**Evolução Diferencial – Balanceamento nutricional de refeição (Base)**

**Problemática:** suponha que em determinada refeição, deve haver **30% de proteínas**, **15% de gorduras** e **55% de carboidratos**. As opções de alimentos são (porções de 100g):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nutrientes** | **Peito de Frango** | **Batata Doce** | **Arroz Integral** | **Ovo** | **Feijão** |
| *Proteína* | 23 g | 2 g | 2,6 g | 13 g | 9,5 g |
| *Gordura* | 5 g | 0 g | 1 g | 8,9 g | 1,4 g |
| *Carboidrato* | 5 g | 24 g | 26 g | 1,5 g | 29 g |

**Objetivo:** determinar qual combinação de quantidades desses alimentos resulta nos % desejados.

**Função-objetivo:** a **soma absoluta** dos carboidratos (diferença em relação a 55), gorduras (diferença em relação a 15) e proteínas (diferença em relação a 30) tem que chegar a **ZERO (minimização)**.

**Classe Prato –Introdução**

import java.util.Vector;

class **Prato** {

private Vector alimentos;

public Prato() {

alimentos = new Vector(initialCapacity: 5);

for (int i=0; i<5; i++)

alimentos.add(index: i, java.lang.Math.random() \* 200);

}

public Prato(int tamanho){

alimentos = new Vector(initialCapacity: tamanho);

}

public void **add**(double valor) {

getAlimentos().add(e: valor);

}

public String **toString**() {

String prato = "[";

for (int i=0; i<5; i++){

prato += ((Double)getAlimentos().get(index: i)).toString();

if (i<4)

prato += ", ";

}

prato += "]";

return prato;

}

/\*\*

\* @return the alimentos

\*/

public Vector **getAlimentos**() {

return alimentos;

}

/\*\*

\* @param alimentos the alimentos to set

\*/

public void **setAlimentos**(Vector alimentos) {

this.alimentos = alimentos;

}

}

**Classe Evolucao Diferencial – População**

class **EvolucaoDiferencial** {

private Vector pratos;

private int tamanho;

private static double CR = 0.3;

private static double F = 0.8;

public **EvolucaoDiferencial**(int tamanho) {

this.tamanho = tamanho;

pratos = new Vector(initialCapacity: this.getTamanho());

}

public Vector **populacao**() {

for (int i=0; i<this.getTamanho(); i++){

getPratos().add(new Prato());

}

return getPratos();

}

public double **fitness**(Prato umPrato) {

// prato/refeição deve possuir 30% de proteína, 15% de gorduras e 55% de carboidratos

double carboidratos = ((Double)umPrato.getAlimentos().get(0))\*0.05 +

((Double)umPrato.getAlimentos().get(index: 1))\*0.24 +

((Double)umPrato.getAlimentos().get(index: 2))\*0.26 +

((Double)umPrato.getAlimentos().get(index: 3))\*0.15 +

((Double)umPrato.getAlimentos().get(index: 4))\*0.29;

double proteinas = ((Double)umPrato.getAlimentos().get(0))\*0.23 +

((Double)umPrato.getAlimentos().get(index: 1))\*0.02 +

((Double)umPrato.getAlimentos().get(index: 2))\*0.026 +

((Double)umPrato.getAlimentos().get(index: 3))\*0.13 +

((Double)umPrato.getAlimentos().get(index: 4))\*0.095;

double gorduras = ((Double)umPrato.getAlimentos().get(0))\*0.05 +

((Double)umPrato.getAlimentos().get(index: 1))\*0.00 +

((Double)umPrato.getAlimentos().get(index: 2))\*0.01 +

((Double)umPrato.getAlimentos().get(index: 3))\*0.089 +

((Double)umPrato.getAlimentos().get(index: 4))\*0.014;

double total, porcaoCarboidratos, porcaoProteinas, porcaoGorduras;

double diffTotal, diffCarboidratos, diffProteinas, diffGorduras;

total = carboidratos + proteinas + gorduras;

porcaoCarboidratos = (carboidratos/total) \* 100;

porcaoProteinas = (proteinas/total) \* 100;

porcaoGorduras = (gorduras/total) \* 100;

diffCarboidratos = java.lang.Math.abs(porcaoCarboidratos-55);

diffProteinas = java.lang.Math.abs(porcaoProteinas-30);

diffGorduras = java.lang.Math.abs(porcaoGorduras-15);

diffTotal = diffCarboidratos + diffProteinas + diffGorduras;

return diffTotal;

