Anais do
Terceiro Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial

Campinas, São Paulo
18 e 19 de março de 2010

Editora: Wu Shin-Ting

Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas

Prefácio

Esta publicação reúne as 21 contribuições ao Terceiro Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial da Faculdade de Engenharia Elétrica e de Computação da Unicamp (III EADCA), realizado nos dias 18 e 19 de março de 2010. As contribuições, que cobrem temas de computação gráfica, redes de computadores, sistemas distribuídos, sistemas embarcados, engenharia de software, processamento de imagens, sistemas inteligentes, automação e processamento de sinais, revelam claramente a abrangência da pesquisa realizada no departamento bem como a riqueza das duas grandes áreas que o caracterizam.

Gostaríamos de agradecer, em primeiro lugar, o apoio de todos os membros do DCA, que são, sem dúvida, os principais responsáveis pela existência do evento. Graças a esse apoio, o encontro, em sua terceira edição, torna-se uma iniciativa plenamente consolidada no âmbito de nossa faculdade. Também desejamos manifestar nossa gratidão aos palestrantes deste ano, Profs. Léo Pini Magalhães e Márcio Luiz de Andrade Netto, à Diretoria da FEEC, que tem apoiado esta iniciativa desde a sua concepção, ao Prof. José Raimundo de Oliveira e a seu orientando Rodrigo C. V. Dias, que mais uma vez nos auxiliaram no processo de submissão eletrônica, ao Prof. José Antenor Pomílio que gentilmente nos cedeu o *software Acrobat Distiller* para edição desta publicação, aos *chairmen* de todas as sessões do evento, e a todos os que, de alguma forma, colaboraram com a organização.

Por fim, não poderíamos deixar de externar nossos agradecimentos aos autores e orientadores, que cuidadosamente prepararam os manuscritos que compõem esta publicação, e a todos que vieram prestigiar essa terceira edição do evento. Esperamos que este evento tenha aberto diversas possibilidades de interação científica e que tenha cumprido o papel de divulgar os trabalhos elaborados por pesquisadores vinculados a nosso departamento.

Desejamos que esta coletânea como os dois dias do evento sejam produtivos e enriquecedores!

Romis Ribeiro De Faissol Attux
Wu Shin-Ting
Comissão Organizadora

Diretoria da Faculdade de Engenharia Elétrica e de Computação

Prof. Max Costa - Diretor

Prof. José Raimundo de Oliveira – Diretor Associado

Conselho Departamental do Departamento de Engenharia de Computação e Automação Industrial

Alice Maria Bastos Hubinger. Tokarnia

Clésio Luiz Tozzi - Vice-Chefe

Daniel Camilo

Eleri Cardozo

Fernando Antonio Campos Gomide

Fernando José Von Zuben

Ivan Luiz Marques Ricarte

José Mario De Martino

José Raimundo de Oliveira

Léo Pini Magalhães

Marco Aurélio Amaral Henriques

Mario Jino

Maurício Ferreira Magalhães

Rafael Santos Mendes

Ricardo Ribeiro Gudwin - Chefe

Roberto de Alencar Lotufo

Romis Ribeiro De Faissol Attux

Wagner Caradori do Amaral

Wu Shin –Ting

EADCA 2010

Palestras Convidadas

Simulação de Multidões - Atividades no DCA

Prof. Léo Pini Magalhães

Na palestra será abordado um método para simulação de multidões baseado no algoritmo de colonização do espaço. Este algoritmo foi originalmente proposto para modelar padrões de nervuras em folhas vegetais e de ramificações em árvores. A técnica baseia-se na competição por espaço entre nervuras ou ramificações durante o crescimento vegetal. Adaptado à simulação de multidões, o algoritmo de colonização do espaço visa simular a competição por espaço durante o movimento dos pedestres. Este trabalho foi tema da tese de doutorado de A. L. Bicho, concluída em julho de 2009 com a participação na orientação da Profa. Dra. Soraia R. Musse e é tema atual de um trabalho de Iniciação Científica de Igor C. Pinheiro / E.Elétrica. Serão apresentados vídeos para mostrar resultados obtidos.

Inovação, Educação e Sociedade

Prof. Márcio Luiz de Andrade Netto

O tema Inovação é complexo e alguns de seus componentes serão tratados na apresentação, buscando traçar um panorama abrangente e visando a compreensão dos problemas e de meios para que a sociedade seja beneficiada.

A Educação, em todos os seus níveis, configura-se em ingrediente essencial para os processos inovadores e deve ser tratada com a importância devida.

A Sociedade é, em última análise, a responsável pelos caminhos que o País trilhará, imprimindo aos governos as características que serão incentivadoras ou inibidoras das atividades inovadoras.

Empreendedorismo, competitividade, regulações econômicas e financeiras, incentivos fiscais e de outras naturezas são alguns dos fatores que incidem sobre a Inovação e que devem ser abordados e entendidos.

Computação Gráfica

Modelo de transcrição da Língua de Sinais Brasileira voltado a implementação de agentes virtuais sinalizadores

Wanessa Machado do Amaral (Orientador: José Mario De Martino)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

{wmamaral, martino}@dca.fee.unicamp.br

Abstract – Accessibility is a growing concern in computer science. Since virtual information is mostly presented visually, it may seem that access for deaf people is not an issue. However, for prelingually deaf individuals, those who were deaf since before acquiring and formally learn a language, written information is often of limited accessibility than if presented in signing. Further, for this community, signing is their language of choice, and reading text in a spoken language is akin to using a foreign language. Sign language uses gestures and facial expressions and is widely used by deaf communities. To enabling efficient production of signed content on virtual environment, it is necessary to make written records of signs. Transcription systems have been developed to describe sign languages in written form, but these systems have limitations. Since they were not originally designed with computer animation in mind, in general, the recognition and reproduction of signs in these systems is an easy task only to those who deeply know the system. The aim of this work is to develop a transcription system to provide signed content in virtual environment. To animate a virtual avatar, a transcription system requires explicit enough information, such as movement speed, signs concatenation, sequence of each hold-and-movement and facial expressions, trying to articulate close to reality. Although many important studies in sign languages have been published, the transcription problem remains a challenge. Thus, a notation to describe, store and play signed content in virtual environments offers a multidisciplinary study and research tool, which may help linguistic studies to understand the sign languages structure and grammar.

Keywords – computer graphics, sign language, XML, accessibility, virtual reality.

1. Introdução

De acordo com o IBGE[1] o Brasil possui atualmente 5,7 milhões de brasileiros com algum grau de deficiência auditiva. Kennaway[2] demonstra que a performance de leitura de crianças surdas geralmente é inferior quando comparada à performance de leitura de crianças com audição normal. A acessibilidade de deficientes auditivos em ambientes virtuais pode ser melhorada provendo conteúdo em língua de sinais. Conteúdo em língua de sinais vem sendo reproduzido nos computadores em forma de arquivos de vídeo. Essa opção é bastante custosa, uma vez que se faz necessário o uso de infraestrutura física específica, bem como a participação de pessoas treinadas que conheçam em detalhes a língua de sinais. Para a criação de um vídeo consistente é necessário haver continuidade, utilizando a mesma pessoa para reproduzir os sinais, com as mesmas roupas e o mesmo fundo. Dessa forma, criar pequenas partes de vídeo e depois agrupá-las para formar um único material não é tarefa trivial. A cada detalhe alterado no conteúdo, novo vídeo precisa ser produzido, tornando difícil a manutenção do material e aumentando os custos. A transmissão e o armazenamento de vídeos é outra dificuldade, uma vez que geralmente são arquivos grandes. Na internet, por exemplo, é necessária uma conexão rápida e estável para a transmissão e recepção de vídeos.

A animação de humanos virtuais mostra-se, portanto, como uma alternativa conveniente. Entre as vantagens, destaca-se que a criação de conteúdo em língua de sinais poderá ser realizada por uma única pessoa utilizando um computador, sem a necessidade de equipamentos especiais para captura e processamento de vídeos. O conteúdo também pode ser criado mais facilmente, por pessoas não necessariamente treinadas e com fluência em língua de sinais. Um agente virtual possibilita a geração de conteúdo em tempo real. Dessa forma, a continuidade também deixa de ser um problema, uma vez que o conteúdo poderá ser alterado a qualquer momento, sem a necessidade de regravar a sequência de sinalização inteira. O armazenamento do conteúdo é outra vantagem. O espaço em disco no computador requerido para armazenar a descrição dos sinais é bastante inferior se comparado ao armazenamento de arquivos de vídeo. A transmissão do conteúdo também é facilitada, uma vez que o conteúdo transcrito pode ser armazenado em arquivos de texto, que são menores e mais fáceis de serem transmitidos que arquivos de vídeo. Existe ainda a possibilidade de oferecer ao usuário controle adicional sobre o material transmitido, como

alteração do ponto de vista durante a reprodução para que o sinal seja melhor visualizado, o que é impossível na reprodução por vídeo.

Para implementar um sinalizador virtual é necessário utilizar um sistema de transcrição da LIBRAS que registre todos os detalhes relevantes com o objetivo de reproduzir a naturalidade e espontaneidade presentes no trabalho do intérprete real, na tentativa de garantir o entendimento do sinal reproduzido. Entretanto, é importante salientar que o objetivo deste trabalho não é substituir o intérprete. As habilidades humanas são indispensáveis para a atividade de tradução, que não é o foco deste trabalho.

As soluções apresentadas na literatura até o momento para a animação de agentes virtuais sinalizadores possuem algumas limitações. Os sistemas de transcrição tradicionais não foram desenvolvidos com o intuito de gerar animações. Muitas informações importantes para a reprodução do sinal não aparecem nas notações existentes. Algumas informações implícitas podem facilmente ser deduzidas por intérpretes reais, mas o mesmo não acontece com o uso de um intérprete virtual. Surge então a necessidade da criação de um sistema de transcrição robusto o suficiente, contendo o maior número de informações relevantes, para garantir a animação realista de agentes virtuais.

2. Proposta

A notação proposta neste trabalho primeiramente separa as informações em dois grupos, os elementos *suspensao* e *movimento*. O elemento *sinal* é a raiz da descrição. Um sinal pode ter uma ou mais suspensões, unidas ou não por movimentos globais A Figura 1 ilustra essa hierarquia.

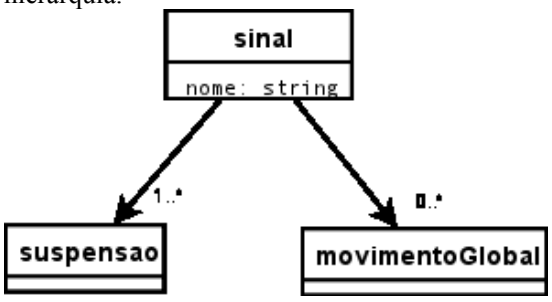


Figura 1. Hierarquia simplificada da descrição de um sinal.

A sequência da realização de cada componente do sinal é descrita na notação através do atributo *numero*. O elemento *movimentoGlobal* também possui atributo de mesmo nome. Deve-se observar no entanto que não é sempre que uma suspensão será sucedida por um movimento. O valor do atributo *numero* em *movimentoGlobal* está portanto diretamente relacionado ao número da suspensão. Foram criados elementos separados para aos mãos. Tanto a mão direita como a esquerda contém elementos para descrever configuração, localização, orientação da palma e movimento local. A mão esquerda contém o atributo *espelhada*, que permite ser atribuído valor igual a "*sim*" quando sua configuração de mão for igual da mão direita.

A configuração da mão é a maneira como estão dispostos os dedos, unidos ou separados, e a situação das juntas, se flexionadas ou distendidas, por exemplo. Existem configurações de mão mais utilizadas na LIBRAS, de maneira que é possível estabelecer um conjunto finito de opções. Em geral, as configurações mais utilizadas são as letras do alfabeto e os números (Figura 2).

Figura 2. Alfabeto e números da LIBRAS.[3]

O espaço de sinalização é representado como um região de três dimensões. Alternativamente, o espaço de sinalização pode ser representado como um ponto de contato, que pode ser com outra a mão, com partes do corpo ou rosto.

A orientação da palma da mão pode estar na horizontal ou vertical. Na horizontal, a palma da mão pode estar voltada para: cima, palma visível, o lado, dorso voltado para direita, ou para baixo, dorso visível. Na vertical, a palma da mão pode estar voltada para: o sinalizador, palma visível, dorso para frente o interlocutor, dorso visível, palma para frente ou para o lado, dorso voltado para direita.

Um sinal pode conter zero ou mais movimentos. Os movimentos são divididos em dois grandes grupos: locais e globais. Os movimentos locais são aqueles em que apenas a movimentação das articulações dos dedos,

rotação do pulso ou do antebraço são realizadas, onde a localização das mãos no espaço não se altera. Os movimentos locais foram divididos em três categorias: antebraço, pulso e dedos.

Os movimentos possíveis para o pulso e o antebraço são: *baixo*: da posição de repouso o pulso (ou antebraço) realiza rotação para baixo; *cima*, movimento oposto ao anterior; *baixocima*, rotação que parte da posição de repouso para baixo, volta e depois sobe; *cimabaixo*, movimento oposto ao anterior.

O movimento dos dedos podem ser os seguintes: articulações proximais abrem; articulações proximais fecham; articulações proximais abrem e fecham (juntas); articulações proximais fecham e abrem (juntas); articulações proximais abrem e fecham alternadas; articulações proximais fecham e abrem alternadas; articulações distais abrem; articulações distais fecham; articulações distais abrem e fecham; articulações distais fecham e abrem; esfregar; circular horário; circular anti-horário.

. O elemento dedos pode ter os seguintes elementos vazios como filhos: *polegar*, *indicador*, *dedoMedio*, *anelar* e *dedoMinimo*, que quando preenchidos indicam quais dedos realização o movimento.

As expressões faciais foram divididas em nove componentes: testa: franzida; sobrancelhas: para cima, retas, para baixo, para cima lado de dentro, para baixo lado de dentro; olhos: abertos, espremidos, fechados, meio abertos, bem abertos; olhar: é a direção do olhar, e pode ser para cima, para cima e um dos lados, para os lados, para baixo, para baixo e um dos lados; bochechas: estufadas, sugadas, tensas, soprar; nariz: franzido; boca: fechada, sorriso fechado, sorriso aberto, bocejo, beijo, tensa, dobras ao redor da boca; língua: visível dentro da boca; dentes: superiores tocando lábio inferior, inferiores tocando lábio superior.

O atributo *preDefinida* foi criado para facilitar descrições de expressões "prontas", como feliz ou triste. Este atributo pode ser utilizado quando não é desejada uma precisão muito grande na descrição da face, bastando dizer que a expressão é de alegria ou tristeza para uma boa articulação.

Os movimentos globais são as trajetórias entre uma suspensão e outra, dentro de um mesmo sinal. Este movimento também pode ser automático e inconsciente, como por exemplo uma acomodação para a posição inicial, o que ocorre ao soletrar uma palavra. Neste caso, a trajetória não precisa ser descrita, uma vez que a reprodução computacional do sinal deverá resolver o problema. No entanto, para movimentos intencionais, onde a maneira como a trajetória entre as suspensões acontece é parte da sinalização e faz-se necessária para o entendimento do sinal, a descrição do movimento deve ser realizada.

São classificados em circular (horário ou anti-horário), meio circulo (horário ou anti-horário), reto (para direita, esquerda, frente ou trás) ou em zigue-zague (começando da direita e para frente, da direita e para trás, da esquerda e para frente ou da esquerda e para trás).

O atributo *maos* descreve a dinâmica do movimento, ou seja, como o movimento é realizado, com uma ou duas mãos, e de que maneira, alternado, consecutivo, simultâneo ou espelhado. São valores possíveis para o atributo *maos*: *direita*: só a mão direita se move; *esquerda*: só a mão esquerda se move; *simultâneo*: ambas as mãos se movem, juntas; *alternado*: mão direita move na direção contrária à mão esquerda, e vice versa; *consecutivoD*: uma das mãos move enquanto outra fica parada. Depois inverte. Movimento começa com a mão direita; *consecutivoE*: uma das mãos move enquanto outra fica parada. Depois inverte. Movimento começa com a mão esquerda; *espelhado*: as duas mãos se movem, de forma espelhada; *espelhadoConsecutivoD*: as duas mãos se movem, em movimentos espelhados, uma de cada vez. Mão direita move primeiro; *espelhadoConsecutivoE*: as duas mãos se movem, em movimentos espelhados, uma de cada vez. Mão esquerda move primeiro.

A velocidade de execução do movimento global pode ser rápida, lenta ou padrão. Quando não preenchido, o atributo é considerado com valor "*padrao"*. O movimento pode ter também sua velocidade acelerada ou desacelerada durante a articulação.

O atributo *repetir*, assim como no movimento local, serve para descrever quantas vezes o movimento é repetido. Se igual a 0, o movimento ocorre e a mão não volta ao seu local de repouso. Se o valor de repetir for 2, quer dizer que a mão vai, volta e vai novamente, assim por diante.

Pode acontecer o contato com a mão, os dedos, parte do corpo ou rosto, durante ou no final da realização do movimento global. O atributo tempo define em qual momento do movimento o contato é realizado. O contato pode ser do tipo *toque*, *bater*, *escovar* (entra e sai de contato), *esfregar* (move, mas permanece na superfície) e *pegar*. O contato pode ocorrer de

uma local para o outro ou entre dois locais. Os atributos *local1* e *local2* são referentes a pontos de contato com a mão, partes do corpo ou rosto.

Para exemplificar o modelo de transcrição, alguns sinais serão descritos com a notação proposta.

O sinal "computador" é articulado com as duas mãos, de forma espelhada e configuração de mão em C. É realizado movimento global circular horário com a mão direita, e movimento espelhado com a mão esquerda. O XML que descreve o sinal "computador" é mostrado a seguir:

```
<sinal nome="computador">
  <suspensao numero="1">
        <maoDireita>
          <configuracao predefinida="c" />
          <localizacao>
            <espaco vertical="3" horizontal="3" />
          </localizacao>
          <palma orientação = "vertical"
                                posicao="dorso"/>
        </maoDireita>
        <maoEsquerda espelhada="sim"/>
      </suspensao>
   <movimentoGlobal numero="1" orientacao="vertical"
movimento="circularH" maos="espelhado" repetir="2"/>
</sinal>
```

Até o momento foram descritos sinais isolados da LIBRAS. No entanto para a reprodução de conteúdo, faz-se necessária a descrição de frases inteiras. Com isso, surgem os problemas de concatenação de sinais, omissão de partes de sinais, e articulações que não possuem nenhum sinal correspondente.

Considerando que os sistemas de transcrição tradicionais costumam representar os sinais isoladamente, a maneira como o sinal está inserido no contexto da frase também tem de ser interpretada pelo sinalizador virtual. A concatenação de sinais ocorre quando em uma frase existe a omissão de parte de um sinal com a sobreposição do sinal seguinte. Por exemplo, pode acontecer do movimento do sinal A começar antes que o movimento do sinal B termine, ocorrendo a sobreposição destes sinais. Não existe portanto garantia de que sempre os sinais serão reproduzidos em sua totalidade nas sentenças da LIBRAS. A composição de sinais na frase deve, portanto ser considerada na animação do avatar para que a reprodução não seja uma mera seqüência de sinais, não correspondendo a conversação real dos surdos.

Por exemplo, o sinal de "árvore" é articulado com o braço direito erguido na vertical, com a palma da mão aberta e os dedos afastados. O braço esquerdo serve como base, apoiando o cotovelo direito na mão esquerda. O sinal pegar começa com a mão aberta, dedos separados, e termina com a mão fechada.

O menino pegou uma fruta na árvore

A frase acima pode ser articulada da seguinte maneira: o braço esquerdo, passivo no sinal "árvore", pode ser usado para articular o gesto pegar, direcionado para a mão direita, que está simbolizando a copa da árvore, onde está a fruta. Neste caso houve uma concatenação de apenas dois sinais, "árvore" e "pegar", para formar uma frase inteira.

```
<sentenca>
    <sinal nome="bicicleta">
    <concatenar>
      <sinal nome="bicicleta" omitir="direita"/>
      <suspensao> (descrição de chapeu cair, mão
                                              direita)
      </suspensao>
    </concatenar>
</sentenca>
```

4. Conclusões

Neste trabalho foi apresentada uma notação XML para a língua de sinais brasileira.

Percebe-se que a descrição textual de uma língua de sinais não é tarefa trivial. Mesmo com o uso das notações já existentes, para o entendimento inequívoco de como reproduzir os sinais faz-se necessária a utilização de outras fontes de informações, como imagens e anotações adicionais.

O modelo de transcrição aqui proposto tem o objetivo de oferecer uma ferramenta de descrição dos sinais o mais detalhada possível, com o maior número de informações relevantes para sua reprodução computacional.

Com um modelo de descrição das línguas de sinais aprimorado, criado com o objetivo de reproduzir os sinais por um agente virtual, é possível aumentar a acessibilidade computacional aos portadores de deficiência auditiva, melhorando assim a interação homem-máquina para estes usuários.

Referências

[1] Censo Demográfico 2000. Características gerais da população. ISSN 0104-3145 Censo demogr., Rio de Janeiro, p. 1-178, 2000.

[2] Kennaway, R.J.., Glauert, J.R.W. e Zwitserlood, I. Providing Signed Content in the Internet by Synthesized Animation. ACM Trans Comput Hum Interact (TOCHI) 14, 3 (2007).

[3] http://www.unisc.br/universidade/estrutura_administrativa/nucleos/naac/alfabeto.htm, acessado em 04/01/2010.

Uma nova abordagem CBIR baseada em realimentação de relevância e classificação por OPF

André Tavares da Silva , Léo Pini Magalhães (Orientador) , Alexandre Xavier Falcão (Co-Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

{atavares,leopini}@dca.fee.unicamp.br, afalcao@ic.unicamp.br

Abstract – More recently, some CBIR approaches have shown the use of relevance feedback to train a pattern classifier which selects relevant images for retrieval. This paper revisits this strategy by using an optimum-path forest (OPF) classifier. During relevance feedback iterations, the proposed method uses the OPF classifier to decide which database images are relevant or not. Images just classified as relevant are sorted and presented to the user for a new iteration. Such images are ordered according to the normalized distance using relevant and irrelevant prototypes, computed previously by the OPF classifier. Our experiments show that the proposed approach requires few iterations, being faster and more effective than methods based on support vector machines.

Keywords – CBIR, relevance feedback, image procesing.

1. Introdução

Com o crescimento da internet e a popularização dos dispositivos para captura de imagens como câmeras digitais e *scanners*, a disponibilidade de coleções de imagens tem crescido rapidamente nos últimos anos [3]. Por isso, os usuários necessitam cada vez mais de ferramentas eficientes para pesquisar, navegar e recuperar essas informações em diferentes domínios, como sensoriamento remoto, moda, prevenção de crime, publicidade, medicina, arquitetura, entre outros. Para este propósito, têm sido desenvolvidos muitos sistemas de recuperação de imagens.

Existem duas linhas principais: baseados em texto e em conteúdo. Na abordagem baseada em texto, o processo de recuperação consiste em comparar os termos de uma consulta textual, definida por um usuário, com as anotações associadas às imagens e, a partir dessa comparação, retornar um conjunto de imagens. Existem duas principais desvantagens nesta abordagem: a necessidade de um trabalho humano considerável para realizar as anotações e a imprecisão das anotações devido à subjetividade da percepção humana [1], já que diferentes pessoas podem associar diferentes anotações para uma mesma imagem.

Para superar essas desvantagens em sistemas de recuperação de imagens, foram introduzidos os sistemas de recuperação de imagens baseados em conteúdo (CBIR - *content-based image retrieval*). Nos sistemas CBIR, as imagens são indexadas pelo seu conteúdo visual, tais como cor, textura e forma. Nesses sistemas, a anotação manual não é necessária. O processo de busca consiste basicamente em, dado um padrão de consulta (por exemplo uma imagem), calcular a sua similaridade em relação às imagens armazenadas na base, exibindo as mais similares.

Um grande desafio na recuperação de imagens é saber interpretar o desejo do usuário [5]. E esse desejo varia conforme a realidade cultural de cada pessoa, ou seja, imagens têm significado diferente para cada indivíduo e depende do conhecimento e vivência que cada um tem sobre as imagens.

A técnica de realimentação de relevância tem sido bastante utilizada para diminuir a lacuna semântica existente entre os sistemas computacionais e a subjetividade das pessoas. Essa técnica possibilita ao usuário expressar sua necessidade na especificação de uma consulta sem precisar recorrer a propriedades de mais baixo nível para representação da imagem. O usuário informa quais as imagens ele considera relevantes em um conjunto de imagens retornado pelo sistema. O algoritmo de realimentação de relevância aprende a vontade do usuário durante um determinado número de iterações. Dessa forma, o sistema retorna imagens cada vez mais similares à vontade do usuário, aprendendo o conceito estabelecido por ele.

Neste trabalho, propomos uma nova abordagem para CBIR com realimentação de relevância utilizando informações de imagens relevantes e ir-

relevantes selecionada pelo usuário. Para um determinado conjunto de imagens relevantes e irrelevantes, o método calcula uma OPF (Optimum-Path Forest) [4]. Apenas as imagens classificadas como relevantes são ordenadas pela distância e apresentadas ao usuário na próxima iteração. Mostramos que essa estratégia é realmente muito eficaz reduzindo consideravelmente o número de iterações necessárias.

Este artigo está organizado da seguinte forma: na seção 2 são expostos alguns conceitos básicos sobre realimentação de relevância, na seção 3 é apresentada a proposta da tese e por fim, são apresentadas as contribuições e perspectivas sobre o presente trabalho.

2. Conceitos Básicos

Esta seção apresenta alguns conceitos básicos sobre recuperação de imagens por conteúdo adotados neste trabalho.

2.1. Descritores

A extração de características (descritores) é a base da recuperação de informação visual [6]. A percepção visual dos objetos é subjetiva e por isso não existe uma única representação e nem mesmo uma melhor representação para uma dada característica. As principais características extraídas de imagens são cor, textura e forma.

A cor é provavelmente a característica mais utilizada para recuperação visual. Ela é relativamente robusta por apresentar independência do tamanho da imagem e da orientação da mesma. Uma vantagem do uso desse tipo de classificador é que as cores podem ser facilmente associadas a descrições textuais (nome da cor), facilitando a utilização em muitos sistemas CBIR. O histograma de cores é a característica mais utilizada para representar imagens.

Textura é uma propriedade presente em praticamente todas as estruturas, como nuvens, vegetação, paredes, cabelo e outros. Ela contém informação importante sobre o arranjo estrutural da superfície e sua relação com o ambiente.

Os primeiros descritores para descrever a forma de objetos em uma imagem se resumiam em definir informações simples como comprimento, perímetro, área e algumas relações entre essas medidas (regularidade e compacidade). Mais tarde, outros descritores mais complexos foram definidos, como os pontos de saliência, saliências de segmentos (*Segment Saliences*), Fourier, invariantes de momento (*moment invariants*), entre outros.

2.2. Realimentação de Relevância

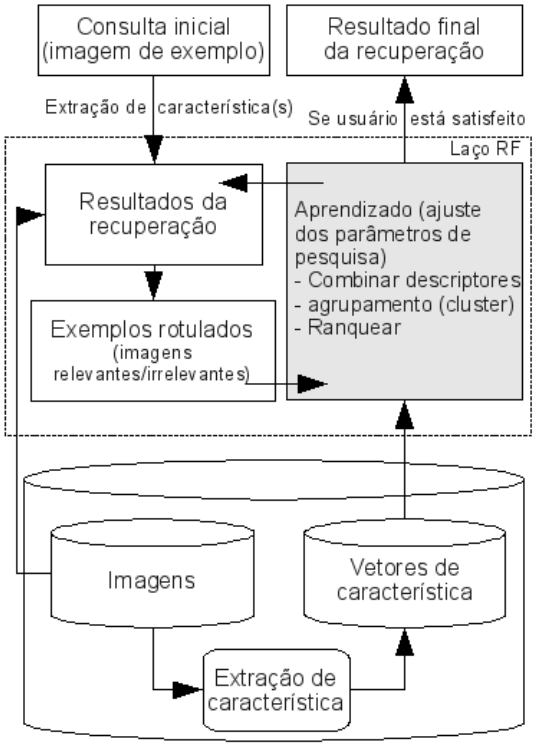


Figura 1. Arquitetura de um sistema de recuperação de imagens por conteúdo com realimentação de relevância.

Esse mecanismo (figura 1) tem por objetivo possibilitar que o usuário expresse a sua necessidade na especificação de uma consulta, sem recorrer a ajustes de propriedades de baixo nível utilizadas na representação de imagens. Para isso, o usuário apenas precisa indicar as imagens relevantes e, em certos casos, também as irrelevantes dentre um conjunto retornado pelo sistema. A cada iteração, o algoritmo de realimentação de relevância "aprende" quais propriedades visuais melhor definem as imagens relevantes a partir das informações fornecidas pelo usuário, ou seja, das imagens por ele indicadas. Assim, após um determinado número de iterações, o sistema retorna as imagens mais similares à imagem de consulta.

Realimentação de relevância endereça duas questões referentes ao processo de recuperação de

imagens por conteúdo. A primeira delas reside na lacuna semântica (*semantic gap*) entre as propriedades visuais de alto nível, através dos quais o usuário tem a percepção da informação visual, e a descrição de baixo nível utilizada para a representação das imagens [3]. A outra diz respeito ao caráter subjetivo da percepção da imagem pelo usuário. Diferentes pessoas, ou a mesma em diferentes circunstâncias, podem ter percepções visuais distintas de uma mesma imagem. Com realimentação de relevância essas duas questões são contornadas de forma transparente para o usuário.

O algoritmo de aprendizado é um ponto crucial para a definição de um mecanismo de realimentação de relevância. Em alguns trabalhos, o aprendizado consiste em estimar o vetor de característica que melhor representa o padrão de consulta. Em outros, atribuem-se pesos para cada posição do vetor de características e para cada descritor utilizado. Assim, o aprendizado consiste em estimar esses pesos, de forma a melhor representar a percepção visual do usuário. Existem diversos métodos para combinação de descritores, como Movimento de Ponto de Consulta, Aprendizado Probabilístico e Máquinas de Vetores de Suporte.

3. Proposta

OPF é um método de classificação, que representa cada classe de objetos por uma ou mais árvores de caminhos ótimos cujas raízes são amostras chamadas de protótipos. As amostras de treinamento são os nós de um grafo completo, cujos arcos são ponderados pela distância entre os vetores de características de seus nós. Na realimentação de relevância, temos duas classes: imagens relevantes escolhidas pelo usuário e as irrelevantes. Os protótipos escolhidos pelo classificador OPF, são então utilizados para classificar as imagens de acordo com a seleção do usuário.

Seja $\mathcal{Z}$ um de banco de dados de imagem. Para cada imagem $t \in \mathcal{Z}$, temos um vetor de características $\vec{v}(t) \in \Re^n$. Ou seja, cada imagem pode ser considerada um ponto no espaço $\Re^n$. A distância $d(s,t)$ entre duas imagens s e t é a distância entre seus vetores de características. Para um ponto inicial de pesquisa s, o método proposto retorna as N imagens $t \in \mathcal{Z}$ mais próximas a s (pesquisa por similaridade). Devido à lacuna semântica, as imagens mais próximas a s podem não ser as mais relevantes para um determinado usuário. Marcando as imagens que um usuário considera relevante ou não, são criados dois conjuntos: uma lista $\mathcal{I} \subset \mathcal{Z}$ de imagens irrelevantes e uma lista $\mathcal{R} \subset \mathcal{Z}$ de imagens relevantes. O método usa então os conjuntos $\mathcal{R}$ e $\mathcal{I}$ para treinamento da OPF. Apenas N imagens classificadas como relevantes mais próximas serão retornadas ao usuário na próxima interação.

Para realizar essa ordenação, são usados os protótipos relevantes ($\mathcal{A}$) e irrelevantes ($\mathcal{B}$) escolhidos na fase de treinamento. O método calcula a distância média $\bar{d}_{\mathcal{A}}(t,\mathcal{A})$ entre cada imagem $t \in \mathcal{Z}$ do banco de dados e imagens do conjunto de protótipos relevantes $\mathcal{A}$. Também é calculada a distância média $\bar{d}_{\mathcal{B}}(t,\mathcal{B})$ entre imagens do conjunto de protótipos irrelevantes $\mathcal{B}$. Finalmente, uma a distância média normalizada $\bar{d}(t,\mathcal{A},\mathcal{B})$ é calculada entre protótipos relevantes e irrelevantes: $\frac{\bar{d}_{\mathcal{A}}(t,\mathcal{A})}{\bar{d}_{\mathcal{A}}(t,\mathcal{A})+\bar{d}_{\mathcal{B}}(t,\mathcal{B})}$.

Após classificar cada imagem do banco de dados, o método retorna ao usuário um novo conjunto de N imagens relevantes, que contém os menores valores de $\bar{d}(t,\mathcal{A},\mathcal{B})$. Esse processo é repetido durante algumas interações T e, finalmente, o sistema retorna todas as imagens relevantes obtidas nesse processo.

Usamos o descritor BIC com a distância dLog [7] para avaliar nosso método e comparamos sua eficiência usando a curva precisão-revocação contra dois outros métodos: um baseado em SVM proposto por Tong et al. [8] e outro usando *multi-point query* apenas com imagens relevantes. O primeiro, denominado SAL (SVM Active Learning), é também chamado por SVM_{ACTIVE} ou SVM_{AL} na literatura. Foi escolhido por ser baseado em uma técnica que é considerada o estado da arte em classificação de imagens. O segundo, chamado de QPM (Query Point Movement) [8], foi selecionado para mostrar a importância das imagens selecionadas como irrelevantes. Nossa abordagem é chamada aqui como OPF, já que é baseada em um classificador OPF.

As imagens a seguir, mostram a curva média de precisão-revocação para duas bases de dados (Corel e PASCAL) após três iterações comparando os métodos QPM, SAL e OPF. Quanto mais alta a curva, melhor o resultado. Podemos ver que nosso método supera a performance dos outros dois. Resultados obtidos por este trabalho podem ser vistos

em [2].

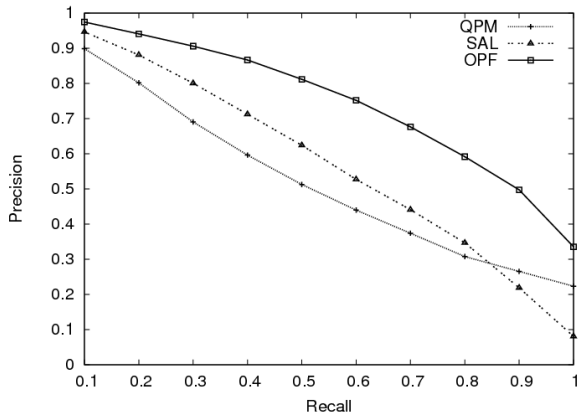


Figura 2. Curva média de precisão-revocação para a base Corel após três iterações.

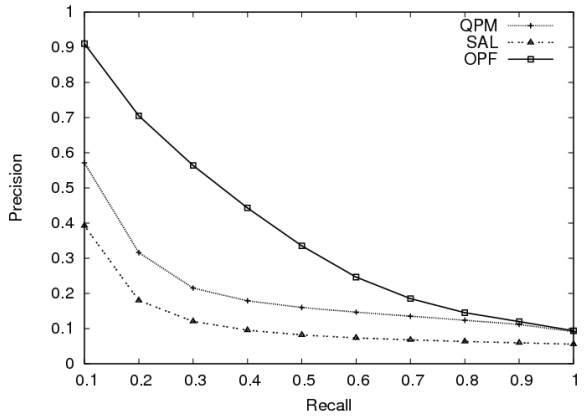


Figura 3. Curva média de precisão-revocação para a base PASCAL após três iterações.

4. Considerações Finais

Pretende-se neste trabalho desenvolver um algoritmo de aprendizado que consiga aprender o mais rápido possível a vontade do usuário.

Existe atualmente uma grande quantidade de imagens digitais disponível, principalmente na Web. Essas imagens vêm sendo geradas, manipuladas e armazenadas em diferentes locais. Para recuperar imagens, é necessário um método eficiente e eficaz para isso.

Os resultados obtidos mostram que nosso método, chamado OPF, necessita de poucas iterações e supera a performance dos outros métodos testados. O número de imagens perdidas pela classificação é insignificante (entre 0,2 e 3%). Além disso, uma vantagem do nosso método é que ele é aproximadamente vinte vezes mais rápido que o SAL.

Como trabalho futuro, pretendemos usar múltiplos descritores e técnicas para combiná-los. Também pretendemos comparar nosso método com outros mais recentes.

Referências

[1] P. Alshuth, T. Hermes, J. Kreyb, and M. Roper. *Intelligent Retrieval for Images and Videos*, volume 8. World Scientific, 1997.

[2] A. T. da Silva, A. X. Falcão, and L. P. Magalhães. A new cbir approach based on relevance feedback and optimum-path forest classification. In *Proc. WSCG 2010*, Plzen, Czech Republic, 2010.

[3] Ying Liua, Dengsheng Zhanga, Guojun Lua, and Wei-Ying Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, (40):262–282, 2007.

[4] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, 19(2):120–131, 2009.

[5] Yong Rui, Thomas S. Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, (10):39–62, 1999.

[6] Arnold W.M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, Dezembro 2000.

[7] R. O. Stehling, M. A. Nascimento, and A. X. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 102–109, New York, NY, USA, 2002. ACM.

[8] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM.

Visualização de Propagação de Ondas Mecânicas

Rodrigo Mologni Gonçalves dos Santos, José Mario de Martino (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

{mologni, martino}@dca.fee.unicamp.br

Abstract – This paper presents a author's master degree dissertation project proposal; whose objective is develop a software that enables the interactive visualization of wave propagation described in files generated by simulation, in order to solve the need of FEM's Computational Mechanics Department. The theoretical basis to be applied on the proposal comes from scientific visualization, which provides a kit of techniques for data visualization. A study was conducted prior to choose the environmental development and more favorable experiences that demonstrated the advantages of adding the VTK to project.

Keywords – Computer graphics, scientific visualization, mechanical waves.

1. Introdução

A onda mecânica é uma perturbação que se propaga, sem transporte de matéria, ao longo de um meio material elástico. O som é um exemplo de onda mecânica; cuja energia é transmitida por oscilações longitudinais e a propagação é visivelmente imperceptível. Mesmo quando visíveis, como é o caso das ondas oceânicas, a velocidade com que a propagação ocorre pode prejudicar a visualização. Isto porque a velocidade de propagação de uma onda depende das propriedades inercial e elástica do meio material. A percepção visual e a velocidade de propagação das ondas, tal como exemplificados, são dois dentre outros fatores que dificultam o processo cognitivo de propagação de ondas mecânicas.

As ondas podem ser descritas matematicamente e, portanto, simuladas por computador. E com o apoio da visualização científica é possível visualizar a propagação de ondas mecânicas, sejam simuladas por computador ou amostradas de um meio físico por equipamentos específicos de captação. Com isso, as ondas mecânicas, mesmo quando visivelmente imperceptíveis, podem ser visualizadas e a velocidade de propagação, dentre outras propriedades que descrevem o comportamento das ondas, controlada durante a simulação. Portanto, com a visualização científica é possível ampliar a capacidade do processo cognitivo humano no que se refere à propagação de ondas mecânicas.

A possibilidade de visualização e exploração de dados que descrevem o comportamento de ondas mecânicas é um desafio levantado pelo Prof. Dr. Euclides de Mesquita Neto (E.M.N.) do Departamento de Mecânica Computacional (DMC) da Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas. No qual auxiliaria, por exemplo, na detecção de regiões onde as interferências das ondas poderiam provocar abalamento de estruturas.

2. Proposta

O objetivo deste projeto de dissertação de mestrado é desenvolver um programa por meio do qual o usuário (especialista em mecânica das ondas) possa visualizar e explorar, de maneira interativa, a propagação de ondas mecânicas descrita em um arquivo de texto gerado pelo Wanglay (um programa desenvolvido no DMC que simula a propagação de ondas mecânicas).

