

História e terminologia a respeito da computação evolutiva

Adriano Donizete Pila*

Mestre em Ciência da Computação - USP
Coordenador e professor do Curso de Ciência da Computação da Faculdade Comunitária de Santa Bárbara
e-mail: adriano.pila@unianhanguera.edu.br

■ Resumo

Muitos são os termos para designar os programas computacionais desenvolvidos sob o paradigma evolutivo. Esses termos foram aparecendo ao longo da linha histórica da computação evolutiva, quase que em paralelo, devido a núcleos de desenvolvimento distintos. O objetivo deste trabalho é apresentar o que motivou o aparecimento da Computação Evolutiva, uma breve conceituação a respeito da seleção natural e evolução, bem como traçar uma linha histórica sobre a computação evolutiva em termos de sua origem, funcionamento e aplicações.

Palavras-chave: Computação Evolutiva.

■ Abstract

There are a lot of terms to name the computational programs developed under the evolutionary paradigm. These terms came up during the evolutionary computation historical line, almost at the same time, because of distinct development groups. The aim of this work is to present the reason that made the Evolutionary Computation came up, some concepts about natural selection and evolution as well as a historical line about the evolutionary computation in terms of its birth, how it works and some applications.

Key-word: Evolutionary Computing.

■ Introdução

A Computação Evolutiva (CE) utiliza-se de modelos e funcionalidades naturais para melhorar o desempenho de sistemas computacionais. O preceito da CE esta calcado nas descobertas feitas por Charles Darwin e sua Teoria da Evolução das Espécies, na qual o mecanismo fundamental que guia a evolução é a seleção natural (Darwin, 1859). A CE é uma área muito ampla que envolve os Algoritmos Genéticos (AGs), Programação Genética (PG), Estratégias Evolutivas (EE), Programação Evolutiva (PE) e os Sistemas de Classificação (SC). Cada qual possui sua peculiaridade nas representações dos indivíduos e funcionalidades dos operadores, mas a base modelada segundo a teoria de Darwin é semelhante para todas. Na CE o problema computacional é representado de forma a tirar vantagem do processo de seleção natural (implementado na forma de um método de seleção artificial), de forma que várias soluções candidatas concorrem entre si e são recombinadas até originarem a melhor solução (Mitchell, 2000). Devido à grande gama de terminologias que apareceram durante a linha histórica, muitos dos termos são confundidos e outros são apenas termos novos para designar conceitos idênticos. Assim, neste trabalho serão apresentadas as motivações para o aparecimento da CE,

*Bolsistas FUNADESP - Fundação Nacional de Desenvolvimento do Ensino Superior Particular

os conceitos advindos da Teoria da Evolução das Espécies, bem como as principais diferenças entre as áreas que compõem a Computação Evolutiva.

■ Métodos de Otimização

Os problemas do mundo real são representados no mundo computacional por meio de modelos que variam de equações matemáticas às estruturas complexas com múltiplos apontadores. Após a representação desses problemas muitas tarefas podem ser desempenhadas, como simulações, visualizações, integrações com outros modelos, etc. Uma dessas tarefas consiste em otimizar o problema representado pelo modelo, dando assim origem a uma área que trata dos Problemas de Otimização.

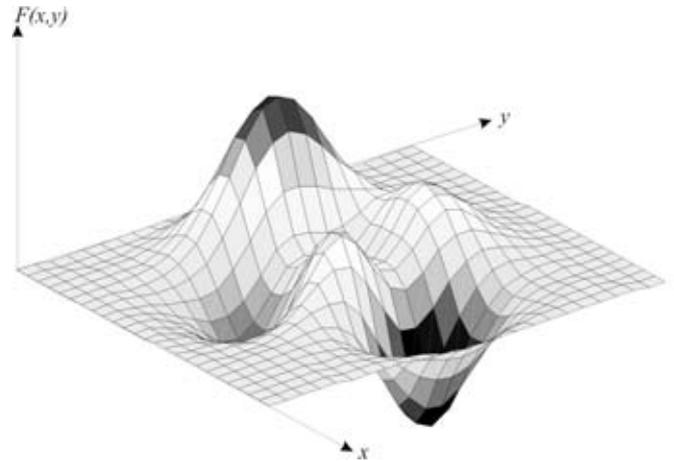
Muitos estudos foram feitos e diversos métodos de otimização foram desenvolvidos em virtude desses estudos. Existem basicamente três classes de métodos:

1. baseados em cálculo;
2. enumerativos; e,
3. aleatórios.

Os métodos de otimização baseados em cálculo têm sido exaustivamente estudados, os quais se subdividem em duas classes: indiretos e diretos. Métodos indiretos buscam por extremos locais pela solução do sistema de equações não-lineares ajustando o gradiente da função objetivo como zero (Goldberg, 1989). Dada uma função contínua e sem restrições - Figura 1 - a busca por um possível ponto de pico começa pela restrição da procura naqueles pontos cuja inclinação em todas as direções é igual à zero. Por outro lado, os métodos diretos buscam por ótimos locais movendo-a na direção do gradiente local. Essa é a noção de busca usada pelo método *hill-climbing*, o qual encontra o máximo local escalando na direção do maior pico. Embora os métodos baseados em cálculo tenham sido aprimorados, algumas razões simples mostram sua falta de robustez.

