

Prof. Anderson Ribeiro (duarte.andersonr@gmail.com)

1) Heurísticas para detecção de clusters espaciais irregulares

Métodos para detecção e inferência para clusters espaciais são usuais em diversas áreas como em epidemiologia e vigilância sindrômica. Grande parte dos procedimentos em estudo se baseiam na estatística Scan. Um caso particular de aplicação desta estatística pode ser observado através do Scan Circular. Trata-se do método mais difundido para detecção e inferência de clusters. Entretanto, um grande problema desta técnica é o fato de estabelecer uma forma fixa para os clusters detectados em contradição a muitos problemas aplicados que revelam clusters de forma bastante irregular. Heurísticas de otimização multi-objetivo são uma importante ferramenta para a detecção de clusters de forma irregular. A proposta deste trabalho é implementar técnicas heurísticas de otimização ainda não utilizadas que sejam capazes de se ajustar bem ao procedimento de detecção de clusters irregulares.

2) Funções de aproveitamento para detecção e inferência em clusters espaciais irregulares

O procedimento inferencial nas técnicas de detecção de clusters é dependente da distribuição associada à estatística de teste em uso. Quando utilizando algoritmos multi-objetivo para o procedimento de detecção de clusters irregulares, o procedimento inferencial se torna mais complexo. Os procedimentos usuais trabalham com problemas de otimização bi-objetivo e utilizam no procedimento inferencial técnicas de funções de aproveitamento em Pareto. Problemas de otimização com mais que dois objetivos poderiam se tornar mais eficientes, entretanto as funções de aproveitamento ainda não estão bem estabelecidas para aplicação neste problema específico com mais que duas dimensões. A proposta deste trabalho é ajustar a técnica das funções de aproveitamento para mais que duas dimensões no estudo de problemas para detecção de clusters irregulares.

Prof. Alvaro Junior (alvarorpj@yahoo.com)

1) Eficiência de métodos de recomendação.

Métodos de recomendação precisam ser eficazes e eficientes. Muitos tratam bem a primeira qualidade, mas se esquecem da segunda. O estudo e aprimoramento de métodos existentes, ou o desenvolvimento de novos métodos, ou ainda a distribuição de métodos existentes, sempre visando eficiência, representam desafios na linha de projeto e análise de algoritmos no contexto de sistemas de recomendação. Um trabalho específico nesta linha poderia ser a análise de complexidade de classes de métodos existentes. Outra possibilidade seria a proposta de métodos eficientes.

2) Mineração de dados em tempo real para tomada de decisão no trânsito de uma cidade.

No contexto do projeto Olhos da Cidade em execução no DECOM/UFOP, na linha de mineração de dados, há a necessidade de desenvolvimento de sistemas que consigam processar os dados colhidos pelas câmeras em campo em tempo real, descobrindo informação importante para tomada de decisão, com o menor atraso possível, já que a ação como resposta a um evento precisa ser disparada no menor tempo factível. Poderia ser um trabalho específico o estudo de algoritmos eficientes para mineração de dados e descoberta de informação, que se apliquem ao contexto do trânsito.

Prof. David Menotti Gomes (menottid@gmail.com)

1) Classificação de Batimentos Cardíacos a partir de sinal de ECG

A análise do sinal de ECG é a forma mais utilizada para diagnóstico de doenças do coração, visto sua natureza não invasiva. Métodos de aprendizado de máquina para classificação automática de batimentos do coração têm sido largamente usados. Estes métodos sofisticados usam diversos algoritmos e o estudo deles envolvem aprofundados conhecimentos de algoritmos e matemática.

2) Auto Calibração de Câmeras em Visão Estéreo:

Sistemas de visão computacional envolvendo múltiplas câmeras com sobreposição na área de observação vêm sendo utilizados em várias aplicações de monitoramento de ambientes devido em grande parte ao barateamento de câmeras digitais e ao aumento de exatidão em contagens de objeto. No entanto, para inicializar tais sistemas, faz-se necessário a calibração das câmeras para contagens precisas. Existem vários algoritmos para a realização da auto calibragem das câmeras, cada um com suas restrições, que envolvem vários aspectos de complexidade computacional.

3) Estimação de Fluxo Ótico para Localização de Veículos:

Sistemas automáticos de controle de acesso de veículos são compostos basicamente de três partes: Localização do veículo; localização da placa de identificação do veículo e posterior reconhecimento dos caracteres a compõem. As duas últimas tarefas tem sido amplamente estudadas, enquanto que a primeira geralmente é eliminada com o uso de outros componentes como sensores e/ou gatilhos. Uma outra proposta seria utilizar sequências de imagens (um vídeo) obtidas por uma câmera para detectar automaticamente a presença do veículo. Uma boa parcela dos algoritmos de detecção, localização e rastreamento de objetos utilizam a estimação do fluxo ótico do vídeo. Estes algoritmos de fluxos óticos envolvem complexas modelagem matemáticas e apresentam diferentes ordens de complexidade de espaço e tempo.

