Aula 6

Inteligência Artificial Aplicada

Prof. Dr. Luciano Frontino de Medeiros

Temas

- Algoritmos Genéticos
- Etapas do AG
- Operadores de AG
- Exemplo de Aplicação de AG
- AG em Linguagem Java

Conversa Inicial

- A teoria da evolução darwiniana aplicada à computação
- Evolução de soluções parciais ao longo de várias iterações
- Algoritmos Genéticos

Algoritmos Genéticos

- Um algoritmo genético (AG) faz parte da classe de algoritmos de busca
- O algoritmo procura uma solução dentro de um espaço para um problema de otimização
- Os AG podem ser uma boa opção para efetuar a busca em problemas considerados intratáveis

Exemplos de Aplicações

- Arquitetura de circuitos eletrônicos
- Planejamento e roteirização
- Programação de games
- Previsão do tempo
- Descoberta de identidades matemáticas

Algoritmo de Busca

- Algoritmo de busca em feixe estocástico
- Os estados sucessores são criados a partir da combinação de dois (ou mais) estados "pais", ao invés de serem criados a partir da variação de um único estado

Fatores de Sucesso do AG

- Simplicidade da operação
- Facilidade de implementação
- Eficácia na busca da região onde provavelmente está o máximo global
- Aplicável nas situação onde se tem pouco ou nenhum conhecimento do modelo, ou este é impreciso

Função Objetivo

- A produção de uma nova população é avaliada por uma função objetivo, ou função de *fitness*
- Esta função retorna à nova população priorizando os estados melhores

Diferença das Técnicas Convencionais

- Busca em feixe
- Busca cega (sem conhecimento)
- Usam operadores estocásticos ou probabilísticos e não regras determinísticas

Etapas do Algoritmo Genético

Etapas do AG (1)

- 1. Cria uma população aleatória de candidatos à solução (pop)
- 2. Enquanto as condições de terminação não são satisfeitas: (cada iteração ou geração)

- (a) Cria uma nova população vazia (*new-pop*)
- (b) Enquanto a *new-pop* não estiver completa:
  - i. Faz a seleção de dois indivíduos aleatoriamente de *pop* (dando preferência na seleção aos indivíduos de maior *fitness*)

Etapas do AG (2)

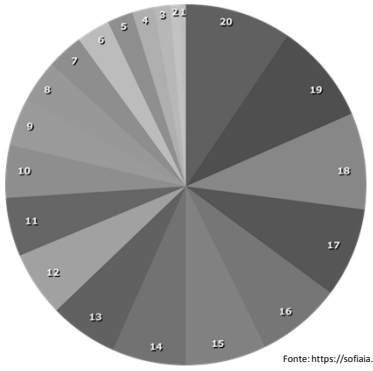
  - ii. Faz o *crossover* de dois indivíduos para obter dois novos indivíduos
- (c) Dá a cada membro da *new-pop* a chance de mutação
- (d) Substitui *pop* por *new-pop*

- 3. Seleciona o indivíduo da população com o melhor *fitness* como a solução do problema

Operadores de Algoritmos Genéticos

Seleção

- Análogo à sobrevivência dos mais adaptados no mundo natural

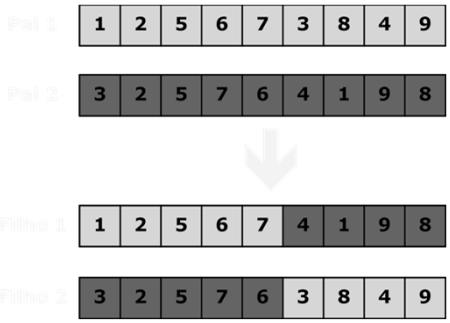


Fonte: https://sofiaia.wordpress.com/category/c/

- **Indivíduos mais aptos (melhor *fitness*) são selecionados para "procriação"**
- **Utiliza um critério de maior probabilidade de cruzamento para os que têm maior *fitness***

Crossover

- **O *crossover* (cruzamento) ocorre pela mistura de duas soluções (indivíduos) com o objetivo de criar dois novos indivíduos**

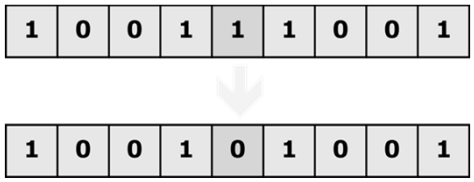


Fonte: Autor.