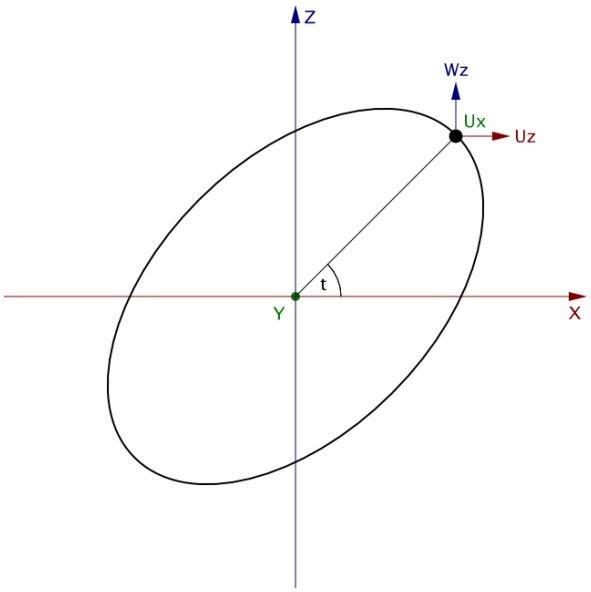


Figura 1. Trajetória do movimento de um ponto influenciado pelos vetores complexos de deslocamento ao longo do tempo.

2.1 Simulação pelo Wanglay

A simulação de uma onda mecânica superficial, por exemplo, gera um conjunto de pontos igualmente espaçados. E para cada um destes pontos os vetores complexos de deslocamento *Ux*, *Uz* e *Wz* que influenciam no movimento oscilatório de um ponto ao longo do tempo, respectivamente sobre os eixos *Y*, *X* e *Z*, tal como ilustrado na Figura 1.

Para visualizar o efeito de propagação da onda é necessário calcular para cada instante de tempo as novas coordenadas dos pontos sob influência de seus respectivos vetores complexos de deslocamento. A Equação 1 descreve as novas coordenadas (*x'*, *y'*, *z'*) de um ponto (*x*, *y*, *z*) em função do valor angular *t* que varia entre 0 e 1.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \Re(Uz) & \Im(Uz) \\ \Re(Ux) & \Im(Ux) \\ \Re(Wz) & \Im(Wz) \end{bmatrix} \cdot \begin{bmatrix} \cos(2 \cdot \pi \cdot t) \\ -\sin(2 \cdot \pi \cdot t) \end{bmatrix} \quad (1)$$

2.2 Ambiente de Desenvolvimento

Para o desenvolvimento do programa serão utilizados os seguintes recursos computacionais: a linguagem de programação Python, a biblioteca gráfica Qt (por meio do empacotador PyQt) e o conjunto de ferramentas de visualização de dados VTK.

O Python é uma linguagem de programação de alto nível, interpretada, de extensão, orientada a objetos, portável, flexível, de código aberto, e livre para uso e distribuição [4]. As principais vantagens para se adotar esta linguagem ao invés de outras comumente utilizadas, tais como C++ e Java, são: Qt e VTK possuem interfaces para Python e entre si; o tempo gasto para escrever um código em Python é, em média, duas vezes menor do que para C++ ou Java; o número de linhas de código necessário para desenvolver um aplicativo em Python é bem inferior que em C++ ou Java; e o tempo de execução de um aplicativo escrito em Python tende a ser inferior ao mesmo escrito em Java [6].

O Qt é uma biblioteca em C++ própria para o desenvolvimento de aplicações com interface gráfica. Como o Python é a linguagem de programação adotada para o desenvolvimento do programa, então será necessário utilizar o PyQt, um empacotador Qt para Python [7]. Em comparação com as bibliotecas gráficas AWT e Swing (ambas do Java), o Qt fornece os seguintes benefícios: as funcionalidades são mais intuitivas; necessita de um número menor de linhas de código; o controle dos eventos é mais simples; e não impõe paradigmas de programação [1].

O VTK é um conjunto de ferramentas em C++ próprio para visualização de dados [2]. Contém as principais técnicas para visualização de dados escalares, vetoriais e tensoriais, tais como: mapeamento em cores, efeito banda, contorno, glifos e linha de fluxo; além de pré e pós-classificação; visão estereoscópica; conversão entre espaços de cores; mapeamento em altura; etc. São duas as motivações para utilizá-lo no desenvolvimento do projeto: aproveitamento de funcionalidades já implementadas; e economia de tempo. Além disto, o VTK demonstrou-se eficaz para solução de problemas apresentados em trabalhos semelhantes: visualização de propagação de ondas acústicas em teatros [3] e visualização de campos eletromagnéticos (ondas não mecânicas) [5].

2.3 Processos de Desenvolvimento

A Figura 2 ilustra o fluxo dos processos que serão adotados para visualizar a propagação de ondas mecânicas, desde a obtenção dos dados até a representação visual com interação do usuário. Os processos, sinalizados por retângulos em branco, são cinco: importação, filtragem, mapeamento, renderização e interface.

A importação consiste na obtenção e conversão dos dados contidos em arquivo para uma representação processável pelo VTK, no caso uma malha estruturada, já que os pontos contidos em arquivo são igualmente espaçados e quando deslocados não geram sobreposição. Na filtragem os dados podem ser reamostrados, eliminados ou enriquecidos com informações

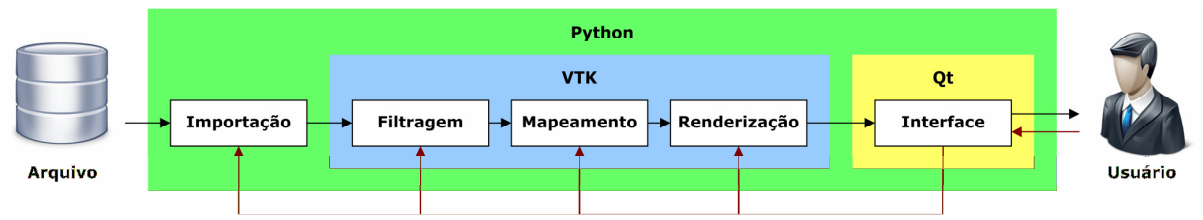


Figura 2. Fluxo dos processos necessários para visualização da propagação de ondas mecânicas.

adicionais. No mapeamento os dados pré-processados são mapeados em atributos gráficos, tais como pontos, polígonos e vetores. Na renderização ocorrem a tonalização, o cálculo de iluminação, o processamento de imagem, etc. A interface é o meio pelo qual o usuário interage com as funcionalidades do programa, visando permitir a visualização e exploração de regiões de interesse.

3. Resultados

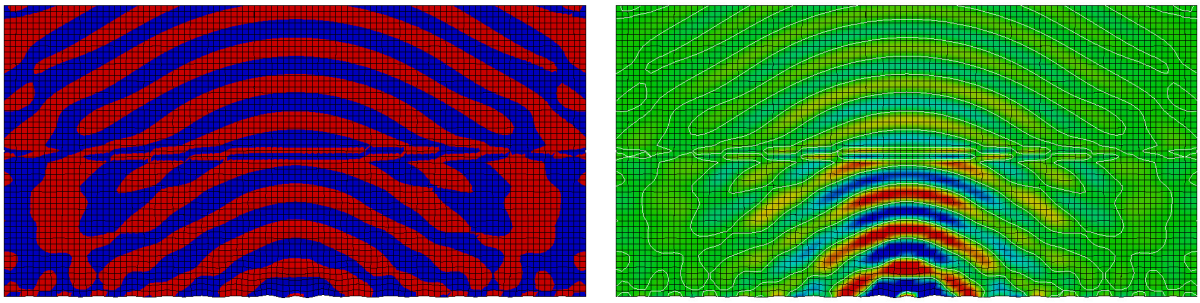
Para avaliar os benefícios da incorporação do VTK ao projeto, foram realizadas experiências pelo Paraview, um programa de visualização de dados que utiliza o VTK como base de suas funcionalidades. Entretanto, para que o Paraview pudesse interpretar os dados contidos em um arquivo gerado pelo Wanglay, foi necessário desenvolver um aplicativo que convertesse os dados para uma representação processável pelo VTK: o Wanglay2Paraview.

Para a realização dos testes foram utilizados dados que simulam a propagação de ondas mecânicas superficiais contendo 5.151 pontos. A Figura 3 apresenta a aplicação do efeito banda e contorno sobre a malha estruturada. Em "a" foi possível verificar claramente o comportamento das ondas ao longo da superfície, além de identificar uma barreira que interfere na propagação (centro da imagem). Em "b", a visualização propiciou a identificação de regiões onde a intensidade da onda é menor ou maior, e o contorno permitiu distinguir locais de transição.

Os vetores de deslocamento *Uz* e *Wz* sobre cada ponto podem ser visualizados na Figura 4, na qual foi possível verificar como a propagação ocorre sobre os eixos *X* (a) e *Z* (b).

A Figura 5 apresenta a aplicação do recorte e contorno por intersecção na malha representada tridimensionalmente por mapeamento em altura. Ambas permitiram visualizar regiões de interesse.

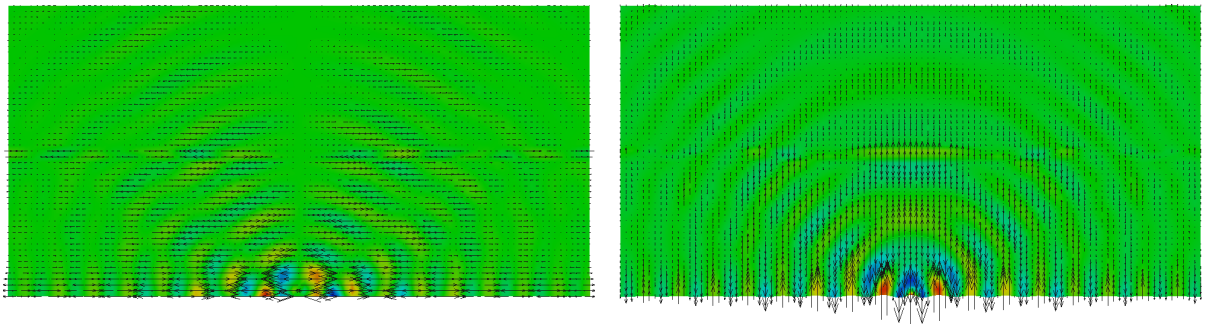
Outras experiências, não ilustradas, foram realizadas para visualizar os vetores normais, a trajetória de deslocamento dos pontos e o espaço ocupado pela onda. Todos os testes foram apresentados ao E.M.N. e estes demonstraram-se eficazes para visualização de propagação de ondas mecânicas.



a) Efeito banda: os pontos localizados na parte positiva do eixo *Y* foram mapeados em vermelho e os demais em azul.

b) Contorno: as linhas em branco sinalizam os locais de transição entre as partes positiva e negativa do eixo *Y*. Os pontos foram mapeados em cores variando do azul ao vermelho no modelo de cores HSV.

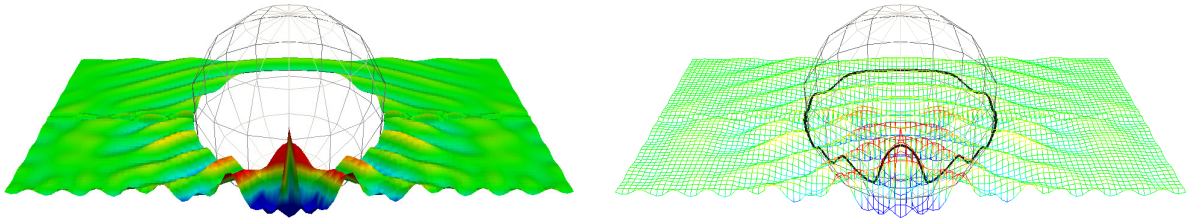
Figura 3. Efeito banda e contorno.



a) Vetores *Uz*: representados por glifos; indicam a propagação das ondas ao longo do eixo *X*.

b) Vetores *Wz*: representados por glifos; indicam a propagação das ondas ao longo do eixo *Z*.

Figura 4. Vetores *Uz* e *Wz*.



a) Recorte: resultado da subtração da malha estruturada com uma esfera.

b) Contorno por interseção: resultado da intersecção da malha estruturada com a superfície de uma esfera.

Figura 5. Recorte e contorno por intersecção.

4. Conclusões

A visualização científica disponibiliza um conjunto de técnicas para visualização de dados, que quando aplicados à visualização de propagação de ondas mecânicas, possibilita a visualização de ondas visivelmente imperceptíveis e o controle das propriedades que descrevem o comportamento das ondas.

As experiências realizadas com o Paraview demonstraram que a incorporação do VTK ao projeto favorecerá o desenvolvimento do programa, eliminando o tempo que seria gasto com as implementações das funcionalidades.

Com isso, pode-se concluir que o prognóstico é favorável ao cumprimento do objetivo do projeto: propiciar ao especialista condição para visualizar e explorar dados que descrevem a propagação de ondas mecânicas.

Referências

[1] Matthias Kalle Dalheimer. Qt vs. Java: a comparison of Qt and Java for large-scale, industrial-strength GUI development. Klarälvdalens Datakonsult AB, 2002. `http://turing.iimas.unam.mx/~elena/PDI-Lic/qt-vs-java-whitepaper.pdf`. (acessado em 19/01/2010).

[2] Inc. Kitware. The VTK user's guide: updated for VTK version 5. Kitware, Inc., 2006.

[3] Sami Khoury, Adrian Freed, and David Wessel. Volumetric modeling of acoustic fields in CNMAT's sound spatialization theatre. In: Proceedings of the Conference on Visualization '98 (Research Triangle Park, North Carolina, United States, October 18 - 23, 1998). IEEE Visualization. IEEE Computer Society Press, Los Alamitos, CA, 439-442.

[4] Mark Lutz. Learning Python: third edition. O'Reilly Media, Inc., 2008.

[5] Augusto Carlos Pavão, Eduardo Victor dos Santos Pouzada, and Marcio Antonio Mathias. Electromagnetic field visualization through VTK software. In: IEEE International Microwave and Optoelectronics Conference IMOC 2001, 2001, 2001. p. 21-24.

[6] Lutz Prechelt. An empirical comparison of C, C++, Java, Perl, Python, Rexx, and Tcl for a search/string-processing program. Fakultät für Informatik, Universität Karlsruhe, Germany, 2000. `http://page.mi.fu-berlin.de/prechelt/Biblio/jccpprtTR.pdf`. (acessado em 19/01/2010).

[7] Mark Summerfield. Rapid GUI programming with Python and Qt: the definitive guide to PyQt programming. Pearson Education, Inc., 2008.

Redes de Computadores e Sistemas Distribuídos

Ambiente de ensino de línguas baseado em Sistema Tutor Inteligente com agentes-lexemas ativos

Ismael Ávila, Ricardo Gudwin (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

i960946@dac.unicamp.br, gudwin@dca.fee.unicamp.br

Abstract – Este artigo descreve resumidamente um Sistema Tutor Inteligente para ensino de línguas no qual o objeto de ensino (a L2) é modelado em termos de seus lexemas e da inter-relação desses. Cada lexema da L2 comporta-se como um agente que busca ser aprendido em um ambiente multi-agente. Para ser aprendido, ele compete por um recurso limitado, a interface do curso, onde ele pode apresentar o seu significado por meio de imagens e outras pistas. Uma vez apresentado, o lexema ajuda outros lexemas com os quais ele pode formar sintagmas válidos para criar sentenças-exemplo. A meta individual de cada agente está sujeita a metas globais que, dependendo da estratégia didática, pode priorizar e favorecer alguns agentes em um mecanismo de seleção de ação. É aqui apresentada a estrutura do agente-lexema e são discutidas algumas implicações dessa abordagem.

Keywords – CALL, ITS, multiagent, domain model, NLP.

1. Introdução

Este artigo apresenta uma pesquisa em curso na área de Sistemas Tutores Inteligentes (STI) para o ensino de línguas. Sua principal contribuição é a abordagem em que cada lexema da língua-alvo (L2) é tratado como unidade pedagógica que age como um agente autônomo cujo objetivo é ser aprendido. Isso cria um arranjo *bottom-up* que é flexível para adaptar-se a mudanças nas metas didáticas e que reflete a natureza incremental da aquisição da língua materna (L1)[1]. Além disso, as relações entre os agentes simulam as típicas dependências que os lexemas correspondentes têm na gramática da L2. A seguir é apresentada a implementação e são discutidas resumidamente suas principais implicações.

2. Estratégia do ambiente de ensino

O ambiente STI descrito aqui visa a ensinar cerca de 2700 lexemas da L2, sendo 1500 substantivos, 800 verbos, 300 adjetivos e 80 conectores (que são preposições e conjunções). Cada lexema é programado para agir como um agente autônomo que busca ser aprendido, em um processo de seleção de ação como definido em [2]. Em outras palavras, cada agente-lexema tem o objetivo individual de ensinar seu próprio significado ao aluno, tanto isoladamente como na combinação com outros lexemas. Uma exibição na tela é uma condição necessária para que o aluno aprenda o sentido de qualquer lexema em particular, e a sequência de apresentações deve ser coerente para atender os objetivos e critérios pedagógicos.

O aprendizado de um lexema em resposta a sua exibição na tela pode ocorrer:
1) Com a ajuda de uma imagem ilustrativa (desenho ou foto)
2) Pela similaridade que o lexema possa ter com o lexema correspondente na L1 do aluno.
3) Pela relação com outros lexemas no contexto das sentenças-exemplo
4) Por meio de uma definição formal de seu significado

Para tornar possível cada uma dessas quatro possibilidades, cada agente-lexema tem: (1) uma imagem associada; (2) um valor que indica quão similar ele é em relação ao(s) correspondente(s) lexema(s) na L1 [3]; (3) uma lista de lexemas com os quais ele forma sintagmas e sentenças válidos na L2; e (4) uma definição formal de seu significado (escrita com outros lexemas da L2).

Visto que a tela é um recurso limitado, o agente-lexema precisa competir para ter acesso a ela. Assim, um agente permanece oculto na maior parte do tempo, e em qualquer momento a maior parte dos agentes não aparece, isto é, sua forma textual e sua imagem ilustrativa não são exibidas. Em qualquer momento somente um ou alguns poucos agentes (formando uma cena) são exibidos, e a sequência de exibições resulta de uma seleção de ações que é modulada pelas metas globais da estratégia pedagógica ou pelo tópico dado. Se o contexto, por exemplo, requer a exploração de temas tais como “alimentação” ou “viagem”, os lexemas pertinentes a esses temas ganham maior ativação e com isso têm maior probabilidade de serem exibidos primeiro. Ademais, a meta individual de um agente não é incompatível com os objetivos de outros agentes,

ela de fato depende deles para a construção de sentenças válidas. Isso implica a cooperação dos lexemas certos, na ordem correta. No ambiente, isso ocorre autonomamente e se assemelha às coalizões tais como discutidas em [4] e [5]. Portanto, além de competir por espaço na tela, cada agente também ajuda (envia ativação para) aqueles outros agentes com os quais ele precisa se associar para formar sentenças usuais na L2. Esses agentes candidatos a parceiros são listados dentro de cada agente, e o critério para pertencer a essa lista é frequência da combinação dos dois lexemas em usos reais da L2. Se o agente é um substantivo, possíveis parceiros são adjetivos e verbos, e vice-versa. Assim, as listas de agentes parceiros são usadas como as *condition-lists* e *add-lists* do mecanismo de seleção de ação. Essa é uma forma de traduzir as relações linguísticas em temos computacionalmente tratáveis. Tais listas podem ser vistas na Tabela 1, um exemplo resumido de dados contidos em agentes-lexemas.

Campo	*Tipo*	*Exemplo*
Substantivo	texto	luno
Imagem	imagem	G:\Images\luno.png
Nível de ativação	double	100
Temas	lista	[Nature; Science]
Simil. lexema L1	double	0,75*
Freq. no corpus	inteiro	10**
Possíveis adjetivos	lista	[bela, blua, kreskanta, nova, malkreskanta ronda, plena,]
Subjeito de	lista	[esti, brili, aperi, lumi, ravi, kaŝiĝi]
Objeto direto de	lista	[vidi, observi, rigardi, admiri, esplori]
Hiperônimos	lista	[satelito, astro]
Hipônimos	lista	[lunbrilo, lunradio, lunfazo, lunmonato]
Sinônimos	lista	[]
Antônimos	lista	[]
Definição	texto	La natura satelito de la Tero.

Tabela 1. Campos do agente-lexema "Luno" (lua).

Como mostrado na Figura 1, há uma fase inicial na qual todos os agentes-lexemas são importados de planilhas externas (as quais podem ser atualizadas ou corrigidas pelos tutores humanos se necessário). Uma vez importados, os agentes começam a competir por espaço na tela, e isso cria subsequentes ciclos de exibição. A aplicação tem também agentes auxiliares especializados, que são responsáveis por monitorar o acesso à tela, por fornecer modelos para a construção de sentenças e também por avaliar o progresso dos alunos por meio de exercícios e testes.

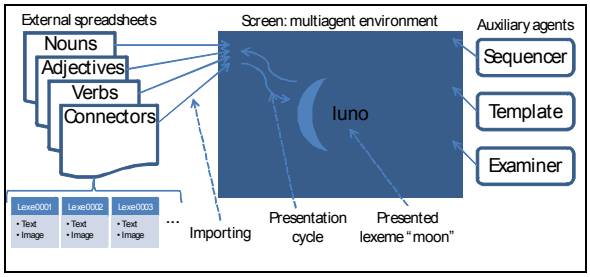


Figura 1. Esquema do STI com agentes-lexemas.

4. Conclusão

O ambiente STI descrito tem atributos adequados ao contexto de ensino de línguas. Primeiro, ele combina uma capacidade de tratar os conteúdos léxicos em seu nível atômico à oferta de recursos de alto nível para conduzir o processo de ensino a atender requisitos e cobrir temas específicos. Segundo, ele facilita a avaliação do processo de aprendizado ao garantir aos tutores humanos um acesso direto ao ciclo de vida instrucional de cada lexema em particular. Por fim, a resultante flexibilidade permite à aplicação lidar com a complexidade do ensino de línguas e com a diversidade de alunos.

Referências

[1] Bloom, P.: How Children Learn the Meanings of Words. MIT Press, Cambridge, MA (2000).

[2] Maes, P.: How to Do the Right Thing. Connection Science Journal. v.1, n.3, 291-323 (1989).

[3] Ávila, I; Gudwin, R.: Lexical similarity metrics for vocabulary learning modeling in Computer-Assisted Language Learning (CALL). Workshop: NLP in support of Learning: Metrics, Feedback and Connectivity, AIED'09, Brighton, 2009.

[4] Baars, B.J.: In the theater of consciousness: The Workspace of the Mind. New York, NY: Oxford University Press, 1997.

[5] Dubois, D.: Constructing an Agent Equipped with an Artificial Consciousness: Application to an ITS. PhD Thesis at the University of Quebec, Montreal, 2007.

* Supondo-se que a L1 é o inglês, que tem "lunar".

** De 1 a 10 , indicando uma frequência decrescente.

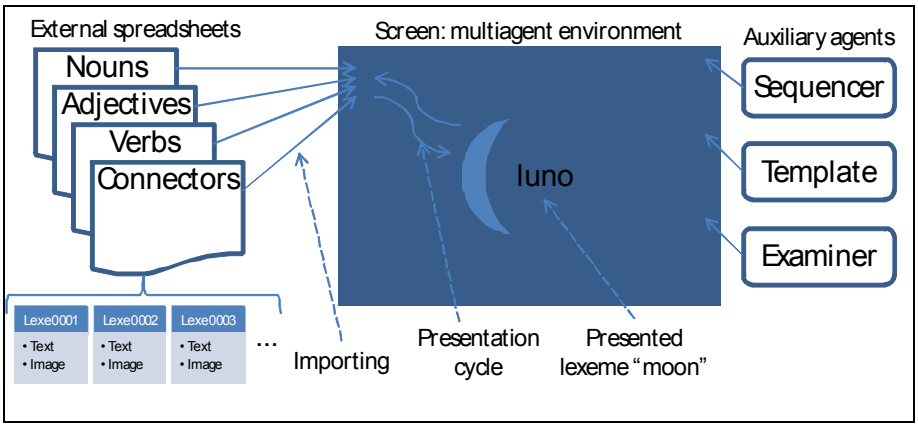


Fig. 1. Scheme of the ITS CALL tool with autonomous lexeme-agents.

As depicted in Figure 1, there is an initial phase during which all lexeme-agents are imported from external spreadsheets (which, by the way, can be updated or corrected by human tutors if necessary). Once imported, the agents start competing for space on the screen, and this creates subsequent presentation cycles. The application also uses some specialized auxiliary agents, which are responsible for monitoring the access to the interface, for providing some templates for the construction of sentences and also for assessing the learners' progress through exercices and tests.

3 Conclusion

The described ITS CALL tool has features well suited to a language-teaching context. Firstly, it combines a capacity of managing the lexical contents at their atomic level to the offering of higher-level resources to direct the teaching process to meet specific needs and cover specific themes. Secondly, it facilitates the assessment of the learning process by granting to the human tutors a direct access to the teaching life-cycle of any particular lexeme. Finally, the resulting flexibility enables the application to cope with the complexity of the language-teaching area and with the diversity of learners.

References

1. Bloom, P.: How Children Learn the Meanings of Words. MIT Press, Cambridge, MA (2000)
2. Maes, P.: How to Do the Right Thing. Connection Science Journal. v.1, n.3, 291-323 (1989)
3. Ávila, I; Gudwin, R.: Lexical similarity metrics for vocabulary learning modeling in Computer-Assisted Language Learning (CALL). Workshop: NLP in support of Learning: Metrics, Feedback and Connectivity, AIED'09, Brighton, (2009)
4. Baars, B.J.: In the theater of consciousness: The Workspace of the Mind. New York, NY: Oxford University Press, (1997)
5. Dubois, D.: Constructing an Agent Equipped with an Artificial Consciousness: Application to an ITS. PhD Thesis at the University of Quebec, Montreal, (2007)

Redes de Data Center com filtro de Bloom nos pacotes

Carlos A. B. Macapuna , Christian Esteve Rothenberg , Maurício F. Magalhães (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)

{macapuna, chesteve, mauricio}@dca.fee.unicamp.br

Abstract – This paper describes a networking approach for cloud data centers architectures based on a novel use of in-packet Bloom filters to encode randomized network paths. We present the design principles and testbed implementation of a data center architecture governed by Rack Managers, which are responsible to transparently provide the networking and support functions to cost-efficiently operate the DC network. We evaluate the proposal in terms of state requirements, our claims of false-positive-free forwarding, and the load balancing capabilities.

1. Introdução

Com o advento de serviços em nuvem para Internet, o suporte às redes de *data center* (DCN - *data center networks*) tornou-se um assunto de intensa pesquisa para aumentar sua escala, desempenho e relação custo-eficiência [3]. A fim de cumprir essas metas, sem comprometer a qualidade do serviço, inovações são solicitadas em muitas áreas do ambiente de *data center*, incluindo a própria infraestrutura de hospedagem (por exemplo, eficiência energética, fiação, acondicionamento) e a engenharia de rede (roteamento, virtualização, monitoramento).

Pesquisas recentes em re-arquitetura de *data center* têm estimulado projetos inovadores para interligar servidores, incluindo encaminhamento baseado na posição de pseudo endereço MAC [7], topologias *fat-tree* [1], ou balanceamento de carga usando uma camada 2 virtual [2].

Neste trabalho, descrevemos o encaminhamento utilizando filtro de Bloom nos pacotes, uma proposta DCN motivada por mudanças na rede e impulsionado pelo baixo custo dos *switches* com um substrato de programabilidade (OpenFlow [6]). Nosso projeto empresta algumas características de uma nova geração de DCN, por exemplo, a incorporação de controladores logicamente centralizados (4D [4]).

Basicamente, a ideia é interligar qualquer par de nós conectados dentro da DCN através de codificação da rota na origem em um filtro de Bloom, adicionado nos campos MAC Ethernet. Os objetivos do projeto incluem a conservação da semântica IP e eliminação de falsos-positivos explorando os múltiplos caminhos disponíveis. A solução proposta permite uma melhor utilização do espaço de 96-bits de origem e destino dos campos MAC, evitando, assim, o encapsulamento e, ao mesmo tempo, conserva a boa propriedade de *plug and play* do endereçamento Ethernet.

O resto do artigo está organizado da seguinte forma: a Seção 2 introduz informações relacionadas à lógica sobre a arquitetura DCN; a Seção 3 apresenta os princípios de projeto adotados para a nossa solução e descreve os principais blocos funcionais; na Seção 4 detalhamos a implementação do protótipo e do ambiente de testes; a Seção 5 avalia a proposta em termos de requisitos e estado de rede, falsos positivos, e capacidades de balanceamento de carga e, finalmente, a Seção 5 conclui o artigo.

2. Arquitetura do Data Center

O *data center*, como uma rede de interconexão para realizar tarefas de processamento distribuído, tem três principais elementos dominantes que determinam o seu desempenho: (1) a arquitetura de rede, (2) o esquema de roteamento, e (3) a topologia de interconexão. Nesta seção, descrevemos a estratégia adotada para resolver (1) e (2) e que pode ser resumida como uma abordagem separação identificador/localizador, onde os endereços IP atuam apenas como identificadores, e o roteamento é fornecido pelo encaminhamento baseado em filtro de Bloom nos pacotes. Quanto a (3), assumimos uma topologia em 3 camadas, uma inferior de *switches* ToR (*Top Of Rack*), uma camada intermediária de *switches* de agregação (AGGR), e uma camada superior de *switches* CORE (ver Fig. 1).

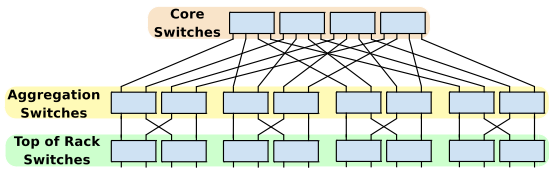


Figura 1. Arquitetura *fat tree* de 3 camadas.

2.1. Princípios de projeto

Com base na proposta de arquitetura de *data center*, nossos princípios de projeto são:

Separação identificador/localizador: A divisão entre identificador e localizador possui um papel fundamental para permitir o compartilhamento de recursos dos serviços com endereçamento IP. O IP é usado para identificar servidores físicos (e virtuais) dentro do DC, ou seja, não são impostas restrições de como os endereços são atribuídos ou utilizados para acesso externo (Internet), tornando-os não significativos para o roteamento de pacotes.

Rota na origem: Aproveitando o pequeno diâmetro das topologias de redes de *data center*, nossa abordagem utiliza o esquema de rota na origem (*strict source routing*). O roteamento nas topologias DCN em 3 camadas é bastante simplificado, ou seja, qualquer rota entre dois ToRs, tem uma trajetória ascendente em direção a um *switch* CORE e, em seguida, uma trajetória na direção do ToR destino, ambas passam pelos *switches* intermediários (AGGR). O encaminhamento é baseado em filtro de Bloom nos pacotes (iBF - *in-packet Bloom filter*) contendo apenas três elementos, nomeados de identificadores *Bloomed MAC* ($\langle CORE_i, AGGR_{down}, ToR_{dst} \rangle$) de *switches*. O ToR de origem calcula as rotas e envia para o AGGR mais próximo, com isso, três decisões são realizadas utilizando o iBF.

Diretório centralizado e controle de rede direto: Utilizamos a filosofia 4D [4] que simplifica o plano de dados e centraliza o plano de controle. Introduzimos o Gerenciador de *Rack* (RM) para tomar as decisões de roteamento e inserir as entradas dos fluxos nos *switches* programáveis.

Balanceamento de carga: A abordagem para fornecer o balanceamento de cargas é baseada no encaminhamento aleatório (*oblivious routing*). O RM é responsável pela seleção aleatória dos caminhos até o ToR destino.

2.2. Encaminhamento com identificadores *Bloomed MAC*

A originalidade, no caso, está associada à tomada de decisão nos *switches* AGRR e CORE. Inicialmente, as tabelas de encaminhamento estão vazias. Após a descoberta da topologia, as tabelas são preenchidas com uma entrada de fluxo para cada *switch* vizinho detectado. No lugar da tradicional combinação exata dos campos MAC, cada entrada de fluxo contém uma máscara de 96-bits gerada a partir de k funções de espalhamento (*hashing*) sobre o endereço MAC interno do *switch* vizinho. Um *ID Bloomed MAC* é um vetor de 96-bits onde apenas k bits são definidos com o valor igual a 1, como resultado da aplicação das k funções de *hashing*. Na chegada do pacote, apenas os 1s de cada *ID Bloomed MAC* são verificados quanto à presença nos campos MAC Ethernet (origem e destino) utilizados para transportar a rota na origem codificada no iBF. No caso de todos os 1s do *ID Bloomed MAC* corresponderem aos 1s definidos no iBF, o pacote é encaminhado para a interface correspondente.

2.3. Protocolo de descoberta

Uma questão não trivial é a descoberta da topologia da árvore e o papel de cada *switch* (isto é, ToR, AGGR ou CORE). Este conhecimento da topologia é um pré-requisito para viabilizar o roteamento na origem o que, além de reduzir os esforços operacionais, permite o encaminhamento correto e otimizado dos pacotes. Para este fim, criamos um protocolo (*Role Discovery Protocol*) que automatiza a inferência da árvore de *switches*, adicionando uma extensão no protocolo de descoberta LLDP. Nosso protocolo é bastante simples e requer apenas a identificação da camada em que o *switch* está situado.

3. Implementação e testbed do protótipo

A implementação do mecanismo de transmissão iBF é baseada em *switches* OpenFlow [6], enquanto o RM é implementado como uma aplicação adicionada ao controlador NOX [5]. A seguir, descrevemos as principais questões relacionadas à implementação do protótipo e do ambiente de testes.

3.1. OpenFlow

Um *switch* OpenFlow (OF) separa o encaminhamento de pacotes rápido (plano de dados) do nível de decisões de encaminhamento (plano de controle) de um roteador ou *switch*. Embora parte do plano de dados ainda encontre-se residente no *switch* e execute sobre o mesmo hardware (portas lógicas, memória), as decisões de manipulação de pacotes em alto-nível são movidas para um controlador separado. Dispositivos com OF habilitado e o(s) respectivo(s) controlador(eres) comunicam-se através do

protocolo OF (OFP - *Open Flow Protocol*), que define mensagens como *packet-received, send-packet-out, modify-forwarding-table* e *get-stats*.

O aspecto principal do OF é definir uma interface de captação na forma de uma tabela de fluxos, com entradas que contêm um conjunto de campos de pacote que combina uma tupla formada por 10 elementos: $\langle inport, Eth_{src}, Eth_{dst}, VLAN, EthType, IP_{proto}, IP_{src}, IP_{dst}, TCP_{src}, TCP_{dst} \rangle$, e uma lista de ações suportadas em hardware como, por exemplo, encaminhar para uma porta, encapsular e transmitir para o controlador ou descartar o pacote. A fim de apoiar o encaminhamento com base no iBF, apenas uma pequena alteração foi necessária na implementação do OpenFlow (v. 0.89rev2).

3.2. Gerenciador de Rack (RM)

O RM (*Rack Manager*) atua como um controlador de *switches* e a sua implementação é instanciada através de um aplicativo que executa no contexto do controlador NOX [5]. O NOX está disponível gratuitamente traduzindo-se em um importante *framework* para construir novas aplicações para interação com os dispositivos que possuem o OF habilitado. Em poucas palavras, a interface de programação do NOX é construída sobre os eventos, seja por componentes principais do NOX (*core*), ou definidas por usuários, e gerenciados diretamente a partir de mensagens OF como `packet-in`, `switch join`, `switch leave`, etc.

3.3. Ambiente de Teste (Testbed)

O ambiente de teste é composto por 5 nós físicos, um deles hospeda o controlador NOX com o componente RM e os 4 restantes são compartilhados, cada um, por 9 máquinas virtuais: 5 instâncias de OF *switches* e 4 nós finais. A Figura 2 mostra o *testbed*, onde as linhas sólidas representam ligações diretas entre as máquinas virtuais e as linhas tracejadas representam as conexões entre as máquinas virtuais de diferentes máquinas físicas. A topologia em cada máquina física é configurada com o OpenFlowVMS, o qual dispõe de um conjunto útil de *scripts* para automatizar a criação de máquinas virtuais em rede utilizando o QEMU. *Scripts* adicionais foram desenvolvidos para distribuir o ambiente em diferentes máquinas físicas usando conexões SSH e *switches* virtuais desprovidos de inteligência e baseados no VDE (*Virtual Distributed Ethernet*). O conjunto de *scripts* desenvolvido neste trabalho permite definir rapidamente uma topologia e automatizar a iniciação dos nós virtuais e do OF *switch*, incluindo a configuração de IP, a criação de *data path*, a iniciação do módulo OFP e conexão ao controlador.

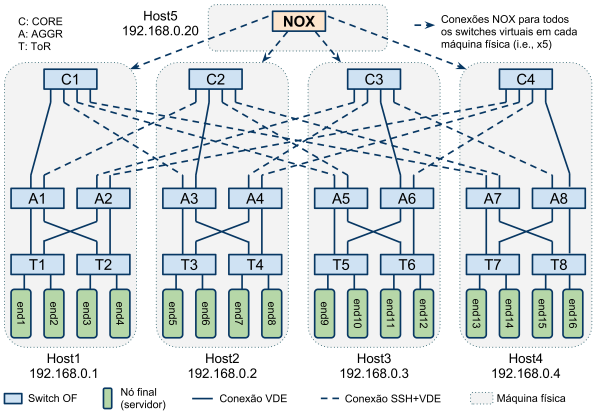


Figura 2. Ambiente de teste.

4. Avaliação

Depois de validar a implementação do protótipo através da verificação da conectividade total entre o conjunto de servidores (16 VMs), a próxima questão é avaliar o encaminhamento baseado no iBF em termos dos seguintes itens: (i) custo das informações de estado, (ii) os potenciais efeitos de falsos positivos, e (iii) a capacidade de balanceamento de carga. Devido às limitações de um ambiente virtualizado, os aspectos de desempenho não são considerados.

4.1. Análise do Estado

A configuração de rede é uma topologia em 3 camadas, com ToRs conectados a 2 servidores e dois AGGRs. As p_1 portas (p.e, p_1=4) do AGGRs são usados para conectar-se a $p_1/2$ ToRs e $p_1/2$ COREs. Em consonância com a literatura, assumimos uma média de 10 fluxos simultâneos por servidor (5 subindo e 5 descendo). Devido ao roteamento baseado na origem, nossa implemantação requer um estado mínimo nos COREs e AGGRs, ou seja, apenas uma entrada por interface (vizinho). Além disso, a escalabilidade no DCN não tem impacto sobre o número de entradas de fluxo nos *switches*, que é constante e igual ao número de vizinhos.

4.2. Falsos positivos

Avaliou-se o desempenho de falso positivo em filtro de Bloom de 96-bits quando se observa apenas

3 elementos, chamados de três endereços *Bloomed MAC*, que representam um caminho na rede na topologia DCN. A estimativa normalmente utilizada para a probabilidade de falso positivo de um filtro de Bloom de tamanho m, inserido com n elementos, com número de k funções *hash* é:

$$p^k = \left[1 - \left(1 - \frac{1}{m}\right)^{k*n}\right]^k \quad (1)$$

Temos que avaliar a viabilidade e a eficiência da nossa escolha de projeto para evitar falsos positivos, com base em descartar candidatos iBF propensos a falsos positivos antes da sua utilização. A nossa tese é que, dada a baixa fpr (*false positive rate*) de um iBF de 96-bits, há uma abundância de caminhos livres de falsos positivos entre quaisquer ToRs. A partir da teoria de filtros de Bloom, existe um número ideal de funções de *hash* ($k_{opt} = ln2 * m/n$, com $m = 96, n = 3$) que minimiza a probabilidade de falsos positivos que, no nosso caso, seria até 22 funções de *hashing*. Em nossa configuração prática, porém, o menor fpr foi obtido para k em torno de 7.

4.3. Balanceamento de carga

Dada uma matriz de tráfego (TM), o objetivo é avaliar como o tráfego é espalhado entre os enlaces disponíveis. Comparamos a utilização do enlace da nossa implementação com uma execução simplificada do *Spanning Tree Protocol* (STP) sobre a mesma topologia. Usamos ITG como gerador de tráfego, configurado com fluxos de TCP com duração de 10s, com tamanhos de carga exponencialmente distribuídos em torno de 850 bytes. Estes parâmetros são adequados para a maioria dos tráfegos DCN. A Figura 3 mostra a utilização normalizada após a repetição de dez experimentos. Como esperado, na STP as ligações de rede são muito utilizadas, enquanto o tráfego com iBF espalha adequadamente, com a utilização máxima e mínima normalizada de qualquer ligação desviando apenas cerca de 20% do valor ideal, ou seja, 1.

