode ser compreendida e aproveitada de forma automatizada. Logo, a extração de informações não se faz de forma ágil e eficiente.

Existem modelos e padrões para a preparação e disponibilização de dados na *web*, visando torná-los Dados Abertos e Conectados [2], permitindo realizar conexões entre bases de dados diferentes e navegar por esses dados como se fossem de uma única fonte.

A interligação dessas fontes de dados pode gerar grafos RDF ainda maiores, sobre os quais as consultas SPARQL podem levar um tempo demasiado de espera pela resposta.

Entretanto, muitos usuários não estão interessados no conjunto de dados completo, mas sim em partes específicas desse conjunto. De acordo com Ibragimov *et al*. (2016), o custo computacional é alto em consultas agregadas sobre *endpoints* SPARQL, especialmente se detalhes específicos do RDF precisarem ser considerados. Sendo assim, faz-se necessário o

emprego de técnicas que visem acelerar a execução das consultas agregadas. Nesse sentido, este trabalho propõe a materialização de consultas em Dados Abertos Conectados para a verificação do ganho de tempo na resposta de consultas realizadas em grandes conjuntos de dados modelados como grafos RDF.

2. Materiais e métodos

A fonte de dados utilizada foi o conjunto de dados *QualisBrasil* [3], contendo o histórico do índice Qualis do Sistema Brasileiro de Avaliação de Periódicos, no qual, buscando-se minimizar o tempo de recuperação do conjunto de dados, propõe-se a materialização de consultas SPARQL, criando-se visões de subconjuntos dos dados.

A materialização de consultas corresponde à geração de todas as combinações de campos por nível de granularidade, seguida pela compactação desses resultados. A materialização pode ser realizada por meio de um *script* que gera as combinações automaticamente em uma única vez ou sob demanda, na qual toda vez que uma consulta é solicitada é feita uma busca no conjunto de dados materializados. Caso o resultado não esteja armazenado, a busca é feita no conjunto de dados original e o resultado é então materializado.

Para o desenvolvimento do *script* de consultas SPARQL foram identificadas as dimensões do grafo RDF *QualisBrasil*, objetivando a construção de um esqueleto de consultas SPARQL, no qual são geradas todas as combinações possíveis de consultas, eliminando-se redundâncias (por exemplo: Título e Ano; Ano e Título). Esse *script* foi escrito em linguagem PHP[1] (PHP *HyperText Preprocessor*) e sua biblioteca ZipArchive[2].

O ganho de desempenho ocorre pelo fato de que as visões são bem menores que o conjunto de dados original. Porém, manter pré-calculadas todas as agregações de campos requer muito mais espaço do que o conjunto original consome. Sendo assim, é importante encontrar um conjunto adequado de visões materializadas para minimizar o tempo de resposta das consultas.

3. Resultados e discussão

Uma vez utilizado o *script* de materialização de consultas, mesmo que esse leve um tempo considerável para ser processado, esse custo de tempo é amortizado pelo fato de que

1URL: https://secure.php.net/manual/pt_BR/intro-whatis.php
2URL: http://php.net/manual/en/class.ziparchive.php

todas as próximas consultas realizadas sobre o conjunto materializado terão um tempo de resposta minimizado. Porém, a frequência de atualização da base de dados tem que ser baixa, pois a realização frequente do *script* de materialização pode tornar o custo de tempo elevado.

Ao utilizar a abordagem de materialização sob demanda, o custo de tempo será menor quando o conjunto de dados original possui alta frequência de atualização, como, por exemplo, quando os dados são atualizados semanalmente na base.

Alguns resultados obtidos relacionados ao tempo de resposta de consultas feitas diretamente na base de dados original, em comparação com o tempo de resposta sobre o conjunto materializado de dados, são apresentados na Tabela 1.

Tabela 1 – Comparativo de tempo de resposta entre consultas na base original e materializada.

Título	Score	Ano	ISSN	Área	Consulta à Base Original de Dados	Consulta aos Dados Materializados	Economia
X					5.4309s	1.9915s	63%
X	X				6.0033s	2.0015s	67%
X	X	X			6.7137s	2.0024s	70%
X	X	X	X		6.7035s	2.0055s	70%
X	X	X	X	X	10.5150s	2.3245s	78%

4. Considerações finais

É importante a busca por técnicas que visem acelerar o processo de requisição e consumo de dados, pois a sociedade precisa, progressivamente, de decisões rápidas sobre uma quantidade grande de dados.

A utilização de visões materializadas se mostrou eficiente porque nem sempre o usuário está interessado no conjunto de dados completo, mas sim em partes específicas desse conjunto. Mantendo-se os dados materializados, observou-se um ganho em tempo de resposta.

Referências

[1] IBRAGIMOV, D., HOSE, K., PEDERSEN, T. B. e ZIMÁNYI, E. Optimizing aggregate sparql queries using materialized rdf views. In *International Semantic Web Conference*, pages 341-359, 2016. Springer.

[2] ISOTANI, S. e BITTENCOURT, I. I. G. **Dados Abertos Conectados: Em busca da Web do Conhecimento**. 2009, Novatec.

[3] RAUTENBERG, S., MARX, E., ERMILOV, I., AND AUER, S. Linked data workflow project ontology: uma ontologia de domínio para publicação e preservação de dados conectados. **Tendências da Pesquisa Brasileira em Ciência da Informação**, 2017 9(2).

Protótipo de um Sistema Cliente/Servidor para Gerenciamento de Dados da Atenção Básica a Saúde

Paula Daiane Penteado Machula[1], Tony Alexander Hild[1]
pauladpm@outlook.com, thild@unicentro.com.br

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *O monitoramento da saúde pública é feito por Agentes Comunitários de Saúde (ACS) que se deslocam às residências familiares para coletar informações gerais sobre as condições de saúde dos indivíduos, inserindo tais informações em fichas de papel padronizadas, que são carregadas durante o monitoramento diário. Para melhorar e aprimorar o trabalho dos ACS e da coleta de dados, este trabalho apresenta o protótipo de um sistema de back-end cliente/servidor para coleta, processamento e armazenamento de dados de maneira ágil. A implementação foi feita na plataforma .NET Core, utilizando a linguagem C# e seguiu a arquitetura em camadas para atender os requisitos levantados.*

Palavras-chave: Agentes Comunitários de Saúde; Servidor; Gerenciamento.

