



Universidade Federal Rural de Pernambuco  
Pró-Reitoria de Pesquisa e Pós-Graduação  
**Programa de Pós-Graduação em  
Informática Aplicada**



# Computação Evolutiva

AULA 06 – VARIANTES DE ALGORITMOS EVOLUCIONÁRIOS – PARTE 1

# Roteiro

---

Algoritmos Genéticos

Estratégias de Evolução

Programação Evolucionária

Programação Genética

Evolução Diferencial

# Algoritmos Genéticos

---

Algoritmo Genético (GA) é a variante mais popular de algoritmos evolucionários...

- Foi proposto por Holland no livro *Adaptation in Natural and Artificial Systems*
- O algoritmo genético “padrão” é mais próximo da proposta da tese de De Jong

Representação	Cadeia de bits
Recombinação	Crossover de um ponto
Mutação	Troca de bits
Seleção dos pais	Proporcional ao fitness implementada por roleta
Seleção de sobreviventes	Por geração

Esqueleto de um GA básico.

# Algoritmos Genéticos

---

Um GA geralmente apresenta um fluxo de execução típico:

- Dada uma população de  $\mu$  indivíduos, a seleção de pais preenche um conjunto intermediário de tamanho  $\mu$ , permitindo duplicatas
- A população intermediária é embaralhada para criar pares aleatórios para serem usados na recombinação
- A recombinação é aplicada à cada par de indivíduos com probabilidade  $P_c$  e os filhos substituem os pais imediatamente
- A população intermediária passa pelo processo de mutação, indivíduo à indivíduo (cada bit de cada indivíduo é modificado com probabilidade  $p_m$ )
- A população intermediária substitui a anterior completamente

# Algoritmos Genéticos

---

Nos primeiros anos uma grande atenção foi dada à busca por valores ótimos para os parâmetros dos GAs:

- Taxas de mutação típicas ficam entre  $1/l$  e  $1/\mu$
- É comum usar probabilidades de recombinação entre 0,6 e 0,8
- O tamanho da população fica por volta de 50 (mas é comum usar também poucas centenas)

Cuidado, pois muitas das escolhas sobre parâmetros levavam em conta os recursos computacionais da época!

# Algoritmos Genéticos

---

Estudos recentes demonstram o GA básico apresenta várias desvantagens, tais como:

- Ausência do conceito de elitismo
- Seleção de pais baseado em *fitness* (*fitness x ranking*)
- Modelo de substituição muito simples
- Representação binária não adequada para vários problemas

# Estratégias de Evolução

---

Estratégias de Evolução (EE) foram inventadas no início da década de 60 por Rechenberg e Schwefel para resolver problemas de otimização de formas. Eram basicamente dois algoritmos:

- O primeiro algoritmo é conhecido como EE (1 + 1)
  - Usa um vetor para representar a solução do problema e o descendente é gerado por meio da adição de um número aleatório independente à cada posição do vetor e o filho é aceito apenas se for melhor
- O segundo algoritmo é conhecido como EE (1, 1)
  - Neste esquema o pai é sempre substituído pelo descendente gerado

# Estratégias de Evolução

---

Os descendentes em EE são sempre criados por uma mutação baseada em uma distribuição Gaussiana de média zero e desvio padrão  $\sigma$ :

- O parâmetro  $\sigma$  é conhecido como tamanho do passo de mutação
- Uma das grandes contribuições dos estudos em EE foi desenvolver um método adaptativo para controlar o passo de mutação, que ficou conhecido como **regra de 1/5 de sucesso** (Rechenberg)
- Razão de 1/5 entre sucessos de mutações.
- Tamanho do passo deve ser mudado a cada  $k$  gerações.

$$\begin{cases} \sigma/c & \text{se } p_s > 1/5 \\ \sigma \cdot c & \text{se } p_s < 1/5 \\ \sigma & \text{se } p_s = 1/5 \end{cases}$$

- Onde  $p_s$  é a chance de sucesso e  $c$  uma constante tal que  $0.817 \leq c \leq 1$ .



# Estratégias de Evolução

---

As versões modernas de EE consideram vários indivíduos e são representadas por  $(\mu + \lambda)$  e  $(\mu, \lambda)$ :

- A versão de EE que usa vários indivíduos exigiu um mecanismo mais sofisticado de adaptação do passo de mutação:
  - O cromossomo contém também o valor de passo de mutação e este valor coevolui com os indivíduos
  - Essas ideias foram pioneiras no conceito de autoadaptação de algoritmos evolucionários
- A forma básica de recombinação envolve dois pais que criam um filho
- O esquema de seleção  $(\mu, \lambda)$  é mais usado do que  $(\mu + \lambda)$ :
  - Como  $(\mu, \lambda)$  descarta todos os pais, isso evita que o algoritmo fique preso em mínimos locais (bom para problemas multimodais)
  - Se o fitness das soluções muda com o passar do tempo,  $(\mu + \lambda)$  favorece informações desatualizadas, então  $(\mu, \lambda)$  é melhor para seguir na direção do ponto ótimo neste tipo de problema
  - $(\mu + \lambda)$  dificulta a autoadaptação

# Estratégias de Evolução

---

Finalmente, um algoritmo baseado em EE pode ser resumido pela seguinte tabela:

Representação	Vetores de valores reais
Recombinação	Discreta ou intermediária
Mutação	Perturbação gaussiana
Seleção dos pais	Aleatória uniforme
Seleção de sobreviventes	Substituição elitista determinística por $(\mu + \lambda)$ ou $(\mu, \lambda)$

Esqueleto de EE.

# Programação evolucionária

---

Programação Evolucionária (PE) foi proposta inicialmente por Fogel e colaboradores para simular a evolução como um processo de aprendizagem...

- Nos dias atuais corresponde à uma técnica de otimização com representação real e possui pontos em comum com EE
- A principal diferença entre PE e EE é a inspiração biológica (com suas implicações) → em PE cada indivíduo é como se fosse uma espécie diferente, que compete com as demais
  - Não há recombinação em PE (apenas mutação)
  - Em PE cada pai gera exatamente um indivíduo e a seleção de sobreviventes é feita por torneio round-robin

# Programação evolucionária

---

Finalmente, um algoritmo baseado em PE pode ser resumido pela seguinte tabela:

Representação	Vetores de valores reais
Recombinação	Não há
Mutação	Perturbação gaussiana
Seleção dos pais	Determinística (um pai produz um filho por mutação)
Seleção de sobreviventes	Probabilística ( $\mu + \mu$ )

Esqueleto de PE.



# Programação em pares

---

Atividade para executar até 03/05:

- Definir duplas
- Cada dupla ficará com uma variante de AE apresentada (GA, EE, PE)
- Implementar, testar e subir no repositório



Universidade Federal Rural de Pernambuco  
Pró-Reitoria de Pesquisa e Pós-Graduação  
**Programa de Pós-Graduação em  
Informática Aplicada**



# Computação Evolutiva

---

AULA 06 – VARIANTES DE ALGORITMOS EVOLUCIONÁRIOS – PARTE 1