5. Conclusões

Apresentamos uma arquitetura de rede de *data center* com base em uma simples camada de plano de dados sob o IP, que encaminha pacotes baseado no

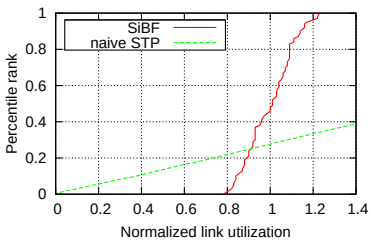


Figura 3. Testes.

conteúdo de um filtro de Bloom nos pacotes. A proposta DCN apresenta muitas características atraentes como, por exemplo, não exigir qualquer modificação nos nós finais, reutilizando o cabeçalho do pacote Ethernet, e um controle preciso sobre as rotas dos pacotes no *data center*. A avaliação sobre uma implementação de teste virtualizado em pequena escala, não só oferece uma prova de conceito, mas também lança luz sobre a capacidade de fornecer balanceamento de carga com iBFs. Em implementações futura, o protótipo será melhorado (por exemplo, para lidar com casos de falha) e estendido com características adicionais, como gerenciamento de banco de dados distribuídos.

Referências

[1] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. A scalable, commodity data center network architecture. *SIGCOMM CCR*, 38(4):63–74, 2008.

[2] Albert Greenberg, James Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David Maltz, Parveen Patel, and Sudipta Sengupta. Vl2: a scalable and flexible data center network. *SIGCOMM CCR*, 2009.

[3] Albert Greenberg, James Hamilton, David Maltz, and Parveen Patel. The cost of a cloud: research problems in data center networks. *SIGCOMM CCR*, 2009.

[4] Albert Greenberg, Gisli Hjalmtysson, David Maltz, Andy Myers, Jennifer Rexford, Geoffrey Xie, Hong Yan, Jibin Zhan, and Hui Zhang. A clean slate 4d approach to network control and management. *SIGCOMM CCR*, 2005.

[5] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. Nox: towards an operating system for networks. *SIGCOMM CCR*, 2008.

[6] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *SIGCOMM CCR*, 2008.

[7] Radhika Mysore, Andreas Pamboris, Nathan Farrington, Nelson Huang, Pardis Miri, Sivasankar Radhakrishnan, Vikram Subramanya, and Amin Vahdat. Portland: a scalable fault-tolerant layer 2 data center network fabric. In *SIGCOMM '09: ACM SIGCOMM 2009 conference on Data communication*, 2009.

Aperfeiçoando a navegação hiperbólica em um repositório de documentos por meio das tecnologias da Web Semântica

kadu Neves Batista Pereira , Ivan L. M. Ricarte (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil
Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA)
Caixa Postal 6141, 13083-886 – Campinas, SP, Brasil

{eng_kadu,ricarte}@dca.fee.unicamp.br

Abstract – Access to a set of documents whose content was classified according to a hierarchical structure of topics can be facilitated with the use of hyperbolic trees. However, this mechanism restricts the navigation to the topics selected for the organization of the hierarchy, eventually leaving to take for shipping other similarities or relationships between documents that have been classified into distinct branches of the hierarchy. The objective of this study is to explore this possibility, using for that purpose metadata associated with documents. Considering the particular case of documents on the Web, the appropriate mechanism for the organization of metadata is the use of Resource Description Framework (RDF), one of the key technologies of the Semantic Web. The adoption of such technology could allow, in future, the integration of this application to other Semantic Web.

Keywords – Semantic Web, Metadata, Resource Description Framework, Hyperbolic Tree.

1. Introdução

A representação e descrição dos recursos eletrônicos, tais como áudio, vídeo, imagens e textos podem ser feitos por meio de metadados. Dentre os tipos de formatos existentes para descrição dos metadados, estão o *Dublin Core* e o *Resource Description Framework* (RDF). Sua utilização favorece a representação de recursos eletrônicos, tornando-os mais visíveis aos motores de busca e sistemas de recuperação [2]. A Agência de Informação da Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA), serviço de informação voltado ao tratamento, qualificação e gestão da informação, possui um grande volume de informação técnico-científica, resultante de atividades de pesquisa e desenvolvimento agropecuários. Na implementação atual do sistema tais recursos são disponibilizados através dos hipertextos. Além disso, todos os hipertextos da Agência de Informação já possuem embutidos em seus códigos HTML metadados no padrão *Dublin Core*. Isso permite a recuperação de seus conteúdos pelos robôs de busca da web.

Uma das formas de acesso aos recursos disponibilizados pela Agência na *Web* utiliza a navegação por árvore hiperbólica [8], cujo conteúdo foi classificado de acordo com uma estrutura hierárquica de tópicos. No entanto, esse mecanismo restringe a navegação aos tópicos selecionados para a organização da hierarquia, deixando eventualmente de aproveitar para a navegação outras similaridades ou relações entre documentos que tenham sido classificados em ramos distintos da hierarquia. Nesse contexto, o trabalho visa explorar tal possibilidade, usando para tal fim metadados associados aos documentos. Considerando o caso particular de documentos na *Web*, o mecanismo apropriado para a organização desses metadados é o uso de RDF, uma das tecnologias básicas da Web Semântica.

2. Web Semântica

A Web Semântica é uma extensão da Web atual que permitirá aos computadores e humanos trabalharem em cooperação. Interliga significados de palavras e, neste âmbito, tem como finalidade conseguir atribuir um significado (sentido) aos conteúdos publicados na Internet de modo que seja perceptível tanto pelo ser humano como pelo computador [1]. RDF provê uma infraestrutura para representar informações sobre recursos na *Web* por meio de arquivos com metadados. É uma tecnologia endossada e recomendada pelo *World Wide Web Consortium* (W3C), tendo como principais objetivos criar um modelo simples de dados com uma semântica formal, usar o vocabulário baseado em *Uniform Resource Identifier* (URI) e uma sintaxe baseada em XML. Os arquivos RDF têm três componentes básicos formando uma tripla: Sujeito, propriedade e objeto, o que torna a linguagem altamente escalável [7]. No domínio de aplicação do sistema Agência

de Informação, o sujeito representa uma instância de entidade, representada por uma URI. A propriedade representa um atributo da entidade, enquanto o objeto representa o valor da propriedade.

2.1. Trabalhos Relatados

No que se refere a mecanismos de busca, uma das estratégias mais utilizadas é a expansão de consultas por meio de informações contidas em estruturas conceituais. Em [5] os autores propõem um algoritmo de expansão de consultas semânticas para recuperação de informações médicas. Sua abordagem consiste em identificar conceitos em um conjunto de termos MeSH (de Medical Subject Headings) em consultas de usuários e aplicar um algoritmo de expansão para incluir outros termos relacionados. Também é possível aumentar a eficiência de busca por conteúdos multimídia por meio da extensão de consultas feitas por usuários com a adição de informação semântica [9], bem como agregar termos a uma consulta de usuário por meio de uma expansão baseada em relações semânticas entre estruturas conceituais de domínios de conhecimento distintos [3].

A busca de um recurso por meio de navegação também pode ser melhorada com a associação de informação conceitual aos documentos. Tecnologias da Web Semântica como ontologias e metadados baseados em RDF podem ser utilizadas para prover melhor acesso e navegação em conteúdos disponibilizados na rede. Em [6], os autores abordam o desenvolvimento de um sistema de navegação por documentos baseado em similaridade semântica, a qual é representada por anotações semânticas obtidas automaticamente pela análise de conteúdos. Com a base de conhecimento gerada com a extração dessas informações, o sistema oferece ao usuário, na forma de um serviço Web, possibilidade de navegação pelos documentos organizados pela similaridade dos conceitos. No domínio de informações médicas, é abordado o desenvolvimento de um browser semântico que habilite, além de busca, navegação contextualizada com a integração de conhecimento obtido de portais de informações médicas. O sistema proposto utiliza ferramentas linguísticas para analisar artigos científicos e gerar anotações semânticas no formato RDF [4] .

3. Complementação de informações

Nesta seção descreveremos brevemente a metodologia utilizada no desenvolvimento do trabalho, tendo como base um subconjunto de documentos cujo domínio de aplicação é o cultivo da cana-de-açúcar, disponibilizados pela Agência de Informação Embrapa com foco na complementação de informações para o sistema de navegação hiperbólica utilizando RDF. A figura 1 traz um exemplo do conteúdo de nó contendo os recursos de informação selecionados para a seção **Informações Complementares**.

Figura 1. Informações Complementares disponível em conteúdos de nó.

3.1. Metadados

Os recursos de informação catalogados estão associadas aos nós selecionados na árvore do conhecimento como um complemento à informação contida nesse nó. Todos os recursos têm associados metadados no formato RDF composto por um conjunto de descritores aderentes ao padrão Dublin Core. Segue um exemplo dos elementos descritores *Dublin Core* encorporado ao modelo RDF de metadados.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/
    02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/
    elements/1.1/">
  <rdf:Description rdf:about=
    "http://www.cnptia.embrapa.br/
     index.hmtl">
     <dc:creator>Suzi</dc:creator>
     <dc:title>Embrapa Information
     Technology </dc:title>
     <dc:description>Research Unit Main
       Page.</dc:description>
     <dc:date>2001-01-20</dc:date>
```

