

Inteligência Artificial

Tópico 03 - Parte 05

Resolução de Problemas por Buscas

Algoritmos Genéticos

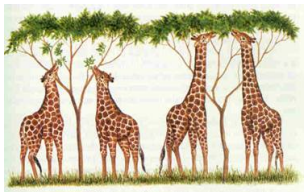
Profa. Dra. Priscila Tiemi Maeda Saito
✉ priscilasaito@ufscar.br

1 Resolução de Problemas por Buscas

- Busca Local - Além da Busca Clássica
 - Hill-Climbing
 - Simulated Annealing
 - Local Beam
 - Algoritmos Genéticos

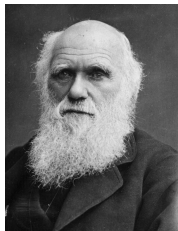
Algoritmos Genéticos

- Engloba métodos e técnicas computacionais inspirados
 - ▶ na teoria da evolução das espécies, de seleção natural (Darwin)
 - ▶ na Genética iniciada por Mendel
- Bases da evolução:
 - ▶ seres mais adaptados aos seus ambientes sobrevivem
 - ▶ características genéticas de tais seres são herdadas pelas próximas gerações
 - ▶ diversidade é gerada por cruzamento e mutações



Algoritmos Genéticos

- 1859 - Charles Darwin publica o livro “A Origem das Espécies”



As espécies evoluem pelo princípio da seleção natural e sobrevivência do mais apto

- 1865 - Gregor Mendel, pai da genética, apresenta experimentos do cruzamento genético de ervilhas



Algoritmos Genéticos

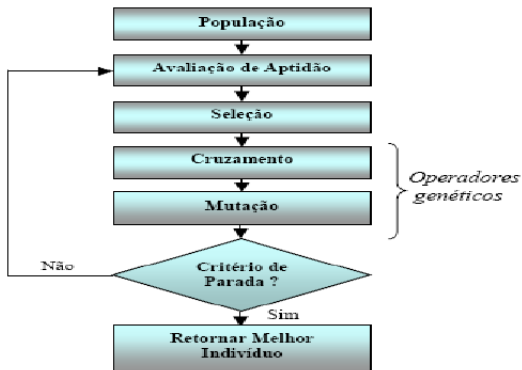
- Nos anos 1960, John Holland e seus alunos propuseram a construção de um algoritmo de busca e otimização: os algoritmos genéticos



Algoritmos Genéticos

- Algoritmos genéticos usam como base e procuram combinar:
 - ▶ teoria da **evolução das espécies** - a sobrevivência das estruturas/soluções mais adaptadas a um ambiente/problema
 - ▶ estruturas **genéticas** - utiliza conceitos de hereditariedade e variabilidade genética para troca de informações entre as estruturas, visando a melhoria das mesmas

Algoritmos Genéticos



Algoritmos Genéticos - População

- A população de um algoritmo genético é o conjunto de indivíduos que estão sendo cogitados como solução

Cada indivíduo é uma possível solução do problema

Algoritmos Genéticos - Indivíduo

- Um indivíduo no AG é um **cromossomo**
- Ou seja, um indivíduo é um conjunto de atributos da solução
- Geralmente é uma cadeia de bits que representa uma solução possível para o problema

Exemplo

População de tamanho $n=4$

Geração de indivíduos, com seus cromossomos

Cada elemento do vetor é um gene, um atributo da solução

Indivíduo 1 = [1 1 1 0 1]

Indivíduo 2 = [0 1 1 0 1]

Indivíduo 3 = [0 0 1 1 0]

Indivíduo 4 = [1 0 0 1 1]

Algoritmos Genéticos - Indivíduo

- Outras representações são possíveis
- Boa representação depende do problema

Exemplo - Problema da Mochila

- Dada uma lista de coisas com preços e tamanhos
- É fornecido o valor da capacidade da mochila
- Escolha as coisas de forma a maximizar o valor daquilo que cabe dentro da mochila, sem ultrapassar sua capacidade

Cada bit é usado para dizer se a coisa correspondente está ou não na mochila

- Crom: A = 1 0 1 1 0 0 1 0 1 1
- Crom: B = 1 1 1 1 1 1 0 0 0 0

Codificação Binária

Cada cromossomo é uma string de bits - 0 ou 1

Algoritmos Genéticos - Indivíduo

- Outras representações são possíveis
- Boa representação depende do problema

Exemplo - Problema do caixeiro viajante

- São dadas cidades e as distâncias entre elas
- O caixeiro viajante tem que visitar todas as cidades
- Encontrar a sequência de cidades em que as viagens devem ser feitas de forma que a distância percorrida seja a mínima possível