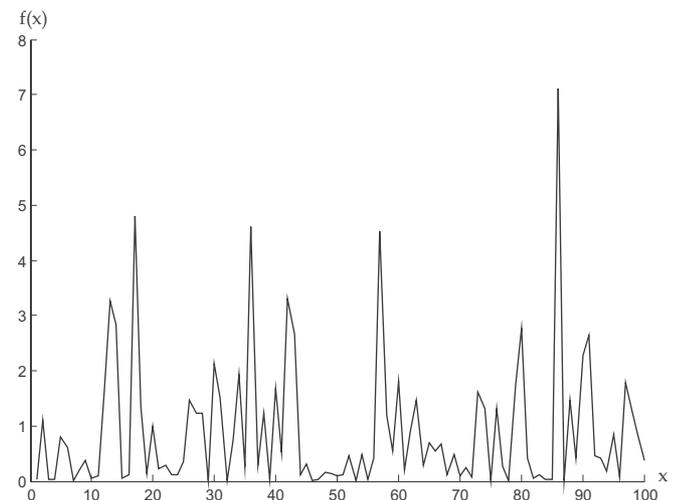
Primeiramente, ambos os métodos, indiretos e diretos, procuram por pontos locais na vizinhança do ponto corrente. Se a busca iniciar na vizinhança do menor pico, considerando como exemplo a Figura 1, isso fará com que o ponto máximo não seja atingido. Ainda, ambos os métodos assumem que as funções são contínuas e, portanto admitem derivadas para a obtenção do gradiente. Entretanto, os espaços de busca dos problemas do mundo real são compostos por funções descontínuas, multimodais e ruidosas, tal como a

Figura 1 - A função $f(x,y)$ possui pontos de mínimo e máximo obtidos por meio da combinação precisa dos parâmetros x e y .



representada na Figura 2. Não há dúvidas que métodos que dependam da existência de continuidade e derivadas são aplicáveis apenas a um domínio muito restrito (Haupt, 1998).

Figura 2 - A função $f(x)$ possui pontos de mínimos e máximos que não podem ser encontrados por métodos que utilizem o gradiente da função.



Esquemas enumerativos foram desenvolvidos e considerados em muitos formatos. A idéia é bastante simples: dentro de um espaço finito de busca, ou um espaço de busca discretizado e infinito, o algoritmo de busca inicia procurando por valores da função objetivo em todo o espaço de busca, um de cada vez. Embora esse esquema de busca seja muito simples, seu comportamento na busca é semelhante a dos humanos quando o espaço de busca é pequeno. Entretanto, esse método torna-se extremamente ineficiente quando

aplicado a espaços de busca maiores.

A terceira classe de métodos de busca são os métodos aleatórios. Nessa classe os parâmetros da função assumem valores aleatórios que são armazenados e comparados uns com os outros em relação ao valor da função objetivo. Na verdade, os métodos de busca aleatórios possuem uma eficiência muito próxima à dos métodos enumerativos. Deve-se fazer uma distinção entre os métodos aleatórios e os métodos que utilizam técnicas aleatórias. Os Algoritmos Genéticos são um exemplo desses últimos métodos, os quais utilizam escolhas aleatórias para guiar métodos altamente exploratórios.

Um outro método de busca bastante conhecido é o *simulated annealing*. O nome vem do processo de enrijecimento na metalurgia, que é uma técnica que envolve o aquecimento e o esfriamento controlado do material para a melhoria das suas propriedades. Nesse processo, o metal atinge sua configuração mais estável, minimizando sua energia interna total. O princípio do método *simulated annealing* é similar. Cada ponto no espaço de busca tem uma energia associada, que indica o quão bom aquele ponto é na solução do problema. O objetivo é encontrar o ponto com a menor energia. O essencial a ser notado é que métodos de busca aleatórios não implicam necessariamente em buscas sem direção (Koza, 1992).

Um método de busca mais recente é aquele baseado em colônias de formigas no qual há uma interação entre os membros da colônia como forma de cooperação, adaptação e solução de problemas (Parpinelli, 2002).

Os três métodos aqui citados - Algoritmos Genéticos, *Simulated Annealing* e Algoritmos baseados em Colônias de Formigas - são métodos de busca que vem sendo amplamente empregados em espaços de busca complexos, nos quais outras técnicas mais específicas não são factíveis. O emprego desses métodos, com destaque para os AGs, se deve ao fato de não ser necessária à modelagem de funções matemáticas que representem o problema e nem o cálculo de derivadas e gradientes. Esses três métodos foram criados com base na observação de fenômenos naturais que se mostram muito eficazes quando aplicados na natureza e, portanto, crê-se, que o mesmo desempenho seja possível quando aplicados a problemas modelados computacionalmente (Mitchell, 1998).

Dessa forma, uma vez que existia uma lacuna entre

os problemas do mundo real e os métodos de otimização baseados em cálculo de derivadas e gradiente, pensou-se em simular nos computadores o mecanismo evolutivo e de seleção que ocorre na natureza. Na próxima seção é dada uma visão geral do que vem a ser a evolução e a seleção natural.

■ Seleção Natural e Evolução

Nos anos que antecederam o século XIX, a única teoria aceita em relação às diferentes espécies animais e vegetais existentes no planeta é chamada *Criacionismo*. Segundo essa teoria todas as espécies animais e vegetais são obras Divinas e suas características atuais são as mesmas daquelas iniciais quando Deus as criou. No entanto, esta teoria deixa uma lacuna em relação ao aparecimento de novas espécies e à extinção de outras. Para complementar o Criacionismo, surgiu uma outra teoria denominada *Catastrofismo*. Nessa teoria, que procurava não entrar em atrito com a Bíblia, e ainda condizia com o Dilúvio, novas espécies surgiam e outras eram extintas devido à catástrofes naturais.

