**Centro de Ensino Unificado de Teresina - CEUT**

Dann Luciano de Menezes

**Algoritmos Genéticos na Otimização de Consultas por Similaridade**

Teresina, PI

2010

**Centro de Ensino Unificado de Teresina - CEUT**

Dann Luciano de Menezes

**Algoritmos Genéticos na Otimização de Consultas por Similaridade**

Projeto apresentado como requisito para conclusão da disciplina de Trabalho de Conclusão de Curso I e pré requisito para cursar a disciplina de Trabalho de Conclusão de Curso II

Teresina, PI

2010

**Sumario**

**1 Tema**

**2 Problema**

**3 Objetivos**

**4 Justificativa**

**5 Metodologia**

**6 Referenciais Teóricos**

**6.1 Consultas por Similaridade**

**6.2 Algoritmos Genéticos**

**7 Cronograma**

**8 Recursos**

**9 Referências**

**1 Tema**

Algoritmos Genéticos na Otimização de Consultas por Similaridade.

**2 Problema**

Como otimizar o tempo computacional na obtenção de consultas por similaridade? A técnica dos Algoritmos Genéticos otimizaria o processo?

**3 Objetivos**

* Implementar uma ferramenta de busca em Domínios Métricos utilizando a técnica dos Algoritmos Genéticos.
* Acelerar as operações de busca de dados em Domínios Métricos
* Criar uma melhor representação do problema

**4 Justificativa**

O tempo computacional gasto para realizar a operação de recuperação de dados em SGBDs é relativamente baixa, considerando o volume de informações armazenadas, isso ocorre somente nos tipos de dados mais simples como números e letras. Quando aumentamos a complexidade do dado a ser trabalhado aumentamos também o tempo computacional do processo de busca. O pior caso é quando temos que trabalhar com dados binários, como imagens, musicas, vídeos. Esses dados complexos são armazenados em estruturas de dados especificas nomeadas de Domínios Métricos. Apesar das características de indexação dos Domínios Métricos as operações de consultas dos dados são muitas vezes ineficientes devido ao problema da “maldição da dimensionalidade” apud Bueno et al (2005). Buendo et al (2005) ainda afirmam que “em alguns casos a busca sequencial mais vantajosa”. Neste trabalho, procuramos mostrar uma abordagem diferente e eficiente pois segundo Zuben (***~~ANO~~***) são os problemas de otimização que mais recebem contribuições a partir das técnicas de computação evolutiva. Sendo os Algoritmos Genéticos a técnica que iremos utilizar neste trabalho.

**5 Metodologia**

**6 Referenciais Teóricos**

**6.1 Consultas por Similaridade**

Os primeiros Sistemas Gerenciados de Base de Dados (SGBDs) tinham a finalidade de facilitar e agilizar o processo de recuperação exata dos dados. Essa recuperação exata foi e continua sendo uma das necessidades das aplicações que os SGBDs devem suportar. Com o passar do tempo novas necessidades foram surgindo, e a recuperação exata dos dados deixar de ser necessária em algumas aplicações especificas como, sistemas de apoio à decisão. Segundo Buendo et al (2005) isso porque essas aplicações possuem uma característica exploratória. Outra funcionalidade que as aplicações atuais necessitam é o armazenamento e recuperação de tipos de dados complexos, como já foi afirmado acima. As operações de recuperação não são realizadas em cima dos dados complexos propriamente dito e sim de características extraídas quando os dados são inseridos ou alterados. Os dados complexos devem passar antes por extratores de características que codificam as características dos dados complexos em características que os descrevem e depois adicionam essas características descritivas em estruturas de dados denominadas de Métodos de Acesso Métricos (MAMs). Os Métodos de Acesso Métricos servem para indexar os dados pertencentes a um Domínio Métrico.

**6.1.1 Domínios Métricos**

Um Domínio Métrico é definido como M = (D, d( )), onde D é o conjunto de todos os objetos que atendem às propriedades do domínio e d( ) é uma função de distância, ou métrica, entre esses objetos. Dados x,y e z E D, uma função de distancia (métrica) d : D x D 🡪 R+ deve satisfazer as seguintes propriedades:

*1 Simetria*: d(x,y) = d(y,x);

*2 Não Negatividade*: 0 <= d(x,y) <= 8, d(x,x) = 0;

*3 Desigualdade Triangular*: d(x,y) <= d(x,z) + d(z,y);

A propriedade de numero um garante que a distancia entre dois pontos tem que ser igual independente da ordem em que é calculada. Já a segunda propriedade, diz que a distancia entre dois pontos deve pertencer ao intervalo de zero ate o infinito, e que a distancia entre dois ponto somente é zero se esses forem os mesmos pontos. A terceira propriedade merece um maior destaque, visto que a partir dela é possível diminuir os cálculos de distâncias em busca por similaridade em Domínios Métricos como afirma Bueno et al (2005).