Cromossomos descrevem ordem de visita das cidades

- Crom: A = 1 5 3 2 6 4 7 9 8
- Crom: B = 8 5 6 7 2 3 1 4 9

Codificação por Permutação

Cada cromossomo é uma string de números que representa uma posição em uma sequência

Algoritmos Genéticos - Indivíduo

- Outras representações são possíveis
- Boa representação depende do problema

Exemplo - encontrar pesos para uma rede neural dada uma estrutura

- É dada uma rede neural com arquitetura definida
- Encontre os pesos entre os neurônios da rede de forma a obter a resposta desejada da rede

Valores reais representam pesos em uma rede neural

- Crom: A = 1.2324 5.3243 0.4556 2.3293 2.4545
- Crom: B = ABDJEI FJDHDI ERJFDL DFLFEGT
- Crom: C = (back), (back), (right), (forward), (left)

Codificação por Valor

Cada cromossomo é uma sequência de valores

Algoritmos Genéticos - Função de Aptidão

- Função de *fitness*, função de custo \rightarrow determina uma nota a cada indivíduo
- Nota avalia quão boa é a solução que este indivíduo representa

Se indivíduo I_1 representa uma solução melhor do que $I_2 \rightarrow$ avaliação de I_1 deve ser maior do que de I_2

O objetivo de um AG pode ser maximizar o número de 1s

Indivíduos	Função de aptidão (fitness)
[11101]	4
[01101]	3
[00110]	2
[10011]	3
Aptidão média	3

Algoritmos Genéticos - Função de Aptidão

- Função de avaliação deve ser escolhida cuidadosamente
- Cada problema tem sua própria função de aptidão
 - ▶ embutir todo o conhecimento que se possui sobre o problema a ser resolvido
- Pode envolver uma (otimização de função) ou mais medidas (otimização multi-objetivo)
 - ▶ ex.: projeto de ponte → custo, tempo de construção, capacidade máxima
- Métodos para calcular índice de aptidão
 - ▶ padrão
 - ▶ baseada em ranking

Algoritmos Genéticos - Função de Aptidão

- Função de Aptidão Padrão
 - ▶ utiliza apenas informação sobre “qualidade do cromossomo”

$$f_i = \frac{q_i}{\sum_j q_j}$$

q = aptidão do cromossomo

Algoritmos Genéticos - Função de Aptidão

- Função de Aptidão Padrão

Cromossomo	Grau	Aptidão Padrão
1 4	4	0.4
3 1	3	0.3
1 2	2	0.2
1 1	1	0.1

Algoritmos Genéticos - Função de Aptidão

- Função de Aptidão baseada em Ranking
 - ▶ aptidão padrão → escolha da escala do índice de aptidão pode prejudicar reprodução
 - ▶ ex.: 1, 10, 100
 - ▶ utilizar ranking de cromossomos por aptidão
 - ★ utiliza medida de qualidade apenas para definir um “ranking” de cromossomos por aptidão
 - ★ seleção baseada apenas nos ranks

Cromossomo	Valor de Aptidão	Aptidão Padrão	Rank	Aptidão Rank
1 4	4	0.4	1	0.667
3 1	3	0.3	2	0.222
1 2	2	0.2	3	0.074
1 1	1	0.1	4	0.025

Algoritmos Genéticos - Seleção

- De acordo com a teoria de Darwin, o melhor sobrevivente para criar a descendência é selecionado
- Privilegiar indivíduos com função de **avaliação alta**
 - ▶ não desprezar completamente indivíduos com função de **avaliação** extremamente **baixa**
 - ▶ indivíduos com péssima avaliação podem ter características genéticas favoráveis à criação de um indivíduo ótimo
 - ★ características podem não estar presentes em nenhum outro cromossomo

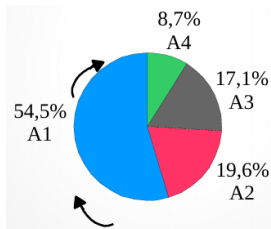
Algoritmos Genéticos - Seleção

- Há muitos métodos para selecionar o melhor cromossomo
 - ▶ seleção por **roleta**
 - ▶ seleção por **torneio**
 - ▶ seleção por **amostragem universal estocástica**
 - ▶ ...