No início do século XIX surgiram muitas teorias que contradiziam as demais teorias aceitas até aquele momento. Uma das mais importantes foi aquela desenvolvida por Jean-Baptiste Lamarck (1744 - 1829), cujo pressuposto do surgimento de novas espécies estava relacionado à “*Lei do Uso e Desuso*”, segundo a qual, as características dos indivíduos eram melhoradas de acordo com seu uso repetitivo ou, ainda, essas características eram enfraquecidas, ou até mesmo removidas, por causa do desuso. Essas características modificadas eram então passadas para as gerações futuras dos descendentes desses indivíduos. Em 1830, o geólogo inglês Sir Charles Lyell refutou a Teoria do Catastrofismo, mas não fez o mesmo com a teoria de Lamarck, denominada *Lamarckismo*.

A grande modificação em relação às teorias aceitas e como novas espécies surgem e outras são extintas mudou completamente em 1859. Foi neste ano que o cientista inglês Charles Darwin - Figura 3 - publicou sua obra mais famosa, intitulada “*Teoria da Evolução das Espécies*” (Darwin, 1859). Segundo sua teoria, todos os indivíduos de uma população são diferentes. Devido a essas diferenças, alguns deles são mais bem adaptados ao ambiente, se comparados aos demais e, portanto, possuem melhores chances de sobreviver e procriar. Essas características vantajosas são herdadas pelos

indivíduos das gerações descendentes, tornando-se através do tempo características predominantes na população. Essa teoria ficou conhecida como *Darwinismo*.

Figura 3 - Cientista inglês Charles Robert Darwin (1809 - 1882), criador da Teoria da Evolução das Espécies.



Seguindo os preceitos do Darwinismo, podemos definir o termo *evolução* como sendo as mudanças ocorridas em uma espécie ao longo do tempo devido a pressões do ambiente no qual essa espécie vive. Assim, as espécies sobreviventes são aquelas que ao longo do tempo evoluem satisfazendo as pressões do ambiente, enquanto que as espécies extintas são aquelas incapazes de satisfazer essas pressões. A esse mecanismo evolutivo em função das pressões do ambiente dá-se o nome de *seleção natural*.

A seleção natural é o mecanismo essencial para a evolução cuja explicação foi proposta por Charles Darwin e, nos dias de hoje, ainda é aceita pela comunidade acadêmica como a melhor explanação para o aparecimento e extinção das espécies.

O conceito envolvido na seleção natural está no fato que as condições ambientais determinam como certas características particulares dos organismos servem para a sobrevivência e reprodução desses organismos. Organismos que não possuem características favoráveis em relação ao meio em que vivem podem não se reproduzirem, ou até mesmo morrerem. Como as condições ambientais permanecem as mesmas, ou similares o suficiente para serem favoráveis àquelas características, estas tendem a serem mais comuns dentro da população.

A teoria da evolução das espécies pela seleção

natural, proposta por Darwin, parte da premissa que as características dos organismos variam de uma forma não-determinística dos parentes para seus descendentes. Esse processo é chamado individualização. Embora a teoria de Darwin não explique como esse processo ocorre, descobertas científicas recentes no campo da genética mostram seu mecanismo de funcionamento (Wikipedia, 2003d).

O mecanismo da seleção natural não faz qualquer distinção relacionada às características serem favoráveis em relação ao ambiente ou em relação à aptidão reprodutiva. O mecanismo de seleção natural simplesmente considera que se uma variação particular nos organismos manifestar um melhor poder de sobrevivência ou reprodução, seus descendentes terão mais chances de sobreviver e procriar do que os demais organismos que não manifestaram aquela variação. As características originais irão desaparecer, assim como qualquer variação que implique em má adaptação. Então, certas características são preservadas devido às vantagens seletivas que elas provêm aos indivíduos, permitindo assim que esses indivíduos produzam mais descendentes que os indivíduos que não as possuem. Eventualmente, após muitas iterações desse processo, os organismos irão desenvolver características adaptativas cada vez mais complexas.

O que torna uma característica mais propensa ao sucesso é altamente dependente dos fatores ambientais. Quando membros de uma espécie são separados eles enfrentam diferentes condições ambientais, tendendo a se desenvolverem em direções diferentes. Após um grande período, suas características terão se desenvolvido através de caminhos diferentes, e os indivíduos anteriormente separados não poderão mais se reproduzir. Nesse ponto eles são considerados espécies distintas. Esta é a razão pela qual uma espécie às vezes se separa em múltiplas espécies, ao invés de simplesmente ser substituída por uma nova espécie.

Conforme explanada no livro *Teoria da Evolução das Espécies* (Darwin, 1859), a seleção natural pode ser expressa como a seguir:

1. Se *Existem organismos que reproduzem, e*
2. Se *Os descendentes herdarem características de seus progenitores, e*
3. Se *Existe variabilidade das características,*
e
4. Se *O ambiente não pode suportar todos os membros de uma população em crescimento*

5. Então *Os membros da população com características menos adaptativas irão morrer, e Os membros com características mais adaptativas irão ter sucesso.*

O resultado desse processo é a evolução das espécies. Note que este é um processo contínuo e considera como as espécies sofrem mudanças, podendo tanto levar à extinção de espécies quanto à criação de novas espécies. Note também que a lei acima não se aplica somente a organismos biológicos. Ela se aplica a todos os organismos que reproduzem numa forma que envolva herança e variabilidade (Darwin, 1859). Portanto, uma forma de seleção natural poderia ocorrer no mundo não-biológico – *i.e.* Computação Evolutiva em geral.

Na próxima seção são apresentados os primeiros relatos do uso desses conceitos como forma de contribuir para a solução de problemas computacionais.