Prof. Fabrício Benevenuto (benevenuto@gmail.com)
Análise de algoritmos para amostragem de redes sociais

Vários trabalhos recentes que estudam dados coletados da Web utilizam uma estratégia de amostragem para obter esses dados. Muitos desses resultados podem ser tendenciosos devido à estratégia de amostragem utilizada para coletar os dados. Nesse projeto, pretendemos investigar diferentes algoritmos de amostragem de grandes grafos e quantificar o quanto cada algoritmo leva a resultados tendenciosos. Como forma de comparar resultados vamos utilizar uma coleção completa do Twitter (<http://twitter.mpi-sws.org/>), que possui 55 milhões de usuários do Twitter, todos os elos de seguidores e seguidos (grafo com quase 2 bilhões de elos) e todos os tweets postados por esses usuários (1.8 bilhões de tweets).

Como trabalho de PAA, esperamos que o aluno implemente alguns algoritmos de amostragem, avalie suas complexidades e compare os resultados obtidos com os dados completos do Twitter. Algumas sugestões de estratégias de amostras.

- Amostragem por timeline: A API do Twitter permite que usuários colem uma amostra de todos os tweets postados. Uma hipótese é que essa API possa levar à coleta de usuários com mais tweets.
- Amostragem por snowball: Snowball consiste em coletar um conjunto inicial de usuários e então seguir recursivamente elos de seguidor/seguidor dos usuários coletados de forma a coletar novos usuários. Uma hipótese é que essa estratégia leva à coleta de usuários com mais seguidores na rede.

Algumas referências com vários outros algoritmos e estratégias.

Sampling Techniques for Large, Dynamic Graphs
<http://net.cs.uni-tuebingen.de/gi2006/papers%20in%20program/stutzbach.pdf>

Statistical properties of sampled networks
<http://arxiv.org/abs/cond-mat/0505232>

Statistical Properties of Sampled Networks through Random Walks
http://arxiv.org/PS_cache/physics/pdf/0612/0612037v1.pdf

Choudhury, M. D., Lin, Y-R., Sundaram, H., Candan, K. S., Xie, L., Kelliher, A. (2010). *How Does the Sampling Strategy Impact the Discovery of Information Diffusion in Social Media?* In Proceedings of the 4th Int'l AAI Conference on Weblogs and Social Media. ICWSM 2010.

Meeyoung Cha, Hamed Haddadi, Fabrício Benevenuto, Krishna P. Gummadi. **Measuring User Influence in Twitter: The Million Follower Fallacy**. Int'l AAI Conference on Weblogs and Social Media (ICWSM'10). Washington DC, USA. May, 2010.

Prof. Fabrício Benevenuto (benevenuto@gmail.com)

Algoritmos para compressão de URLs

O uso de mensagens curtas tem sido amplamente explorado em sistemas como o Facebook, Twitter e Orkut, permitindo que os usuários compartilhem informações com seus amigos. Em particular, o Twitter é um sistema social voltado unicamente para a postagem de mensagens curtas, que podem conter no máximo 140 caracteres.

Com a grande popularidade desses sistemas o uso de encurtadores de URL tais como bit.ly e tinyurl está se tornando cada vez mais comum.

Sistemas encurtadores de URL funcionam da seguinte forma. Eles traduzem uma URL (que pode consistir de centenas de caracteres) em uma nova URL, tipicamente com poucos caracteres que retorna os códigos HTTP 301 ou 302 de redirecionamento para a URL longa original. Tais serviços podem introduzir alguns atrasos para seus usuários e têm sido amplamente utilizados como forma de ofuscar spam, phishing e malware.

Sendo assim, nesse projeto pretendemos investigar o uso de algoritmos que dispensam o uso de servidor. A idéia é que esses algoritmos sejam executados no momento de envio da mensagem na rede social (o algoritmo encurta a URL) e posteriormente a descompressão da URL ocorre nos clientes que recebem a URL.

Como trabalho de PAA, duas abordagens devem ser investigadas.

- 1) Compressão através de dicionário fixo, construído através do levantamento de padrões freqüentes encontrados em um conjunto de URLs.
- 2) URLs não permitem qualquer tipo de caractere, porém as APIs de sistemas o Twitter permitem que qualquer caractere utf8 seja postado. Este passo consiste em investigar se alterações na codificação da URL podem ser utilizadas como forma de compressão.

Para testar as abordagens de compressão já possuímos uma base com cerca de 150 milhões de URLs encurtadas e suas respectivas versões originais.

Algumas referências relacionadas:

Fabrício Benevenuto, Gabriel Magno, Tiago Rodrigues, Virgílio Almeida. **Detecting Spammers on Twitter**. Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS). Redmond, Washington, USA. July, 2010.

