

Aula Prática 4

Inteligência Artificial Aplicada

Prof. Dr. Luciano Frontino de Medeiros

Temas

- **Problemas de otimização**
- **Funções objetivo de 5 e 10 variáveis.**
- **AG com permutação de 8 nós em um grafo.**

Solucionando Equações com AG

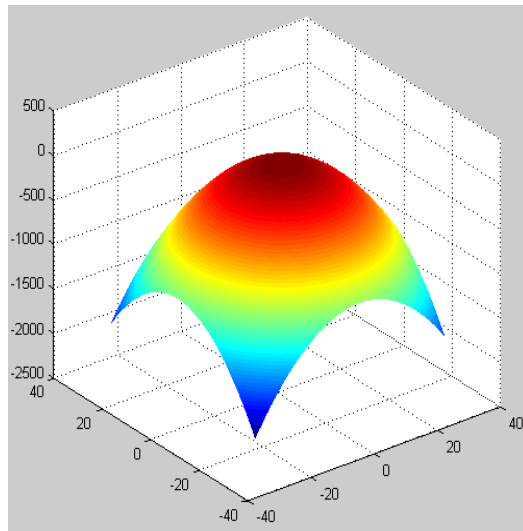
- **Certos problemas matemáticos que possuem métodos determinísticos ou analíticos de resolução *a priori*, são bons para comparar o desempenho de um AG na busca por soluções ótimas.**

Exemplo: Otimização com 2 Variáveis

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

- **Por métodos analíticos (ou mesmo por visualização), esta equação tem um máximo no ponto $x=3$ e $y=2$.**

Gráfico



Classe Example1.java

```
public class Example1 {

    public static int NVAR = 2;
    public static int NEXEC = 1000;
    static Utility utility;
    static IFunctionFitness ff;

    public static void main(String[] args) throws FileNotFoundException {
        utility = new Utility();

        PrintStream ps = new PrintStream(new FileOutputStream("Example1.txt", true));
        utility.printlnHeader(System.out, NVAR);
        ff = new FunctionFitness1();

        for(int j=0; j<NEXEC; j++) {
            GenAlg g = new GenAlg(60, 15, 2, 100000, 0.01, 0.6, 0.8, 1.9999, 40, false, ff);
            utility.println(System.out, j, g.fGen, g.maxFitness, g.indGlobalFitMax.xNorm);
            utility.println(ps, j, g.fGen, g.maxFitness, g.indGlobalFitMax.xNorm);
        }
        ps.close();
    }
}
```

Função Objetivo (1)

```
public class FunctionFitness1 implements IFunctionFitness {
    private double bias = 0.0; // Para ajustar valores para o fitness
    private double f;

    public FunctionFitness1() {
    }

    public double process(double[] x, boolean normalized) {
        if(normalized) {
            return(function(x) + bias);
        } else {
            return(function(x));
        }
    }
}
```

Função Objetivo (2)

```
public void setMinMax(double[] xMin, double[] xMax) {
    double fmin, fmax;
    for(int i=0; i < xMin.length; i++) {
        xMin[i] = 0;
        xMax[i] = 6;
    }
    fmin = Math.abs(function(xMin));
    fmax = Math.abs(function(xMax));
    bias = fmin > fmax ? fmin : fmax;
}

public double function(double[] x) {
    f = 2 - Math.pow(x[0] - 3, 2) - Math.pow(x[1] - 2, 2);
    return(f);
}
```

