

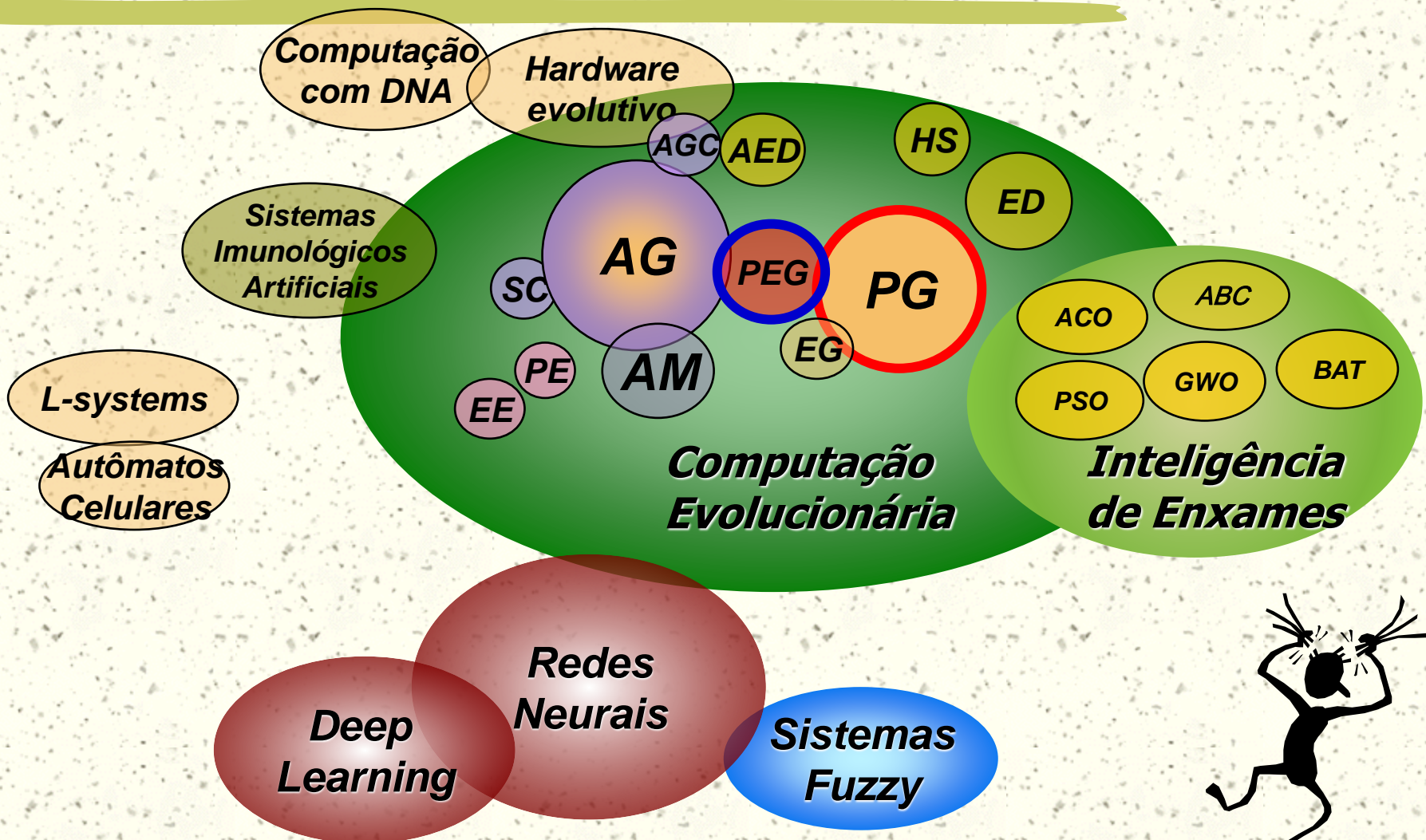
Computação Evolucionária

Prof. Heitor Silvério Lopes

hslopes@utfpr.edu.br
<http://silverio.net.br/heitor>



Inteligência Computacional



Programação de Expressão Genética (*Gene Expression Programming*)

- # Foi criada por Cândida Ferreira em 2001
- # PEG é um novo algoritmo evolucionário que herda características de ambos, AG e PG
- # É considerado como uma variante de PG
- # A diferença fundamental é na forma de representação dos indivíduos



Livros e artigos

- Ferreira, C. *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*. 2nd Edition, Springer, 2005.
- Ferreira, C. *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*. Angra do Heroísmo, Portugal, 2002 (<http://www.gene-expression-programming.com/GepBook/Introduction.htm>)
- Repositório:
- <http://www.gene-expression-programming.com>



Representação dos indivíduos em AG, PG e PEG

- # AG: cromossomo (*string* de bits) de tamanho fixo
- # PG: programas sob a forma de árvores de tamanho variável
- # PEG: cromossomo (*string* de símbolos) de tamanho fixo que geram árvores de expressão de tamanho variável

Pseudocódigo do GEP

1. Gerar população inicial
2. Expressar cromossomos
3. Avaliar programas (*fitness*)
4. Enquanto não (critério de término) faça:
 1. Conservar o melhor programa
 2. Selecionar programas com base no *fitness*
 3. Aplicar operadores genéticos e gerar novos indivíduos
5. Apresentar resultado

