

UC – CTC II

Programa Doutoral em Informática

Computação Natural

E

Inteligência Evolutiva

Professor: José Neves

Aluno: José Antônio da Cunha

Orientador: Cesar Analide

Braga, 03 de abril de 2014

Introdução a Teoria da Evolução Natural e Genética

Depois de muitos anos de observações e experimentos, Charles Darwin apresentou em 1858 sua teoria da evolução por meio da seleção natural, simultaneamente com outro naturalista inglês Alfred Russel Wallace. Em seguida, ele publica o livro “A origem das espécies por meio da seleção natural” com sua teoria completa, sustentada por muitas evidências colhidas durante suas viagens a bordo do navio Beagle (Darwin, 1859). Segundo Futuyama (1992), este trabalho influenciou muito diversas áreas de pesquisa, tais como da Biologia e Zoologia, mas também o pensamento religioso, filosófico, político e econômico. A teoria da evolução e a computação nasceram praticamente na mesma época: Charles Babbage, um dos precursores da computação moderna e amigo pessoal de Darwin, desenvolveu sua máquina analítica em 1833.

A moderna teoria da evolução combina a genética e as ideias de Darwin e Wallace sobre a seleção natural, criando o princípio básico de Genética Populacional: a variabilidade entre indivíduos em uma população de organismos que se reproduzem sexualmente é produzida pela mutação e pela recombinação genética. Este princípio foi desenvolvido durante os anos de 1930 e 1940, por biólogos e matemáticos (Resende, 2003).

A partir dos anos 1950 e 1960, muitos biólogos começaram a desenvolver simulações computacionais de sistemas genéticos. No entanto, foi John Holland quem começou, seriamente, a desenvolver as primeiras pesquisas sobre o tema. Holland foi gradualmente refinando suas ideias e em 1975 publicou o seu livro “Adaptation in Natural and Artificial Systems”, uma das referências básicas de Algoritmos Genéticos (Holland, 1992). Desde então, estes algoritmos vêm sendo aplicados com sucesso nos mais diversos problemas de otimização e Aprendizado de Máquinas (Goldberg, 1989).

Computacional Natural

As pesquisas em Computação Natural, tiveram início na década de 1950, e tratam de sistemas para a resolução de problemas que utilizam modelos computacionais baseados na teoria da evolução natural (Back, 1996 citado por Resende, 2003). Em seu trabalho Back (1996), propõe o agrupamento dos sistemas de Computação Evolutiva, também chamados de Algoritmos Evolutivos, em três grandes categorias: **Algoritmos Genéticos, Estratégias de Evolução e Programação Genéticas**.

Através de técnicas inspiradas na natureza, a inteligência computacional busca, o desenvolvimento de sistemas inteligentes, que imitem aspectos do comportamento humano, tais como: aprendizado, percepção, raciocínio, evolução e adaptação (Resende, 2003).

Pode-se considerar a Computação Natural, como o universo científico que, incorpora as áreas de pesquisas, como por exemplo Inteligência Computacional, Vida Artificial e Sistemas Dinâmicos Não-Linear.

Vida Artificial

É um campo de pesquisa que busca aumentar o papel da síntese no estudo dos fenômenos biológicos. Seu potencial é atribuído a possibilidade de, através de seu estudo, se desvendar segredos da vida e levantar a discussão sobre uma série de temas científicos, técnicos e psicológicos. Alguns dos campos nesta área incluem os seguintes tópicos: Auto-Organização e funcionalidade emergente; Definições e Origens da Vida; Auto-Reprodução; Vírus em Computadores; Evolução e Populações Genéticas; e Dinâmica Ecológica; Organização e comportamento de organismos coloniais e sociais; Agentes Autônomos (Robôs Autônomos e Agentes de Software); Inteligência Coletiva; Comportamento Animal; Ecossistemas Locais e Globais; etc.

Sistemas Dinâmicos Não-Lineares

Pode-se considerar os Sistemas Dinâmicos Não-Lineares (**SDNL**), como sendo processos em movimento ou processos que mudam no tempo ou no espaço (caos e fractal). Estes variam de sistemas muito simples a extremamente complexos e são encontrados com facilidade. A motivação para a pesquisa nesta área reside na habilidade em se prever a direção futura ou o estado de tais sistemas através do tempo ou espaço. Tipicamente, e sobre certas circunstâncias, mesmo os mais simples sistemas dinâmicos exibem um comportamento imprevisível e quase aleatório. Tal comportamento é denominado *caótico*. Alguns exemplos desses sistemas incluem: o movimento dos corpos celestes, os altos e baixos das bolsas de valores, as condições meteorológicas, etc.

