



Jordi Pereira

*Faculty of Engineering and Sciences
Universidad Adolfo Ibáñez,
Viña del Mar, Chile*

Uso de CPLEX junto a Excel

Varias alternativas

Contenidos

1. Introducción
2. Cplex+Excel
3. OPL+Excel
4. JuMP (+Excel)

Diferentes opciones

- Hay diferentes maneras de asociar Cplex y Excel dependiendo del objetivo buscado.
- Veremos aquí varias maneras que van desde la sustitución del Solver de Excel por Cplex hasta referirse a Excel como una fuente de entrada y salida de datos.
- Alguna de las opciones que mostraremos requiere cierto conocimiento de programación
- IBM ha ido dejando de ofrecer soporte para alguna de las opciones que veremos (específicamente llamar a Cplex desde Excel).

Modelos

A lo largo de los ejemplos veremos dos modelos diferentes que servirán para ilustrar el proceso de modelización y resolución mediante Cplex. Los modelos que veremos son dos problemas clásicos del área:

- Uncapacitated Lot Sizing Problem
- Set Covering Problem

Vamos a ver una formulación de cada problema.

Uncapacitated Lot Sizing

Modelo estándar de inventarios con demanda determinista y heterogénea en T periodos.

$$\min \sum_{t=1}^T c_I y_t + \sum_{t=1}^T c_s s_t$$

s.t.

$$x_t + s_{t-1} = d_t + s_t \quad 1 \leq t \leq T;$$

$$x_t \leq M y_t \quad 1 \leq t \leq T;$$

$$s_0 = 0$$

$$y_t \in \{0, 1\}, \quad x_t \geq 0, \quad s_t \geq 0 \quad 1 \leq t \leq T.$$

Set Covering

Cubrimiento de conjuntos y subproblema de otros modelos más complejos (itinerarios, localización, *timetabling*,...).

$$\min \sum_{i=1}^n c_i x_i$$

s.t.

$$\sum_{i=1}^n a_{ij} x_i \geq 1 \quad 1 \leq j \leq m;$$

$$x_i \in \{0, 1\} \quad 1 \leq i \leq n.$$

Introducción

Pros:

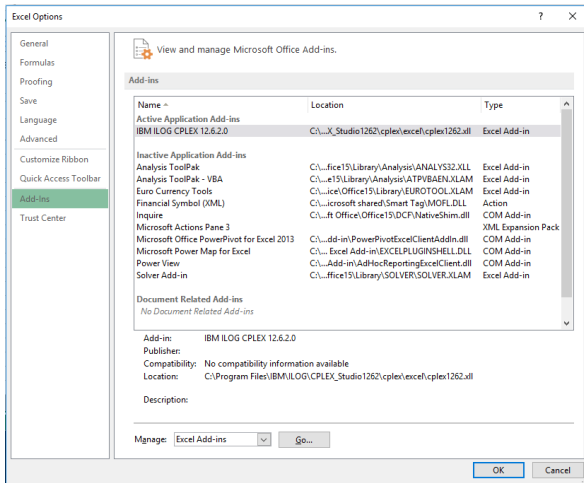
- A priori, la forma más sencilla de hacer las cosas.
- Cplex sustituye a Excel Solver utilizando un interfaz muy similar al del segundo.
- Puede incluso llamarse desde una macro.

Contras:

- IBM ya no ofrece soporte para este método (esta opción ya no existe en Cplex 12.7 ni Cplex 12.8, además el módulo dejó de desarrollarse en Excel 2010).
- En pocas palabras: *“Use at your own risk”* a sabiendas que tarde o temprano dejará de funcionar según se vayan actualizando los equipos.

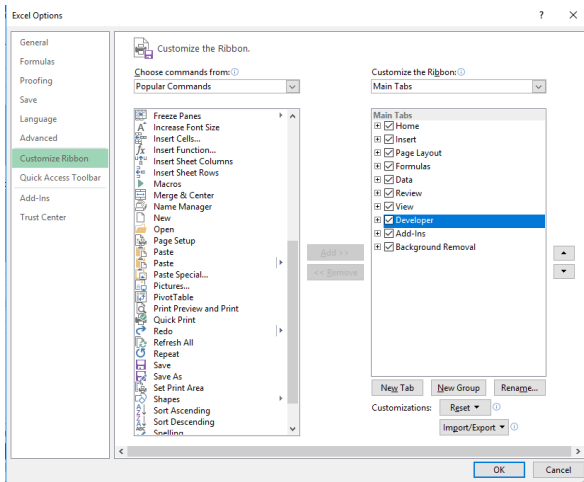
Activando Cplex en Excel

Tenemos que activar un módulo en Excel (menú Opciones).



