

Inteligência Artificial

Aula 08- Solução de Problemas (Busca com Informação - Algoritmos por Refinamentos Sucessivos)¹

Sílvia M.W. Moraes

Faculdade de Informática - PUCRS

March 29, 2017

¹Este material não pode ser reproduzido ou utilizado de forma parcial sem a permissão dos autores.

Sinopse

- Nesta aula, apresentamos uma **introdução a solução de problemas por algoritmos de busca local por refinamentos sucessivos**.
- Este material foi construído com base nos capítulos:
 - 4 do livro Artificial Intelligence – a Modern Approach de Russel & Norvig
 - 4 do livro Inteligência Artificial de Luger.

Sumário

- 1 O que vimos ...
- 2 Algoritmos de Busca Local por Refinamentos Sucessivos

Aulas anteriores

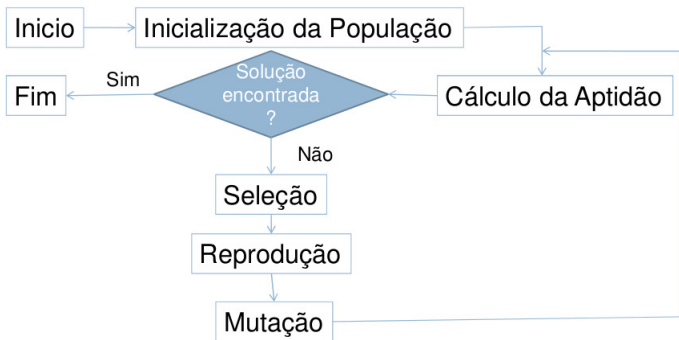
- Agente Reativos e Cognitivos
- Solução de Problemas
 - Representação, Espaço de Estados, Plano: sequência de ações
 - Busca sem informação
 - Busca com informação
 - A*
 - Algoritmos de busca local por refinamentos sucessivos:
Hill Climbing e Simulated Annealing

Algoritmo Genético: Introdução

- Como já vimos, algoritmos de busca por refinamento sucessivo partem de soluções propostas e tentam melhorá-las.
- **Algoritmos Genéticos:**
 - Métodos de otimização e busca inspirados nos mecanismos de evolução de populações de seres vivos.
 - Foram introduzidos por John Holland (1975) e popularizados por um dos seus alunos, David Goldberg (1989).
 - Seguem o princípio da seleção natural e sobrevivência (Charles Darwin - livro “A Origem das Espécies”, 1859):
 - **“Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes” .**

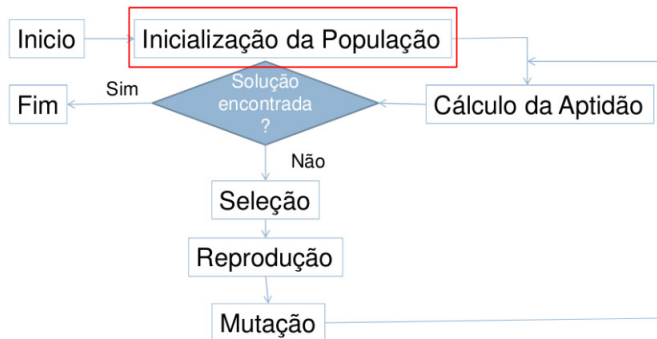
Algoritmo Genético: Ciclo

- Ciclo de execução:



Algoritmo Genético: Inicialização da população

- Ciclo de execução:

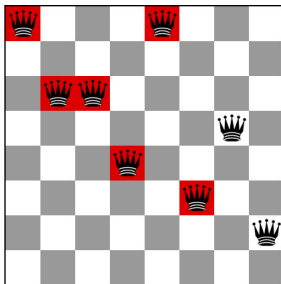


Algoritmo Genético: Inicialização da população

- **População Inicial:** conjunto de soluções geradas aleatoriamente. Corresponde à população da geração 0 (ponto de partida do algoritmo).
 - cada elemento da população é chamado de indivíduo ou cromossomo.
 - um cromossomo é formado por genes – seqüências de DNA - que servem para determinar as características da solução.
 - os cromossomos podem ser codificados em binário (mais tradicional), inteiro, real, alfabeto específico, ...
 - tamanho da população (número de soluções) é um parâmetro do algoritmo.

