Computação Evolutiva

- Teoria da Evolução Natural - C. Darwin
  - A vida na terra é o resultado de um processo de seleção, pelo meio ambiente, dos mais adaptados/aptos, e por isso mesmo, com mais chances de reproduzir-se.

- A computação evolutiva é um paradigma para resolver problemas.
- Não exige, para resolver um problema, um conhecimento prévio de uma maneira de encontrar a solução.
- Baseada em mecanismos evolutivos da natureza
  - Auto-organização
  - Comportamento adaptativo

- Em Computação Evolutiva, abrimos mão da garantia de obtenção de uma solução ótima para se conquistar a tratabilidade via uma ferramenta de propósito geral.

Para que serve:

1. Validar teorias e conceitos associados à biologia da evolução.
2. Lidar com problemas com os quais não é possível ou é muito custoso obter uma solução detalhada. E.g.: funções não deriváveis (otimização)
3. Não é necessário reiniciar todo o processo de busca de uma solução frente a pequenas mudanças nas especificações do problema dado que refinamentos podem ser feitos a partir da solução atual.

Como simulam este comportamento

- (CE) está baseada em algumas ideias básicas:
  1. População de soluções (e.g., inicialmente aleatória) na qual os indivíduos registram os parâmetros que descrevem uma possível solução.
  2. Função de avaliação: julga a aptidão de cada indivíduo. Apenas atribui uma "nota".

③ Operadores : aplicados a uma dada geração para obtenção de uma próxima geração.

Ⓐ operador seleção - permite escolher os indivíduos (reprodução assexuada) ou um par deles (reprodução sexuada) para gerar descendência.

- A probabilidade de escolha recai sobre os indivíduos mais bem avaliados.

Ⓑ Recombinação - Simula a troca de material genético entre os ancestrais e determina a carga genética dos descendentes.

Ⓒ mutação : - Operador que realiza mudanças aleatórias no material genético.

Como avaliar a adaptação?

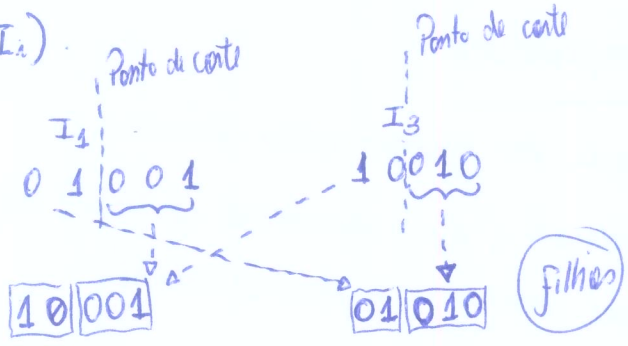
Uma população inicial de soluções evolui ao longo das gerações que são simuladas no processo em direção a soluções mais adaptadas por meio dos operadores de seleção, mutação e recombinação.

Indivíduos formam uma população

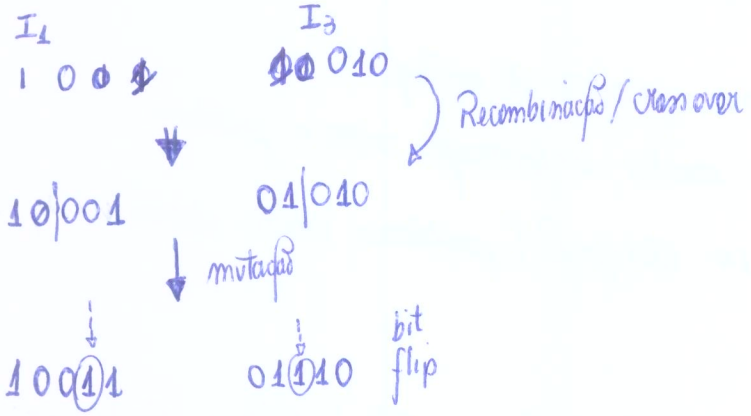
População		I		I
*	①	0 1 0 0 1	④	0 0 1 1 0
	②	1 0 0 1 1	⑤	0 0 0 0 1
*	③	1 0 0 1 0	⑥	0 1 0 1 0