1. Introdução

É tarefa do Sistema Único de Saúde (SUS) promover e proteger a saúde, assegurando que indivíduos com diferentes necessidades recebam atenção contínua e de qualidade. Parte dessa atenção é realizada pelos Agentes Comunitários de Saúde (ACS), que coletam dados e informações de famílias e indivíduos [1]. As informações coletadas são inseridas de maneira manual em fichas de papel, sendo possível registrar e acompanhar a situação dessas famílias.

Todo ACS carrega consigo as fichas necessárias para realizar o monitoramento de saúde em uma determinada região, o que acarreta carregar peso durante toda a trajetória, normalmente percorrida a pé. Após a realização da coleta de dados, os ACS se dirigem para as Unidades Básicas de Saúde (UBS) onde inserem os dados em um sistema informatizado interno.

O principal objetivo deste resumo é descrever o desenvolvimento de um protótipo de sistema de *back-end* cliente/servidor para gerenciar dados da Atenção Básica a Saúde, com o intuito de substituir a utilização das fichas físicas, evitando as dificuldades de carregá-las, facilitando a procura, o preenchimento e a organização dessas, assim como gerenciar os dados e informações relevantes inseridas nas fichas, melhorando, dessa forma, a eficiência da

utilização de recursos públicos, a qualidade de vida da população e as atividades rotineiras dos ACS.

Com a utilização do protótipo desenvolvido, espera-se que as atividades dos ACS seja simplificada, com o uso da informatização e com a melhoria do método de coleta de dados. Em consequência, espera-se aumentar a qualidade e confiabilidade dos dados necessários para a geração de informações gerenciais e de apoio à decisão, buscando a utilização eficiente dos recursos públicos.

2. Materiais e métodos

2.1. C# - Lê-se *C sharp*, é uma linguagem de programação orientada a objetos e utilizada para desenvolver aplicativos executados na plataforma .NET Core [2].

2.2. .NET Core – É uma plataforma gratuita para desenvolvimento de aplicativos a partir das linguagens de programação C# e F# [3].

2.3. ASP.NET Core – É um *framework* de código-fonte aberto e multiplataforma para desenvolvimento de aplicativos móveis, que podem ser executados no .NET Core [4].

2.4. Visual Studio Code (VSC) – É um software gratuito e multiplataforma para edição visual de código-fonte, conta com um conjunto de recursos e extensões para diversas linguagens [5].

2.5. SQLite – É um banco de dados multiplataforma e de domínio público que acessa arquivos de armazenamento sem a necessidade de intermédios [6].

2.6. Entity Framework (EF) Core – É um mapeador objeto-relacional multiplataforma que permite abstrair dos desenvolvedores detalhes sobre o banco de dados [7].

2.7. Modelo de Prototipação – É um Modelo de Processo Evolucionário iterativo e que evolui ao decorrer do desenvolvimento do protótipo [8].

2.8. Padrão MVC – É um padrão de arquitetura bastante utilizado para o desenvolvimento de aplicações web. O padrão divide a aplicação em três partes: Modelo, Visão e Controlador [9].

O desenvolvimento do protótipo foi feito na linguagem C# e *framework* Asp.Net Core, no editor VSC. Os dados são armazenados no banco SQLite, o qual foi criado e configurado automaticamente pelo EF. O protótipo foi desenvolvido seguindo o Modelo de Prototipação e a arquitetura do sistema seguiu o padrão MVC.

Para levantar os requisitos foi necessário fazer consultas em documentos do Ministério da Saúde, entrevistar os ACS da UBS Dourados, da cidade de Guarapuava-PR, para obter informações sobre como é feita a coleta de dados da população, além de acompanhar visitas domiciliares, juntamente aos ACS, para ver na prática como é feita a coleta de dados.

3. Resultados e discussão

Os requisitos levantados resultaram em uma lista de dados, informações cadastrais e registros necessários para realizar o cadastro de pessoas, famílias, agentes, visitas, áreas e unidades, bem como poder fazer o controle do registro das visitas realizadas e das responsabilidades de uma determinada área.

O protótipo contém os campos de dados necessários (presentes nas fichas de papel) para que o cadastramento e o acompanhamento da situação de saúde das famílias possa ser feito da melhor maneira possível. Pode-se realizar operações de criar, ler, editar e excluir dados, bem como realizar a validação e verificação desses, substituindo as fichas de papel utilizadas pelos ACS.

4. Considerações finais

Foram realizados estudos bibliográficos sobre a saúde brasileira, sobre o trabalho dos ACS e sobre sistemas distribuídos e tecnologias necessárias para o desenvolvimento do protótipo. Entrevistas com ACS e acompanhamentos em visitas domiciliares também foram feitas para que os requisitos pudessem ser levantados e o protótipo pudesse substituir, de forma prática, rápida e eficiente, as fichas de papel.

Considerando os resultados obtidos, é possível concluir que o protótipo do sistema tem potencial para substituir as fichas de cadastro e de acompanhamento da saúde da família brasileira, melhorar o trabalho rotineiro das visitas familiares feitas pelos ACS, otimizando, portanto, a forma de promover a saúde e prevenir doenças da população brasileira.

Referências

[1] World Health Organization. WHO | Constitution of World Health Organization. http://www.who.int/about/mission/en/. Acesso em 02 mai. 2017.

[2] Uma tour pela linguagem C#. https://docs.microsoft.com/pt-br/dotnet/csharp/tour-of csharp/index. Acesso em 15 mar. 2017.

[3] .NET Core. https://docs.microsoft.com/pt-br/dotnet/core/. Acesso em 19 fev. 2017.

[4] Introduction to ASP.NET Core. https://docs.microsoft.com/en-us/aspnet/core/. Acesso em 19 mar. 2017.

[5] Visual Studio Code. https://code.visualstudio.com/docs. Acesso em 11 mar. 2017.

[6] About SQLite. https://sqlite.org/about.html. Acesso em 2 jun. 2017.

[7] Entity Framework Core. https://docs.microsoft.com/en-us/ef/core/. Acesso em 2 jun. 2017.

[8] PRESSMAN, R. S. *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2010.