Activando VBA

Tenemos que mostrar las opciones de desarrollo.



Usar Cplex

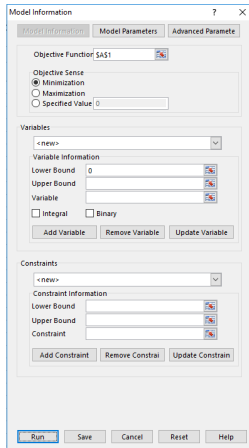
Usar Cplex como sustituto de Solver requiere de los mismos pasos que usaríamos con Solver.

- 1 Construir la hoja de cálculo
- 2 Determinar las variables (casillas)
- 3 Determinar la función objetivo (casilla)
- 4 Construir el lado izquierdo y derecho de las restricciones

Puede haber algunos “problemillas” de uso.

Ejemplo ULS

Podemos descargar la hoja de cálculo inicial para hacer el ejercicio de goo.gl/n919co



Ejemplo ULS

La versión finalizada puede descargarse de goo.gl/xyVyzr

The screenshot shows the 'Model Information' dialog box in CPLEX. It has three tabs: 'Model Information' (selected), 'Model Parameters', and 'Advanced Parameters'. The 'Objective Function' is set to '\$A\$1'. The 'Objective Sense' is set to 'Minimization'. The 'Variables' section shows a dropdown menu with '<new>' selected. The 'Variable Information' section shows 'Lower Bound' set to 0, 'Upper Bound' set to 0, and 'Variable' set to 0. The 'Constraints' section shows a dropdown menu with '<new>' selected. The 'Constraint Information' section shows 'Lower Bound' set to 0, 'Upper Bound' set to 0, and 'Constraint' set to 0. At the bottom, there are buttons for 'Run', 'Save', 'Cancel', 'Reset', and 'Help'.

Model Information

Model Information Model Parameters Advanced Parameters

Objective Function: \$A\$1

Objective Sense

- ☒ Minimization
- ☐ Maximization
- ☐ Specified Value: 0

Variables

<new>

Variable Information

Lower Bound: 0

Upper Bound: 0

Variable: 0

☐ Integral ☐ Binary

Add Variable Remove Variable Update Variable

Constraints

<new>

Constraint Information

Lower Bound: 0

Upper Bound: 0

Constraint: 0

Add Constraint Remove Constraint Update Constraint

Run Save Cancel Reset Help

Ejemplo SCP (versión 1)

- De forma parecida al caso anterior, podemos resolver el modelo de cubrimiento. Descargar hoja de cálculo de goo.gl/8M46Ry y de goo.gl/fFw9Tj para la hoja completa.
- En este caso podemos visualizar que la técnica anterior no escala bien para problemas grandes ni formulaciones en que la matriz A está formada mayoritariamente por entradas iguales 0 (lo que se entiende como una matriz dispersa).
- Aquí es cuando se aprecia la oportunidad de mejorar el flujo de trabajo construyendo el modelo de forma programática.

Utilizando macros

- Generalmente, la información del modelo anterior se hubiera entregado como una lista de costos y otra de no-ceros.
- Partiendo de esas listas podemos construir el modelo matemático.
- Vamos a usar macros de Excel (lenguaje Visual Basic for Applications).

No se preocupen, pueden bajar el archivo directamente (goo.gl/6v72sD) y podemos analizar el código paso a paso.

