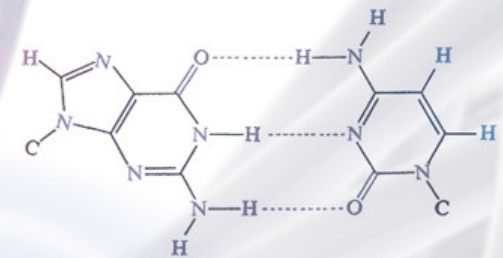


# Evolução Diferencial

Marcelo Burégio



deoxyribonucleic acid

# Roteiro

---

- Introdução
  - Inspiração biológica
  - Motivação
  - Histórico
  - Onde utilizar
- O Algoritmo
  - Visão Geral
  - Representação individual
  - Avaliação de sucesso
  - Inicialização
  - Operadores
  - Parâmetros
  - Critérios de parada
  - Pseudo-código
  - Variações
- Funcionamento
- Referências



# INTRODUÇÃO

---

Baseado na teoria da evolução

# Inspiração biológica

---

- Compartilha as demais inspirações das outras técnicas evolucionárias, tais como:
  - Ser baseada na teoria da evolução
  - Os indivíduos mais aptos tem maiores chances de sobrevivência
- Diferencial:
  - As mutações são efetuadas com o cruzamento aleatório dos próprios indivíduos da população

# Motivação

---

- Evolução Diferencial é uma técnica estocástica de otimização de funções não-lineares no **espaço contínuo** baseada em populações de indivíduos.
- Surgiu das tentativas de resolver o problema de ajuste polinomial de Chebychev.
- Considera-se que uma boa inicialização dos indivíduos provê uma boa representação do espaço de busca.
- Os novos indivíduos são gerados por um vetor de diferenças produzido pela população.
- Menor número de chamadas a função *fitness*.

# Histórico

---

- 1995 – Foi apresentado por Rainer Storn e Kenneth Price seus primeiros resultados iniciais no ICSI.
- 1996 – Em maio, Rainer e Ken participaram de um concurso de otimização evolucionária, onde terminaram em terceiro lugar, porém perdendo para métodos não tão versáteis.
- 1997 – Em abril, foi escrito um dos principais artigos sobre o algoritmo, que ganhou amplo conceito internacional



Rainer Storn

R. Storn and K. Price. **Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces.** *Journal of Global Optimization*, 11(4):431–359, 1997.

K.V. Price, R.M. Storn, and J.A. Lampinen. **Differential Evolution: A Practical Approach to Global Optimization.** *Springer*, 2005.

# Onde utilizar

---

- Problemas de otimização de funções não-lineares, de espaços contínuos, ambientes estáticos multidimensionais (mais de uma dimensão)
  - Múltiplos objetivos
  - Treinamento de redes MLP
  - Problemas reais de otimização:
    - Processamento de imagens
    - Identificação de objetos
    - ...
    - Em fim, **Busca!**