[9] MVC. https://msdn.microsoft.com/en-us/library/ff649643. Acesso em 11 mai. 2017.

Jornada de Atualização em Informática da Unicentro
13, 14 e 15 setembro 2017
UNICENTRO PARANÁ DECOMP

Redes Neurais Convolucionais na Classificação de Imagens: Levantamento Bibliográfico

Giovanni Klein Campigoto[1], Mauro Miazaki[1], Evanise Araujo Caldas[2]
giovanniklincampigoto@gmail.com, maurom@gmail.com, EvaniseCaldas@ufgd.edu.br

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

[2]Universidade Federal da Grande Dourados – UFGD

Resumo: *Redes Neurais Convolucionais (RNCs) têm se destacado nos últimos anos, principalmente na tarefa de classificação de imagens. Assim, buscando investigar as diversas aplicações de RNCs atualmente encontradas na literatura, o objetivo deste trabalho foi realizar um levantamento bibliográfico sobre a aplicação de RNCs na classificação de imagens. Os resultados foram sintetizados em uma tabela contendo informações sobre as aplicações encontradas. Destaca-se que os resultados são bastante positivos e promissores.*

Palavras-chave: Redes Neurais Convolucionais; *Deep Learning*; Classificação de Imagens.

1. Introdução

Um dos objetivos em Processamento Digital de Imagens é extrair dados para classificação de objetos em imagens. Para que essa classificação seja mais eficiente e em larga escala existem diferentes abordagens, como as Redes Neurais Artificiais [1].

O conceito de *Deep Learning* (Aprendizado Profundo) [2] consiste em uma rede neural "profunda", ou seja, com várias camadas, por isso esse nome. As Redes Neurais Convolucionais (RNCs) são redes de aprendizado profundo que, nos artigos pesquisados, foram utilizadas para a classificação de imagens como: folhas, objetos em geral, carros, placas, entre outros. A rede utiliza várias camadas para fazer uma "compressão" da imagem e extrair suas principais características, fazendo com que a classificação seja muito mais eficiente.

RNC é um método eficiente para realizar processamento de imagens, sendo demonstrado em vários *benchmarks* [3]. Apesar de ser uma técnica conhecida há bastante tempo, somente nos últimos anos as RNCs foram aperfeiçoadas o suficiente para obter ótimos

resultados. Com o propósito de investigar as atuais aplicações de RNCs na classificação de imagens, o objetivo deste trabalho foi fazer um levantamento bibliográfico a respeito.

2. Materiais e métodos

A metodologia de pesquisa foi a realização de um levantamento bibliográfico de artigos científicos relacionados à aplicação de RNCs na classificação de imagens. A ferramenta de busca Google Acadêmico foi utilizada, com as palavras-chaves: redes neurais convolucionais; redes neurais convolucionais folhas; redes neurais convolucionais imagens; *convolutional neural networks*; *convolutional neural networks plants*; *convolutional neural networks images*. Também foram analisadas as referências citadas nos artigos selecionados na busca.

3. Resultados e discussão

A Tabela 1 sumariza as aplicações encontradas. As referências correspondentes às citações na tabela podem ser encontradas no documento disponibilizado no Google[1].

Tabela 1. Tabela comparativa das aplicações encontradas. São informados: artigo em que foi publicado, aplicação, técnicas utilizadas, banco de dados (BD), número de amostras de treinamento (ATr), número de amostras de testes (ATe) e precisão dos resultados. "N" significa que os dados não foram reportados. Técnicas utilizadas: Camadas Convolucionais (CC), Max-Pooling (MP), GoogleLeNet (GLN), Dropout (D), AlexNet (AN), DeconvNet (DN), Softmax (S), MultiColumn Deep Neural Network (MC-DNN), Gaussianas (G), Mapas de Calor (MC), PReLU (P), Mixed Deep Convolutional Neural Network (MDCNN) e Transfer Learning (TL).

Artigo	Aplicação	Técnicas	BD	ATr	ATe	Precisão
Ciresan 2012	Classificação de caracteres, dígitos e imagens	CC, MP, MC-DNN	MNIST	N	N;	99,77 %
			NIST SD 19	N	82.000	88,37 %
			Chinese Characters ON	938.679	234228	94,39 %
			Chinese Characters OFF	939.564	234.800	93,5 %
			NORB	291.600	38.320	97,3 %
			Traffic Signs	26.640	12.569	99,46 %
			CIFAR 10	5.000	1.000	88,79 %
Ranzato 2007	Classificação de pólen e células da urina	CC, G	Células da urina	5640	360	84,5 %
			Airborne Pollen	N	3.686	93,9 %
Szegedy 2015	ILSVRC 2014	GLN, CC, MP, S	ILSVRC2014	50.000	100.000	43,9 %
Farfade 2015	Reconheci-mento facial com caixas	AN, CC, MP, S, MC	PASCALVOC	341	N	91,79%

1 https://sites.google.com/site/maurom/d/Referencias_Aplicacoes_RNCs_JAI2017.pdf

	de fronteira		AFW	473	N	96,26%
			FDDB	5.171	N	84 %
Lee 2015	Reconheci-mento de folhas	CC, MP, DN	Base de dados própria	34.672	8.800	99,5 %
Zhang 2016	Reconheci-mento de folhas	D, LRN, P, CC, MP, S	Flavia	1.586	320	94,68 %
Mohanty 2016	Reconheci-mento de doenças em culturas (folhas)	GLN	Plant Village	43.444	10.861	99,35 %
Ge 2016	Reconheci-mento Inter e Intra-classe (pássaros e flores)	MDCNN, GLN, TL	CUB200-2011	6.000	6.000	81,1 %
			Birdsnap	47.386	2.443	74,1 %
			PlantCLEF2015	25.025	3.200	52,1 %

As redes neurais convolucionais têm várias técnicas para realizar a "compressão" e a classificação das imagens. Cada técnica tem como objetivo construir um mapa de características, otimizar a imagem e classificá-la na camada final da rede. As bases de dados utilizadas são, em sua maioria, públicas, o que possibilita a replicação dos resultados. Na maioria dos casos a precisão possui valores altos e satisfatórios, segundo os autores.