Ejemplo SCP (versión 2)

```

Sub SCP()
    nVar = Sheet1.Cells(1, 2)      'vamos a leer el número de variables y restricciones del modelo
    nConstr = Sheet1.Cells(2, 2)
    For i = 2 To nVar + 1          'de la hoja original leemos costos (valor) y escribimos nombre
        ActiveSheet.Cells(i, 1).Value = "x" + Str(i - 1)
        Valor = Sheet1.Cells(i + 1, 2)
        ActiveSheet.Cells(i, 3).Value = Valor
    Next
    Dim Vars As Range              'creamos un rango de excel para indicar variables
    Set Vars = Range(ActiveSheet.Cells(2, 2), ActiveSheet.Cells(1 + nVar, 2))
    Dim ObjCoefs As Range          'creamos un rango de Excel para indicar los coeficientes y escribir la fórmula de coste
    Set ObjCoefs = Range(ActiveSheet.Cells(2, 3), ActiveSheet.Cells(1 + nVar, 3))
    ActiveSheet.Cells(1, 4).Formula = "=SUMPRODUCT(" & Vars.Address & "," & ObjCoefs.Address & ")"      'escribimos fórmula en casilla D1 (D == 4)
    CFXclear                      'Limpiamos datos de CPLEX para empezar el cálculo.
    CFXsetObjective ObjCell:=ActiveSheet.Cells(1, 4), Sense:=2          'Sense = 2 indica minimizar
    CFXaddVariable Variable:=Vars, Binary:=True                      'Definimos variables como binarias.
    For cCounter = 1 To nConstr    ' vamos a construir las nConstr restricciones
        ActiveSheet.Cells(cCounter + 1, 5).Value = "c" + Str(cCounter)      'nombre
        'contamos el número de elementos a sumar
        numNz = Sheet1.Cells(cCounter + nVar + 3, 2).Value 'el +3 son las líneas no usadas para variables
        Texto = ""
        For nzCounter = 1 To numNz
            'Replace elimina un texto en blanco creado por Str
            'Str transforma un número en un texto para poder pegarlo al texto de la fórmula
            'Sheet1.Cells(cCounter+nVar+3, 2+nzCounter).Value devuelve el valor de la celda (la variable)
            Texto = Texto + "+B" + Replace(Str(Sheet1.Cells(cCounter + nVar + 3, 2 + nzCounter).Value + 1), " ", "")
        Next
        'escribo la fórmula
        ActiveSheet.Cells(cCounter + 1, 6).Formula = Texto
    Next
    'creo rango para las restricciones
    Dim Constraints As Range
    Set Constraints = Range(ActiveSheet.Cells(2, 6), ActiveSheet.Cells(1 + nConstr, 6))
    'añado restricciones
    CFXaddConstraint Constraint:=Constraints, Lb:=1
End Sub

```

Introducción

Pros:

- Es la forma recomendada por IBM.
- Puede integrarse de forma “profesional” (a costa de programar un poco).
- Proporciona un modelizador.

Contras:

- La curva de aprendizaje es un poco mayor.
- El trabajo se hace fuera de Excel.

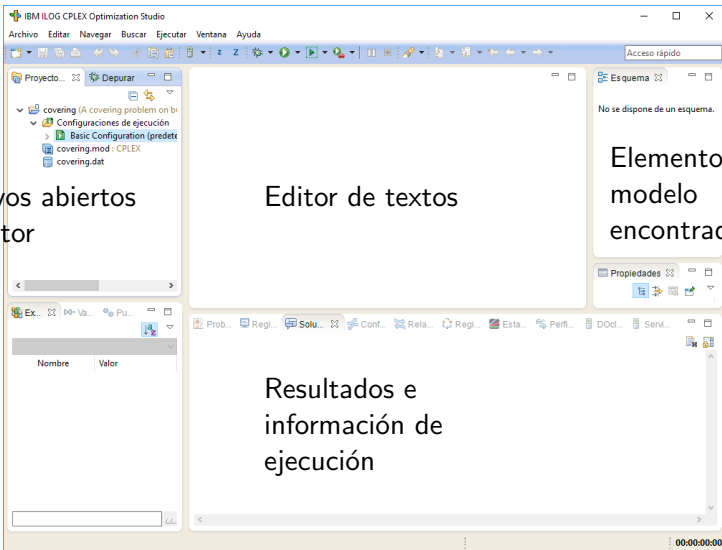
Modus Operandi

Archivos abiertos
en editor

Editor de textos

Elementos del
modelo
encontrados

Resultados e
información de
ejecución



Entorno de trabajo

- OPL estructura la información en dos partes: **Modelo y Datos**.
- La parte de modelo se divide en tres partes: lectura de datos, modelo matemático (las ecuaciones) y escritura de datos
- La fuente de datos para la lectura y escritura son un archivo de texto (por defecto)
- Haremos que el archivo de texto sea una hoja de Excel.

Podemos descargar un zip con todos los archivos necesarios de <http://goo.gl/oSWy1M>

Código del modelo (independiente del archivo)

```
6  int nPeriodos = ...;
7  int cl=...;
8  int cs=...;
9  int d[1..nPeriodos]=...;
10
11 dvar float+ s[0..nPeriodos];
12 dvar float+ x[1..nPeriodos];
13 dvar boolean y[1..nPeriodos];
14 dvar float+ obj;
15 minimize obj;
16 subject to {
17     calculoObj: obj ==
18         sum(t in 1..nPeriodos) cl * y[t] + sum(t in 1..nPeriodos) cs * s[t];
19     initStock: s[0]==0;
20     flujo: forall( t in 1..nPeriodos) x[t]+s[t-1]==d[t]+s[t];
21     pedido: forall(t in 1..nPeriodos) (sum(tt in t..nPeriodos)d[tt]) * y[t]>= x[t];
22 }
23
```