```
  </rdf:Description>
</rdf:RDF>
```

Para o gerenciamento eficiente dos dados RDF foi utilizado Banco de Dados Relacional para armazenar o conjunto de triplas RDF, permitindo inferir sobre os dados por meio de consultas SQL. Inicialmente o trabalho se propõe e relacionar recursos de informação que contenha similaridades como mesmo autor, categoria e palavra-chave, de forma que posteriormente possam ser utilizados outros descritores para relacionar os recursos de informação.

3.2. Visualização hiperbólica

Ao acessar o conteúdo de um nó e navegar até o item **Informações Complementares**, o usuário terá disponível o *hyperlink* **Documentos Relacionado**, que irá direcioná-lo a outra página *Web* contendo a listagem dos recursos relacionados, bem como a opção de visualização por árvore hiperbólica, conforme pode ser visto na figura 2.

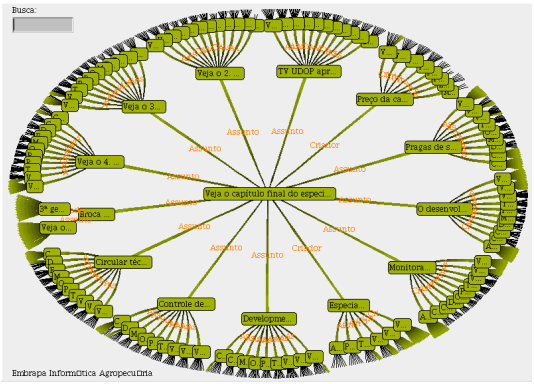


Figura 2. Visualização hiperbólica dos documentos relacionados.

4. Conclusão

Este trabalho apresentou uma abordagem para organização e visualização de recursos de informação na *Web* que tem por objetivo possibilitar ao usuário acesso a documentos relacionados através de similaridades e relações pertinentes ao contexto do ambiente informacional e que tenham sido classificados em ramos distintos da hierarquia.

O uso de metadados como mecanismo para estabelecer ligações semânticas entre recursos em diferentes ramos da árvore do conhecimento pode permitir, em trabalhos futuros, a integração dessa aplicação a outras da Web Semântica.

Referências

[1] T. Berners-Lee, J. Hendler, and O. Lassla. The semantic web. 5:34–43, May 2001.

[2] A. L. Boica and L. H. M. Oliveira. Conversão de metadados do padrão dublin core para o rdf. 1:11–17, 2008.

[3] M.A. de Leite and I.L.M. Ricarte. Fuzzy information retrieval model based on multiple related ontologies. In *Proc. 20th IEEE International Conference on Tools with Artificial Intelligence ICTAI '08*, volume 1, pages 309–316, 2008.

[4] G. Diallo, K. Khelif, O. Corby, P. Kostkova, and G. Madle. Semantic browsing of a domain specific resources: The corese-neli framework. In *Proc. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology WI-IAT '08*, volume 3, pages 50–54, 2008.

[5] V. Jalali and M.R.M. Borujerdi. The effect of using domain specific ontologies in query expansion in medical field. In *Proc. International Conference on Innovations in Information Technology IIT 2008*, pages 277–281, 2008.

[6] W. Kasper, J. Steffen, and Yajing Zhang. Using semantics for news navigation. In *Proc. IEEE International Conference on Semantic Computing*, pages 261–267, 2008.

[7] G. Klyne and J. Carroll. Resource description framework: concepts and abstract syntax. `http://www.w3.org/TR/rdf-concepts/`.

[8] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[9] M. Tvarozek and M. Bielikova. Personalized faceted navigation for multimedia collections. In *Proc. Second International Workshop on Semantic Media Adaptation and Personalization*, pages 104–109, 2007.

Mecanismos de segurança para grades de computação voluntária

Leonardo Laface de Almeida , Marco Aurélio Amaral Henriques (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

{lalmeida,marco}@dca.fee.unicamp.br

Abstract – Grid computing environments are used to process algorithms that require a large amount of computing power. Some of these environments take advantage of computing power from personal computers connected to the internet. Several security services are required by this kind of grid computing. In this document, we propose three security mecanisms. The first provides data authenticity, integrity and confidentiality. The second provides computer identification and the last improves the application result reliability using trust models.

Keywords – Data security, Volunteer grid computing, Trust models

1. Introdução

Uma grade computacional (*grid computing*) é um tipo de sistema de processamento paralelo que faz uso do poder computacional de computadores geograficamente dispersos e interligados por redes de alto desempenho para resolver problemas que requerem grande volume de cálculo. Algumas grades, chamadas de grades de computação voluntária, fazem uso de milhares de computadores pessoais conectados à *internet*, que sozinhos tem baixo poder de cálculo, mas juntos agregam uma alta capacidade computacional.

Para se usar eficientemente um sistema de processamento paralelo é necessário dividir um problema que exige grande poder computacional (aplicação) em pequenos problemas que exigem baixo poder computacional (tarefas) e os distribuir pelos computadores pertencentes ao sistema (trabalhadores). Depois que as tarefas foram computadas pelos trabalhadores, os resultados são agrupados de forma a se obter a solução da aplicação.

Assim como outros sistemas, as grades computacionais podem ser alvo de vários tipos de ataques, tais como invasão de computadores, proliferação de vírus, falsificação de resultados entre outros. As grades de computação voluntária estão mais vulneráveis a estes ataques porque utilizam computadores pessoais que, normalmente, não são administrados pela grade e nem sempre podem ser vinculados aos seus donos. Este trabalho propõe mecanismos para aumentar a segurança em grades de computação voluntária.

2. Segurança em grades

Do ponto de vista dos provedores de serviços de grade, dois aspectos de segurança se destacam:

1. proteção dos trabalhadores contra ataques provenientes de aplicações da grade;
2. proteção das aplicações contra ataques provenientes de trabalhadores.

2.1. Proteção para trabalhadores

Uma forma de oferecer segurança aos trabalhadores é identificando e autenticando os computadores responsáveis pela infaestrutura da grade. Isto pode ser feito utilizando a infraestrutura de chaves públicas (PKI), que ainda oferece a vantagem de garantir integridade das conexões.

A infraestrutura de chaves públicas se baseia em um modelo de confiança que utiliza certificados digitais criados e assinados por autoridades certificadoras. O dono de um certificado recebe duas chaves, uma privada e outra pública. A privada deve ser mantida sob sigilo. Caso contrário, a autenticidade do dono do certificado não poderá ser comprovada. Portanto, cabe aos trabalhadores confiarem nas autoridades que assinaram os certificados utilizados pelos computadores na premissa de que as chaves privadas são mantidas sob sigilo pelos seus donos.

2.2. Proteção para aplicações

Um dos serviços de segurança requeridos para aplicações é a identificação de trabalhadores. Esta identificação é necessária para que se possa punir de alguma forma os trabalhadores nocivos. Não é viável

a identificação utilizando certificados digitais devido ao custo elevado para gerenciar tais certificados em uma grade com grande número de trabalhadores. Utilizar o endereço IP também não resolve o problema, porque trabalhadores podem estar conectados a NAT's, impedindo sua identificação. Outros identificadores, tais como o endereço MAC, também não atendem as necessidades porque é um identificador facilmente clonável que está restrito a uma rede local. Algumas grades utilizam arquivos de texto (similares a arquivos *cookies* em *browsers*) para identificar trabalhadores [1].

Um outro serviço de segurança necessário é a detecção de resultados incorretos vindo dos trabahadores, uma vez que é difícil garantir que todos os resultados retornados são corretos. Alguns trabalhos propõem a utilização de réplicas de tarefas para verificar os resultados, utilizando métodos de inspeção, de votação e de reputação [2].

3. Mecanismos de segurança para grades de computação voluntária

Propomos a implementação de três mecanismos de segurança neste trabalho. O primeiro deles visa proteger os trabalhadores contra ataques oriundos de computadores responsáveis pela infraestrutura da grade. O segundo e o terceiro visam proteger as aplicações contra ataques oriundos de trabalhadores.

3.1. Uso de PKI para computadores da infraestrutura da grade

Algumas grades já utilizam PKI, alternativa que reforçamos neste trabalho. Ao utilizar a infraestrutura de chaves públicas, é possível garantir aos trabalhadores a autenticidade dos computadores responsáveis pela infraestrutura da grade. Além disso, por meio do uso de certificados digitais nestes computadores principais, é possível ainda prover sigilo e integridade das mensagens trocadas entre eles e os trabalhadores.

3.2. Identificação de trabalhadores

Algumas grades criam para os trabalhadores arquivos do tipo *cookie*, que possuem uma identificação única criada pela grade. Diferentemente desses trabalhos, propomos que este arquivo seja assinado utilizando uma chave privada de algum computador da infraestrutura da grade para garantir a origem e a integridade do arquivo. Isto diminui bastante a chance dos arquivos serem alterados, possibilitando a detecção de trabalhadores nocivos ao sistema. A proposta de gerenciamento pela grade dos arquivos que identificam trabalhadores é ilustrada na Figura 1.

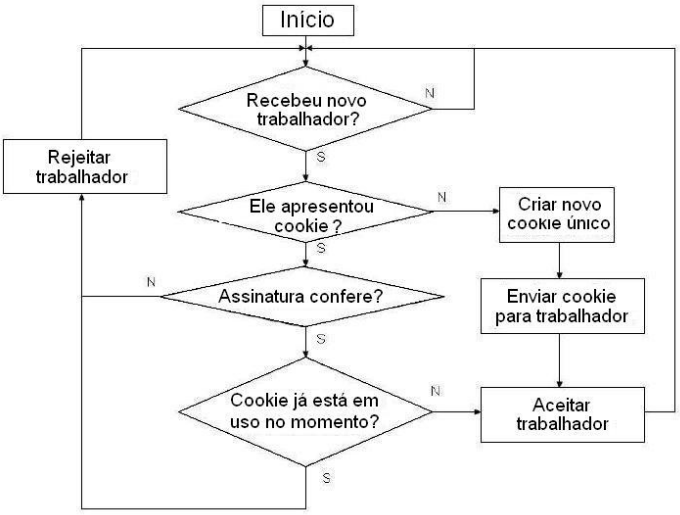


Figura 1. Gerenciamento de arquivos *cookies*

Com a utilização de *cookies*, é possível punir o trabalhador se a grade identificar qualquer ataque proveniente dele. Contudo, a identificação pode falhar porque o dono do trabalhador pode apagar o *cookie* depois de ser punido e inserir a mesma máquina como um novo trabalhador, recebendo uma nova identificação. Os donos dos trabalhadores devem restringir o acesso aos *cookies* para dificultar que terceiros os removam ou os copiem.

3.3. Verificação de resultados de tarefas

É proposto um mecanismo que utiliza os métodos de inspeção, de votação e de reputação, aproveitando a confiabilidade dos computadores responsáveis pela infraestrutura da grade. Nesta proposta, a grade deve classificar os trabalhadores, comparando os resultados retornados de tarefas e de suas réplicas.

Note que o número de réplicas por aplicação deve ser escolhido de forma a garantir maior confiabilidade sem comprometer significativamente o desempenho do sistema. É fundamental também evitar que os trabalhadores percebam que estão sendo testados. Para isso, propõe-se que:

- nenhum trabalhador execute a mesma tarefa mais de uma vez;
- as réplicas sejam criadas por amostragem a partir de tarefas convencionais e verificadas durante a execução de aplicações;
- a classificação (reputação) dos trabalhadores deve ser perene, isto é, mantida entre execuções de diferentes aplicações.

O mecanismo de verificação que segue o diagrama ilustrado na Figura 2. Trabalhadores classificados como desconhecidos são aqueles nunca testados pelo algoritmo. Trabalhadores honestos são aqueles aprovados no seu último teste. Trabalhadores suspeitos são aqueles reprovados no seu último teste. O trabalhador reprovado em dois testes consecutivos é testado isoladamente com um computador totalmente confiável, ou seja, pertencente à infraestrutura da grade. Se falhar também neste teste, ele será banido da grade. Certas precauções devem ser tomadas ao implementar esta proposta:

- todos os rótulos de trabalhadores honestos devem ter prazo de validade;
- trabalhadores suspeitos devem ser priorizados nos testes, mas nunca comparados entre si.

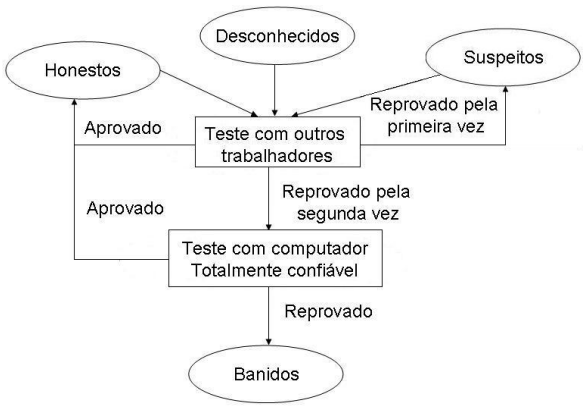


Figura 2. Definição de rótulos

Esta proposta propicia uma maior confiabilidade nos resultados de tarefas visto que alguns resultados coincidem em mais de um trabalhador. Além disso, os resultados de tarefas executadas pelos computadores totalmente confiáveis são considerados como corretos. Outra vantagem da proposta é que ela possui menor sensibilidade a erros transitórios, já que os trabalhadores não são banidos por retornar um ou outro resultado incorreto esporadicamente.

A proposta possui duas dificuldades. A primeira é detectar trabalhadores que retornam resultados corretos e incorretos alternadamente. A segunda é identificar grupos de trabalhadores que retonam resultados incorretos, porém idênticos entre si.

4. Simulações

Consideramos que um trabalhador que retorna ao menos um resultado incorreto é um trabalhador nocivo e que trabalhadores combinados são os que apresentam resultados incorretos e idênticos. Foram feitas simulações do algoritmo que seguem as seguintes condições:

- total de trabalhadores: 500 (constante);
- trabalhadores nocivos: entre 0% e 90% do total de trabalhadores;
- trabalhadores combinados: entre 0% e 100% do total de trabalhadores nocivos.

Uma maneira de avaliar o desempenho do algoritmo foi mantê-lo em loop até que todos os trabalhadores nocivos fossem banidos do sistema e cada trabalhador fosse testado ao menos uma vez. Estas condições são muito rigorosas e não são viáveis na prática, mas servem para avaliar o comportamento do mecanismo. Cada situação foi testada 500 vezes para cálculo de média e erro padrão e foi medido o volume de trabalho extra gasto pelo algoritmo até ele parar. Este volume é dado pelo número R de réplicas necessárias. Para efeito de comparação, foi utilizado um volume de trabalho relativo extra tendo como referência o volume de trabalho extra quando não há trabalhadores nocivos. Para rotular todos os trabalhadores, o volume de trabalho extra mínimo (usado como referência) é $R_{min} = \lceil W/2 \rceil$ réplicas, onde W é o total de trabalhadores.

5. Resultados

Em todas as figuras estão plotados também os erros padrão, que são muito pequenos e quase imperceptíveis. As Figuras 3 e 4 mostram o volume de trabalho relativo extra gasto pelo algoritmo quando não há trabalhadores nocivos combinados e quando todos os trabalhadores nocivos são combinados, respectivamente.

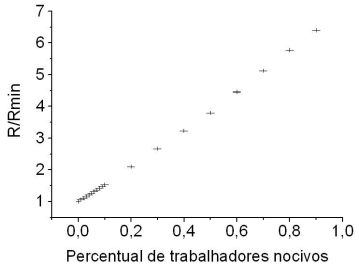


Figura 3. Volume de trabalho relativo extra sem trabalhadores combinados

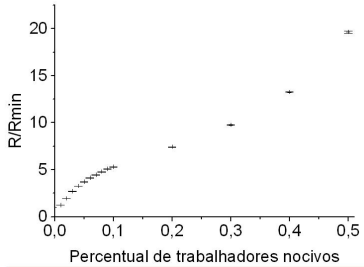


Figura 4. Volume de trabalho relativo extra sendo todos os trabalhadores nocivos combinados

É possível verificar que o mecanismo cria menos réplicas para situações mais comuns na prática, isto é, aquelas em que há mais de 90% de trabalhadores que retornam resultados corretos. Isso significa que o custo ao usar o algoritmo não aumenta muito em relação ao caso referência. Seja T o número total de tarefas de uma aplicação. Seja R/T o número relativo de réplicas necessárias para testar e banir todos os trabalhadores nocivos. Assim, é possível demonstrar que $R/T = 1/2 \times R/R_{min} \times (T/W)^{-1}$. As Figuras 5 e 6 mostram os valores de R/T necessários para que o algoritmo identifique todos os trabalhadores nocivos a medida que varia a relação T/W quando não há trabalhadores combinados e quando todos os trabalhadores nocivos são combinados, respectivamente.

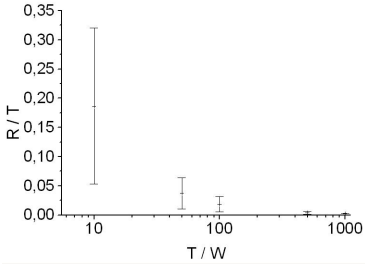


Figura 5. Valores de R/T sem trabalhadores combinados

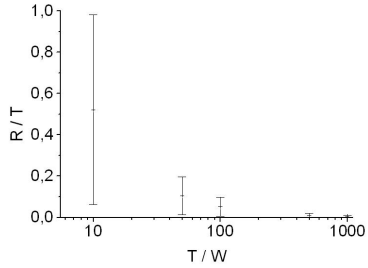


Figura 6. Valores de R/T sendo todos os trabalhadores nocivos combinados

É possível verificar que a medida que aumenta a relação T/W, o algoritmo se torna mais eficiente porque a relação R/T diminui. Em situações práticas, o valor de T/W costuma ser maior que 10 em grades de computação voluntária. Algumas aplicações possuem essa relação maior do que 100. Nestes casos, o mecanismo se mostrou eficiente, principalmente quando não há trabalhadores combinados.

6. Conclusões

Este trabalho propõe a implementação de três mecanismos para prover segurança para grades de computação voluntária. A primeira é a adoção de certificados digitais para identificar os computadores da infraestrutura da grade a fim de oferecer maior segurança aos trabalhadores. A segunda é a criação de arquivos *cookies* para identificar trabalhadores, viabilizando a proteção de aplicações contra ataques de trabalhadores. A última é a implementação de um algoritmo que utiliza réplicas de tarefas para classificar trabalhadores, oferecendo maior confiabilidade ao resultado da aplicação.

Simulações preliminares do algoritmo proposto mostraram que ele é eficaz para os casos em que o número total de trabalhadores é constante, sem causar grande impacto ao sistema. Testes em ambientes reais estão sendo feitos para comprovar a eficácia dos mecanismos propostos.

Referências

[1] Erik Elmroth, Mats Nylen, and Roger Oscarsson. A user-centric cluster and grid computing portal. *Int. J. Comput. Sci. Eng.*, 4(2):127–134, 2009.

[2] Luis F. G. Sarmenta. Sabotage-tolerance mechanisms for volunteer computing systems. In *CCGRID '01*, page 337, USA, 2001.

Sistemas Embarcados e Engenharia de *Software*

A Methodology for Effectiveness Analysis of Vulnerability Scanning Tools

Tania Basso, Regina L. O. Moraes (Co-orientadora), Mario Jino (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

taniabasso@gmail.com, {regina@ft, jino@dca.fee}.unicamp.br

Abstract – Software systems developed nowadays are highly complex and subject to strict time constraints and are often deployed with critical software faults. In many cases, software faults are responsible for security vulnerabilities which are exploited by hackers. Automatic web vulnerability scanners can help to reveal these vulnerabilities. Trustworthiness of the results these tools provide is important; hence, relevance of the results must be assessed. We analyzed the effect on security vulnerabilities of Java software faults injected into source code of Web applications. We assessed how these faults affect the behavior of the vulnerability scanner tool, to validate the results of its application. Software fault injection techniques and attack trees models were used to support the experiments. The injected software faults influenced the application behavior and, consequently, the behavior of the scanner tool. A high percentage of uncovered vulnerabilities as well as of false positives points out the limitations of the tool.

Keywords – Fault injection, Vulnerability Scanner tool, Web Application Security.

1. Introduction

Web applications are extremely popular nowadays. This type of application is becoming increasingly exposed as any security vulnerability can be exploited by hackers.

Automatic vulnerability scanner tools are often used to assess Web applications with respect to security vulnerabilities. Reliable results from vulnerability scanners are essential and the analysis of the scanners' effectiveness is important to guide the selection as well as the use of these tools. Previous research [1][2] shows that, in general, Web vulnerability scanners present a high number of false-positives (i.e., vulnerabilities detected by the tool that do not exist in the application) and low coverage (i.e., vulnerabilities that do exist in the application but were not identified by the tool), highlighting the limitations of this kind of tool.

Although other potential causes for vulnerability do exist, the root cause of most security attacks are vulnerabilities created by software faults [3][4]. Our proposal is to investigate the effect that Java software faults may have on security vulnerabilities and, then, analyze how they affect the behavior of the vulnerability scanner tool. The paper describes a method based on modelling of attack trees to define how to perform security tests by attacking the application.

The approach consists of injecting software faults into small Java applications to check if the scanner can detect potential vulnerabilities caused by the injected faults. Creation of vulnerabilities is confirmed through manual attacks, guided by the models of attack trees, to get accurate measures of detection coverage and false positives rate.

2. Software fault injection

Few works address the relationship between software faults and security vulnerabilities. A study by Fonseca and Vieira [5] analyzed security patches of web applications developed in PHP. The types of faults that are most likely to lead to security vulnerabilities are characterized.

The work by Basso *et al* [4] presents a field data study on real Java software faults, including security faults. The field study was based on security correction patches analysis available in open source repositories. More than 550 faults were analyzed and classified, determining the representativeness of these faults. The authors also define new operators, specific to this programming language structure, guiding the definition of a Java faultload.

The software fault injection technique used in this paper is the G-SWFIT [6], which is based on a set of fault injection operators that reproduce directly in the target executable code the instruction sequences that represent the most common types of high-level software faults.

To inject the faults, a use case of the application was selected. Each fault was injected in all possible locations of this specific use case,

one at time, forming different scenarios to be analyzed.

3. Effectiveness of vulnerability scanner tools

Web vulnerability scanners are regarded as an easy way to test applications against vulnerabilities. Most of these scanners are commercial tools (e.g., Acunetix [7], IBM Rational AppScan [8], N-Stalker [9] and HP WebInspect [10]).

Vieira *et al* [1] present an experimental evaluation of security vulnerabilities in publicly available web services. Four well known vulnerability scanners have been used to identify security flaws in web services implementations. Many differences in vulnerabilities were detected and a high number of false-positives and low coverage were observed when different tools were used to analyze the same application.

Fonseca *et al* [2] propose a method to evaluate and benchmark automatic Web vulnerability scanners using software fault injection techniques. Three leading commercial scanning tools were evaluated and the results have also shown that in general the coverage is low and the percentage of false positives is very high. However, these studies were focused on a specific family of applications: web services and PHP applications, respectively. Thus, the results obtained cannot be easily generalized. Furthermore, they do not present a clear methodology to validate the vulnerabilities detected by scanner tools. We investigate the behavior of scanner tools in the presence of injected Java faults, show a method using attack trees to model the possible ways to perform attacks to specific vulnerabilities, and analyze the results obtained by the scanner. This is addressed in the next sections.

4. Attack trees and security vulnerabilities

Attack trees provide a formal way of addressing security attacks on software systems [11]. In our work the attack trees are used to describe the various ways of attacking a specific type of security vulnerability. This is important to guide the security tests to validate the scanner results. We consider three types of security vulnerabilities: Cross-Site Scripting (XSS) [12], SQL Injection [13] and Cross-Site Request Forgery (CSRF) [14]. They were selected because they are widely spread and may cause major damage to the victims.

For each of these three types of vulnerability an attack tree was created. Figure 1 presents the attack tree for CSRF vulnerabilities. Due to space restrictions, the other trees are not presented, but they can be seen elsewhere [14].

In Figure 1, the first step to perform a CSRF attack is to have the user logged in the site because the attack will use the trust in user authentication. The next step is to analyze the request from the site that the attack will target in order to be able to reproduce it. If the site does not have CSRF countermeasures this step will lead to the next one because the request will be considered valid and will take effect on the site. If the site uses any defensive measure it will be necessary to analyze the request and take additional actions.

A known defensive method consists in appending different tokens to each request, but

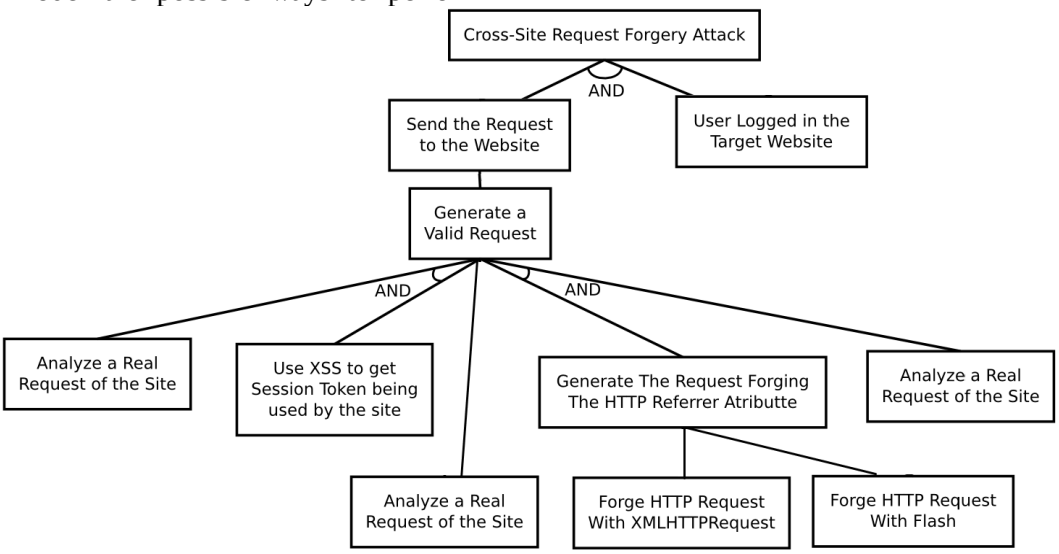


Figure 1. CSRF attack tree

this approach can be bypassed if the application is vulnerable to XSS attacks. The three remaining leaf nodes show how to overcome applications that use verification of the HTTP (Hypertext Transfer Protocol) Referrer attribute, although this is not a recommended defensive measure.

5. The experimental study

Two small open source Web applications developed in Java, App1 and App2, were selected to carry out the experiment. They are, respectively, a Customer Relationship Manager (CRM) and a management system for Distance Education, developed by the Brazilian federal government. They use technologies such as Hibernate and Ajax. We have chosen similar use cases from both applications to be the target piece of code of injected faults.

The types of fault to be injected were the most frequent ones observed by Basso et al. [4]. The security vulnerability scanner was selected because it is widely used and available. We do not identify it because commercial licenses do not allow the publication of tool evaluation results.

5.1 Injecting faults, executing the scans and validating the results

The tests start with a "Gold Run", where the application is tested once by the scanner tool without any fault injected. After the "Gold Run", one fault is injected. The context of the code where the fault is injected is analyzed to understand the effect of this fault in the applications behavior. Next, the code and the database are versioned, defining a scenario to be tested.

The scanner application is run and a verification of the results is done. If new vulnerabilities are detected, attacks are performed in the current scenario using the attack trees. This aims to verify if the new vulnerability actually exists or if it is a false positive. Then, the same attacks are performed in the original application scenario (without any fault injected) to verify if the vulnerability was present in the application before fault injection and was not identified by the tool (lack of coverage) when the Gold Run was performed.

The procedure is done for each possible location in the source code where faults can be injected in accordance with G-SWFIT technique (for the selected use case).

6. Results and discussions

For both Web applications, we analyzed, respectively, 11 and 23 different scenarios. Table 1 shows the total of scenarios that presented new security vulnerabilities detected by the scanner due to the fault injection.

Table 1. Applications scenarios and vulnerabilities

	App1	App2
Total scenarios analyzed	11	23
Scenarios with new vulnerabilities	5	7
% of faults that affected the scan	46%	30%

According to Table 1, about 35% of the injected software faults affected the scanner results. The lack of coverage and false positives rate are shown in Table 2.

In Table 2, the CSRF vulnerability represents 60% of the lack of coverage. In most of the cases, when scanning the application with fault injected, a new vulnerability detected by the tool was, in fact, one that already was present in the original application, not identified in the "Gold Run".

Table 2. Percentage of security vulnerabilities: lack of coverage and false positives

	XSS	SQL inject	CSRF	Total
Vulnerabilities	2	2	15	19
Lack of coverage (%)	0%	0%	60%	47%
False positive (%)	50%	100%	34%	42%

Also in Table 2, the false positives come from the three types of security vulnerabilities: XSS, SQL injection and CSRF, representing, respectively, 50%, 100% and 34% of the vulnerabilities detected. The false positive associated to the XSS vulnerabilities is considered because the scanner tool integrates outdated version of internet browsers. An attack successfully executed by the tool, when executed in the later versions of internet browsers, has no effect, because these versions implement features that do not permit the execution of common XSS attacks.

The SQL injection false positives were identified through the attacks and the analysis of the source code. Both applications use the Hibernate technology, and the way that the application was coded, i.e., extremely encapsulated, does not give opportunities to develop successful attacks.

Most of cases where CSRF false positives were identified were in error pages. A hacker

performing a CSRF attack to access an error page can be dangerous if the error page presents links or buttons which permit to access back the application (as "back" buttons which bring back the user to the last page he/she accessed). For both applications, the error pages do not present any way of accessing application functionalities or private information. Hence, we considered theses cases as false positives because a CSRF attack when accessing the error pages is useless.

The last column of Table 2 shows the total percentage of lack of coverage and false positives. From the 19 vulnerabilities investigated, 42% are false positives and 47% were not identified by the scanner tool. It indicates the limitations of this tool found in this study.

7. Conclusions

In this paper we present an experimental study where we analyze the effect of Java software faults, injected in the source code of Web applications, on security vulnerabilities. We also analyze the influence of these faults on the security vulnerabilities detection by a well known security vulnerability scanner tool. Fault injection techniques and attack tree modeling were used to support the experiments.

Results show that, according to the context of both the target code applications and the security vulnerabilities structure considered, the injected faults did affect the behavior of the application and, consequently, the behavior of the scanner tool in detecting new vulnerabilities. The scanner presented high percentage of lack of coverage and many false positives, showing its limitations. Factors that influenced this percentage are, in addition to the activation of the faults injected into the source code of the applications, the use of different development technologies (such as Hibernate) and some outdated features of the tool (as the internal internet browser).

We intend to extend this experiment by investigating the effect of other types of faults and the effectiveness of other vulnerability scanner tools. We also intend to develop a tool to perform the attacks (based on attack trees) automatically.

References

[1]M. Vieira, N. Antunes, H. Madeira. "Using Web Security Scanners to Detect Vulnerabilities in Web Services". *IEEE/IFIP Intl Conf. on Dependable Systems and Networks, DSN 2009*, Lisboa, Portugal, June 2009.

[2]J. Fonseca, M. Vieira, H. Madeira. "Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks", *13º IEEE Pacific Rim Dependable Computing Conference (PRDC 2007)*, Melbourne, Victoria, Australia, December 2007.

[3]J. Fonseca, M. Vieira. "Mapping software faults with web security vulnerability". *IEEE/IFIP Int. Conf. on Dependable Systems and Networks*, Anchorage, USA, 2008.

[4]T. Basso, R. Moraes, B. P. Sanches, M. Jino. "An Investigation of Java Faults Operators Derived from a Field Data Study on Java Software Faults." *In: Workshop de Testes e Tolerância a Falhas - WTF2009*, João Pessoa, Brazil, 2009, pp. 1-13.

[5]J. Fonseca, M. Vieira. "Mapping software faults with web security vulnerability". *IEEE/IFIP Int. Conf. on Dependable Systems and Networks*, Anchorage, USA, 2008.

[6]J. Durães, H. Madeira. "Emulation of Software Faults: A Field Data Study and Practical Approach". *IEEE Trans. on Software Engineering*, Nov. 2006, pp.849-867.

[7]Acunetix Web Appication Security. Available in http://www.acunetix.com, November/2009.

[8]IBM Rational AppScan. Available in http://www01.ibm.com/software/awdtools/appscan/, November/2009.

[9]N-Stalker. Available in http://www.nstalker.com/, November/2009.

[10]HP WebInspect. Available in https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-201-200%5E9570_4000_100__, November/2009.

[11]B. Schneir. "Attack Trees: Modeling Security Threats", Dr. Dobb's Journal, December, 1999

[12]CGISecurity.com. "The Cross Site Scripting FAQ." Available in http://www.cgisecurity.com/xss-faq.html, November/2009.

[13]W. G. Halfond, J. Viegas, A. Orso, "A classification of SQL injection attacks and countermeasures". In Proc.IEEE International Symposium on Secure Software Engineering, Arlington, Virginia, March/2006.

[14]R. Auger. "The Cross-Site Request Forgery (CSRF/XSRF) FAQ". Available in http://www.cgisecurity.com/csrf-faq.html, November/2009.

[15]Research Test Group. Available in http://www.ceset.unicamp.br/docentes/regina/projeto/, December/2009.

Aplicação de Dados Históricos para Seleção de Casos de Teste de Regressão

Camila Socolowski, Mario Jino (Orientador), Marcelo Fantinato (Co-orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

{camila.socolowski@gmail.com, jino@dca.fee.unicamp.br, m.fantinato@usp.br}

Abstract – Regression testing is applied to ensure the software quality across its multiple versions. To minimize the cost of this activity, test cases with higher capability of detecting faults should be selected, keeping the efficiency of the test suite resulting from the selection. In this paper, we propose an approach in which historical data are used as a selection technique in regression testing. The approach feasibility and usefulness are validated through its application to test a real world software. In the experiment, the proposed approach is compared with other two black box testing selection techniques, in terms of both the efficiency in the fault detection and the software reliability achieved with their application.

Keywords – regression testing, test case selection techniques, historical data, software reliability.

1. Introdução

Após seu desenvolvimento, um produto de software usado na indústria precisa evoluir devido à inclusão de novas funcionalidades requeridas pelo cliente à medida que seus negócios mudam. Além disso, defeitos são encontrados durante o uso do software em produção, provocando a necessidade de sua manutenção. Diante dessa evolução e manutenção, manter a qualidade do software após diversas versões é um desafio. Algumas vezes, a qualidade se deteriora devido às alterações realizadas [1].

O teste de regressão é usado para garantir a qualidade do software após diversas versões terem sido geradas. No entanto, ele é muito custoso, por requerer muitas execuções de Casos de Teste (CTs), cuja quantidade aumenta consideravelmente conforme o software evolui [2]. Esse alto custo leva à aplicação de técnicas de Seleção de Testes de Regressão (STR), que direcionam a seleção de apenas um subconjunto de CTs para re-execução. Em geral, a seleção ótima de testes, ou seja, aquela que seleciona exatamente os CTs que revelam todos os defeitos existentes, é impossível. Portanto, a análise de custo-benefício aplicada às diversas técnicas STR é um interesse central da pesquisa e da prática em teste de regressão [3].

Em geral, as técnicas STR são estudadas a partir de versões-base criadas ou identificadas de um sistema que acompanham os conjuntos de teste. Então, alguns algoritmos STR são executados e comparados em termos de tamanho e eficiência com o conjunto de testes original. No entanto, esses estudos empíricos consideram o teste de regressão em uma única sessão de testes, sem considerar restrições de tempo e de recursos do mundo real. De acordo com Kim e Porter [4], uma forma mais proveitosa de estudar as técnicas STR seria por meio de sessões de teste sujeitas a restrições que existem na prática. Além disso, dados históricos de desempenho dos CTs podem ser usados para melhorar o desempenho do teste de regressão a longo prazo, uma vez que as técnicas STR existentes são *memoryless*, ou seja, consideram apenas as informações obtidas a partir das versões atuais ou das imediatamente precedentes do software em teste.

Park *et al.*[5] usam dados históricos para uma tarefa similar à realizada pelas técnicas STR: priorizar CTs, definindo uma ordem em que todos os CTs do conjunto devem ser re-executados. Neste artigo, uma abordagem é proposta em que a técnica de priorização de Park chamada aqui de *Retest Using Historical Data*, é aplicada como uma técnica STR. Essa abordagem é comparada com duas outras técnicas STR, em termos tanto da eficiência na detecção de defeitos quanto da confiabilidade de software atingida com sua aplicação. A comparação é realizada por meio de um experimento em que os dados históricos são coletados em função do teste de um software do mundo real.

2. Proposta

Esse trabalho apresenta o projeto de mestrado que propõe o estudo e a implementação de uma

técnica de seleção de casos de teste caixa preta baseada em dados históricos, coletados de aplicações do mundo real. Um experimento será conduzido para validar essa técnica, comparando-a com outras técnicas de STR "caixa preta", como o Reteste de Casos de Uso de Maior Risco [6] e o Reteste por Perfil Operacional [7].

Por fim, será especificada uma ferramenta para registro de casos de teste e de defeitos encontrados que facilite o armazenamento e a recuperação dos dados históricos na execução de baterias de teste e que calcule automaticamente os valores históricos das execuções de teste para uso posterior nos testes de regressão. A especificação dessa ferramenta será composta pelos requisitos necessários para a construção da ferramenta, diagramas da UML (do inglês, *Unified Modelling Language*), alguns protótipos da interface gráfica e principais algoritmos utilizados.

2.1 Trabalhos relacionados

Outros trabalhos abordam técnicas para teste de regressão englobando seleção de testes de regressão, priorização de testes de regressão e redução da suíte de teste.

No trabalho de Graves et al. [8] é apresentado um experimento que demonstra os custos e benefícios de diversas técnicas de teste de regressão enfatizando a redução da suíte de teste e a detecção de defeitos. As técnicas consideradas pelos autores são as de minimização da suíte de testes, segura e fluxo de dados. Os autores consideram que as técnicas segura e de fluxo de dados são eficientes em detectar falhas, mas selecionam números amplamente variados de casos de teste. Em um ambiente restrito, tal abordagem pode ser simplesmente inviável [8].

Os trabalhos de Wong e Rothermel [2, 3] baseiam-se nas técnicas de priorização sem memória e modelam o teste de regressão como uma atividade de uma única vez, ignorando possíveis efeitos sobre múltiplas releases de software. Finalmente, não levaram em consideração as restrições de tempo e as restrições de recursos [9].

Kim e Porter [4] apresentam resultados iniciais de um estudo empírico sobre o uso de dados históricos de execução de teste para priorizar a seleção de casos de teste em um processo restrito de teste de regressão. Os resultados experimentais relatados pelos autores apóiam fortemente o princípio de que o teste de regressão pode ter que ser feito de forma diferente em ambientes restritos e não restritos. Os autores também apóiam que a informação histórica pode ser útil em reduzir custos e aumentar a eficiência de processos de teste de regressão de longa duração.

Park et al. [5] conduzem um experimento para validar e provar a eficiência de sua abordagem baseada em valor histórico para priorização de casos de teste que considera o fator custo. No experimento realizado foi usada a métrica APFDc (*Average Percentage of Faults Detected per cost*) para comparar a abordagem em questão com outra técnica de priorização de casos de teste baseada em cobertura. Como resultado do experimento foi comprovado que a abordagem que considera o fator custo produz melhores resultados, em termos de APFDc, do que as técnicas de priorização de casos de teste baseadas em cobertura.

Uma análise mais específica dos trabalhos mencionados nesta seção mostrou que na área de testes a maior parte dos trabalhos relativos a regressão baseiam-se na aplicação de técnicas que dependem da análise de código fonte do software em teste. Isso pode acarretar dificuldades para selecionar estratégias de regressão em situações em que não é possível ter acesso ao código fonte para análise.

Outra questão importante mencionada por Kim e Porter [4] e Park et al. [5] é a necessidade de realizar experimentos para abranger uma variedade mais ampla de defeitos que ocorrem naturalmente em sistemas reais com restrições de custos, tempo e recursos e comparar os resultados com outros métodos não baseados em histórico descritos na literatura.

2.2. Experimento

A abordagem proposta foi aplicada em um software para Web do mundo real, desenvolvido para o Serviço da Receita Federal Brasileira (SRFB). Desse software, foram levados em conta apenas CTs caixa preta, ou seja, apenas aspectos funcionais foram considerados para a coleta de dados e para a aplicação da abordagem proposta. Por ser um software do mundo real, os defeitos encontrados no experimento são reais.

Para fins de comparação de resultados com a técnica *Retest using Historical Data*, ambas as técnicas Reteste de Casos de Uso de Maior Risco e Reteste por Perfil Operacional foram aplicadas no experimento.

Para o experimento, foram considerados inicialmente uma média 40 CTs (casos de teste)

executados em cinco baterias de teste. Na prática, o número de CTs variou entre as baterias, pois alguns CTs foram incluídos e outros, à medida que se tornaram obsoletos, foram excluídos de uma bateria para outra.

Para medir a eficiência das técnicas nas cinco baterias, o conjunto original de testes foi inteiramente executado seguindo a técnica *Retest All* para que seus resultados fossem usados como referência. O alto custo da criação desses resultados de referência limitou o tamanho inicial a 40 CTs.

Para coletar dados do experimento, um protótipo de software foi criado para:

1. Calcular o valor histórico a partir do custo dos CTs, do total de severidade dos defeitos e dos dados históricos armazenados;
2. Armazenar os dados históricos em um repositório de dados;
3. Selecionar os CTs com maior valor histórico, com base em um critério de corte definido pelos engenheiros de software;
4. Armazenar as informações obtidas a partir da aplicação das técnicas Reteste de Casos de Uso de Maior Risco e Reteste por Perfil Operacional usadas neste artigo;
5. Calcular a confiabilidade do programa após a execução de cada bateria de teste, para cada técnica abordada neste artigo.

A partir da execução dos CTs os custos derivados a partir dos tempos de execução de cada CT e as severidades das falhas de defeitos derivadas a partir das criticidades dos defeitos detectados pelos CTs foram armazenados em um repositório de dados históricos. Essas informações foram então usadas para o cálculo do valor histórico, calibrando a técnica *Retest using Historical Data* a cada bateria de teste.

Finalmente, os subconjuntos de teste selecionados para cada técnica puderam ser comparados em termos de capacidade de detecção de defeitos e em termos da confiabilidade alcançada para o software. Alguns resultados dessa comparação são apresentados a seguir.

3. Resultados

Entendemos como eficiência de um conjunto de testes T' selecionado a partir da aplicação de uma técnica STR sobre o conjunto de testes original T, a capacidade de detectar o maior número de defeitos com o menor número de CTs possível. A partir disso, as técnicas Reteste de Casos de Uso de Maior Risco (RUcR) e Reteste por Perfil Operacional (RpO) foram comparadas com a técnica Retest Using Historical Data (RUhD), em termos do total de defeitos detectados, conforme ilustra a Tabela 1, após as baterias de teste. A diferença entre os totais de defeitos detectados ao longo das baterias mostra que a técnica Retest Using Historical Data apresentou os melhores resultados. Por meio dessa técnica, um número maior de defeitos foi detectado, com o mesmo número de CTs.

Tabela 1. Defeitos por criticidade

Técnicas	*Def. CRI*	*Def. ALT*	*Def. MED*	*Def. BAI*
RUcR	9	8	5	14
RpO	9	8	8	14
RUhD	14	12	22	24

A partir dos dados das Tabelas 2 e 3, pode-se concluir que a capacidade de detecção de defeitos da técnica Retest Using Historical Data foi, para esse experimento, praticamente igual à da técnica de referência Retest All considerando cada uma das baterias de teste. Isto pode ser considerado uma grande evidência de sua eficiência.

Tabela 2. Aplicação da técnica Retest All

Bateria	*CT selecionados*	*Total CT*	*Def. Detectados*
0	35	35	68
1	39	39	35
2	41	41	22
3	43	43	14
4	44	33	4

Tabela 3. Aplicação da técnica RUhD

Bateria	*CT selecionados*	*Total CT*	*Def. Detectados*
0	-	35	68
1	19	39	34
2	12	41	21
3	8	43	13
4	5	33	4

Além disso, as técnicas foram comparadas também em termos da confiabilidade de software alcançada após cada bateria de teste. A confiabilidade do software do SRFB foi medida após cada bateria de teste, sendo que, na bateria 0, as três técnicas tiveram o mesmo percentual de confiabilidade, para uma comparação justa entre as técnicas.

Tabela 4. Percentuais de confiabilidade

Técnicas	*Bt. 0*	*Bt. 1*	*Bt.2*	*Bt.3*	*Bt.4*
RUcR	31,54	52,62	75,07	90,35	90,77
RpO	31,54	52,87	75,32	90,58	90,77
RUhD	31,54	56,46	78.34	90,58	91,09

4. Conclusões

Este trabalho apresenta uma abordagem em que a técnica *Retest Using Historical Data* originalmente proposta para a priorização de casos de teste de regressão foi usada para a seleção de casos de teste de regressão. Por meio de um experimento prático, executando testes em um software do mundo real, a eficiência e a confiabilidade dessa abordagem foi avaliada, por meio de uma comparação com outras duas técnicas STR caixa preta. Como resultado do experimento, a técnica *Retest Using Historical Data* mostrou-se mais eficiente na detecção de defeitos e produziu uma melhora maior na confiabilidade do software do SRFB, quando comparada com as técnicas Reteste de Casos de Uso de Maior Risco e Reteste por Perfil Operacional.

Alguns trabalhos futuros estão previstos para aperfeiçoar a abordagem proposta neste artigo: i) novos experimentos com mais dados devem ser realizados, visto que os 40 CTs usados no experimento atual constituem um número relativamente baixo; ii) a técnica Retest Using Historical Data poderá ser comparada com um número maior de técnicas STR; iii) pretende-se elaborar a especificação de uma ferramenta para registro de CTs e de defeitos encontrados que facilite o armazenamento de dados históricos e calcule automaticamente os valores históricos das execuções de teste para uso posterior nos testes de regressão.

Referências

[1] S. Elbaum, A. Malishevsky, and G. Rothermel. Test case prioritization: A family of empirical studies. IEEE Transactions on Software Engineering, 28(2):159–182, February 2002.

[2] G. Rothermel and M. J. Harrold. Analyzing regression test selection techniques. IEEE Transactions on Software Engineering, 22(8):529– 551, August 1996.

[3] G. Rothermel and M. J. Harrold. A safe, efficient regression test selection technique. ACM Transactions on Software Engineering. Methodology, 6(2):173–210, April 1997.

[4] J.-M. Kim and A. Porter. A history-based test prioritization technique for regression testing in resource constrained environments. In ICSE ’02: Proceedings of the 24th International Conference on Software Engineering, pages 119–129, New York, NY, USA, 2002. ACM.

[5] H. Park, H. Ryu, and J. Baik. Historical value-based approach for cost-cognizant test case prioritization to improve the effectiveness of regression testing. SSIRI, 0:39–46, 2008.

[6] R. V. Binder. Testing Object-Oriented Systems: Models, Patterns and Tools. Addison-Wesley Longman, 2000.

[7] J. D. Musa. Software-reliability-engineered testing practice (tutorial). In ICSE ’97: Proceedings of the 19th International Conference on Software Engineering, pages 628–629, New York, NY, USA, 1997. ACM.

[8] T. L. Graves, M. J. Harrold, J.-M. Kim, A. Porter and G. Rothermel. An empirical study of regression test selection techniques. ACM Transactions on Software Engineering Methodology, pages 184–208, New York, NY, USA, 2001. ACM.

[9] W. E. Wong, J.R. Horgan, S. London and H. A. Bellcore. A Study of Effective Regression Testing in Practice. In ICSE ’97: ISSRE '97: Proceedings of the Eighth International Symposium on Software Reliability Engineering, pages 264-273, Washington, DC, USA, 1997. IEEE Computer Society.

Ferramentas de Projeto baseado em Plataforma UML de Sistemas de Gerência de Configuração Multi-Placas em Chassi

Rodrigo de A. Moreira, Alice M. Tokarnia (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

ramoreira@gmail.com, tokarnia@dca.fee.unicamp.br

Abstract – The design and implementation of embedded software for chassi management is a hidden part of the development of many communication systems. This embedded software is usually developed from scratch for each system using ad-hoc techniques. In this paper, we present a design toolbox for chassi management software that performs several steps of development automatically. Our toolbox includes a set of UML configurable software components; a simplified hardware modeling tool; a software synthesis package that constructs a module from functional components and communication proxies; and a simple estimation tool. A small management system is used to illustrate the use of our toolbox.

Keywords – Embedded system, Chassi management, UML platform, Software synthesis.

1. Introdução

Estruturas de múltiplas placas, agrupadas em chassi, são comumente encontradas em sistemas complexos, que incluem funções implementadas em hardware e software. Como exemplo, considere os sistemas de telecomunicações, que vêm se tornando mais complexos e podem incluir muitos dispositivos [1]. Atualmente, estes dispositivos requerem projeto de hardware e software embarcado para executar suas funções.

Um chassi é composto por placas, montadas em trilhas e conectadas através de um *backplane*. As placas são classificadas em placas de linha e placas de controle, central ou periféricas [2]. As placas de linha executam funções específicas do sistema, enquanto as placas de controle se encarregam de funções de gerenciamento do chassi [2], tais como: i) monitoramento de temperatura, ventilação e potência; ii) realização de inventário do sistema; iii) detecção de falhas e iv) atualização de softwares. Estas funções visam garantir que o sistema satisfaça as especificações de desempenho, temperatura, ventilação e confiabilidade, que são independentes da função específica do sistema e, na verdade, encontradas em várias estruturas multi-placas em chassi. Apesar disto, para cada chassi, geralmente um novo projeto é realizado sem reaproveitar partes de projetos anteriores e o desenvolvimento é baseado em métodos informais de projeto e técnicas manuais de codificação.

O objetivo deste trabalho é tornar mais rápido o projeto de sistemas de gerenciamento de configuração multi-placas em chassi, através de uma metodologia implementada por um conjunto de ferramentas de projeto. Estas ferramentas, reunidas no CMS-SEUP (*Chassi Management Software Synthesis and Estimation with UML Platforms*), dispõe dos seguintes recursos: 1) Biblioteca de componentes configuráveis de software, descritos em UML (*Unified Modeling Language*) [6]; 2) Ferramenta para geração automática de software, com inclusão do software de comunicação entre processadores, localizados na mesma placa ou em placas distintas e 3) Modelo simplificado de hardware que reúne informações para configurar componentes de software e calcular estimadores simples de desempenho, temperatura, potência e tamanho de memória.

Os conceitos e parte das ferramentas empregados no ambiente CMS-SEUP estão disponíveis em outras publicações. Ke *et. al.* descrevem um método de projeto baseado em componentes [3]. Marcio *et. al.* apresentam um método e uma ferramenta para explorar o espaço de projeto de software embarcado usando uma plataforma de software descrita em UML [4], mas não apresenta nenhum método para automatização do projeto de software. Shourong *et. al.* apresenta um método para síntese de componentes de software a partir de modelos UML [5]. Nosso trabalho se diferencia dos anteriores por reunir, num mesmo ambiente, ferramentas de projeto de software baseado em plataforma para gerenciamento de estruturas multi-placas em chassi.

O restante deste artigo está organizado da seguinte forma. A seção 2 apresenta a metodologia de projeto usada no CMS-SEUP. Um exemplo de projeto de software de gerenciamento é descrito na seção 3. A seção 4 traz a conclusão e os trabalhos futuros.

2. Proposta

O software embarcado de gerenciamento de configuração multi-placas em chassi é comumente armazenado em memórias *flash* e executado pelos processadores de uma plataforma de hardware. As ferramentas introduzidas neste trabalho permitem realizar automaticamente várias etapas do projeto de software embarcado, incluindo estimadores simples para alguns requisitos não-funcionais.

As ferramentas do CMS-SEUP auxiliam no projeto de sistemas de gerenciamento de configuração multi-placas em chassi desde a captura da especificação do sistema até a geração do código fonte e estimativa de parâmetros não-funcionais, conforme ilustrado na Figura 1. As ferramentas utilizam as quatro entradas a seguir: i) descrição das funcionalidades do sistema ii) descrição simplificada da plataforma de hardware de cada placa; iii) fatores de redundância e iv) biblioteca de componentes de software [6].

A especificação do sistema é realizada pelo projetista através das seguintes etapas:

1. Escolha das funcionalidades do sistema e especificação de restrições de desempenho, consumo de energia, tamanho de memória e temperatura.
2. Especificação simplificada da plataforma de hardware para configuração dos componentes de software e cálculo dos estimadores.
3. Especificação de *fatores de redundância*, que permitem ao projetista descrever características de operação em caso de falha e de manutenção. Estes fatores influenciam a seleção de componentes e o número de placas de controle do projeto. A Tabela 1 apresenta alguns fatores de redundância e suas implicações no projeto.

Após a captura da especificação, as próximas etapas, demarcadas pelo retângulo da Figura 1 são realizadas automaticamente. Em primeiro lugar, é feita a seleção dos componentes de software de uma biblioteca de acordo com as funcionalidades especificadas. Os componentes são customizados, suprimindo as funções não utilizadas. Em seguida, é realizado o mapeamento dos componentes de software nos componentes de hardware, levando em conta os *fatores de redundância*. O mapeamento define a configuração dos componentes e a necessidade de inclusão de *proxies* de comunicação.

Neste ponto, são calculados estimadores simples de desempenho, tamanho de memória e temperatura, apenas para uma primeira avaliação da especificação. O estimador de desempenho

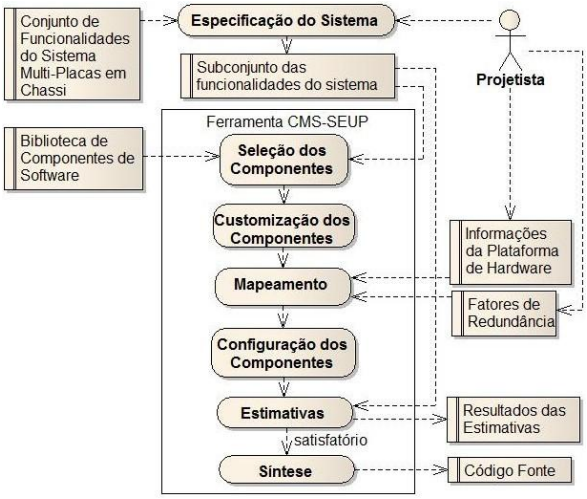


Figura 1 - Fluxo do Projeto CMS-SEUP.

Tabela 1 - Alguns fatores de redundância.

Fator 1: Redundância do placa central
Implicações: Replicação da placa de controle central e uso de componentes de sincronização entre as placas de controle central e de detecção de falha de placa.
Fator 2: Substituição de placas de linha sem a parar o sistema
Implicações: Componentes para armazenar as configurações das placas de linha e realizar a sincronização entre a placa central e placa de linha.
Fator 3: Redundância do sistema de ventilação
Implicações: Placas separadas para ventilação de subconjuntos de placas e componentes individuais de controle de ventilação.

leva em conta o algoritmo de escalonamento, o maior tempo de execução das tarefas (WCET-*worst case execution time*) nos processadores para as quais foram mapeadas e os períodos de cada tarefa. O estimador de temperatura leva em conta informações do fabricante e da plataforma de hardware para determinar se há necessidade de dissipador e/ou de ventilador para manter a temperatura especificada para o chassi. O estimador de tamanho de memória soma os tamanhos dos arquivos de programa e verifica se a memória não-volátil especificada é suficiente. É feita também a soma dos tamanhos dos dados e calculada a fração que pode ser armazenada na memória volátil. Como os dados podem não ser usados simultaneamente, cabe ao projetista avaliar se é suficiente.

A etapa final consiste na síntese de códigos fonte e de arquivos de comandos para geração do código. É possível gerar códigos executáveis, se os compiladores estiverem disponíveis.

3. Exemplo

Descrevemos a seguir as etapas do projeto de um sistema de gerenciamento de chassi e os

resultados obtidos pelas ferramentas CMS-SEUP. As funcionalidades deste sistema de gerenciamento estão descritas na Tabela 2. Neste caso, o projetista selecionou sete funcionalidades e especificou seis restrições de tempo, que são os períodos para execução das tarefas.

As informações da plataforma de hardware, apresentadas na Tabela 3, são usadas para estimativa de desempenho e temperatura e na síntese do software de comunicação. A descrição de memória é usada para verificar se o código sintetizado pode ser armazenado. Neste exemplo não foram especificados fatores de redundância.

3.1. Seleção dos Componentes

A ferramenta CMS-SEUP faz uma busca na biblioteca pelos componentes de software necessários para satisfazer as funcionalidades da Tabela 2. Esta biblioteca, apresentada em [6], inclui componentes de software específicos para projeto do sistema de gerenciamento de configuração multi-placas em chassi. Os componentes eventos selecionados para este projeto são mostrados na Tabela 4: Interface de Linha de Comando (*CLI*), Diagnóstico, Controlador Central e Gerenciador de Eventos.

3.2. Customização dos Componentes

Esta etapa suprime os componentes passivos do componente evento Diagnóstico, que não são usados para implementar as funcionalidades da especificação, descrita pela Tabela 2.

Tabela 2 – Subconjunto de funcionalidades.

Funcionalidades	Restrições
	Período
Monitorar Temperatura	7
Monitorar *Slots*	7
Controlar *Slots*	12
Monitorar Ventiladores	7
Controlar Ventiladores	12
Enviar Eventos	20
Interface de Linha de Comando	-

Tabela 3 - Informações da plataforma de hardware.

Processador	
• Nome	P1
• Freqüência	500 MHz
• Potência dissipada	10 W
• Temperatura do Chip	130 °C
Comunicação (Com.) entre as placas	
• Ethernet	100 Mpbs
Com. entre processador e sensores/atuadores	
• Serial RS232	19900 bauds
Memória Flash / Memória RAM	
• Tamanho	64 KB / 32 KB
Chassi	
• Número de Slots	7
• Temperatura Máxima	60 °C
• Dimensão das placas (mm)	233x250x22

Tabela 4 - Especificação do Sistema e Componentes.

Funcionalidade	Algumas Funções no Projeto de Software	Componentes
Monitorar *Slots*	GET_SLOT_STATUS GET_SLOT_STATE	Diagnóstico e Ger. de Eventos.
Controlar *Slots*	SET_SLOT_ENABLE SET_SLOT_DISABLE	Cont. Central e Ger. de Eventos.
Monitorar Temperatura	GET_TEMPERATURE	Diagnóstico e Ger. de Eventos.
Monitorar e Controlar Ventiladores	GET_FAN_RPM GET_FAN_STATUS SET_FAN_ROTATION	Cont. Central e Ger. de Eventos.

3.3. Mapeamento

Como não há redundância, a ferramenta utiliza uma implementação *básica* com duas placas de controle, uma central e outra periférica. Na placa de controle central são mapeados os componentes evento: *CLI*, Diagnóstico, Controlador Central e Gerenciador de Eventos. Na placa de controle periférica, que contém os ventiladores, é mapeada a função passiva Monitor de Velocidade dos Ventiladores.

3.4. Configuração dos Componentes

Com base nas informações de hardware, a ferramenta configura os componentes conforme descrito na Tabela 5. De acordo com o modelo de software básico [6], a comunicação entre um componente passivo e o componente evento que o contém é realizada através de chamadas com bloqueio. Por outro lado, a comunicação entre os componentes eventos é realizada sem bloqueio através de um *Proxy*. Um *Proxy* utiliza chamadas para rotinas *IPC* (*Inter Process Communication*), responsáveis pela troca de dados entre processos [8]. Desta maneira, como os componentes eventos são implementados como processos, é necessário configurar o tipo de *IPC* do *Proxy* para efetuar a comunicação entre os componentes *CLI* e Diagnóstico. Como eles estão mapeados no mesmo processador, a ferramenta configurou o *IPC* como uma *fifo* (*named pipe*). O *IPC* do *Proxy* entre os componentes Diagnóstico da placa central e o Diagnóstico da placa periférica foi configurado como um *Socket-Udp*, pois estão mapeados em processadores diferentes e possuem uma rede ethernet de comunicação.

Tabela 5 - Configuração dos componentes.

Componentes		
Evento	**Passivo**	**Configuração**
CLI	-	Comandos da interface
Diagnóstico	Monitor de Temperatura	Temperatura máxima
	Monitor de Slot	Quantidade de slots do chassi
	Monitor de Velocidade dos Ventiladores	Quantidade de placas de ventiladores

3.5. Estimativas

Neste exemplo, o estimador de desempenho considera um modelo de processo simples, onde todos os processos têm seus *prazos* (D) iguais aos períodos (T) especificados. As prioridades são atribuídas segundo o critério de taxa monotônica e é realizada uma análise de tempo de resposta para escalonamento baseado em prioridades (P) com preempção descrita em [7]. A ferramenta leva em conta o período de cada tarefa e o tempo de computação (C) dos componentes anotado na Tabela 2.

O pior caso para os tempos de resposta dos componentes (W), calculado usando a técnica apresentada em [7], é mostrado na Tabela 6. Neste caso, todos os componentes satisfazem os tempos de os prazos de execução.

O estimador de temperatura utiliza fórmulas fornecidas pelo fabricante e informações da plataforma de hardware para determinar o dissipador e os ventiladores necessários para manter a temperatura especificada para o chassi.

Para estimar o tamanho de memória, a ferramenta leva em conta as informações da plataforma de hardware na Tabela 3 e o tamanho anotado de cada componente utilizado. Estas informações são mostradas na Tabela 7.

3.6. Síntese

A etapa de síntese gera automaticamente o software executável de cada componente para cada processador. Para isto, a ferramenta utiliza: i) o código fonte dos componentes que estão armazenados na biblioteca; ii) as informações da plataforma de hardware e iii) o endereço dos *cross-compiladores* de cada processador [9].

4. Resultados

A Figura 2 apresenta o projeto de software em UML gerado automaticamente pelas ferramentas do CMS-SEUP seguindo as etapas descritas na seção 2.

Tabela 6 - Componentes Eventos Utilizados (processos).

Componentes Eventos	T	C	P	W
Diagnóstico	7	3	3	3
Controlador Central	12	3	2	9
Gerenciador de Eventos	20	5	1	20

Tabela 7 - Estimativa de tamanho de memória.

Componentes	Tamanho	Restrição	Res.
Diagnóstico	450 (KB)	64000(KB)	OK
Ger. de Eventos	300 (KB)		
Cont. Central	300 (KB)		
Total	1050 (KB)	64000 (KB)	OK

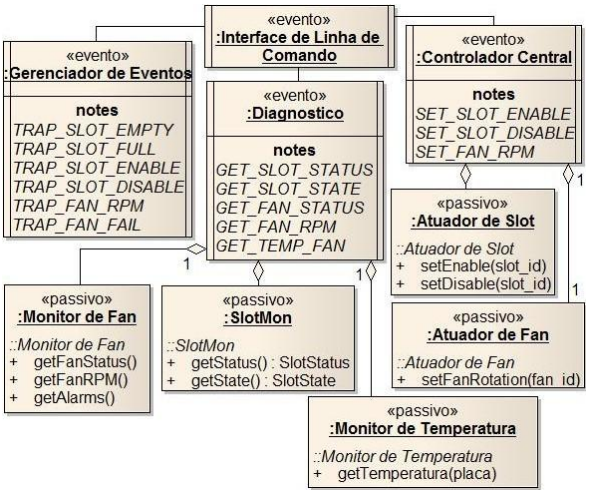


Figura 2 - Exemplo de Projeto de Software.

5. Conclusões

As ferramentas do CMS-SEUP tornam mais simples o reaproveitamento de partes de projetos anteriores; utilizam métodos formais de projeto e técnicas de síntese automática de código; permitem uma primeira verificação dos requisitos não-funcionais e reduzem o tempo de codificação.

Trabalhos futuros considerados são o aperfeiçoamento dos estimadores e a introdução de novos fatores de redundância e de uma etapa com exploração automática com novas opções de mapeamento e seleção de componentes.

Referências

[1] T. Sridhar, “Designing Embedded Communication Software”. Elsevier: CMP Books, 2003.

[2] “PICMG” http://www.picmg.org, Oct., 2009.

[3] X. Ke, K. Sierszecki, C. Angelov, “COMDES-II: A Component-Based Framework for Generative Development of Distributed Real-Time Control Systems”, Proc. of Embedded and Real-Time Computing Systems and Applications, 199-208, 2007.

[4] M. Oliveira, L. Brisolara, “Early Embedded Software Design Space Exploration Using UML-based Estimation”, Proc. of the 17th IEEE Intern. Workshop on Rapid System Prototyping, 2006.

[5] L. Shourong, W. Halang, L. Zhang, “A component-based UML profile to model embedded real-time systems designed by the MDA approach”, Proc. of Embedded and Real-Time Computing Systems and Applications, 563-566, 2005.

[6] R. Moreira, A. M. Tokarnia, “Ferramentas de Projeto baseado em Plataforma UML de Sistemas de Gerência de Configuração Multi-Placas”, Plano de Dissertação de Mestrado, FEEC, Unicamp, 11/2009.

[7] A. Burns, A. Wellings, “Real-Time Systems and Programming Languages”, Addison Wesley, 1997.

[8] “Inter-Process-Communication”, http://en.wikipedia.org/wiki/, Feb., 2010.

[9] “Linux Target Image Builder”, http://ltib.org. Feb., 2010.

Descrição de Padrões de Cenário de Operação de Sistemas Embarcados e Aplicação no Desenvolvimento de Monitores

Alice M. Tokarnia (Orientadora), Emerson Cruz

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

{torkarnia, ecruz}@dca.fee.unicamp.br

Abstract – Specification, performance analysis, and testing are key steps in the design of an embedded system. In many designs, these steps are independently executed and rely solely on the skills of the designers. The observation that same scenarios that specify system behavior are used in all these steps has led to software design methodologies that reduce design time. In this paper, the concept of scenario pattern is extended with a formal description that captures both functional and temporal behavior. An example of a scenario for a fire alarm system, which follows a pattern commonly found in several application domains, is presented. The objective of this formal description of scenario pattern is the development a configurable monitor that verifies system behavior using traces generated during tests or in the field.

Keywords – System specification, Scenario patterns, Trace monitor, Embedded systems.

1. Introdução

A especificação de sistemas embarcados geralmente inclui características funcionais e não-funcionais, que precisam ser repetidamente verificadas em várias etapas do desenvolvimento. Para reduzir o tempo de projeto, existem várias propostas de ferramentas que permitem reunir informações da especificação e utilizá-las na geração automática de software de verificação. A proposta deste trabalho é estender os trabalhos apresentados em [1], [2], [3] e [4], introduzindo uma notação para descrever uma variedade maior de padrões de cenários de operação e desenvolver um monitor configurável capaz de verificar os cenários de operação de um sistema.

Cada cenário de operação descreve uma interação entre o sistema e seu ambiente [1]. Como muitas interações podem ser descritas usando estruturas e elementos comuns, é possível associar muitos cenários a um mesmo padrão de cenário [2]. Associando a cada padrão de cenário de operação um padrão de software de teste, é possível obter a uma redução significativa no tempo de desenvolvimento de testes [3]. Conforme descrito em [3], apenas oito padrões de cenário foram suficientes para descrever aproximadamente 95% cenários de operação de um dispositivo médico implantável.

Muitos trabalhos fazem uso de padrões para descrever a especificação de requisitos de sistemas embarcados. Em [2], os autores utilizam uma linguagem natural estruturada para descrever cenários de operação e expressam requisitos de tempo real usando lógica temporal. Em [5], os autores introduzem padrões de operação baseados em linguagem natural. O objetivo é expressar os requisitos de um sistema usando construções padrões precisas e isentas de ambigüidades.

Este trabalho pode ser dividido em duas partes. A primeira parte consiste na elaboração de uma notação que permita descrever uma variedade de cenários de operação, incluindo requisitos funcionais e de tempo real. Na segunda parte, o objetivo é apresentar o núcleo de um monitor configurável a ser desenvolvido para verificar se os cenários especificados para um sistema são atendidos num rastro temporizado.

Este artigo está organizado da forma descrita a seguir. A seção 2 apresenta o modelo de descrição formal para o cenário de operação. A seção 3 fornece um exemplo de cenário de operação empregado em um sistema de detecção de incêndio. A seção 4 apresenta as informações do rastro e o algoritmo do monitor. A seção 5 traz as conclusões e os próximos passos deste trabalho.

2. Cenário de Operação

Para introduzir uma notação que permita a especificação de uma variedade de cenários de operação, foram utilizados conceitos em máquinas de estados programáveis (PSM) [6] e lógica temporal [7]. Os requisitos de tempo,

acrescentados a este modelo, são descritos por uma lógica temporal [7].

2.1 Elementos do cenário de operação

Os seguintes elementos são usados na descrição de um cenário de operação:

1. *Variáveis:* Correspondem às entradas, saídas, e estado. Uma variável V_i é descrita por <*função, tipo, domínio*>, onde a *função* pode ser entrada, saída, entrada/saída, estado; o *tipo* pode ser Booleano, inteiro, caractere, texto ou ponto flutuante e o *domínio* apresenta os valores possíveis na forma de conjunto ou intervalo. Todas as variáveis de estado podem ser lidas e que algumas podem também ser diretamente modificadas. O valor de V_i no tempo T_m é representado por $V_i(T_m)$. O tempo no qual a assume o valor *Val* é *Tempo*[V_i, *Val*]. Exemplos:
EstadoCentral : <estado, inteiro, {0, 1}>
EstadoCentral (20) = 0.

2. *Condição:* Descrita por expressões lógico-aritméticas usando variáveis, uma condição pode ser verdadeira (**V**) ou falsa (**F**).
Uma condição C_i é descrita por <[*elemento* 1] *relação* [*elemento* 2] >, onde [*elemento* i] é uma expressão usando variáveis e *relação* é um operador lógico-aritmético. O tempo no qual C_i se torna verdadeira é indicado por *Tempo*[C_i, **V**]. Exemplo:
$C_{sensor1}$: = < [SsrFumaça==1] || [SsrTérmico==1]>;

3. *Estado:* Um estado St_i é definido por um conjunto de condições sobre as variáveis de estado [St-V_1, St-V_2,⋯ St-V_n]. O estado determina relações entrada-saída e respostas a eventos. O estado do sistema no tempo T_m é dado por ST_SYS(T_m)= [St-V_1(T_m), St-V_2(T_m),⋯, St-V_n(T_m)]. representa o estado do sistema no tempo T_m. Exemplos: stMonitor := <EstadoCentral == 0>;
stAlarme := < EstadoCentral == 1 >.

4. *Evento:* Um evento E_i está associado a uma condição C-E_i sobre as variáveis de entrada e o tempo. Diz-se que o E_i ocorreu no instante *Tempo*[C-E_i, **V**] no qual a condição C-E_i se torna verdadeira. Um evento pode provocar uma transição de estado e/ou uma ação em um tempo posterior a *Tempo*[C-E_i, **V**]. Transição de estado e ação são definidas a seguir.
Exemplo: C-E_{sensor} := $C_{sensor1}$.

5. *Transição:* Mudança de estado ocasionada por um evento. Uma transição Tr_i é descrita por < C-E_i, St_a, St_b>, onde C-E_i representa a condição para que ocorra a transição St_a e St_b, são os estados inicial e final, respectivamente. A transição Tr_i ocorre atomicamente em *Tempo*[Tr_i] que pode ser posterior a *Tempo* [C-E_{if}, **V**]. Exemplo:
$Tr_{central}$:= < C-E_{sensor}, stMonitor, stAlarme >.

6. *Ação:* Atribuição de valores a um conjunto de variáveis de saída ocasionada por um evento. Uma ação A_i associada ao evento E_A é descrita por <C-E_A; [OV_1= Valor$_1$],⋯, [OV_k=Valor$_k$]> onde C-$E_{A\ i}$ é a condição, OV_i e $Valor_i$ são variáveis de saída e seus novos valores, respectivamente. A ação A_i ocorre atomicamente no *Tempo*[A_i], posterior a *Tempo*[C-E_A]. Exemplo: $A_{central}$:= < C-E_{sensor}, [Sirene = 1] >.

7. *Atividade:* Descreve a relação entrada-saída correspondente a um estado e a uma condição. A resposta do sistema é caracterizada por atribuições de uma seqüência de valores a variáveis de saída ao longo de um intervalo de tempo. Uma atividade At_i é descrita por <St_i; C_i; [OV= Val_1,Val_2,⋯,Val_n]; T >, onde os valores Val_1,Val_2,⋯,Val_n são atribuidos à variável de saída OV durante um intervalo de tempo T, quando o sistema se encontra no estado St_i e a condição C_i é satisfeita. Exemplo:
$At_{central}$ < stAlarme; $C_{sensor2}$; [Sprinkler=1]; 60s >.

8. *Tempo:* Conforme as definições anteriores o tempo é usado para caracterizar o comportamento do sistema. Na especificação de um cenário, apresentada a seguir, são estabelecidas condições sobre o tempo no qual as variáveis assumem valores. Todas as condições de tempo podem ser escritas na forma de lógica temporal. Exemplos:
Tempo [SsrFumaça, 1] – *Tempo* [SsrTérmico, 1] ≤ 2.

2.2 Cenários de operação

Um cenário de operação *CO* é descrito por dez elementos < *Id, Var, ST, Cond,* Stin, *Tc, Ac, Atvd, Tr, Tce, Te, Tend* >, divididos em quatro seções, conforme descrito a seguir.

Seção identificação:
Id é o número de identificação do cenário
Var é o conjunto de variáveis usadas na descrição do cenário;
ST é conjunto de estados usados na descrição do cenário;

Seção causa:
***St*$_{in}$** é o estado inicial do cenário;

Cond é um conjunto de condições sobre variáveis de entrada e de estado, que caracteriza o cenário de operação $\{C_1,..., C_n\}$. Pode incluir eventos.
$\boldsymbol{T_C}$ é um conjunto de restrições sobre intervalos de tempo entre os elementos da seção causa;
Seção efeito:
Ac é um conjunto de ações $\{ A_1, \cdots, A_n\}$;
Atvd é um conjunto de atividades $\{ At_1, \cdots, At_n\}$; com seus tempos máximo e mínimo;
Tr é uma transição para o estado final $\boldsymbol{St_f}$;

$\boldsymbol{T_{CE}}$ é um conjunto de restrições sobre intervalos de tempo definidos entre um elemento da seção efeito e outro da seção causa;
$\boldsymbol{T_E}$ é um conjunto de restrições sobre intervalos de tempo entre os elementos da seção efeito;

Seção restrição de tempo de término:
$\boldsymbol{T_{end}}$ é prazo máximo para término do cenário.

3. Exemplo de Cenário

No cenário de operação descrito a seguir, a condição para a transição de estados é a ocorrência de dois ou mais eventos, em qualquer order.

A descrição textual do cenário: "A central de incêndio deverá sair do modo monitoramento para o modo incêndio e acionar a saída de sirene se os sensores termo-velocimétrico e de fumaça, que monitoram o ambiente, enviarem eventos de emergência, em qualquer ordem, num intervalo de até dois segundos."

Descrição formal

Seção identificação:
Id = 1.

Var = {EstadoCentral, SensorFumaça, SensorTérmico, Sirene};
EstadoCentral: < estado, inteiro, {0, 1}>;
SensorFumaça: < entrada, inteiro, {0, 1}>;
SensorTérmico: <entrada, inteiro, {0, 1}>;
Sirene: <saída, inteiro, {0, 1}>;

ST = {stMonitor, stIncedio}
stMonitor = < EstadoCentral == 0>;
stIncedio = < EstadoCentral == 1 >

Seção causa:
***Cond*:**
cSensorTérmico = < SensorFumaça ==1>
cSensorFumaça = < SensorTérmico ==1>

$\boldsymbol{St_{in}}$**:**
stMonitor
$\boldsymbol{T_C}$**:**
***Tc*-1:** | Tempo [*stMonitor* AND *cSensorTérmico*, **V**] - Tempo [*stMonitor* AND *cSensorFumaça*, **V**] | $\leq$ 2s ;

Seção efeito:
***AC*:**
AC-1 = < ***Tc*-1**, [Sirene = 1] >;

***Tr*:**
Tr-1 = < ***Tc*-1**, *stMonitor, stIncedio*>;

Seção restrição de tempo de término:
$\boldsymbol{T_{end}}$**:**
T_{final} = 10s

Como a descrição textual não especifica até quanto tempo depois da condição ***Tc*-1** ser satisfeita devem ocorrer a transição de estado e ação de tocar a sirena, foi necessário arbitrar um prazo máximo para término do cenário.

4. Rastro e Monitor

O verificador possui dois arquivos de entrada um contendo o rastro de operação (*trace*) e o outro uma lista de cenários. O rastro de operação é o resultado de uma amostragem periódica das variáveis de entrada, saída e estado do sistema durante um intervalo de tempo. A linha K do rastro é constituída por $<T_K, V_1(T_K), V_2(T_K), \cdots, V_n(T_K)>$, onde T_K é o tempo correspondente a k amostragem e $V_i(T_K)$ é o valor da variável V_i no tempo T_K e n é o número de variáveis monitoradas. Os cenários da lista estão classificados em padrões. Os padrões diferem apenas no nome das variáveis e nos valores numéricos considerados nas condições. Para cada cenário padrão existe uma rotina de verificação específica.

O monitor apresenta como resultado um arquivo que lista para cada cenário as seguintes informações:

i) Identificação do cenário
ii) Descrição do cenário (copiada do arquivo de entrada).
iii) Número N_A de vezes que o cenário foi *acionado*, isto é, a seção causa do cenário foi satisfeita.

iv) Número N_S de vezes que o cenário foi *satisfeito*, isto é, tanto seção causa quanto a seção efeito do cenário foram satisfeitas.
v) Número N_I de vezes que não foi possível determinar se o cenário foi satisfeito ou não.
vi) Fração F_S de vezes que o cenário foi satisfeito, que é calculada como N_S /(N_A - N_I).

A Figura 1 apresenta o pseudocódigo geral para as rotinas de verificação de um cenário de um padrão e a Figura 2 apresenta o pseudocódigo para o monitor. Para cada cenário, o monitor faz uma chamada à rotina correspondente ao padrão deste cenário. A rotina percorre o trace para identificar quando as seções causa e efeito do cenário são satisfeitas. O monitor é configurado de forma a incluir apenas as rotinas correspondentes aos padrões de cenários utilizados na especificação do sistema.

