clear all; %Limpa todas as variáveis

ND = 15; %Indica a dimensao da matriz no. nos x no. de nos

MQ = zeros(ND,ND); %Cria a matriz de vazao nula

% ====== PASSA OS VALORES NÃO NULOS PARA A MATRIZ DE VAZAO =========

MQ(4,1) = 74.83;

MQ(4,2)= 36.86;

MQ(9,1) = 33.7;

MQ(5,2) = 76.74;

MQ(6,3) = 56.89;

MQ(7,3) = 50.98;

MQ(8,4) = 59.16;

MQ(10,4) = 22.53;

MQ(8,5) = 6.24;

MQ(12,5) = 40.50;

MQ(9,6) = 9.25;

MQ(11,7) = 20.98;

MQ(12,8) = 20.40;

MQ(10,9) = 12.95;

MQ(11,10) = 5.48;

MQ(12,11) = 9.10;

MQ(13,12) = 20;

MQ(15,13) = 20;

MQ(11,6) = 17.64;

% ------- PASSA VETOR DE DEMANDAS NODAIS (EM LINHA) -----------------

DM = [30 25 30 30 30 30 30 45 30 30 35 50 0 0 20]; %Vetor de demandas

criterio = 0.5; %criterio de cobertura

%---------------- PARAMETROS DO AG ----------------------------------

NGen = 60; %numero de gerações

ind\_pop = 20; %Numero de individuos na população

num\_est = 4; %Numero de estações de montoramento (deve ser par)

num\_nos = 15; %Numero total de nós

prob\_mut = 0.07; %probabilidade de mutação

% --------------- FIM DA ENTRADA DE DADOS ----------------------------

%================== PROCEDIMENTO DE CÁLCULO ==========================

% Determinação da localização ótima de estações de monitoramento de

% qualidade de água ..................................................

% -------------- TRABALHO DE FINAL DE CURSO --------------------------

% Aluno: Hugo S. Hirose

% Orientador: Prof. Dr. Edevar Luvizotto Junior

% Conclusão: Dezembro/2010

$=====================================================================

% ================ INICIO DO PROCEDIMENTO ============================

DEM = DM'; %vetor demandas em coluna;

[f] = M\_FRAC\_AGUA (ND, MQ); %Gera matriz da fração de água

[f] = MAT\_COBERTURA (ND,f,criterio); %Gera matriz de cobertura global

[POP] = POP\_INICIAL(ind\_pop, num\_est,ND);%Gera população inicial aleatória

%Calcula função objetivo da pop.

[vajuste,POP] = F\_OBJETIVO (ind\_pop, f, POP, ND,num\_est,DEM);

%-------------- Chama a rotina de AG ------------------------------------

new\_gera = 0; %zera contador de gerações

clc; %limpa espaço de trabalho para saída de resultados

disp('Geração Ajuste Media Melhor Individuo')

while new\_gera < NGen

new\_gera = new\_gera+1;

[POP] = AG (num\_est, ND, ind\_pop, POP, prob\_mut); %gera nova população

%Avalia população

[vajuste,POP] = F\_OBJETIVO (ind\_pop, f, POP, ND,num\_est,DEM);

% -------- Retem dados para impressão e graficos ------------------------

meida = mean(vajuste); %Media do vetor auxiliar das funções objetivos

[maximo,ind\_max] = max(vajuste); %Localiza o maximo e o indice da melhor

Melhor\_individuo = POP(ind\_max).cromossomo; %Cromossomo da melhor solução

Tab= [new\_gera POP(ind\_max).ajuste meida Melhor\_individuo]; %Impressão

disp(Tab);

% ----------------------------------------------------------------------

end %Fim do processamento das gerações

**Rotina 01 – Rotina Principal**

function [ f ] = M\_FRAC\_AGUA( ND, MQ )

%== M\_FRAQ\_AGUA Calcula Matriz de fração de água =====================

% ND - Numero de nós que compõe a rede

% %numerados sequencial à partir de 1

% MQ - Matriz das vazões dos elementos entre os nós (linha e coluna)