■ Revisão Histórica

Os primeiros relatos do uso de modelos computacionais para o melhor entendimento do processo natural da evolução começou na década de 50. As primeiras descrições do uso desses modelos aparecem nos artigos de Friedberg (1958) e Friedberg (1959). Esses trabalhos representam os primeiros passos da computação evolutiva e aprendizado de máquina. Outro trabalho importante da época foi o desenvolvido por Fisher (1958), segundo o qual a evolução é uma forma de adaptação tal qual ocorre no aprendizado, diferindo tão somente quanto à escala de tempo. Ainda, houve o trabalho publicado por Bremermann (1962), que se utilizou da simulação evolutiva como forma de otimização de problemas numéricos. Bremermann também desenvolveu alguns dos primeiros algoritmos evolutivos (Bremermann, 1965).

Antes do uso da seleção natural nos métodos baseados em computação evolutiva, vários biólogos utilizaram computadores para executar simulações de sistemas genéticos (Barricelli, 1957, 1962; Fraser, 1960, 1962; Martin, 1960). Embora esses estudos fossem devotados ao estudo dos fenômenos naturais, os trabalhos de Fraser não estavam tão distantes da noção moderna de algoritmo genético. Em seu trabalho ele utilizou um conjunto formado por 15 bits. Embora sua representação sugira o que atualmente é conhecido como algoritmo genético, nos trabalhos de Fraser não ficou

evidenciado que as idéias de seus trabalhos estavam baseadas nos algoritmos de busca por seleção natural (Goldberg, 1989).

Todos os trabalhos desenvolvidos até então contribuíram para o desenvolvimento da computação evolutiva. Entretanto, na metade da década de 60 foi desenvolvida a base do que hoje se pode identificar como as três principais áreas dentro da computação evolutiva (De Jong, 2000). A base da PE foi desenvolvida por Fogel (1966) em San Diego, Califórnia. O início dos AGs ocorreu na Universidade de Michigan em Ann Arbor por Holland (1967). Enquanto que, do outro lado do Atlântico, Estratégias Evolutivas foi um desenvolvimento conjunto de um grupo de três estudantes, Bienert, Rechenberg e Schwefel, em Berlin (Rechenberg, 1965).

Na Figura 4 são apresentadas as subáreas envolvidas na CE. Essa é uma visão de como a CE esta subdividida, a qual foi esquematizada seguindo descrições contidas em várias publicações que são referência para essa área (Goldberg, 1989; Michalewicz, 1997; Haupt, 1998; Mitchell, 1998; Bäck, 2000).

Figura 4 - A Computação Evolutiva pode ser vista como uma grande área que envolve todas as áreas cujo desenvolvimento está calcado no preceito da evolução.



A seguir são brevemente descritas cada uma das sub-áreas que compõem a Computação Evolutiva.

■ Programação Evolutiva

Na sua forma padrão, um programa evolutivo utiliza-se das quatro características principais de um algoritmo evolutivo:

1. inicialização;
2. variação;

3. avaliação;
4. seleção.

Para um programa evolutivo o aprendizado é decorrente de um processo evolutivo que retém os indivíduos de sucesso que são treinados por meio de um processo estocástico de tentativa e erro. A função de avaliação mede diretamente a aptidão, ou equivalentemente o erro comportamental, de cada membro da população com relação ao ambiente. A seleção, retira probabilística-mente as soluções sub-ótimas da população (Porto, 2000).

Um algoritmo básico de PE começa com uma população de soluções que são inicializadas de forma aleatória, heurística ou outro método apropriado. O tamanho da população pode variar, mas geralmente seu tamanho permanece constante ao longo do processo. Cada uma das soluções é avaliada com respeito a uma função de avaliação específica. Após a criação da população de soluções iniciais, cada um dos indivíduos é alterado pela aplicação do processo de mutação. Uma característica forte que diferencia PE das demais abordagens é a não adoção de um mecanismo de recombinação. Cada um dos indivíduos gera descendentes, os quais são replicados com um mecanismo estocástico de erro (mutação).

Programação Evolutiva difere das outras técnicas da Computação Evolutiva, tal como Algoritmos Genéticos, de uma maneira crucial. Programação Evolutiva é uma abordagem de otimização *top-down* ao invés de uma abordagem *bottom-up*. A idéia de que a soma de partes ótimas raramente levam a uma solução ótima geral é a chave da diferença nessa abordagem. Algoritmos Genéticos são baseados na identificação, combinação e sobrevivência dos melhores blocos de construção (esquema), os quais são iterativamente combinados para formar blocos de construção maiores e melhores. Na hipótese dos blocos de construção há uma premissa implícita que pressupõe a separação da função de aptidão ao longo do espaço de busca, tornando os AGs um método de otimização localmente focalizado. Já na PE a abordagem é global. Em PE, da mesma forma que em EE, o operador de variação permite a mudança simultânea de todas as variáveis ao mesmo tempo. A aptidão, descrita em termos do comportamento de cada indivíduo da população, é avaliada indiretamente, sendo a base da sobrevivência de cada indivíduo (Porto, 2000).

A PE foi desenvolvida por Lawrence J. Fogel em

1960. Naquela época a inteligência artificial estava concentrada principalmente na busca por heurísticas e simulação de redes neurais. Para Fogel estava claro que essas abordagens eram limitadas pelo fato de modelar os humanos ao invés de modelar a essência maior que produz a evolução das criaturas. Fogel considerava inteligência como sendo o comportamento adaptativo que levava à satisfação de uma gama de metas. Na verdade, a predição era vista como sendo a chave para o comportamento inteligente e isso levou a uma série de experimentos que foram relatados e publicados. Em De Jong (2000) são descritas algumas dessas publicações.