função de avaliação $f(I_i)$

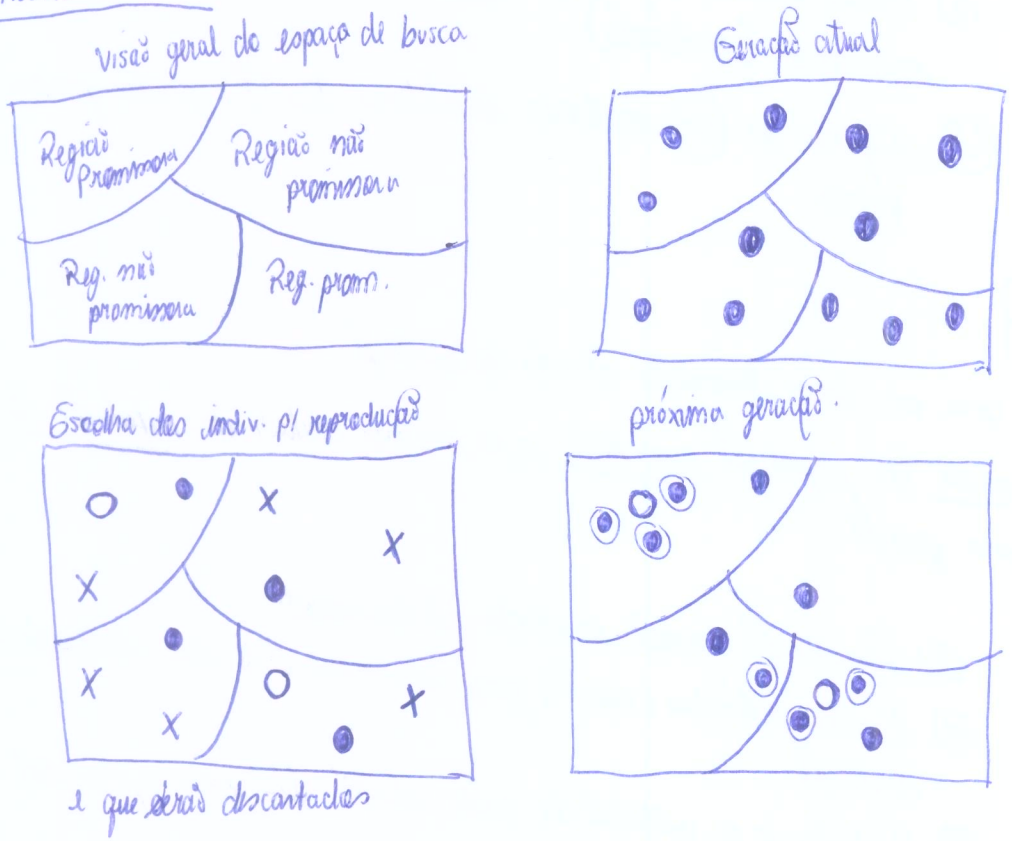
Recombinação



Mutação



Entendendo melhor



Diferentes Abordagens de Computação Evolutiva

1. Estratégias Evolutivas
2. Programação Evolutiva
3. Algoritmos Genéticos
4. Programação Genética.

Estratégias Evolutivas

① Empregam apenas operadores de mutação.
② Não necessita de muitas informações sobre o problema.
③ Muito utilizada em otimização (problemas multi-dimensionais, multi-modais e não lineares.

Algoritmo Básico

① População com (m) indivíduos. Cada um c/ (n) genes
② Cada indivíduo produz (K/m) descendentes com pequenas mudanças (mutações).
③ Apenas os (m) melhores indivíduos dos (K) gerados permanecem vivos.

Programação Evolutiva

① Cada indivíduo gera um único descendente através de mutação.
② A melhor metade da população ascendente e a melhor metade da população descendente formam a nova geração.

Algoritmo Básico

① População inicial escolhida aleatoriamente.
② Cada indivíduo (solução) gera um novo indivíduo utilizando-se mutação.
③ Calcula-se a aptidão de cada solução. Os mais aptos são retidos.

Algoritmos Genéticos

① Combinam variações aleatórias com seleção de indivíduos mais aptos.
② mantém uma população de ~~sol~~ soluções candidatas.
③ Busca multi-direcional e ~~mas~~ paralelismo. Algoritmos genéticos possuem o que chamamos de paralelismo intrínseco.

