

## ANNEX B: TRADUCCIÓ DEL MODEL AL LENGUATGE DE PROGRAMACIÓ DE L'OPLStudio

/\* Inicio de la declaración de parámetros de dimensión asociados al problema\*/

/\*alimentados desde tabla excel\*/

SheetConnection datos("z:\Desktop\dades.xls", 1);

/\*nº de personal en plantilla\*/

{int} PP from SheetRead(datos, "P");

int+ P=first (PP);

/\*nº de tareas a asignar\*/

{int} TT from SheetRead(datos, "T");

int+ T=first (TT);

/\*nº de intervalos\*/

{int} HH from SheetRead(datos, "H");

int+ H=first (HH);

/\*intervalos máximos seguidos realizando misma tarea\*/

{int} EE from SheetRead(datos, "E");

int+ E=first (EE);

/\*Activar o desactivar restricción 1\*/

{int} r1r1 from SheetRead(datos, "rr1");

int+ r1=first (r1r1);

/\*Activar o desactivar restricción 2\*/

{int} r2r2 from SheetRead(datos, "rr2");

int+ r2=first (r2r2);

/\*Activar o desactivar restricción 3\*/

{int} r3r3 from SheetRead(datos, "rr3");

int+ r3=first (r3r3);

/\*Activar o desactivar restricción 4\*/

{int} r4r4 from SheetRead(datos, "rr4");

int+ r4=first (r4r4);

/\*Activar o desactivar restricción 5\*/

{int} r5r5 from SheetRead(datos, "rr5");

int+ r5=first (r5r5);

/\*Peso para el criterio o función objetivo 1\*/

{int} C1C1 from SheetRead(datos, "CCC1");

int+ C1=first (C1C1);

/\*Peso para el criterio o función objetivo 2\*/

{int} C2C2 from SheetRead(datos, "CCC2");

int+ C2=first (C2C2);

/\*Peso para el criterio o función objetivo 3\*/

{int} C3C3 from SheetRead(datos, "CCC3");

int+ C3=first (C3C3);

/\*Peso para el criterio o función objetivo 4\*/

{int} C4C4 from SheetRead(datos, "CCC4");

int+ C4=first (C4C4);

/\*inicio de los rangos para los contadores\*/

## ANNEX B: TRADUCCIÓ DEL MODEL AL LENGUATGE DE PROGRAMACIÓ DE L'OPLStudio

range

binaria 0..1,  
tareas 1..T,  
personal 1..P,  
intervalos 1..H;

//definición de los datos

```
/*Alimentados desde tabla excel*/
/*Demanda para la tarea t en el intervalo h*/
float+ Dth[tareas, intervalos] from SheetRead(datos, "Dth");
/*Carga de trabajo para la tarea t*/
float+ Nt[tareas] from SheetRead(datos, "Nt");
/*Rendimiento del trabajador p realizando t*/
float+ Rpt[personal, tareas] from SheetRead(datos, "Rpt");
/*Matriz de presencia para cada trabajador en cada intervalo*/
float+ MPph[personal, intervalos] from SheetRead(datos, "MPph");
/*Matriz de preferencia a las tareas por los trabajadores*/
float+ PPpt[personal, tareas] from SheetRead(datos, "PPpt");
/*Rendimiento de calidad de cada trabajador realizando cada tarea*/
float+ RQpt[personal, tareas] from SheetRead(datos, "RQpt");
/*Nº de intervalos ideales a realizar misma tarea en todo H*/
float+ ITt[tareas] from SheetRead(datos, "ITt");
/*Matriz de polivalencia de los trabajadores*/
int+ MPOLpt[personal, tareas] from SheetRead(datos, "MPOLpt");

/*Parámetros calculados a partir de los datos de partida*/
/*Trabajadores presentes en cada h*/
float+ TPh[intervalos];
initialize{
forall (h in intervalos)
TPh[h]=sum(p in personal)MPph[p,h];
};
/*Demanda equivalente de persona para cada tarea en cada intervalo*/
float+ DEth[tareas, intervalos];
initialize{
forall (t in tareas, h in intervalos)
DEth[t,h]=Dth[t,h]*Nt[t];
};
/*Carga equivalente media en cada intervalo*/
float+ XIMh[intervalos];
initialize{
forall (h in intervalos)
XIMh[h]=(sum(t in tareas)DEth[t,h])/TPh[h];
};
```