```
Rotina de verificação dos cenários padrão L:

Para cada linha k no trace:
  Se o estado inicial do cenário for satisfeito:
    Para cada condição na seção causa do cenário:
        Se a condição for satisfeita, anotar o tempo
        T_k para esta condição.
        Se todas as condições estiverem satisfeitas
        usar os tempos anotados para verificar
        equações em T_C
       Se todas as equações em T_C forem
        satisfeitas:
        i)  Incrementar N_A
        ii) Verificar se a seção efeito do cenário
            é satisfeita até o tempo T_k + T_end.

            Se seção efeito for satisfeita,
              Incrementar N_S.
            Se seção efeito não for satisfeita,
               Se fim do trace antes de T_k + T_end
                 Incrementar N_I

 Calcular  F_S = N_S /( N_A - N_I).
```

Figura 1. Rotina de verificação

```
Monitor

Para cada cenário do arquivo de entrada:
 i)   Ler cenário,
 ii)  Ler padrão correspondente a este cenário
 iii) Determinar variáveis, estados e demais
      parâmetros.
 iv)  Chamar rotina de verificação do padrão,
      passando variáveis e parâmetros
      correspondentes.
```

Figura 2. Monitor

5. Conclusão

Este trabalho apresenta uma proposta de descrição formal para especificação de cenários de operação de sistemas embarcados, um exemplo de descrição de um cenário e uma descrição inicial para um monitor que utiliza padrões de cenários para analisar um rastro e determinar se cada cenário é satisfeito.

Os próximos passos neste trabalho são a determinação de um conjunto de padrões de cenários de operação, a codificação de rotinas de verificação e do monitor e a aplicação do monitor a sistemas reais.

Referências

[1] S. Some, R. Dssouli, J. Vaucher, "From Scenarios to Timed Automata: Building Specifications from Users Requirements," *Second Asia-Pacific Software Engineering Conf.*, p. 48, 1995.

[2] S. Konrad, B. Cheng, "Real-time specification patterns," *Proc. of the 27th International Conf. on Software Engineering*, p. 372, 2005.

[3] W. Tsai, L. Yu, F. Zhu, R. Paul, "Rapid Embedded System Testing Using Verification Patterns," *IEEE Software*, v. 22, p. 68, July/Aug. 2005.

[4] W. Tsai, L. Yu, R. Paul, X. Wei, "Rapid Pattern-Oriented Scenario-Based Testing for Embedded Systems," *Software Evolution with UML and XML*, 2004.

[5] C. Denger, D. M. Berry, E. Kamsties, "Higher Quality Requirements Specifications through Natural Language Patterns," *Proc. of the IEEE International Conference on Software-Science, Technology & Engineering*, p. 80, 2003.

[6] F. Vahid, T. Givargis, *Embedded System Design: A Unified Hardware/ Software Introduction*. Wiley, 2002.

[7] X. Chen, H. Hsieh, Y. Watanabe, F. Balarin, "Automatic trace analysis for logic of constraints," *Proc. of the 40th annual Design Automation Conference*, p. 460, 2003.

Processamento de Imagens

Análise de Algoritmos da Transformada *Watershed*

André Körbes , Roberto de Alencar Lotufo (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

{korbes,lotufo}@dca.fee.unicamp.br

Abstract – This paper presents an analysis of two issues of the watershed transform algorithms. The first issue regards the general behaviour of those, concerning the scanning order of pixels on the image, providing means for classification and generalization. The second issue is about the practical implications of the different definitions implemented by the algorithms in the literature. These are discussed on a real application of image segmentation.

Keywords – watershed transform, watershed algorithms

1. Introdução

A transformada *watershed* propõe uma abordagem morfológica para o problema de segmentação de imagens, interpretando estas como superfícies, onde cada *pixel* corresponde a uma posição e os níveis de cinza determinam as altitudes. A partir desta noção, deseja-se então identificar bacias hidrográficas, definidas por mínimos regionais e suas regiões de domínio. Intuitivamente, a transformada *watershed* trata de encontrar os pontos em uma superfície onde uma gota d'água possa escorrer para dois mínimos regionais diferentes. Esta analogia pode ser também criada inversamente, onde um nível d'água é elevado através de uma superfície, inundando a partir dos mínimos regionais, e, nos pontos onde águas provenientes de mínimos diferentes se tocarem ergue-se uma barreira, que constitui a linha de divisão das bacias.

Todavia, este trabalho se trata de uma análise dos algoritmos da transformada *watershed*, que implementam as diversas definições existentes. Para tal efeito, considerou-se a literatura a partir da introdução da primeira transformada rápida em 1991 por Vincent e Soille [10] até os trabalhos recentes de Cousty *et al.* [5], englobando 14 algoritmos. Assim, são identificadas características que permitem classificá-los e compara-se os resultados das definições destes em uma aplicação real.

Este trabalho é organizado conforme: a Sec. 2. apresenta as definições de transformada *watershed* de base para os algoritmos, a Sec. 3. propõe os métodos de exploração para classificação dos algoritmos, a Sec. 4. verifica a influências das definições em uma aplicação real, e a Sec. 5. apresenta as conclusões.

2. Definições de Watershed

Nesta seção introduzimos brevemente as principais definições de transformada *watershed*: inundação (Flooding-WT), distância topográfica (TD-WT), condição local (LC-WT), transformada imagem floresta com custo máximo de caminho (IFT-WT), zona de empate da IFT-WT (TZ-IFT-WT) e *watershed cut* (WC-WT). Estas definições são a base dos algoritmos disponíveis na literatura [6], e por consequência seus resultados afetam como as aplicações se comportam e o que pode ser esperado como saída destas.

A def. Flooding-WT consiste em um processo iterativo de limiarizações, onde as regiões crescem de acordo com suas zonas de influência calculadas em relação ao nível anterior. As linhas de *watershed* são os *pixels* que não pertencem a nenhuma região [10].

A def. TD-WT minimiza uma função de custo entre quaisquer pontos em uma imagem, baseada na soma das distâncias entre os pontos intermediários e relacionada às diferenças de níveis de cinza. Determina-se assim as regiões da imagem, provendo uma solução única onde um *pixel* com custo igual para dois ou mais mínimos regionais é marcado como *watershed* [9]. A def. LC-WT é derivada da def. TD-WT, onde a condição de unicidade da solução é removida. Desta forma, os *pixels* de *watershed* deixam de existir, sendo atribuídos a uma das regiões vizinhas [3].

A def. IFT-WT é uma abordagem baseada em floresta de caminhos mínimos para o problema, onde o custo do caminho é o máximo dos arcos. Todavia, em uma zona plana estes custos são iguais para todos os *pixels* nesta. Para tratar isto, é intro-

duzida uma segunda componente, o custo lexicográfico, que determina a menor distância até a borda da zona plana, desempatando a primeira [8]. A zona de empate da IFT-WT é uma transformada que unifica soluções. Dado que a IFT-WT produz um conjunto de soluções ótimas, a TZ-IFT-WT unifica-as em apenas uma solução, onde as regiões são definidas por aqueles *pixels* que tem um caminho ótimo para o mesmo conjunto de sementes em todas as soluções possíveis [1].

A def. WC-WT é uma transformada de corte de grafo, onde os cortes são feitos quando uma aresta satisfaz as condições de que os vértices que conecta levam a dois mínimos diferentes, e que seu valor é maior do que o valor de qualquer aresta nestes caminhos até os mínimos [5]. Entretanto, não é produzida uma solução única pois diversos cortes podem satisfazer estes critérios.

Uma consideração importante é que independentemente do algoritmo, seus resultados serão dependentes da definição implementada. Isto implica que algoritmos em uma mesma definição devem ser todos capazes de produzir o mesmo conjunto de soluções, ou um subconjunto deste.

3. Métodos de Exploração

A análise de exploração da imagem realizada neste trabalho busca caracterizar os algoritmos de acordo com a estratégia utilizada para a rotulação dos *pixels*. Entre as formas de exploração identificadas, há duas vertentes representativas e ainda outra alternativa pouco utilizada. Adotamos aqui a nomenclatura utilizada por Cormen *et al.* [4] e clássica na avaliação de algoritmos de busca em grafos: busca em largura e busca em profundidade. A terceira linha constitui-se da varredura aleatória, onde não é imposta nenhuma ordem no acesso aos *pixels*, pouco utilizada no âmbito da transformada por questões de desempenho. É importante ressaltar que esta caracterização não implica em qual definição o algoritmo implementa.

3.1. Busca em Largura

Métodos de busca em largura são bem conhecidos na literatura de computação, sendo os fundamentos de diversos procedimentos, como os algoritmos de busca em grafos de Dijsktra para construção de SPFs e de Prim para construção de MSFs [4]. A principal característica deste tipo de algoritmo é dada por sua natureza de expansão, sempre da última borda e uniformemente em sua largura. A respeito da distância da semente original, todos os vértices a distância k são visitados antes de visitar qualquer vértice a distância $k + 1$ [4].

No campo dos algoritmos de transformada *watershed*, pode ser vista uma similaridade entre a busca em largura e o *watershed* por marcadores, onde um conjunto de sementes é expandido para se encontrar a partição ótima formada por estas, ou mesmo considerando-se como sementes os mínimos regionais. Estas propostas se diferenciam em uma série de características e implementam definições distintas, mas preservam a varredura em largura, onde a distância k passa a ser o custo do caminho.

Desta forma, pode-se generalizar os algoritmos que aplicam esta estratégia em 3 passos, onde permite-se variar a forma de expansão e rotulação de acordo com a definição adotada, com a variação fundamental sendo as conexões iterativas no passo 2:

1. Defina as sementes/marcadores
2. Calcule as conexões dos *pixels* da iteração atual com a anterior
3. Rotule os *pixels* de acordo com as conexões calculadas. Vá para o passo 2 e expanda as regiões da iteração atual, até visitar todos os *pixels*

De maneira geral, algoritmos de transformada *watershed* com varredura em largura tem como desvantagem a detecção inicial dos mínimos regionais, seja por um procedimento independente, ou por ordenação dos *pixels* por seus valores, para detecção dos componentes conexos mínimos. No entanto, se estes marcadores constituírem parâmetros de entrada do algoritmo, o processamento da vizinhança torna-se uma operação localizada e de baixo custo computacional.

3.2. Busca em Profundidade

Assim como os algoritmos de busca em largura, a busca em profundidade também é muito comum na literatura de grafos, representada comumente por algoritmos gulosos. A estratégia adotada aqui é continuar a busca sempre a partir do vértice visitado mais recentemente, enquanto possível, e então retornar para analisar vértices pendentes. De fato, em

contraste com a busca em largura, a busca em profundidade prioriza vértices a distância $k+1$ assim que são descobertos, antes de visitar todos à distância k [4].

Pode-se facilmente ver a semelhança entre este procedimento descrito e uma gota d'água descendo sobre uma superfície, onde esta segue um caminho único. Em relação a transformada *watershed*, segue-se o mesmo raciocínio, representado pela técnica de *arrowing*, onde identifica-se para cada *pixel* o vizinho com menor valor. Diversos algoritmos utilizam esta abordagem, seguindo um caminho até que este seja esgotado em um mínimo regional, enquanto outros são baseados em construir os caminhos em etapas anteriores, identificar os mínimos, e por último executar a rotulação percorrendo o caminho, caracterizando-os como buscas em profundidade.

Assim como o algoritmo de transformada *watershed* por busca em largura, pode-se generalizar a busca em profundidade em 3 passos, conforme a caracterização mencionada anteriormente:

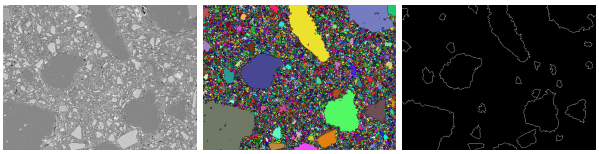
1. Conecte cada *pixel* ao(s) seu(s) vizinho(s) de acordo com o custo desejado
2. Rotule os mínimos regionais
3. Percorra para cada *pixel* o caminho até o mínimo regional e rotule-o

Estes passos são geralmente distinguíveis nos algoritmos, no entanto podem ser unidos, na tentativa de atingir melhor desempenho. Em especial deve-se comentar sobre o passo 1, que depende da resolução das zonas planas. Estas regiões não podem, de maneira geral, ser conectadas utilizando um algoritmo de busca em profundidade, optando-se por utilizar variações de algoritmos em largura para propagação e divisão correta, aplicando-se uma segunda componente de custo para resolução destes empates. Entretanto, mesmo nestes casos, o passo 3 é característico de algoritmos de busca em profundidade, percorrendo o caminho até encontrar um *pixel* já rotulado ou um mínimo regional.

4. Análise de Resultados

Nesta seção comparam-se os algoritmos de *watershed* por seus resultados em uma aplicação real. Desta forma, busca-se destacar a questão da definição implementada pelos algoritmos que pode impactar no resultado da aplicação. Considerando-se que cada um dos algoritmos está associado a uma definição da transformada [6], serão apresentados apenas os resultados para cada uma destas. A def. Flooding-WT, que não é implementada corretamente por nenhum algoritmo, é representada aqui pelo algoritmo de Imersão de Vincent e Soille [10].

A aplicação apresentada aqui é um verificador de imagens de concreto em microscópio, onde deseja-se medir algumas regiões de interesse (ROI's), sendo a transformada *watershed* usada como um detector de texturas para regiões homogêneas, filtradas com base em um critério de área. A Fig. 1 apresenta as etapas para extração das ROI's: (a) imagem de entrada; (b) transformada *watershed* rotulada aplicada sobre o gradiente filtrado; (c) contornos internos das regiões obtidas após filtragem por área para selecionar regiões e remover ruído.



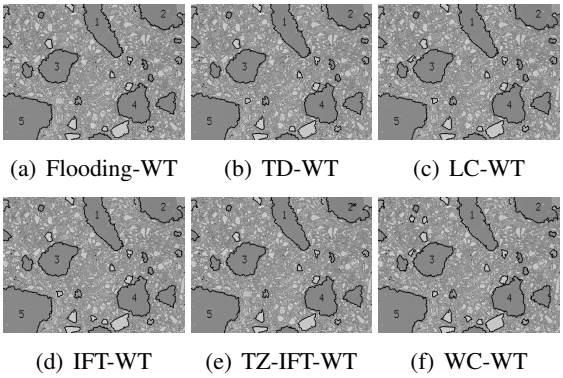
(a) (b) (c)

Figura 1. Etapas da aplicação *concrete*

Alternando a transformada *watershed* aplicada no passo (b) entre as seis definições, mede-se o número de regiões homogêneas detectadas, a área e variação para as cinco ROI's. A Fig. 2 apresenta as imagens produzidas pelos algoritmos, com o contorno interno destacado. A Fig. 2 (a)-(f) apresenta as cinco ROI's numeradas, e as soluções obtidas pelos algoritmos de Vincent e Soille [10], de Lin *et al.* [7], de Bieniek e Moga [3], de Beucher e Meyer [2, 8], de Audigier, Lotufo e Couprie [1] e de Cousty *et al.* [5], respectivamente.

Conforme mencionado, o objetivo desta aplicação é medir regiões homogêneas, sendo que também há interesse no número de regiões detectadas. Para as imagens (a) - (f), foram encontradas 26, 27, 28, 28, 23 e 30 regiões respectivamente. A respeito das cinco ROI's, a área média e a máxima variação positiva (MVP) e negativa (MVN) foram calculadas, sendo apresentadas na Tab. 1, onde cada linha corresponde a uma ROI.

De forma geral, a análise dos resultados aponta para consistência entre as soluções. Entretanto, em alguns problemas uma análise mais aprofundada pode ser necessária, de modo a determinar



(a) Flooding-WT (b) TD-WT (c) LC-WT

(d) IFT-WT (e) TZ-IFT-WT (f) WC-WT

Figura 2. Comparação de resultados da aplicação de watershed na identificação de regiões homogêneas

Região	Média	MVP	MVN
1	12749,66	2,3%	3,5%
2	11343,5	1,9%	2,8%
3	11030,16	1,8%	2,9%
4	11041,83	6,2%	4,0%
5	22776	0,3%	1,0%

Tabela 1. Medidas nas cinco ROI's da aplicação

a abordagem mais adequada, especialmente no que diz respeito ao comportamento da definição, sendo que os algoritmos são dependentes destas, e pode ocorrer variação. Esta conclusão deve ser considerada em aplicações com baixa tolerância a erros. Todavia, quando há pré-processamento com grande simplificação da imagem, as diferenças tendem a ser pequenas, variando abaixo de 1%.

5. Conclusões

Neste trabalho foram analisados dois aspectos da transformada *watershed*: o comportamento de visitação da imagem dos algoritmos, classificados em largura e profundidade; e a influência das diferentes definições em uma aplicação real. Propusemos duas generalizações para os algoritmos de *watershed* de modo a compreender melhor o funcionamento destes, e permitir associar mais a transformada a princípios da computação como otimização de custos em caminhos e menos a princípios naturais como a gota d'água. A aplicação real foi testada com seis definições de modo a se verificar que, embora aproximados, os resultados variam e isso deve ser estudado caso a caso.

Referências

[1] R. Audigier, R. Lotufo, and M. Couprie. The tie-zone watershed: Definition, algorithm and applications. In *Proceedings of IEEE International Conference on Image Processing (ICIP'05)*, volume 2, pages 654–657, 2005.

[2] S. Beucher and F. Meyer. *Mathematical morphology in image processing*, chapter The Morphological Approach to Segmentation: The Watershed Transformation. Optical Engineering. M. Dekker, New York, 1993.

[3] A. Bieniek and A. Moga. An efficient watershed algorithm based on connected components. *Pattern Recognition*, 33(6):907–916, 2000.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, Cambridge , Massachusetts, 2 edition, 2001.

[5] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1362–1374, 2009.

[6] A. Körbes and R. Lotufo. Analysis of the watershed algorithms based on the breadth-first and depth-first exploring methods. In *SIBGRAPI'09*, pages 133–140, Rio de Janeiro, Brazil, Oct. 2009. IEEE Computer Society.

[7] Y. Lin, Y. Tsai, Y. Hung, and Z. Shih. Comparison between immersion-based and toboggan-based watershed image segmentation. *IEEE Transactions on Image Processing*, 15(3):632–640, 2006.

[8] R. Lotufo and A. Falcão. The ordered queue and the optimality of the watershed approaches. In *Proceedings of the 5th International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing*, volume 18, pages 341–350. Kluwer Academic Publishers, June 2000.

[9] F. Meyer. Topographic distance and watershed lines. *Signal Processing*, 38(1):113–125, 1994.

[10] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

Segmentação de imagens de tensores de difusão no contexto da morfologia matemática

Leticia Rittner , Roberto A. Lotufo (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

{lrittner,lotufo}@dca.fee.unicamp.br

Abstract – The main goal of this work is to present a segmentation method for diffusion tensor images, based on the watershed transform. Instead of adapting the watershed to work with tensorial images, the tensorial morphological gradient (TMG), retaining relevant information from tensors, was defined. The desired segmentation is achieved by applying the watershed over the TMG. Segmentation results obtained by the hierarchical watershed over the TMG are comparable to atlas-based segmentation. The proposed method is used to segment the thalamic nuclei, an important task for neuroscience.

Keywords – segmentation, mathematical morphology, watershed, tensorial morphological gradient, diffusion tensor images.

1. Introdução

A imagem de difusão é uma nova modalidade de imagem por ressonância magnética, muito utilizada na área médica em estudos relacionados ao cérebro. Ela trabalha com a mensuração das tendências do movimento aleatório das moléculas de água em um dado meio. Normalmente estas moléculas se movem desordenadamente em altas velocidades em todas as direções, colidindo umas com as outras. Em regiões onde o tecido cerebral é altamente organizado, o movimento das moléculas fica restrito à direção paralela à orientação da estrutura do tecido. A este movimento com direção preferencial, dá-se o nome de difusão anisotrópica. Neste caso, como o movimento das moléculas de água não pode mais ser caracterizado por um único coeficiente de difusão, é necessário lançar mão de um modelo mais complexo, o tensor de difusão, que consegue descrever deslocamentos por unidade de tempo diferentes em cada direção.

A imagem de tensores de difusão (DTI) contém, portanto, um tensor para representar cada pixel e é capaz de revelar detalhes da anatomia da substância branca do cérebro e fornecer pistas para entender a conectividade do cérebro. Em patologias onde o estudo anatômico não é suficiente para explicar determinados comportamentos e sintomas, somente uma imagem que forneça informação das conexões preservadas ou comprometidas parece poder ajudar em seu diagnóstico e tratamento.

Nos últimos anos, um assunto bastante estudado na área de imagens de tensores de difusão é a segmentação destas imagens, etapa necessária para permitir a análise quantitativa de difusão em uma determinada estrutura do cérebro. Normalmente, o delineamento da estrutura a ser estudada é feito em uma imagem de ressonância convencional e depois transferido para imagens de tensores de difusão após registro destas imagens. A utilização de um método de segmentação baseado em DTI elimina a necessidade de registro das imagens, reduzindo o tempo de processamento e evitando eventuais erros introduzidos pelo registro. Outra razão para se buscar um método de segmentação baseado em DTI é a capacidade desta modalidade de imagem de distinguir regiões cerebrais, não identificadas por nenhuma outra modalidade de imagem, graças à informação direcional que ela contém.

Dentre os métodos de segmentação de imagens de tensores de difusão propostos na literatura, podemos citar os baseados em grafo [11, 9], *level-set* [10, 8, 4], evolução de superfície [5] e crescimento de regiões [6]. Assim, o objetivo principal deste trabalho é propor um método de segmentação para imagens de tensores de difusão baseado em conceitos da morfologia matemática e na transformada de *watershed*.

Este trabalho esta organizado da seguinte forma: os conceitos básicos relacionados estão descritos na Seção 2.; Seção 3. apresenta os resultados dos experimentos de segmentação utilizando-se o método proposto. Finalmente, conclusões foram resumidas na Seção 4..

2. Base conceitual

2.1. Imagens de tensores de difusão

O cérebro é composto por diferentes tipos de tecidos e que possuem níveis distintos de organização celular, resultando em movimentos de difusão com diferentes graus de anisotropia. Nestas regiões, a difusão não pode mais ser descrita por um único escalar, mas sim, por um tensor, que representa não só a mobilidade molecular ao longo de todas as direções, mas também a correlação entre elas [1].

Isso quer dizer que a cada voxel da imagem de tensores de difusão está associado um tensor de segunda ordem, que representa a difusão das moléculas de água no cérebro humano. Ele normalmente é escrito na forma de uma matriz 3×3:

$$\mathbf{T} = \begin{pmatrix} T_{xx} & T_{xy} & T_{xz} \\ T_{yx} & T_{yy} & T_{yz} \\ T_{zx} & T_{zy} & T_{zz} \end{pmatrix} \qquad (1)$$

Como, no caso da difusão, T é uma matriz SDP, os autovalores (λ_1, λ_2, λ_3) da matriz são reais e seus autovetores (e_1, e_2, e_3) são ortogonais. Nesse caso, T pode ser representado por um elipsóide, cuja orientação é definida pelos autovetores e cujos raios correspondem à raiz quadrada dos autovalores.

2.2. *Watershed* hierárquico

A transformada de *watershed* é uma ferramenta de morfologia matemática para segmentação de imagens [2]. Uma forma fácil de se entender a transformada de *watershed* por marcadores é compará-la a um processo de inundação, onde a imagem pode ser vista como uma superfície topográfica, cuja altitude corresponde ao nível de cinza. Quando buracos são abertos em alguns pontos marcados da imagem, água colorida começa a subir por estes buracos, sendo que cada cor está associada a um buraco (marcador). A medida que a inundação vai ocorrendo, barragens são construídas cada vez que águas de cores diferentes se encontram, de forma a mantê-las separadas. Estas barragens são as linhas do *watershed* e os lagos coloridos são as regiões resultantes da segmentação da imagem (Fig. 1).

Nesta trabalho, é usada uma variante do *watershed*, denominado *watershed* hierárquico, onde os marcadores são escolhidos usando-se o critério do valor de extinção dos mínimos regionais. Em outras palavras, após o cálculo dos mínimos regionais, são utilizados como marcadores para o *watershed* os "n" mínimos regionais com os maiores valores de extinção segundo o critério de volume [3]. Deste modo, estamos segmentando a imagem nas n regiões mais significativas.

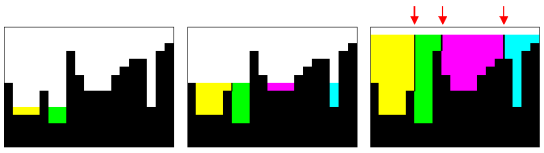


Figura 1. A transformada de *watershed* - analogia com um processo de inundação

2.3. Gradiente morfológico tensorial (TMG)

Para realizar a segmentação de uma imagem de tensores de difusão do cérebro utilizando-se a transformada de *watershed*, primeiro é necessário calcular o gradiente desta imagem. Ou seja, é necessário transformar a imagem tensorial em uma imagem escalar, que contenha de preferência, informação das transições existentes na imagem original. Propusemos, então, o gradiente morfológico tensorial (TMG), inspirado no conceito do gradiente morfológico adaptado para imagens tensoriais [7]:

$$\nabla_B^T(f)(x) = \bigvee_{y,z\in B_x} d_n(\mathbf{T}_y, \mathbf{T}_z), \qquad (2)$$

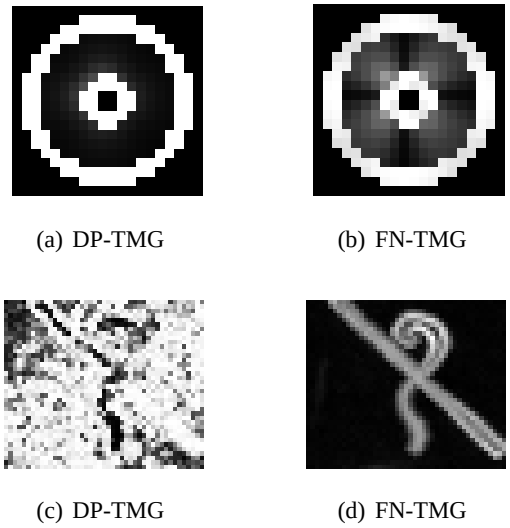
$\forall x \in E$, onde d_n representa uma medida intervoxel, $B \subset E$ é o elemento estruturante, $\mathbf{T}_y$ e $\mathbf{T}_z$ são tensores que representam a difusão em y e em z, respectivamente (y e z estão na vizinhança de x, definida por B_x). ∇_B^T é o TMG proposto. Enquanto o gradiente morfológico é dado pela dilatação menos a erosão, que no caso mais simples, é uma diferença entre a filtragem de ordem de máximo menos a de mínimo, na proposta do TMG, busca-se o máximo das diferenças ou dissimilaridades entre os voxels vizinhos.

3. Resultados

O método de segmentação proposto foi testado, tanto em imagens de difusão geradas sinteticamente, imagens de difusão adquiridas de *phantom*, quanto em imagens de tensores de difusão do cérebro.

3.1. Segmentação de DTI sintéticas

Os resultados obtidos pela utilização do *watershed* nas imagens de TMGs calculadas mostrou que o método proposto realmente tem a capacidade de segmentar imagens de tensores de difusão, desde que a medida intervoxel e o elemento estruturante sejam corretamente escolhidos, de acordo com as características das estruturas que se quer segmentar. Um exemplo da adequação da medida intervoxel à imagem a ser segmentada pode ser visto na Fig. 2. Enquanto que para o torus sintético, ambas medidas foram capazes de preservar a borda do torus no cálculo do TMG, no caso do phantom, o produto escalar não consegue preservar a informação necessária para a etapa de segmentação. Neste caso então, a norma de Frobenius seria a única a permitir a segmentação correta pela aplicação do *watershed* na imagem do TMG calculado.



(a) DP-TMG (b) FN-TMG

(c) DP-TMG (d) FN-TMG

Figura 2. TMG calculado para o torus sintético e para o phantom (DP e FN)

3.2. Segmentação de DTI do cérebro

Também foram realizados diversos experimentos com imagens do cérebro, tanto em estruturas de substância branca, quanto substância cinzenta do cérebro, mostrando que o método independe do tipo de tecido a ser segmentado. Para segmentar automaticamente o corpo caloso, por exemplo, primeiramente o gradiente morfológico tensorial (TMG) foi calculado usando a Norma de Frobenius e um elemento estruturante 6-conexo (3D). Uma vez calculado, o TMG foi usado pelo *watershed* para segmentar o corpo caloso. O critério utilizado para escolher os marcadores para o *watershed* foi definir em quantas regiões a imagem deveria ser segmentada. No experimento apresentado, marcadores foram impostos às bases com os 60 maiores valores de extinção, segmentando, desta forma, a imagem em 60 regiões.

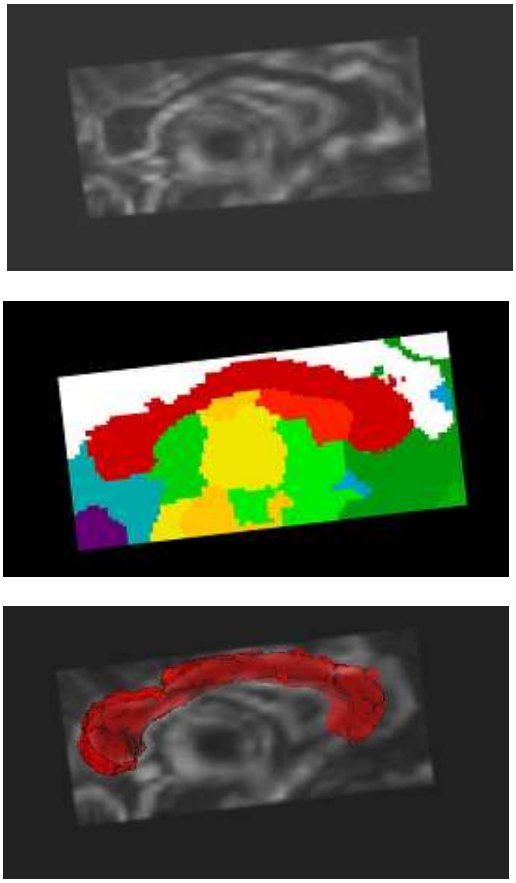


Figura 3. Segmentando o corpo caloso: TMG, resultado do *watershed* e resultado 3D

Fig. 3 apresenta os três passos do processo de segmentação: Fig. 3(a) contém uma fatia do TMG calculado, Fig. 3(b) mostra a mesma fatia rotulada de acordo com o resultado do *watershed* e Fig. 3(c) mostra o resultado da segmentação $3D$. Como pode ser visto na Fig. 3(a), o TMG conseguiu detectar as bordas do corpo caloso, e a área escura mostra que a difusão dentro do corpo caloso é bem homogênea, levando a um gradiente quase nulo. Na Fig. 3(b) cada cor representa um rótulo, mostrando que a transformada de *watershed* foi capaz de atribuir um único rótulo ao corpo caloso (área vermelha) na parte superior da imagem.

É importante ressaltar que, apesar da Fig. 3(a) e da Fig. 3(b) mostrarem apenas uma fatia, tanto o TMG quanto o *watershed* foram calculados

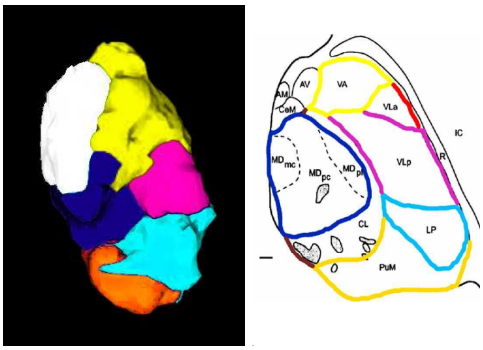


Figura 4. Segmentação do tálamo: método proposto *versus* atlas histológico

levando em conta a informação tridimensional. Isto é garantido pela escolha do elemento estruturante (neste caso, um elemento 6-conexo).

Finalmente, escolhemos o problema de segmentação dos núcleos do tálamo para validar o método de segmentação proposto. A segmentação dos núcleos do tálamo é de extrema importância para neuro-cientistas e neuro-cirurgiões e sua solução através de imagens só passou a ser possível depois do surgimento da DTI. Até então, a divisão do tálamo em núcleos só era feita *post-mortem*.

Para segmentar os núcleos do tálamo usando o método proposto, primeiro foi necessário usar um método de segmentação automática para encontrar a borda externa do tálamo. Esta borda foi então usada como marcador externo para o *watershed*, enquanto que os marcadores internos foram escolhidos pela dinâmica dos mínimos regionais. Os resultados obtidos pelo método proposto foram bastante surpreendentes, correspondendo à divisão em núcleos descrita em atlas histológico (Fig. 4).

4. Conclusões

O método de segmentação de DTI baseada no TMG e na transformada de *watershed* mostrou-se capaz de segmentar diversos tipos de imagens tensoriais. De maneira geral, a norma de Frobenius usada no cálculo do TMG levou aos melhores resultados de segmentação, apesar de não ser invariante à rotação. Isso se deve ao fato de a invariância à rotação ter se mostrado menos importante para o cálculo de um gradiente a ser usado na segmentação, do que a resposta linear às diferenças de anisotropia e traço.

Apesar de termos conseguido uma boa segmentação das estruturas do cérebro, mesmo em problemas complexos como o dos núcleos do tálamo, ainda há necessidade de aumentar a robustez nas medidas, pois a relação sinal ruido ainda é baixa. Os bons resultados de segmentação obtidos em um conjunto de dados de difusão não se repetiram em outros conjuntos de dados, mesmo os adquiridos em um mesmo indivíduo em instantes distintos.

Referências

[1] P.J. Basser and C. Pierpaoli. Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI. *J. Magn. Reson.*, 111(3):209–219, June 1996.

[2] S. Beucher and F. Meyer. The morphological approach to segmentation: The watershed transformation. In *Math. Morph. Image Proces.*, chapter 12, pages 433–481. 1992.

[3] E. R. Dougherty and R. A. Lotufo. *Hands-on Morphological Image Processing*, volume TT59. SPIE, 2003.

[4] L. Jonasson, X. Bresson, P. Hagmann, O. Cuisenaire, R. Meuli, and J. Thiran. White matter fiber tract segmentation in DT-MRI using geometric flows. *Medical Im. Anal.*, 9(3):223–236, 2005.

[5] C. Lenglet, M. Rousson, and R. Deriche. A statistical framework for DTI segmentation. In *ISBI*, pages 794–797. IEEE, 2006.

[6] S.N. Niogi, P. Mukherjee, and B.D. McCandliss. Diffusion tensor imaging segmentation of white matter structures using a reproducible objective quantification scheme (roqs). *NeuroImage*, 35:166–174, 2007.

[7] L. Rittner and R. Lotufo. Diffusion tensor imaging segmentation by watershed transform on tensorial morphological gradient. *Brazilian Symp. on Computer Graph. and Image Proc.*, 0:196–203, 2008.

[8] Z. Wang and B.C. Vemuri. DTI segmentation using an information theoretic tensor dissimilarity measure. *IEEE Trans. Med. Imag.*, 2005.

[9] Y.T. Weldeselassie and G. Hamarneh. DT-MRI segmentation using graph cuts. In *Medical Imaging: Image Processing*. SPIE, 2007.

[10] L. Zhukov, K. Museth, D. Breen, R. Whitaker, and A. Barr. Level set modeling and segmentation of DT-MRI brain data. *J. Electronic Imaging*, 12:125–133, 2003.

[11] U. Ziyan, D. Tuch, and C.F. Westin. Segmentation of thalamic nuclei from DTI using spectral clustering. In *MICCAI'06*, Lect. Notes Comp. Sci., pages 807–814, Denmark, 2006.

Comparação e classificação de algoritmos de rotulação de componentes conexos em processamento de imagens

Victor Matheus de Araujo Oliveira , Roberto Alencar lotufo (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

victormatheus@gmail.com,lotufo@unicamp.br

Abstract – In this article we propose a classification and a comparison of some Connected Components Labeling algorithms, we will consider algorithm complexity, cache memory use and paralelization aspects, specially in GPU architecture. Also, we're going to use the collaborative scientific writing and programming platform Adessowiki [3] to make our benchmarks public.

Keywords – CCL, image processing, GPU, parallel algorithms

1. Introdução

O uso de algoritmos de rotulação de componentes conexos em grafos remonta ao próprio início da ciência da computação [2]. Do mesmo modo, seu uso em imagens remonta ao começo do estudo do processamento digital de imagens [4].

Sendo a rotulação, como também passaremos a chamá-la a partir de agora, uma ferramenta importante não só em processamento de imagens, mas também em engenharia e física, é natural que muitos algoritmos tenham sido propostos até o momento na literatura. Nestes artigos, muitas vezes são mostrados algoritmos, implementações e benchmarks.

Porém, a longo prazo existem mudanças graduais de linguagem de programação, na arquitetura das máquinas, entre outros, que são capazes de fazer com que algoritmos que possuíam baixa performance sejam eficientes hoje em dia. Podemos tomar como exemplo desse fenômeno a recente renovação no ramo da computação paralela, algoritmos destinados à execução em máquinas massivamente paralelas se tornam mais competitivos com o auxílio de placas gráficas.

Assim, para verificar o desempenho de um algoritmo não basta apenas possuirmos o próprio algoritmo, sua implementação e benchmarks, mas também sua *execução*.

Para isso, utilizaremos a plataforma de programação colaborativa Adessowiki [3], em que os códigos dos algoritmos na linguagem Python e em C estarão disponíveis publicamente, assim como as imagens de teste, resultados de benchmarks e a própria execução do algoritmo, que acontecerá no servidor do Adessowiki.

2. Descrição do problema

Para definir o problema da rotulação, precisamos dos conceitos de *imagem*, *vizinhança*, *conectividade*, *caminho conexo*, *componente conexo* e *partição*.

Uma *imagem binária* é um par $\hat{I} = (D_I, I)$ em que $D_I \subset \mathbb{Z}^2$ e I é um mapeamento $I(p) \in \{0,1\} \forall p \in D_I$. Também podemos estabeler dois conjuntos: A *frente*, em que $I(p) = 1$ e o *fundo*, onde $I(p) = 0$. Chamaremos a frente de $\mathbf{F}$ e o fundo de $\mathbf{B}$.

Nesse artigo, trataremos apenas de imagens dimensionais binárias, já que alguns conceitos como o de componente conexo são mais simples de serem descritos em imagens binárias e o funcionamento dos algoritmos de rotulação pode ser facilmente generalizado para dimensões maiores e para imagens multi-banda.

Vizinhança é uma relação binária $D_I \times D_I$ entre os pixels em uma imagem que depende de suas posições relativas e possivelmente de algum outro atributo da imagem. Por exemplo, a *4-Vizinhança* de um pixel p de coordenadas (x, y) é o conjunto $\boldsymbol{V}(p) = \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\}$.

A *conectividade* entre pixels se determina se além deles serem vizinhos, também atendem a algum critério com relação à seus atributos. Por exemplo, a relação *4-conexa*, em que dois pixels p e q estão 4-conectados se são 4-vizinhos e ambos estão em $\mathbf{F}$. Se dois pixels atendem a uma relação de

conectividade dizemos que são *adjacentes*.

Um *caminho conexo* entre p e q, respectivamente de coordenadas (x_0, y_0) e (x_n, y_n), é uma sequência de pixels distintos de coordenadas $(x_0, y_0), (x_1, y_1), \cdots, (x_n, y_n)$ em que (x_i, y_i) é adjacente a (x_{i-1}, y_{i-1}) para $1 \leq i \leq n$.

Um subconjunto S de uma imagem binária é um *componente conexo* se dado um pixel $p \in S$, S é formado por todos os pixels que possuem um caminho conexo até p e é maximal, ou seja, todos os pixels da imagem que possuem algum caminho conexo até p então em S.

Uma *partição* de uma imagem é um conjunto de componentes conexos disjuntas cuja união é D_I.

A *rotulação de componentes conexos* é o problema de obter uma partição da imagem e atribuir um *rótulo* $l \in \mathbb{N}$ para cada componente conexo em $\mathbf{F}$.

Existem dois modos geralmente utilizados na literatura de atribuir rótulos, a atribuição *sequencial* e a de *menor endereço*.

Na atribuição sequencial atribuímos o rótulo à componente de acordo com a ordem em que o componente é processado, portanto, teremos um conjunto $l \in \{1, 2, \cdots, L\}$ de rótulos, em que L é o número total de componentes conexos na imagem.

Na atribuição de menor endereço, o rótulo do componente fica sendo o menor endereço raster dos pixels naquele componente conexo.

A estratégia que será usada ao longo deste artigo é a de menor endereço, pois ela não escalona melhor em algoritmos paralelos e é mais fácil de implementar no geral. Todos os algoritmos aqui apresentados serão adaptados para seguir essa regra. Assim garantimos que todas as rotulações terão o mesmo resultado independentemente do algoritmo utilizado.

3. Exemplo de aplicação

Usaremos a rotulação para identificar letras, palavras e parágrafos em uma imagem binária em que o texto está em $\mathbf{F}$.

Para separar letras, usamos o critério tradicional de 8-conectividade. Portanto, dado um certo pixel $p \in \mathbf{F}$ e um pixel q ao redor de p, em que

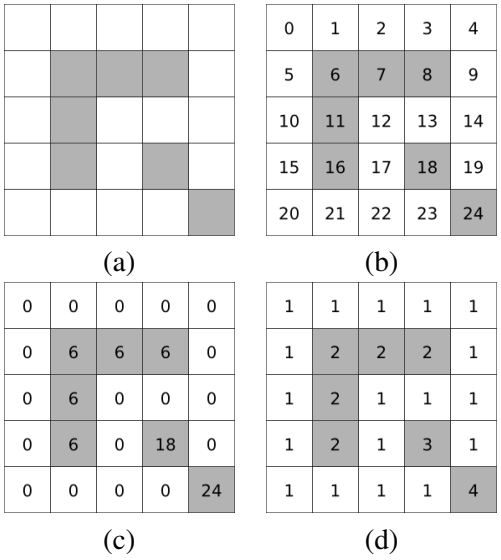


Figura 1. (a) Imagem binária; (b) Endereços na ordem raster; (c) Rotulação sequencial; (d) Rotulação usando menor endereço raster

$q \in \mathbf{F}$, então p e q são adjacentes.

Palavras são letras próximas uma da outra. Para separar as palavras, usamos uma relação de conexidade diferente. Dado um certo pixel $p \in \mathbf{F}$, fazemos um retângulo de 7 pixels de altura e 11 de altura e colocamos p no centro deste retângulo. Todos os pixels que estiverem dentro do retângulo e estão em $\mathbf{F}$ são adjacentes a p. Os valores 7 e 11 foram escolhidos experimentalmente e dependem do tamanho da fonte.

Do mesmo modo, parágrafos são palavras próximas uma da outra. Neste caso, a conectividade vai ser dada por um retângulo de 35 por 20 pixels.

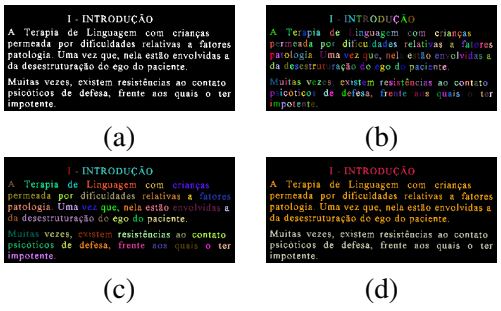


Figura 2. (a) Exemplo de imagem binária com texto; (b) Segmentação das letras; (c) Segmentação de palavras; (d) Segmentação de parágrafos

4. Busca em largura

A busca em largura é uma técnica muito utilizada para percorrer grafos [2]. Uma imagem binária $\hat{I}$ pode ser vista como um grafo G em que os pixels são os vértices e a relação de adjacência estabelece as arestas. Esse procedimento faz uma busca em largura em G para descobrir as componentes conexas.

No seguinte código Python podemos ver o funcionamento do algoritmo. Há como entrada uma imagem I e como saída, a imagem rotulada L, Q é uma fila FIFO que guarda os pixels que ainda serão rotulados.

