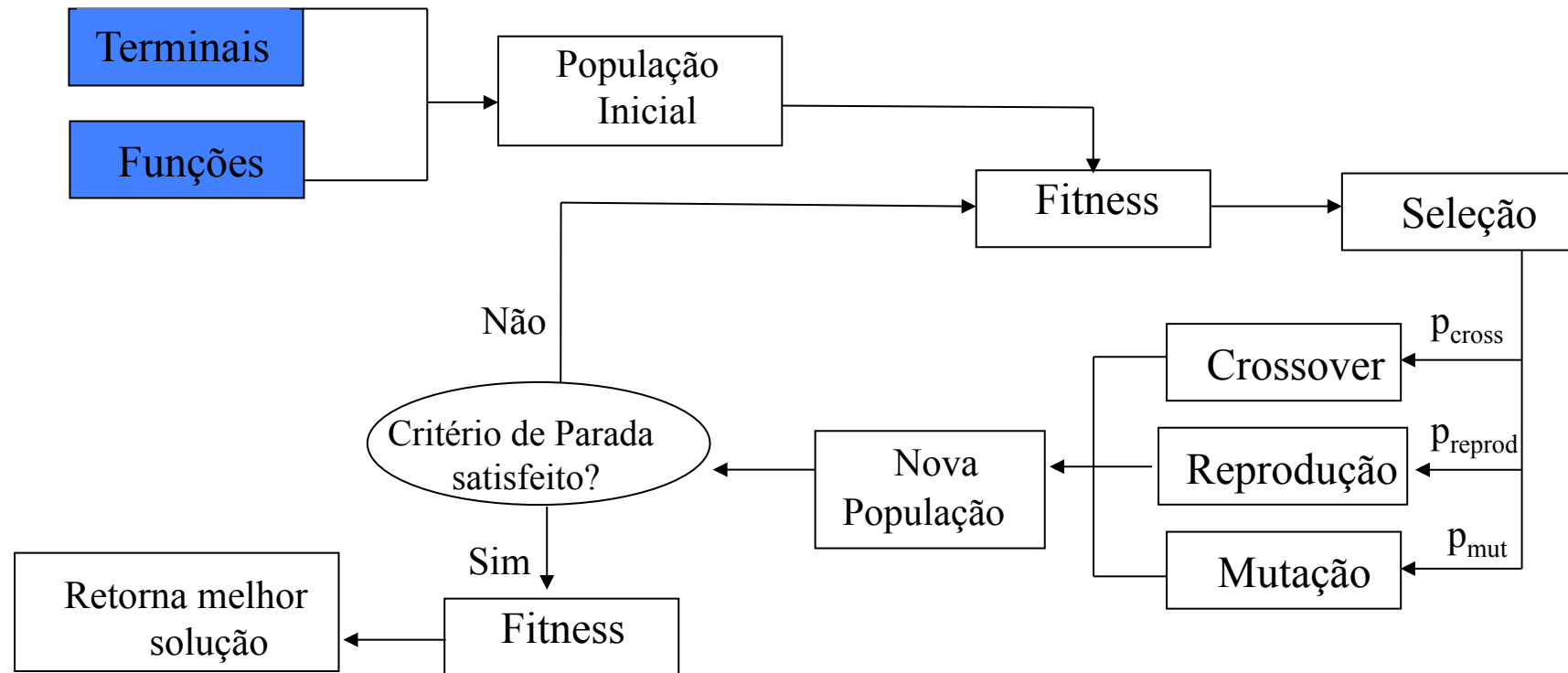


# Programação Genética

## Parte 2/2

Gisele L. Pappa

# Programação Genética

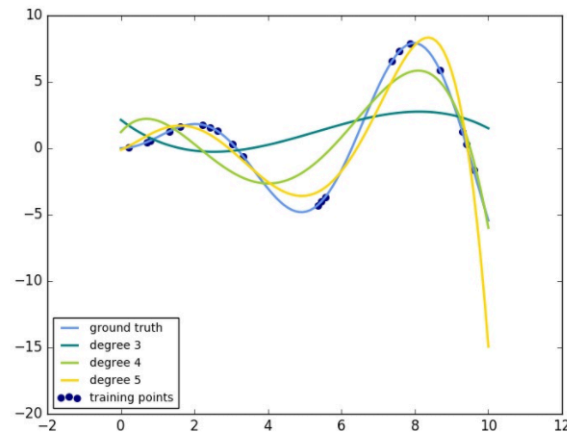


# Regressão Simbólica

- Problema clássico onde GP baseado em árvores obtêm muito sucesso
- Em comparação com outras técnicas de regressão, tem a vantagem de não precisar definir a forma da função sendo buscada a priori
  - Diferença para regressão linear