Open Reading Frame - ORF

- # É um conceito da genética e representa a parte de uma sequência de DNA que tem potencial para codificar uma proteína.
- # Um ORF é uma sequência contínua de códon (triplas de bases) que não contém um códon de terminação (TAA, TAG ou TGA).
- # A transcrição do DNA ocorre entre o códon de início e o de terminação

```
1. ATG CAA TGG GGA AAT GTT ACC AGG TCC GAA CTT ATT GAG GTA AGA CAG ATT TAA
2. A TGC AAT GGG GAA ATG TTA CCA GGT CCG AAC TTA TTG AGG TAA GAC AGA TTT AA
3. AT GCA ATG GGG AAA TGT TAC CAG GTC CGA ACT TAT TGA GGT AAG ACA GAT TTA A
```

Mapeamento genótipo-fenótipo

- ✦ Assim como em Algoritmos Genéticos, PEG também faz um mapeamento entre genótipo e fenótipo:
 - Genótipo → um cromossomo com um ou mais genes de tamanho fixo que utilizam a notação "Karva"
 - Fenótipo → cada gene gera uma árvore de expressão de tamanho variável

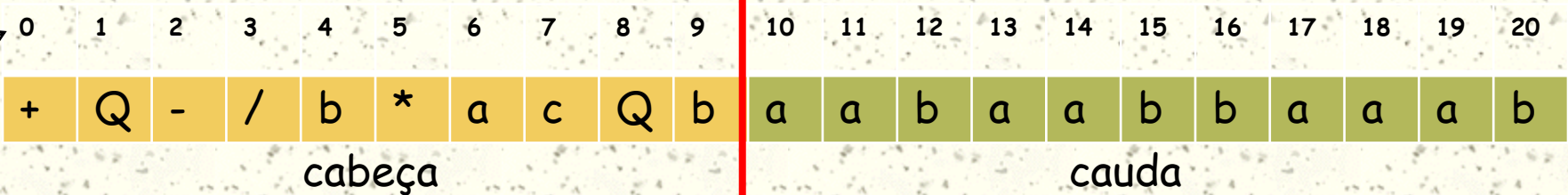
Notação Karva (*K-expression*)

- # Cada gene é dividido em duas partes:
 - Cabeça (head): sempre inicia com uma função e pode conter funções ou terminais
 - Cauda (tail): contém somente terminais
 - O comprimento da cauda é dado por: $t = h(n_{max} - 1) + 1$, onde n_{max} é a máxima aridade
- # Construção da árvore de expressão (ET):
 - Lê-se a *K-expression* e constrói-se a árvore da esquerda para a direita e um nível de cada vez (top-down) até completar todos os nós-folha.

Mapeamento genótipo-fenótipo

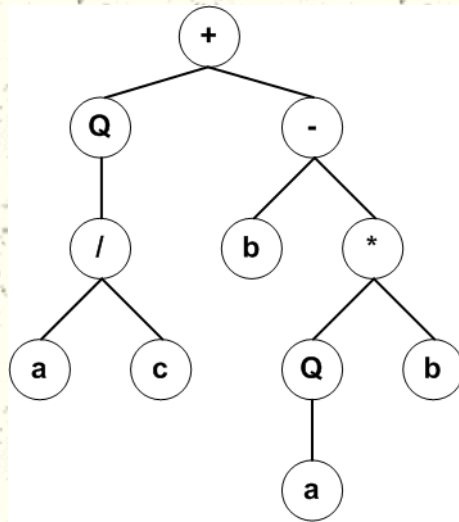
Sejam: $\mathcal{F}=\{+,-,/,* ,Q\}$ e $\mathcal{T}=\{a,b,c\}$

cromossomo



elementos do cromossomo

ORF



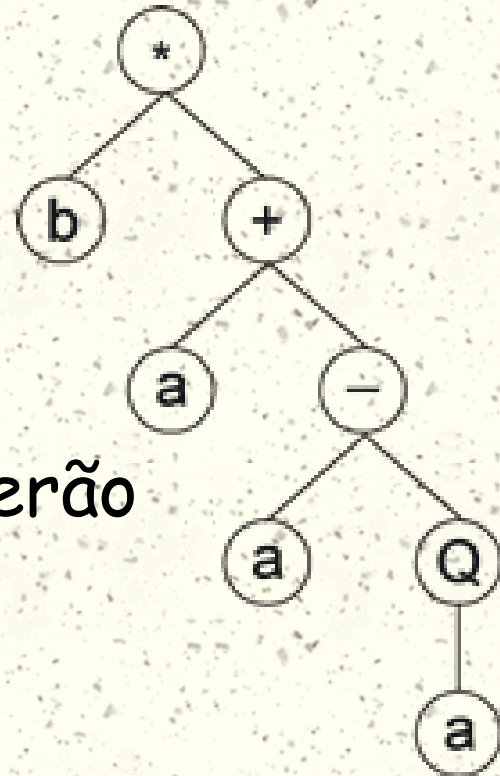
$$\left(\frac{a}{c}\right)^2 + [b - (a^2 * b)]$$