4. Considerações finais

Os trabalhos encontrados evidenciam a utilização bem-sucedida de RNCs na classificação de diversos tipos de imagens, inclusive com os melhores resultados no estado da arte em alguns casos [3]. Portanto, os conceitos biológicos de redes neurais têm demonstrado sua efetividade na resolução de problemas envolvendo a classificação de imagens. Porém, muitos avanços são ainda necessários para aprimorar os algoritmos [4].

Referências

[1] BRAGA, A.P. et al. Redes neurais artificiais: teoria e aplicações. 2ª ed. LTC, 2007.
[2] FUKUSHIMA, K. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, vol. 36, n. 4, p. 193-202, 1980.
[3] CIRESAN, D. et al. Multi-column deep neural networks for image classification. In: *IEEE Conference on Computer Vision and Pattern Recognition*, p. 3642-3649, 2012.
[4] GE, Z.Y. et al. Finegrained classification via mixture of deep convolutional neural networks. In: *IEEE Winter Conference on Applications of Computer Vision*, p. 1–6, 2016.

SMLCompiler: Desenvolvendo um Protótipo de Compilador para a Linguagem *Sparqlification Mapping Language*

Bruno Francisco Passaglia[1]

bruno.passaglia2013@gmail.com

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *A publicação de dados na Web de Dados é uma tarefa complexa, sobretudo por ser necessário mapear dados brutos em recursos enriquecidos semanticamente. Uma das abordagens de mapeamento é a utilização da Sparqlification Mapping Language (SML) juntamente à ferramenta Sparqlify, utilizando-se de um arquivo de configuração. Porém, esse processo é suscetível a falhas. Visando auxiliar o engenheiro de dados na construção dos arquivos de configuração, propõe-se o desenvolvimento de uma abordagem tecnológica para analisar os arquivos SML e sugerir correções. Como resultado, foi desenvolvido um protótipo de compilador intitulado SMLCompiler, capaz de identificar erros léxicos e sintáticos.*

Palavras-chave: Compilador; *Sparqlify*; Web de Dados; SML.

1. Introdução

Apesar do avanço do conceito e das tecnologias referentes à Web Semântica, grande parte do conhecimento disponível ainda está na forma de base de dados relacional. Logo, boa parte do esforço da comunidade da Web Semântica reside em disponibilizar, em formato *Resource Description Framework* (RDF), o conhecimento que ainda se encontra nas bases de dados relacionais [1].

Algumas abordagens, que focam no mapeamento de dados relacionais para recursos RDF, foram desenvolvidas, entre elas, a *Sparqlify* [2]. Seu funcionamento consiste na entrada de um arquivo de configuração descrito em *Sparqlification Mapping Language* (SML), juntamente a uma massa de dados primários. Entretanto, o processo de mapeamento é suscetível a erros que, comumente, são detectados somente após a verificação dos recursos resultantes. Ressalta-se que a *Sparqlify* não oferece mensagens de erro adequadas. Diante disso, a identificação e correção de erros exige do engenheiro de dados um esforço cognitivo adicional, que poderia ser minimizado ao pré-processar o arquivo SML. Nesse sentido, o desenvolvimento de um protótipo de compilador foi proposto e desenvolvido neste trabalho, intitulado *SMLCompiler*.

2. Materiais e métodos

Aho *et al* [3] definem um compilador como um programa que lê um código escrito em uma linguagem (a linguagem fonte) e o traduz em um código equivalente em outra linguagem. Compiladores trabalham em fases. Cada uma dessas fases transforma a representação do programa-fonte de maneira que a próxima fase do processo possa utilizá-la. Para cada uma das fases existe um componente associado. Para este protótipo foram desenvolvidos três destes componentes: o analisador léxico, o analisador sintático e parte do analisador semântico.

Além da linguagem Java, foi utilizada a ferramenta JavaCC [4], a qual auxilia na geração dos analisadores léxico e sintático. A ferramenta permite realizar a definição de cada *token* na forma de expressões regulares e descrever a linguagem na forma de regras de produção.

3. Resultados e discussão

A fim experimentar o compilador desenvolvido, uma bateria de testes foi realizada. Foram gerados arquivos contendo diferentes tipos de erros e executados na ferramenta *Sparqlify* e no *SMLCompiler*. A Tabela 1 apresenta o comparativo entre as mensagens de erro produzidas pela *Sparqlify* e o pelo *SMLCompiler*, bem como uma descrição do erro contido no arquivo de configuração.

Tabela 1: Comparativo das mensagens de erro entre *Sparqlify* e *SMLCompiler*.

Descrição do erro	**Mensagem Sparqlify**	**Mensagem SMLCompiler**
A definição da variável deve terminar com '.', foi encontrado um ';'	2017-05-27 01:23:43,496 ERROR org.aksw.sparqlify.csv.CsvMapper\-CliMain: line 27:19 [templateConfig, templateConfigStmt, viewTemplateDefStmt, viewTemplateDef, constructTemplateQuads, quadPattern] mismatched input [@147,1290:1298='rdf:Class'<158>\-,27:19] expecting CLOSE_CURLY_BRACE 2017-05-27 01:23:43,497 ERROR org.aksw.sparqlify.csv.CsvMapper\-CliMain: line 28:9 [templateConfig] missing EOF at [@151,1316:1323='rdf:type', <158>\-,28:9]	Encountered " <VARIAVEL> "?City "" at line 27, column 3. Was expecting: <IDENT> ... 0 Lexical Errors found 1 Syntactic Errors found
A linguagem não aceita os símbolos '*' e '@' nos identificadores	2017-05-27 01:15:02,447 ERROR org.aksw.sparqlify. csv.CsvMapperCliMain: line 2:7 [templateConfig,	Line 2 - Invalid string found: *@ 1 Lexical Errors found 1 Syntactic

	templateConfigStmt, prefixDefStmt, prefixDecl] mismatched input [@4,113:113='f',-<130>,2:7] expecting PNAME NS	Errors found
Um parêntese não foi fechado	2017-05-27 01:28:13,075 ERROR org.aksw.sparqlify.csv.CsvMapperCli-Main: line 68:0 [templateConfig, templateConfigStmt, view-TemplateDefStmt, viewTemplateDef, varBinding-Part, varBinding, typeCtorExpression] missing CLOSE BRACE at [@353,2630:2634='City',<236->,68:0]	Encountered "" <VARIAVEL> """?City """" at line 68, column 1. Was expecting one of: "")"" ... "","" ... 0 Lexical Errors found 1 Syntactic Errors found