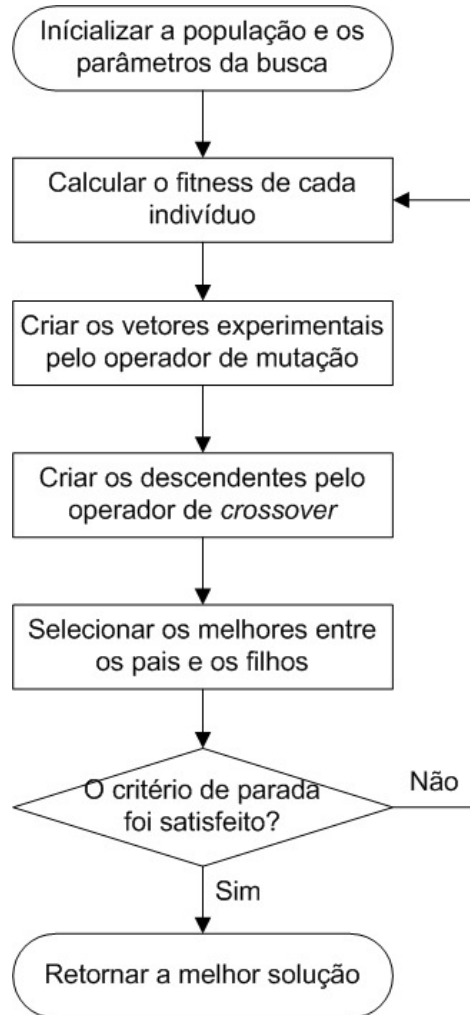
# O ALGORITMO

---

Detalhes da implementação do algoritmo



# Visão geral



- Deve ser levado em consideração uma boa inicialização dos indivíduos
- Há uma inversão dos operadores:
  - Primeiro a mutação
  - Depois o cruzamento
- A idéia principal consiste na criação de vetores experimentais, pela adição da diferença entre dois indivíduos aleatórios da população a um terceiro indivíduo aleatório.

# Representação individual

---

- Cada indivíduo representa uma possível solução no espaço de busca.
- Os indivíduos são codificados com valores reais, em um vetor.

# Avaliação de sucesso

---

- Baseada em uma função de avaliação *fitness*
  - Utiliza uma heurística, ou função matemática para descrever o ambiente.
- Simulação de uma solução proposta
  - Na ausência de uma heurística, a técnica de simulação pode ser utilizada como avaliação do *fitness*.
- Pode ser utilizado restrições na busca.

# Inicialização

---

- Distribuição uniforme aleatória dos indivíduos pelo espaço de busca
  - Uma distribuição não-uniforme pode comprometer a qualidade da solução encontrada
- Estudos empíricos demonstram que o número de indivíduos deve ser 10x o número de dimensões do problema

# Operadores

---

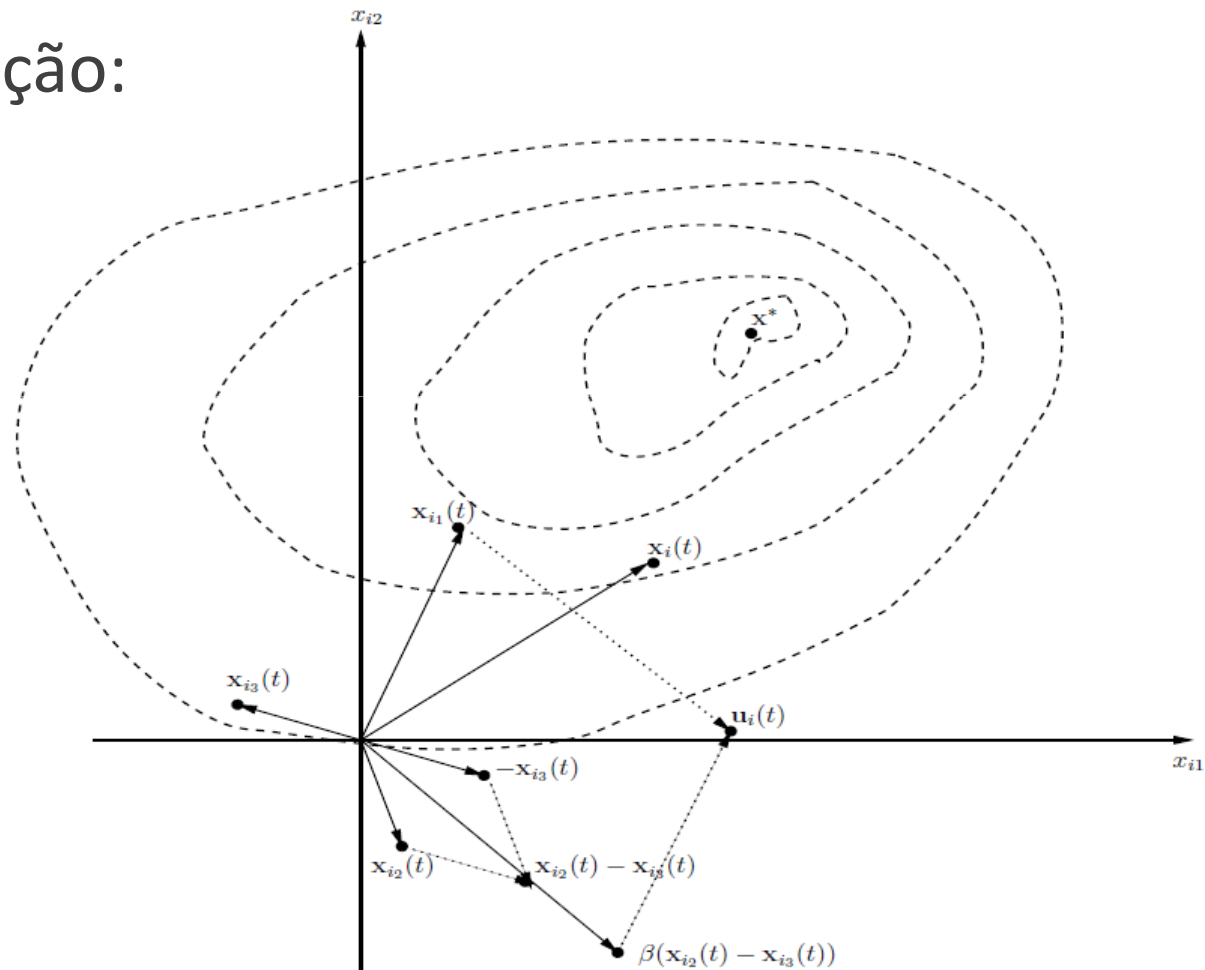
- Operador de Mutação:
  - Para cada pai da população:
    - São escolhidos aleatoriamente 2 indivíduos (diferentes);
    - É calculada a diferença vetorial entre eles;
    - Essa diferença é amplificada pelo fator de escala ( $\beta$ );
    - Esse resultado é atribuído a um terceiro vetor (destino), que resulta no vetor experimental:

$$\mathbf{u}_i(t) = \mathbf{x}_{i_1}(t) + \beta(\mathbf{x}_{i_2}(t) - \mathbf{x}_{i_3}(t))$$

- O vetor de destino pode ser:
  - Um terceiro indivíduo qualquer da população, ou
  - O melhor indivíduo da população atual.

# Operadores

- Mutaç o:



# Operadores

---

- Operador de Cruzamento:
  - Proporciona o aumento da diversidade
  - Para cada dimensão, é gerado um número aleatório que caso seja menor que a probabilidade de recombinação ( $p_r$ )
    - A componente do vetor experimental será utilizada
  - Caso contrário
    - A própria componente do pai sem alterações será utilizada

$$x'_{ij}(t) = \begin{cases} u_{ij}(t) & \text{if } j \in \mathcal{J} \\ x_{ij}(t) & \text{otherwise} \end{cases}$$

# Operadores

---

- Existem dois tipos de operadores de cruzamento:
  - Binomial
    - Os pontos de cruzamento são selecionados aleatoriamente dentro da dimensão do problema
  - Exponencial
    - Os pontos de cruzamento são selecionados em seqüência dentro da dimensão do problema, em forma de uma lista circular

```
para cada dimensão  $j$  faça  
    se  $U(0,1) < p_r$  então  
         $x_{ij} = u_{ij}$   
    fim  
fim
```

```
qtd = 0  
 $j = U(0, nx-1)$   
repete  
     $x_{ij} = u_{ij}$   
     $j = (j + 1) \bmod nx$   
     $qtd = qtd + 1$   
até  $U(0,1) \geq p_r$  ou  $qtd = nx$ 
```



- Cruzamento:



# Operadores

---

- Operador de Seleção:
  - Se o vetor experimental possuir *fitness* melhor que o vetor escolhido, então esse será substituído pelo experimental
  - Caso o vetor escolhido seja maior que o experimental, o vetor escolhido permanece na população