A seleção dirige o AG para as melhores regiões do espaço de busca

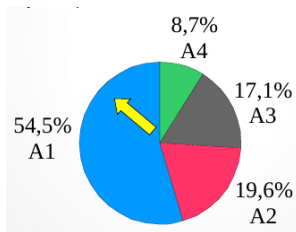
Algoritmos Genéticos - Seleção por Roleta

- Para visualizar este método considere um círculo dividido em N regiões (tamanho da população)
 - ▶ onde a área de cada região é proporcional à aptidão do indivíduo



Algoritmos Genéticos - Seleção por Roleta

- Coloca-se sobre este círculo uma “roleta”
- A roleta é girada um determinado número de vezes, dependendo do tamanho da população
- São escolhidos como indivíduos que participarão da próxima geração, aqueles sorteados na roleta



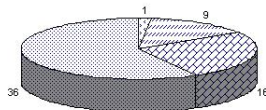
Algoritmos Genéticos - Seleção por Roleta

Exemplo

Indivíduo	Avaliação
0001	1
0011	9
0100	16
0110	36
Total	62

Aptidão Relativa (parte da roleta)
1.61
14.51
25.81
58.07
100.0

Roleta Viciada para População Exemplo



□ "0001" □ "0011" □ "0100" □ "0110"

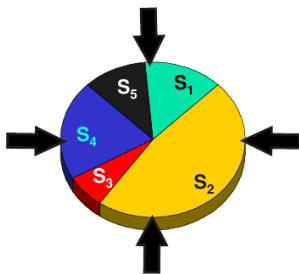
Algoritmos Genéticos - Seleção por Torneio

- Escolhe n indivíduos (e.g., $n=3$) aleatoriamente, com mesma probabilidade
- Cromossomo com maior aptidão é selecionado para a população intermediária
- Processo se repete até que a população intermediária seja preenchida
- Parâmetro n , tamanho do torneio, define pressão seletiva
 - ▶ \uparrow número de indivíduos que participam do torneio $\rightarrow \uparrow$ pressão seletiva
 - ▶ indivíduo tem que ser melhor do que uma quantidade maior de competidores

Indivíduo	Aptidão	Candidatos \rightarrow Vencedor
10110	2.23	$S_1, S_2, S_5 \rightarrow S_2$
11000	7.27	$S_2, S_4, S_5 \rightarrow S_2$
11110	1.05	$S_5, S_1, S_3 \rightarrow S_1$
01001	3.35	$S_4, S_5, S_3 \rightarrow S_4$
00110	1.69	$S_3, S_1, S_5 \rightarrow S_1$

Algoritmos Genéticos - Seleção por Amostragem Universal Estocástica

- Conhecido como SUS (*Stochastic Universal Sampling*)
- Variação do método da roleta
 - ▶ utiliza n agulhas igualmente espaçadas ao invés de uma
 - ★ n é o número de indivíduos a serem selecionados
 - ★ ao invés de n vezes, a roleta é girada uma única vez
 - ▶ exibe menos variância do que as repetidas chamadas do método da roleta



Algoritmos Genéticos - Operadores Genéticos

- Um conjunto de operações é necessário para que, dada uma população, seja possível gerar populações sucessivas que (espera-se) melhorem sua aptidão com o tempo
- Estas operações são os **operadores genéticos**
 - ▶ **cruzamento**
 - ▶ **mutação**
 - ▶ **elitismo**

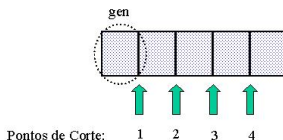
Os operadores genéticos permitem explorar áreas desconhecidas do espaço de busca

Algoritmos Genéticos - Cruzamento

- O operador **cruzamento** (*crossover* ou recombinação) cria novos indivíduos, misturando características de **dois indivíduos pais**
- O resultado desta operação é **um indivíduo** que potencialmente combine as melhores características dos indivíduos usados como base
- Alguns tipos de cruzamento são:
 - ▶ cruzamento em um ponto
 - ▶ cruzamento em dois pontos
 - ▶ cruzamento multi-pontos
 - ▶ uniforme