}

public Prato **mutacao**(int parental, Vector vetores3) {

int cont = 0;

Prato A, B, C;

double X;

Prato pratoParental = (Prato)getPratos().get(parental);

Prato tentativa = new Prato(tamanho: 5);

A = (Prato)vetores3.get(index: 0);

B = (Prato)vetores3.get(index: 1);

C = (Prato)vetores3.get(index: 2);

while (cont < pratoParental.getAlimentos().size()){

double R = java.lang.Math.random();

if (R < getCR()){

X = (Double)A.getAlimentos().get(index: cont) +

getF()\*((Double)B.getAlimentos().get(index: cont) -

(Double)C.getAlimentos().get(index: cont));

} else {

X = (Double)pratoParental.getAlimentos().get(index: cont);

}

tentativa.add(Valor: X);

cont++;

}

return tentativa;

}

public int **melhorVetor**() {

Vector notas = new Vector();

int melhor = 0;

for (int i=0; i<this.getTamanho(); i++){

notas.add(e: fitness((Prato)getPratos().get(index: i)));

}

for (int i=0; i<notas.size(); i++){

if ((Double)notas.get(index: melhor) > (Double)notas.get(index: i))

melhor = i;

}

return melhor;

}

public void **removeVetor**(int indice) {

getPratos().remove(index: indice);

}

public void **addPrato**(Prato umPrato) {

getPratos().add(e: umPrato);

}

public Vector **seleciona3**(int parental) {

Vector p2 = new Vector();

Vector vetores3 = new Vector(initialCapacity: 3);

for (int i=0; i<this.getTamanho(); i++){

if (parental != i)

p2.add((Prato)getPratos().get(index: i));

}

for (int i=0; i<3; i++){

int indice = (int)(java.lang.Math.random() \* (p2.size()-i));

vetores3.add((Prato)p2.get(index: indice));

p2.remove(index: indice);

}

return vetores3;

}

public String **toString**() {

String populacao = "[";

for (int i=0; i<this.getTamanho(); i++)

populacao += getPratos().get(index: i);

populacao += "]";

return populacao;

}

/\*\*

\* @return the pratos

\*/

public Vector **getPratos**() {

return pratos;

}

/\*\*

\* @param pratos the pratos to set

\*/

public void **setPratos**(Vector pratos) {

this.pratos = pratos;

}

/\*\*

\* @return the tamanho

\*/

public int **getTamanho**() {

return tamanho;

}

/\*\*

\* @param tamanho the tamanho to set

\*/

public void **setTamanho**(int tamanho) {

this.tamanho = tamanho;

}

/\*\*

\* @return the CR

\*/

public static double **getCR**() {

return CR;

}

/\*\*

\* @param aCR the CR to set

\*/

public static void **setCR**(double aCR) {

CR = aCR;

}

/\*\*

\* @return the F

\*/

public static double **getF**() {

return F;

}

/\*\*

\* @param aF the F to set

\*/

public static void **setF**(double aF) {

F = aF;

}

}

public class **Teste** {

public static void **main**(String[] args) {

double F = 0.8; // peso diferencial (entre 0 e 2)

double CR = 0.3; // taxa mutação (entre 0 e 1)

Vector tres;

Prato tentativa;

EvolucaoDiferencial ed = new EvolucaoDiferencial(tamanho: 5);

ed.populacao();

for (int i=0; i<1000; i++){

for (int j=0; j<5; j++){

tres = ed.seleciona3(parental: j);

tentativa = ed.mutacao(parental: j, vetores3: tres);

if (ed.fitness(umPrato: tentativa) <

ed.fitness((Prato)ed.getPratos().get(index: j))){

ed.removeVetor(indice: j);

ed.addPrato(umPrato: tentativa);

}

}

System.out.println(ed.melhorVetor() + " - " +

(Prato)ed.getPratos().get(index: ed.melhorVetor()));

System.out.println("Fitness: " + ed.fitness((Prato)ed.getPratos().get(ed.melhorVetor())));

}

}

}