4. Considerações finais

O desenvolvimento deste trabalho visa prover ao engenheiro de dados uma forma mais amigável de apontar erros contidos em seus *scripts*, do que a ofertada pela ferramenta *Sparqlify*. Para tanto, foi desenvolvido um compilador denominado *SMLCompiler.*

Além de proporcionar mensagens mais diretas e legíveis, o compilador também é capaz de apontar a posição exata no código em que encontrou o erro, além de informar se o mesmo trata-se de um erro léxico ou sintático. Pondera-se, porém, que o protótipo desenvolvido não apresenta todas as funcionalidades observadas em um compilador moderno. Uma delas é a recuperação de erros sintáticos. Outra limitação presente é o fato do analisador semântico não tratar de erros de sobreposição de identificadores, isto é, é possível que um prefixo seja declarado duas vezes, por exemplo.

Pretende-se, em trabalhos futuros, além da superação das limitações já citadas, desenvolver uma interface gráfica para o compilador, tornando seu uso ainda mais amigável e estender sua funcionalidade a todas as seções de um arquivo SML.

Referências

[1] STADLER, C., UNBEHAUEN, J., WESTPHAL, P., SHERIF, M. A., and LEHMANN, J. Simplified rdb2rdf mapping. In Bizer, C., Auer, S., Berners-Lee, T., and Heath, T., editors, LDOW@WWW, volume 1409 of CEUR Workshop Proceedings, 2015.

[2] SPARQLIFY. Disponível em: http://sparqlify.org/wiki/Sparqlification_mapping_language. Acesso em: 30/08/2017.

[3] AHO, A. V., SETHI, R., and ULLMAN, J. D. **Compilers: Principles, Techniques and Tools**. Addison-Wesley, 1986.

[4] JAVACC. Diponível em: https://javacc.org/. Acesso em 30/08/2017.

Uma Hiper-Heurística para o *Flowshop* Multiobjetivo

Geovani F. Antunes, Carolina Paula de Almeida, Richard Gonçalves, Sandra Mara G. S. Venske
geovanyantunes@hotmail.com, carol@unicentro.br, richard@unicentro.br, ssvenske@unicentro.br

Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *Inúmeros problemas de otimização apresentam mais de uma função-objetivo a ser minimizada ou maximizada. Tais problemas pertencem à classe de Problemas de Otimização Multiobjetivo (POMs). O Flowshop é um problema de sequenciamento presente na indústria, com objetivo de alocar da melhor maneira as atividades em cada máquina. Hiper-Heurísticas são uma metodologia de alto nível para seleção ou geração automática de heurísticas para solucionar problemas complexos. Este trabalho propõe uma Hiper-Heurística de seleção de operadores no MOEA/D para o Flowshop multiobjetivo. Os resultados obtidos, considerando instâncias do benchmark de Taillard, são promissores e vários apontamentos para trabalhos futuros foram identificados.*

Palavras-chave: Otimização Combinatória; *Multi-Armed Bandit;* MOEA/D.

1. Introdução

Hiper-Heurísticas são técnicas propostas como suficientemente genéricas para serem aplicadas em domínios diferentes. Realizando a seleção automática ou gerando heurísticas para resolver problemas computacionais complexos, tomando decisões em tempo de execução e de acordo com o momento do processo de otimização. O *Flowshop* é um problema de alocação de atividades na cadeia de produção de indústrias e tratar a versão multiobjetivo deste é importante, pois, em sua grande maioria, os problemas dessa natureza são multiobjetivos. Assim, este trabalho propõe uma Hiper-Heurística para resolvê-lo.

2. Materiais e métodos

Grande parte dos problemas do mundo real possuem objetivos conflitantes, como no processo de produção na indústria, no qual deve-se minimizar o tempo total das atividades, maximizando a eficiência da produção e mantendo a qualidade. Esses problemas, contendo mais de uma função objetivo a serem otimizadas, são denominados Problemas de Otimização Multiobjetivo (POMs) [6]. A solução para esses problemas não é única e, geralmente, é encontrada em um conjunto de soluções-compromisso entre os diferentes objetivos, chamado de Fronteira de Pareto [2].

Nem sempre é possível resolver POMs de maneira exata. Nesse contexto, é

interessante o uso de Algoritmos Evolucionários Multiobjetivo (AEMOs), os quais são métodos aproximativos para solucionar tais problemas. No MOEA/D (*Multi-Objective Evolutionary Algorithm based on Decomposition*) o problema é decomposto em problemas de otimização mono-objetivo, e cada um deles é resolvido de maneira colaborativa [5].

O *Flowshop* é um problema de alocação de atividades na cadeia de produção de indústrias, no qual se deseja programar *n* atividades a serem executadas por *m* máquinas diferentes [4]. Os objetivos são minimizar o tempo de execução de todas as tarefas (*makespan*) e o tempo total de uma tarefa (*flowtime*), o que diminui o custo de produção. O algoritmo proposto (MOEA/D-DRA-MAB) foi testado em 12 instâncias do *benchmark* de Taillard, que consideram diferentes quantidades de máquinas e atividades.

Hiper-Heurísticas (HHs) são algoritmos de alto nível capazes de resolver diferentes instâncias de diferentes problemas, sem modificações ou com modificações mínimas na implementação do algoritmo, sendo que as modificações necessárias geralmente se limitam às modificações para atender os novos domínios. As HHs podem ser classificadas como sendo de seleção ou geração. As HHs de seleção visam escolher a melhor heurística a ser aplicada em cada instante do processo de otimização, enquanto as HHs de geração visam criar melhores heurísticas para resolverem o problema em questão [1].

O problema do *Multi-Armed Bandit* (MAB) consiste em uma máquina com um determinado número de alavancas em que o apostador tem que escolher uma para puxar, de modo a maximizar seus ganhos em uma série de tentativas. Inicialmente não se sabe nada sobre as alavancas e através de várias tentativas é que pode-se determinar quais alavancas dão melhor retorno [3]. Vários problemas de otimização e aprendizado do mundo real podem ser modelados com essa abordagem, em que se deseja maximizar ganhos ou minimizar perdas.