Algoritmos Genéticos - Cruzamento

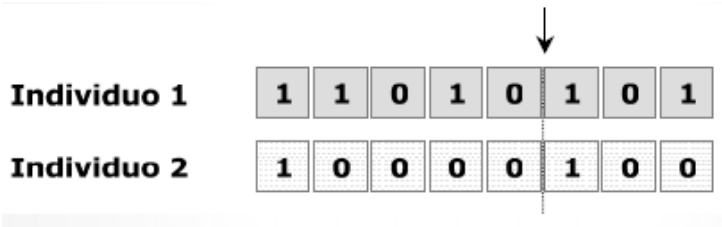
- Um ponto de corte deve ser selecionado
- Constitui uma posição entre dois genes de um cromossomo
- Cada indivíduo de n genes contém $n - 1$ pontos de corte possíveis



- Separação do pai em duas partes: esquerda e direita do ponto de corte
 - ▶ partes não necessariamente têm o mesmo tamanho
- Primeiro filho: concatenação da parte esquerda do primeiro pai + parte direita do segundo pai
- Segundo filho: concatenação da parte esquerda do segundo pai + parte direita do primeiro pai

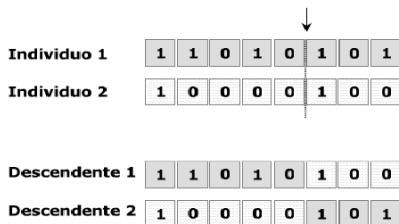
Algoritmos Genéticos - Cruzamento de um Ponto

- No cruzamento de um ponto, divide-se cada progenitor em duas partes, em uma localidade k (escolhida aleatoriamente)

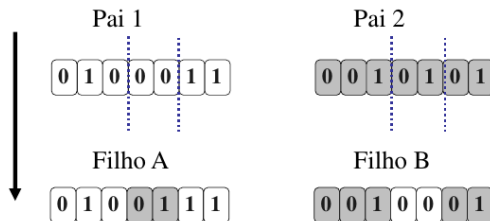


Algoritmos Genéticos - Cruzamento de um Ponto

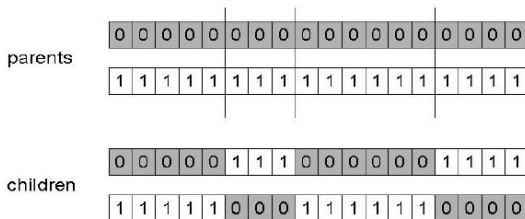
- O descendente 1 consiste em genes 1 a k do progenitor 1, e genes $k + 1$ a $n - 1$ do progenitor 2
- O descendente 2 é “reverso”



Algoritmos Genéticos - Cruzamento de Dois Pontos

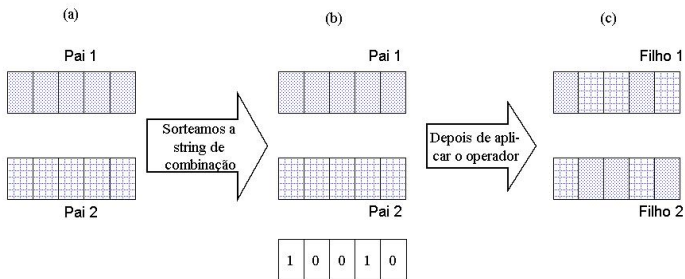


Algoritmos Genéticos - Cruzamento de n Pontos



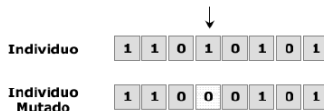
Algoritmos Genéticos - Cruzamento Uniforme

- Para cada gene é sorteado um número zero ou um
- Se o sorteado for 1, um filho recebe o gene do primeiro pai e o segundo filho o gene do segundo pai
- Se o sorteado for 0, o primeiro filho recebe o gene do segundo pai e o segundo filho recebe o gene do primeiro pai



Algoritmos Genéticos - Mutação

- A mutação modifica aleatoriamente alguma característica do indivíduo, sobre o qual é aplicada



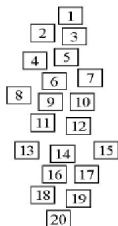
- O operador de mutação é necessário para a introdução e manutenção da diversidade genética da população
- Desta forma, a mutação assegura que a probabilidade de se chegar a qualquer ponto do espaço de busca, possivelmente, não será zero
 - ▶ reduz chance de ficar preso em mínimos locais
 - ▶ taxa de mutação pequena \rightarrow 0.5% ou 1%