- **Este cruzamento tende a formar indivíduos que possuem características dos "pais", e que têm a possibilidade de atender melhor o *fitness***

Mutação

- **Durante cada geração, existe uma pequena chance de que um individuo dentro da população sofra uma mutação, que vai mudar a característica do indivíduo levemente**
- **A mutação acontece de forma aleatória no algoritmo**



Fonte: Autor

Tamanho da População

- Quanto maior é a população, maior a quantidade de soluções possíveis, o que significa maior variação da população

- A população deve ser a maior possível
- O limite é o tempo que o algoritmo vai demorar na execução, e a memória para guardar a população

Formas de Término do AG

- A abordagem mais simples é rodar o algoritmo de busca por um número fixo de gerações
- Encerrar o algoritmo se, depois de passadas algumas gerações, não se obtém uma melhora significativa na adaptação do melhor indivíduo da população

Exemplo de Aplicação de AG

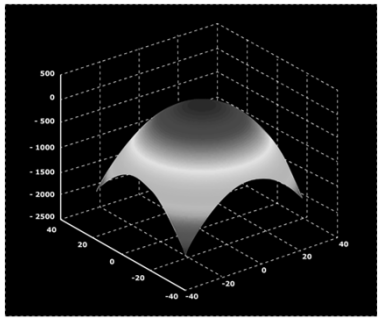
Encontrar o Máximo de uma Equação

$f(x) = 2 - (x-3)^2 - (y-2)^2$

$x = 3$

$y - 2$

- A seguinte equação tem o seu valor máximo, f(x) = 2, no ponto (3,2)



Fonte: Autor.

Definição do Cromossomo

- **Faixa de valores: [0,7]**

Decimal	Binário
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

- **Exemplo:**
  - **x = 1 (001b)**
  - **y = 5 (101b)**

  001101

Fonte: Autor.

Geração Aleatória da População

x	y	Cromossomo
3	6	011110
6	4	110100
3	5	011101
7	0	111000
3	7	011111
1	6	001110
1	2	001010
5	4	101100
6	1	110001
4	2	100010

Fonte: Autor.

Função Objetivo (*Fitness*)

x	y	Cromossomo	f(x,y)
3	6	011110	-8
6	4	110100	-15
3	5	011101	-3
7	0	111000	-32
3	7	011111	-15
1	6	001110	-8
1	2	001010	0
5	4	101100	-8
6	1	110001	-18
4	2	100010	-3

Fonte: Autor.

Ordenação

x	y	Cromossomo	f(x,y)
1	2	001010	0
3	5	011101	-3
4	2	100010	-3
3	6	011110	-8
1	6	001110	-8
5	4	101100	-8
6	4	110100	-15
3	7	011111	-15
6	1	110001	-18
7	0	111000	-32

Fonte: Autor.

Seleção

x	y	Cromossomo	f(x,y)
1	2	001010	0
3	5	011101	-3
4	2	100010	-3
3	6	011110	-8
1	6	001110	-8
5	4	101100	-8
6	4	110100	-15
3	7	011111	-15
6	1	110001	-18
7	0	111000	-32

Fonte: Autor.

Crossover

x	y	Cromossomo	f(x,y)	Ponderação
1	2	001010	0	4
3	5	011101	-3	3
4	2	100010	-3	2
3	6	011110	-8	1

x	y	Cromossomo	f(x,y)
		001001	
		001001	
		001010	
		001010	
		011110	
		011110	
		011001	
		100010	
		100010	
		011110	

Fonte: Autor.

