

Programação Genética

Danilo Araújo



Roteiro

- Inspiração e Motivação
- Histórico
- Visão Geral
- Representação de Indivíduos
- Inicialização
- Avaliação de sucesso
- Operadores
- Engorda
- Aplicações

Inspiração e Motivação

- Teoria da evolução por seleção natural de Darwin (adaptação ao ambiente)
“Mais aptos têm mais chances de sobreviver e reproduzir”
- A criação de novas estruturas se dá pela seleção natural, recombinação sexual (cruzamento genético) e mutações



Inspiração e Motivação

- Questão chave para motivação em PG:

"Como os computadores podem aprender a resolver problemas sem estarem explicitamente programados? Em outras palavras, como os computadores podem ser criados para fazer o que precisava ser feito, sem ser dito exatamente como fazê-lo?"

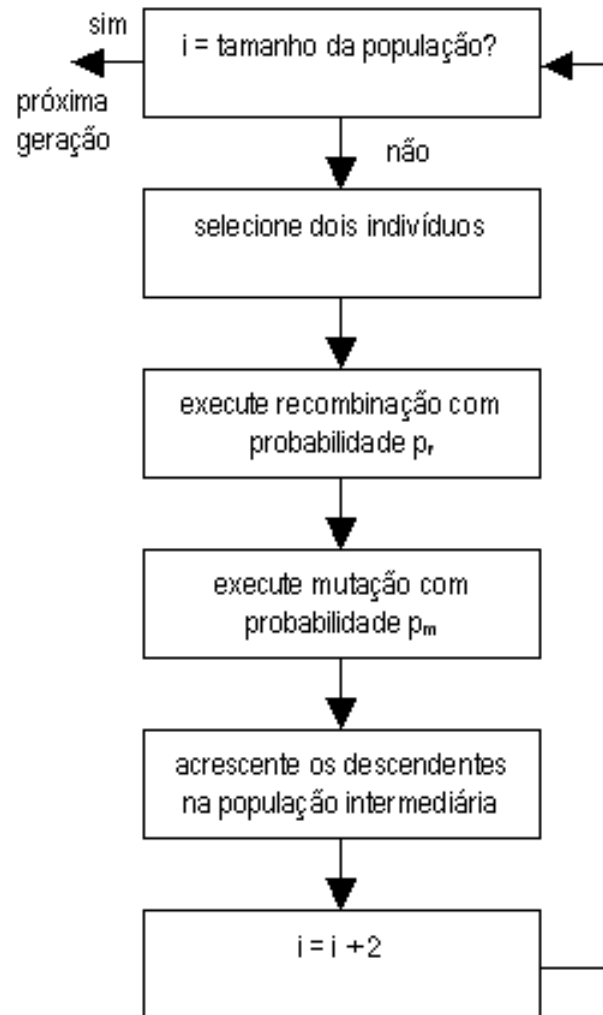
Histórico

- Pesquisas em Computação Evolucionária nas décadas de 1950, 1960, 1970. Destaque trabalhos John Holland
- N. Cramer, em 1985 – primeiros trabalhos sobre PG
- John R. Koza, em 1992 – popularização da técnica
- Que paralelo podemos fazer com invenção da imprensa?
- Já foram registrados 36 problemas em que PG produziu um resultado competitivo
- 21 destes trabalhos correspondem a soluções que infringiam ou duplicavam patentes já registradas
- Em 2 dos 36 casos a PG criou uma invenção completamente nova e patenteável