Algoritmos Genéticos - Elitismo

- Conjunto de indivíduos mais adaptados é automaticamente selecionado para a próxima geração
- Evita modificações deste(s) indivíduo(s) pelos operadores genéticos
 - ▶ utilizado para que os melhores indivíduos não desapareçam da população

Algoritmos Genéticos - Gerações

- Algoritmo é iterado até algum critério de parada
 - ▶ tempo de execução, número de gerações, valores de aptidão mínimo ou médio
- A cada passo, um novo conjunto de indivíduos é gerado a partir da população anterior
- A este novo conjunto dá-se o nome de geração
- Com a criação de uma grande quantidade de gerações que é possível obter resultados dos AGs



Algoritmos Genéticos - Algoritmo

Algoritmo_Genetico

- p = tamanho da população
 - r = taxa de cruzamento
 - m = taxa de mutação
- 1 $P \leftarrow$ gerar aleatoriamente p indivíduos
 - 2 **Para cada** i em P , computar Aptidão(i)
 - 3 **Enquanto** critério_parada não é atingido
 - 1 selecionar p membros de P para reprodução
 - 2 aplicar cruzamento a pares de indivíduos selecionados segundo taxa r , adicionando filhos em PS
 - 3 realizar mutação em membros PS , segundo taxa m
 - 4 $P \leftarrow PS$
 - 5 **Para cada** i em P , computar Aptidão(i)
 - 4 **Retornar** o indivíduo de P com maior aptidão

Codificação e avaliação de aptidão são pontos chave

Algoritmos Genéticos - Parâmetros

- O desempenho dos algoritmos genéticos é fortemente influenciado pela definição dos seus parâmetros
- **Tamanho da população**
 - ▶ populações pequenas: cobrem pouco o espaço de busca
 - ▶ populações grandes: apesar de evitar mínimos locais, requer mais recursos computacionais e tempo
- **Intervalo de geração:**
 - ▶ porcentagem da população que será substituída
 - ★ grande (comum): filhos substituem pais
 - ★ pequena: “pais e filhos convivem”

Algoritmos Genéticos - Parâmetros

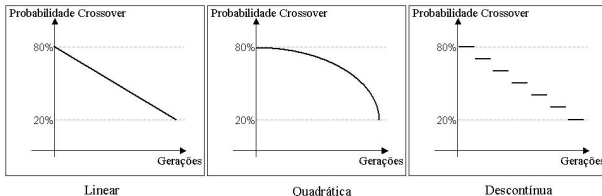
- O desempenho dos algoritmos genéticos é fortemente influenciado pela definição dos seus parâmetros
- **Taxa de cruzamento**
 - ▶ se for muito baixa: busca pode estagnar
 - ▶ se for muito alta: boas estruturas podem ser perdidas
- **Taxa de mutação:**
 - ▶ possibilita que qualquer ponto de espaço de busca seja atingido
 - ▶ se for muito alta: busca aleatória

Algoritmos Genéticos - Interpolando Operadores

- É possível aumentar ou diminuir a **incidência de cada um dos operadores** sobre a população e assim ter mais controle sobre o desenvolvimento dos cromossomos
- Cada operador pode receber uma avaliação
- Normalmente o operador de cruzamento recebe um fitness bem maior que o operador de mutação
- A porcentagem de aplicação de cada operador não precisa ser fixa

Algoritmos Genéticos - Interpolando Operadores

- No início, deseja-se executar **muita reprodução e pouca mutação**
 - ▶ há muita diversidade genética e deseja-se explorar o máximo possível o espaço de soluções
- Após um grande número de gerações, há pouca diversidade genética na população
 - ▶ extremamente interessante que o operador de mutação fosse escolhido mais frequentemente

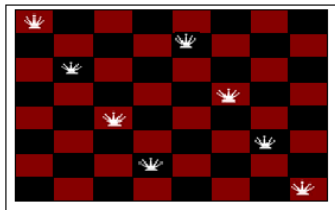


Questões Importantes

- Como criar cromossomos?
- Qual tipo de codificação usar?
 - ▶ primeiras perguntas que devem ser feitas ao resolver um problema com AG
 - ▶ codificação dependerá fortemente do problema
- Como escolher os pais para a realização do crossover?
- A geração de uma população a partir de duas soluções pode causar a perda da melhor solução. O que fazer?