Entrada y salida del modelo

```
6 SheetConnection sheetData("uls.xls");  
7  
8 nPeriodos from SheetRead(sheetData,"nPeriodos");  
9 cl from SheetRead(sheetData,"costeLanzamiento");  
10 cs from SheetRead(sheetData,"costePosesion");  
11 d from SheetRead(sheetData,"demanda");  
12  
13 s to SheetWrite(sheetData,"variableS");  
14 x to SheetWrite(sheetData,"variableX");  
15 y to SheetWrite(sheetData,"variableY");  
16 obj to SheetWrite(sheetData,"objetivo");
```

Vayan con cuidado, es necesario que el archivo Excel esté cerrado

Nombres en Excel

Hemos tenido que poner nombres a las casillas en Excel. Esto se puede conseguir usando el botón derecho y escogiendo el nombre. Así hemos definido las zonas para la solución y la lectura de datos:

- nPeriodos
- costeLanzamiento
- costePosesion
- demanda
- variableX, variableY, variableS, objetivo

Ejercicio

Hagamos el mismo proceso con el problema de cubrimiento.

Introducción

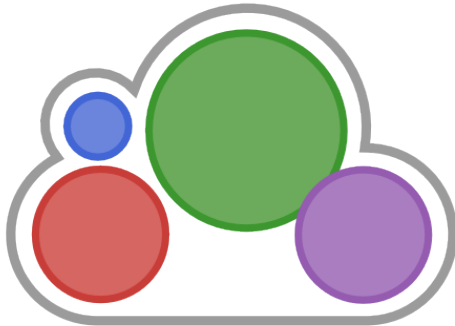
- Personalmente no uso ninguna de las soluciones anteriores.
- Recientemente han aparecido diversas opciones para facilitar el trabajo con modelos matemáticos.
- Mi favorita es Julia/JuMP. No es exactamente una forma de trabajar con Excel y Cplex pero puede integrarse con las anteriores.

Pros:

- Lenguaje moderno completo.
- Un modelizador que integra directamente con el lenguaje (importante para usar algoritmos “complejos” como branch and cut, Dantzig-Wolfe,...).
- “Solver agnostic”.
- Podemos practicar sin necesidad de tener instalado un solver.

Online

Podemos ejecutar Julia desde una página web utilizando www.juliabox.com y una cuenta google (o ucn.cl)



Offline

- Podemos ejecutar Julia de varias maneras (línea de comando, Jupyter local, Juno ...)
- Podemos ejecutar diferentes solvers de programación lineal (Cplex, Gurobi, GLPK) cambiando únicamente una línea del archivo.
- Los primeros ejemplos lo vamos a hacer desde Jupyter local. Descargar un ejemplo simple de goo.gl/4yFCwM

Modelo “as-is”

```
#creamos un modelo
m = Model(solver = CplexSolver())
m = Model(solver = GLPKSolverMIP())
m = Model(solver = GLPKSolverLP())

#y sus variables
@variable(m, 0 <= x <= 200 )
@variable(m, 0 <= y <= 100 )

#función objetivo a minimizar
@objective(m, Max, 250*x + 400*y )

#restricción
@constraint(m, 3*x + 5*y <= 750.0 )

#escribimos el modelo
print(m)

#resolvemos
status = solve(m)

#mostramos resultados
println("Status: ", status, " Objective value: ", getobjectivevalue(m))
println("x = ", getvalue(x))
println("y = ", getvalue(y))
```

Modelo inventarios

Descargar de goo.gl/DzTY8v

```

modelo=Model(solver = CplexSolver())
@variable(modelo, x[1:T]>= 0)
@variable(modelo, y[1:T],Bin)
@variable(modelo, s[0:T]>=0)

#función objetivo a minimizar
@objective(modelo, Min, sum(cs*s[t] for t in 1:T)+sum(c1*y[t] for t in 1:T))

for t in 1:T
    @constraint(modelo,s[t-1]+x[t]==d[t]+s[t])
end
for t in 1:T
    @constraint(modelo,x[t]<=M*y[t])
end
@constraint(modelo,s[0]==0)

#escribimos el modelo
print(modelo)

#resolvemos
status = solve(modelo)

#mostramos resultados
println("**** Status: ",status," Objective value: ", getobjectivevalue(modelo))
println("**** x = ", getvalue(x))
println("**** y = ", getvalue(y))
println("**** s = ", getvalue(s))

```

Integración con Excel

Podemos trabajar directamente desde Excel para ello:

- Escribir la rutina en un archivo .jl
- Crear la macro que llama el archivo .jl
- Leer el archivo generado por julia y guardarlo en la hoja de Excel

El paso más complejo es personalizar las rutas hacia y desde los archivos. El problema es algo tedioso pero no difícil. Por si acaso el excel (goo.gl/UHE9Dn) y el archivo julia (goo.gl/gj9zZw)