```
def labeling_BFS(I):
  L = Array(I.N, NOLABEL)
  Q = Queue()

  for p in I.domain:
    if L[p] == NOLABEL and I[p]:
      L[p] = p
      Q.enqueue(p)
      while Q != 0:
        q = Q.dequeue()
        for n in I.adjacency(q):
          if L[n] == NOLABEL:
            L[n] = p
            Q.enqueue(n)
```

5. Conjuntos disjuntos

Esse algoritmo usa uma estrutura de conjuntos disjuntos [1]. Percorre-se a imagem fazer a união de um pixel com seus adjacentes, Quando um pixel p se une a um pixel q, buscamos os representantes dos componentes a que p e q pertencem e fazemos com que um dos representantes se ligue a outro, juntando os componentes um um só. No fim desse processo teremos os componentes conexos da imagem.

Abaixo está o código Python da implementação do algoritmo. Osamos a estratégia do encurtamento de caminho no procedimento *find*. Temos como entrada a imagem I, como saída a imagem rotulada L e como estrutura auxiliar o vetor P, que guarda a estrutura de conjuntos disjuntos, o vetor P é inicializado com a sequencia $\{0, 1, \cdots, \left\|I\right\| - 1\}$, pois no início, todo pixel é sua própria componente conexa.

```
def merge(P, x, y):
  a = self.find(P, x)
  b = self.find(P, y)
  if a < b:
    P[b] = a
  elif a > b:
    P[a] = b

def find(P, x):
  k = x
  while P[k] != k:
    k = P[k]
    equiv = k
    k = j = x
    while P[k] != k:
      j = k
      k = [k]
      P[j] = equiv
  return equiv

def labeling_UnionFind(I):
  P = Sequence(I.N)

  for i in I.domain:
    if I[i]:
      for j in I.adjacency(p):
        merge(P,i,j)
    else:
      P[i] = NOLABEL

  for i in I.domain:
    find(P,i)

  return P
```

6. Propagação

O algoritmo de propagação difere dos mencionados anteriormente porque é *iterativo*, portanto, o número de passos do algoritmo depende da entrada fornecida.

O funcionamento é simples, em uma iteração do algoritmo cada pixel verifica se seus adjacentes possuem um rótulo menor que o seu, se for o caso, ele troca seu rótulo pelo menor. Essa etapa é repetida até a convergência.

No código Python a seguir temos a implementação do algoritmo de propagação, além da etapa de verificação dos rótulos das adjacências (*scan*), é feita uma etapa em que se resolve as cadeias de rótulos (*analysis*). Essa etapa serve para acelerar a convergência do algoritmo.

```
def scan(I, L):
  changed = False
  for p in I.domain:
    tmp_label = L[p]
    for q in I.adjacency(p):
      tmp_label = \
        min(tmp_label, L[q])
    if tmp_label < L[p]:
      L[p] = tmp_label
      changed = True
  return changed

def analysis(I,L):
  for p in I.domain:
    q = p
    while q != L[q]:
      q = L[q]

    equiv = q
    r = q = p
    while L[q] != L[L[q]]:
      r = q
      q = L[q]
      L[r] = equiv

def labelling_Propagation(I):
  L = Sequence(I.N)

  changed = True
  while changed:
    changed = scan(I,L)
    analysis(I,L)