árvore de expressão

Mapeamento genótipo-fenótipo

- # Exemplo: Sejam $h = 15$ e $n_{max} = 2$, então $t = 16$.
- # Portanto o gene terá 31 elementos

0123456789012345678901234567890
 *b+a-aQab+//+b+babbabbbababbaaa



- # Nem todos os elementos do gene serão utilizados na árvore de expressão:

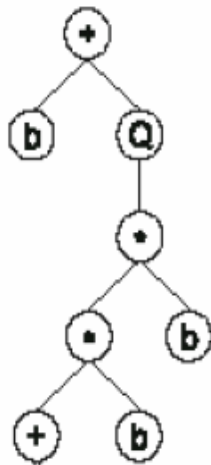
Cromossomos multigênicos

São compostos de múltiplos genes de mesmo tamanho

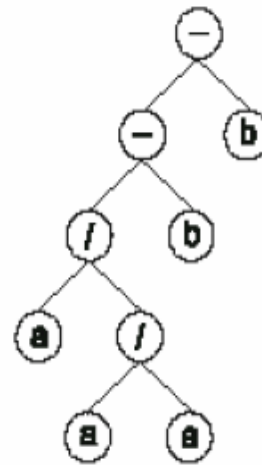
012345678901234012345678901234012345678901234
 +bQ**b+**bababbbb**--b/ba/**aaababab***Q*a*-/abaaaaab