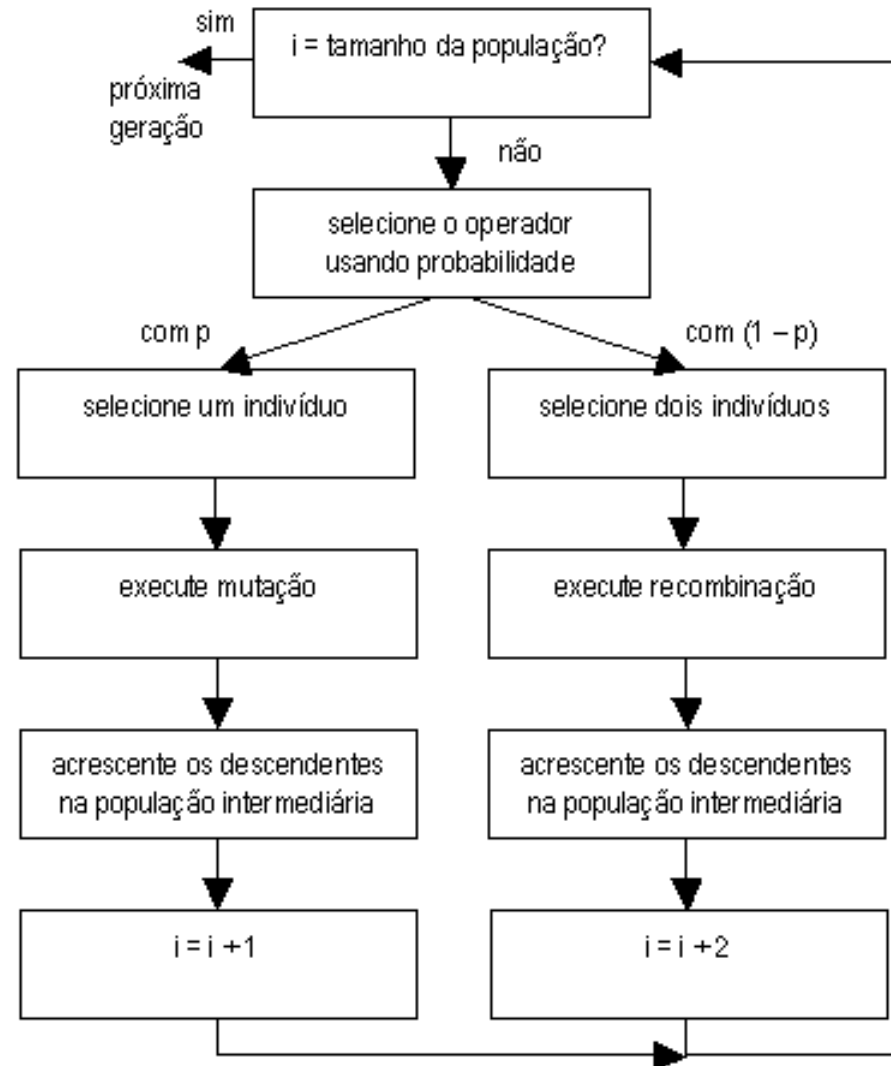
Visão Geral

- PG é "programação de computadores através da seleção natural" ou "a evolução natural de programas de computadores"
- PG x AG
- Implicações da ordem de grandeza do espaço de busca → tamanho médio da população
- Uso de Mutação e Recombinação é diferente

Visão Geral



fluxo de execução de GA



fluxo de execução de PG

Representação de Indivíduos

- O que um indivíduo pode representar em PG?

- Fórmula aritmética

$$2x^2 - 4x + 10$$

- Fórmula lógica

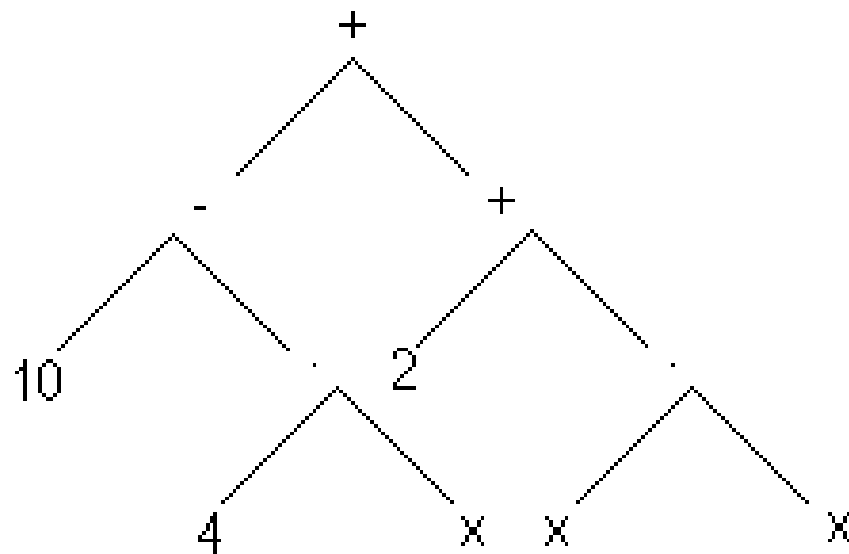
$$(p \vee \text{true}) \rightarrow (p \wedge q) \vee (\neg q)$$