Algoritmos Genéticos - Exemplo

- **Codificando** o problema
- Exemplo: problema 8 rainhas
 - ▶ cada estado deve especificar posição de 8 rainhas, em coluna com 8 quadrados
 - ★ $8 \times \log_2 8 = 24$ bits se codificação binária



- Binário = 111 101 011 001 110 100 010 000
- Inteiro = 8 6 4 2 7 5 3 1

Algoritmos Genéticos - Exemplo

- (a) Gerando **população inicial**
- 8 dígitos, com valores de 1 a 8
 - ▶ exemplo: população com 4 cadeias de 8 dígitos
 - ★ representam estados de 8 rainhas

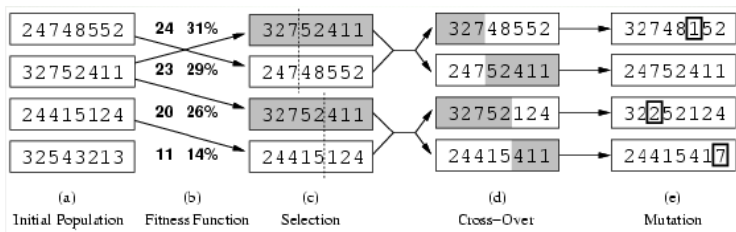
2	4	7	4	8	5	5	2
3	2	7	5	2	4	1	1
2	4	4	1	5	1	2	4
3	2	5	4	3	2	1	3

Algoritmos Genéticos - Exemplo

24748552	24
32752411	23
24415124	20
32543213	11

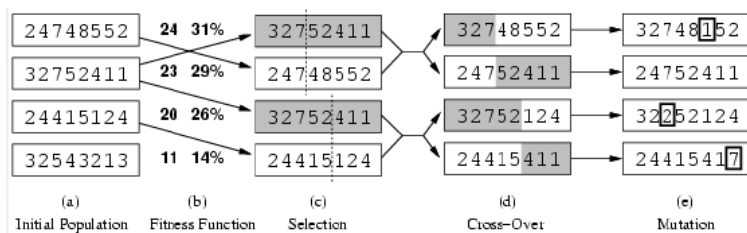
- (b) **Avaliação**
- Função de avaliação (aptidão)
 - ▶ deve retornar valores maiores para estados melhores
 - ▶ ex.: 8 rainhas: número de pares de rainhas não-atacantes
 - ★ valor 28 para uma solução
 - ★ $\min = 0$, $\max = 8 \times 7/2 = 28$

Algoritmos Genéticos - Exemplo



- (c) **Seleção**
- Proporcional à aptidão do indivíduo
 - ▶ vários métodos
 - ▶ todos tendem a privilegiar indivíduos mais aptos
 - ★ $24/(24+23+20+11) = 31\%$
 - ★ $23/(24+23+20+11) = 29\%$

Algoritmos Genéticos - Exemplo

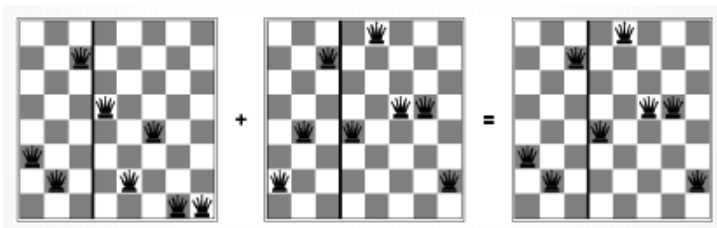


- (d) **Cruzamento**
- Indivíduos selecionados formam pares
- Operador de cruzamento combina pares
 - ▶ ponto de cruzamento gerado ao acaso

Algoritmos Genéticos - Exemplo

- Cruzamento

- ▶ $327|52411 + 247|48552 = 327|48552$



Algoritmos Genéticos - Exemplo



- (e) **Mutação**
- Mudança aleatória do valor de um gene
 - ▶ com pequena probabilidade
- Introduz variações aleatórias
 - ▶ permitindo soluções pularem para diferentes partes do espaço de busca

Algoritmos Genéticos - Exemplo

- Problema: Caixeiro Viajante
- Deve-se encontrar o caminho mais curto para percorrer n cidades sem repetição
- Como representar os indivíduos?
 - ▶ cada indivíduo pode ser representado por uma lista ordenada de cidades
 - ▶ indica a ordem em que cada uma será visitada

Exemplo

3 5 7 2 1 6 4 8

Algoritmos Genéticos - Exemplo

- Cada cromossomo tem que conter **todas as cidades** do percurso, **apenas uma vez**
- Considerando 8 cidades:

Exemplo - Cromossomos válidos

1 2 3 4 5 6 7 8

8 7 6 5 4 3 2 1

1 3 5 7 2 4 6 8

...