Cada gene codifica uma sub-árvore de expressão (sub-ET) ou sub-programa

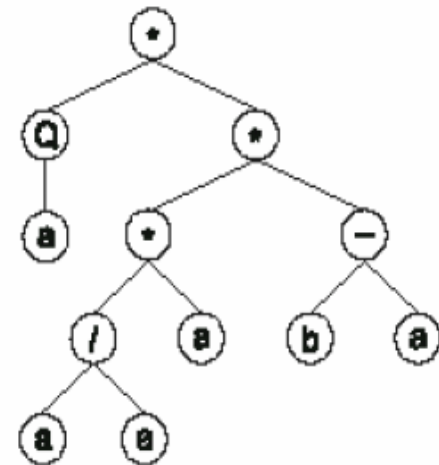
Sub-ET₁



Sub-ET₂

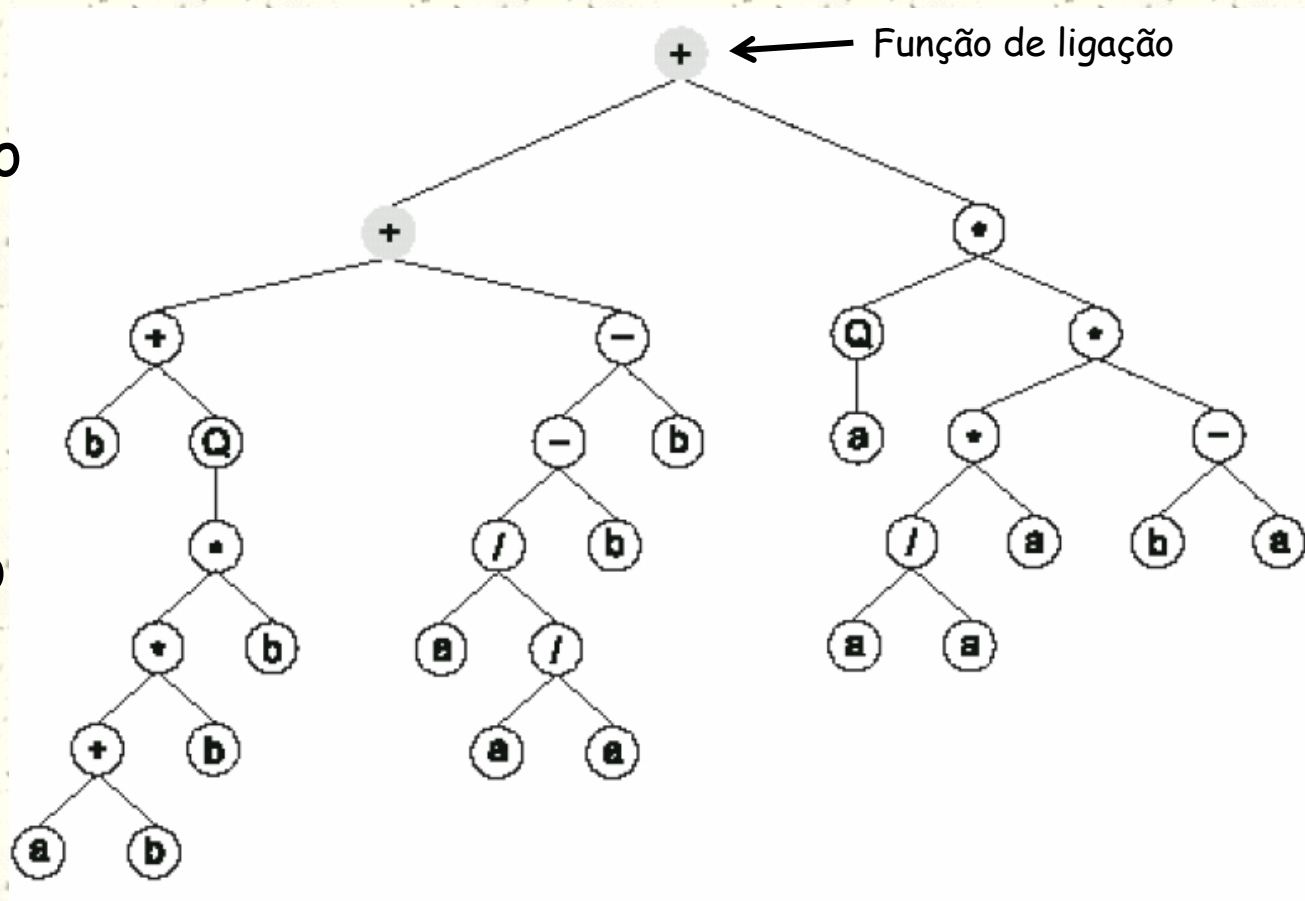


Sub-ET₃



Cromossomos multigênicos

- # Sub-Ets são interligadas por uma Função de ligação
- # Esta Função deve ser do mesmo domínio das demais funções

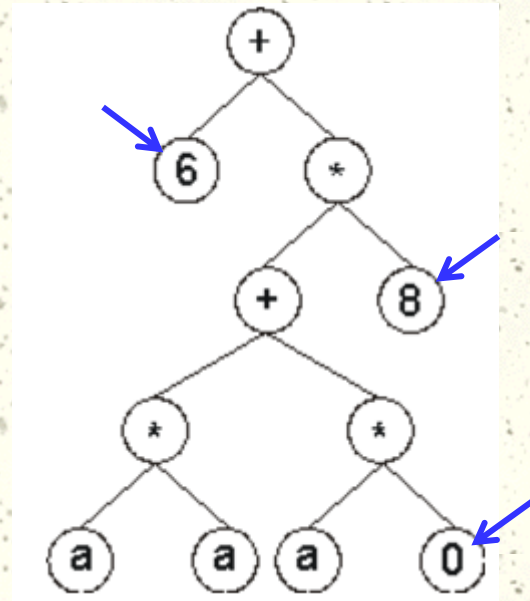
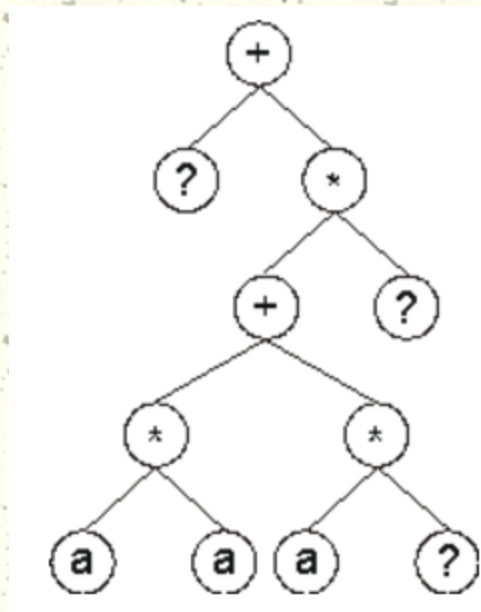


Representação de constantes numéricas

Utiliza-se o terminal "?" na K-expression e uma sequência adicional de constantes logo após a cauda.

Exemplo:

```
01234567890123456789012
+?*+?***aaa??aaa68083295
```



Passos para a aplicação de GEP

1. Definir o conjunto de funções (e aridades), conjunto de terminais
 - 1.a) Para cromossomos multigênicos, definir a função de ligação
2. Definir os casos de *fitness*
3. Definir a função de *fitness*
4. Criar aleatoriamente a população inicial
5. Definir quais operadores serão utilizados e sua probabilidade de aplicação
6. Definir o critério de parada

Função de *fitness*

- # Similar a PG, mas é dependente do problema:
 - Para problemas de regressão simbólica pode ser o somatório dos erros de ajuste, mas pode ser também: erro médio quadrático, erro absoluto, coeficiente de correlação, R-quadrado, etc
 - Para problemas de classificação pode ser a taxa de acerto, área sob a curva ROC, coeficiente de Jaccard, sensibilidade versus especificidade, etc

Método de seleção e elitismo

- # O tamanho da população é constante
- # O GEP usa normalmente o método de seleção da roleta
- # Não há operador de elitismo, mas pode-se copiar o melhor indivíduo de uma geração e utilizar o operador de replicação

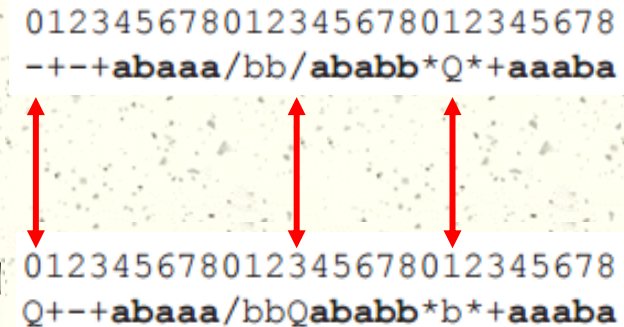
Operadores genéticos

Replicação:

- copia um indivíduo de uma geração para a seguinte (=reprodução de PG)

Mutação:

- É o mais importante do GEP.
- Qualquer função ou terminal pode sofrer mutação por outra função ou terminal
- O GEP preserva a integridade da árvore após a mutação



Operadores genéticos

Recombinação:

- Pode envolver dois ou mais cromossomos-pai
- Equivalente ao crossover, pode ser de um ou múltiplos pontos. Exemplo: 1-ponto:

012345678012345678
 -b+Qbbabb/aQbbbaab
 /-a/ababb-ba-abaaa

012345678012345678
 -b+/ababb-ba-abaaa
 /-aQbbabb/aQbbbaab

- Caso especial: Recombinação Gênica (trocam-se genes de mesma posição em dois cromossomos).