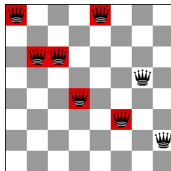
Algoritmo Genético: Inicialização da população

- **Exemplo:** Retomando o **problema das 8 rainhas** ...
 - **Objetivo:** Dispor as oito rainhas em um tabuleiro sem que uma ataque a outra.
 - Uma rainha ataca outra se esta estiver na mesma linha, coluna ou diagonal.



Algoritmo Genético: Inicialização da população

- **Passo 1:** Escolher a **codificação** mais indicada para o problema



- Binário: necessário 3 bits por rainha

coluna	→	0	1	2	3	4	5	6	7
linha	→	000	010	010	100	000	101	011	110

- Inteiro: valores de 0 a 7

coluna	→	0	1	2	3	4	5	6	7
linha	→	0	2	2	4	0	5	3	6

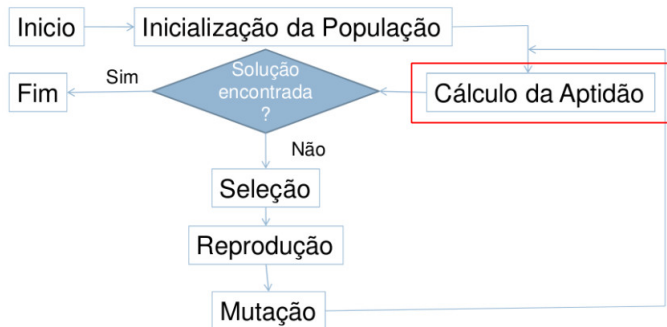
Algoritmo Genético: Inicialização da população

- **Passo 2:** Definir o tamanho da população (número de indivíduos) e gerá-la aleatoriamente.
 - recomenda-se que a população inicial contenha soluções distintas (uma diferente da outra).
 - Exemplo: **tamanho = 5**

	Cromossomos							
0	0	2	2	4	0	5	3	6
1	1	2	3	5	0	6	1	2
2	0	3	3	5	7	4	3	5
3	1	2	3	4	0	0	1	2
4	7	3	4	4	5	5	6	2

Algoritmo Genético: Cálculo da Aptidão

- Ciclo de execução:



Algoritmo Genético: Cálculo da Aptidão

- **Passo 3:** Logo após a geração de uma população, verifica-se a aptidão dos cromossomos.
 - Aptidão: é determinada por uma função que diz o quanto os indivíduos são aptos. Ela mede a qualidade da solução.
 - Vamos usar a mesma função heurística h , usada nos exemplos anteriores, para calcular a aptidão
 - h : corresponde ao número de pares de rainhas que estão atacando umas às outras, seja direta ou indiretamente.
 - objetivo é $h = 0$, logo, quanto menor o valor de h melhor (indivíduo mais apto).

Algoritmo Genético: Cálculo da Aptidão

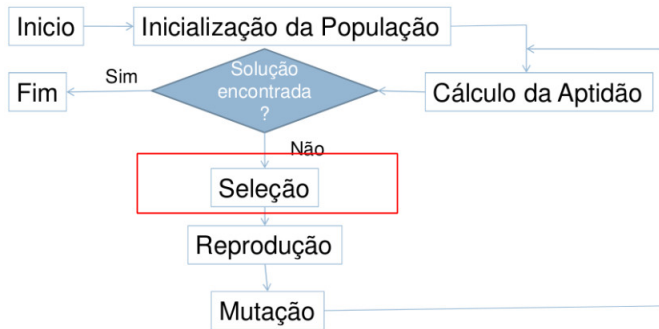
- Cada um dos indivíduos da população é avaliado pela função de aptidão (h).