Meeyoung Cha, Hamed Haddadi, Fabrício Benevenuto, Krishna P. Gummadi. **Measuring User Influence in Twitter: The Million Follower Fallacy**. Int'l AAAI Conference on Weblogs and Social Media (ICWSM'10). Washington DC, USA. May, 2010.

Grier, C., Thomas, K., Paxson, V., Zhang, M. 2010. @spam: The underground on 140 characters or less. In ACM conference on Computer and communications security (CCS). 27–37.

Antoniades, D., Polakis, I., Kontaxis, G., Athanasopoulos, E., Ioannidis, S., Markatos, E., and Karagiannis, T. 2011. we.b: The web of short urls. In Int'l Conference on World wide web. 715–724.

Prof. Fabrício Benevenuto (benevenuto@gmail.com)
Análise de algoritmos para redes complexas

Uma rede complexa é nada mais que um grafo. Exemplos de redes complexas são redes sociais online, redes P2P, a Web, rede de transportes aéreos, etc. Em comum várias redes existentes compartilham propriedades estruturais comuns e existem várias métricas para se medir essas propriedades estruturais.

Neste contexto, o objetivo desse seminário implementar algumas dessas métricas, testar suas execuções em redes publicamente disponíveis na Web e computar suas complexidades. Alguma biblioteca para o cálculo dessas métricas pode ser utilizada para a implementação das métricas, mas a complexidade de tais abordagens deve ser analisada.

Uma explicação detalhada sobre as métricas que devem ser implementadas está na Seção 1.3 do seguinte capítulo de livro:
<http://homepages.dcc.ufmg.br/~fabricio/download/jai2011.pdf>

Uma lista de *datasets* pequenos e interessantes pode ser encontrada em:
<http://www-personal.umich.edu/~mejn/netdata/>

Prof. Guillermo Camara (gcamarac@gmail.com)

1) Reconhecimento de linguagem de sinais:

Recentemente, tem havido uma grande necessidade para a comunidade surda ser capaz de se comunicar e se integrar com o resto da sociedade. A comunidade surda está acostumada a realização do seu cotidiano de forma isolada e somente com pessoas capazes de entender a linguagem de sinais. Esse isolamento priva esse segmento considerável da sociedade de uma adequada socialização, educação, e da aspiração ao crescimento da carreira. Essa falta de comunicação dificulta a comunidade surda de desenvolver seus talentos e habilidades em benefício da sociedade em geral. A proposta deste trabalho é implementar algoritmos de tratamento de imagens e reconhecimento de padrões que sejam capazes de realizar o reconhecimento de frases em uma língua de sinais

2) Resumo de vídeo:

Sumarização de vídeo é o processo de extração de um resumo do conteúdo original do vídeo, cujo objetivo é fornecer rapidamente a informação concisa do conteúdo do vídeo, preservando a mensagem do vídeo original. Recentemente, o resumo automático de vídeos tem atraído o interesse dos pesquisadores devido ao seu potencial comercial em diversas aplicações. Como consequência, novos modelos e algoritmos têm sido propostos na literatura da área. Os tipos de resumos gerados podem ser classificados em duas categorias principais: estáticos ou dinâmico. A primeira categoria, consiste na extração de um conjunto de quadros-chave do vídeo original. Já a segunda categoria, coleta um conjunto de tomadas através de análise da similaridade ou da relação temporal entre os quadros. Uma vantagem do uso de resumos dinâmicos é a possibilidade de incluir elementos de áudio e movimentos realçando assim, tanto a expressividade quanto a informação presente no vídeo. O objetivo deste trabalho é gerar resumos dinâmicos de vídeos

3) Rastreamento de pessoas em vídeo:

O processo denominado de rastreamento é a inferência probabilística sobre o movimento de um objeto, dada uma sequência de medidas de suas posições. Dois grandes desafios para o rastreamento de pessoas são a oclusão e a imprevisibilidade do movimento. A oclusão acontece quando os atributos de uma pessoa ficam parcialmente ou totalmente ocultos por um determinado período de tempo. A imprevisibilidade do movimento é algo inerente ao ser humano que precisa ser tratado em sistemas de monitoramento. Os filtros probabilísticos tentam prever o movimento das pessoas entre frames consecutivos de um vídeo. Utilizando um filtro para realizar a estimativa de posições futuras das pessoas, é possível contornar situações como as mencionadas acima, relacionadas a oclusões e imprevisibilidade. Este trabalho visa desenvolver um sistema de monitoramento visual de pessoas que permita rastrear a movimentação de pessoas em cenas variadas.

Otimização no Sistema de Transporte Público

As propostas apresentadas neste documento estão voltadas para a otimização de problemas encontrados na operação de sistemas de transporte público por ônibus urbano. Neste cenário destacamos: *i*) o Problema de Programação dos Veículos – PPV (*vehicle scheduling problem*); *ii*) o Problema de Programação das Tripulações – PPT (*crew scheduling problem*) e, *iii*) o Problema do Rodízio das Tripulações – PRT (*crew rostering problem*). Estes problemas, com suas variações, tais como a programação dos veículos com várias garagens, a integração entre a programação dos veículos com a tabela de horários, a integração da programação dos veículos com a programação das tripulações entre outros, são de grande interesse prático e econômico.