## ANNEX B: TRADUCCIÓ DEL MODEL AL LENGUATGE DE PROGRAMACIÓ DE L'OPLStudio

```
float+ DFO1;
initialize{
DFO1=sum(p in personal, h in intervalos)MPph[p,h]*XIMh[h]
};

float+ DFO2;
initialize{
DFO2=sum(t in tareas, h in intervalos)Dth[t,h]
};

float+ DFO3;
initialize{
DFO3=sum(t in tareas, h in intervalos)Dth[t,h]
};

float+ DFO4;
initialize{
DFO4=sum(p in personal, t in tareas)MPOLpt[p,t]*H
};

//limitación de los datos

assert forall (h in intervalos) sum(p in personal)(MPph[p,h])<=P; //1
assert forall (p in personal) sum(t in tareas)(PPpt[p,t])<=1; //2
assert forall (p in personal) sum(t in tareas)(Rpt[p,t])>0; //3
assert forall (t in tareas) sum(p in personal)(Rpt[p,t])>0; //4

//declaración de variables
/*variable principal del modelo. Binaria -- Si a p se le asigna t en h o no.*/
var binaria Xpth[personal, tareas, intervalos];

/*Carga equivalente asignada a cada trabajador en cada intervalo*/
var float XIph[personal, intervalos];
/*Personal equivalente asignado realizando cada tarea en cada intervalo*/
var float XEth[tareas, intervalos];
/*Nº de intervalos que cada trabajador realiza la misma tarea en todo H*/
var float XITpt[personal, tareas];
/*Variable que indica el valor obtenido para la función objetivo 1*/
var float FO1;
/*Variable que indica el valor obtenido para la función objetivo 2*/
var float FO2;
/*Variable que indica el valor obtenido para la función objetivo 3*/
var float FO3;
/*Variable que indica el valor obtenido para la función objetivo 4*/
var float FO4;
```

## ANNEX B: TRADUCCIÓ DEL MODEL AL LENGUATGE DE PROGRAMACIÓ DE L'OPLStudio

```
/*Discrepancia por arriba entre XIph y XIMh (carga real y carga media o ideal)*/
var float+ dXISUPph[personal, intervalos];
/*Discrepancia por abajo entre XIph y XIMh (carga real y carga media o ideal)*/
var float+ dXIMINph[personal, intervalos];
/*Discrepancia por arriba entre XITph y ITt*/
var float+ dXITSUPpt[personal, tareas];
/*Discrepancia por arriba entre XITph y ITt*/
var float+ dXITMINpt[personal, tareas];
/*Equivalente a Xpth, pero necesaria para cálculo de intervalos seguidos misma t*/
var binaria Xptj[personal,tareas,intervalos];

/*definición de la función objetivo (Multicriterio)*/

maximize

C1*FO1+C2*FO2+C3*FO3+C4*FO4

subject to{

//definición de las variables declaradas anteriormente

forall (p in personal, h in intervalos) //carga real asignada a p en h
XIph[p, h]=sum(t in tareas)(Xpth[p,t,h]*Nt[t]);

forall (p in personal, t in tareas) //h's asignando t a p en H
XITpt[p,t]=sum(h in intervalos)Xpth[p,t,h];

forall (t in tareas, h in intervalos) //cantidad equiv. p's a t en h
XEth[t,h]=sum(p in personal)(Xpth[p,t,h]*Rpt[p,t]);

forall (p in personal, h in intervalos) //discrepancias XIph y XIMh
XIph[p,h]-dXISUPph[p,h]+dXIMINph[p,h]=XIMh[h];

forall (p in personal, t in tareas) //Discrepancias XITpt y ITt
XITpt[p,t]-ITt[t]=(dXITSUPpt[p,t]-dXITMINpt[p,t]);

forall (p in personal, t in tareas, h in intervalos) //Xpth para intervalos seguidos
Xptj[p,t,h]=Xpth[p,t,h];

//Función objetivo 1 norm.
FO1=100*(1-(sum(p in personal, h in
intervalos)((dXISUPph[p,h]+dXIMINph[p,h])*MPph[p,h])/DFO1));

//Función objetivo 2 norm.
FO2=100*(sum(p in personal, t in tareas, h in intervalos)RQpt[p,t]*Xpth[p,t,h])/DFO2;
```

## ANNEX B: TRADUCCIÓ DEL MODEL AL LENGUATGE DE PROGRAMACIÓ DE L'OPLStudio

//Función objetivo 3 norm.

FO3=100\*(sum(p in personal, t in tareas, h in intervalos)PPpt[p,t]\*Xpth[p,t,h])/DFO3;

//Función objetivo 4 norm.

FO4=100\*(1-(sum(p in personal, t in tareas)(dXITSUPpt[p,t]+dXITMINpt[p,t])/DFO4));

//definición de las restricciones

```
if (r1=1) then //Restricción 1
  forall (t in tareas, h in intervalos)
    DEth[t,h]<=XEth[t,h]
  else
  endif;
```

```
if (r2=1) then //Restricción 2
  forall (p in personal, h in intervalos)
    XIph[p,h]<=1
  else
  endif;
```

```
if (r3=1) then //Restricción 3
  forall (t in tareas, h in intervalos, p in personal)
    Xpth[p,t,h]<=Rpt[p,t]*10
  else
  endif;
```

```
if (r4=1) then //Restricción 4
  forall (t in tareas, h in intervalos, p in personal)
    Xpth[p,t,h]<=MPph[p,h]
  else
  endif;
```

```
if (r5=1) then //Restricción 5
  forall (p in personal, t in tareas){
    forall (h in [E+1..H]){ //R5-limitar asignaciones seguidas a E
      sum (j in [h-E..h]) Xptj[p,t,j]<=E; }
    else
  endif;
};
```