	Cromossomos								h
0	0	2	2	4	0	5	3	6	7
1	1	2	3	5	0	6	1	2	9
2	0	3	3	5	7	4	3	5	6
3	1	2	3	4	0	0	1	2	14
4	7	3	4	4	5	5	6	2	7

- Esse é o momento de analisarmos se o objetivo já foi alcançado.
 - Quando isso acontece, o algoritmo pára e apresenta o cromossomo que satisfaz o objetivo como solução final.
 - Caso contrário, inicia a seleção ...

Algoritmo Genético: Seleção

- Ciclo de execução:



Algoritmo Genético: Seleção

- A partir desse ponto, inicia o processo de construção da **população intermediária**, ou seja, a população da próxima geração.
 - A seleção é o operador genético que escolhe indivíduos da população atual que serão usados para gerar a próxima população.
 - É projetada para escolher preferencialmente indivíduos de maior aptidão, embora não exclusivamente, a fim de manter a diversidade da população.
- Há vários tipos de **seleção**. Eis alguns:
 - Seleção por **elitismo**: consiste em passar o melhor indivíduo da população atual diretamente para a próxima.
 - Seleção por **torneio**: é usada para selecionar os indivíduos da população atual com fins de cruzamento.

Algoritmo Genético: Seleção por Elitismo

- **Passo 4:** Aplicação da seleção por elitismo.
 - É recomendada pois melhora o desempenho do algoritmo por preservar a melhor solução encontrada até o momento.

População Atual

	Cromossomos								h
0	0	2	2	4	0	5	3	6	7
1	1	2	3	5	0	6	1	2	9
2	0	3	3	5	7	4	3	5	6
3	1	2	3	4	0	0	1	2	14
4	7	3	4	4	5	5	6	2	7

População Intermediária

	Cromossomos							
	0	3	3	5	7	4	3	5

Algoritmo Genético: Seleção por Torneio

- **Passo 5:** Aplicação da seleção por torneio para selecionar os indivíduos que serão cruzados.
 - A **seleção por torneio** binária é a mais usada e consiste em
 - escolher **randomicamente dois indivíduos** da população atual e escolher o **melhor para ser o pai**.
 - escolher **randomicamente** novamente **dois indivíduos** da população atual e escolher o **melhor para ser a mãe**.

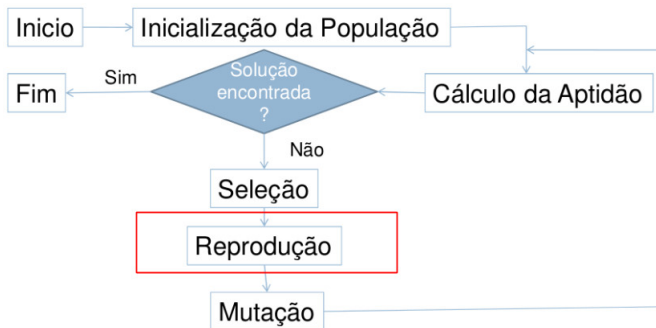
Algoritmo Genético: Seleção por Torneio

População Atual										
		Cromossomos							h	
	0	0	2	2	4	0	5	3	6	7
	1	1	2	3	5	0	6	1	2	9
mãe	2	0	3	3	5	7	4	3	5	6
	3	1	2	3	4	0	0	1	2	14
pai	4	7	3	4	4	5	5	6	2	7

- Seleção por torneio:
 - gera randomicamente: 1 e 4, escolhe **4 para ser o pai**
 - gera randomicamente: 3 e 2, escolhe o **2 para ser a mãe**
- O passo seguinte é o cruzamento ...