Entre 2008 e 2009 o coordenador realizou a implementação de uma nova técnica de busca local, a busca em vizinhança de grande porte - BVGP (*very large-scale neighborhood search*), proposta por Ahuja et. al. (2002). Esta é uma técnica de busca ainda pouco explorada, principalmente para abordar problemas do setor de transporte público, abrindo assim uma nova frente de pesquisa.

Nos trabalhos de Silva e Cunha (2009, 2010a, 2010b, 2011) foram desenvolvidos modelos que utilizam a técnica BVGP para a programação dos veículos e das tripulações com desempenho bastante promissor. Nestes trabalhos foi observado que o resultado de uma busca BVGP é fortemente dependente da solução corrente de partida. Sendo assim, pretende-se combinar a BVGP com outras heurísticas construtivas que, periodicamente, realizam perturbações na solução corrente. Nesta classe destacam-se a heurística *Variable Neighborhood Search* –VNS (Mladenovic e Hansen, 1997), método de busca local que consiste em explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança, e a heurística *Iterated Local Search* – ILS (Baxter, 1981), baseada na idéia de que um procedimento de busca local pode ser melhorado gerando-se novas soluções de partida, as quais são obtidas por meio de perturbações na solução ótima local.

Bibliografia

- AHUJA, R.K.; ERGUN, O.; ORLIN, J.B. e PUNNEN, A.P. (2002) A survey of very large scale neighborhood search techniques. *Discrete Applied Mathematics*, v.123., p. 75-102.
- BAXTER, J. (1981) Local optima avoidance in depot location. *Journal of the Operational Research Society*, v. 32, p. 815-819.
- MLADENOVIC, M e HANSEN, P. (1997) Variable Neighborhood Search. *Computers and Operations Research*, v. 24, p. 1097 - 1100.
- SILVA, G. P. (2010) Programação de veículos com múltiplas garagens: um estudo de caso de empresas de transporte público. In: *Panorama Nacional da Pesquisa em Transportes*, v. 1. p. 127-140.
- SILVA, G. P.; CUNHA, C. B. (2009) Aplicação da técnica de busca em vizinhança de grande porte para otimizar a frota de empresas de transporte público. *XLI Simpósio Brasileiro de Pesquisa Operacional. Porto Seguro-BA*.
- SILVA, G. P.; CUNHA, C. B. (2010a) Uso da técnica de busca em vizinhança de grande porte para a programação da escala de motoristas de ônibus urbano. *Revista Transportes*, v. 18, p. 64-75.
- SILVA, G. P.; CUNHA, C. B. (2010b) GRASP and Very Large-scale Neighborhood Search for Solving Complex Vehicle Scheduling Problems. In: *Proceedings of the XVI Pan-American Conference on Traffic and Transportation Engineering (PANAM)*, v. 1. p. 125-137. Lisboa, Portugal.
- SILVA, G. P.; CUNHA, C. B. (2011) GRASP with Very Large-Scale Neighborhood Search and Path-Relinking for the Integration of Vehicle and Crew Scheduling. In: *Computers in Urban Planning and Urban Management 2011*. Calgary, Canadá. Artigo completo.

Prof. Haroldo Lima (haroldo.santos@gmail.com)

1) Algoritmos de Branch & Bound para Problemas de Otimização Combinatória

Baseada no paradigma de divisão e conquista, a técnica de Branch & Bound é a técnica de maior sucesso na resolução exata de problemas de otimização combinatória complexos (NPdifíceis). Suas aplicações incluem problemas nas mais diversas áreas como planejamento de redes, logística, biologia computacional e escalonamento de tarefas.

2) Estratégias de Programação Dinâmica para Problemas de Otimização Combinatória :

A técnica de programação dinâmica permite que a solução de subproblemas auxilie na construção da solução ótima para problemas complexos. O projeto de algoritmos eficientes de programação dinâmica passa pela definição da função de recorrência e da avaliação da estratégia mais apropriada para o problema em questão: deve-se considerar o uso da estrutura matricial com bottomup ou a estratégia topdown que utiliza memoização.