Em domínios métricos, existem vários operadores de consulta por similaridade, mas dois se destacam:

**RQ(sq, rq) – Consulta por abrangência (“Similarity Range Query”):** retorna todos os objetos cujas distâncias até objeto central da consulta sq sejam menores ou iguais ao raio da consulta rq;

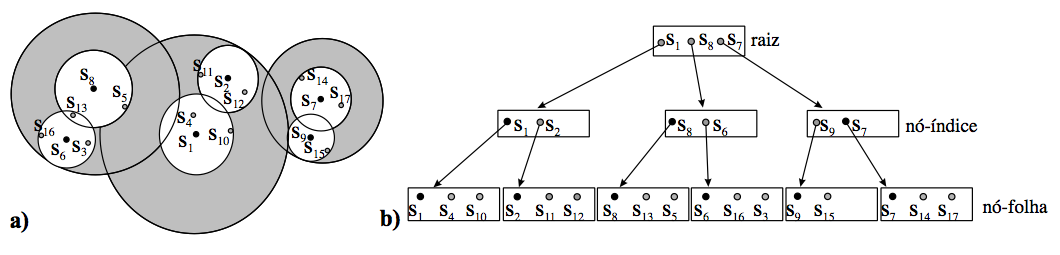
**k-NNQ(sq, k) – Consulta aos k-vizinhos mais próximos (“k-Nearest Neighbor Query”):** retorna os k objetos mais próximos do objeto central de consulta sq;

**6.1.2 Métodos de Acesso Métrico**

Como já mencionado acima os Métodos de Acesso Métricos (MAMs) são estruturas de indexação. Segundo Bueno et al (2005) o ponto de partida para a criação dos Métodos de Acesso Métrico foram os estudos de Burkhard e Keller em 1973, e só depois em 1997 é que os MAMs dinâmicos foram criados com a M-*tree*, e em 2000 com Slim-*tree*.

Segundo Bueno et al (2005) Depois destes trabalhos pioneiros vários outros MAMs foram criados, como é o caso da *Pyramid-tree* e a *P-Sphere tree* (*Probabilistic sphere tree*).

A implementação do algoritmo para resolução do problema pode utilizar qualquer um MAMs propostos. Será feito um estudo sobre qual estrutura de dados melhor complementa a representação do problema.

Figura 1: a) Representação de uma Slim-tree e b) sua estrutura lógica

**6.2 Algoritmos Genéticos**

Os algoritmos genéticos (AG) do inglês *genetic algotithms* (*GA*) são uma técnica de busca extremamente eficiente no seu objetivo de varrer o espaço de soluções e encontrar soluções próximas da solução ótima. Seus mecanismos foram fortemente inspirados na teoria da evolução proposta por Charles Darwin em 1859, os primeiros cientistas a introduzir e disseminar os Algoritmos Genéticos foram John Holland e seu aluno David Goldberg em 1975 e 1989 respectivamente.

Uns dos princípios dos algoritmos genéticos se baseiam na frase do livro **A Origem das Espécies** de Charles Darwin, que diz “Quanto melhor um individuo se adaptar ao seu ambiente, maior será sua chance de sobreviver e gerar descendentes”. Os algoritmos de otimização se caracterizam por buscarem a melhor solução para um determinado problema, pro isso os Algoritmos Genéticos se encaixam tão bem nessa classe de problemas. Segundo Lacerda Estéfane e Carvalho André (ANO) os algoritmos genéticos consistem em tentar varias soluções e utilizar a informação obtida neste processo de forma a encontrar soluções cada vez melhores.

Melhorar AQUIi

**6.2.1 Terminologia**

Os algoritmos genéticos possuem muitos termos originados biologia, isso porque segundo Lacerda Estéfane e Carvalho André(ANO) eles são a metáfora dos processos de reprodução dos indivíduos mais adaptados, e durante a reprodução ocorrem processos de *crossover* e mutação . Para entendermos melhor esse paradigma precisamos conhecer alguns dos principais termos encontrados no âmbito dos Algoritmos Genéticos.

* **Cromossomo e Genoma:** Representam um ponto no espaço de busca, são a representação da solução codificada em uma estrutura de dados, geralmente não passam de um vetor ou uma árvore.
* **Gene:** Nada mais são do que características ou atributos do problema em questão, codificados em uma posição ou nó do cromossomo.
* **Indivíduo:** Um membro da população. São formados por um ou mais cromossomos e sua aptidão.
* **Genótipo:** Representa a informação contida dentro dos cromossomos
* **Fenótipo:** Representa o objeto construído a partir da informação dos do genótipo.
* **Alelo:** São os diferentes valores que um gene pode assumir.
* **Epistasia:** Interação ente genes do cromossomo, ou seja, quando um gene é influenciado por outro gene, “problemas com alta Epistasia são de difíceis solução por algoritmos genéticos” segundo Lacerda Estéfane e Carvalho André(ANO)

**6.2.2 Representação Real**

Quando foram idealizados as representações dos problemas eram codificados utilizando dados binários (zeros e uns), mas existem casos onde uma representação real é mais cabível, por exemplo em problemas com parâmetros contínuos matemáticos onde a precisão de casas decimais é muito grande. A representação binária nestes casos chega a ser inviável “pois para cada ponto decimal acrescentado na precisão, é necessário adicionar 3,3 bits na cadeia” (LACERDA Estéfane e CARVALHO André, ANO). Em nosso trabalho será realizado um estudo para melhor representação do problema.