Algoritmo Genético: Crossover

- Ciclo de execução:



Algoritmo Genético: Crossover

- **Passo 6:** Cruzamento dos indivíduos selecionados.
 - O operador genético **crossover** é responsável por
 - cruzar a carga genética dos indivíduos escolhidos pela seleção no passo anterior
 - gerar os indivíduos da próxima geração.
 - Há vários tipos de cruzamento:
 - **n -ponto:** consiste em cortar os cromossomos em n pontos. Quando $n = 1$, é chamado de **uniponto**. Para diferentes valores de n é chamado de multiponto.
 - **uniforme:** aplica uma máscara binária que determina como os genes serão misturados para formar os filhos.

Algoritmo Genético: Crossover uniponto

- No cruzamento uniponto:
 - escolhe-se o ponto de corte aleatoriamente.
 - para formar o filho1, unimos o início da mãe com o final do pai.
 - para formar o filho2, unimos o início do pai com o final da mãe.

mãe	2	0	3	3	5	7	4	3	5
pai	4	7	3	4	4	5	5	6	2

Filho 1	0	3	3	4	5	5	6	2
Filho 2	7	3	4	5	7	4	3	5

Algoritmo Genético: Crossover uniponto

- Os indivíduos gerados (filhos) são colocados na população intermediária.
- O processo de seleção e cruzamento (passos 4 e 5) devem ser repetidos até completar a população.
 - Observação: toda a população tem sempre a mesma quantidade de elementos.

População Intermediária

Cromossomos							
0	3	3	5	7	4	3	5
0	3	3	4	5	5	6	2
7	3	4	5	7	4	3	5

Algoritmo Genético: Crossover multiponto

- Exemplo de cruzamento n -ponto, para $n = 2$.
 - Os pontos de corte são escolhidos aleatoriamente.
 - É mais interessante para cromossomos longos.

mãe	2	0	3	3	5	7	4	3	5
pai	4	7	3	4	4	5	5	6	2

Filho 1	0	3	4	4	5	5	3	5
Filho 2	7	3	3	5	7	4	6	2

Algoritmo Genético: Crossover uniforme

- No cruzamento **uniforme**, é necessário inicialmente gerar uma máscara binária.
 - A máscara determina como os genes deve ser copiados para formar os filhos.
 - Filho1: contém os genes do pai cuja posição na máscara é 1 e os genes da mãe cuja posição na máscara é 0.
 - Filho2: o oposto.

máscara								
1	0	0	1	0	1	1	0	

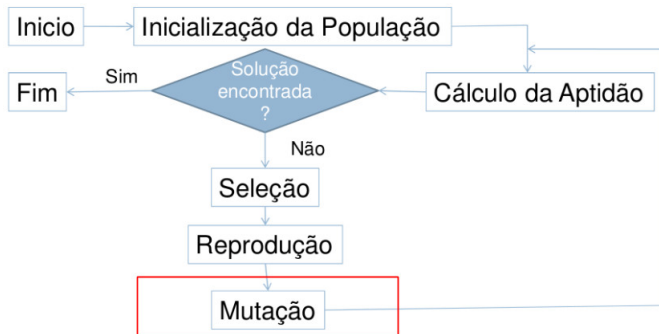
2	0	3	3	5	7	4	3	5	mãe
4	7	3	4	4	5	5	6	2	

Filho	1	7	3	3	4	7	5	6	5	
Filho	2	0	3	4	5	5	4	3	2	

- O próximo passo é a mutação ...

Algoritmo Genético: mutação

- Ciclo de execução:



Algoritmo Genético: mutação

- **Passo 7:** Aplicação da operação de **mutação**
 - O operador genético mutação consiste em uma mudança genética em cromossomos da população.
 - A taxa de mutação deve ser baixa.
 - Escolhe randomicamente um dos filhos recém gerados pelo cruzamento e um dos seus genes para mutar.
 - Substitui o gene escolhido por outro, também gerado aleatoriamente.