Prof. Joubert Lima (joubertlima@gmail.com)

Hoje em dia precisamos analisar grandes volumes de dados. Os dados normalmente são estruturados em tabelas e estas tabelas possuem atributos. Por exemplo, podemos ter a tabela aluno com os atributos departamento matriculado, ano de ingresso, semestre de ingresso, ano de egresso, semestre de egresso, nome, endereço (rua, complemento, bairro, cidade, estado, cep), nome do responsável legal, tel fixo, celular, {nota na disciplina 1, professor na disciplina 1, frequência na disciplina 1, nota na disciplina 2,}. A estrutura de uma tabela é ilustrada na Figura 1. Note que, em nosso exemplo não buscamos criar múltiplas tabelas para obedecer às formas normais propostas por Codd na década de 70-80. Apenas focamos no problema a ser atacado e assumimos que o mesmo poderá ser feito para uma ou n tabelas. É importante ressaltar que o termo tabela é usado comumente, mas o termo formal na área de Banco de Dados é relação e não tabela.

Depto matriculado	Ano ingresso	Sem. Ingresso	...	Nome	Rua	...	Nome resp	tel	Nota D1	Prof D1	Freq D1
-------------------	--------------	---------------	-----	------	-----	-----	-----------	-----	---------	---------	---------	------

Figura 1. Estrutura de uma tabela

Já na Figura 2 temos a tabela da Figura 1 com alguns dados. Note que, uma forma de analisar os dados de tal tabela seria saber qual a média de notas dos últimos cinco anos de um determinado departamento, anos estes em que o Brasil viveu um enorme crescimento de suas universidades. Daí, buscamos também a média (ou mediana ou qualquer outra métrica) de tal departamento desde sua criação. Com isso, sabemos se neste departamento o crescimento do ensino brasileiro de terceiro grau afetou positivamente ou negativamente as notas, as frequências, o número de professores, o índice de repetência, etc...

Depto matriculado	Ano ingresso	Sem. Ingresso	...	Nome	Rua	...	Nome resp	tel	Nota D1	Prof D1	Freq D1
DECOM	2000	01		Zé	www		Jesus	35591692	6.3	Joubert	3	
DECOM	2000	01		Toim	Online		Deus	1234567	5.4	Joubert	12	
DECOM	2010	02			6.7	Joubert	7	
DECOM	2011	01							8.9	Joubert	0	
DECOM	2006	01							6.3	Joubert	8	
DEMAT	2006	02							8.7	Toim	14	
DEMAT	2009	02							6.7	ZeZim	1	
...												

Figura 2. Estrutura com dados operacionais

Para responder estas e muitas outras análises dos dados, temos que varrer a tabela a todo momento? A resposta é não. Conseguimos armazenar na mesma tabela as chamadas agregações de dados, úteis para responder análises como fizemos em nosso exemplo. Uma das maneiras de se fazer isto é adicionar o atributo ALL na tabela anterior. Assim, poderemos saber facilmente qual média de notas no departamento DECOM em 2009, semestre 01, por exemplo, adicionando o registro $r = \{ \text{DECOM, 2009, 01, all, all, ..., 77(Nota D1), all, 5(Freq D1), etc...} \}$ na tabela, conforme ilustra Figura 3. Se quisermos saber a média de notas, etc..., de toda a escola em 2009-01, basta inserir o registro $r1 = \{ \text{all, 2009, 01, all, all, ..., 68(Nota D1), all, 4(Freq D1), etc...} \}$. Note que, bastou inserir um atributo ALL em nome de departamento para se obter tal informação.

Depto matriculado	Ano ingresso	Sem. Ingresso	...	Nome	Rua	...	Nome resp	tel	Nota D1	Prof D1	Freq D1
DECOM	2000	01		Zé	www		Jesus	35591692	6.3	Joubert	3	
DECOM	2000	01		Toim	Online		Deus	1234567	5.4	Joubert	12	
DECOM	2010	02			6.7	Joubert	7	
DECOM	2011	01							8.9	Joubert	0	
DECOM	2006	01							6.3	Joubert	8	
DEMAT	2006	02							8.7	Toim	14	
DEMAT	2009	02							6.7	ZeZim	1	
DECOM	2009	01	...	ALL	ALL	...	ALL	ALL	77	ALL	5	...
ALL	2009	01	..	ALL	ALL	...	ALL	ALL	68	ALL	4	

...												
ALL	ALL	ALL	..	ALL	ALL	..	ALL	ALL	71	ALL	6	

Figura 3. Uma das representações de um cubo de dados

Pode parecer fácil, mas não é. O grande problema é como gerar todas as possíveis agregações da tabela alunos, varrendo a tabela em questão a menor quantidade de vezes possível. A Figura 3 é chamada cubo completo de dados. Montar um cubo completo de dados é um problema exponencial em tempo e espaço, portanto desde sua definição formal em Gray et al. [1] muito esforço tem sido gasto em propor abordagens que tendem a reduzir tal complexidade, valendo-se de novas formas de representar um cubo e novas ideias para montá-lo mais rápido.

Dentre as abordagens existentes, podemos citar as abordagens que utilizam grafos para representar cubos como uma das mais eficientes em termos de runtime e consumo de memória. Entre as abordagens baseadas em grafos, podemos citar as abordagens Star-cubing [6], Dwarf [3] e MCG [2] como as mais eficientes da literatura para computar e representar um cubo usando grafos (lembrando que árvores são sub-tipos de grafos). Leia-se eficiente por ser mais rápida e ao mesmo tempo consumir menos memória.