3. Resultados e discussão

Uma HH de seleção baseada em MAB foi implementada no algoritmo MOEA/D-DRA-MAB para resolver o *Flowshop* multiobjetivo. A Tabela 1 apresenta os resultados obtidos para o indicador ε-Unário, comparando a seleção automática com os operadores individuais, o *Cycle Crossover* (CX), *Order Crossover* (OX) e o *Partially-Mapped Crossover* (PMX). Os valores em cinza escuro significam os melhores resultados e os valores em cinza claro correspondem ao segundo melhor resultado. O algoritmo proposto obteve os melhores resultados para cinco instâncias e o segundo melhor para três instâncias. O operador PMX obteve o melhor resultado para três instâncias, o OX e o CX obtiveram o melhor resultado para duas instâncias.

Tabela 1. ε-Unário – média e desvio padrão – efeito da Hiper-Heurística.

Instância	MOEA/D-DRA-MAB		MOEA/D-DRA-PMX		MOEA/D-DRA-OX		MOEA/D-DRA-CX	
	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
tai20_5	1,66e+02	1,2e+02	1,68e+02	1,0e+02	1,99e+02	1,2e+02	2,14e+02	9,9e+01
tai20_10	3,76e+02	2,8e+02	3,20e+02	1,8e+02	3,11e+02	2,1e+02	4,96e+02	2,2e+02
tai20_20	3,80e+02	1,8e+02	3,37e+02	1,9e+02	3,94e+02	2,1e+02	4,43e+02	1,5e+02
tai50_5	1,13e+03	5,8e+02	1,04e+03	5,7e+02	9,64e+02	4,4e+02	9,08e+02	4,2e+02
tai50_10	2,00e+03	7,7e+02	2,05e+03	9,1e+02	2,09e+03	8,0e+02	2,06e+03	7,4e+02
tai50_20	1,75e+03	9,6e+02	2,73e+03	1,1e+03	2,14e+03	9,5e+02	2,35e+03	1,1e+03
tai100_5	3,07e+03	1,1e+03	3,13e+03	1,3e+03	3,01e+03	1,3e+03	3,25e+03	1,4e+03
tai100_10	4,91e+03	2,3e+03	3,82e+03	2,1e+03	5,45e+03	2,0e+03	4,28e+03	1,6e+03
tai100_20	5,91e+03	2,7e+03	6,54e+03	2,6e+03	7,08e+03	2,3e+03	6,17e+03	2,6e+03
tai200_10	1,02e+04	2,4e+03	1,07e+04	4,4e+03	1,38e+04	5,3e+03	1,35e+04	4,9e+03
tai200_20	1,50e+04	6,0e+03	1,42e+04	7,4e+03	1,60e+04	6,9e+03	1,34e+04	6,3e+03
tai500_20	4,25e+04	1,8e+04	4,00e+04	2,0e+04	8,32e+04	2,6e+04	7,36e+04	2,7e+0

4. Considerações finais

O algoritmo foi testado para 12 instâncias e nas simulações em que os operadores individuais foram confrontados com a escolha automática feita pela HH. O MOEA/D-DRA-MAB obteve desempenho satisfatório com o melhor ou o segundo melhor resultado, e a vantagem de não exigir que o usuário escolha o operador a ser aplicado ao longo do processo de busca.

Os resultados são promissores e foram identificados vários pontos a serem explorados, sendo que a literatura mostra diferentes estratégias e componentes que podem melhorar o desempenho de AEMOs [6], tais como: o uso de buscas locais, o uso de um arquivo externo para armazenar soluções não dominadas e diferentes funções de agregação.

Referências

[1] Burke, E. K. et al. Hyper-heuristics: A survey of the state of the art. *Journal of the Operational, Research Society*, v. 64, n. 12, p. 1695–1724, 2013.

[2] Coello, C. A. C. et al. **Evolutionary algorithms for solving multi-objective problems**, Springer, 2007.

[3] Robbins, H. Some aspects of the sequential design of experiments. *In Herbert Robbins Selected Papers*, v. 55, p. 169–177, 1985.

[4] Yenisey, M. M. e Yagmahan, B. Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. *Omega*, v. 45, p. 119–135, 2014.

[5] Zhang, Q. e Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, v.11, n.6, p. 712–731, 2007.

[6] Zhou, A. et al. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, v. 1, n. 1, p. 32–49, 2011.

Uma proposta de solução para o Problema da Árvore Geradora Mínima com incertezas em grafos completos

Cassiano Blonski Sampaio[1], Fábio Hernandes[1]

cassianobsampaio@gmail.com, hernandes@unicentro.br

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *Considerando a importância do problema da árvore geradora mínima com incertezas na Engenharia e na Informática, neste trabalho é proposto um algoritmo genético para esse tema, sendo as incertezas nos pesos das arestas e na estrutura em si, além da rede ser um grafo completo. As incertezas são modeladas por meio da Teoria dos Conjuntos Fuzzy.*

Palavras-chave: Algoritmos Genéticos; Teoria dos Conjuntos *Fuzzy*; Número de Prüffer.

1. Introdução

O Problema da Árvore Geradora Mínima, PAGM, consiste em, a partir de um grafo não orientado, conexo e com pesos em cada aresta, gerar uma árvore que conecte todos os vértices do grafo e que tenha o menor peso total. Esse problema possui aplicações nas mais diversas áreas da Engenharia e da Informática [1], porém, em algumas aplicações, os parâmetros da rede ou a estrutura em si não são precisos, pois em certos casos estes representam tempo, custos e conexões [2], parâmetros que dificilmente são tratados de forma exata. Com isso, surgiu o PAGM com incertezas, proposto por Delgado et al. [3] e abordado por meio da Teoria dos Conjuntos *Fuzzy* [4], passando a ser denominado de Problema da Árvore Geradora Mínima Fuzzy, PAGM-*Fuzzy*.