  return L
```

Por ter um funcionamento simples e acesso de memória bem organizado, o algoritmo de propapagação é um bom candidato à paralelização.

Referências

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Chapter 21: Data structures for disjoint sets. In *Introduction to Algorithms*, pages 498–524. MIT Press, 2001.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Chapter 22: Elementary graph algorithms. In *Introduction to Algorithms*, pages 552–556. MIT Press, 2001.

[3] Lotufo et. al. Adessowiki - on-line collaborative scientific programming platform. *Wikisym*, 2009.

[4] Azriel Rosenfeld. Connectivity in digital pictures. *J. ACM*, 17(1):146–160, 1970.

Sistemas Inteligentes

Algoritmo Genético com Coordenação Fuzzy para Resolução do Problema de Roteamento de Veículos com Janelas de Tempo

Vitor Marques , Fernando Gomide (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

{vmarques,gomide}@dca.fee.unicamp.br

Abstract – This paper presents a genetic algorithm coordinated by fuzzy rule models to solve the vehicle routing problem with time windows. The fuzzy rule-based coordinators play distinct roles during the genetic algorithm execution. The aim is to trade-off exploration and exploitation behavior for route and distance minimization. Experimental results using classic benchmark test instances suggest that the fuzzy coordinated genetic approach is competitive against classic genetic algorithm.

Keywords – GFS, VRPTW, Genetic Algorithm

1. Introdução

O problema de roteamento de veículo com janelas de tempo (PRVJT) é um problema combinatorial e tem sua origem nos problemas clássicos do caixeiro viajante (PCV) e no problema do empacotamento (PE), dois respeitados e conhecidos problemas np-difíceis. Aplicações para PRVJT são inúmeras, especialmente em transportes e sistemas de logística como serviços de entrega, postagem e distribuição.

Já foi gasto um grande esforço para resolver PRVJT até 100 clientes e enfatizando minimização de distânica [7],[16]. Instâncias mais práticas, todavia, necessitam meta-heurísticas para obtenção de solução com tempo de computação razoável.

Este trabalho propõe o uso de um sistema baseado em regras fuzzy (SBRF) para coordenar um algoritmo genético (AG) para resolver PRVJT. O SBRF coordenador é um conjunto de modelos baseados em regras cujo propósito é manter o equilíbrio entre intensificação e diversificação no AG. A idéia principal é tentar evitar paradas precipitadas em ótimos locais; e como é bem sabido, boas soluções para PRVJT está em pequenas partes de todo espaço de busca [20]. Ainda, uma dificuldade adicional para AGs quando resolvem PRVJT é que pequenas alterações na geração de um AG geralmente direciona o algoritmo para uma parte do espaço de busca onde ele não consegue evoluir. A solução adotada para superar tal dificuldade tem sido especializar os operadores do GA [20],[26]. Claramente, a natureza do PRVJT requer um controle cuidadoso das características de intesificação e diversificação e sistema baseados em regras fuzzy [21] proporciona um mecanismo poderoso para resolver isto.

O uso de SBRF para controle dinânico de algoritmos genéticos, foi inicialmente proposto em [18], onde taxa de recombinação e tamanho da população eram dinamicamente escolhidos durante execução do AG. [17] usa controlador fuzzy dinâmico para resolver o PRV multi-objetivo com múltiplos depósitos e produtos. Aqui o controle é sobre a variação da taxa de recombinação e mutação, usando fitness médio ao longo das gerações e uma medida de diversidade média da população. Mais especificamente, o controle é realizado em um procedimento de dois passos. O primeiro passo coleta valores do fitness médio, diversidade da população e tamanho médio das rotas dentro da população. O segundo passo define taxa de recombinação, mutação e da busca local. O tamanho da população não está sujeito a controle.

Entre as várias meta-heurísticas existentes, uma de grande sucesso para PRVJT é a busca tabu (BT) [8]. Uma outra importante meta-heurística é sistema de colônia de formigas. [10] apresenta um algoritmo com múltiplas colônias, uma para minimização de veículos e outra para minimização de distância. Uma outra abordagem efetiva é a combinação de meta-heurísticas. Por exemplo, [5] propõe um busca reativa em vizinhaça variável, uma variação da BVV com quatro fases: inicialização, redução de rotas, minimização da distância total usando quatro novas buscas locais e uma modificação da função objetivo para escapar do máximo local.

Muitas das mais bem sucedidas meta-heurísticas para instâncias grandes do PRVJT usam

alguma forma de computação paralela. Em [14], busca com estratégias cooperativas e paralela são criadas usando meta-heurísticas em duas fases; a primeira fase tenta minimizar o número de veículos usando uma meta-heurística evolucionária, a segunda fase tenta minimizar a distância total percorrida com uma busca tabu. Como uma alternativa, [4] apresenta um método de múltiplas buscas cooperativas e paralelas baseado na estratégia de armazém ("warehouse"), na qual várias "threads"de busca cooperam trocando informações assincronamente a respeito das melhores soluções identificadas. A pesquisa [9] exibe mais detalhes e realiza comparações de resultados.

Diferentemente das abordagens correntes visitadas na literatura, este trabalho propõe modelos baseados em regras fuzzy para coordenar algoritmos genéticos na resolução do PRVJT. O propósito dos coordenadores com SBRF é dinamicamente escolher taxa de reprodução e mutação, estruturas de vizinhaça local e o comportamento de busca do AG (intensificação, diversificação) usando uma taxa de melhoria, uma medida para diversidade, proporção média para incumbente e proporção média de rotas pequenas como entradas.

Após esta introdução o artigo segue da seguinte maneira. Seção 2. apresenta a formulação do PRVJT e a seção 3. detalha o AG coordenado com fuzzy. Resultados experimentais são resumidos na seção 4.. Seção 5. conclui o artigo e sugere itens para sequência de trabalhos.

2. Formulação do PRVJT

Este trabalho trata do PRVJT com um depósito. Mais especificamente, existe um conjunto de clientes que possuem uma demanda de transporte específica e uma janela de tempo determinando quando a demanda deve ser atendida. Veículos com capacidades idênticas realizam o atendimento do serviço. O objetivo é achar o conjunto de rotas com custo mais baixo começando e terminando no depósito dentro do horário de trabalho. Factível significa que os veículos devem respeitar suas capacidades e atender a demanda do cliente dentro da janela de tempo especificada por este. Cada cliente é atendido uma vez. O serviço pode ser definido antes da janela de tempo (neste caso o veículo espera), mas não pode ultrapassar a janela. Essa é a abordagem rígida ("hard") para o PRVJT.

O PRVJT pode ser associado a um grafo $G(V, A)$ onde $V = C \cup \{v_0, v_{n+1}\}$ e $C = \{v_1, \ldots, v_n\}$ represetam os n clientes e v_0, v_{n+1} o depósito. O conjunto $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ define arestas entre nós clientes, e v_0 e v_{n+1} representam o depósito no início e fim da rota, respectivamente. Conseqüentemente, não existe aresta começando em v_{n+1} nem uma aresta terminando em v_0. Cada aresta (v_i, v_j) têm um custo c_{ij} em um tempo de viagem associados t_{ij}. Um tempo de serviço s_i é associado ao cliente i que tem demanda d_i, $i \in C$. O depósito tem um conjunto K veículos com capacidade q para servir os clientes. Uma janela de tempo $[e_i, l_i]$ de ve ser obedecida quando inicia-se o serviço do cliente i. As variáveis de decisão são $X^k ij$ que assume o valor 1 se o veículo k atravessa aresta (i, j) $\forall k \in K, \forall (i, j) \in A$.

Deixe $S^k i$ ser o tempo que o veículo k começa atender o cliente i, $\forall k \in K, \forall i \in C$. O problema PRVJT pode ser assim descrito:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} X_{ij}^k \tag{1}$$

sujeito a:

$$\sum_{k \in K} \sum_{j \in C \cup \{v_{n+1}\}, j \neq i} X_{ij}^k = 1, \quad \forall i \in C \tag{2}$$

$$\sum_{i \in C} \sum_{j \in C \cup \{v_{n+1}\}, j \neq i} d_i X_{ij}^k \leq q, \quad \forall k \in K \tag{3}$$

$$\sum_{j \in C \cup \{v_{n+1}\}} X_{0j}^k = 1, \quad \forall k \in K \tag{4}$$

$$\sum_{i \in C} X_{i,n+1}^k = 1, \quad \forall k \in K \tag{5}$$

$$\sum_{i \in C \cup \{v_0\}} X_{ih}^k - \sum_{j \in C \cup \{v_{n+1}\}} X_{hj}^k = 0, \forall h \in C, \forall k \in K \tag{6}$$

$$X_{ij}^k (S_i^k + s_i + t_{ij} - S_j^k) \leq 0, \quad \forall (i, j) \in A, \forall k \in K \tag{7}$$

$$e_i \leq S_i^k \leq l_i, \quad \forall i \in C \cup \{v_{n+1}\}, \forall k \in K \tag{8}$$

$$X_{ij}^k \in \{0, 1\}, \quad \forall (i, j) \in A, \forall k \in K \tag{9}$$

Portanto, o objetivo é minimizar o custo geral de transporte considerando as seguintes restrições: (2) todos clientes devem ser atendidos por um único veículo, (3) toda demanda atendida por um veículo não pode ultrapassar sua capacidade de carga, (4) toda rota deve iniciar no nó v_0, (5) todo

rata deve terminar no nó v_{n+1}, (6) um veículo deve atender e sair do cliente, (7) se um cliente j vêm depois do cliente i em uma rota, o tempo de serviço i e o tempo total de viagem (i, j) são considerados, (8) um veículo deve atender o cliente dentro de sua janela de tempo.

3. Coordenação Fuzzy do AG

A coordenação fuzzy do algoritmo genético (CFAG) tem três grandes componentes: algoritmo genético, algoritmo de busca local e coordenação com base de regras fuzzy, respectivamente. Fig. 1 ilustra a abordagem CFAG. Segue os detalhes destes três componentes.

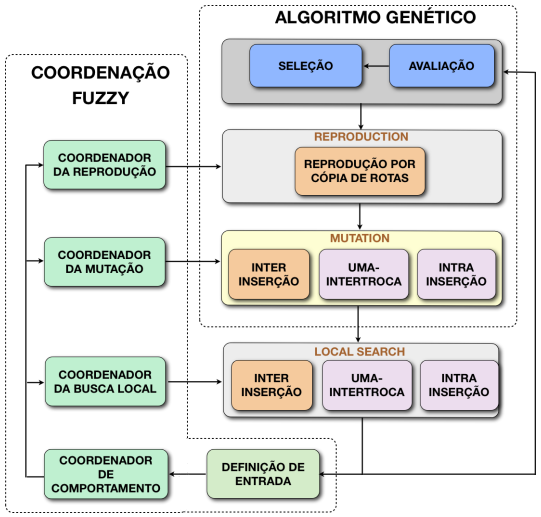


Figura 1. CFAG

3.1. Algoritmo Genético

Um cromossomo representa a ordem dos clientes a serem visitados. Os clientes são colocados em sequência. O depósito é omitido, mas é considerado implicitamente. Fig. 2 mostra a estrutura do cromossomo. Note que o cromossomo é composto por subsequências, cada uma correspondendo à uma rota factível. A população inicial é gerada da seguinte maneira. O primeiro indivíduo é gerado utilizando *"push forward insertion heuristics"*. Os indivíduos restantes são gerados das mesmas estruturas de vizinhaça usadas na mutação. Diversidade populacional é garantida usando *medida de Jaccard* para evitar indivíduos similares.

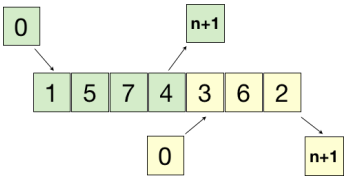


Figura 2. estrutura do cromossomo

3.1.1. Operadores do AG

A *reprodução* é baseada no operador de recombinação por cópia de rotas. Este operador copia rota dos dois parentes para formar os filhos. Se um cliente não está em alguma rota, ele é inserido em alguma rota que permaneça factível. Se isto não for possível, novas rotas são formadas com estes clientes. O procedimento de *mutação* usa três estruturas de vizinhaças: *inter-inserção* , *intra inserção* e *uma-intertroca*, respectivamente. Vizinhaças são escolhidas usando uma distribuição uniforme. A *seleção* é executada usando uma variação da roleta russa, chamado de procedimento de amostragem universal estocástica [3], que busca reduzir a chance de repetir indivíduos com alto fitness, na nova população.

3.1.2. Fitness

A função de fitness considera tanto distância total (dt) quanto número de veículos (nv) na solução: $fitness(dt, nv) = \left(w_d \frac{1}{dt}\right) + \left(w_v \frac{1}{nv}\right)$, onde w_d e w_v são pesos com valores reais. O propósito do AG é maximizar o fitness, que acontece para os menores valores de dt e nv.

3.2. Busca Local

O procedimento de busca local usa as mesmas estruturas de vizinhaça da mutação. Ela sempre tenta melhorar o incumbente. A taxa da busca local pode ficar mais alta ou mais baixa dependendo do coordenador fuzzy da busca local. Ainda, o coordenador pode enfatizar redução de rotas (aumentando quantidade de movimentos de inter-inserção), ou redução da distância (aumentando quantidade de movimentos intra-inserção e uma-intertroca) dependendo da fase da busca e da proporção média das rotas pequenas.

3.3. Coordenação por Base de Regras Fuzzy

O propósito dos coordenadores por base de regras fuzzy é ajudar o AG é diversificar a busca sempre que ele fica parado em ótimos locais, e intensificar a busca sempre que atingir uma nova região do espaço de busca. Eles decidem também quando minimização de veículo ou distância deve ser reforçada. Os coordenadores fuzzy são modelos de base de regras linguísticas. Por simplicidade, este trabalho usa o procedimento de inferência clássico max-min com defuzificação usando centro de gravidade. As variáveis de entrada e saída e a base de regras de cada coordenador são explanadas na sequência.

Entradas: TG, taxa de ganho, melhoria. Mede a capacidade atual da busca local de melhorar uma solução; é a fração *número de todos movimentos factíveis que melhoram a solução* e o *número de todos movimentos factíveis*. DIV, diversidade da população. Mede quão diferente são os indivíduos da população. É a fração *números de arestas usadas da população* e *número total de arestas factíveis*. PMI, proporção média para incumbente. Mede quão distante estão os indivíduos da população do incumbente atual em termos da função de fitness. Proporção da população para incumbente (PI) é a fração entre *fitness de um indivíduo* e *fitness do incumbente* . Logo PMI é a média do PI da população. PRP, proporção média de rotas pequenas. Mede a proporção de rotas pequenas no conjunto de todas as rotas. Uma rota pequena é definida da seguinte maneira. Deixe mcr ser a média de clientes por rota, cv a capacidade do veículo, md a média de demanda entre todos clientes, tfd tempo de fechamento do depósito, tms o tempo médio de serviço dos clientes, e tmv o tempo médio de viagem entre todos clientes. Define-se mcr como

$$mcr = \min\left(\left\lceil \frac{cv}{md} \right\rceil, \left\lceil \frac{tfd}{(tmv + tms)} \right\rceil\right),$$

onde$\lceil x \rceil$ significa o maior inteiro mais perto de x. Uma rota pequena é uma rota com menos clientes que um percentual de mcr. Este artigo adota 20% em todos experimentos.

Saídas: FASE, indica se o algoritmo deve iniciar intensificação, diversificação ou uma fase intermediária de busca. TR, taxa de reprodução. TM, taxa de mutação. TB, taxa de busca. Proporção da população para a qual a busca local é aplicada. TMV, taxa de minimização de veículos, percentual da taxa de busca que reforça a minimização de veículos durante a busca local. TMD, taxa de minimização de distância, percentual da taxa de busca que reforça a minimização de distância durante a busca local.

Todas variáveis linguísticas, exceto FASE, usa *P* para pequeno, *M* para médio, *G* para grande, *B* para baixo *A* para alto. Para FASE, *I* significa intensificação, *D* diversificação, e *IN* intermediário.

3.3.1. Coordenadores Fuzzy

O **coordenador do comportamento da busca** recebe a taxa de ganho e diversidade da população durante a execução do AG e processa a base de regras para escolher entre intensificação, diversificação ou comportamento intermediário de busca. Um exemplo de regra é: SE TG é *M* e DIV é *P* ENTÃO FASE é *D*. A base de regras especifica, se a busca local não está melhorando a solução satisfatoriamente, o algoritmo deve enfatizar a diversificação, entrando na fase de diversificação. Quando altas taxas de ganhos são observadas, o algoritmo entra em uma fase de intensificação. Com taxas de ganhos intermediárias, se a diversidade é baixa, a diversificação é incentivada, caso contrário a intensificação é favorecida. O **coordenador da reprodução** recebe a informação da fase do coordenador de comportamento de busca e a proporção média para incumbente para definir a taxa de reprodução. A base de regras diz, por exemplo, que a taxa de reprodução deve ser alta quando algoritmo está em fase de intensificação e baixa durante diversificação. A proporção média para incumbente (PMI) evita empates quando a fase não é de intensificação nem diversificação. O **coordenador da mutação** recebe a informação da fase do coordenador de comportamento de busca e a proporção média para incumbente para definir a taxa de mutação. O **coordenador da busca local** usa a informação de fase do coordenador de comportamento da busca e a proporção média de rotas pequenas para definir a taxa da busca, a taxa de minimização de veículos e taxa de minimização de distância. A base de regras determina, por exemplo, que quanto mais alta a diversificação mais baixa é a taxa de busca, quanto mais alta é a intensificação maior é a taxa de busca.

4. Resultados Experimentais

O algoritmo coordenado por regras fuzzy sugerido neste trabalho foi testado com as instâncias de Solo-

mon [23], um benchmark clássico para PRVJT. Ele usa três tipos de geração de clientes e suas localizações: randômica (instâncias R), de agrupamento (instâncias C), e combinação dos dois (instâncias RC). O resultados reportados são os melhores de 10 rodadas de execução, com 200 gerações cada, usando um Pentium Intel Core 2 duo, 2.4GHz with 2GB RAM. O algoritmo foi implementado usando Java e JFuzzyLogic [15]. Os resultados alcançados foram comparados contra o AG clássico com parâmetros fixados em TR=0.9, TM=0.1, TB=0.25, TMV=0.5, TMD = 0.5 para todos experimentos. O AG clássico levou em média 10 minutos em média para executar todo conjunto de instâncias. O algoritmo coordenado por base de regras fuzzy levou em média 12 minutos.

Tabela 1 resume os melhores resultados para PRVJT encontrados na literatura. O AG clássico com busca local e parâmetros fixados melhorou levemente a qualidade da solução, quando comparado com o melhor resultado [14], aproximadamente 1% para tanto, número de veículos (NV) e total de distância viajada (D). O AG coordenado por fuzzy se mostrou mais efetivo, apresentando ganho de 2.7% e 6.8% para número de veículos e distância total, respectivamente.

Tabela 1. Resultados para Instâncias de Solomon

	Resultados por grupo de instância		
Autor	**Instância**	**NV**	**D**
AG	R1	11.25	1147.28
	C1	10.00	939.06
	RC1	11.75	1304.72
	R2	2.91	915.90
	C2	3.125	724.69
	RC2	3.125	1011.85
	Accum.	**401.00**	**56624.15**
AG + Fuzzy	R1	11.00	1100.88
	C1	10.00	874.85
	RC1	11.37	1264.62
	R2	2.91	843.91
	C2	3.00	690.30
	RC2	3.125	947.96
	Accum.	**394.00**	**53590.52**
[4]	R1	12.08	1209.19
	C1	10.00	828.38
	RC1	11.50	1386.38
	R2	2.73	960.95
	C2	3.00	589.86
	RC2	3.25	1133.30
	Accum.	**407.00**	**57412.37**
[13]	R1	11.91	1212.73
	C1	10.00	828.38
	RC1	11.50	1386.44
	R2	2.73	955.03
	C2	3.00	589.38
	RC2	3.25	1108.52
	Accum.	**405.00**	**57192.00**
[10]	R1	12.00	1217.73
	C1	10.00	828.38
	RC1	11.63	1382.42
	R2	2.73	967.75
	C2	3.00	589.86
	RC2	3.25	1129.19
	Accum.	**407.00**	**57525.00**

5. Conclusão

Este artigo introduziu um algoritmo genético com coordenação fuzzy para resolver o PRVJT. O propósito da coordenação é balancear diversificação e intensificação, e reforçar minimização de veículos ou minimização de distância durante a execução do algoritmo genético. Resultados experimentais revelam que AG com coordenação fuzzy é competitivo comparado aos melhores resultados da literatura para PRVJT por melhorar a qualidade da solução. Estamos trabalhando também com objetivo de construir um framework para automatização de parâmetros para meta-heurísticas.

Referências

[1] R. Ah King, B. Radha, and H. Rughooputh. A fuzzy logic controlled genetic algorithm for optimal electrical distribution network reconfiguration. In *Networking, Sensing and Control, 2004 IEEE International Conference on*, volume 1, pages 577–582, March 2004.

[2] P. Badeau, F. Guertin, M. Gendreau, J. Potvin, and E. Taillard. A parallel tabu search heuristic for the vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 5(2):109 – 122, 1997.

[3] J. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceeding of the Second International Conference on Genetic Algorithms and their application*, pages 14–21. L. Erlbaum Associates Inc., 1987.

[4] A. Bouthillier and T. Crainic. A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. 32(7):1685–1708, 2005.

[5] Olli Bräysy. A reactive variable neighborhood search for the vehicle-routing problem with time windows. *Informs J. on Computing*, 15(4):347–368, 2003.

[6] Olli Bräysy et al. A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research*, 159(3):586 – 605, 2004.

[7] Alain Chabrier. Vehicle routing problem with elementary shortest path based column generation. *Computers Operations Research*, 33(10):2972 – 2990, 2006.

[8] J. Cordeau and Gilbert Laporte. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52:928–936, 2001.

[9] T. Crainic and M. Gendreau. Cooperative parallel tabu search for capacitated network design. *Journal of Heuristics*, 8(6):601–627, 2002.

[10] L. Gambardella, E. Taillard, and G. Agazzi. Macs-vrptw: A multiple ant colony system for vehicle routing problems with time windows. Technical report, Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, 1999.

[11] H. Gehring and J. Homberger. Two evolutionary metaheuristics for the vehicle routing problem with time windows. *Infor*, 37:297–318, 1999.

[12] Michel Gendreau, Alain Hertz, and Gilbert Laporte. A tabu search heuristic for the vehicle routing problem. *Manage. Sci.*, 40(10):1276–1290, 1994.

[13] J. Homberger and H. Gehring. A parallel two-phase metaheuristic for routing problems with time windows. *Asia-Pacific Journal of Operational Research*, 13(1):35–47, 2001.

[14] J Homberger and H. Gehring. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*, 162(1):220–238, 2005.

[15] Open source fuzzy logic library and fcl language implementation.

[16] B. Kallehauge, J. Larsen, and Oli Madsen. Lagrangean duality applied on vehicle routing with time windows - experimental results. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2001.

[17] H Lau, T. Chan, W. Tsui, F. Chan, G. Ho, and K. Choy. A fuzzy guided multi-objective evolutionary algorithm model for solving transportation problem. *Expert Systems with Applications*, 36(4):8255 – 8268, 2009.

[18] M. Lee and Hideyuki Takagi. Dynamic control of genetic algorithms using fuzzy logic techniques. In *Proc. of the Fifth International Conference on Genetic Algorithms*, pages 76–83, 1993.

[19] Oli Madsen, Niklas Kohl, J. Desrosiers, M. Solomon, and F. Soumis. 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33:101–116, 1999.

[20] B. Ombuki, B. Ross, and F. Hanshar. Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, 24:2006, 2006.

[21] W. Pedrycz and F. Gomide. *Fuzzy Systems Engineering: Toward Human-Centric Computing*. Wiley Interscience/IEEE, Roboken-NJ, USA, first edition, 2007.

[22] J. Potvin and S. Bengio. The vehicle routing problem with time windows - part ii: Genetic search. *Informs J. on Computing*, 8(2):165–172, 1996.

[23] M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.*, 35(2):254–265, 1987.

[24] E. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J. Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31(2):170–186, 1997.

[25] K. Tan, L. Lee, and K. Ou. Artificial intelligence heuristics in solving vehicle routing problems with time window constraints. *Engineering Applications of Artificial Intelligence*, 14(6):825 – 837, 2001.

[26] S. Thangiah. Vehicle routing with time windows using genetic algorithms. In *Application Handbook of Genetic Algorithms: New Frontiers, Volume II*, pages 253–277. Lance Chambers, CRC Press, 1995.

Mecanismo Adaptativo e Evolutivo de Seleção de Comportamentos em um Agente de Software Consciente Aplicado a Programação de Semáforos no Tráfego Urbano

André Paraense , Ricardo Gudwin (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

{paraense,gudwin}@dca.fee.unicamp.br

Abstract – This article gives some guidelines for preparing a four-page paper. Its abstract must be written in English and is limited to 150 words. It should be a fully self-contained description of your work, conveying motivation, problem statement, the way that you plan to solve/have solved your problem, expected/achieved/on-the-road results, and the implications of your results. It is a way to convince people that it is worth taking their time to acquire and read the rest of the paper. We exceptionally cite [3] in this abstract for further reading.

Keywords – They must be in English. Put here keywords that people looking for your paper might use or that may help the review committees or editors assign it to appropriate reviewers.

1. Introdução

- Falar da importância de se estudar consciência e cognição artificial em agentes de software.

- Mostrar a evolução dos trabalhos em consciência artificial, mais especificamente na arquitetura baars-franklin, citando as teses que temos.

- Citar as hipóteses que guiarão a pesquisa.

2. Proposta

- Explicar os modos de seleção de comportamento de um agente: planejador e reativo. Falar que o ser humano implementa um modelo híbrido. Falar que a rede de comportamentos, segundo Pattie Maes, é um modelo híbrido, no qual os objetivos injetam ativação na rede de um lado (planejamento) e os sensores injetam ativação do outro (reativo).

- Explicar a proposta como uma rede de comportamento que evolui, modificando sua arquitetura e seus pesos, escolhendo comportamentos disponíveis em um repositório (pool) de comportamentos baseado em realimentação das suas ações no ambiente. Funciona como se o agente no início fosse um bebê que já soubesse todos os comportamentos, mas não soubesse encadeá-los em redes de comportamento, e aprendesse com suas experiências

- Explicar a aplicação a um problema real, que é o de resolver congestionamento reprogramando os semáforos.

3. Resultados

- mostrar os resultados já alcançados com o simulador validado nos arredores da universidade.

- citar os resultados esperados, com o simulador expandido para a cidade de Campinas inteira e o agente programador de semáforos consciente atuando nisto.

4. Conclusões

Retomar o problema mencionado na seção 1. e sintetizar contribuições e perspectivas.

Referências

[1] James F. Blinn. How to write a paper for siggraph. *IEEE Computer Graphics and Applications*, 7(12):62–64, Dezembro 1987.

[2] Steven L. Kleiman. Writing a math phase two paper. `http://www.mit.edu/course/other/mathp2/www/piil.html`, 1999. (acessado em 08/11/2007).

[3] Philip Koopman. How to write an abstract. `http://www.ece.cmu.edu/~koopman/essays/abstract.html`. (acessado em 07/11/2007).

[4] Rogério Lacaz-Ruiz. Notas e reflexões sobre redação científica. `http://www.hottopos.com.br/vidlib2/Notas.htm`. (acessado em 24/10/2007).

Rede Neural Granular para Modelagem Evolutiva de Sistemas

Daniel F. Leite, Fernando Gomide

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

{danfl7, gomide}@dca.fee.unicamp.br

Abstract – In this paper, we propose a framework to granulate and model partially supervised drifting data streams. The modeling approach consists in evolving granular neural networks capable of processing nonstationary data streams from one-pass incremental algorithms. Evolving granular neural classifiers employ fuzzy hyperboxes and T-S neurons to granulate data and aggregate features then proving discriminant boundaries between classes. Additionally, classification results provide interpretable explanation about the model decision from IF-THEN statements. The associated learning algorithm allows model's structural and parametric adaptation whenever the environment changes. The algorithm needs no prior knowledge about statistical properties of data and classes, and can compute data originally numeric or in the form of confidence intervals. The experiments conducted give clues to the behavior of evolving granular neural networks in nonstationary environments. In particular, the approach has demonstrated to be robust against concept drift, and to be able to cope with missing classes.

Keywords – Evolving Intelligent Systems, Data Stream Mining, Concept Drift, Semi-supervised Learning.

1. Introdução

O aumento da disponibilidade de grandes quantidades de dados tem promovido uma busca por novos algoritmos *on-line* para aprendizagem a partir de fluxos de dados [1]-[8]. A mineração de fluxos de dados em ambientes dinâmicos não-estacionários tem criado problemas únicos e inspirado pesquisa na direção do desenvolvimento de abordagens construtivas para modelagem *on-line*.

Em ambientes não-estacionários, a distribuição estatística que gera os dados (médias, variâncias, estrutura de correlação) muda ao longo do tempo. Tais mudanças podem ser graduais ou abruptas, contractíveis ou expansíveis, determinísticas ou aleatórias, ou mesmo cíclicas, devido a sazonalidades. Modelos de aprendizagem *on-line* devem detectar prontamente estas mudanças.

Desafios envolvidos neste contexto compreendem ainda: **(i)** a impossibilidade de armazenar dados históricos. O modelo deve reter o conhecimento previamente adquirido e que ainda é relevante, enquanto usar apenas os dados mais recentes para adaptação; **(ii)** novos dados podem trazer novas características e novas classes. Estes clamam por adaptação estrutural do modelo; **(iii)** dados com ruído e valores perdidos são ocorrências comuns. Claramente, técnicas padrões de mineração de dados, que assumem estacionariedade e requerem múltiplos passos sobre bases de dados se tornam infactíveis neste contexto.

2. Proposta

Sugerimos para o problema da modelagem *on-line* de fluxos de dados não-estacionários uma abordagem baseada em redes neurais granulares evolutivas (eGNN) [6], [8]. eGNN é uma variante evolutiva de sistemas neuro-fuzzy capaz de lidar com ambientes dinâmicos. eGNN originou-se de nossa pesquisa recente em processamento de fluxos de dados sem a necessidade de re-treinamento de modelos. Neste artigo, focamos classificação parcialmente supervisionada.

Colocado de uma maneira geral, classificadores eGNN sumarizam o comportamento do sistema no espaço característico e as classes associadas a dados de treinamento em regras SE-ENTÃO. Para isto, eGNN usa hiper-caixas fuzzy para granular dados, e neurônios T-S [6] para agregar características. Seu algoritmo de aprendizagem incrementalmente adapta a estrutura e parâmetros de eGNN tão logo que o ambiente muda. Além disso, o algoritmo pode lidar com dados rotulados e não-rotulados simultaneamente usando um procedimento único. Ele pode processar dados originalmente numéricos ou granulares (na forma de intervalos de confiança). Na próxima seção, descrevemos as características gerais da modelagem eGNN. Na seção 4 avaliamos o comportamento de redes neurais eGNN quando estas são submetidas a mudanças de conceito e a diferentes proporções de dados rotulados. O trabalho é concluído com a Seção 5.

3. Redes Neural Granular Evolutiva

3.1 Introdução

O conceito de redes neurais granulares (GNN) foi inicialmente estabelecido em [9], enquanto que o de eGNN foi proposto em [6]. Ambas as abordagens enfatizam redes neurais artificiais capazes de processar dados originalmente numéricos ou granulares. Entretanto, eGNN focaliza aprendizagem incremental *on-line* a partir de fluxo de dados.

A aprendizagem em GNN e eGNN segue um princípio comum que envolve dois estágios conforme ilustrado na Fig. 1. Primeiro, grânulos de informação – intervalos ou conjuntos fuzzy – são construídos a partir de uma base de representação numérica. Em seguida, a aprendizagem – construção e refinamento – da rede neural é baseada nos grânulos de informação ao invés de nos dados originais. Assim, a rede neural não é exposta a todos os dados de treinamento, muito mais numerosos que os grânulos, e.g., quando não transportando novas informações, exemplos são descartados.

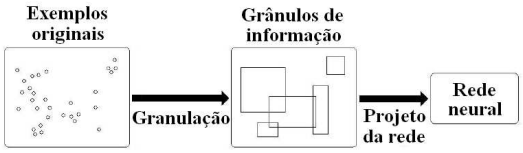


Figura 1. Projeto de redes neurais granulares

Fundamentalmente, modelos eGNN processam dados observando um fluxo somente uma vez. eGNN começa a aprender a partir de uma base de regras vazia e sem conhecimento prévio das propriedades estatísticas dos dados e classes. A abordagem consiste em formar limites discriminantes entre classes a partir da granulação do espaço característico usando hiper-caixas fuzzy. Adicionalmente, os resultados de classificação provêem explicações interpretáveis sobre a decisão do modelo a partir de proposições do tipo SE-ENTÃO. Em suma, entre as características principais de eGNN estão as seguintes, eGNN: ajusta sua estrutura e parâmetros para aprender um novo conceito, enquanto esquece o que não é mais relevante; lida com dados rotulados e não-rotulados utilizando um procedimento único; detecta mudanças no ambiente e lida com incerteza nos dados; possui habilidade não-linear de separação de classes; e desenvolve aprendizado "ao longo da vida" usando mecanismos construtivos *bottom-up* e destrutivos *top-down*.

3.2. Princípio de funcionamento

Redes eGNN aprendem a partir de um fluxo de dados $x^{[h]}$, h = 1, 2, ..., onde os exemplos de treinamento podem ou não ser acompanhados de um rótulo de classe $C^{[h]}$. Cada grânulo de informação γ^j da coleção finita dos grânulos existentes $\gamma = \{\gamma^1, ..., \gamma^c\}$ definido no espaço característico $X \subseteq \Re^n$ é associado a uma classe C_k da coleção finita de classes $C = \{C_1, ... C_m\}$ em um espaço de saída $Y \subseteq \aleph$. eGNN associa o espaço característico e de saída usando grânulos (extraídos do fluxo de dados) e neurônios T-S nos passos de processamento.

A rede neural tem uma estrutura em cinco camadas como ilustrado na Fig. 2. A camada de entrada basicamente insere vetores característicos $x^{[h]} = (x_1, ..., x_n)^{[h]}$, h = 1, 2, ..., na rede neural; a camada evolutiva consiste de um conjunto de grânulos de informação $\gamma^j \forall i$ formado como um escopo do fluxo de dados. Sobreposição parcial de grânulos são permitidas; a camada de agregação contém os neurônios T-S, $TSn^i \forall i$. Eles agregam características para gerar medidas de compatibilidade $o^i \forall i$ entre exemplos e grânulos; na camada de decisão, as medidas de compatibilidade são comparadas e a classe C_k associada ao grânulo γ^v que apresentou a maior compatibilidade para um dado exemplo é induzida na saída da rede como uma estimação para $C^{[h]}$; e a camada de saída (que também evolui durante a aprendizagem), consistindo de rótulos de classes $C_k \forall k$.

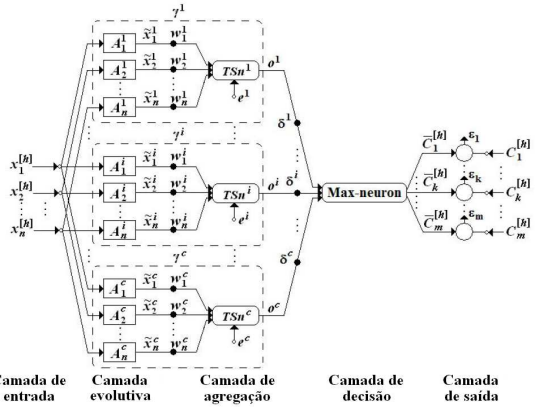


Figura 2. Modelo eGNN para classificação de fluxo de dados

Alternativas para o controle do crescimento estrutural de eGNN compreendem: controle do número de classes (usualmente empregado quando a quantidade de classes nos dados é conhecida); controle automático do número de grânulos na estrutura do modelo (usado quando

memória e tempo de processamento são requerimentos principais); ou nenhum dos citados. A evolução não controlada de grânulos e classes é justificada em ambientes imprevisíveis. A ênfase neste modo é colocada na operação do sistema. Claramente, todos os modos de controle também objetivam desempenho de classificação.

3.3. Aprendizagem em eGNN

Omitiremos as derivações e formulações referentes ao algoritmo de aprendizagem neste trabalho. Pedimos ao leitor que se refira a [6] ou [8] para uma abordagem completa. Basicamente, o algoritmo eGNN compreende os seguintes mecanismos: **(i)** atualização da granularidade; **(ii)** criação/refinamento de grânulos; **(iii)** monitoramento da matriz de distâncias; **(iv)** pré-rotulação de dados não-rotulados; **(v)** adaptação dos pesos da camada de agregação; **(vi)** poda de neurônios e conexões inativas; **(vii)** ajuste de elementos neutros; **(viii)** decisão de classificação; e **(ix)** detecção de *outliers*. Postos de forma sistemática, estes procedimentos formam o algoritmo de aprendizagem eGNN.

4. Resultados Experimentais

Nesta seção provemos resultados empíricos para estabelecer a efetividade da abordagem eGNN. Usamos dados simulados para manter controle sobre os tipos de mudanças nos conceitos e suas implicações na modelagem. Conduzimos dois experimentos com diferentes propósitos. Primeiro, consideramos duas funções Gaussianas parcialmente sobrepostas girando em torno de um ponto central. Cada Gaussiana representa uma classe. Buscamos encontrar um limite discriminante entre as classes tomando por base apenas os exemplos mais recentes. Fazendo isso, avaliamos a habilidade da modelagem eGNN em capturar mudanças graduais do conceito ao longo da evolução. O último experimento ilustra a importância do uso de todas as informações disponíveis no fluxo de dados para guiar o processo de classificação. Além disso, evidenciamos a habilidade de eGNN em classificar fluxos de dados não-estacionários parcialmente rotulados. Maiores detalhes dos experimentos e as parametrizações adotadas podem ser encontrados em [8].

4.1. Rotação das Gaussianas gêmeas

Neste experimento, duas Gaussianas parcialmente sobrepostas, centradas em (4,4) e (6,6) com desvio padrão .8, giram gradualmente no sentido anti-horário até um ângulo de 90º em torno do ponto central (5,5) como mostra a Fig. 3. Queremos encontrar o limite discriminante entre as classes usando apenas os exemplos mais recentes aleatoriamente selecionados.

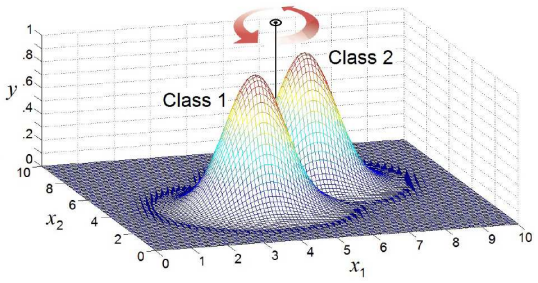


Figura 3. Problema das Gaussianas rotativas

A Fig. 4 mostra o limite de decisão e os últimos 200 exemplos em diferentes instantes da evolução: h = 200 (quando a mudança do conceito se inicia) e h = 400 (quando a mudança termina). Em h = 200, eGNN usa 5 grânulos (dois representando a classe 1 e três representando a classe 2) para modelar os dados. A rede neural obteve um desempenho de classificação correto/errado de 189/11 (94.5%) em h = 200. Após a rotação, em h = 400, a rede neural utiliza 5 grânulos (três para a classe 1 e o restante para a classe 2) e alcança um desempenho de 195/5 (97.5%).

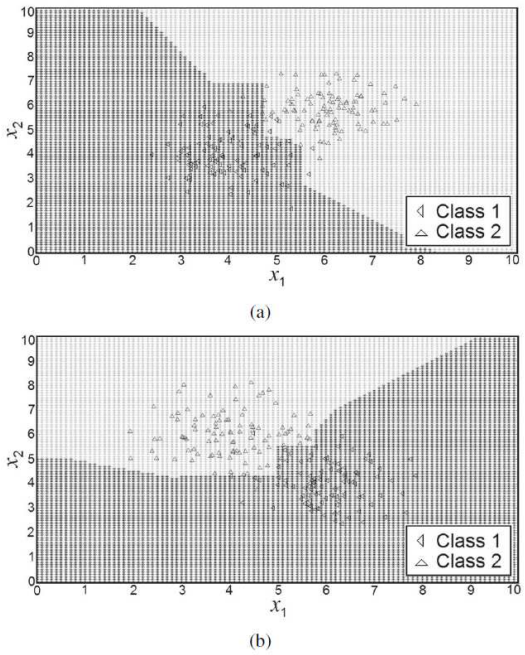


Figura 4. Limite de decisão e últimos 200 exemplos em (a) h = 200; e (b) h = 400

Enquanto hiper-caixas foram preferidas neste trabalho como representação dos grânulos de informação, nós evidenciamos neste experimento que tal consideração não restringe eGNN a lidar somente com distribuições de dados da mesma natureza (na forma de intervalos de confiança). Ao contrário, eGNN não precisa de informação prévia sobre os dados para capturar as mudanças de conceito ocorrendo no fluxo de dados.

4.2. Dados rotulados e não-rotulados

Aqui, o desempenho de eGNN é investigado considerando variações da proporção de dados não-rotulados entre 0% e 100%. Admitimos para isso o problema das Gaussianas rotativas e também um problema de surgimento repentino de uma nova classe no fluxo de dados [8]. A Fig. 5 ilustra o desempenho médio do algoritmo em 5 simulações para cada ponto do gráfico.

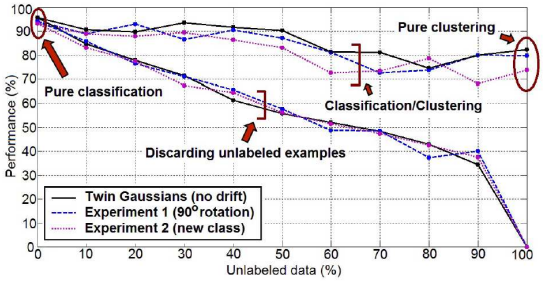


Figura 5. Desempenho de eGNN usando frações de dados não-rotulados

Referindo-se a Fig. 5, removemos sucessivamente a informação referente ao rótulo dos exemplos. A variação em desempenho se mostrou robusta ao número crescente de rótulos perdidos. A performance da rede eGNN com 50% de exemplos não-rotulados degradou ligeiramente em ambos os experimentos usando aprendizagem híbrida. Além disso, claros benefícios da combinação de dados rotulados e não-rotulados no treinamento foram noticiados. Os experimentos aproveitando toda a informação do fluxo de dados guiaram a classificação com sucesso, enquanto aqueles experimentos simplesmente descartando exemplos não-rotulados degradaram o poder de reconhecimento do modelo proporcionalmente a quantidade de dados descartados.

5. Conclusão

Uma abordagem híbrida baseada em redes neuro-fuzzy evolutivas para classificação e clusterização de fluxo de dados não-estacionários (especificamente, rede neural granular evolutiva) foi sugerida neste trabalho. Verificamos que a abordagem eGNN pode classificar dados sujeitos à mudanças de conceito em tempo real. Testamos a efetividade do algoritmo em dois experimentos compreendendo deslize gradual de conceito, e diferentes proporções de dados rotulados. Os resultados mostraram que o algoritmo é robusto a ambientes dinâmicos e também capaz de capturar não-estacionariedades. Trabalhos futuros na direção dos fundamentos da modelagem considerarão processamento de dados originalmente granulares; e aprendizagem participativa. Na direção de novas aplicações, admitiremos problemas de controle e predição de séries.

Referências

[1] P. Angelov; D. Filev. An approach to on-line identification of evolving Takagi-Sugeno models. IEEE Trans. on SMC – Part B, 34(1): 484-498, 2004.

[2] B. Grabrys; A. Bargiela. General fuzzy min-max neural network for clustering and classification. IEEE Trans. on Neural Networks, 11(3): 769-783, 2000.

[3] M. Muhlbaier; A. Topalis; R. Polikar. Learn ++.NC: Combining Ensemble of Classifiers With Dynamically Weighted Consult-and-Vote for Efficient Incremental Learning of New Classes. IEEE Trans. on Neural Networks, 20(1): 152-168, 2009.

[4] S. Ozawa; S. Pang; N. Kasabov. Incremental Learning of Chunk Data for Online Pattern Classification Systems. IEEE Trans. on Neural Networks, 19(6): 1061-1074, 2008.

[5] D. Leite; P. Costa Jr.; F. Gomide. Interval-based evolving modeling. IEEE Workshop on Evolving and Self-Developing Intelligent Systems, 1: 1-8, 2009.

[6] D. Leite; P. Costa Jr.; F. Gomide. Evolving Granular Classification Neural Networks. Int. Joint Conference on Neural Networks, 1: 1736-1743, 2009.

[7] D. Leite; P. Costa Jr.; F. Gomide. Granular Approach for Evolving System Modeling. Lecture Notes in Computer Science (to appear), 2010.

[8] D. Leite; P. Costa Jr.; F. Gomide. Evolving Granular Neural Networks for Semi-supervised Data Stream Classification. World Congress on Computational Intelligence (submitted), 2010.

[9] W. Pedrycz; W. Vukovich. Granular Neural Networks. Neurocomputing, 36: 205-224, 2001.

Automação e Processamento de Sinais

Algoritmo Nebuloso de Distribuição de Vagões Vazios

Joelma Cristina Costa , Rodrigo Gonçalves , Fernando Gomide (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

{joelma,gomide}@dca.fee.unicamp.br,rodrigo@cflex.com.br

Abstract – This paper suggests a fuzzy algorithm to plan empty car distribution considering strategic and operational information. The algorithm is based on a network flow model that considers the train schedule, trains routes, and railroad operation rules. The fuzzy algorithm consider strategic information such as customer reliability, planning horizon, and uncertainty in cars demand. The classic models found in the literature do not consider strategic informations in car distribution plans. In practice strategic information is essencial to develop satisfactory plan and are always taken into account when the distribution planning is constructed by experienced railway planners. Experimental results show that the fuzzy algorithm provides realistic and efficient solutions from the point of view of solution quality and computational performance.

Keywords – Empty Car Distribution, Fuzzy Optimization, Railway Planning.

1. Introdução

O transporte ferroviário inclui uma série de oportunidades de otimização no planejamento tático, estratégico e operacional. Dentre as oportunidades destaca-se o planejamento da distribuição e alocação de vagões.

O problema de distribuição de vagões consiste basicamente no planejamento da movimentação dos vagões vazios na ferrovia de modo a atender à demanda de transporte da ferrovia e minimizar os custos associados a movimentação dos vagões. Portanto, a distribuição de vagões consiste basicamente na atribuição dos pares vagão-demanda com o objetivo de maximizar o atendimento a demanda de transporte e minimizar os custos de movimentação.

O problema da alocação de vagões vazios envolve muito mais o conhecimento estratégico do que o conhecimento puramente formalizável em modelos matemáticos [3]. Em geral, o processo de alocação combina o conhecimento e a experiência operacional com modelos formais para encontrar soluções ótimas para o sistema real.

Exemplos de conhecimento estratégico considerado por distribuidores e alocadores de vagões incluem: a) confiabilidade dos clientes; b) previsibilidade do pedido no hozizonte de tempo; e c) imprecisão da quantidade de vagões solicitados. O conhecimento estratégico influencia diretamente a qualidade da solução gerada pelos algoritmos de distribuição e alocação e não é considerado pelos modelos matemáticos da literatura.

Este trabalho propõe um algoritmo nebuloso para a distribuição de vagões capaz de considerar informações estratégicas.

2. Algoritmo fuzzy de distribuição de vagões vazios

O algoritmo nebuloso de distribuição de vagões utiliza números nebulosos para modelar e tratar a imprecisão na quantidade de vagões das demandas de transporte. As entradas para o algoritmo são: a) demandas de transporte; b) grade de trens; c) posição dos vagões na malha; e d) informações estratégicas. As saídas do algoritmo de distribuição são as atribuições vagão-demanda.

2.1. Modelagem

O algoritmo nebuloso de distribuição de vagões é similar ao algoritmo de distribuição clássico. A diferença é que o algoritmo nebuloso considera a imprecisão na demanda de transporte. A teoria dos conjuntos nebulosos é utilizada para tratar esta imprecisão. No algoritmo de distribuição nebuloso uma demanda de transporte é representada por um número nebuloso.

O algoritmo nebuloso de distribuição de vagões foi inspirado no trabalho de Chanas e Kuchta (1998) que propuseram um algoritmo para resolver problemas de transporte de maneira geral.

Considere

O: conjunto de nós oferta ($\forall$ origem, tipo, instante);

n_o: número de vagões disponíveis em o, $\forall o \in O$;

D: conjunto de nós demanda ($\forall$ destino, tipo, instante);

n_d: número de vagões necessários para atender a demanda d, $\forall d \in D$.

No modelo descrito em [2], n_d é o número de vagões necessários para atender a demanda d, $\forall d \in D$. No modelo de distribuição nebuloso, n_d é substituído por um número nebuloso $\tilde{n}_d$ para representar a imprecisão quanto ao número de vagões necessário para atender a demanda d. Neste caso, as variáveis de decisão $y_d, \forall d \in D$,(demanda não atendida) não são necessárias, visto que a formulação das demandas como números nebulosos considera o equilíbrio entre demanda e oferta de vagões na própria modelagem.

Neste trabalho, de maneira geral, considera-se que $\tilde{n}_d$ é um número nebuloso triangular, pois não se tem outras informações disponíveis baseadas na percepção da prática operacional, que não a demanda aproximada. Nos experimentos computacionais (Sec. 3.1.) utiliza-se uma função de pertinência trapezoidal para representar a imprecisão inerente às demandas pouco confiáveis.

Sendo assim, pode-se reescrever o modelo de distribuição [2] e obter o modelo de distribuição nebuloso como segue:

$$min \sum_{o \in O} \sum_{d \in D} \zeta_{od} x_{od} \quad (1)$$

$$\sum_{o \in \hat{O}_d} x_{od} \cong \tilde{n}_d \quad \forall d \in D$$

$$\sum_{d \in \hat{D}_o} x_{od} + z_o = n_o \quad \forall o \in O$$

$$x_{od} \geq 0 \quad \forall o \in O \quad \forall d \in D$$

$$z_o \geq 0 \quad \forall o \in O$$

onde $\tilde{n}_d$ é um número nebuloso. Este modelo considera os custos de tranporte ζ_{od}, $\forall o \in O$, $\forall d \in D$ e o número de vagões de oferta n_o, $\forall o \in O$, são números reais.

As variáveis de decisão para o problema nebuloso são:

x_{od}: número de vagões de o para d, $\forall o \in O$, $\forall d \in D$.

z_o: oferta não alocada, $\forall o \in O$

Este problema é resolvido pelo algoritmo nebuloso de distribuição resumido na Fig. 1. Este algoritmo foi inspirado no trabalho de [1]. A Fig. 1 dá uma visão geral do algoritmo de distribuição, suas interfaces no ambiente ferroviário, as etapas do algoritmo que foram inspiradas no trabalho de [1] e as contribuições deste trabalho.

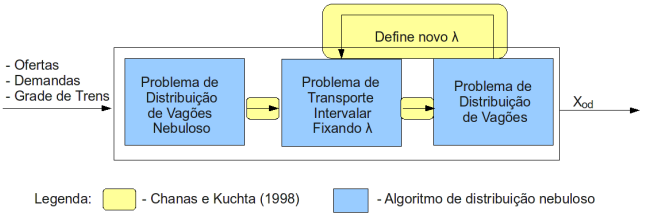


Figura 1. Algoritmo Nebuloso de Distribuição de Vagões

Como mostra a Fig. 1, o algoritmo é dividido em 3 etapas principais:

Problema de distribuição de vagões nebuloso: este problema é similar ao problema de distribuição [2], exceto que, nesta etapa são definidos os números nebulosos que representam a imprecisão na quantidade de vagões para cada demanda de transporte;

Problema de transporte intervalar para λ fixo: neste etapa é fixado um λ entre 0 e 1. O λ define o λ-corte. Para toda demanda, que é representada por um número nebuloso, tem-se um intervalo de números reais resultante do cálculo do λ-corte para um λ fixo. O problema de transporte intervalar é criado a partir do maior e menor número inteiro que estiverem contidos neste intervalo.

Problema de distribuição de vagões: Nesta etapa cria-se um problema de distribuição [2] apartir do problema de transporte intervalar. O problema de distribuição é resolvido e verifica-se se a solução é ótima ou infactível. Se a solução não é ótima nem infactível, o valor de λ é alterado e um novo problema de transporte intervalar é criado. Estas etapas se repetem até que ou a solução ótima, ou ouma solução infactível seja encontrada.

A partir da Fig. 1 verifica-se que resolver o algoritmo de distribuição nebuloso consiste em resolver várias instâncias do algoritmo de distribuição [2]. O tempo de processamento do algoritmo de dis-

tribuição nebuloso é equivalente ao tempo de processamento do algoritmo de distribuição multiplicado pelo número de vezes que o λ foi atualizado. O algoritmo de distribuição tem um tempo de processamento muito pequeno, devido a estrutura do modelo de distribuição criado. Apesar do algoritmo de distribuição nebuloso ser mais lento, este algoritmo ainda obtém resultados num tempo de processamento satisfatório devido a rapidez apresentada na resolução do problema de distribuição.

Na próxima seção mostramos os resultados e experimentos realizados para ilustrar a utilização do algoritmo de distribuição de vagões nebuloso em diferentes situações e avaliar a qualidade das soluções obtidas.

3. Resultados

Esta seção apresenta e discute os resultados obtidos pelo algoritmo de distribuição de vagões nebuloso. Os resultados obtidos são comparados algoritmo de distribuição de vagões [2].