Infelizmente não existe um estudo sobre a comparação da complexidade de tais abordagens (nem em tempo e nem em espaço). Um estudo que merece atenção é o feito para os chamados closed cubes [5], que são cubos que não precisam armazenar determinadas agregações. As abordagens Star, Dwarf e MCG prometem até maiores reduções do que os chamados closed cubes. Caberia a três alunos do curso de PAA a formalização da complexidade das abordagens Star e MCG, a verificação da prova feita em [4] e o estudo comparativo entre as três abordagens. Maiores e muito mais detalhadas informações com o Prof. Joubert, sala 13 do DECOM (email: joubertlima@gmail.com, página: www.joubertlima.com.br).

Referências

- [1] GRAY, J.; CHAUDHURI, S.; BOSWORTH, A.; LAYMAN, A.; REICHART, D.; VENKATRAO, M.; PELLOW, F.; PIRAHESH, H. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. jan. 25 1997.
- [2] LIMA, J. de C.; HIRATA, C. M. Multidimensional Cyclic Graph Approach: Representing a Data Cube without Common Sub-Graphs. Article accepted for publication. Information Sciences Journal. Elsevier. 2010. DOI information: 10.1016/j.ins.2010.05.012
- [3] SISMANIS, Y.; DELIGIANNAKIS, A.; ROUSSOPOULOS, N.; KOTIDIS, Y. Dwarf: Shrinking the PetaCube, abr. 04 2002.
- [4] SISMANIS, Y.; ROUSSOPOULOS, N. The Dwarf Data Cube Eliminates the High Dimensionality Curse, UM Computer Science Department; CS-TR-4552 UMIACS; UMIACS-TR-2003-120, 2003.
- [5] SISMANIS, Y.; ROUSSOPOULOS, N. The polynomial complexity of fully materialized coalesced cubes, Proceedings of the Thirtieth international conference on Very large data bases - Volume 30, VLDB '04, 2004.
- [6] XIN, D.; HAN, J.; LI, X.; SHAO, Z.; WAH, B. W. Computing iceberg cubes by top-down and bottom-up integration: The starcubing approach. IEEE Trans. Knowl. Data Eng., v. 19, n. 1, p. 111-126, 2007.

Prof. Luiz Merschmann (luizhenrique@iceb.ufop.br)

1) O processo de descoberta de conhecimento em bases de dados -- processo KDD (*Knowledge Discovery in Databases*) -- é formado basicamente por três etapas: pré-processamento dos dados, mineração de dados e pós-processamento. Uma das possíveis tarefas no pré-processamento dos dados é a redução do número de atributos das bases de dados. Essa redução do número de atributos é realizada a partir de técnicas de seleção de atributos. Algumas técnicas de seleção de atributos utilizam um dos paradigmas de programação estudados na disciplina de PAA. Estude sobre técnicas de seleção de atributos para verificar qual paradigma de programação é adotado e explique com detalhes pelo menos uma dessas técnicas.

Prof. Marcone Freitas (marcone.freitas@gmail.com)

Título: O Problema de Corte de Estoque Unidimensional com Reaproveitamento das Sobras de Material: Estudo de complexidade da heurística FFD-modificada

Este trabalho tem seu foco em Problemas de Corte de Estoque (Cutting Stock Problem, CSP), que consiste na otimização do processo de corte de peças maiores (objetos), disponíveis em estoque, para a produção de peças menores (itens), satisfazendo a uma determinada demanda. Entre os objetivos a serem atingidos destaca-se a minimização das perdas, dos excessos da produção e dos padrões de corte a serem utilizados. Entretanto, se uma sobra é suficientemente grande para ser reaproveitada posteriormente, em outro lote de encomendas, não faz sentido contabilizá-la como perda. Isto introduz uma postura diferente frente ao problema típico de corte e soluções com perda mínima passam a não ser necessariamente as mais interessantes, já que as sobras podem ser reaproveitadas. Este tipo de problema ocorre em diversos processos industriais, como no corte de bobinas de papel e alumínio, barras de aço, chapas metálicas e de madeira, placas de circuito impresso, caixas de papelão, rolos de tecido, entre outros. Assim, o CSP é essencial para o planejamento da produção nessas indústrias e a redução dos custos de produção e melhoria da eficiência estão frequentemente associadas à utilização de estratégias adequadas de cortes. O objetivo no presente trabalho é o de analisar a complexidade do método de solução heurística FFD-M (Modified First Fit Decreasing) para esse problema. Esse método consiste em gerar padrões por meio da heurística FFD clássica e aplicá-los para atender a demanda, obedecendo às restrições de estoque e não permitindo que ocorra excesso de produção de itens. Caso o padrão gerado pela FFD seja inaceitável, ou seja, se a parte não utilizada do padrão não estiver em conformidade com os parâmetros de perda e sobra, então esse padrão é corrigido por meio da resolução de um problema da mochila.

