

Algoritmos Evolutivos

Diego Buchinger

Universidade do Estado de Santa Catarina (UDESC)

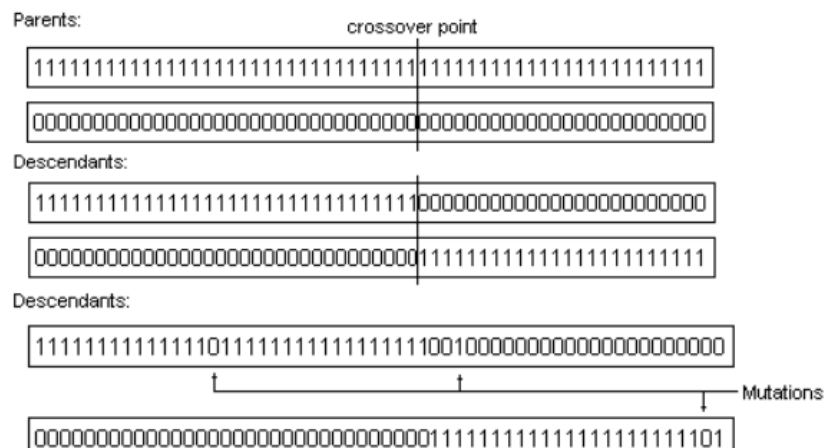
Programa de Pós Graduação em Computação Aplicada (PPGCA)

Computação Natural

Quais são os Algoritmos Evolutivos?

- Algoritmos Genéticos ou *Genetic Algorithms* (Holland, 1975; Holland, 1992): se inspiram na genética, sendo que os atributos e por consequência as soluções (indivíduos) do problema são representadas como uma palavra de bits (*bit-string*) que representa os genes. As operações de crossover e mutação são bastante simples de serem implementadas.

A maior dificuldade da abordagem está na codificação dos atributos em palavra de bits de forma a permitir que, tanto o crossover quanto a mutação, gerem soluções válidas. Esta abordagem também pode não ser muito adequada para problemas contínuos, pois é necessário discretizar os valores, levando a possíveis perdas de informação.



- Estratégias Evolutivas ou *Evolutionary Strategies* (Rechenberg, 1973; Schwefel, 1975): é um método que se baseia na expressão fenotípica, sendo que cada solução (indivíduo) é representada por um vetor de valores, chamados de parâmetros de decisão, que simbolizam as suas características, e um vetor adicional de números reais positivos que representam os parâmetros de estratégias (um parâmetro de estratégia para cada parâmetro de decisão, por exemplo).

No crossover desta abordagem, quando os dados são discretos é realizada uma escolha estocástica dos atributos dos pais. Por outro lado, se os dados forem contínuos, é possível escolher essa mesma abordagem, ou então usar um método de intermédio, no qual o filho recebe a média dos valores dos pais (gera um efeito de concentração). Já a mutação é baseada nos parâmetros de estratégia:

$$\sigma'_j = \sigma_j \times \exp(\tau_1 \times N(0,1) + \tau_2 \times N_j(0,1))$$

$N(0,1)$ representa um número randômico entre $[0,1]$ (N_j é específico para cada parâmetro)

σ_j representa um parâmetro de estratégia

τ_i representa um parâmetro de estratégia exógeno (uma função)

E1:	2,5	8,2	7,0	1,3
E2:	4,5	9,4	1,8	2,1
C:	4,5	8,2	7,0	2,1

E1:	2,5	8,2	7,0	1,3
E2:	4,5	9,4	1,8	2,1
C:	3,5	8,8	4,4	1,7

- Programação Evolutiva ou *Evolutionary Programming* (Fogel, Owens e Walsh, 1966): mantém o foco sobre toda a espécie e não nos indivíduos. Nesta abordagem, a evolução se refere ao comportamento geral, usando apenas os operadores relacionados ao fenótipo, e como os indivíduos representam uma espécie, não há reprodução entre espécies e, portanto, o operador de *crossover* não é utilizado.

A seleção utiliza uma abordagem de seleção por torneio com escolha dos melhores em cada grupo (similar a Estratégias Evolutivas). O operador de mutação, por sua vez é dependente de problema e de tipo de dado, podendo ser igual ao operador utilizados nas demais abordagens. O processo de seleção também é chamado de seleção por sobrevivência.

- *Learning Classifier Systems* - LCS (Holland, 1975; Holland e Reitmann, 1978; Geyer-Schulz, 1995): é um método de aprendizagem de máquina baseado em indução que é capaz de aprender regras condicionais básicas (se{condições} então {ação}). As condições e ações são codificadas como *bit-strings*, assim como no algoritmo genético, com a adição de um símbolo extra '#' que representa um "não importa" permitindo generalização de condições. Nesta abordagem um indivíduo representa uma regra (abordagem de Michigan) ou várias regras (abordagem de Pittsburgh).