Execução

#	Data	Hora	Gen	Fitness	x[0]	x[1]
0	8/03/2017	18:14:19	70	2,0000	3,00	1,99
1	8/03/2017	18:14:19	117	2,0000	3,00	1,99
2	8/03/2017	18:14:19	28	1,9999	3,00	1,99
3	8/03/2017	18:14:19	254	2,0000	3,00	2,00
4	8/03/2017	18:14:19	281	2,0000	3,00	2,00
5	8/03/2017	18:14:19	328	1,9999	3,00	1,99
6	8/03/2017	18:14:19	17	2,0000	3,01	2,00
7	8/03/2017	18:14:19	75	1,9999	3,00	1,99
8	8/03/2017	18:14:19	168	1,9999	3,00	1,99
9	8/03/2017	18:14:19	40	2,0000	3,00	2,00
10	8/03/2017	18:14:19	9	1,9999	3,00	1,99
11	8/03/2017	18:14:19	657	2,0000	3,00	2,00
12	8/03/2017	18:14:19	428	2,0000	3,00	2,00
13	8/03/2017	18:14:20	132	2,0000	2,99	2,00
14	8/03/2017	18:14:20	34	1,9999	3,00	1,99
15	8/03/2017	18:14:20	52	1,9999	3,00	2,01

Exemplo: Otimização com 5 Variáveis

$$f(x) = 55 - (x_1 - 1)^2 - (x_2 - 3)^2 - (x_3 - 4)^2 - (x_4 - 5)^2 - (x_5 - 2)^2$$

- **Por métodos analíticos (ou por visualização), esta equação tem um máximo no ponto (1,3,4,5,2)**

Classe Example2.java

```
public class Example2 {

    public static int NVAR = 5;
    public static int NEXEC = 1000;
    static Utility utility;
    static IFunctionFitness ff;

    public static void main(String[] args) throws FileNotFoundException {
        utility = new Utility();

        PrintStream ps = new PrintStream(new FileOutputStream("Example2.txt", true));
        utility.printHeader(System.out, NVAR);
        //utility.printHeader(ps, NVAR);
        ff = new FunctionFitness2();

        for(int j=0; j<NEXEC; j++) {
            GenAlg g = new GenAlg(60, 15, 5, 100000, 0.02, 0.6, 1.0, 54.9999, 42, false, ff);
            utility.println(System.out, j, g.fGen, g.maxFitness, g.indGlobalFitMax.xlnorm);
            utility.println(ps, j, g.fGen, g.maxFitness, g.indGlobalFitMax.xlnorm);
        }
        ps.close();
    }

}
```

Execução

#	Data	Hora	Gen	Fitness	x[0]	x[1]	x[2]	x[3]	x[4]
0	8/03/2017	18:25:08	58666	55,0000	1,00	3,00	4,00	5,00	2,00
1	8/03/2017	18:25:09	10756	54,9999	1,00	3,01	4,00	5,00	2,01
2	8/03/2017	18:25:10	18386	55,0000	1,00	3,00	4,00	5,00	2,01
3	8/03/2017	18:25:11	22108	55,0000	1,00	3,00	4,00	5,00	2,00
4	8/03/2017	18:25:12	11179	55,0000	1,00	3,00	4,00	5,00	2,00
5	8/03/2017	18:25:12	5227	54,9999	1,00	3,00	4,00	4,99	2,00
6	8/03/2017	18:25:12	1848	54,9999	1,00	3,00	4,00	5,00	2,01
7	8/03/2017	18:25:14	27414	54,9999	1,00	2,99	4,00	5,00	2,00
8	8/03/2017	18:25:15	7016	54,9999	1,00	3,00	4,00	5,00	1,99
9	8/03/2017	18:25:15	8796	54,9999	1,00	3,00	3,99	5,00	2,00
10	8/03/2017	18:25:16	11107	54,9999	1,00	2,99	4,00	4,99	2,00
11	8/03/2017	18:25:17	10011	54,9999	1,00	3,00	4,00	5,00	2,01
12	8/03/2017	18:25:17	5060	54,9999	0,99	3,00	4,00	5,00	2,00
13	8/03/2017	18:25:18	9299	54,9999	1,00	3,00	4,00	5,01	2,00
14	8/03/2017	18:25:18	1122	54,9999	1,00	3,00	4,00	4,99	2,00
15	8/03/2017	18:25:19	15434	55,0000	1,00	3,00	4,01	5,00	2,00