Desde o aparecimento da PE muito esforço vem sendo feito de forma a otimizar seus mecanismos. Em especial, nas décadas de 80 e 90 houve um grande avanço e interesse por parte da comunidade científica no que se refere aos algoritmos evolutivos. Na literatura encontram-se muitas aplicações da PE, tais como treinamento, construção e otimização de redes neurais, otimização de rotas (em duas, três ou mais dimensões), projeto de novos medicamentos, automação de controles e teoria dos jogos. Notavelmente, aplicações de sucesso incluem problemas de ordem NP, descritos em Porto (2000).

■ Estratégias Evolutivas

O esforço em minimizar o atrito total de corpos tridimensionais em fluxos turbulentos foi, e ainda é, o objetivo geral das pesquisas nos institutos de hidrodinâmica. Três estudantes - Peter Bienert, Ingo Rechenberg e Hans-Paul Schwefel - encontraram-se no Instituto Hermann Föttinger da Universidade Técnica de Berlin em 1964. Como eles estavam fascinados, não somente com aerodinâmica, mas também com a cibernética, eles decidiram solucionar o problema intratável de como desenvolver formas com o auxílio de algum tipo de robô. O robô deveria ser capaz de manipular um modelo flexível posicionado dentro de um túnel de vento. Esperava-se que ao final o robô fosse capaz de modelar o objeto de forma a obter um objeto delgado. Entretanto, o robô não conseguiu alcançar esse objetivo, porque o tratamento de uma variável de cada vez e os métodos baseados em gradientes acabavam por levar a pontos de mínimo local (Rudolph, 2000). Ingo Rechenberg teve a idéia de introduzir pequenas mudanças aleatórias que só eram permitidas se essas mudanças levassem à melhorias no modelo. Essa idéia

foi o passo inicial para o nascimento das EE (Rechenberg, 1965).

Essa abordagem, da forma como foi criada, não animou a comunidade acadêmica da época devido a sua estratégia aleatória. Mais tarde, a idéia foi estendida para a aplicação em problemas multidimensionais e, ainda, foi incluída a operação de seleção. Muitas outras melhorias foram feitas, tais como a implementação de operações de cooperação e deterioração (Schwefel, 1995). Na verdade, embora a EE pareça uma abordagem aleatória, o operador de mutação – principal guia de exploração do espaço de busca nessa abordagem – é fortemente baseado em probabilidades, sendo que a viabilidade desse operador pode ser demonstrada algebricamente (Rudolph, 2000).

Relacionados a essa área pode-se citar, entre muitos outros, os trabalhos desenvolvidos por Kursawe (1992), Ostermeier (1994) e (Yao, 1997).

■ Algoritmos Genéticos

O uso dos métodos nos moldes que os levaram a serem chamados de algoritmos genéticos surgiram na década de 60. Esses métodos foram desenvolvidos por John Holland e seu grupo de estudos na Universidade de Michigan e sua publicação “*Adaptação em Sistemas Naturais e Artificiais*” (Holland, 1975), lhe rendeu o título de criador dos Algoritmos Genéticos. Entretanto, o uso de seu trabalho se deve em grande parte a um de seus alunos, David Goldberg e sua publicação intitulada “*Algoritmos Genéticos em Busca, Otimização e Aprendizado de Máquina*” (Goldberg, 1989).

Embora os méritos da criação dos AGs se devam à Holland, outros trabalhos anteriores à sua publicação já mostravam sinais da utilização de algumas de suas idéias. A primeira referência ao termo *algoritmo genético* surgiu com um trabalho desenvolvido por Bagley (1967), cujo objetivo era desenvolver um algoritmo genético capaz de calcular conjuntos de funções de avaliação de jogos. Seu trabalho foi aplicado a um jogo de damas restrito à um tabuleiro quadrado com nove casas. Entretanto, seu algoritmo mostrou-se insensível à não linearidade do jogo.

A simulação de células biológicas foi estudada por Rosenberg (1967). Nesse trabalho foi simulada uma população de organismos unicelulares com uma estrutura simplificada, porém com uma bioquímica muito rigorosa. Esse trabalho foi importante para o posterior

desenvolvimento da teoria de algoritmos genéticos aplicados a sistemas artificiais (Goldberg, 1989).

Uma aplicação aproximada de algoritmos genéticos foi realizada por Cavicchio (1970) no problema de reconhecimento de padrões. Embora esse estudo não tenha trabalhado diretamente com o problema mencionado, foi aplicado um algoritmo genético a um projeto de um conjunto de detectores para uma máquina de reconhecimento de padrões.

A primeira aplicação de algoritmos genéticos a um problema de otimização matemática foi desenvolvida por Hollstien (1971). O problema consistia na otimização de funções de duas variáveis. O objetivo era realizar o controle de um modelo de engenharia que possuísse realimentação. O algoritmo genético utilizado nesse trabalho usava operadores genéticos de *crossover*, mutação e esquemas de criação baseados na prática tradicional de criação de animais e horticultura.

A codificação dos indivíduos utilizando números reais é resultado do trabalho publicado por Bosworth (1972). Nesse estudo, os indivíduos eram compostos por seqüências de quatro a quarenta números reais, sendo que os operadores genéticos foram por eles adaptados para tal situação.