Algoritmos Genéticos

④ Utiliza os operadores de Recombinação e mutação

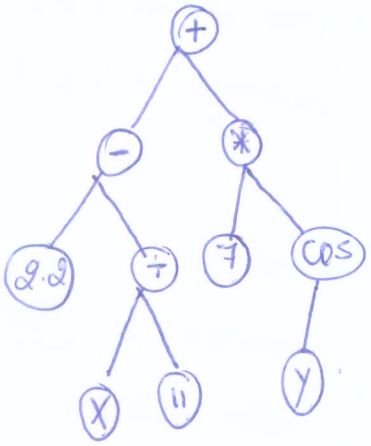
Algoritmo Básico

① Inicia população (soluções candidatas).
② Avalia cada cromossomo (solução).
③ Cria novos cromossomos a partir da população atual (mutação ⊕ recombinação).
④ Substitui ascendentes por descendentes.
⑤ Se atingir critério de paragem, termina.

Programação Genética

① Extensão de algoritmos genéticos.
② Indivíduos são programas.
③ Representação comum: árvores.

Exemplo



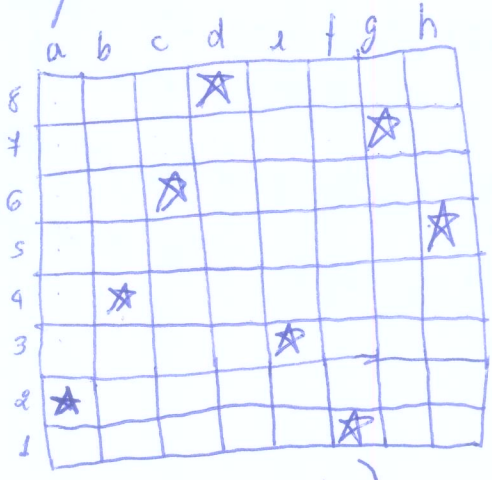
$$\left(2.2 - \frac{X}{11}\right) + \left(7 * \cos(y)\right).$$

percurso em pré-ordem.

Problemas Comuns

① Codificação dos indivíduos : binária (vs) ponto flutuante.

↝ Uma codificação errada pode levar a problemas de convergência prematura (mín. local) e valores inválidos (fora de domínio).



Problema das ⑧ rainhas

(2, 4, 6, 8, 3, 1, 7, 5).

② Como criar uma população inicial ? Aleatório ou usar um método simples ?

③ Operadores genéticos
- (a) Quais os seus parâmetros
- (b) mutação deve ser de valor alto ou baixo ?
- (c) Como deve ser a recombinação ? Qual o ponto de corte ?

④ Como selecionar os indivíduos para a próxima geração ?

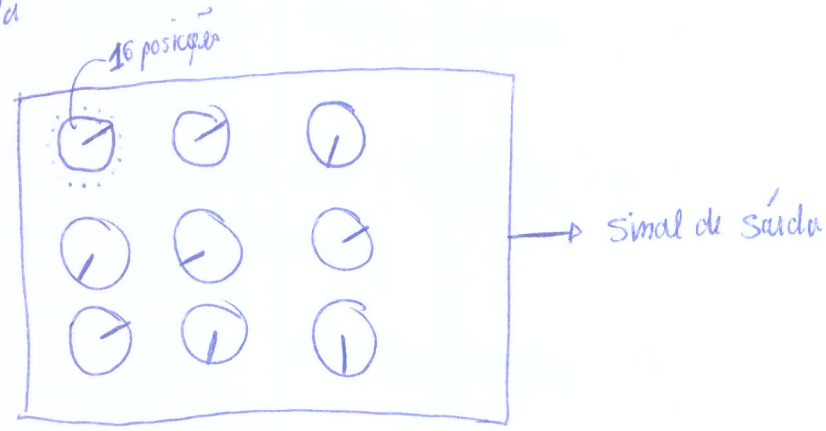
(a) modelo roleta : probabilidade proporcional ao fitness. Com isso, podemos perder o melhor indivíduo. Solução : usar elitismo.

(b) seleção baseada em rank : ordena pelo fitness.