**6.2.3 População Inicial**

A população inicial como o próprio nome já diz é a primeira população de indivíduos que os algoritmos genéticos trabalham. Elas podem ser geradas a partir de vários métodos. O primeiro desses métodos é o aleatório, a cada execução do algoritmo uma nova e diferente população inicial é formada, e não se tem garantia que essa população inicial seja igual a população inicial de outra execução. Outro método seria utilizar como população inicial soluções encontradas por outras técnicas de otimização, ou algumas soluções aceitáveis para o problema. Esse método é conhecido como “*seeding*” segundo Lacerda Estéfane e Carvalho André(ANO).

**6.2.4 Função Objetivo**

Na sua maioria os problemas do mundo real são bastante complexos fazendo com que a função objetivo seja também bastante complexa e com alto custo dos recursos computacionais. É importante simplificar ao máximo a função objetivo, mas essa simplificação trará perdas na informação na solução final. Para reduzirmos o tempo computacional é interessante utilizarmos uma simplificação da função objetivo nas primeiras gerações de indivíduos e deixarmos a função objetivo real para as ultimas gerações. Uma outra técnica bastante difundida para a redução dos recursos computacionais é utilizar os algoritmos genéticos para localizar a região da função objetivo mais promissora a ser o máximo da função e depois utilizar um algoritmo de Subida de Encosta para localizar a possível melhor solução. Isso porque os métodos de Subida de Encosta são extremamente eficientes na obtenção do ponto de máximo local.

**6.2.5 Critérios de Parada**

Critérios de parada são responsáveis por fazer com que o algoritmo genético não fiquem evoluindo infinitamente. Para cada tipo de problema pode ser utilizado um critério de parada diferente. Em alguns casos pode se priorizar o tempo ou os recursos computacionais, em outros a quantidade de gerações, ou então a chegada ao valor ótimo da função (somente se conhecermos esse valor). E por ultimo o algoritmo poderia para sua execução quando a convergência esta acima de um certo ponto. Convergência ocorre quando a aptidão dos indivíduos de uma determinada população não melhorar por um dado numero de gerações.

**6.2.6 Seleção**

**6.2.6.1 Roda da roleta**

**6.2.6.1 Seleção por Torneio**

**6.2.7 *Crossover***

**6.2.7.1 Uniforme**

**6.2.7.2 Multiponto**

**6.2.8 Mutação**

**6.2.9 Elitismo**

A cada interação do algoritmo genético uma nova população de soluções candidatas a resolução do problema é formada. Nessa nova população podemos perder informações valiosas, como a melhor solução já encontrada para o problema. Isso ocorre devido as operações de *crossover* ou mutação. Segundo Lacerda Estéfane e Carvalho André(ANO) “é interessante transferir o melhor cromossomo de uma geração para a outra”. Quando mantemos a melhor solução (cromossomo) da população anterior na nova população dizemos que o algoritmo é elitista.

**7 Cronograma**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Atividades** | **Janeiro** | **Fevereiro** | **Março** | **Abril** | **Maio** | **Junho** |
| Estudo da Teoria Envolvida | **X** | **X** |  |  |  |  |
| Estudo sobre a Representação do Problema |  | **X** |  |  |  |  |
| Implementação dos Algoritmos |  | **X** | **X** | **X** |  |  |
| Suíte de Testes |  |  |  | **X** |  |  |
| Analise dos Resultados |  |  |  |  | **X** |  |
| Elaboração da Monografia |  |  |  | **X** | **X** | **X** |

**Referências**

BUENO Renato, TRAINA Agma J. M. e Traina Caetano Jr. **Algoritmos Genéticos para Consultas por Similaridade**. 2005

ZUBEN Fernando J. Von. **Computação Evolutiva: Uma Abordagem Pragmática**

GOLDBERG David E. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Company Inc. 1989

BÄCK Thomas, FOGEL David B e MICHALEWICZ Zbigniew. Evolutionary Computation1 : Basic Algorithms and Operators. Institute of Physics Publishing. 2000

RUSSEL Stuart e NORVIG Peter. Inteligência Artificial. 2˚ Ed. Rio de Janeiro: Editora Campus. 2004.

LACERDA Estéfane e CARVALHO André. **Introdução aos Algoritmos Genéticos**. Capítulo 3