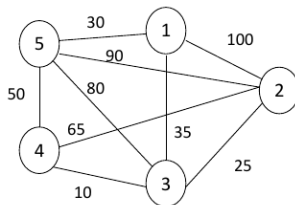
Exemplo - Cromossomos inválidos

1 5 7 8 2 3 6 → Falta a cidade 4

1 5 7 8 2 3 6 5 → Falta a cidade 4 e a cidade 5 está representada 2 vezes

Algoritmos Genéticos - Exemplo

- Qual a função de avaliação?
- A função de avaliação consiste em somar todas as distâncias entre cidades consecutivas



Exemplo

O cromossomo 1 3 5 4 2 tem avaliação igual a
 $35 + 80 + 50 + 65 = 230$

Algoritmos Genéticos - Exemplo

Exemplo - Recombinação (uniforme)

- Pai 1 → 3 5 7 2 1 6 4 8
- Pai 2 → 2 5 7 6 8 4 3 1
- Gera-se uma string de bits aleatória do mesmo tamanho que os pais
 - ▶ 1 0 0 1 0 1 0 1
- Copia-se para o filho 1 os elementos do pai 1 referentes àquelas posições onde a string de bits possui um 1
 - ▶ 3 _ _ 2 _ 6 _ 8
- Elementos não copiados do pai 1
 - ▶ 5 7 1 4
- Permuta-se essa lista de forma que os elementos apareçam na mesma ordem que no pai 2 e copia-se eles para dentro do filho 1
 - ▶ 3 5 7 2 4 6 1 8

Algoritmos Genéticos - Exemplo

Exemplo - Mutação

Indivíduo 3 5 7 2 1 6 4 8

- Escolhem-se dois elementos aleatórios dentro do cromossomo e trocam-se as suas posições
 - ▶ 3 5 7 2 1 6 4 8
- Novo indivíduo mutante:
 - ▶ 3 1 7 2 5 6 4 8

Algoritmos Genéticos

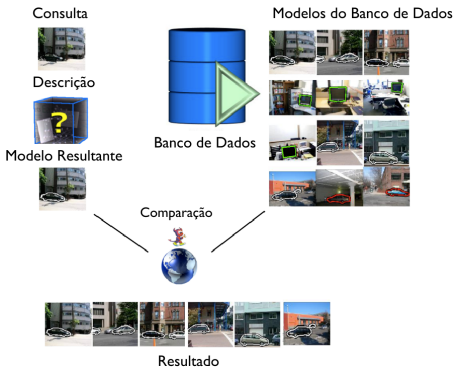
- Questões importantes na definição de um problema em algoritmos genéticos
 - ▶ representação dos indivíduos
 - ▶ parâmetros do sistema (tamanho da população, taxa de mutação, ...)
 - ▶ políticas de seleção e eliminação de indivíduos
 - ▶ operadores genéticos (cruzamento e mutação)
 - ▶ critérios de parada
 - ▶ função de avaliação (a mais importante e mais complicada de ser definida)

Observações

- Se o AG estiver corretamente implementado, a população evolui em gerações sucessivas
- Aptidão do melhor indivíduo e do indivíduo médio aumentam em direção a um ótimo global

Exemplos

- Problema: CBIR
- Descrição da imagem:
 - ▶ Cor: (1, 20, 50, 50, 30, 25)
 - ▶ Forma: (0, 1, 1, 0, 1, 1)
 - ▶ Textura: (234, 50, 45, 11, 13, 14)

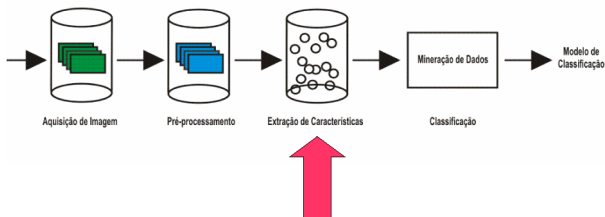


Combinação

Programação Genética

Exemplos

- Problema: Classificação de Imagens



Algoritmos Genéticos



Referências e Leituras Complementares

- Cap. 04 → livro Russel e Norvig
- Cap. 14 → livro Ben Coppin
- Algoritmos Genéticos → livro Ricardo Linden