- Programa em linguagem de alto nível

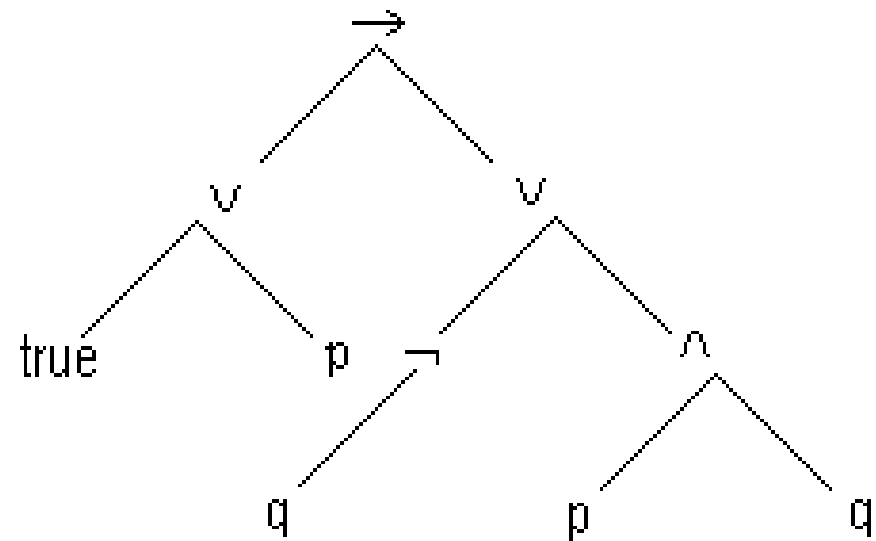
```
int numero = 2;  
if (numero < 3)  
    numero *= numero;
```


Representação de Indivíduos

- Representação em árvore



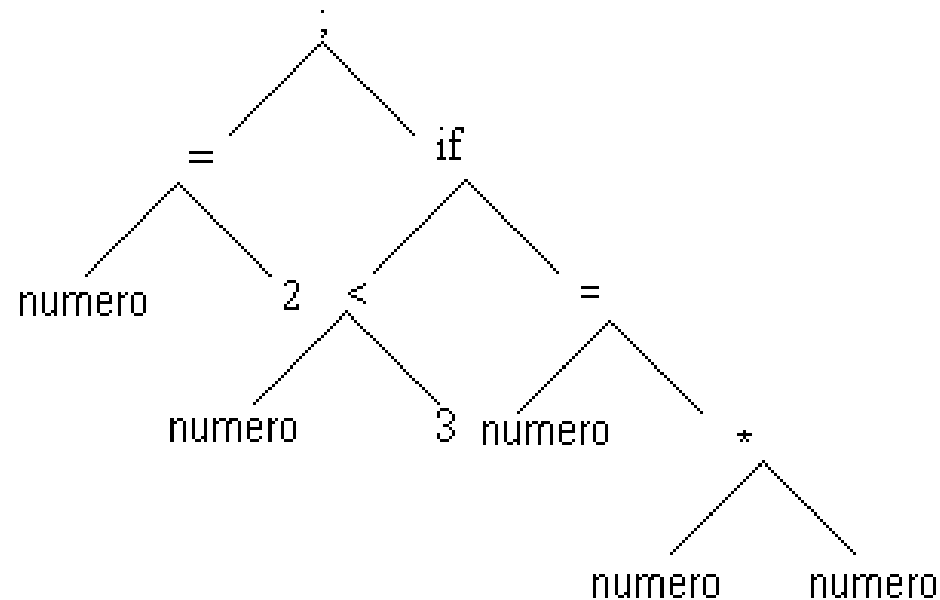
$$2x^2 - 4x + 10$$



$$(p \vee \text{true}) \rightarrow (p \wedge q) \vee (\neg q)$$

Representação de Indivíduos

- Representação em árvore



```
int numero = 2;  
if (numero < 3)  
    numero *= numero;
```