Um marco importante no desenvolvimento dos algoritmos genéticos é devido ao trabalho de De Jong (1975) intitulado “*Uma Análise do Comportamento de uma Classe de Sistemas Adaptativos Genéticos*”. Nessa publicação foi combinada a teoria dos esquemas proposta por Holland, com os resultados dos trabalhos realizados por De Jong. Esse estudo é considerado de grande importância ao desenvolvimento dos algoritmos genéticos e possui diversas conclusões cuidadosamente formuladas (Haupt, 1998).

Embora os AGs sejam utilizados com representações de seqüências de bits, outras representações foram propostas, tal como os parâmetros reais. Além disso, novas formas de *crossover* e mutação, além de outros operadores, vêm sendo introduzidos gerando variações de AGs (Eshelman, 2000). Contudo, há quem defenda que aplicações baseadas em algoritmos genéticos devam estar representadas como seqüências de bits e utilizar somente os operadores clássicos dessa abordagem (Michalewicz, 1997).

Na literatura, a Programação Genética e os Sistemas de Classificação são tratados como subáreas dos AGs, tal como representado na Figura 4, e serão descritos a seguir.

■ Programação Genética

A PG é uma forma de algoritmo evolutivo, distinguindo-se pelo uso de um conjunto particular na escolha da representação, dos operadores genéticos e da função de avaliação. A PG é implementada como um algoritmo evolutivo no qual as estruturas de dados que sofrem adaptação são na verdade programas computacionais. A função de avaliação na PG envolve a execução desses programas (Kinnear, 2000). Portanto, PG utiliza uma busca direcionada pela evolução, sendo que o espaço de busca é composto por programas que solucionam determinado problema, entre os quais existe um possível programa que produz o melhor resultado.

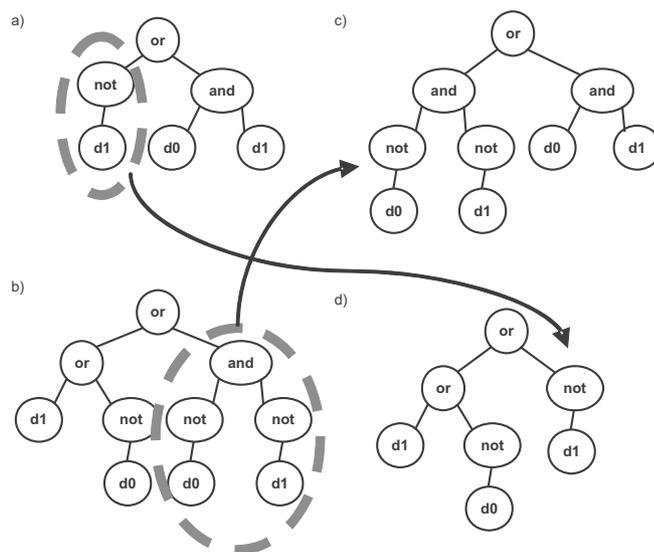
Na PG, para criar uma população inicial de soluções, um grande número de programas é gerado aleatoriamente. Cada um desses programas é executado e o resultado da execução é utilizado para compor o valor da avaliação. Então, uma nova população de programas é criada por meio da cópia de alguns programas selecionados. Essa população é complementada com a criação de novos programas que são gerados pela aplicação dos operadores genéticos em programas já existentes e previamente selecionados com base em sua aptidão. Novamente, todos os indivíduos são avaliados e novos valores de aptidão são atribuídos. Esse ciclo é repetido até que o critério de parada seja satisfeito.

Em alto nível, PG é definida como um AG com algumas escolhas diferenciadas feitas na representação do problema, nos operadores genéticos e na função de avaliação empregada.

Na Figura 5 as árvores *a*) e *b*) são exemplos de indivíduos envolvidos dentro de um ciclo típico na PG. Aplicando o operador genético obtém-se os indivíduos *c*) e *d*). Note que o operador não executa necessariamente uma operação de *crossover*, podendo as estruturas serem alteradas em qualquer ordem, caracterizando um operador de recombinação.

Na PG, diferentemente das demais abordagens, a forma como os indivíduos são representados para um problema em particular pode ser vista como uma linguagem. Essa visão na forma de representação pode ser útil na modelagem dos operadores e da função de avaliação (Kinnear, 2000).

Figura 5 - Os indivíduos *a*) e *b*) são recombinados de forma a originar os novos indivíduos *c*) e *d*)



■ Aprendizado de Sistema de Classificação

O Sistema de Classificação é normalmente referenciado como sendo a primeira técnica de Aprendizado de Máquina (AM) que utiliza AGs (Mitchell, 1999). Também, é descrita como uma técnica de AM que utiliza AGs como meio principal para a descoberta de regras de conhecimento. Entretanto, os detalhes de operação de um SC variam muito entre a vasta gama de implementações, não existindo, portanto, nenhuma versão padrão. Na verdade, um SC é mais um conceito do que um algoritmo (Smith, 2000).

Para entender o que trata um SC, deve-se ter em mente os conceitos de AM, tal como descritos por (Mitchell, 1997). Entretanto, um conceito muito importante da área de AM é utilizado na área de SC numa forma um pouco diferenciada. Em AM, conceitualmente, um classificador é uma estrutura formada por um conjunto de regras (implícitas ou explícitas), de forma que dado um novo exemplo, o classificador é capaz de prever com boa precisão à qual classe aquele novo exemplo pertence. No caso do classificador ser simbólico, a predição da classe de novos exemplos é feita utilizando todo o conjunto de regras que constituem o classificador (Monard & Baranauskas, 2003a,b). Podem existir casos raros nos quais o classificador é formado por uma única regra. Entretanto, para os SC desenvolvidos sob a ótica da CE, esse conceito pode ser um pouco divergente da área de AM, sendo que uma única regra é chamada de classificador.