Analisando os trabalhos da literatura que abordam o PAGM-*Fuzzy*, verifica-se que esses, em sua maioria, resolvem o problema com incertezas nos parâmetros ou na estrutura da rede, exceto o trabalho de Hernandes e Sampaio [5], que propõe um algoritmo exato para o problema, porém, caro para redes de porte médio. Logo, visando continuar a abordagem com incertezas nos parâmetros e na estrutura da rede, simultaneamente, neste trabalho é proposto um Algoritmo Genético, AG, para o PAGM com incertezas nos parâmetros e na estrutura da

rede. Com a implementação desse algoritmo, buscou-se encontrar um conjunto solução que se aproximasse ao máximo do ótimo, visando reduzir a questão da complexidade presente no método exato, podendo, assim, ser aplicado em redes de qualquer tamanho.

2. Algoritmo proposto

Considerando que o objetivo deste trabalho é abordar o PAGM-*Fuzzy*, os pesos das arestas foram abordados como números *fuzzy* triangulares, enquanto que cada aresta tem uma pertinência *fuzzy*, $\mu \epsilon [0,1]$, de pertencer à rede. Para eliminar as árvores de pesos maiores, mais caras, foi utilizado o conceito da distribuição de credibilidade inversa de Zhou et al. [6].

Ressalta-se que, na codificação do cromossomo, foi utilizado o conceito do Número de Prüfer [6], a fim de contornar o problema de encontrar a solução, o que ocorre com os métodos exatos.

O algoritmo é composto por seis passos, sendo: (i) crie a população inicial aleatória *P*, com X indivíduos, sendo possível apenas valores de 0 até a quantidade de vértices para cada posição do cromossomo, e calcule o *fitness* de cada indivíduo; (ii) selecione uma percentagem dos melhores indivíduos de *P* para o elitismo e aloque os indivíduos na População elite (*Pe*); (iii) selecione uma percentagem dos indivíduos de *P* para cruzamento, proceda a operação e aloque os novos indivíduos na população intermediária (*Pi*) até que *Pi* seja igual *P-Pe*; (iv) selecione uma percentagem dos indivíduos de *Pi* para a mutação, proceda a operação; (v) faça $P \leftarrow Pi + Pe$ e calcule o valor do *fitness* (soma dos pesos das arestas do indivíduo) de cada indivíduo de *P*; (vi) se (i = *Geração*) → Fim. Senão retorne ao Passo (i).

3. Resultados e discussão

O algoritmo proposto foi implementado em Python e executado para o grafo da Figura 1, sendo que na Tabela 1 estão os pesos e as pertinências das arestas.

Durante as execuções foram utilizadas várias combinações de geração {5, 15, 20, 50}, população inicial {10, 30, 50, 100}, taxa de mutação {0.01, 0.1}, taxa de seleção {0.1, 0.01} e credibilidade {0.2, 0.5, 0.8}, sendo executadas 10 vezes cada combinação.

Após a execução, verificou-se que o melhor indivíduo teve solução 239.8, com os seguintes parâmetros: credibilidade 0.2, geração 20, população inicial 50, taxa de mutação 0.01 e taxa de seleção 0.01.

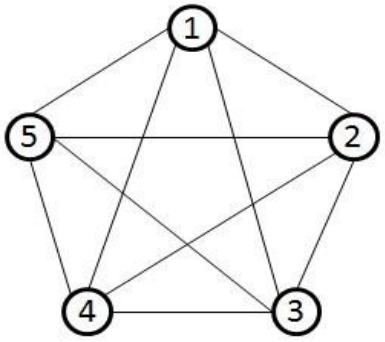

Figura 1. Rede de Zhou et al. [6]

Origem	Destino	Peso	Pertinência
1	2	(60, 65, 70)	0,3125
2	3	(58, 70, 75)	0,6823
3	4	(70, 74, 82)	0,7684
4	5	(62, 72, 80)	0,3357
1	5	(58, 66, 76)	0,2528
1	4	(64, 70, 74)	0,8132
1	3	(58, 62, 70)	0,2987
2	5	(65, 70, 80)	0,5255
2	4	(55, 60, 65)	0,6349
3	5	(62, 68, 72)	0,4966

Tabela 1. Dados da rede da Figura 1

4. Considerações finais

O PAGM é abordado em diversas aplicações da Engenharia e da Informática, sendo que em algumas destas as redes possuem incertezas nos pesos e até mesmo em suas estruturas. Logo, o objetivo deste trabalho foi estudar o PAGM-*Fuzzy*, propondo um algoritmo genético que abordasse as incertezas nos pesos e na estrutura da rede, simultaneamente. O diferencial do algoritmo proposto foi a utilização do Número de Prüfer, o que fez com que este convergisse rapidamente, pois os indivíduos da população sempre eram factíveis.

Referências

[1] Ahuja, R.K., Magnanti T.L. e Orlin J.B. **Network Flows: Theory, Algorithms and Applications**, Prentice Hall, 1993.

[2] Takahashi M.T. e Yamakami A. Um estudo sobre o problema da árvore geradora mínima com estrutura do grafo fuzzy, *Anais do XXXVI Simpósio Brasileiro de Pesquisa Operacional*, vol. 1, p. 2278--2285, 2004.

[3] Delgado M., Verdegay J.L. e Vila M.A. On fuzzy tree definition, *European Journal of Operational Research*, vol. 22, p. 243--249, 1985.

[4] Zadeh L. Fuzzy sets, *Information and Control*, vol. 8, p. 338--353, 1965.

[5] F. Hernandes F. e Sampaio C.B. O problema da árvore geradora mínima fuzzy: um algoritmo para o caso envolvendo incertezas nos pesos das arestas e na estrutura da rede, *Anais do IV Congresso Brasileiro de Sistemas Fuzzy*, vol. 1, p. 69--80, 2016

[6] Zhou J., Chen L., Wang K. e Yang F. Fuzzy α-minimum spanning tree problem: definition and solutions, *International Journal of General Systems*, vol. 45, p. 311--335, 2016.

Uso do Vocabulário Data Cube em Dados Abertos Conectados

Gianluca Bine[1], Josiane Hauagge Dall' Agnol[1]
gian_bine@hotmail.com, jhauagge@unicentro.br

[1]Universidade Estadual do Centro-Oeste – UNICENTRO

Resumo: *O Consórcio W3C recomenda a utilização do RDF Data Cube Vocabulary para a publicação de dados estatísticos na Internet. O uso das características multidimensionais do RDF Data Cube facilita a apresentação visual desses dados. Esta pesquisa tem como objetivo principal usar o Vocabulário RDF Data Cube para a semantificação de um conjunto de Dados Governamentais Abertos. Como resultado, produziu-se uma base de dados modelada com o RDF Data Cube Vocabulary, sendo possível concluir que o uso de um vocabulário padrão em Dados Abertos Conectados padroniza e facilita a utilização desses dados.*

Palavras-chave: RDF Data Cube; Dados Abertos Conectados; Dados Governamentais Abertos.