Referências:

CHERRI, A. C., ARENALES, M. N. (2005a). O Problema de Corte de Estoque com Reaproveitamento das Sobras de Material – Heurística FFD Modificada. In: Anais do XXXVII Simpósio Brasileiro de Pesquisa Operacional, Gramado, p. 1700-1711.

CHERRI, A. C.; ARENALES, M. N. (2005b). Heurísticas para o problema de corte com reaproveitamento das sobras de material. In: XXVIII Congresso Nacional de Matemática Aplicada e Computacional, 2005, São Paulo. Anais do XXVIII SBMAC. v. 1, p. 1-6.

PILEGGI, G. C. F. (2002) “Abordagens para otimização integrada dos problemas de geração e seqüenciamento de padrões de corte”, Tese de Doutorado em Ciências no Curso de Ciência da Computação e Matemática Computacional da USP, São Carlos, SP.

POLDI, K. C. (2002) Heurísticas para o problema de corte de estoque unidimensional inteiro. In: XXXIV Simpósio Brasileiro de Pesquisa Operacional, 2002, Rio de Janeiro. Anais do XXXIV SBPO.

POLDI, K. C. (2003) Algumas extensões do problema de corte de estoque. Dissertação de mestrado, ICMC – USP.

POLDI, K. C.; ARENALES, M. N. (2003) O problema de corte de estoque unidimensional inteiro com restrições de estoque. In: XXXV Simpósio Brasileiro de Pesquisa Operacional, 2003, Natal/RN. Anais do XXXV SBPO, p. 1498-1509.

Prof. Ricardo Rabelo (rrabelo@gmail.com)

Os algoritmos distribuídos são um paradigma no qual o sistema computacional consiste em sistemas de memória distribuída, interligados por um canal de comunicação.

Os sistemas de memória distribuída consistem numa coleção de processadores interconectados por algum canal que permita comunicação ponto a ponto. Os processadores não compartilham fisicamente qualquer memória, de forma que a troca de mensagens é necessária.

Uma das principais funções de um sistema de memória distribuída neste ambiente é atuar como um redistribuidor de mensagens de outros sistemas de memória distribuída. Considerando um sistema de memória distribuída como um grafo não direcionado $G_P=(N_p,E_p)$, no qual N_p é o conjunto de nós e E_p as arestas que ligam esses nós, a mensagem recebida num nó pode ser definida pelo par (q,Msg) , que significa que a mensagem Msg pode ser enviada para o processador q . Quando um processador r recebe a mensagem (q,Msg) ele verifica se $r=q$, caso positivo, ele processa a mensagem, caso negativo, ele a encaminha para frente.

A partir deste sistema de troca de mensagens, define-se uma representação para o algoritmo distribuído consiste em um grafo direcionado conectado $G_t=(N_t,D_t)$ onde N_t são as instancias do algoritmo. Para cada um temos duas arestas, In , que define o ponto de entrada, e Out , que define o ponto de saída. Cada instancia tem a seguinte forma:

Algoritmo T

Efetua alguma computação;

Envia uma mensagem em um subconjunto de arestas de saída;

Repeat

Recebe mensagens no subconjunto de arestas de entrada;

Efetua alguma computação

Envia mensagem em um subconjunto (possivelmente vazio) de arestas de saída;

Enquanto (Condição de parada de todas instancias não acontecer)

Assume-se que todas as instancias executam o mesmo código. Neste tipo de algoritmo, além da complexidade de tempo e espaço, normalmente encontrada em algoritmos de memória simples ou compartilhada, é levada em conta também a complexidade da troca de mensagens para a convergência da computação. Além disso, a complexidade tempo de computação de cada instancia introduz um problema de sincronização, associado a essa complexidade.

Problemas de pesquisa para seminário:

- 1- Conectividade do grafo. Dado um grafo G_t , deseja-se descobrir se o mesmo é conectado, considerando que cada vértice executa uma instância do mesmo algoritmo distribuído. Um estudo comparativo do algoritmo não distribuído, visão global do grafo, com a versão distribuída. Quais são as soluções necessárias para encontrar a melhor complexidade de troca de mensagens
- 2- Eleição de líder: Dado um grafo G_t , a eleição de líder consiste no problema de escolher o vértice líder, considerando que inicialmente, cada vértice conhece a si mesmo apenas. Se G_t é um grafo completo, a complexidade de troca de mensagens pode chegar a $O(n^3)$. Efetue um estudo para diminuir a complexidade de troca de mensagens para $O(n \log n)$ caracterizando os grafos envolvidos
- 3- Auto-estabilização. Se a execução de um determinado algoritmo distribuído sobre um grafo G faz com que aconteça uma convergência a um estado global no qual alguma propriedade em particular se mantém, este algoritmo é considerado de auto-estabilização. Um exemplo de propriedade é por exemplo a remoção de ciclos no grafo. Este algoritmo distribuído será