Embora exista essa divergência, neste trabalho o termo classificador sempre será referido como o conjunto de regras que o constitui.

A fim de exemplificar, considere que a seqüência

$$0110 \rightarrow 010$$

é uma regra que assume um entre 16 possíveis estados e uma entre 8 possíveis ações. Assumindo ainda uma representação binária. Essa regra pode ser lida como

$$\text{If estado} = 0110 \text{ then ação} = 010$$

Considerando agora uma forma de generalização pela introdução do símbolo (*) (“não importa”), poder-se-ia ter a seqüência

$$*11* \rightarrow 010$$

significando uma regra que diz

$$\begin{aligned} \text{If estado} &= 0110 \text{ or} \\ \text{estado} &= 0111 \text{ or} \\ \text{estado} &= 1110 \text{ or} \\ \text{estado} &= 1111 \\ \text{then ação} &= 010 \end{aligned}$$

Este tipo de generalização permite que um SC utilize a mesma ação para um conjunto de estados. De forma semelhante, um algoritmo de AM simbólico supervisionado procura induzir um conjunto de regras de conhecimento, o qual constitui o classificador. Em geral, o objetivo do indutor é descobrir um conjunto de regras, ao invés de uma única regra, que seja o mais completo e consistente possível com os exemplos de treinamento. Existem duas diferentes abordagens na forma como os indivíduos são representados e como a população de regras é tratada. Essas duas abordagens são descritas a seguir.

■ Abordagem Michigan

Proveniente dos trabalhos de Holland e Reitman na década de 70 na Universidade de Michigan, o SC chamado CS-1 emprega uma forma de representação que ficou conhecida como *Michigan* em referência ao nome da universidade de origem.

Nessa abordagem, cada indivíduo da população representa uma única regra, *i.e.* uma parte da solução candidata composta por todas as regras (população) (Michalewicz, 1997). Na abordagem Michigan, após

vários ciclos a solução é dada, em geral, por uma única regra evoluída que representa o melhor indivíduo da população. Entretanto, nem sempre uma única regra é capaz de representar toda a solução de um problema. Existem formas de co-evoluir os indivíduos de forma que toda a população faça parte da solução (Smith, 2000). Ainda, outra forma é repetir sucessivas vezes o ciclo evolutivo do AG de forma a obter-se várias regras. Entretanto, esse é um processo lento que demanda um grande esforço computacional.

■ Abordagem Pittsburgh

Por outro lado, os trabalhos desenvolvidos por De Jong e Smith na Universidade de Pittsburgh, em especial o SC chamado LS-1, empregam outra abordagem na representação dos indivíduos, a qual ficou conhecida como *Pittsburgh*, em referência ao nome da universidade de origem. Na verdade essa abordagem é mais conhecida na comunidade acadêmica pelo nome *Pitt* (Smith, 2000).

Diferentemente da abordagem Michigan, nessa abordagem cada indivíduo da população representa um conjunto de regras candidatas à solução do problema. Dessa forma, a população contém vários conjuntos de regras, sendo que cada indivíduo (conjunto de regras) representa uma solução homogênea do problema. Comparativamente com a abordagem Michigan, a abordagem Pitt requer um esforço computacional menor para obter a solução, embora o cálculo da aptidão dos indivíduos seja mais complexa que na outra abordagem (Freitas, 2002).

■ Considerações Finais

Uma visão geral das sub-áreas que compõem a Computação Evolutiva foi apresentada neste trabalho. Essas sub-áreas estão calcadas no funcionamento artificial do mecanismo evolutivo natural que ocorre na natureza. O aparecimento da CE foi motivado, entre outros, pela impossibilidade de aplicação dos métodos de otimização numérica para alguns problemas do mundo real. Muitos são os termos para designar conceitos muito próximos, os quais são explanados neste trabalho com o objetivo principal de esclarecer e situar cada um deles. Ainda, embora as áreas sejam diversas, não há uma divisão clara em termos de aplicabilidade, de forma que não ocorra uma intersecção. Na verdade, muitas das

aplicações são modelos híbridos que consideram o melhor que cada conceito pode oferecer.