1. Introdução

Os dados do Governo Brasileiro estão disponíveis no Portal Brasileiro de Dados Abertos [4]. Nesse portal podem ser encontrados conjuntos de dados sobre diferentes domínios, tais como: lixo, saúde e finanças. Esses dados, assim como a maioria dos Dados Abertos disponíveis na *web*, ainda precisam ser tratados para serem considerados como Dados Abertos Conectados (*Linked Open Data* – LOD). Vocabulários são usados em Dados Abertos Conectados para definir conceitos e promover a interoperabilidade semântica [5].

Nesta pesquisa foi utilizado o Vocabulário RDF *Data Cube* [3] para a semantificação de um conjunto de Dados Governamentais Abertos, o qual fornece um modelo para publicação de dados multidimensionais, como os dados estatísticos, no formato de Dados Abertos Conectados [3]. Dados multidimensionais são informações que possuem relações com vários atributos, como, por exemplo, a previsão meteorológica de uma região, na qual cada dado sobre a quantidade de chuva está relacionado a uma cidade e à data da medição.

2. Materiais e métodos

A metodologia utilizada para o desenvolvimento desta pesquisa foi a *Linked Data Lifecycle*, por meio da abordagem LOD2 *Stack* [6]. O ciclo de vida da metodologia *Linked Data Lifecycle* é iniciado antes mesmo da estruturação da base de dados no formato RDF. Segundo Auer [2], suas oito atividades são: Extração, Armazenamento/Consulta, Revisão Manual/Autoria, Interligação/Fusão, Classificação/Enriquecimento, Análise de Qualidade, Evolução/Reparação e Busca/Navegação/Exploração.

O Vocabulário RDF *Data Cube* é focado exclusivamente na publicação de dados multidimensionais na *web* e divide o conjunto de dados em quatro partes: observação, estrutura organizacional, metadados estruturais e metadados de referência [3].

A base de dados utilizada no desenvolvimento desta pesquisa foi referente ao Índice de Desenvolvimento Humano Municipal do Brasil (IDHM) dos anos 1991, 2000 e 2010 [1]. Foram usados os campos: Lugar, COD IBGE e IDHM. A base de dados foi separada em diferentes arquivos CSV (*Comma-Separated Values*), de acordo com o ano de medição do IDHM, para facilitar a modelagem do grafo RDF na ferramenta OpenRefine.

3. Resultados e discussão

O processo de modelagem dos grafos RDF da base de dados foi feito de forma manual e usando a ferramenta OpenRefine, seguindo o tutorial do autor [7]. A URI (*Uniform Resource Identifier*) base para o grafo foi: **http://lod.unicentro.br/**. A URI base teve o prefixo **dc/** acrescentado em seu final para identificar os grafos que usam o Vocabulário RDF *Data Cube*. A modelagem das URI's da base de dados IDHM obtida foi:

- *Data Cube* [dc]: **http://lod.unicentro.br/dc/idhm/**
- *DataSet* [ds]: **http://lod.unicentro.br/dc/idhm/dataset/**
- *Properties* [prop]: **http://lod.unicentro.br/dc/idhm/prop/**
- *Data Cube Component Specifications* [dccs]: **http://lod.unicentro.br/dc/idhm/dccs/**

Nas Tabelas 1 e 2 pode-se observar medidas e dimensões identificadas na base IDHM.

Tabela 1. Medidas identificadas na base de dados IDHM

Medidas		
Coluna	Descrição	Valores
Resultado	O resultado obtido (IDHM)	Vários, do arquivo original

Tabela 2. Dimensões identificadas na base de dados IDHM

Dimensões		
Coluna	Descrição	Valores
Código	Código IBGE do município da observação	Vários, do arquivo original
Município	Nome do município da observação	Vários, do arquivo original
UF	Unidade federativa da observação	(SP, RJ, etc)
Ano	Ano da observação	1991, 2000 e 2010

4. Considerações finais

Esta pesquisa teve como objetivo principal usar o Vocabulário RDF *Data Cube* para a semantificação de um conjunto de Dados Governamentais Abertos e teve como resultado um grafo RDF da base de dados IDHM. A contribuição principal desta pesquisa foi mostrar que é possível semantificar um conjunto de Dados Governamentais Abertos por meio de um vocabulário padronizado. Assim, os dados estão disponíveis em um *endpoint,* podendo ser consultados e apresentando informações que ajudam na tomada de decisões.

Referências

[1] ATLASBRASIL. Consulta – Atlas do Desenvolvimento do Brasil. Disponível em: http://www.atlasbrasil.org.br/2013/pt/consulta/ - Acesso em: 16/08/2017.

[2] AUER, Sören et al. Managing the life-cycle of linked data with the LOD2 stack. In: **International semantic Web conference**. Springer, Berlin, Heidelberg, 2012. p. 1-16.

[3] CYGANIAK, Richard; REYNOLDS, Dave; TENNISON, Jeni. The RDF data cube vocabulary. **W3C Recommendation (January 2014)**, 2013.

[4] GOVERNO FEDERAL DO BRASIL. Sobre o dados.gov.br. Disponível em: http://dados.gov.br/paginas/sobre – Acesso em: 16/08/2017.

[5] LINKED OPEN VOCABULARIES. About Linked Open Vocabularies. Disponível em: https://lov.okfn.org/dataset/lov/about – Acesso em: 16/08/2017.

[6] LOD2. Welcome - LOD2 - Creating Knowledge out of Interlinked Data. Disponível em: http://lod2.eu/Welcome.html – Acesso em: 16/08/2017.

[7] WILLIAMS, Tim. OpenRefine RDF Cube Creation Workbook. Disponível em: http://phusewiki.org/wiki/index.php?title=File:TT03_CompanionDoc-OpenRefine-CubeWorkbook.docx – Acesso em: 16/08/2017.