Inicialização

- Método *ramped half-and-half*
- Definir profundidade máxima D_{\max} e selecionar terminais (T) ou funções (F) usando os métodos abaixo com igual probabilidade:
 - Método completo: cada ramo tem profundidade D_{\max} . É escolhido F se $d < D_{\max}$ ou T se $d = D_{\max}$
 - Método de crescimento: os ramos podem ter profundidades diferentes, limitadas a D_{\max}

Avaliação de Sucesso

- Distância euclidiana entre os pontos fornecidos (problema) e os pontos dados pelo indivíduo

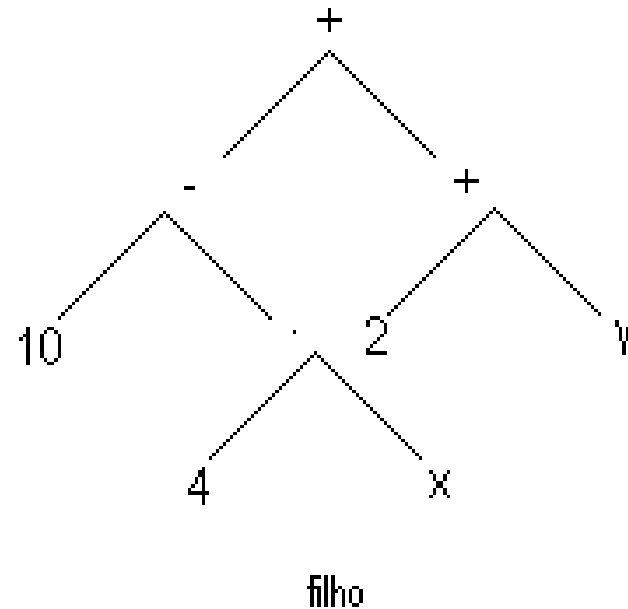
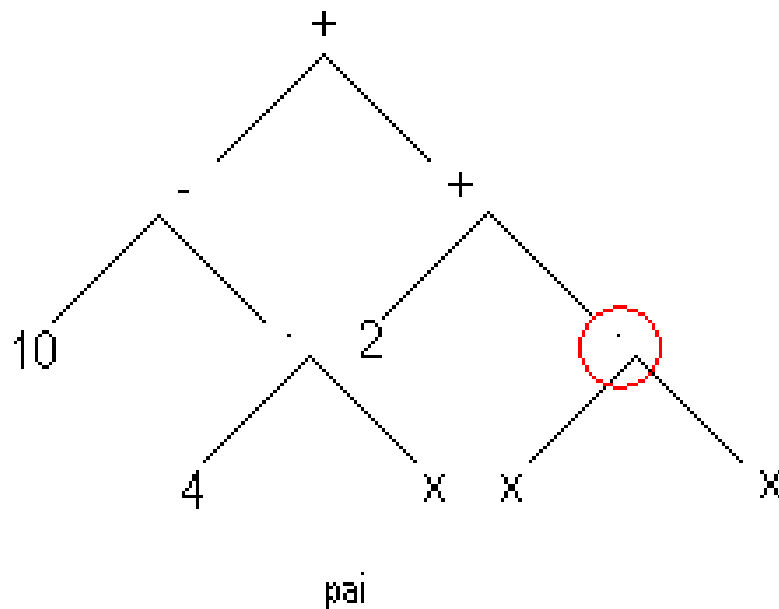
$$d(y, \bar{y}) = \sqrt{|y_1 - \bar{y}_1|^2 + |y_2 - \bar{y}_2|^2 + \dots + |y_n - \bar{y}_n|^2}.$$

- Observar *outliers*!
- Lembrar que a avaliação consiste em executar o programa correspondente ao indivíduo → custo!