Exemplo: Otimização com 10 Variáveis

$$f(x) = \frac{611}{4} - (x_1 - 1)^2 - (x_2 - 3)^2 - (x_3 - 4)^2 - (x_4 - 5)^2 - (x_5 - 2)^2 - (x_6 - \frac{1}{2})^2 - (x_7 - \frac{3}{2})^2 - (x_8 - 2)^2 - (x_9 - 1)^2 - (x_{10} - \frac{7}{2})^2$$

- Esta equação tem um máximo no ponto **(1,3,4,5,2,0.5,1.5,2,1,3.5)**

Classe Example3.java

```
public class Example3 {

    public static int NVAR = 10;
    public static int NEXEC = 1000;
    static Utility utility;
    static IFunctionFitness ff;

    public static void main(String[] args) throws FileNotFoundException {
        utility = new Utility();

        PrintStream ps = new PrintStream(new FileOutputStream("Example3.txt", true));
        utility.printHeader(System.out, NVAR);
        //utility.printHeader(ps, NVAR);
        ff = new FunctionFitness3();

        for(int j=0; j<NEXEC; j++) {
            GenAlg g = new GenAlg(100, 15, 10, 200000, 0.005, 0.6, 1.0, 152.749, 50, false, ff);
            utility.println(System.out, j, g.fGen, g.maxFitness, g.indGlobalFitMax.xNorm);
            utility.println(ps, j, g.fGen, g.maxFitness, g.indGlobalFitMax.xNorm);
        }
        ps.close();
    }
}
```

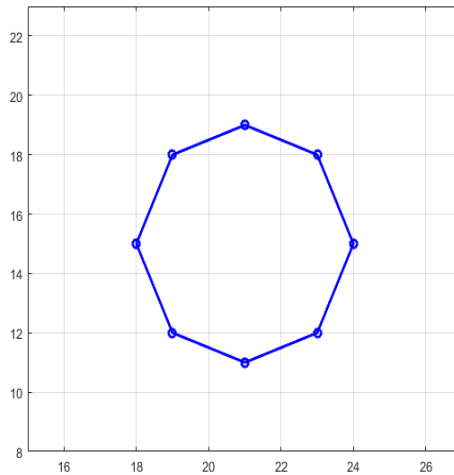
Execução

#	Data	Hora	Gen	Fitness	x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]	x[7]	x[8]	x[9]
0	8/03/2017	18:40:53	2410	152,7493	0,98	3,00	4,01	5,00	2,00	0,52	1,50	1,99	1,00	3,50
1	8/03/2017	18:40:53	1881	152,7491	1,00	3,00	4,00	5,01	2,01	0,50	1,49	1,99	0,98	3,49
2	8/03/2017	18:40:56	8290	152,7491	1,00	3,00	3,98	5,00	2,01	0,50	1,49	2,01	0,98	3,50
3	8/03/2017	18:40:58	7119	152,7493	0,98	3,00	4,01	5,02	2,01	0,50	1,50	2,00	1,00	3,49
4	8/03/2017	18:41:03	14311	152,7490	1,00	3,00	4,02	5,00	1,99	0,50	1,50	2,00	1,00	3,52
5	8/03/2017	18:41:04	2602	152,7494	1,00	3,00	4,01	5,02	2,00	0,51	1,50	2,02	1,00	3,50
6	8/03/2017	18:41:05	3634	152,7491	1,01	3,00	3,98	5,00	2,02	0,50	1,50	2,02	1,00	3,50
7	8/03/2017	18:41:06	2996	152,7490	1,00	3,00	4,00	5,00	2,00	0,52	1,49	2,02	1,00	3,52
8	8/03/2017	18:41:07	3496	152,7497	1,00	3,00	4,00	5,00	2,00	0,50	1,50	2,00	1,00	3,52
9	8/03/2017	18:41:08	2675	152,7491	1,00	3,00	4,00	5,00	2,00	0,52	1,50	1,98	1,00	3,52
10	8/03/2017	18:41:09	2596	152,7491	1,00	3,00	4,00	4,99	2,00	0,49	1,50	1,98	0,98	3,49
11	8/03/2017	18:41:10	2076	152,7491	1,01	3,00	4,00	5,00	2,00	0,48	1,51	2,00	0,99	3,50
12	8/03/2017	18:41:11	1733	152,7497	1,01	3,00	4,00	5,00	2,00	0,50	1,50	2,00	0,98	3,50
13	8/03/2017	18:41:12	4154	152,7493	1,00	3,00	4,00	4,99	2,00	0,48	1,50	2,02	1,00	3,50
14	8/03/2017	18:41:13	3185	152,7493	1,01	3,00	4,01	5,00	2,00	0,52	1,50	2,02	1,01	3,50
15	8/03/2017	18:41:13	888	152,7491	1,00	3,01	3,98	5,02	1,99	0,50	1,50	1,99	1,00	3,50