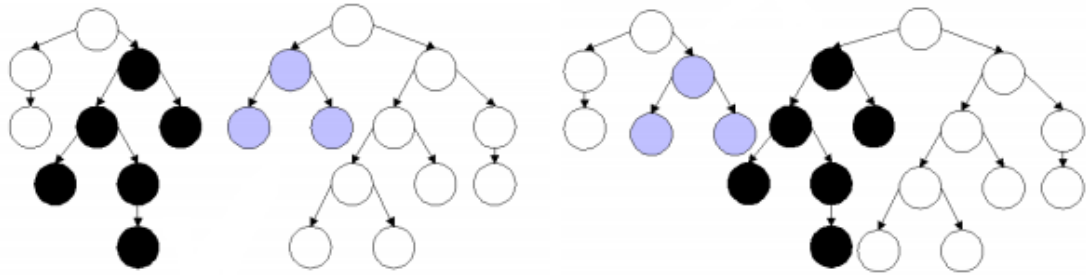
Os indivíduos são avaliados pelo seu desempenho ao interagir com o ambiente. Dentre os indivíduos com melhor desempenho um é escolhido (proporcional ao seu desempenho), recompensado (reforço) e tem sua ação executada, a qual altera o ambiente para a próxima iteração.

- Programação Genética ou *Genetic Programming* (Koza, 1992): intuito de usar processos evolucionários para gerar programas automaticamente. Programas são considerados como soluções (indivíduos) e suas instruções são consideradas atributos do problema. O *fitness* de um programa é calculado com base na avaliação da execução deste sobre diversas condições e na saída gerada. As estruturas desses programas são alteradas durante o processo de otimização.

Os primeiros experimentos envolveram a linguagem de programação LISP que é orientada a interpretação de listas. Os operadores de mutação e *crossover* garantiam que apenas novos programas válidos eram gerados. Posteriormente os programas foram codificados como árvores de instruções, perdendo a dependência com linguagens de programação.

Nesta abordagem, a seleção ocorre via torneio. Em uma seleção por torneio as soluções são alocadas aleatoriamente em grupos e os melhores indivíduos de cada grupo serão parte da população de pais. O *crossover*, por sua vez, ocorre através da seleção e troca entre duas subárvores quaisquer dos pais. Já na mutação, seleciona-se uma subárvore qualquer e troca-a por uma nova subárvore gerada arbitrariamente. A técnica canônica costuma usar populações

grandes, mas um número pequeno de gerações, uma vez que as árvores tendem a crescer rapidamente (efeito “*bloat*”)



Aspectos em comum dos Algoritmos Evolutivos

- Todos são inspirados de alguma forma em algum processo evolutivo;
- Todos são algoritmos populacionais;
- Todos são utilizados para otimização;
- Todos seguem as etapas fundamentais de: (1) inicialização de população, (2) seleção, (3) variação [mutação e/ou crossover], (4) avaliação, e (5) critério de parada.

Aspectos que diferem os Algoritmos Evolutivos

- O algoritmo genético tem a característica marcante em que os atributos devem ser todos codificados em uma *bit-string*, sendo fácil de implementar os operadores de crossover e mutação, mas ruim de trabalhar com dados contínuos.
- Os operadores de variação crossover e mutação dos Algoritmos Genéticos e das Estratégias Evolutivas possuem a mesma fundamentação, mas são implementados de forma significativamente diferente, sendo que o segundo é mais apropriado para dados contínuos e exige mais parametrização.
- A Programação Evolutiva é a única abordagem onde os indivíduos representam espécies, não havendo o operador de variação *crossover*.
- Os *Learning Classifier Systems* tem como fim um objetivo específico voltado à aprendizagem de máquina, diferentemente dos demais algoritmos.
- A programação genética tem como fim um objetivo específico de obtenção de programas para uma dada finalidade, diferentemente das outras abordagens que se focam em encontrar soluções.
- A classe de problemas que os algoritmos evolutivos se propõem a resolver também é uma diferença entre eles. Enquanto que Algoritmos Genéticos, Estratégias Evolutivas e Programação Evolutiva são voltados para otimização de funções/problemas genéricos, os *Learning Classifier Systems* são voltados exclusivamente para problemas de aprendizagem de máquina, e a Programação Genética é voltada para a geração automática de programas.

Quais classes de problemas cada um deles se propõem a resolver (na forma canônica)