OPERADORES

Operadores: Mutação

- Mecanismo de para gerar diversidade
- Substituição de uma sub-árvore do indivíduo que sofrerá a mutação:

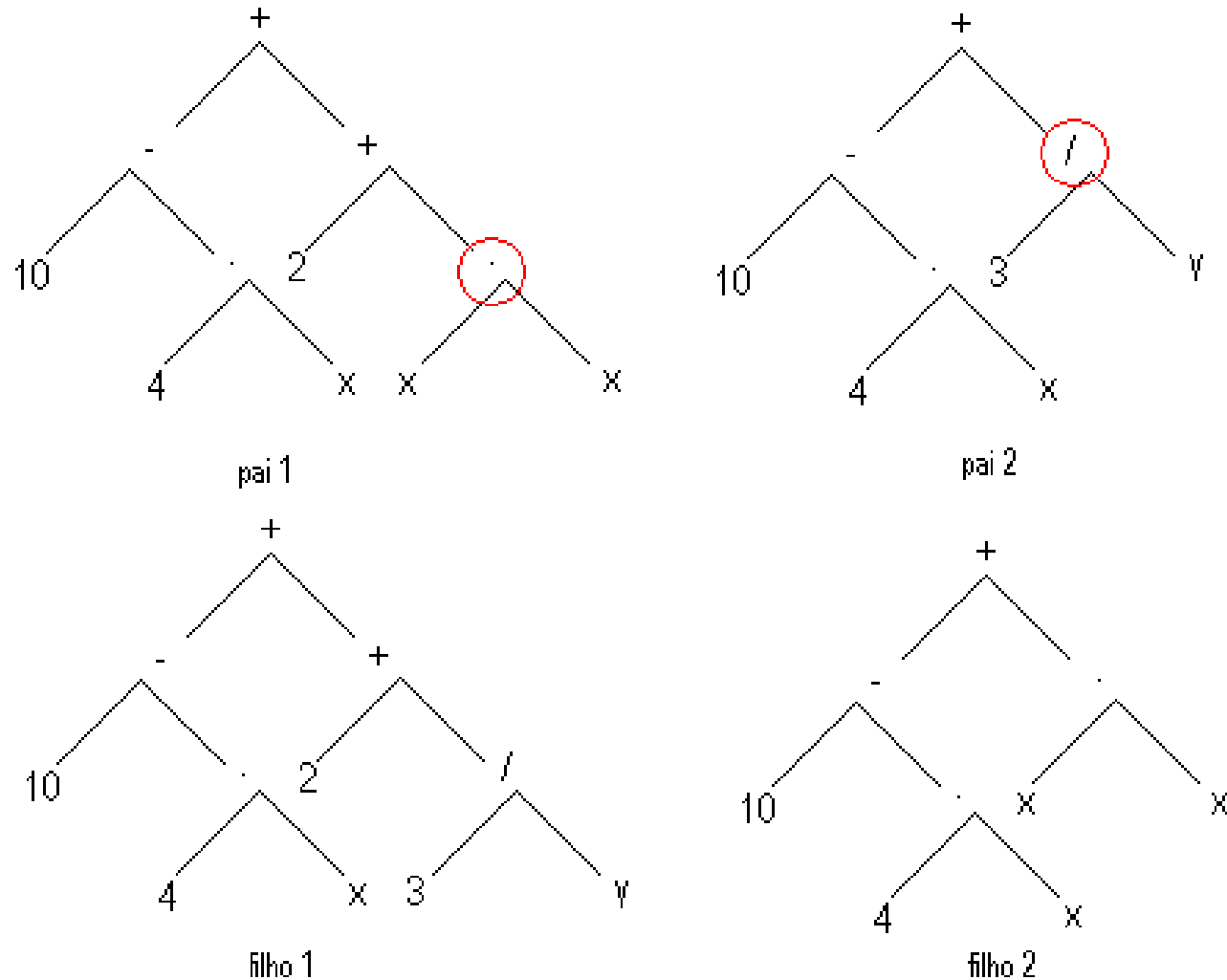


Operadores: Mutação

- Parâmetros:
- A probabilidade de realizar uma mutação no indivíduo considerado.
- A probabilidade de selecionar um dos nós da árvore representativa do indivíduo para ser substituído.
- A nova árvore gerada pode ser de tamanho diferente!