# Regressão Simbólica

- GP recebe como entrada um conjunto de variáveis que descrevem o problema e uma variável de saída
  - Encontra a melhor função que aproxima as variáveis de entrada com a saída esperada, usando o erro como uma fitness



# Regressão Simbólica

- Pode ser utilizada em qualquer problema onde existam dados de entrada contínuos e uma saída esperada
  - Exemplos incluem:
    - Encontrar funções de “regressão”
    - Encontrar funções de distância
    - Encontrar funções de recomendação de itens

# Introns

- Em biologia, introns são partes “inúteis” do DNA
  - Partes dos genes que não são utilizadas durante a produção de proteínas
- Em GP, introns são partes do genótipo do indivíduo que não tem nenhum efeito na saída do “programa”
- Exemplos:
  - $X = X + 0$
  - $X = X + X - X$
- Introns são comuns em indivíduos de GP

## *Bloat*

- Como resultado do aparecimento de introns, execução de um GP normalmente “incha” (*bloat*)
  - Faz com que um indivíduo cresça incontrolavelmente até que o tamanho máximo seja alcançado

## Introns e *Bloat*

- Possíveis “vantagens” de introns
  - Efeito protetor contra o efeito destrutivo do crossover
  - Quanto maior a porcentagem de introns, maior a probabilidade do crossover trocar introns ao invés de trocar partes úteis do código
- Desvantagens de *bloating*
  - População utilizada muito mais memória
  - Execução do GP torna-se mais lenta

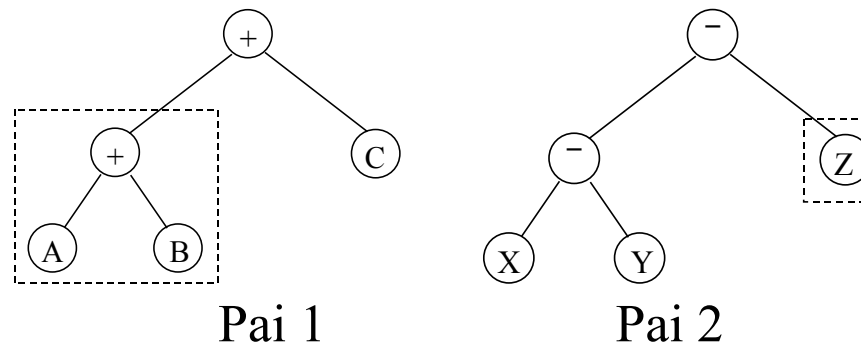


## Efeito destrutivo do Cruzamento

- Cruzamento – representação linear

$I_1$  A B C D E F G     $C_1$  A **B** C D E G **B**    A **B** C D E G **F**  
 $I_2$  C A D F E G B     $C_2$  C A D **F** E **F** G    C A D **F** E **B** G

- Cruzamento – representação por árvores



Gisele L Pappa

## O que fazer em relação a introns e *bloating*

- Estimular a evolução de indivíduos mais simples, utilizando uma pressão de parsimônia (penalizando soluções complexas) na fitness
- Eliminar crossovers destrutivos, i.e., crossover é realizado apenas se os filhos gerados tiveram fitness pelo menos tão boas quanto as fitness dos pais (medida um pouco drástica)
- Implementar um operador que explicitamente remove introns depois que um indivíduo é gerado

# Diferenças “Padrão” entre GA e GP

- **Algoritmos Genéticos (GA)**
  - Representação: originalmente, um vetor binário.
  - Principal operador: crossover (altas probabilidades)
  - Operador secundário: mutação (baixa probabilidade)
- **Programação Genética (GP)**
  - Representação: utilizada não apenas dados, mas funções
  - Objetivo original é evoluir programas ao invés de soluções para uma instância particular do problema
  - Acredita-se que crossover possa ter um efeito destrutivo

## GA versus GP

- Diferença está na interpretação da representação [Woodward 2003]
  - GA o mapeamento entre a descrição e o objeto sendo descrito é sempre um para um
  - GP esse mapeamento é de muitos para um
    - Em regressão simbólica, a mesma função pode ser descrita por diversos indivíduos

## Leitura Recomendada

- A Field Guide to Genetic Programming, Livro online, <http://www.gp-field-guide.org.uk/>

## Curiosidades

- <http://rogeralsing.com/2008/12/07/genetic-programming-evolution-of-mona-lisa/>

# Agradecimentos

- A maioria dos slides dessa aula foram retiradas das aulas de Computação Natural de Alex A. Freitas

# Programação Genética

## Parte 2/2

Gisele L. Pappa