**comprabaseaerea.mod**

#Datos (Conjuntos y parámetros)

#Conjuntos:

set C; #Conjunto de condicionantes

set L; #Conjunto de casillas en las que colocar la base aérea

#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 G {j in L} binary; #Existe supremacía aérea del propio jugador o no en esa casilla

#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 la misma base aérea en más de una casilla

R1 sum X[j] =1

#2) Se debe colocar la ciudad en una casilla con supremacía aérea del propio jugador

R2 A[j] =1

**compraciudad.dat**

#Como cargar datos en AMPL

#Conjuntos

set C := C1 C2 C3 C4

##Desarrollo de los condicionantes

#C1: cbaeL =1

#C2: cbaeL >=2

#C3: craL >0

#C4: cbaL >0

##Definición de las abreviaturas utilizadas en los condicionantes

#cbaeL = Cantidad de bases aéreas en un radio <= 4 casillas

#craL =Cantidad de radares en un radio <= 4 casillas

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

set B := B(j)

##Casillas con superioridad aérea del J2 (máquina)

#Parámetros

#Coeficientes de importancia de cada condicionante a cada casilla

param Y L(j) :=

C1 -3

C2 -5

C3 2

C4 5

;

#Se da o no cada condicionante

param S:=

;

#Existe o no la supremacía aérea del propio jugador en dicha casilla

param G:=

;

**comprabaseaerea.run**

#Llamar al modelo y a los datos

model comprabaseaerea.mod;

data comprabaseaerea.dat;

#Llamar al compilador o solver (cplex/gurobi)

option solver gurobi;

#Resuelve el modelo;

solve;

#Muestra los resultados de la variable

display X;