**ataqueavocaza.mod**

#Datos (Conjuntos y parámetros)

#Conjuntos:

set C; #Conjunto de condicionantes

set L; #Conjunto de casillas a las que atacar con el avión de caza

#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 aeronave desde donde se ataca

#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 de ataque máximo de la aeronave

R2 D[j] <=2

**ataquedron.dat**

#Como cargar datos en AMPL

#Conjuntos

set C := C1 C2 C3 C4 C5

##Desarrollo de los condicionantes

#C1: acL1

#C2: aaL1

#C3: atL1

#C4: heL1

#C5: drL1

##Definición de las abreviaturas utilizadas en los condicionantes

#aaL= avión de caza en la casilla

#acL=avión de caza en la casilla

#atL =avión de transporte en la casilla

#heL =helicóptero en la casilla

#drL =dron en la casilla

#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 8

C2 10

C3 6

C4 4

C5 2

;

#Se da o no cada condicionante

param S:=

;

# Distancia de la casilla a la aeronave desde donde se ataca param D:=

;

**ataqueavocaza.run**

#Llamar al modelo y a los datos

model ataqueavocaza.mod;

data ataqueavocaza.dat;

#Llamar al compilador o solver (cplex/gurobi)

option solver gurobi;

#Resuelve el modelo;

solve;

#Muestra los resultados de la variable

display X;