Exemplos de problemas

Problema ④ : Voltagem

Suponha que tenhamos um problema de otimização em que queremos a maior voltagem possível em uma saída



$16^9 \sim 68,7 \times 10^9$ combinações

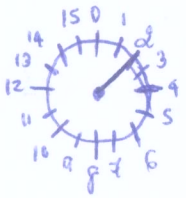
Configuração : ⑯ posições.
⑨ botões

Objetivo : encontrar a combinação que maximiza o sinal de saída.

Nosso primeiro problema é escolher a codificação

↳ ⑯ posições possíveis : ④ bits

↳ Indivíduo ④ bits * ⑨ botões = 36 bits.



Posição	Representação	Posição	Representação
0	0000	4	0100
1	0001	5	0101
2	0010	6	0110
3	0011	7	0111

Posição atual : **0010**

Cromossomo (solução candidata)

operadores genéticos : mutação simples e recombinação uniforme

O que é recombinação uniforme: Para cada posição (gene), escolhe-se com certa probabilidade se pai (X) ou (Y) é que contribui.

função de adaptação: voltagem de saída.

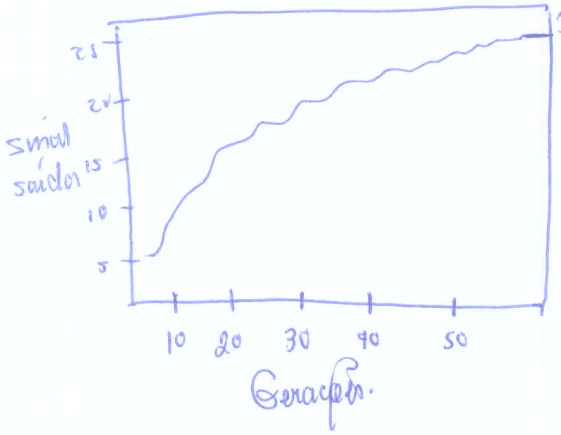
Valores arbitrários (A) Probabilidade de bits (1) na população: 50%.

(B) Taxa de mutação: 3%

(C) Recombinação: 60%

(D) Tipo de seleção na recombinação: bi-classista (50% bons, 10% ruins)

Resultado

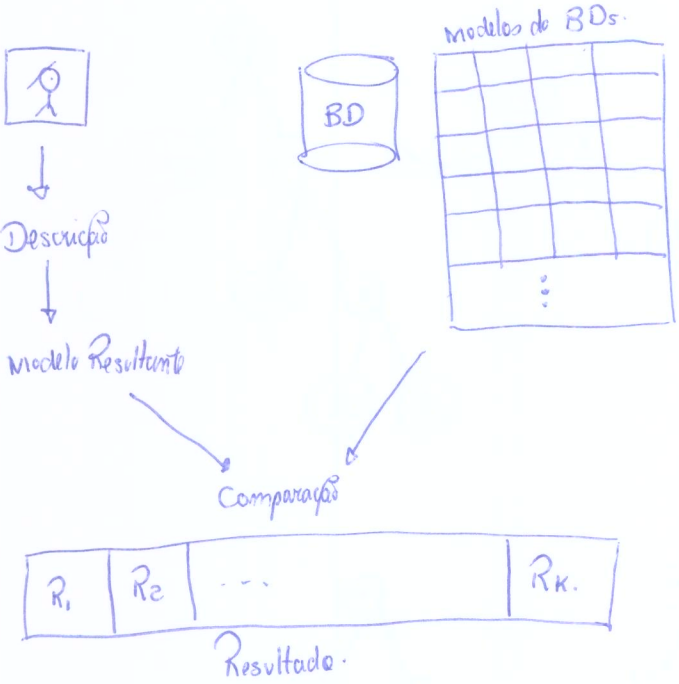


* 1500 indivíduos
* Quantas combinações eram? $\sim 10^9$.
* Resultado em < (1) segundo.

Paralelismo intrínseco!

Problema #2 : CBIR

Consulta Q



Como descrever uma imagem? Inúmeras formas. Qual a melhor?

① Cor: (1, 50, 25, 35, ..., 20)

② ~~Forma~~ Textura: (0, 1, 1, 0, 0, 1 ...)

③ Textura: (235, 35, 200, 120, ...).

Dados esses números que representam imagens, como combiná-los?