# Parâmetros

---

- Número de indivíduos na população
  - Controla a quantidade de indivíduos na população
  - Estudos empíricos mostram que um bom valor é 10x o número de dimensões
- Fator de escala ( $\beta$ )
  - Controla a amplificação da variação diferencial
  - Valores pequenos demais comprometem a convergência
  - Valores grandes demais facilitam a exploração porém o algoritmo pode ficar “saltando” do ponto ótimo
- Probabilidade de recombinação ( $p_r$ )
  - Influencia diretamente na diversidade, controlando a quantidade de elementos que irão mudar
  - Valores altos aumentam a diversidade e exploração
  - Valores baixos aumentam a robustez da busca

# Critérios de parada

---

- Número máximo de gerações for atingido.
- Não for observado nenhuma melhoria no melhor indivíduo em um número consecutivo de gerações.
- Não for observado nenhuma alteração na população em um número de gerações.
- Uma solução aceitável for encontrada.
- O delta da função objetivo for aproximadamente zero.

# Pseudo-código

---

- Segue abaixo o pseudo-código do algoritmo original:

```
Inicializa o contador de gerações,  $t = 0$ ;  
Inicializa os parâmetros de controle  $\beta$  e  $p_r$ ;  
Cria e inicializa uma população  $n$ x-dimensional;  
repete  
  para cada indivíduo da população faça  
    Calcular o fitness  $f(x_i(t))$   
    Cria um vetor experimental pelo operador de mutação  
    Cria um filho  $x'_i(t)$  aplicando o operador de cruzamento  
    se  $f(x'_i(t))$  é melhor que  $f(x_i(t))$  então  
      Adiciona  $x'_i(t)$  para a próxima população;  
    caso contrário  
      Adiciona  $x_i(t)$  para a próxima população;  
  fim  
fim  
até a condição de parada ser verdadeira;  
Retornar o indivíduo com o melhor fitness como sendo a solução
```

# Variações

---

- Os algoritmos de Evolução Diferencial são conhecidos na literatura como **DE/x/y/z**
  - x indica o método de seleção do vetor de destino
  - y indica o número de vetores diferenciais utilizados
  - z indica o método de cruzamento utilizado

# Variações

Número	Mutação	Notação
1	$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$	ED/rand/1/bin
2	$V^{(q+1)} = X_{best}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$	ED/best/1/bin
3	$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\lambda}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/rand/2/bin
4	$V^{(q+1)} = X_{best}^{(q)} + F(X_{\alpha}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/best/2/bin
5	$V^{(q+1)} = X_{old}^{(q)} + F(X_{best}^{(q)} - X_{old}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/rand-to-best/2/bin
6	$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$	ED/rand/1/exp
7	$V^{(q+1)} = X_{best}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$	ED/best/1/exp
8	$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\lambda}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/rand/2/exp
9	$V^{(q+1)} = X_{best}^{(q)} + F(X_{\alpha}^{(q)} - X_{\beta}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/best/2/exp
10	$V^{(q+1)} = X_{old}^{(q)} + F(X_{best}^{(q)} - X_{old}^{(q)} + X_{\gamma}^{(q)} - X_{\delta}^{(q)})$	ED/rand-to-best/2/exp



# FUNCIONAMENTO

---

Exemplo prático do uso de ED





# REFERÊNCIAS

---

Referências importantes com maiores informações sobre o ED

# Referências

---

- Storn R, Price K. ***Differential Evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces.*** International Computer Science Institute, 1995.
- Storn R, Price K. ***Differential Evolution – A simple evolution strategy for fast optimization.*** Dr. Dobb's Journal of Software Tools. 1997
- Storn R, Price K. ***Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces.*** Journal of Global Optimization, 11(4):431–359, 1997.
- K.V. Price, R.M. Storn, and J.A. Lampinen. ***Differential Evolution: A Practical Approach to Global Optimization.*** Springer, 2005.
- Storn R, Price K. ***Differential Evolution for Continuous Function Optimization.*** Disponível em: <http://www.icsi.berkeley.edu/~storn/code.html> Acesso em 20 de março. 2010.
- Engelbrecht A. ***Computational intelligence: An introduction.*** Wiley, 2007.