As instâncias utilizadas neste experimento foram inspiradas em situações reais que ocorrem em ferrovias. Com o intuito de analisar o algoritmo nebuloso foram feitos 3 experimentos. Os experimentos realizados são descritos nas próximas subseções.

3.1. Experimento 1: Confiabilidade dos clientes

Este experimento foi realizado com o objetivo de ilustrar a aplicação do algoritmo de distribuição nebulosa em situações em que a confiabilidade da demanda de transporte é imprecisa. De fato, na prática existe imprecisão nas demandas de uma ferrovia. Geralmente, existem clientes que fazem o pedido de uma quantidade de vagões superior ao que realmente necessitam. Neste experimento foram considerados que os pedidos são classificados de acordo com o confiabilidade da demanda como pouco confiável ou muito confiável.

Neste experimento foi resolvido o problema mostrado na Fig. 3 (Problema 1). As funções de pertinência para as demandas do problema estão ilustradas na Fig. 2.

Os resultados obtidos pelo algoritmo de distribuição nebuloso são comparados com os resultados do algoritmo de distribuição [2].

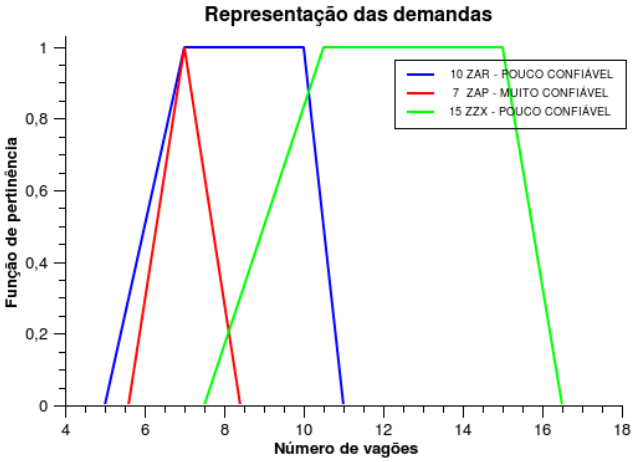


Figura 2. Representação das funções de pertinência das demandas do problema 1

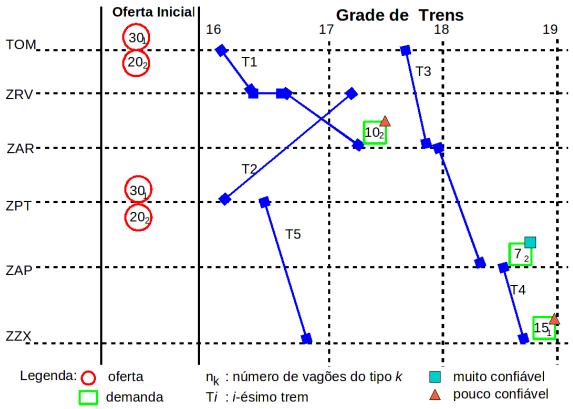


Figura 3. Problema 1

	Distribuição[2]	Nebuloso
Custo	139.21	104.44
Tempo(ms)	4	26
Origem-Destino	Vagões distribuídos	
TOM-ZAR	10	7
TOM-ZAP	7	7
ZPT-ZZX	15	10

Tabela 1. Resultados do algoritmo de distribuição[2] e do algoritmo de distribuição nebuloso para o problema 1

A partir da análise dos resultados da tabela 1 pode-se verificar que o algoritmo de distribuição nebuloso obteve uma solução de menor custo. Este resultado foi possível, pois o algoritmo considera a confiabilidade das demandas.

Assume-se no problema 1 as demandas 10

ZAR e 15 ZZX são pouco confiáveis. Neste caso o algoritmo nebuloso encontrou uma solução de compromisso considerando a confiabilidade destas demandas e o custo de distribuição. O Resultado envia menos vagões, obtendo assim um menor custo de distribuição.

A solução obtida pelo algoritmo nebuloso é mais robusta que a do algoritmo de distribuição[2] considerando que os vagões que deixaram de ser distribuidos não seriam utilizados e iriam gerar um custo maior.

3.2. Experimento 2: Previsibilidade no horizonte de tempo

Neste experimento as demandas tem uma função de pertinência que varia de acordo com seus prazos no horizonte de tempo.

	Distribuição[2]	Nebuloso
Custo	139.21	118.35
Tempo(ms)	4	77
Origem-Destino	Vagões distribuídos	
TOM-ZAR	10	9
TOM-ZAP	7	6
ZPT-ZZX	15	12

Tabela 2. Resultados do algoritmo de distribuição[2] e do algoritmo de distribuição nebuloso para o problema 1

A análise dos resultados da tabela 2 mostra que o algoritmo de distribuição nebuloso obteve uma solução de menor custo. Este resultado foi possível porque o algoritmo considera a imprecisão das demandas no horizonte de tempo.

3.3. Experimento 3: Instância inspirada em uma grade real

Este experimento considera um problema baseado em uma grade de trens real, com 30 trens, 30 tipos de vagões e horizonte de tempo de 1 dia. O mesmo problema foi resolvido para diferentes valores de f (fator de imprecisão), tempo de processamento e custo das soluções obtidas conforme tabela abaixo.

f	Custo	Tempo (ms)
0	7472.8	552
0.25	6436.6	542
0.5	3758.2	553
0.75	1369.5	633
1	0	654

Tabela 3. Resultados do algoritmo nebuloso de distribuição para o problema do experimento 3 e diferentes valores de f

A solução obtida pelo algoritmo de distribuição[2] teve custo 7472.8, ou seja, foi exatamente igual a solução obtida pelo algoritmo nebuloso com fator de imprecisão $f = 0$. A partir do resultado pode-se observar que o algoritmo de distribuição clássico é um caso particular do algoritmo nebuloso, para isto, é necessário utilizar a função de pertinência genérica com fator $f = 0$. O algoritmo nebuloso resolve o problema também para situações onde existe imprecisão na demanda de transporte, ou seja, é mais abrangente que o algoritmo[2].

4. Conclusão

Neste trabalho foi apresentado um modelo e um algoritmo para distribuição de vagões vazios com imprecisão na demanda. A imprecisão foi modelada utilizando números nebulosos. O algoritmo fornece como resultado a quantidade e tipos de vagões que serão movimentados entre os pares origem-destino da ferrovia, considerando imprecisão nas demandas. Os resultados mostram que o algoritmo proposto obteve soluções estrategicamente mais sofisticadas, com menor custo de transporte e maior robustez que algoritmos clássicos de distribuição.

Referências

[1] S. Chanas and D. Kuchta. Fuzzy Integer Transportation Problem. *Fuzzy Sets and Systems*, 98:291–298, 1998.

[2] Joelma Cristina Costa and F. A. C. Gonçalves, Gomide. Algoritmo de Distribuição e Alocação de Vagões em Tempo Real. *IX Simpósio de Automação Inteligente, Brasília, Distrito Federal, Brasil*, Setembro 2009.

[3] P. Lévine and J.C. Pomerol. Railcar Distribution at the French Railways. *IEEE Expert*, October 1990.

Otimização de Modelos BFO com Funções de Laguerre

Jeremias Barbosa Machado , Ricardo J. G. B. Campello , Wagner Caradori do Amaral(Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

{jeremias,wagner}@dca.fee.unicamp.br, campello@icmc.usp.br

Abstract – This paper presents a new approach to optimization of linear dynamic systems models using orthonormal basis functions with Laguerre functions. This approach propose a method to select the poles of the proposed model. This selection is made by optimizing of the model by nonlinear algorithms of optimization, which requires to calculate the gradients of the orthonormal functions outputs with respect to its parameters. These gradients are calculated analytically and provide the exact direction of search in the optimization of functions parameters. In this context, this paper presents the approaches used for the identification and optimization dynamical system models via orthonormal basis functions. A simulation example illustrates the techniques of identification and modeling proposals.

Keywords – Dynamic linear Systems, Orthonormal Basis Functions, Identification, Modeling

1. Introdução

Modelos de sistemas dinâmicos que utilizam base de funções ortonormais - BFO apresentam vantagens em relação a outras abordagens de modelagem. Estes modelos possibilitam a incorporação de conhecimento prévio sobre a dinâmica do sistema [11, 15] o que reduz a ordem da representação do modelo, simplificando os problemas de identificação e controle associados [19, 10]. Outra característica da utilização de modelos BFO em sistemas dinâmicos é a ausência de realimentação da saída, isto é, não existe realimentação de erros de previsão, o que normalmente resulta em modelos mais precisos [14]. Devido à completude da base também é possível aumentar a sua capacidade de representação incrementando o número de funções ortonormais da base [10]. Os modelos BFO são parametrizados por polos e a escolha adequada destes leva a modelos com um número reduzido de termos. A escolha do(s) polo(s) pode ser feita através de informações a priori das características dinâmicas do sistema ou através de métodos de seleção do(s) polo(s). Em trabalhos como [18, 4, 3, 7] apresenta-se métodos analíticos para a seleção de polos para funções de Laguerre e Kautz, contudo, tais soluções são sub-ótimas pois em geral otimiza-se somente um subconjunto dos parâmetros da base. Alguns trabalhos como [1, 17] propõem procedimentos para seleção de polos de modelos BFO através de métodos numéricos. Em [9, 8] é proposta a otimização dos parâmetros de modelos com BFO (Kautz e Funções de Takenaka-Malmquist) através de métodos de otimização não-linear, empregando uma direção de busca dada pelo cálculo analítico dos gradientes das saídas das funções ortonormais.

Neste trabalho propõem-se um um método para o cálculo analítico dos gradientes das funções de Laguerre utilizando a representação em espaço de estados. Tais gradientes, associados a métodos de otimização não-linear permitem a obtenção de modelos mais precisos e que não exijam um conhecimento a priori do sistema a ser modelado. A seguir, na Seção 2., faz-se uma revisão de modelagem com BFO's. Na Seção 3. descreve-se o processo de otimização e o cálculo dos gradientes para os modelos BFO com funções de Laguerre é apresentado na Seção 3.. Na Seção 4. apresenta-se um exemplo que ilustra o desempenho das metodologias propostas. Finalmente, na Seção 5., apresentam-se as conclusões.

2. Modelos BFO

Modelos BFO podem ser genericamente representados em modelos de espaço de estados como [14, 5]:

$$\begin{aligned} \Psi(k+1) &= A_f \Psi(k) + B_f u(k) \\ \hat{y}(k) &= C_f^T \Psi(k) \end{aligned} \quad (1)$$

onde $\Psi(k) = [\psi_1(k) \dots \psi_n(k)]^T$ é o vetor de estados ortonormais (denominação usada por conveniência já que a ortonormalidade é uma propriedade da função e não dos estados) de ordem *n*, sendo estes estados coincidentes com as saídas dos filtros correspondentes às funções ortonormais de ordem equivalente, $u(k)$ a entrada do sistema e $\hat{y}(k)$ é a saída estimada. Este modelo pode ainda incluir uma parcela na parte estática da representação que modela o nível DC da saída do sistema sob estudo.

Duas funções utilizadas com muita frequência na representação através de BFO's são as funções de Laguerre e de Kautz. A base de Laguerre é apropriada para representar sistemas com polos puramente reais ou com parte imaginária de valor reduzido. A base de Kautz é mais apropriada para representar sistemas com dinâmica oscilatória subamortecida por ser parametrizada por polos complexos conjugados [10, 5]. A base de funções ortonormais de Laguerre é caracterizada pela utilização de funções de transferência com apenas um polo real [10]:

$$\Phi_i(z) = \frac{\sqrt{1-p^2}}{z-p} \left(\frac{1-pz}{z-p} \right)^{i-1}, \quad i = 1, 2, \dots \quad (2)$$

onde $p = \{p : p \in \mathbb{R} \text{ e } |p| < 1\}$ é o polo que parametriza as funções ortonormais e i a ordem da função dada. As funções de Laguerre podem ser caracterizadas como modelos FIR com polos fora da origem, sendo o modelo FIR

com polos na origem um caso especial das BFO's [10, 5]. O modelo FIR normalmente requer um número elevado de termos para representar sistemas com dinâmica dominante lenta. Como pode-se notar através da equação (2) os modelos de ordem i podem ser escritos como modelos em cascata de ordem $i-1$. A representação em cascata da função de Laguerre lhe confere a propriedade de recursividade já que a *i*-ésima função pode ser escrita em função da (*i*-1)-ésima. Sendo assim, é possível representar a dinâmica de funções de Laguerre através de uma equação de estados, como apresentado na equação (1). A descrição detalhada do modelo em espaço de estados pode ser encontrada em [13]. Neste trabalho propõem-se uma metodologia para a busca do valor do polo p de Laguerre e dos parâmetros da expansão do modelo.

3. Ajuste dos parâmetros das BFO's

Os parâmetros a serem determinados na modelagem de sistemas dinâmicos lineares por base de funções ortonormais são os polos das funções ortonormais e os coeficientes da expansão da série de funções. Conforme discutido em [16, 2, 5, 13], uma vez selecionados os polos, os termos da expansão da série podem ser determinados através do já bem conhecido método de mínimos quadrados.

A seleção do(s) polo(s) da base de funções ortonormais que representará a dinâmica do sistema normalmente depende de um conhecimento prévio do sistema ou de métodos de otimização que geralmente envolvem algoritmos de busca que minimizem o erro entre a saída do sistema e a saída do modelo estimado como apresentado em [14, 10, 6]. Em [9, 6] é apresentada uma metodologia de otimização dos parâmetros de modelos de Kautz e GOBF com funções de Takenaka-Malmquist através de buscas exatas dadas pelos cálculos dos gradientes da saída do modelo estimado com relação aos parâmetros do modelo. Nesta seção otimiza-se, através de métodos de otimização não-linear [12] em especial empregando o método de Levenberg-Marquardt [14], os parâmetros de um modelo linear BFO com funções de Laguerre apresentando de maneira detalhada as equações para o cálculo dos gradientes necessário ao processo de otimização.

3.1. Estimação e Otimização dos Parâmetros dos modelos BFO's

A estratégia adotada consiste da otimização de ambos o polo p e os termos da expansão da série na matriz C minimizando uma função de custo com relação ao erro quadrático entre a saída do sistema e a saída do modelo:

$$\min_{p\ C^T} J = \frac{1}{2}\sum_{k=0}^{L}(\hat{y}(k)-y(k))^2 \qquad (3)$$

com $\hat{y}(k)$ sendo a saída estimada.

Para se resolver o problema de minimização é necessário calcular o gradiente da função de custo com relação aos parâmetros que se deseja otimizar. Para o cálculo dos gradientes através dos dados coletados do sistema será empregada a técnica de *back-propagation-through-time* [14]. Esta técnica decompõe a dinâmica do sistema sob estudo em uma sequência de representações estáticas, permitindo descrever as derivadas da saída do modelo somente em termos das condições iniciais e dos sinais de entrada, através de uma recursão de k passos no tempo.

Analisando-se individualmente os termos que compõem a equação (3), o gradiente da função de custo com relação aos parâmetros em C é dado por:

$$\begin{aligned}\nabla_C J &= \sum_{k=0}^{L}(y(k)-\hat{y}(k))\nabla_C\hat{y}(k)\\ &= \sum_{k=0}^{L}(y(k)-\hat{y}(k))\Psi(k)\end{aligned}$$

Já o gradiente de J com relação ao parâmetro p é definido por:

$$\begin{aligned}\frac{\partial J}{\partial p} &= \sum_{k=0}^{L}(y(k)-\hat{y}(k))\sum_{i=1}^{n} g_i\frac{\partial\psi_i(k)}{\partial p}\\ &= \sum_{k=0}^{L}(y(k)-\hat{y}(k))C^T\frac{\partial\Psi_i(k)}{\partial p}\end{aligned}$$

Neste trabalho utilizar-se-á o cálculo analítico destes gradientes da função de custo com relação ao(s) parâmetro(s) p e C, fornecendo assim o valor exato do gradiente para a otimização do(s) polo(s) e dos parâmetros da expansão das funções ortonormais através de métodos de otimização não-linear. A seguir é apresentada a solução para o cálculo dos gradiente citados para os modelos BFO com funções de Laguerre.

3.1.1. Gradiente para funções de Laguerre

Para o cálculo do gradiente utilizar-se-á a representação em espaço de estados do modelo de Laguerre como descrito na eq (1). Contudo, uma modificação é necessária já que se deseja conhecer as saídas dos filtros, ou seja, os estados nos instantes k para derivá-las com relação ao polo. Sendo assim, o modelo em espaço de estados para as funções de Laguerre será escrito como:

$$\Psi(k+1) = A_f\Psi(k) + B_f u(k) \qquad (4)$$

onde A_f e B_f são as mesmas definidas na seção 2. para as funções de Laguerre. Desenvolvendo-se o modelo (4) é possível verificar que a solução para $\Psi(k)$ será dada por:

$$\Psi(k) = A_f^k\Psi(0) + \sum_{i=0}^{k-1}A_f^i B_f u(k-1-i) \qquad (5)$$

O gradiente de $\Psi(k)$ com relação ao polo p pode então ser calculado através da derivada da equação (5) com relação ao polo, como apresentado na equação (6):

$$\frac{\partial \Psi(k)}{\partial p} = \frac{\partial A_f^k}{\partial p}\Psi(0) + \sum_{i=0}^{k-1}\left(\frac{\partial A_f^i}{\partial p}B_f + A_f^i\frac{\partial B_f}{\partial p}\right)u(k-1-i) \quad (6)$$

sendo $\frac{\partial A_f^k}{\partial p}$ dado por:

$$\frac{\partial A_f^k}{\partial p} = \sum_{j=1}^{k} A_f^{j-1}\frac{\partial A_f}{\partial p}A_f^{k-j} \quad (7)$$

Das equações (6) e (7) torna-se necessário calcular as derivadas de A_f e B_f com relação ao polo p. O valor destas derivadas é dado por:

$$\frac{\partial A_f}{\partial p} = \begin{bmatrix} dap_{11} & 0 & 0 & \cdots & 0 \\ dap_{21} & dap_{22} & 0 & \cdots & 0 \\ dap_{31} & dap_{32} & dap_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ dap_{n1} & dap_{n2} & dap_{n3} & \cdots & dap_{nn} \end{bmatrix}$$

onde i é igual à linha e j é igual à coluna do elemento dap_{ij}, sendo este dado por:

- No casos em que $j > i$:
  . $dap_{ij} = 0$;
- Com $i = j$
  . $dap_{ij} = 1$;
- Com $j < i$

  . $dap_{ij} = (-1)(i-j-1)(-p)^{(i-j-2)} * \ldots$
$\ldots * (1-p^2) + (-p)^{(i-j-1)}(-2p)$, tal que $i \leq n$
$(i-j-1) \geq 0$;

Já para a matriz B_f, tem-se que a sua derivac com relação ao polo p é dada por:

$$\frac{\partial B_f}{\partial p} = \begin{bmatrix} dbp_{11} & dbp_{21} & dbp_{31} & \cdots & dbp_{n1} \end{bmatrix}^T$$

com seus elementos sendo:

$$dbp_{i1} = \frac{-p}{\sqrt{1-p^2}}(-p)^{(i-1)} + \frac{1-p^2}{-p}(i-1)(-1)(-p)^{(i-2)};$$

onde i é a linha do elemento dbp.

Desta maneira calcula-se analiticamente o gradiente das saídas das funções de Laguerre com relação ao parâmetro p utilizando-se a técnica de retropropagação através do tempo, como apresentada na seção 3.1.

4. Resultados Experimentais

Esta seção exemplifica a aplicação do modelo e a implementação do processo de otimização propostos neste trabalho. Para ilustrar a aplicação de modelos de Laguerre, considere o sistema linear estável de segunda ordem com polo real de multiplicidade 2.

$$F(z) = \frac{0,08037}{z^2 - 1,433z + 0,5134}, \qquad \text{com } t_s = 0,1seg \quad (8)$$

cujos polos estão em $z = 0,7165$.

Este sistema tem como resposta ao degrau uma saída super-amortecida, o que, neste caso justifica a abordagem de Laguerre resultando em modelos com número reduzido de funções. Partindo-se do pressuposto que em sistemas reais não se conhece a ordem do sistema sob análise, os testes para se obter o modelo foram iniciados com uma função de Laguerre. O sinal de entrada $u(k)$ é aleatório com distribuição uniforme entre -1 e 1. A tabela 1 apresenta os resultados comparativos entre os modelos otimizados.

Analisando-se os resultados apresentados na tabela 1 verifica-se que os modelos com somente um polo não foram capazes de modelar o sistema sob estudo. Os modelos com duas funções modelaram perfeitamente com o polo das funções convergindo para um polo próximo ao polo exato do sistema. Já com três funções o polo ficou um pouco mais distante e o modelo com uma precisão semelhante. A figura 1 mostra o resultado do sistema estimado antes e depois do processo de otimização com duas funções de Laguerre.

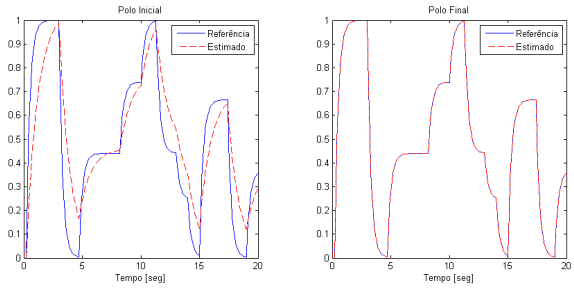


Figura 1. Comparação entre os polos inicial e final

5. Conclusões

Este artigo apresentou uma nova proposta para a otimização de modelos de sistemas dinâmicos através de BFO com funções de Laguerre na forma de espaço de estados. Foram apresentadas expressões analíticas para os gradientes das funções de Laguerre com relação aos parâmetros dos modelo. Utilizando-se estas expressões é possível se obter direções de busca exatas que são utilizadas na otimização dos modelos. A metodologia descrita apresenta vantagens sobre outros modelos pois o cálculo dos gradientes é feito em batelada a partir de sinais de entrada e saída do sistema sob análise e o sistema é identificado sem conhecimento a priori do mesmo. Resultados de simulação ilustram a eficácia da metodologia proposta permitindo-se obter modelos BFO com uma excelente

Tabela 1. Resultados dos modelos de Laguerre para função de 2^a ordem.

N. de Funções	Iter.	Polo Inicial	Polo Final	Avaliação Função de Custo
1	8	0,1000	0,8664	$0,5808$
1	3	0,9000	0,7360	0.5808
2	7	0,1000	0,7160	$3,1111.10^{-15}$
2	5	0,5000	0,7160	$9,3224.10^{-18}$
3	10	0,5000	0,7124	$6,8997.10^{-8}$

aproximação com polos convergindo para os polos reais do sistema modelado. Extensões deste trabalho estão sob investigação e estão sendo aplicadas na modelagem de sistemas não-lineares e a outras funções ortonormais.

Referências

[1] A.C. Den Brinker B.E. Sarroukh, S.J.L. Van Eijndhoven. An iterative solution for the optimal poles in a kautz series. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 6:3949–3952, 2001.

[2] R. J. G. B. Campello. *Arquiteturas e Metodologias para Modelagem e Controle de Sistemas Complexos Utilizando Ferramentas Clássicas e Modernas*. Tese de Doutorado, DCA/FEEC/UNICAMP, 2002.

[3] R. J. G. B. Campello, W. C. Amaral, and G. Favier. A note on the optimal expansion of Volterra models using Laguerre functions. *Automatica*, 42:689–693, 2006.

[4] R. J. G. B. Campello, G. Favier, and W. C. Amaral. Optimal expansions of discrete-time Volterra models using Laguerre functions. *Automatica*, 40:815–822, 2004.

[5] R. J. G. B. Campello, G. H. C. Oliveira, and W. C. Amaral. Identificação e Controle de Processos via Desenvolvimentos em Séries Ortonormais: Partes A (Identificação) e B (Controle). *Controle & Automação*, 18(3):298–332, 2007.

[6] A. da Rosa. *Identificação de Sistemas Não-Lineares Usando Modelos de Volterra Baseados em Funções Ortonormais de Kautz e Generalizadas*. Tese de Doutorado, DCA/FEEC/UNICAMP, 2009.

[7] A. da Rosa, R. J. G. B. Campello, and W. C. Amaral. Choice of free parameters in expansions of discrete-time Volterra models using Kautz functions. *Automatica*, 43(6), 2007.

[8] A. da Rosa, R. J. G. B. Campello, and W. C. Amaral. Exact Search Directions for Optimization of Linear and Nonlinear Models Based on Generalized Orthonormal Functions. *IEEE Transactions on Automatic Control*, 54(12):2757–2772, 2009.

[9] A. da Rosa, R.J.G.B. Campello, and W.C. Amaral. Cálculo de direções de busca exatas para otimização de filtros de laguerre e de kautz. In *XVII Congresso Brasileiro de Automática*, volume 1, 2008.

[10] P. S. C. Heuberger, P. M. J. Van den Hof, and B. Wahlberg. *Modelling and Identification with Rational Orthogonal Basis Functions*. Springer, 2005.

[11] Van den Hof P. M. J. e Bosgra O. H. Heuberger, P. S. C. A generalized orthonormal basis for linear dynamical systems. *IEEE Transactions on Automatic Control*, 40(3):451–465, 1999.

[12] David G. Luenberger. *Linear and Nonlinear Programming*. Springer, 2nd edition, 2003.

[13] Jeremias Barbosa Machado. Modelagem e controle preditivo utilizando multimodelos. Master's thesis, Universidade Estadual de Campinas, Campinas, Brasil, Fevereiro 2007.

[14] O. Nelles. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer-Verlag, 2001.

[15] F. Ninness, B. e Gustafsson. A unifying construction of orthonormal bases for system identification. *IEEE Transactions on Automatic Control*, 42(4):515–521, 1997.

[16] G. H. C. Oliveira. *Controle Preditivo para Processos com incertezas estruturadas baseado em séries de funções ortonormais*. PhD thesis, Universidade Estadual de Campinas, Campinas, 1997.

[17] Shah S.L. Partwardhan, S.C. From data to diagnosis and control using generalized orthonormal basis filters, Part I: Development of state observers. *Journal of Process Control*, 15(7):819–835, 2005.

[18] N. Tanguy, R. Morvan, P. Vilbé, and L. C. Calvez. Pertinent choice of parameters for discrete Kautz approximation. *IEEE Trans. on Automatic Control*, 47:783–787, 2002.

[19] Heuberger P. S. C. e Bokor J. Van den Hof, P. M. J. System identification with generalized orthonormal basis functions. *Automatica*, 31(12):1821–1834, 1995.

Um Breve Estudo sobre Análise de Componentes Esparsos

Everton Z. Nadalin[1], Ricardo Suyama[2], Romis Attux[1]

1 - Departamento de Engenharia de Computação e Automação Industrial (DCA)
2 – Departamento de Microonda e Óptica (DMO)

Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

{nadalin, attux}@dca.fee.unicamp.br, rsuyama@dmo.fee.unicamp.br

Abstract – In this work, we present a discussion concerning some fundamental aspects of sparse component analysis (SCA), a method that has been increasingly employed to solve blind source separation (BSS) problems, especially when there are more sources than sensors. In addition to providing an overview on BSS and SCA, we try to point out relevant paths for future research and analyze the idea of compressive sensing, which seems to raise interesting perspectives for practical application.

Keywords – unsupervised signal processing, blind source separation, sparse component analysis, compressive sensing.

1. Introdução

Devido ao barateamento de processadores e à conseqüente revolução que os sistemas embarcados vêm causando, novas técnicas de processamento de sinais têm sido requisitadas. Sempre buscando novos desafios, a teoria de processamento não-supervisionado tem cada vez mais incorporado técnicas e soluções capazes de fazer efetivo uso de informação *a priori* sobre os sinais que se deseja estimar. Nesse contexto, uma técnica que vem ganhando espaço é a análise de componentes esparsos (SCA – *sparse component analysis*), a qual encontra aplicação em áreas como separação cega de fontes (BSS – *blind source separation*) [1] e *compressive sensing* [2].

Intuitivamente, a idéia de esparsidade de um sinal é relativamente simples: um sinal esparso é fundamentalmente aquele em que predominam, no domínio temporal ou em algum outro domínio relevante, valores nulos ou próximos de zero, sendo que poucos componentes possuem a maior parte da energia do sinal. Porém, quando tentamos quantificar essa idéia, surgem diversas interpretações.

Este artigo pretende mostrar algumas formas existentes de quantificação da idéia de esparsidade em sinais esparsos, bem como fazer uma discussão sobre a aplicação de SCA em BSS.

O artigo está dividido da seguinte forma: na seção 2, é feita uma descrição da idéia de SCA; na seção 3, discute-se a aplicação de SCA em BSS; enquanto a seção 4 contém uma discussão final.

2. Quantificando a esparsidade de um sinal

O conceito de esparsidade nos remete sinais que possuem uma grande quantidade de regiões com valores nulos ou quase nulos em algum domínio, ou seja, sinais em que toda a informação está concentrada em uma quantidade pequena de valores que representam aquilo que se analisa. Uma maneira, neste caso, de se quantificar o grau de esparsidade é contar quantas amostras ou quantos coeficientes possuem valores não-nulos.

Este tipo de medição evoca a chamada norma ℓ_0 [3], porém, ela falha na maioria dos casos práticos pelo simples fato de que quase nunca os coeficientes possuem valor nulo, havendo em geral ruído ou um valor residual. Para fugir disto, alguns trabalhos adotaram a norma $\ell_{0\varepsilon}$, que leva em conta um valor ε residual para definir uma espécie de limiar de "nulidade".

Outro tipo de critério de otimização é a chamada norma ℓ_1 [3], definida pela soma dos valores absolutos das amostras do sinal. Uma vantagem desta abordagem é que o uso dessa norma permite, em alguns casos, que se lance mão de metodologias de otimização convexa, as quais, muitas vezes, permitem que certos passos de um procedimento de separação se vinculem a soluções em forma fechada [4].

É importante frisar que, além do uso direto de critérios de otimização que procuram quantificar a esparsidade do sinal, existem outras possibilidades, como as discutidas no trabalho de Hurley et al. [3]. Uma dessas possibilidades, aliás, relaciona-se ao o emprego adicional de

uma estatística de ordem superior amplamente usada em análise de componentes independentes (ICA, - *independent component analysis*), a curtose.

Ainda em [3], é realizada uma discussão sobre a qualidade de cada critério no âmbito de um conceito mais amplo de esparsidade. No artigo, seis são os atributos considerados para se medir a esparsidade de um sinal, quatro deles derivados das leis de Dalton relacionadas à distribuição de renda [5] e outros dois descritos em [6]. Estes seis atributos são descritos abaixo.

1 – quanto mais a energia for distribuida entre os coeficientes, menor a esparsidade do sinal.

2 – multiplicar cada coeficiente por uma mesma constante não altera a esparsidade.

3 – o acréscimo de uma mesma constante a cada coeficiente diminui a esparsidade do sinal.

4 – a união de dois sinais idênticos não altera a esparsidade.

5 – conforme um determinado coeficiente vai ficando com cada vez mais energia, a esparsidade do sinal vai aumentando.

6 – a inclusão de coeficientes de valor nulo o sinal aumenta a esparsidade.

Apesar de estes atributos parecerem intuitivamente bem consistentes, Hurley et al. mostram que, das 15 medidas de esparsidade mais comumente utilizadas, somente o índice de Gini [7] obedece a todos os seis. Porém, é importante perceber que nem todos estes critérios precisam ser levados em consideração quando tentamos resolver os problemas de separação de fontes e compressão de sinais, os mais comuns em que se utilizam técnicas de SCA. Para que tal fato fique mais claro, estes dois problemas serão explicados no próximo item.

3. Separação cega de fontes

Nos últimos anos, o estudo do problema de separação cega de fontes (BSS) se consolidou como um dos pilares da teoria de processamento não-supervisionado de sinais. Um bom exemplo de problema de BSS é o comumente chamado de *cocktail party problem* [8], que pode ser exemplificado da seguinte forma: existe uma sala com *n* microfones e *m* pessoas falando, e, tendo acesso apenas aos dados dos microfones, queremos encontrar o sinal produzido por cada pessoa separadamente.

Matematicamente, sendo $\mathbf{s}(n)$ o vetor dos sinais das fontes de informação (no exemplo acima, esse vetor seria composto pelos sinais de voz de todos os presentes) e $\mathbf{x}(n)$ o vetor de sinais captados pelos sensores (também chamado de vetor de observações), podemos, supondo que a mistura dos sinais é linear e não envolve espalhamento temporal, escrever

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n)$$

onde $\mathbf{A}$ é chamada de matriz de mistura do sinal, que, tipicamente, é suposta quadrada e inversível (voltaremos a essa hipótese mais adiante). Sendo assim, o problema acima descrito corresponde a estimar as fontes tendo acesso somente às observações. Para realizar esse processo de separação, adota-se, tipicamente, um sistema separador com estrutura análoga à da mistura, ou seja:

$$\mathbf{y}(n) = \mathbf{W}\mathbf{x}(n)$$

sendo $\mathbf{W}$ a matriz de separação. Idealmente, o que se busca é obter um vetor $\mathbf{y}(n)$ que seja igual a $\mathbf{s}(n)$ a menos de fatores de escala e de uma permutação. Portanto, a matriz de separação ideal seria:

$$\mathbf{W} = \mathbf{P}.\mathbf{D}.\mathbf{A}^{-1}$$

onde $\mathbf{P}$ é uma matriz de permutação e $\mathbf{D}$ uma matriz diagonal.

O modelo acima descrito é, sem dúvida, o mais estudado em BSS, e, sob a hipótese de que as fontes são mutuamente independentes, o problema dele originado vem sendo resolvido com sucesso por meio de diversas técnicas de ICA. No entanto, tal modelo se baseia em algumas hipóteses restritivas que podem não ser válidas em determinadas aplicações práticas. Um exemplo é o fato de que a matriz $\mathbf{A}$ é considerada quadrada e inversível (ou, eventualmente, de posto completo), o que nem sempre é razoável, pois pode haver, por exemplo, menos sensores que fontes.

4. SCA aplicada à separação de fontes

Em boa parte dos problemas práticos não é possível garantir que haja um número de sensores ao menos igual ao número de fontes

porque número de fontes não é conhecido. Neste caso, pode não ser possível aplicar um método para obter um **W** eficiente pelo simples fato de a matriz de mistura não ser inversível: tem-se, portanto, um problema indeterminado. Trabalhos mostram que, em algumas situações em que o sistema é indeterminado [1,9,10,11] e os sinais são esparsos, o fato de que nem sempre todas as fontes estarão ativas ao mesmo tempo traz interessantes perspectivas para a aplicação desta ferramenta em problemas de BSS. Isto se deve ao fato de que, nestas situações, um sistema indeterminado ser localmente determinado, sendo possível a identificação da matriz de mistura e em alguns casos até a separação das fontes. Este tipo de abordagem é comumente chamada de análise de componente esparsos (SCA).

Os primeiros trabalhos utilizando SCA em BSS [1] assumiram que no máximo uma fonte estivesse ativa em cada amostra dos sinais de mistura. Desta forma, não existindo sobreposição de fontes, é possível separá-las de forma integral, mesmo que haja mais fontes do que misturas, conforme mostrado em [9]. Neste caso, métodos como o mascaramento binário [10] já são suficientes para a separação das fontes. Em [9] é dito que, quando é exigido em cada instante que apenas uma fonte esteja ativa, o problema acaba demandando, na verdade, duas restrições. Além da esparsidade das fontes, é necessário também que exista uma ortogonalidade disjunta entre elas na mistura.

Já no artigo [11], Aissa-El-Bay et al. extendem o resultado também para fontes não-disjuntas, em que nem sempre é necessário uma fonte ativa por vez. Outros trabalhos seguem a mesma linha, considerando que o número de fontes ativas seja, na média, unitário [12], ou que, para cada fonte, exista pelo menos um instante na mistura em que somente ela esteja ativa [13]. Porém, essas garantias só se justificam em uma mistura sem ruído.

Alguns trabalhos conseguem extrapolar os resultados para situações em quem em cada instante o número de fontes não ultrapassa o número de misturas [14], porém, neste caso o processo se restringe à estimação da matriz de mistura: as fontes não chegam a ser separadas.

O trabalho [15] mostra que se, além de esparsos, os sinais forem independentes entre si (o que é verdade em boa parte dos sinais práticos), é possível estimar a matriz de mistura.

4.1 Estimação do número de fontes

Um ponto que deve ser levado em consideração é que, a partir do momento em que se trabalha com sistemas indeterminados, tem-se que tomar uma decisão sobre o conhecimento *a priori* ou não do número de fontes.

Praticamente todos os trabalhos citados consideram que o número de fontes é conhecido, porém, na maioria dos casos reais, isto não é verdade, principalmente por dois motivos: não se sabe quantas fontes de ruído existem e não é possível precisar o que o algoritmo vai considerar como sendo ou não fonte.

Os trabalhos [16,17] fazem a estimação do número de fontes a partir de uma estimação automática do número de *clusters* mais adequado para representar os sinais de mistura.

4.2 O que ainda não foi feito

Apesar do avanço que vem ocorrendo nos últimos anos, todas as soluções mostradas dependem de condições específicas das misturas, além da esparsidade dos sinais das fontes. Isso evoca a condição clássica para ICA: quando garantimos que os sinais das fontes são independentes entre si e a matriz de mistura tem posto completo em colunas, é possível realizar uma separação perfeita.

Fica agora a pergunta: sabendo que uma mistura é composta por sinais esparsos, se procurarmos otimizar algum critério que determine as componentes mais esparsas possíveis que compõe a mistura, em quais condições estaremos automaticamente separando as fontes?

Em outras palavras, tomando como base o modelo de mistura de ICA

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n)$$

se encontrarmos um vetor $\mathbf{q}(n)$ a partir de um critério de maximização de esparsidade que obedeça a equação

$$\mathbf{x}(n) = \mathbf{G}\mathbf{q}(n)$$

ele será automaticamente uma estimativa de $\mathbf{s}(n)$?

5. Conclusões

O problema de separação cega de fontes (BSS) é, sem dúvida, um dos pilares da moderna teoria de processamento não-supervisionado. Embora, em certos contextos práticos, tal problema possa ser resolvido de modo satisfatório para meio da análise de componentes independentes (ICA), há casos em que outros tipos de informação *a priori* sobre as fontes são necessários.

Uma possibilidade muito interessante nesse sentido é o uso da análise de componentes esparsos (SCA). Neste trabalho, buscamos apresentar a idéia geral de SCA, bem como alguns aspectos de sua aplicação no problema de BSS que julgamos relevantes e promissores.

Referências

[1] Bofill, P. and Zibulevsky, M., "Blind separation of more sources than mixtures using sparsity of their short-time Fourier transform." In: *Proceedings of the ICA2000*, pp. 87-92.

[2] Candès, E. J., Wakin, M. B., "An Introdution to Compressive Sampling", *IEEE Signal Processing Magazine*, vol. 25, pp. 21-30, Março 2008.

[3] Hurley, N., Rickard, S., "Comparing Measures of Sparsity". In: *IEEE Workshop on Machine Learning for Signal Processing*, Cancún, México, 2008.

[4] Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Nova York, 2004.

[5] Dalton, H., "The measurement of the inequity of incomes", *Economic Journal*, vol. 30, pp. 348-361, 1920.

[6] Rickard, S., Fallon. M., "The Gini index of speech". In: *Conference on Information Sciences and Systems*, Princeton, EUA, 2004.

[7] Gini, C., "Measurement of inequity of incomes", *Economic Journal*, vol. 31, pp. 124-126, 1921.

[8] Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons, Nova York, 2001.

[9] Rickard, S., "Sparse sources are separated sources". In: *Proceedings of the 16th Annual European Signal Processing Conference*, Florence, Italy, 2006.

[10] Yilmaz, O.; Rickard, S., "Blind separation of speech mixtures via time-frequency masking," *Signal Processing, IEEE Transactions on* , vol.52, no.7, pp. 1830-1847, Julho 2004.

[11] Aissa-El-Bey, A.; Linh-Trung, N.; Abed-Meraim, K.; Belouchrani, A.; Grenier, Y., "Underdetermined Blind Separation of Nondisjoint Sources in the Time-Frequency Domain," *Signal Processing, IEEE Transactions on*, vol.55, no.3, pp.897-907, Março 2007.

[12] Georgiev, P.; Theis, F.; Cichocki, A., "Sparse component analysis and blind source separation of underdetermined mixtures," *Neural Networks, IEEE Transactions on*, vol.16, no.4, pp.992-996, Julho 2005.

[13] Kim, S.; Yoo, C.D., "Underdetermined Blind Source Separation Based on Subspace Representation," *Signal Processing, IEEE Transactions on*, vol.57, no.7, pp.2604-2614, Julho 2009.

[14] Movahedi Naini, F., Hosein Mohimani, G., Babaie-Zadeh, M., and Jutten, C. "Estimating the mixing matrix in Sparse Component Analysis (SCA) based on partial k-dimensional subspace clustering." *Neurocomput.* 71, 10-12 (Jun. 2008), 2330-2343.

[15] Nadalin, E. Z., Suyama, R., and Attux, R. "An ICA-Based Method for Blind Source Separation in Sparse Domains." In *Proceedings of the 8th international Conference on independent Component Analysis and Signal Separation* (Paraty, Março 15 - 18, 2009).

[16] Arberet, S., Gribonval, R., Bimbot, F.: "A Roubust Method to Count and Locate Audio Sources in a Stereophonic Linear Instantaneous Mixture." In: In *Proceedings of the 6th international Conference on independent Component Analysis and Signal Separation*. EUA (2006).

[17] Nadalin, E. Z., Suyama, R., Attux, R., "Estimating the number of audio sources in a stereophonic instantaneous mixture." In: *7o Congresso de Engenharia de Áudio - AES2009*, 2009, São Paulo.

Métodos para Separação de Misturas com Não-Linearidade Posterior Baseados em Inteligência Computacional

Filipe O. Pereira[1,2], Leonardo T. Duarte[1], Ricardo Suyama[1,3], Romis Attux[1,2]

1 – Laboratório de Processamento de Sinais para Comunicações (DSPCom)
2 – Departamento de Engenharia de Computação e Automação Industrial (DCA)
3 – Departamento de Microonda e Óptica (DMO)

Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6101, CEP 13083-970 – Campinas, SP, Brasil

{filipe, attux}@dca.fee.unicamp.br; {leonardo.tomazeli.duarte}@gmail.com;
{rsuyama}@dmo.fee.unicamp.br

Abstract – This work is concerned with the problem of blind separation of post-nonlinear mixtures. After a brief exposition of the problem of blind source separation in its linear and nonlinear versions, we discuss the elements of our research on the subject and the aspects that shall be investigated in the final stages of our research project.

Keywords – blind source separation, post-nonlinear mixtures, bio-inspired computing.

1. Introdução

O problema de separação cega de fontes (*blind source separation* – BSS) é definido a partir da ideia de recuperar, usando o mínimo de informação *a priori* possível, um conjunto de sinais – denominados fontes – a partir de misturas dos mesmos. Ao longo das décadas de 1980 e 1990, o problema de BSS foi tratado fundamentalmente em sua versão linear, o que deu origem a um notável corpo de resultados teóricos de relevo. No entanto, o interesse por métodos para a realização de BSS não-linear [1] tem crescido nos últimos anos, o que se justifica, por exemplo, pela contínua busca por novos e mais complexos domínios de aplicação [2].

Em BSS não-linear, merece destaque o modelo de mistura com não-linearidade posterior (PNL – *post-nonlinear*) [3], o qual, além de representativo, é interessante por permitir o tratamento do problema de BSS via análise de componentes independentes (ICA-*independent component analysis*). Para que se possa resolver o problema PNL de modo não-supervisionado, é essencial que o sistema separador seja composto de aproximadores monotônicos. Nesse caso, pode-se adotar uma metodologia baseada num contraste de informação mútua, havendo, para tanto, necessidade de realizar convenientemente a estimação da entropia das saídas do misturador.

Tendo em vista essa ideia, Duarte et al. [4][5] estabeleceram uma solução composta de um aproximador monotônico polinomial *ad hoc*, de um processo de estimação de entropia baseado em estatísticas de ordem e num método de otimização bio-inspirado, a rede imunológica artificial opt-aiNet [6]. Em nosso trabalho de mestrado, conforme será exposto neste artigo, pretendemos estender o trabalho de Duarte et al. em dois sentidos: ampliando o leque de ferramentas de otimização por meio da adoção de um sistema imunológico artificial mais simples e de um algoritmo de enxame de partículas e através de adoção de uma gama de estruturas monotônicas mais ampla.

2. Separação de Misturas com Não-Linearidade Posterior Através de ICA

A realização de separação cega de fontes (BSS) via análise de componentes independentes (ICA) se baseia na ideia de estimar, de modo não supervisionado, um conjunto de sinais de interesse, supostos mutuamente independentes e não-gaussianos, a partir de misturas dos mesmos. Seja $\mathbf{s}(t) = [s_1(t), \ldots, s_N(t)]^T$ o vetor de sinais das fontes e $\mathbf{x}(t) = [x_1(t), \ldots, x_N(y)]^T$ o vetor de misturas, ambos, *ex hypothesi*, de mesma dimensão. No caso de misturas instantâneas e lineares – o mais usual da literatura [1] –, matematicamente, as misturas são combinações lineares das fontes e podem ser representadas na forma matricial:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \qquad (1)$$

onde **A** denota a matriz de mistura. Neste caso, uma possibilidade natural é realizar a separação multiplicando o vetor por uma matriz de separação **W**:

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t) \quad (2)$$

A aplicação de ICA ao problema de separação se liga à idéia de escolher **W** de modo que os elementos de $\mathbf{y}(t)$ sejam estatisticamente independentes [1]. Quando é estruturalmente possível inverter a mistura, isso leva à recuperação das fontes a menos de ambigüidades de permutação e fator de escala [7].

A extensão para o caso *post-nonlinear*, ilustrado na Fig. 1, leva ao seguinte modelo de mistura [5]:

$$\mathbf{x}(t) = \boldsymbol{f}(\boldsymbol{A}\boldsymbol{s}(t)) \quad (3)$$

onde $\mathbf{f}(.) = [f_1(.), \ldots, f_N(.)]^T$ corresponde a um conjunto de não-linearidades inversíveis e sem memória. A matriz **A** também deve ser inversível para que a separação seja viável. Um candidato natural ao sistema separador nesse caso é:

$$\mathbf{y}(\iota) = \mathbf{W}\mathbf{g}(\boldsymbol{x}(\iota)) \quad (4)$$

onde $\mathbf{g}(.) = [g_1(.), \ldots, g_N(.)]^T$ são funções não-lineares que devem ser corretamente ajustadas para "anular o efeito"de **f**(.), ou seja, fazer com que a cascata de funções $g_i(.)$ e $f_i(.)$, para i = *1,...,N*, sempre seja uma função linear.

Diante desses modelos, o processo de separação passa a depender de dois aspectos fundamentais: a escolha de um critério que permita quantificar o grau de independência entre as saídas do separador e de um método de parametrização das funções não-lineares **g**(.).

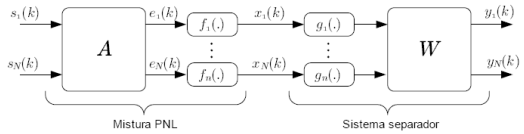


Figura 1. Sistema com Não-Linearidade Posterior

Duarte et al. [5] adotaram a informação mútua (estimada com a ajuda de estatísticas de ordem) como critério de separação e uma parametrização de g(.) baseada em polinômios com restrição de monotonicidade, e esse caminho foi seguido por nós em [8]. Vejamos, pois, o problema de otimização resultante em mais detalhe.

3. Função Custo Baseada na Informação Mútua e Estimação da Entropia Através de Estatísticas de Ordem

A informação mútua, contraste de ICA que temos adotado, é definida como:

$$I(\mathbf{y}) = \sum_i H(y_i) - H(\mathbf{y}) \quad (5)$$

onde $H(\mathbf{y})$ representa a entropia diferencial conjunta da saída do separador **y** e $H(y_i)$ a entropia diferencial de cada um dos elementos desse vetor. Considerando a estrutura de separação mostrada na Figura 1, pode-se expressar a informação mútua das saídas do separador, considerando que as funções $\mathbf{g}_i(.)$ são inversíveis, da seguinte forma:

$$I(\mathbf{y}) = \sum_i H(y_i) \quad H(x) \quad \log|\det \mathbf{W}| \quad E[\log \prod_i |\mathbf{g}_i'(x_i)|] \quad (6)$$

onde $\mathbf{g}_i'$ denota a primeira derivada da i-ésima não-linearidade $\mathbf{g}_i(.)$ do sistema separador. Analisando essa expressão, vemos que a estimação de I(**y**) requer, fundamentalmente, que sejam estimadas as entropias marginais $H(y_i)$, já que H(**x**) é constante e os demais termos são diretamente determinados pelos parâmetros do separador. Em nosso trabalho, a estimação das entropias marginais tem sido realizada por meio de uma metodologia baseada em estatísticas de ordem [9], uma solução robusta e eficiente do ponto de vista computacional. Para uma descrição mais detalhada dessa metodologia, recomendamos ao leitor a leitura de [4][5][8][9].

O problema de minimizar a informação mútua das saídas do separador, ou, equivalentemente, de minimizar a soma de suas entropias marginais engendra uma tarefa de otimização altamente multimodal e para a qual é proibitivo manipular a função custo (e.g. para obter derivadas). Percebemos, então, que se trata de um cenário propício ao uso de ferramentas de computação natural. De fato, inspirados pelo trabalho de Duarte et al. [5], realizamos uma investigação de dois métodos que ainda não haviam sido usados em separação de misturas PNL, uma versão para otimização real do CLONALG [10] e um algoritmo de enxame de partículas [11], sendo a nossa conclusão de que ambos são opções interessantes e plenamente aplicáveis ao problema em questão. Em nosso trabalho de tese, pretendemos realizar novas investigações no sentido de comparar essas ferramentas.

4. Reflexões sobre as Não-Linearidades do Separador

Em [3], trabalho pioneiro na área de separação de misturas PNL, a solução proposta para a parametrização das não-linearidades do separador foi o uso de uma rede do tipo perceptron de múltiplas camadas. Tal solução sempre nos pareceu arriscada, uma vez que o emprego de uma estrutura tão flexível pode levar a soluções espúrias num contexto não-supervisionado. Tendo isso em vista, Duarte et al. [4][5] optaram pelo uso de polinômios com restrição de monotonicidade, uma solução certamente segura. Apesar disso, consideramos que seria uma contribuição interessante buscar novas estruturas monotônicas, preferencialmente estruturas com capacidade de aproximação universal, o que nos levaria a uma proposta genérica para lidar com o problema PNL. Essa investigação está sendo presentemente conduzida por nós e a sua conclusão coincidirá com o término do trabalho de mestrado.

5. Conclusões

Neste trabalho, apresentamos os fundamentos do problema de separação cega de fontes no contexto de misturas com não-linearidade posterior. Ademais, apresentamos as bases de nosso trabalho de mestrado e também algumas possibilidades que temos investigado e que, certamente, formarão parte importante desse esforço.

Agradecimentos

Gostaríamos de agradecer a colaboração do pesquisador Everton Nadalin, que muito tem contribuído para o avanço dessa pesquisa. Também somos gratos à CAPES, que financia este trabalho de mestrado.

Referências

[1] Hyvärinen, A., Karhunen, J., Oja, E., *Independent Component Analysis*, John Wiley & Sons (2001).

[2] Duarte, L.T., *Design of Smart Chemical Sensor Arrays: an Approach Based on Source Separation Methods*. Tese de Doutorado, Institut Polytechnique de Grenoble, Novembro de 2009.

[3] Taleb, A., Jutten, C., "*Source Separation in Postnonlinear Mixtures*", IEEE Trans. Signal Processing, Vol. 47, No. 10, pp. 2807-2820 (1999).

[4] Duarte, L.T., *Um Estudo sobre Separação Cega de Fontes e contribuições ao Caso de Misturas Não-lineares*. Tese de Mestrado, Faculdade de Engenharia Elétrica e de Computação – FEEC – UNICAMP, Campinas, Agosto de 2006.

[5] Duarte, L.T., Suyama, R., Attux, R., Von Zuben, F.J., Romano, J.M.T., "*Blind Source Separation of Post-Nonlinear Mixtures Using Evolutionary Computation and Order Statistics*", Springer Lecture Notes in Computer Science, vol. 3889, pp. 66–73 (2006).

[6] de Castro, L. N., Timmis, J., "*An Artificial Immune Network for Multimodal Function Optimization*", Proceedings of the IEEE Congress on Evolutionary Computation, EUA (2002).

[7] Comon, P., "*Independent Component Analysis, a New Concept?*", Signal Proceessing, Vol. 36, No. 3, pp. 287-314 (1994).

[8] Pereira, F. O. ; Nadalin, E. Z. ; Suyama, R. ; Attux, R. R. de F. "*Análise do Emprego de Ferramentas de Computação Natural no Problema de Separação de Misturas com Não-Linearidade Posterior*". In: XXVII SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES SBrT 2009, 2009, Blumenau. Anais do XXVI Simpósio Brasileiro de Telecomunicações (SBrT'09), 2009.

[9] Pham, D.-T., "*Blind Separation of Instantaneous Mixtures of Sources Based on Order Statistics*", IEEE Trans. Signal Processing, Vol. 48, No. 2, pp. 363-375 (2000).

[10] de Castro, L. N., Von Zuben, F. J., "*Learning and Optimization Using the Clonal Selection Principle*", IEEE Trans. on Evolutionary Computation, Vol. 6, No. 3, pp. 239-251 (2002).

[11] Kennedy, J., Eberhart, R. C., *"Particle Swarm Optimization"*, Proceedings of the IEEE International Conference on Neural Networks, Austrália (1995).