Algoritmo Genético: mutação

- Exemplo:

- Indivíduo escolhido aleatoriamente: 2
- Gene escolhido aleatoriamente: 4 (posição)

Antes

População Intermediária

	Cromossomos							
0	0	3	3	5	7	4	3	5
1	0	3	3	4	5	5	6	2
2	7	3	4	5	7	4	3	5
3								
4								

Depois

População Intermediária

	Cromossomos							
0	0	3	3	5	7	4	3	5
1	0	3	3	4	5	5	6	2
2	7	3	4	5	0	4	3	5
3								
4								

Algoritmo Genético: Critérios de Parada

- Uma vez preenchida a população intermediária,
 - ela se transforma na população atual (geração+1) e
 - o ciclo reinicia no passo 3.
- **Critérios de parada**
 - solução ótima
 - número de gerações
 - convergência: 98 a 99% dos indivíduos representam a mesma solução.

Algoritmo Genético: Terminologia Básica

- **genótipo** ou **cromossomo** ou **indivíduo** :
 - um elemento da população (uma solução possível)
 - estrutura de dados que contém **uma solução codificada** para o problema. **Ex:**

0	1	0	0
---	---	---	---
- **fenótipo** - cromossomo decodificado. **Ex:** 4
- **gene** - representação de cada parâmetro conforme o alfabeto utilizado (binário, inteiro ou real) **Ex:** célula
- **alelos** - valores de um gene. **Ex:** {0,1}

Algoritmo Genético: Terminologia Básica

- **população** - conjunto de pontos (indivíduos/soluções) no Espaço de Busca
- **geração** - iteração (ciclo) responsável por gerar uma nova população
- **aptidão bruta** - valor da função de aptidão para um indivíduo da população
- **aptidão máxima** - melhor indivíduo da população corrente
- **aptidão média** - aptidão média da população corrente

Algoritmo Genético: Dicas

- Contribuem para resultados melhores:
 - Incluir na população inicial uma solução obtida por outro método (o AG tende a melhorar tal solução);
 - Garantir diversidade na geração da população inicial.
 - Iniciar com uma população inicial relativamente grande e reduzi-la com o tempo.
 - Usar de elitismo

Algoritmo Genético: exercícios

- **Atividade I:** Um projeto possui **10 tarefas** que precisam ser distribuídas entre 2 pessoas da equipe. Para cada tarefa foi **estimado o número de horas** necessário para concluí-la.
 - ① **Como poderíamos realizar a distribuição de tarefas entre as duas pessoas de forma harmônica ?**
 - Para responder à questão, considere as tarefas e sua carga (em horas);

t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
5	10	15	3	10	5	2	16	9	7
 - ② Como poderíamos codificar as soluções ?
 - ③ Defina uma função objetivo (de avaliação/aptidão) adequada para o problema.
 - ④ Se a distribuição de tarefas fosse para n pessoas, a função definida no item 2 poderia ser generalizada para n ? (Sim. Como? / Não. Defina uma nova função.)

Algoritmo Genético: exercícios

- **Atividade II:** Desenvolver um AG para descobrir uma frase informada pelo usuário.
 - Representação do Cromossomo em ASCII.
 - Exemplo “Familia” = (102,97,109,105,108,105,97)
 - Espaço de busca: Considerando que existem 26 letras maiúsculas/minúsculas + espaço em branco, uma frase de comprimento N terá um espaço de busca de $(26 + 26 + 1) \times N$, ou seja, $53 \times N$.
 - Teste as funções objetivo:

$$f_1 = \sum_{i=1}^{|fraseUsuario|} (fraseUsuario[i] - cromossomo[i])^2$$

$$f_2 = \sum_{i=1}^{|fraseUsuario|} 1 - sinal(fraseUsuario[i] - cromossomo[i]),$$

$$\text{onde } sinal(x) = \begin{cases} 1 & \text{se } x = 0 \\ 0 & \text{se } x \neq 0 \end{cases}$$