Inteligência Computacional

Esta é uma área da ciência que estuda a teoria e a aplicação de técnicas inspiradas na Natureza, como *Redes Neurais*, *Lógica Nebulosa* e *Computação Evolucionária*.

Redes Neurais

De uma maneira simplificada, pode-se dizer que as Redes Neurais são modelos computacionais não lineares, inspirados na estrutura e operação do cérebro humano, que procuram reproduzir características humanas, tais como: *aprendizado*, *associação*, *generalização* e *abstração*. Devido à sua estrutura, as Redes Neurais são bastante efetivas no aprendizado de padrões a partir de

dados não-lineares, incompletos, com ruído e até compostos de exemplos contraditórios (Resende, 2003).

Lógica Nebulosa

A Lógica Nebulosa (*Fuzzy Logic*) tem por objetivo modelar o modo aproximado de raciocínio humano, visando desenvolver sistemas computacionais capazes de tomar decisões racionais em um ambiente de incerteza e imprecisão. A Lógica Nebulosa fornece um mecanismo para manipular informações imprecisas, tais como os conceitos de muito, pouco, pequeno, alto, bom, quente, frio, etc, fornecendo uma resposta aproximada para uma questão baseada em um conhecimento que é inexato, incompleto ou não totalmente confiável (Resende, 2003).

Computação Evolutiva

Esta é uma área de pesquisa interdisciplinar que compreende diversos paradigmas inspirados no princípio Darwiniano da evolução das espécies. O atual estágio de pesquisa considera, entre outros, os seguintes paradigmas:

- *Algoritmos Genéticos* (Genetic Algorithms - GA)
- *Programação Genética* (Genetic Programming - GP)
- *Hardware Evolucionário* (Evolvable Hardware - EH)

Algoritmos Genéticos

Os algoritmos genéticos são técnicas heurísticas de otimização global, ou seja, um método que encontra boas soluções a cada execução mas, não necessariamente a mesma todas as vezes.

Eles são baseados nas teorias da genética da biologia de onde extrai todo o seu embasamento. Basicamente, nos algoritmos genéticos, populações de indivíduos são criadas e submetidas aos operadores de seleção, cruzamento, mutação. Estes operadores utilizam uma caracterização de qualidade de cada indivíduo como solução do problema em questão chamada de **avaliação** e vão gerar um processo de evolução natural destes indivíduos, que eventualmente deverá gerar um indivíduo que caracterizará uma boa solução, talvez até mesmo a melhor, para o problema que está sendo atacado (Linden, 2006).

Os algoritmos genéticos também são considerados boas técnicas para mineração de dados num processo de descobrimento de conhecimento em base de dados.

Alguns problemas que podem ser atacados com algoritmos genéticos:

- Alocação de recursos;
- Planejamento de Expansão;
- *Unit Commitment*;
- Alocação de Capacitores;
- Bioinformática

Para (Linden, 2006), os Algoritmos Genéticos (**GA**) fazem parte do conjunto dos Algoritmos Evolucionários. Esses usam modelos computacionais dos processos naturais de evolução como uma ferramenta para resolver problemas.

Os algoritmos evolucionários funcionam mantendo uma população de estruturas, denominadas indivíduos ou cromossomos, que se comportam de forma semelhante às evoluções das espécies. A estas estruturas são aplicadas operadores genéticos como recombinação e mutação entre outros. Cada indivíduo recebe uma avaliação que é uma quantificação de sua qualidade como solução para um problema em questão. Com base nesta avaliação serão aplicados os operadores genéticos de forma a simular a sobrevivência do mais apto (Linden, 2006).

Os algoritmos evolucionários se posicionam como técnicas de busca. Eles são parte de um ramo de busca chamado de “Técnicas Aleatórias Guiadas”, isto é, eles têm componentes aleatórios, mas usam as informações do estado corrente para guiar esta pesquisa. Como se pode ver na Figura 2, os algoritmos evolucionários se dividem em vários tipos distintos.