Algoritmos Genéticos: otimização de funções/problemas discretos

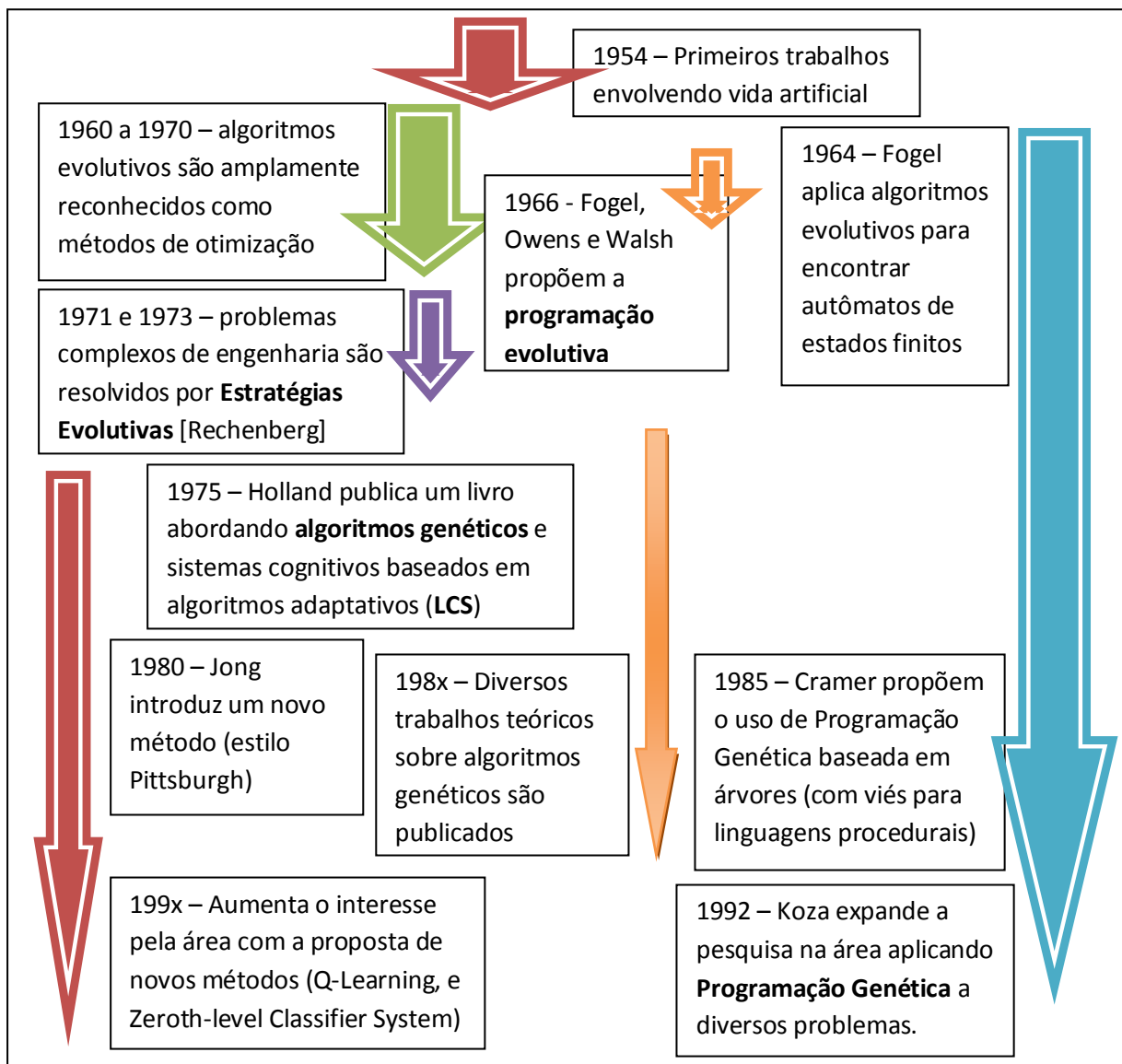
Estratégias Evolutivas & Programação Evolutiva: otimização de funções discretas ou contínuas.

Learning Classifier Systems: aprendizagem de máquina

Programação genética: geração automática de programas

Cronologia

As primeiras abordagens em computação que se inspiraram no processo evolutivo foram realizadas no final da década de 50 e início da década de 60. Foi apenas na década de 60 e 70, porém, que foram apresentadas as três primeiras propostas independentes de algoritmos evolutivos: Programação Evolutiva e Algoritmo Genético, ambos nos Estados Unidos, e Estratégias Evolutivas na Alemanha.



Referências

Fogel, L. J., Owens, A. J. e Walsh, M. J. **Artificial Intelligence through Simulated Evolution**, New York, John Wiley & Sons, 1966.

Geyer-Schulz, A. **Holland classier systems**. *Proceedings of the International Conference on Applied Programming Languages*, June 4.-8., San Antonio, p. 45-55, 1995.

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

Holland, J. **Adaption in Natural and Artificial Systems**: An Introductory Analysis with Applications to Biology, Control and Artificial Systems. The University Press of Michigan Press, Ann Arbor, 1975.

Holland, J. **Genetic Algorithms**, Scientific American, Pages: 44-50, July 1992.

Koza, J. R. **Genetic Programming**. MIT Press Cambridge, MA 1992 *1st Edition HC On the Programming of Computers by Means of Natural Selection*, VG/VG Science 03782, 1992.

Rechenberg, I. **Evolutionsstrategie**: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Problemata, 15. Volume 15 de Reihe Problemata. 170 p. 1973.

Schwefel, H. P. **Evolutionsstrategie und numerische Optimierung**, Dissertation, Technische Universität Berlin, 1975.

Streichert, F. **Introduction to Evolutionary Algorithms**. University of Tuebingen.