Operadores: Recombinação

- Método: recombinação de sub-árvores



Operadores: Recombinação

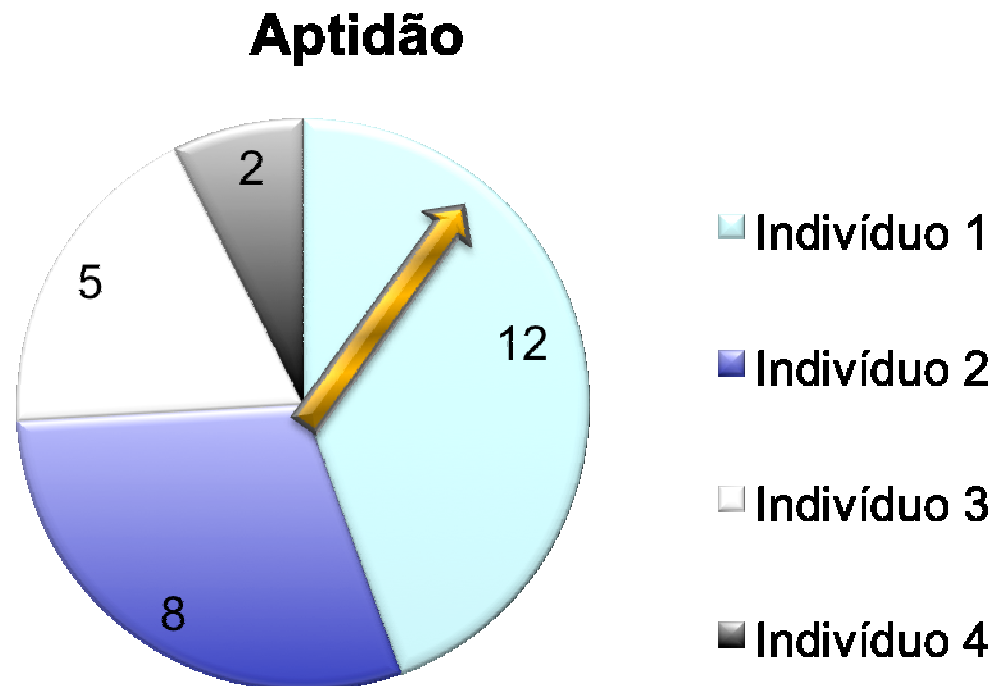
- Parâmetros:
- A probabilidade de realizar a recombinação entre dois indivíduos.
- A probabilidade de selecionar um dos nós em cada árvore-pai para servir como ponto de corte.
- A nova árvore gerada pode ser de tamanho diferente!

Operadores: Seleção

- Seleção de pais
- Seleção de sobrevivência

Operadores: Seleção de pais

- Roleta (escolha proporcional à aptidão)



Operadores: Seleção de pais

- População acima de 1000 → método *over-selection*

Tamanho da população	Proporção da população do grupo 1 (x)
1000	32%
2000	16%
4000	8%
8000	4%

Operadores: Seleção de sobrevivência

- Substituição de pais por filhos → tempo de vida de uma geração
- O original não usa elitismo
- A tendência é passar a incluir mecanismos de elitismo

Critérios de parada

- Número máximo de gerações alcançado
- Determinar se convergiu:
“Nas últimas k gerações houve melhora na aptidão?”

Engorda

- Crescimento da árvore
- Seria *overfitting*?
- Estratégias para evitar:
 - Usar um tamanho máximo, se exceder não usa o operador
 - Usar penalidade na função de avaliação ou otimização multi-objetivo

APLICAÇÕES

Onde Usar?

- Geração de pequenas sub-rotinas;
- Problemas em tempo real (sistemas de controle para robótica);
- Aplicações para análise de series temporais e mineração de dados;
- Aplicações de aproximação de funções através de Redes Neurais Artificiais (RNA) são candidatas obvias;
- Sistemas baseados em regras, como provas automáticas de teoremas.