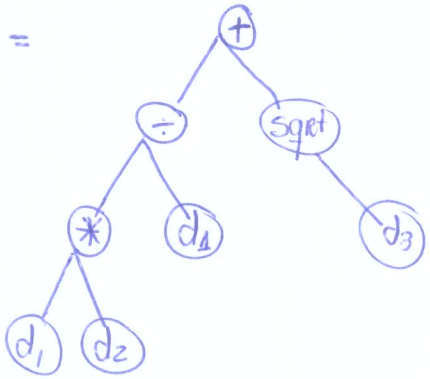
- soma
- subtração
- Raiz
- Log
- multiplicação
- ⋮

Como conseguir boas combinações? Programação Genética.

$f(d_1, d_2, d_3)$ =

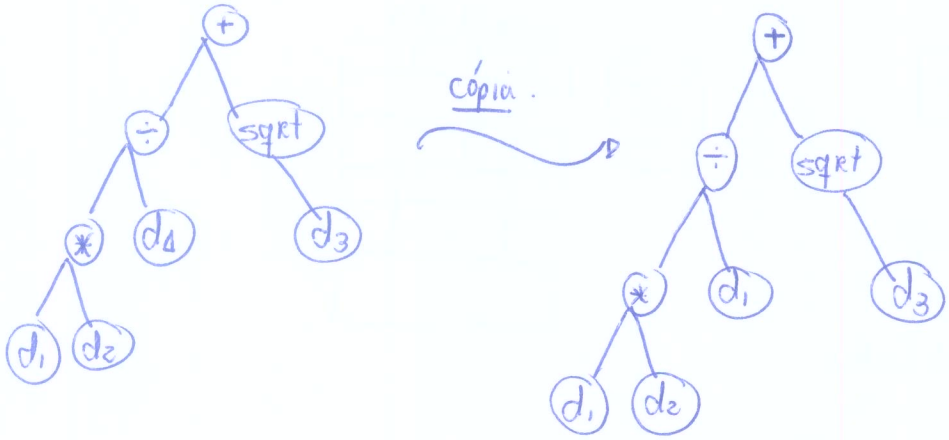
d_1, d_2, d_3: descritores

f: função de combinação

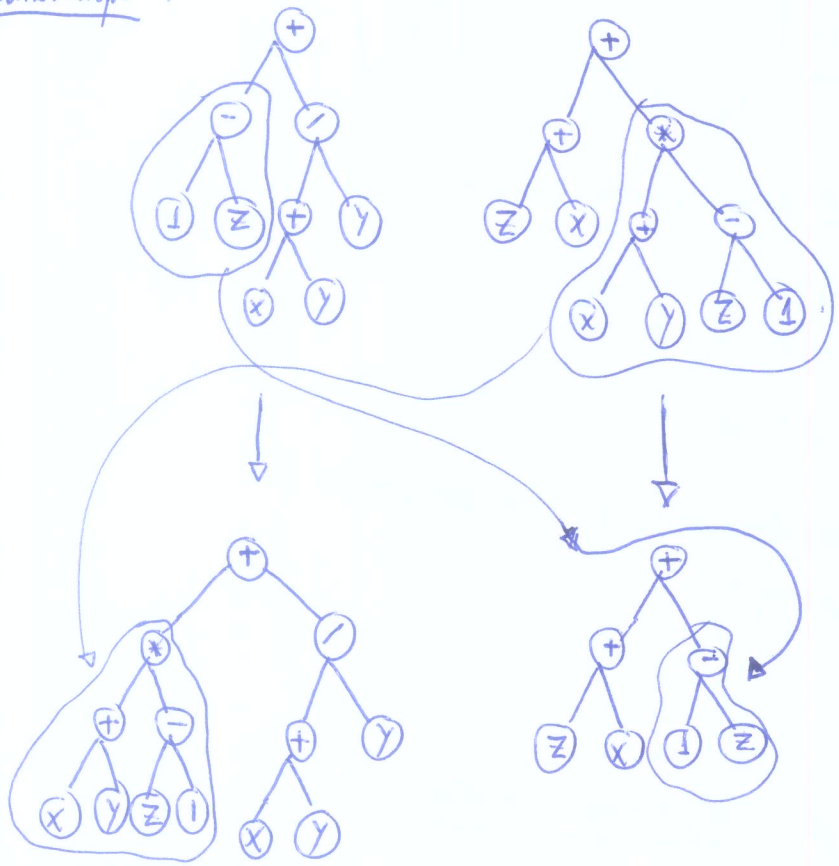


$$\frac{(d_1 * d_2)}{d_1} + \sqrt{d_3}$$

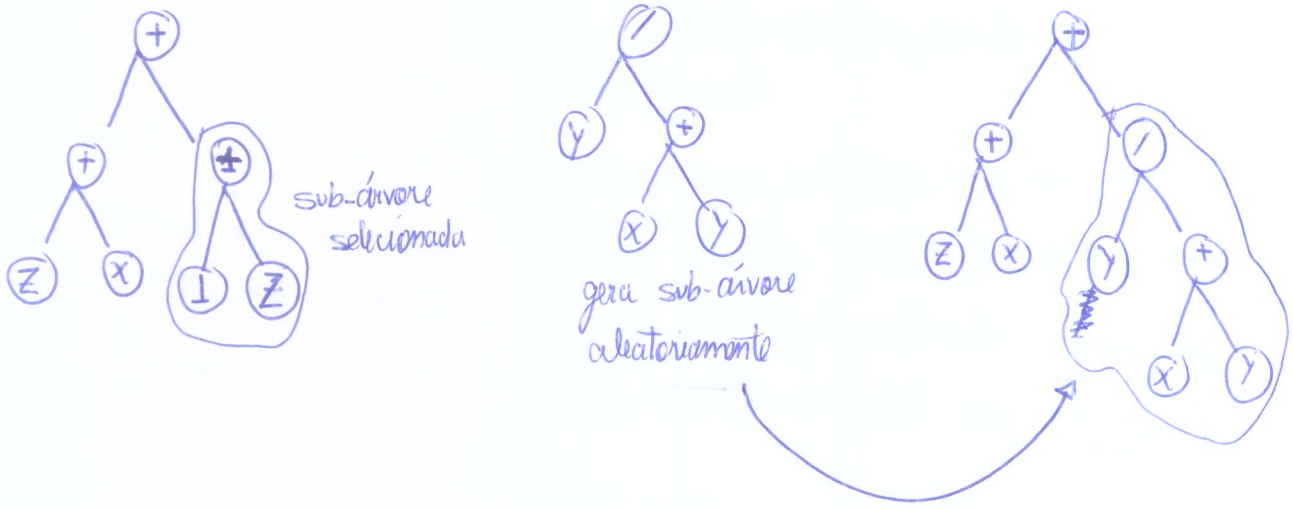
Reprodução



Recombinação:



Mutação



Questões interessantes

(1) Como seria se usássemos algoritmos genéticos?

(2) GP x GA? Quem é melhor?
PG x AG

(3) E no caso específico de recuperação de imagens?

Lições da aula

(1) Computação evolutiva pode ajudar a resolver problemas difíceis.

(2) Uti em machine Learning? Como?

(4) otimização.
(4) Busca de parâmetros.
(4) Elab. de soluções candidatas, etc.

(3) Baseada em leis da natureza { seleção natural, mutação, reprodução.

(4) Algoritmos Evolucionários não devem/não podem ser considerados caixas-preta prontos para o uso mas sim como um conjunto de procedimentos gerais que podem ser adaptados facilmente a diversas aplicações.

Exercícios para treinar

① Considere o clássico problema TSP
(Traveller Salesperson problem)

Suponha que um indivíduo precise partir de um ponto (A) visitar outros (99) pontos diferentes e retornar a (A). Dadas as coordenadas dos 100 pontos, descubra o percurso de menor distância que passe uma única vez por todos os ~~[illegible]~~ pontos e retorne à origem (A).

Pergunta-se ① Podemos usar AG? Porque?
② Quantas combinações teríamos se tentássemos na força bruta?
③ Imagine e proponha uma codificação para o problema.
④ Como gerar uma solução inicial?
⑤ Proponha uma função de adequação.

Leitura Recomendada: * Computação Evolutiva: uma abordagem pragmática
Fernando Von Zuben.

Comparação com técnicas baseadas em Gradiente

* Técnicas de descida do gradiente exigem
① função objetivo diferenciável.
② Baixo custo de diferenciação.
③ O que fazer se a diferenciação não é possível ou muito cara?