% que contribue com a vazão para o nó de jusante

% f - Retorna ma matriz de fração de água

% ======================================================================

f = MQ; %Cria uma copia matriz de vazao

%Que retorna a fração de água

for L = 1:1:ND;

Vsoma(L) = 0;

%Encntra a soma das vazoes para os nos de jusante

for C=1:1:ND

Vsoma(L) = Vsoma(L)+MQ(L,C);

end

end

% ----------- fim totalizacao de vazoes aos nos -----------------------

% --------------- CRIA MATRIZ DE FRAÇÃO DE ÁGUA ------------------------

for L=1:1:ND

for C=1:1:ND

if Vsoma(L) == 0

Vsoma(L)=1;

end

f(L,C) = MQ(L,C)/Vsoma(L);

end

end

% ---------- Da primeita parte (contribuição direta) -------------------

f = f'; % Traspõe a matriz de fração para forma usual - vide texto

%---------------------------------------------------------------------

% Obtençao dos contribuiçoes de nós indiretamente contribuintes

% ----------------------------------------------------------------------

for i = 1:1:ND

f(i,i) = 1; % fixa o valor 1 na diagonal da matriz de fração

end

% Inicio do calculo das contribições e armazenamento na matriz de fração

for C =2:1:ND

for L=1:1:C-1

if f(L,C) ~= 0

ID = L;

MUL = f(L,C);

for K = 1:1:ID-1

f(K,C) = f(K,C)+f(K,ID)\*MUL;

end

end

end

end

% ------ fim da rotina de obtenção da matriz de fração de agua -----------

end

**Rotina 02 – Para obter a Matriz de fração de água**

function [FF] = MAT\_COBERTURA(ND, f, criterio)

%MAT\_COBERTURA - Obtem A matriz de cobertura associada a f

% Sendo: ND numero de nós na rede hidráulica (dimensão NDXND de f)

% f - Matriz de fração de vazão -> Matriz de cobertura FF

% criterio - percentagem da vazao total coberta

%-------------------------------------------------------------------

for L=1:1:ND

for C=1:1:ND

if f(L,C) >= criterio

f(L,C) = 1;

else

f(L,C) = 0;

end

end

end

FF = f;

end

**Rotina 03 – Para obter a Matriz de cobertura**

function [ Pop ] = POP\_INICIAL(ind\_pop,num\_est, num\_nos)

% POP\_INICIAL - Gera populacao inicial

% ind\_pop - Numero de individuos da população

% num\_est - numero de estações de munitoramento

% num\_nos - numero total de nós da rede

%-------------------------------------------------------

pop\_ini = zeros(ind\_pop,num\_est);

%Geração dos indivduos da população inicial

for i=1:1:ind\_pop

gera = randperm(num\_nos);

for j=1:1:num\_est

pop\_ini(i,j) = gera(j);

end

Pop(i).indice = i;

Pop(i).cromossomo = pop\_ini(i,:);

POP(i).ajuste = 0;

end

end

**Rotina 04 – Para obtenção de soluções iniciais aleatórias**

function [vajuste, POP] = F\_OBJETIVO (ind\_pop, f, individuo, ND,NE,DEM)

% F\_ONJETIVO - Calcula a função objetivo -----------------------------

% ind\_pop - individuos da população

% f - matriz de cobertura

% individuo - Individuos da população (tipo estuturado)

% ND - Numero de nós da rede

% NE - numero de estações de monitoramento

% DEM - vetor de demandas nos nós

% z - valor máximo provavel

% FE - fator de escala

% Veajuste - Veotr com os valores da função objetico da população

% sem escala

%-----------------------------------------------------------------------

Vetcob = zeros (ND,NE);

for i=1:1:ind\_pop

ajust = 0;

ESTAMON = individuo(i).cromossomo;