Operadores genéticos

Transposição:

- Envolve a introdução de uma sequência em algum ponto do cromossomo
- Deve preservar o tamanho e a estrutura do gene (*head/tail*)
- Há vários tipos:
 - Transposição IS (*Insertion Sequence*)
 - Transposição RIS (*Root Insertion Sequence*)
 - Transposição Gênica

Transposição IS

- # Exemplo: sequência de inserção "bba", alvo gene 1 entre elementos 6 e 7:

012345678901234567890012345678901234567890
 -+*a-+a*bbabbaabababQ**+abQbb*a**abba**aaabba

- # A sequência *upstream* é preservada e a sequência *downstream* após a inserção perde elementos da cabeça:

012345678901234567890012345678901234567890
 -+*a-**bba**+babbaabababQ**+abQbb*a**abba**aaabba

- # O resultado é sempre um programa sintaticamente correto

Transposição RIS e Transposição Gênica

- # Transposição RIS: similar à transposição IS, mas a sequência de inserção começa sempre com uma função e o ponto de inserção é sempre na raiz do gene
- # Transposição gênica: idem, um gene inteiro é transposto de posição

EGIPSYS

Codificação do Problema

EGIPSYS

Regressão Simbólica

Função de ligação: **Adição**

Escolher Funções

<input checked="" type="checkbox"/> Adição	<input type="checkbox"/> Cosseno
<input checked="" type="checkbox"/> Subtração	<input type="checkbox"/> Tangente
<input checked="" type="checkbox"/> Multiplicação	<input type="checkbox"/> Raiz Quadrada
<input checked="" type="checkbox"/> Divisão	<input type="checkbox"/> Exponencial
<input type="checkbox"/> Seno	<input type="checkbox"/> Logaritmo Neperiano
	<input type="checkbox"/> Logaritmo Base 10

Quantidade de genes por cromossomo:

Comprimento da cabeça de cada gene: a

Quantidade de indivíduos na população:

Número de terminais:

Probabilidade de usar constantes:

Tipos de Constantes

<input checked="" type="checkbox"/> Aleatórias	Range: - <input type="text" value="10"/> a + <input type="text" value="10"/>
<input type="checkbox"/> e - 2,7182818	
<input type="checkbox"/> PI - 3,1415927	
<input type="checkbox"/> 1,0	
<input type="checkbox"/> Outra:	<input type="text"/>

Caminho para o arquivo de casos de fitness:

Número de execuções:

Prefixo a ser anexado ao nome dos arquivos:

Lopes, H.S., Weinert, W.R. [EGIPSYS: a enhanced gene expression programming approach for symbolic regression problems](#). *International Journal of Applied Mathematics and Computer Science*, v. 14, n. 3, p. 375-384, 2004.

Parâmetros Evolucionários

Parâmetros

Seleção

Roleta

Torneio:

PEG Original

Operadores Genéticos

<input checked="" type="checkbox"/> Clonagem	
<input checked="" type="checkbox"/> Replicação	
<input checked="" type="checkbox"/> Mutação	<input type="text" value="0,05"/>
<input checked="" type="checkbox"/> Transposição IS	<input type="text" value="0,1"/>
<input checked="" type="checkbox"/> Transposição RIS	<input type="text" value="0,1"/>
<input checked="" type="checkbox"/> Transposição Gênica	<input type="text" value="0,1"/>
<input checked="" type="checkbox"/> Recombinação	<input type="text" value="0,3"/>
<input checked="" type="checkbox"/> Recombinação Gênica	<input type="text" value="0,1"/>

Operador de Ajuste Fino de Constantes

Probabilidade de Aplicação:

Atuar a cada "N" gerações:

Atuar "%" últimas gerações:

Função de Fitness

Fitness Original fitness = M - |Erro|

M =

Erro Absoluto

Erro Quadrático

Normalização

Multiplicador:

(1/(1+Multiplicador * Erro))

(10/(10+Multiplicador * Erro))

(100/(100+Multiplicador * Erro))

Fitness Escalonado Automaticamente

Número de Gerações:

Precisão Média:

EGIPSYS

Baseado no Lil-GP, mas com diversas melhorias:

- Múltiplos genes por cromossomo
- Diversas constantes predefinidas
- Mais operadores genéticos
- Expansão/compressão de escala
- Ajuste fino (busca local)

Weinert, W.R., Lopes H. S., [Análise de estratégias para melhoria do desempenho do algoritmo de programação por expressão genética na tarefa de regressão simbólica](#). In: *Anais do I Simpósio Brasileiro de Inteligência Computacional, 2007*, Florianópolis, [CD-ROM], 2007

EGIPSYS versus Lil-GP

Função quadrática com ruído

Função Objetivo

$$y = 2x^2 - 3x + 4 + \left(\frac{a}{5} - 0,1\right)$$

Onde a é um número aleatório entre 0 e 1.