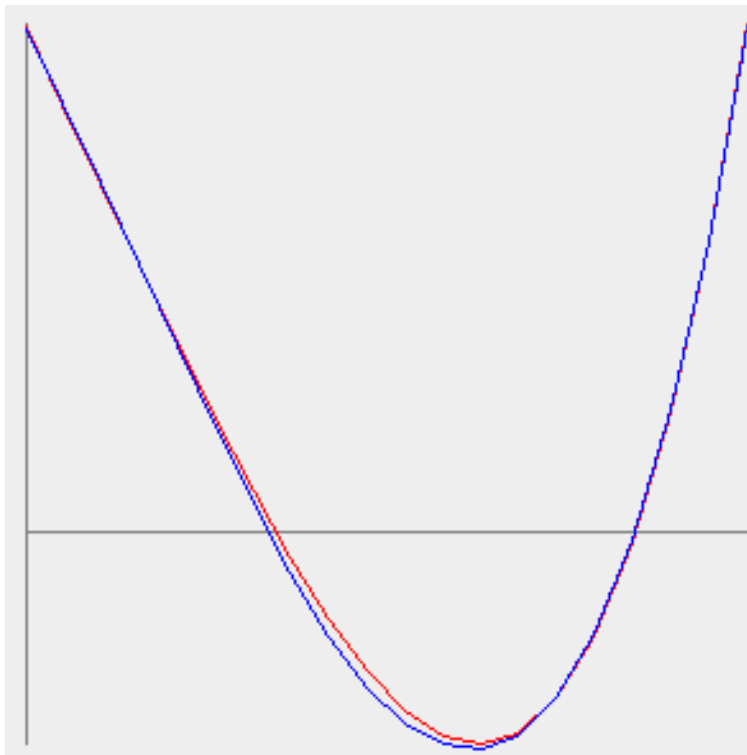
Estudo de caso: Regressão Simbólica

- Dado um conjunto de pontos de entrada-saída, determinar a o texto da função que mais se aproxima desses pontos
- Conjunto de funções $F = \{+, -, /, *, \text{sen}, \text{cos}, \text{exp}, \text{abs}, \text{max}, \text{min}, \text{log}\}$
- Conjunto de terminais $T = \mathbb{R} \cup \{x\}$
- Exemplo: 20 pontos gerados a partir de

$$3x^4 - 3x + 1$$

Estudo de caso: Regressão Simbólica

- Linha vermelha: curva real
- Linha azul: melhor curva encontrada após 747 iterações!



```
sub(  
  sub(  
    sub(  
      mul(x,-1.1625728052968025),  
      add(1.4473329219405509,x)  
    ),  
    mul(  
      sub(-0.1759565063724864,-0.5406932812173544),  
      add(  
        add(  
          x,  
          add(3.3133816502639917,-3.740923112937817)  
        ),  
        x  
      ),  
      x  
    )  
  )  
)  
, ...
```

Perspectivas Futuras e Considerações

- Tendência de uso de paralelismo
- Aplicação em problemas cada vez mais complexos com o passar do tempo (+ recursos)
- Ferramenta para novas invenções ou melhoria de soluções existentes

Referências Bibliográficas

- [1] J.E. Smith Agoston E. Eiben. Introduction to Evolutionary Computation. Springer, 2003.
- [2] N. L. Cramer. A representation for the adaptive generation of simple sequential programs. pages 183{187, 1985.
- [3] A.M.S. Zalzala; D. Green. Mtgp: a multithreaded java tool for genetic programming applications. Congress on Evolutionary Computation, 1999. CEC 99., 1999.
- [4] John R. Koza. Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press, 1992.

Referências Bibliográficas

- [5] John R. Koza. www.genetic-programming.org, 2007.
- [6] Ricardo Linden. Algoritmos Geneticos, volume 2. 2008.
- [7] S. F. Smith. A learning system based on genetic adaptive algorithms. Page 214, 1980.
- [8] C. Ryan L. Spector C. Jacob. W. Banzhaf., J. R. Koza. Genetic programming. Intelligent Systems and their Applications, IEEE, 15:74{84, 2000.