considerado tolerante a falhas caso alguma modificação aconteça no grafo e após uma série de operações, ele volte ao estado global com a propriedade válida. Um estudo sobre como a ordem de complexidade deve ser levada em sistemas tolerantes a falhas, caso esta ordem de complexidade seja um dos fatores de modificação do próprio grafo. Por exemplo, se a ordem de complexidade de um sistema tolerante a falhas for $O(n^3)$, alguns vértices são considerados inativos, ativando novamente o sistema tolerante, até que o grafo “desapareça”.

- 4- Formalize dois problemas clássicos de grafos sob a ótica dos algoritmos distribuídos: o problema de fluxo máximo em redes e a árvore geradora mínima, considerando que o grafo tem em cada vértice uma instância do algoritmo distribuído.
- 5- As redes sem fio Bluetooth possuem um algoritmo de formação do grafo conectado baseado em envio intensivo de mensagens, com baixíssima probabilidade de sucesso. O estudo deve avaliar estratégias eficientes de eleição de líder e de auto-estabilização.
- 6- Cache distribuído. Algoritmos de coerência de cachê são comuns na literatura, considerando memória compartilhada. A pesquisa deve considerar a existência de um cachê num sistema de troca de mensagens, permitindo que sistemas de memória compartilhada possam ser emulados em sistemas de troca de mensagens.

Prof. Tiago Carneiro (tiagogsc@gmail.com)

ROTAS E AMBIENTES VIRTUAIS GEOREFERENCIADOS

1) From GPS traces to a routable road map:

Este trabalho consiste no estudo de algoritmos para a geração automática de um mapa roteável de estradas (grafo dirigido) a partir de bases massivas de trajetórias GPS. Poderiam ser estudadas variações desse mesmo problema: algoritmos capazes de gerar os mapas a partir de dados GPS com baixa taxa de amostragem (uma posição a cada 3 minutos) e algoritmos capazes de refinar mapas em tempo real. O TerraLAB dispõe de massas de dados reais.

2) Melhor caminho com propriedades cinéticas:

Uma vez definida origem e destino em um mapa roteável de estradas (grafo dirigido), estudar algoritmos que determine o melhor caminho (pode ser menor em um caso especial) a ser seguido dadas as restrições cinemáticas do veículo que será utilizado para percorrê-lo. Por exemplo, declividade máxima do terreno suportada pelo veículo ou ângulo de máxima curvatura atingida pelo veículo. Exemplos de algoritmos são: menor caminho de Dijkstra e A*.

3) Terrenos Digitais em Ambiente Virtuais de Larga Escala:

estudo de estruturas de dados, algoritmos e paradigmas de programação destinados a representação em alta-resolução do relevo e cobertura do solo de regiões geográficas de grande extensão. Os modelos de relevo e cobertura do solo precisam ser passíveis de renderização em tempo real para a construções de sistemas de realidade para regiões abertas e de larga escala, por exemplo, município inteiros, estados, países, ou até mesmo o globo. Técnicas como armazenamento em hierarquia de resoluções (escalas) e o paradigmas como MapReduce podem ser aplicados para diminuir o volume de dados a ser tratado em tempo real. Heurísticas baseadas em propriedades da visão humana podem ser utilizadas para adaptar dinamicamente o grau de detalhamento (volume de dados) do terreno apresentado. RESUMINDO: A idéia é conseguir fazer o que o GoogleEarth faz!

Prof. Tiago Carneiro (tiagogsc@gmail.com)

VISUALIZAÇÃO CIENTÍFICA de MODELOS AMBIENTAIS

4) Visualização Científica de Modelos Espaciais Dinâmicos de Fenômenos Geográficos:

Estudos acerca de estruturas de dados e algoritmos para a representação, transmissão, visualização e análise eficiente de dados massivos que representam o estado dinâmico de modelos computacionais para fenômenos geográficos. Para a análise e depuração eficiente de simulações é preciso monitorar em tempo-real o comportamentos dos modelos que representam o modelo em estudo. Modelos espaciais que representam uma determinada paisagem com riqueza de detalhes podem encerrar um grande volume de dados que precisam ser serializados e continuamente transmitidos as ferramentas e interfaces gráficas utilizadas para sua visualização e análise. Estrutura de dados e algoritmos que permitam o armazenamento, transmissão e processamento eficiente dos modelos podem reduzir sobremaneira o tempo de desenvolvimento desses modelos. Tornando por vezes viável algo que antes não seria. Heurísticas como a serialização e transmissão incremental somente das mudanças instantenas sofrida pelo estado do modelo, políticas de caching e algoritmos de compactação podem ser de grande utilidade para a solução desse problema.