%----------- Aptidao da população gerada ------------------------------

sumq = 0;

for L =1:1:ND %Varre as linhas - correspondentes a todos os nós

soma = 0;

for C =1:1:NE %Varre as colunas com as estações de monitoramento

soma = soma+f(L,ESTAMON(C));

end

if soma >= 1 % Verifica se há columa não nula

Vetcob (L) = 1;

else

Vetcob(L) = 0;

end

VCD(L) = Vetcob(L)\*DEM(L);

sumq = sumq + VCD(L);

end

FR = sumq; %substituiu FR = 100-((z-sumq)/z\*100);

vajuste(i) = FR;

end

for i=1:1:ind\_pop

individuo(i).ajuste = vajuste(i);

end

POP = individuo;

%---------- Foi excluida a escala da função prevista ------------------

% (ver texto)

%---------------- Fim da escala de ajuste

end

**Rotina 04 – Para obtenção da aptidão dos indivíduos (soluções)**

function [aux\_indiv] = AG (num\_est, ND, ind\_pop, individuo, prob\_mut)

% AG - Rotina de algoritimo gtenetico

% num\_est - numero de estações de monitoramento

% ND - numero de nos da rede

% ind\_pop - numero de individuos da população

% individuo - vetor de dados dos individuos da popuação

% prob\_mu - probabilidade de mutação

% aux\_indiv - Vetor com individuos da nova população

indx = 0;

while indx < ind\_pop

indx = indx + 1;

% ----------------- Seleção por torneio ------------------------------

for j = 1:1:2 %Pais 1 e 2 - seleção

duelante1 = randperm(ind\_pop);

duelante2 = randperm(ind\_pop);

for i = 1:1:ind\_pop

if individuo(duelante1(i)).ajuste < individuo(duelante2(i)).ajuste

pais(j) = individuo(duelante2(i));

else

pais(j) = individuo(duelante1(i));

end

end

end

%-------------------- operador Crossover -----------------------------

part = 1+(num\_est-1)\*rand;

for im =1:1:num\_est %Cria uma mascara para Cruzamento

if im < part

MASK(im) = 1;

else

MASK(im) = 0;;

end

end;

for kl = 1:1:num\_est

if MASK(kl) == 1

filhos(1).cromossomo(kl) = pais(1).cromossomo(kl);

filhos(2).cromossomo(kl) = pais(2).cromossomo(kl);

else

filhos(1).cromossomo(kl) = pais(2).cromossomo(kl);

filhos(2).cromossomo(kl) = pais(1).cromossomo(kl);

end

end

% --------- verificar se o filho não é defeituoso ----------------------

for id=1:1:(num\_est-1) %verifica se o filho 1 é normal

for jd = (id+1):1:num\_est

if filhos(1).cromossomo(id) == filhos(1).cromossomo(jd)

filhos(1).cromossomo = pais(1).cromossomo;

jd = num\_est;

end

end

end

for id=1:1:(num\_est-1) %verifica se o filho 2 é normal

for jd =(id+1):1:num\_est

if filhos(2).cromossomo(id) == filhos(2).cromossomo(jd)

filhos(2).cromossomo = pais(2).cromossomo;

jd = num\_est;

end

end

end

%------ cria uma nova população intermediaria de individuos -----------

aux\_indiv(indx).cromossomo = filhos(1).cromossomo;

aux\_indiv(indx).indice = indx;

aux\_indiv(indx).ajuste = 0;

indx = indx+1;

aux\_indiv(indx).cromossomo = filhos(2).cromossomo;

aux\_indiv(indx).indice = indx;

aux\_indiv(indx).ajuste = 0;

end %---------------------------------------------------------------------

%--------------------- Operador Mutação -------------------------------

for i=1:ind\_pop

for j =1:1:num\_est

if rand < prob\_mut

repetiu = 1;

while repetiu == 1

repetiu = 0;

no\_aux = round(ND\*rand);

if no\_aux == 0

no\_aux = 1;

elseif no\_aux >= ND

no\_aux = ND;

end

for ik=1:1:num\_est

if no\_aux == aux\_indiv(i).cromossomo(ik)

repetiu = 1;

end

end

end

aux\_indiv(i).cromossomo(j) = no\_aux; %gene sofre mutação

end

end

end

end

**Rotina 05 – Algoritmo Genético - Para obtenção da solução**