Mutação

x	y	Cromossomo	f(x,y)
		001011	
		001001	
		011010	
		001010	
		011110	
		011111	
		011101	
		101010	
		100010	
		011010	

Fonte: Autor.

Decodificação

x	y	Cromossomo	f(x,y)
1	3	001011	
1	1	001001	
3	2	011010	
1	2	001010	
3	6	011110	
3	7	011111	
3	5	011101	
5	2	101010	
4	2	100010	
3	2	011010	

Fonte: Autor.

Função Objetivo (*Fitness*)

x	y	Cromossomo	f(x,y)
1	3	001011	1
1	1	001001	-3
3	2	011010	0
1	2	001010	0
3	6	011110	-8
3	7	011111	-15
3	5	011101	-3
5	2	101010	-8
4	2	100010	-3
3	2	011010	0

Fonte: Autor.

AG em Linguagem Java

JAGA

- ***Java API for Genetic Algorithms***
- **Consiste em uma série de classes para o uso em diversos tipos de problemas**
- **Para exemplos que lidem com aspectos comuns dos AG, podemos criar uma classe de exemplo com o problema em questão**

Exemplo (1)

```
public class Example1 {

	public Example1() {
	}

	public void exec() {
		// Define os parâmetros para o AG
		GAParameterSet params = new DefaultParameterSet();
		params.setPopulationSize(40);
		params.setFitnessEvaluationAlgorithm(new
Example1Fitness());

		// Inclui a reprodução da nova população com crossover e
mutação
		CombinedReproductionAlgorithm repAlg = new
CombinedReproductionAlgorithm();
		repAlg.insertReproductionAlgorithm(0, new
SimpleBinaryXOver(0.9));
		repAlg.insertReproductionAlgorithm(1, new
SimpleBinaryMutation(0.02));
		params.setReproductionAlgorithm(repAlg);
```

Fonte: Autor.

Exemplo (2)

```
		// Define o método da roleta viciada
		params.setSelectionAlgorithm(new RouletteWheelSelection(-
10E10));
		// Número máximo de gerações
		params.setMaxGenerationNumber(10000);
		// Define as variáveis por indivíduo, a precisão decimal e
o tamanho do cromossomo
		NDecimalsIndividualSimpleFactory fact = new
NDecimalsIndividualSimpleFactory(2, 2, 15);
		fact.setConstraint(0, new RangeConstraint(-6, 6));
		fact.setConstraint(1, new RangeConstraint(-6, 6));
		params.setIndividualsFactory(fact);

	// Constrói o AG
	ReusableSimpleGA ga = new ReusableSimpleGA(params);
	// Associa o analisador
	AnalysisHook hook = new AnalysisHook(System.out, false);
	ga.addHook(hook);
```

Fonte: Autor.

Função Objetivo

```
public class Example1Fitness implements FitnessEvaluationAlgorithm {

	public Example1Fitness() {}

	public Class getApplicableClass() {
		return NDecimalsIndividual.class;
	}

	public Fitness evaluateFitness(Individual individual, int age,
Population population, GAParameterSet params) {
		NDecimalsIndividual indiv = (NDecimalsIndividual)
individual;
		double x = indiv.getDoubleValue(0);
		double y = indiv.getDoubleValue(1);
		double f = 2 - (x-3)*(x-3) - (y-2)*(y-2);
		Fitness fit = new AbsoluteFitness(f);
		return fit;
	}
}
```

Fonte: Autor.

Diferentes Execuções

Nº	x	y	f(x,y)	Geração
1	3.00	1.99	1.9999	2475
2	2.99	2.01	1.9998	9816
3	3.14	2.06	1.9768	4019
4	3.00	1.98	1.9996	6431
5	3.01	2.00	1.9999	306
6	3.00	1.99	1.9999	1438
7	2.97	2.04	1.9975	1259
8	3.00	2.01	1.9999	2102
9	2.99	2.00	1.9999	9165
10	2.07	2.01	1.9990	2341

Fonte: Autor.