Função EGIPSYS (31 nós)

$$y = 2x^2 - 3x + 3,981$$

Função Lil- GP (117 nós)

$$y = \frac{-0,0046x^4 - 0,1166x^3 - 2,0726x^2 + 2,5458x + 3,4239}{x + 0,8276}$$

Árvore -EGIPSYS	Árvore - Lil-GP
	<pre>(- (+ (((+ (+ X X) (+ -0,33297 -0,38708)) (/ X 0,93891)) ((+ (+ -0,88409 X) ((+ (- (+ ((+ (+ X X) (- -0,81098 X)) (+ (/ X (- 0,31235 -0,81110)) (- -0,73125 X))) ((+ (/ (+ -0,88409 X) ((+ (- X X) (+ -0,33297 -0,38708)) (/ (- 0,82690 0,31569) (/ -0,96536 -0,05093)))))) ((+ (/ ((+ X 0,30549) (+ 0,82763 X)) (+ -0,88409 X)) (/ (- 0,82690 0,31569) (/ -0,96536 -0,05093)))))) (/ (- 0,82690 0,31569) (/ -0,96536 -0,05093)))))) (+ (- (+ (- 0,79829 0,13051) (+ X X)) -0,38708) (- (/ (X X) ((- -0,78965 0,39240)) (+ X X)))))) (+ X X) (+ (/ -0,19921 0,08084) (- 0,47422 (+ (- (+ (+ X X) (+ -0,33297 -0,38708)) ((- -0,73125 X) (/ X -0,95027)))))) ((- -0,73125 X) (/ ((- -0,73125 X) (/ X -0,95027)))))) 0,95027)))))) (/ (- 0,82690 0,31569) (/ -0,96536 -0,05093)))))) (+ ((+ X (/ X X)) (- (/ (X X) ((- -0,78965 0,39240)) ((+ (/ (+ -0,88409 X) (/ -0,46083 0,47422)) (/ X X))))))</pre>
<p>Onde: a = 6,797 b = 7,529 c = 8,041 d = 2,713 x = terminal de entrada</p>	
<p>Obs: divisão por 0 é protegida e sempre retorna o valor 1.</p>	

GEPCLASS

- # GEP específico para a tarefa de classificação em mineração de dados
- # Gera regras de decisão
- # Permite a seleção de diversos parâmetros, função de ligação e operadores genéticos
- # Utiliza sensibilidade X especificidade como função de *fitness*

GEPCLASS

Codificação do Problema

GEPCLASS

Classificação

Função de ligação ▾

Conjunto de funções:

AND
 NOT (~)
 <
 <>
 OR
 >
 =

Classe para a qual se deseja gerar regras.....

Número de partições para validação cruzada.....

Quantidade de genes por cromossomo.....

Comprimento da cabeça de cada gene. a

Quantidade de indivíduos na população.....

Número de terminais.....

Caminho para o arquivo de casos de fitness

C:\GEP_SOFTWARE_CL\Classificação\FitnessCase.txt

Weinert, W.R., Lopes, H.S., [GEPCLASS: a classification rule discovery tool using gene expression programming](#). *Lecture Notes in Computer Science*, v. 4093, p. 871-880, 2006.

Parâmetros Evolucionários

Parâmetros

Seleção

Roleta
 Torneio.....
 PEG Original (Aleatório)

Operadores Genéticos

Clonagem
 Replicacao
 Mutação.....
 Transposição IS.....
 Transposição RIS.....
 Transposição Gênica.....
 Recombinação.....
 Recombinação Gênica.....

Função de Fitness

fitness=Especificidade x Sensibilidade

Fitness Escalonado Automaticamente

Número de Gerações...

GEPCLASS versus Lil-GP

Ljubliana breast cancer dataset

Bojarczuk, 2001.

Classe	Regra
1	SE ((<i>irradiat</i> = no) E (<i>deg-malig</i> = 1)) OU ((<i>deg-malig</i> = 2) E (<i>inv-nodes</i> < 3-5)) ENTÃO (classe = 1 (não reincidência de câncer))
2	SE ((<i>tumor-size</i> ≥ 10-14) E (<i>inv-nodes</i> ≥ 3-5)) OU ((<i>tumor-size</i> ≥ 25-29) E (<i>deg-malig</i> = 3)) ENTÃO (classe = 2 (reincidência de câncer))

GEPCLASS

Classe	Regra
1	SE (((<i>brest-quad</i> = 0) OU (<i>irradiat</i> ≠ yes)) E (<i>deg-malig</i> ≠ 2)) ENTÃO (classe= 1(não reincidência de câncer))
2	SE (((<i>menopause</i> ≠ lt40) OU (<i>menopause</i> = <i>premeno</i>)) E (((<i>irradiat</i> ≠ yes) E (<i>Age</i> = 30-39)) OU (<i>deg-malig</i> = 2))) ENTÃO (classe=2 (reincidência de câncer))

