/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 \* OPL 12.8.0.0 Model

 \* Author: rodrigo rosa

 \* Creation Date: 10/05/2019 at 13:07:52

 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//NFornecedores

int nf=...;

//NArmazem

int na=...;

//NClientes

int nc=...;

//CONJUNTO fornecedor

range I = 1 .. nf;

//CONJUNTO armazem

range J = 1 .. na;

//CONJUNTO clientes

range K = 1 .. nc;

//custo fornecedor armazem

float cfa[i in I][j in J] =...;

//custo armazem cliente

float cac[j in J][k in K] = ...;

// Capacidade armazem

float cpa[j in J]= ...;

// Custo fixo de instalacao do armazem

float cia[j in J]= ...;

// Oferta do fornecedor i

float o[i in I]=...;

//demanda cliente

float d[k in K] = ...;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// VARIAVEIS

// Yi - Variavel binaria que assume valor 1 se a estacao de transbordo e escolhida para ser implantada e 0 caso contrario;

dvar int y[j in J];

// x - fluxo fornecedor armazem

dvar float x[i in I][j in J];

// w - fluxo armazem cliente

dvar float w[j in J][k in K];

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// FUNCAO OBJETIVO

 minimize

 sum(j in J) cia[j] \* y[j] + sum(i in I, j in J) cfa[i,j] \* x[i,j] + sum(j in J, k in K) cac[j,k] \* w[j,k];

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// RESTRICOES

subject to

{

//RESTRICAO 1: garante que o total transportado do forencedor para o armazem nao exceda a oferta do fornecedor

 forall (i in I){

 Const1: sum (j in J) x[i,j] <= o[i];}

//RESTRICAO 2: garante que tudo que chega ao armazem vindo do fornecedor seja igual ao que sai do armazem para o cliente

 forall (j in J) {

 Const2: sum (i in I) x [i,j] == sum (k in K) w [j,k];}

//RESTRICAO 3:garante que o total de residuos que flui de todos os terminais para uma estacao de transbordo nao exceda a capacidade

 forall (j in J) {

 Const3: sum (i in I) x[i,j] <= y[j] \* cpa[j] ;}

//y deve ser entre 0 e 1, lembrando que e inteira

 forall (j in J) {

 const4: 0 <= y[j] <= 1; }

//RESTRICAO 6: garante que toda demanda seja atendida

 forall (k in K) {

 Const6: d[k] == sum (j in J) w[j,k] ;}

//RESTRICAO 7: garante que as variaveis sejam reais e positivas.

 forall (i in I, j in J){

 const8: x[i,j]>=0;}

 forall (j in J, k in K){

 const9: w[j,k]>=0;}

}

 //

Aqui começa o dat

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 \* OPL 12.8.0.0 Data

 \* Author: rodrigo rosa

 \* Creation Date: 10/05/2019 at 13:07:52

 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

 //NFornecedores

nf=2;

//NArmazem

na=3;

//NClientes

nc=4;

// custo do fornecedor para armazem

cfa = [

 [0.0 0.0 0.0]

 [0.7 2.1 3.4]

 ];

// custo armazem cliente

cac = [

 [23.4 29.9 37.2 31.2]

 [21.9 28.4 35.7 29.7]

 [22.5 29.0 36.3 30.3]

 ];

// custo de implantar o armazem

cia = [1361.3 2841.3 999999.9];

// capacidade armazem

cpa = [2.0 7.0 6.0];

// oferta fornecedor

o = [3.0 6.0];

// demanda cliente

d = [3.00 1.00 1.00 4.00];