**movdron.mod**

#Datos (Conjuntos y parámetros)

#Conjuntos:

set C; #Conjunto de condicionantes

set L; #Conjunto de casillas en las que movilizar el dron

#Parámetros:

param Y {i in C} >=0; #Coeficiente de importancia de cada condicionante a cada acción

param S {i in C} binary; #Se da o no cada condicionante

param D {j in L} >=0; #Distancia de la casilla a la base aérea desde donde se despliega

#Variables:

var X {j in L} binary; #Decisión sobre si se lleva a cabo la acción o no

#Función objetivo: Maximizar la puntuación en función de escoger una casilla u otra

maximize FO: sum {i in C, j in L} [Y[i] \* S[i]] \* X[j];

#Restricciones:

#1) No se puede colocar el mismo avión de caza en más de una casilla

R1 sum X[j] =1

#2) La distancia de la casilla debe ser igual o inferior al alcance máximo del dron

R2 D[j] <=16

**movdron.dat**

#Como cargar datos en AMPL

#Conjuntos

set C := C1 C2 C3 C4 C5 C6 C7 C8

##Desarrollo de los condicionantes

#C1: ial

#C2: saeL1

#C3: maeL1

#C4: sapL2

#C5: craL1 >0

#C6: cbaL1 >0

#C7: cbaeL1 >0

#C8: cciL1 >0

##Definición de las abreviaturas utilizadas en los condicionantes

#iaL=igualdad aérea

#saeL1=superioridad aérea enemiga

#maeL1=supremacía aérea enemiga

#sapL2=superioridad aérea propia

#craL1 =Cantidad de radares enemigos en un radio <= 4 casillas

#cbaL1 =Cantidad de baterías antiaéreas enemigas en un radio <=2 casillas

#cbaeL1 = Cantidad de bases aéreas enemigas en un radio <= 1 casillas

#cciL1 =Cantidad de ciudades enemigas en un radio <= 1 casillas

#xx(1)=parámetro asociado a J1

#xx(2)=parámetro asociado a J2 (máquina)

set L := L(j)

##Casillas

#Parámetros

#Coeficientes de importancia de cada condicionante a cada casilla

param Y L(j) :=

C1 2

C2 4

C3 8

C4 6

C5 3

C6 3

C7 10

C8 9

;

#Se da o no cada condicionante

param S:=

;

# Distancia de la casilla a la base aérea desde donde se despliega

param D:=

;

**movdron.run**

#Llamar al modelo y a los datos

model movavoataque.mod;

data movavoataque.dat;

#Llamar al compilador o solver (cplex/gurobi)

option solver gurobi;

#Resuelve el modelo;

solve;

#Muestra los resultados de la variable

display X;