AG com Permutação

- **AG para permutação**
são utilizados para
minimizar a distância
entre nós em um
determinado grafo.
- **Dessa forma, podemos**
utilizar ao **AG** para
resolver o problema do
viajante, por exemplo.

Exemplo: Permutação com 8 nós



Distância Total

- Calculando a distância euclidiana entre dois pontos, sabemos *a priori* o resultado de para a distância total mínima entre os nós é de 21,59338.

Classe ExOrder3.java

```

public class ExOrder3 {
    public static int TAMANHO = 8;
    public static int TAMPOP = 30;
    public static int NGEN = 3000;
    public static int NEXEC = 1000;

    public static GenAlgoOrder g;
    public static Utility utility;

    public static void main(String[] args) throws FileNotFoundException {
        utility = new Utility();

        // Exemplo de nós
        double[] x = new double[] {21, 23, 24, 23, 21, 19, 18, 19};
        double[] y = new double[] {11, 12, 15, 18, 19, 18, 15, 12};

        PrintStream ps = new PrintStream(new FileOutputStream("ExOrder3.txt", true));
        utility.printHeaderOrder(System.out, TAMANHO);

        for(int j=0; j<NEXEC; j++) {
            g = new GenAlgoOrder(x, y, TAMANHO, TAMPOP, NGEN, 0.9, 0.9, 4, false, 21.59338,
                utility.CROSSOVER_EDGE_RECOMBINATION, utility.MUTATION_INVERSION_LIST);
            utility.println(System.out, j, g.getGen(), g.indGlobalFitMax.getFitness(), g.indGlobalFitMax.getString());
            utility.println(ps, j, g.getGen(), g.indGlobalFitMax.getFitness(), g.indGlobalFitMax.getString());
        }
        ps.close();
    }
}

```

Execução

#	Data	Hora	Gen	Fitness	Cromossomo
0	9/03/2017	08:29:58	3000	21,5934	7,8,1,2,3,4,5,6
1	9/03/2017	08:29:58	3000	21,5934	2,1,8,7,6,5,4,3
2	9/03/2017	08:29:59	3000	21,5934	8,7,6,5,4,3,2,1
3	9/03/2017	08:30:00	3000	21,5934	6,7,8,1,2,3,4,5
4	9/03/2017	08:30:00	3000	21,5934	7,6,5,4,3,2,1,8
5	9/03/2017	08:30:01	3000	21,5934	6,5,4,3,2,1,8,7
6	9/03/2017	08:30:01	3000	21,5934	1,8,7,6,5,4,3,2
7	9/03/2017	08:30:02	3000	21,5934	2,1,8,7,6,5,4,3
8	9/03/2017	08:30:02	3000	21,5934	8,1,2,3,4,5,6,7
9	9/03/2017	08:30:02	3000	21,5934	6,7,8,1,2,3,4,5
10	9/03/2017	08:30:02	3000	21,5934	8,1,2,3,4,5,6,7
11	9/03/2017	08:30:03	3000	21,5934	4,3,2,1,8,7,6,5
12	9/03/2017	08:30:03	3000	21,5934	8,1,2,3,4,5,6,7
13	9/03/2017	08:30:03	3000	21,5934	3,4,5,6,7,8,1,2
14	9/03/2017	08:30:04	3000	21,5934	4,5,6,7,8,1,2,3
15	9/03/2017	08:30:04	3000	21,5934	2,3,4,5,6,7,8,1