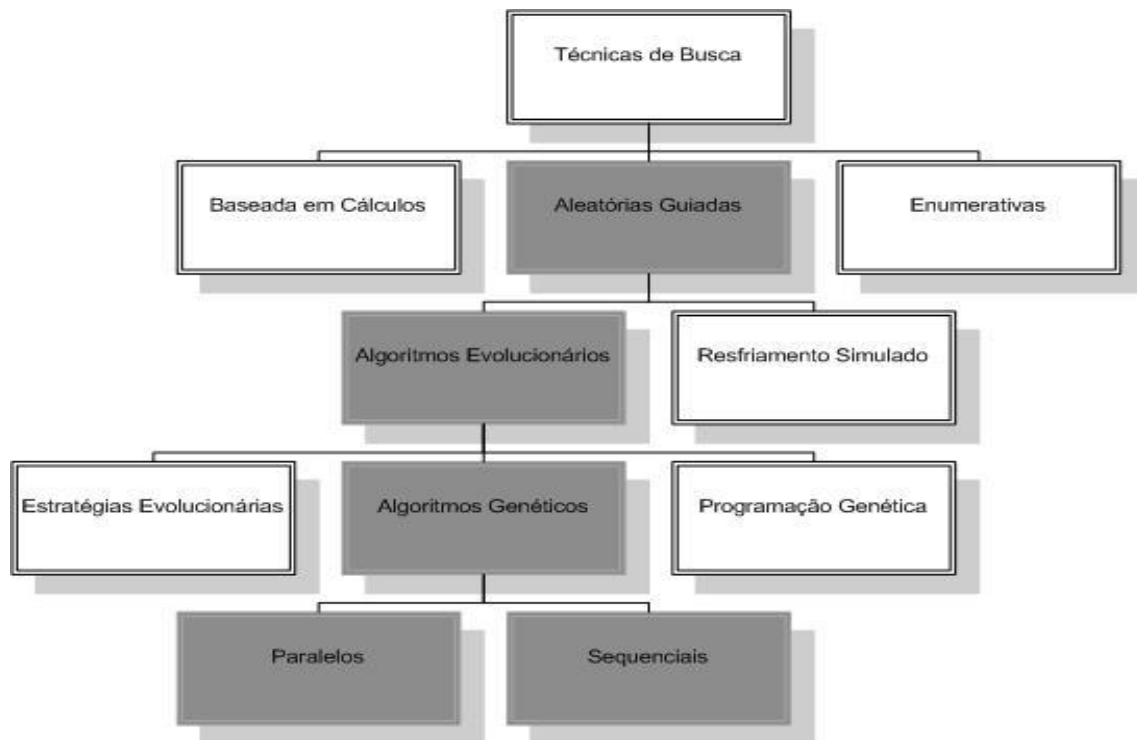


Figura 1: Hierarquia de Técnicas de Busca

Fonte: Linden, 2006

Os algoritmos genéticos (**GA**) são um ramo dos algoritmos evolucionários e como tal podem ser definidos como uma técnica busca baseada em uma metáfora do processo biológico de evolução natural.

Segundo (Linden, 2006), são modelos computacionais de busca e otimização de soluções em problemas complexos. São técnicas que procuram obter boas soluções por meio da evolução de populações de indivíduos que foram criados.

Para (Linden, 2006), os indivíduos depois de criados são submetidos aos operadores genéticos de recombinação e/ou mutação. Estes operadores utilizam uma caracterização de qualidade de cada indivíduo como solução do problema em questão chamada de avaliação e vão gerar um processo de evolução natural destes indivíduos, que eventualmente deverá gerar um indivíduo que caracterizará uma boa solução para o problema.

Os Algoritmos Genéticos empregam um processo adaptativo e paralelo de busca de soluções em problemas complexos. O processo é adaptativo, pois as soluções existentes a cada instante influenciam a busca por futuras soluções. O paralelismo do processo é decorrência natural do fato de que diversas soluções são consideradas a cada momento pelos GA (Linden, 2005).