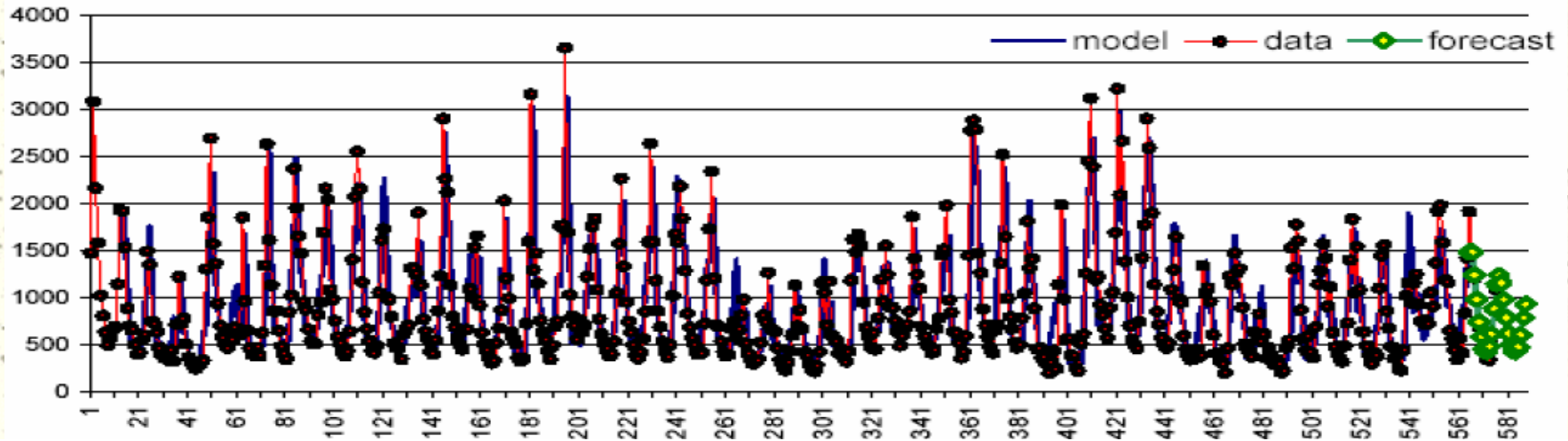
Aplicação: previsão de séries temporais

- # Aplicado às seguintes séries temporais:
 - Previsão de vazão em Furnas
 - Previsão de vazão do rio Nilo
 - Previsão de vazão do rio Tietê
 - Previsão de manchas solares
 - Previsão do valor do salário diário na Inglaterra

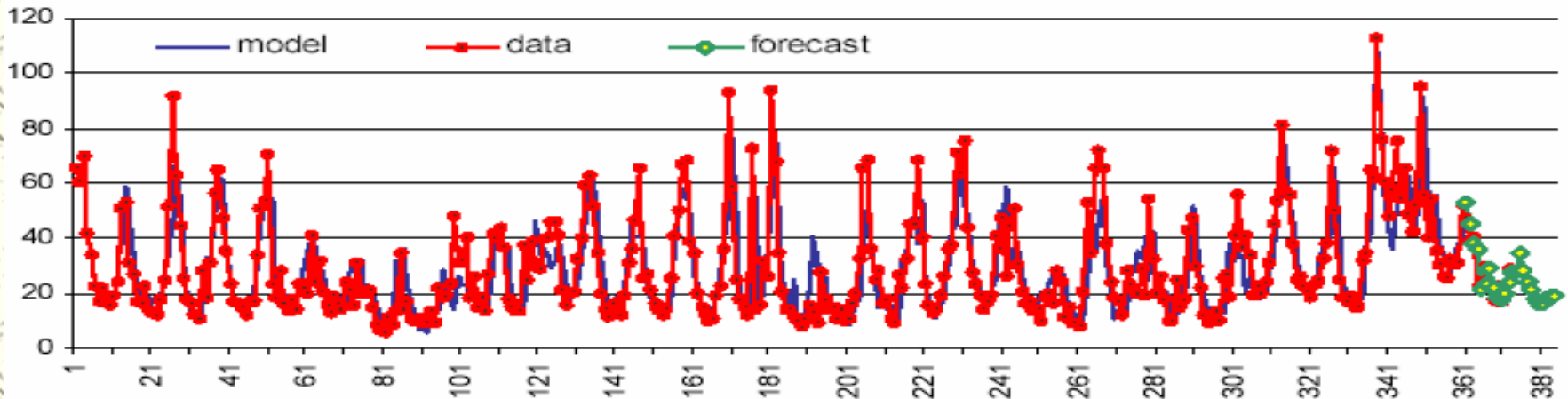
Lopes, H.S., Weinert, W.R. [A gene-expression programming system for time-series modeling.](#)
In: *Proceedings of XXV Iberian Latin American Congress on Computational Methods in Engineering (CILAMCE)*, Recife (PE), 10-12/november, [CD-ROM], 2004.