■ Referências Bibliográficas

- Bäck, T., Fogel, D. B., & Michalewicz, T. (2000). *Evolutionary Computation I – Basic Algorithms and Operators*. Institute of Physics Publishing - IoP.
- Bagley, J. D. (1967). *The behavior of adaptative systems which employ genetic and correlation algorithms*. Tese de Doutorado, Universidade de Michigan.
- Barricelli, N. A. (1957). Symbiogenetic evolution processes realized by artificial methods. *Methods*, 35-36(6):143–182.
- Barricelli, N. A. (1962). Numerical testing of evolution theories. *ACTA Biotheoretica*, páginas 69–126.
- Bosworth, J., Foo, N., & Zeigler, B. P. (1972). Comparison of genetic algorithms with conjugate gradient methods. *National Aeronautics and Space Administration, Washington, DC*.
- Bremermann, J. H. (1962). Optimization through evolution and recombination. *Self-Organizing Systems, M. C. Yovits et al.*
- Bremermann, J. H., Rogson, M., & Salaff, S. (1965). Search by evolution. Em M. Max-field, Callahan, A., & Fogel, L. J., editors, *Proceedings of 2nd. Cybernetic Science Symposium*, páginas 157–167, Whashington, DC.
- Cavicchio, D. J. (1970). *Adaptative search using simulated evolution*. Tese de Doutorado, Universidade de Michigan.
- Darwin, C. R. (1859). *On the Origin of Species*. JohnMurray, London.
- De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptative system*. PhD thesis, Universidade de Michigan.
- De Jong, K., Fogel, D. B., & Schwefel, H.-P. (2000). A history of evolutionary computation, chapter 6, páginas 40–58. Volume I of Bäck et al. (2000).
- Eshelman, L. J. (2000). *Genetic Algorithms*, chapter 8, páginas 64–80. Volume I do Bäck et al. (2000).
- Fisher, R. A. (1958). *The genetic theory of natural selection*. New York: Dover.
- Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). On the evolution of artificial intelligence. Em *Artificial Intelligence through Simulated Evolution*, páginas 131–156, New York, Wesley.
- Fraser, A. S. (1960). Simulation of genetic systems by automatic digital computers: 5-linkage, dominance and epistasis. *Biometrical Genetics*, páginas 70–83.
- Fraser, A. S. (1962). Simulation of genetic systems. *Journal of Theoretical Biology*, páginas 329–346.
- Freitas, A. A. (2002). *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer-Verlag.
- Friedberg, R. M. (1958). A learning machine: parte I. *IBM J.*, 2:2–13.
- Friedberg, R. M., Dunham, B., & North, J. H. (1959). A learning machine: parte II. *IBM J.*, 3:282–287.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company.
- Haupt, R. L. & Haupt, S. E. (1998). *Practical Genetic Algorithms*. Wiley-Interscience Publication.
- Holland, J. H. (1967). Nonlinear environments permitting efficient adaptation. *Computer and Information Sciences II, New York: Academic*.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Second Edition, MIT Press, 1992.
- Hollstein, R. B. (1971). *Artificial genetic adaptation in computer control systems*. Tese de Doutorado, Universidade de Michigan.
- Kinnear Jr., K. E. (2000). *Derivative methods in genetic programming*, chapter 11, páginas 103–113. Volume I do Bäck et al. (2000).
- Koza, J. R. (1992). *Genetic Programming : On the Programming of Computers by Means of Natural Selection*. MIT Press.
- Kursawe, F. (1992). Evolution strategies for vector optimization. Em *Preliminary Proceedings of the Tenth International Conference on Multiple Criteria Decision Making*, páginas 187–193, Taipei, China. <http://citeseer.nj.nec.com/kursawe92evolution.html>.
- Martin, F. G. & Cockerhan, C. C. (1960). High speed selection studies. *Biometrical Genetics*, páginas 35–45.
- Michalewicz, Z. (1997). *Genetic Algorithms + Data Structures = Evolution Programs*. IE-Springer-Verlag.
- Mitchell, M. & Taylor, C. E. (1999). Evolutionary computation: An overview. *Annual Review of Ecology and Systematics*, 20:593–616.
- Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. The MIT Press.
- Mitchell, M. (2000). Life and evolution in computers. Darwinian Evolution Across the Disciplines, M. McPeck et al. (editores). <http://citeseer.nj.nec.com/260969.html>.
- Mitchell, T. M. (1997). *Machine Learning*. WCB/McGraw-Hill.
- Monard, M. C. & Baranauskas, J. A. (2003a). *Conceitos sobre Aprendizagem de Máquina*, capítulo 4, páginas 89–114. Volume 1 de Rezende (2003), 1ª edição.
- Monard, M. C. & Baranauskas, J. A. (2003b). *Indução de Regras e Árvores de Decisão*, capítulo 5, páginas 115–140. Volume 1 de Rezende (2003), 1ª edição.
- Ostermeier, A., Gawelczyk, A., & Hansen, N. (1994). A

- derandomized approach to self adaptation of evolution strategies. *Evolutionary Computation*, 2(4):369–380. <http://citeseer.nj.nec.com/ostermeier93derandomized.html>.
- Parpinelli, R., Lopes, H., & Freitas, A. (2002). Data Mining with an Ant Colony Optimization Algorithm. *IEEE Transactions on Evolutionary Computation, special issue on Ant Colony Algorithms*, 6(4).
- Porto, V. W. (2000). *A history of evolutionary computation*, chapter 10, páginas 89–102. Volume I do Bäck et al. (2000).
- Rechenberg, I. (1965). Cybernetic solution path of an experimental problem. *Royal Aircraft Establishment Library Translation 1122, Farnborough, Hants, UK*.
- Rezende, S. O. (2003). *Sistemas Inteligentes: fundamentos e aplicações*. Editora Manole, Barueri, SP, Brasil, 1ª edição.
- Rosenberg, R. S. (1967). *Simulation of genetic population with biochemical properties*. Tese de Doutorado, Universidade de Michigan.
- Rudolph, G. (2000). *Evolution strategies*, capítulo 9, páginas 81–88. Volume I de Bäck et al. (2000).
- Schwefel, H.-P. & Rudolph, G. (1995). Contemporary evolution strategies. Em *European Conference on Artificial Life*, páginas 893–907. <http://citeseer.nj.nec.com/schwefel95contemporary.html>.
- Smith, R. E. (2000). *Learning classifier systems*, chapter 12, páginas 114–123. Volume I do Bäck et al. (2000).
- Wikipedia (2003d). Natural selection. http://www.wikipedia.org/wiki/Natural_Selection.
- Yao, X. & Liu, Y. (1997). Fast evolution strategies. Em Angeline, P. J., Reynolds, R. G., McDonnell, J. R., & Eberhart, R., editors, *Evolutionary Programming VI*, páginas 151–161, Berlin. Springer. <http://citeseer.nj.nec.com/yao97fast.html>.