Para (Linden, 2006), os GA são técnicas heurísticas, ou seja, eles não encontram a solução ótima para um problema, e quando o faz nem sempre consegue repetir o feito. Eles não ficarão estagnados pelo fato de terem

encontrado a solução melhor possível. Nesse ponto, os GA se parecem com a evolução natural, que só porque encontrou um indivíduo que é instantaneamente o melhor de certo grupo não para de procurar outros indivíduos ainda melhores. Na evolução natural isto também decorre de circunstâncias que mudam de um momento para o outro. Uma bactéria pode ser a melhor em um ambiente livre de antibióticos, mas quando esses são usados, outras que antes eram menos fortes tornam-se as únicas sobreviventes por serem as únicas adaptadas.

No caso dos algoritmos genéticos, o ambiente normalmente é um só. Entretanto, conforme as gerações vão se passando e os operadores genéticos vão atuando, faz-se uma grande busca pelo espaço de soluções, busca esta que seria realizada pela evolução natural se elas ficassem permanentemente em um ambiente imutável.

Mas, ao contrário do que se possa pensar, o processo de evolução natural não é dirigido à obtenção da solução ótima. Na verdade, o processo simplesmente consiste em fazer competir uma série de indivíduos e pelo processo de sobrevivência do mais apto, os melhores indivíduos tendem a sobreviver. Um GA tem o mesmo comportamento que a evolução natural: a competição entre os indivíduos é o que determina as soluções obtidas. Eventualmente, devido à sobrevivência do mais apto, os melhores indivíduos prevalecerão.

Sendo assim, Para (Linden, 2006), apesar de seu nome implicar no contrário, os GA não consistem algoritmos de busca da solução ótima de um problema, mas sim uma heurística que encontra boas soluções a cada execução, mas não necessariamente a mesma todas às vezes.

Segundo Linden (2006), os GA, em geral, são compostos pelos seguintes componentes: problema, representação do cromossomo, avaliação, seleção, operadores genéticos e técnicas e parâmetros.

Para (Linden, 2006), se pode dizer que os Algoritmos Genéticos são técnicas de busca com as seguintes características positivas:

- ✓ **Paralela** – pois mantém uma população de soluções que são avaliadas simultaneamente.
- ✓ **Global** – os Algoritmos Genéticos não usam apenas informação local, logo, não necessariamente ficam presos em máximos locais como certos métodos de busca. Esta característica é uma das mais interessantes dos GA e faz com que eles sejam uma técnica extremamente adequada para funções multidimensionais e de perfis complexos como a maioria das funções de custo associadas a problemas reais.

- ✓ **Não totalmente aleatória** – existem métodos que usam apenas variáveis aleatórias para realizar suas pesquisas. GA têm componentes aleatórios, mas como usam as informações da população corrente para determinar o próximo estado da busca, não podem ser considerados totalmente aleatórios.
- ✓ **Não afetada por descontinuidades na função ou em suas derivadas** – Os GA não usam informações de derivadas na sua evolução nem necessitam de informação sobre o seu entorno para poder efetuar sua busca. Isto faz com que sejam muito adequados para funções com descontinuidades ou para as quais não temos como calcular uma derivada.
- ✓ **Capaz de lidar com funções discretas e contínuas** – os GA são capazes de lidar com funções reais, discretas, booleanas e até mesmo categóricas, sendo possível inclusive misturar as representações sem prejuízo para a habilidade dos GA de resolver problemas (Linden, 2006, pg. 49).

Tendo em vista que os **GA** são orientados a buscas direcionadas e inteligentes, eles são boas técnicas para atacar problemas de busca com espaços muito grandes, que não podem ser resolvidos por técnicas tradicionais. Como exemplo de problemas para aplicação de Algoritmos genéticos tem: Otimização de funções matemáticas, otimização combinatorial, otimização de planejamento, problema do caixeiro viajante, problema de otimização de rota de veículos, otimização de distribuição e logística, seleção de variáveis em mineração de dados, dentre outros.

Estrutura do GA

O algoritmo genético pode ser resumido algoritmicamente através dos seguintes passos:

- a) Inicialize a população de cromossomos.
- b) Avalie cada cromossomo na população.
- c) Selecione os pais para gerar novos cromossomos.
- d) Aplique operadores genéticos.
- e) Apague os velhos membros da população.
- f) Avalie todos os novos cromossomos e insira-os na população.