Aplicação: previsão de séries temporais

Furnas Series

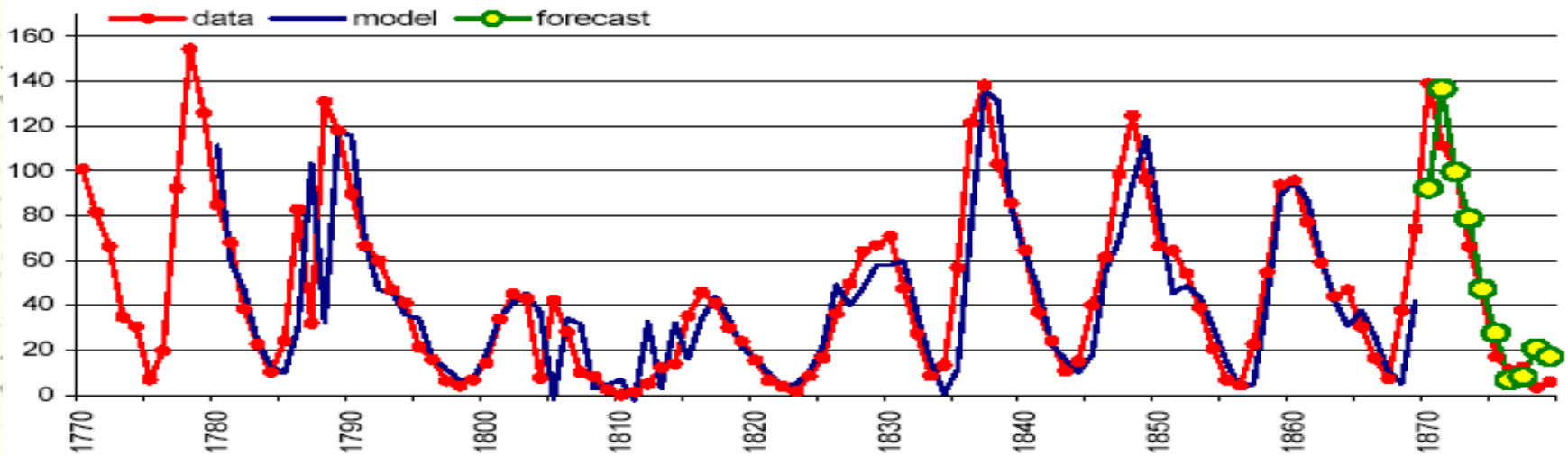


Tietê River Series

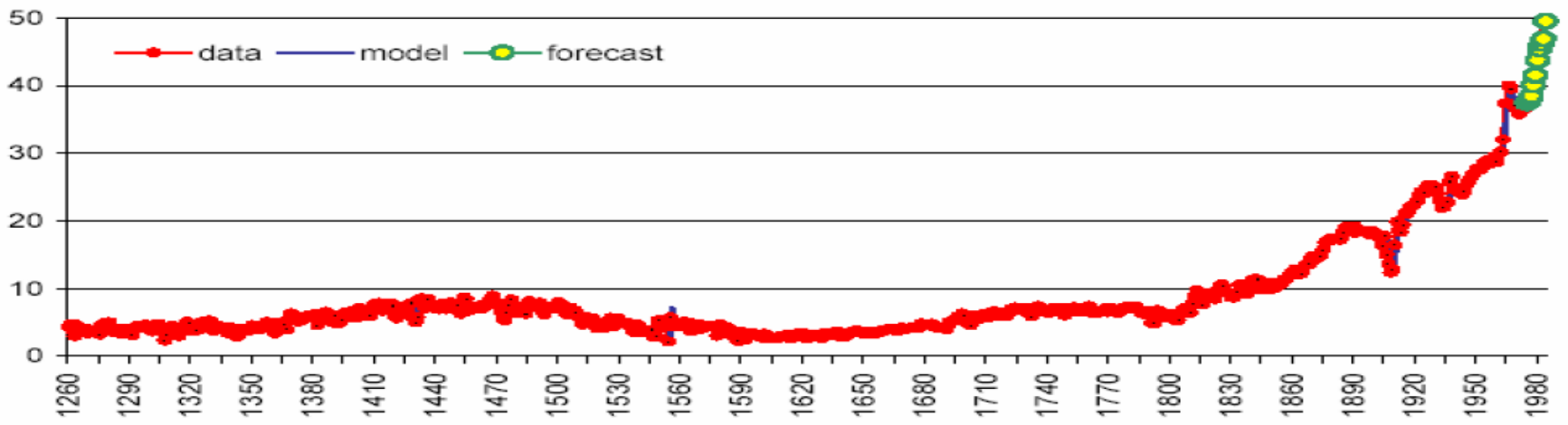


Aplicação: previsão de séries temporais

Sunspots Series



Daily Wages Series



Bibliotecas para GEP

- # GEP4J - GEP for **Java** project
(<http://code.google.com/p/gep4j/>)
- # jGEP - **Java** GEP Toolkit
(<http://sourceforge.net/projects/jgep/>)
- # Software comercial: GeneXproTools 5.0

GEP em Python

<https://github.com/ShuhuaGao/geppy>

geppy: a gene expression programming framework in Python



<https://code.google.com/archive/p/pygep/>



pygep