- g) Se a condição de parada retorne o melhor indivíduo da população, se não retorne ao passo c. (Linden, pg. 53, 2006).

Esta é apenas uma visão de alto nível. Entretanto, como o **GA** é altamente genérico, vários de seus componentes são invariáveis de um problema para o outro. Isto favorece a sua implementação em uma linguagem orientada a objetos, permitindo o reaproveitamento do código para a solução de vários problemas diferentes (Linden, 2006).

Implementando uma solução com GA

Para que o algoritmo genético funcionar existe alguns aspectos que devem ser avaliados e especificados para que o algoritmo tenha o seu desempenho satisfatório. Estes aspectos são: a representação cromossomial adequada ao problema e uma função de avaliação que avalie satisfatoriamente o grau de adequação de cada indivíduo como solução do problema.

Representação do Cromossomo

A representação do cromossomo é fundamental para o **GA**. Basicamente, ela consiste em uma maneira de traduzir a informação do nosso problema em uma maneira viável de ser tratada pelo computador. Quanto mais ela for adequada ao problema, maior a qualidade dos resultados obtidos (Linden, 2006).

Cada pedaço indivisível desta representação é chamado de **gene**, por analogia com as partes fundamentais que compõe as partes fundamentais de um cromossomo biológico.

Segundo Linden (2006), é importante notar que a representação do Cromossomo é completamente arbitrária ficando a sua definição de acordo com o gosto do desenvolvedor e com a adequação ao problema. É interessante apenas que algumas regras gerais sejam seguidas:

- a) A representação deve ser a mais simples possível
- b) Se houver soluções proibidas para o problema, então elas não devem ter uma representação.
- c) Se o problema impuser condições de algum tipo, estas devem estar implícitas dentro da representação (Linden, 2006, pg. 54).

Função de Avaliação

A função de avaliação é a maneira pela qual é utilizada pelo **GA** para determinar a qualidade de um indivíduo como solução de um problema a ser resolvido. Pode-se entendê-la mais facilmente se a visualizar como uma nota dada ao indivíduo na resolução do problema. Esta nota será usada para a escolha de indivíduos pelo módulo de seleção, sendo a forma de diferenciar entre as boas e as más soluções para um problema (Linden, 2006).

Dada a generalidade dos **GA**, a função de avaliação, em muitos casos, é a única ligação verdadeira do sistema como o problema real. Isto decorre do fato que a função de avaliação só julga a qualidade da solução que está sendo apresentada por aquele indivíduo, sem armazenar qualquer tipo de informação sobre as técnicas de resolução do problema.

O mais importante conceito a ter em mente é que a função de avaliação deve refletir os objetivos a serem alcançados na resolução do problema e é derivada diretamente das condições impostas por esse (Linden, 2006).

População

Nos Algoritmos Genéticos, a População representa o conjunto de soluções geradas por ele.

O tamanho da população determina o número de soluções que o **GA** gera, a cada iteração de busca, até que uma condição seja satisfeita. Este tamanho afeta diretamente o desempenho global e a eficiência do algoritmo (Linden, 2006). Caso o número de indivíduos na população seja muito pequeno, não haverá espaço para uma variedade genética suficientemente grande o que fará com que o GA seja incapaz de achar boas soluções e caso o número seja muito grande o algoritmo demorará muito e se pode estar aproximando-se de uma busca exaustiva.

Seleção dos Pais

O método de seleção simula o mecanismo de seleção natural que atua sobre as espécies biológicas, em que os pais mais capazes geram mais filhos, ao mesmo tempo em que os pais menos aptos também conseguem gerar descendentes (Linden, 2006).

Tem que se privilegiar os indivíduos com função de avaliação alta, sem desprezar aqueles indivíduos com função de avaliação extremamente baixa. Esta se dá, pois, mesmo esses indivíduos podem ter características genéticas que sejam favoráveis à criação de um indivíduo que seja a melhor solução para o problema que está sendo atacado.

Vantagens e Desvantagens dos Algoritmos Genéticos

Os algoritmos genéticos são uma ferramenta para efetuar buscas em espaços muito grande de soluções e suas principais vantagens são:

- ✓ São métodos robustos, apresentando bons resultados em espaços de busca complexos.
- ✓ É uma técnica que estabelece uma analogia com a natureza, são simples de serem implementados, pois não é necessário o conhecimento matemático profundo sobre o problema.
- ✓ Trabalham com todo o conjunto de soluções gerado e não apenas com um ponto o que permite realizar buscas em vários espaços.
- ✓ Apresenta bons resultados na resolução de problemas de otimização de grande escala.
- ✓ Utiliza regras de transação probabilística ao invés de determinística.

Mas os algoritmos genéticos não são a solução milagrosa para todos os problemas computacionais existentes. Na verdade eles se limitam a problemas como espaços de busca muito grandes. Ainda existem desvantagens como:

- ✓ Dificuldade em se encontrar a solução ótima.
- ✓ Necessidade de várias avaliações de valores de aptidão
- ✓ Dificuldade na escolha da representação da solução (Cromossomo).

Por essas características é importante saber se o seu problema pode ser resolvido por um algoritmo genético, e não adaptá-lo a uma forma de algoritmo genético, pois assim sua solução pode ser prejudicada.

Programação Genética

A Programação Genética é uma técnica automática de programação que propicia a evolução de programas de computadores que resolvem (ou aproximadamente resolvem) problemas.

Hardware Evolucionário

Hardware Evolucionário utiliza conceitos dos sistemas evolucionários naturais no projeto automático de circuitos, hardware auto reparáveis, projeto de robôs e projeto de circuitos **VLSI**. Hardware Evolucionário é uma extensão do modelo genético de aprendizado no espaço de estruturas complexas como circuitos eletrônicos.

Resumindo

De forma bastante resumida, pode-se dizer que, a Computação natural, é um ramo da ciência da computação destinado a estudar, compreender e aplicar, padrões complexos encontrados na natureza, utilizando-os como base para resolução de problemas, desenvolvimento de novas tecnologias e aperfeiçoamento de sistemas já existentes. Dentre as tecnologias criadas tendo como base conceitos encontrados na natureza, podemos citar o velcro (observando-se as plantas), sonares (observando-se os morcegos), submarinos (observando-se os peixes), entre outros. Além de exemplos práticos, a observação da natureza permitiu também o desenvolvimento de teorias a respeito de como a natureza opera, como, por exemplo, as leis da termodinâmica. Computação natural nada mais é do que uma versão computacional do processo citado anteriormente, em que ideias são obtidas através da observação da natureza.

Bibliografia

Futuyama, D. J. Biologia Evolutiva (2. Ed.). São Paulo: Sociedade Brasileira de Genética, 1992.

Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989

Holland, J. Adaptation in Natural and artificial systems (2. Ed.). Ann Arbor, MI: The University of Michigan Press

Linden, Ricardo. Algoritmos genéticos. Rio de Janeiro: Brasport, 2006

Resende, Solange Oliveira. Sistemas Inteligentes: fundamentos e aplicações. Barueri, SP: Manole, 2003.

Sites:

<http://www.ica.ele.puc-rio.br/compint/index.rails?name=Intelig%C3%A2ncia+Computacional>

Gladstone B. Alves. "Introdução ao Hardware Evolutivo" *Doxa 6* (2001): 55-64. Available at:

<http://works.bepress.com/gladstone/2>

H. Horner, "A C++ class library for genetic programming: The Vienna University of Economics - genetic programming kernel", *Internet Draft*, maio de 1996.

<http://www.wu-wien.ac.at/usr/h88/h8850092>.

Marcio Nunes de Miranda. Algoritmos Genéticos: Fundamentos e Aplicações

<http://www.nce.ufrj.br/GINAPE/VIDA/alggenet.htm>

Maxwel Macedo Dias et al. Aplicação de Técnicas de Mineração de Dados no Processo de Aprendizagem na Educação a Distância.

<file:///C:/Users/Administrador/SkyDrive/Estado%20da%20arte/Minera%C3%A7%C3%A3o%20de%20Dados%20em%20EAD%20Goldschimit.pdf>

Fernando J. Von Zuben. Computação Evolutiva: Uma Abordagem Pragmática.

<file:///C:/Users/Administrador/SkyDrive/Estado%20da%20arte/Computa%C3%A7%C3%A3o%20Evolutiva%20-%20tutorialEC.pdf>