Knapsack

Andrés D. González

Assistant Professor
School of Industrial and Systems Engineering, The University of Oklahoma

ISE 4623/5023: Deterministic Systems Models / Systems Optimization

The University of Oklahoma, Norman, OK, USA

Knapsack Problem

Sets:

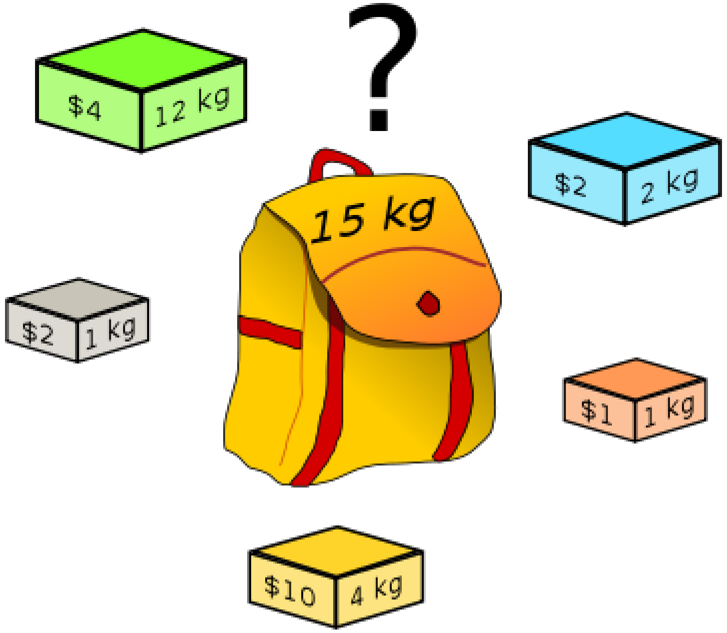
- O = set of *objects* $\{1,2,\ldots,n\}$

Parameters:

- w_o = weight of object $o \in O$
- v_o = value of the object
- k = capacity of the knapsack

Objective:

- Maximize the total value of the objects in the knapsack



Knapsack Problem

Sets:

- O = set of *objects* $\{1,2,\dots,n\}$

Parameters:

- w_o = weight of object $o \in O$
- v_o = value of the object
- k = capacity of the knapsack

Decision variables:

x_o= binary variable that indicates if object $o \in O$ is packed in the knapsack

Objective function:

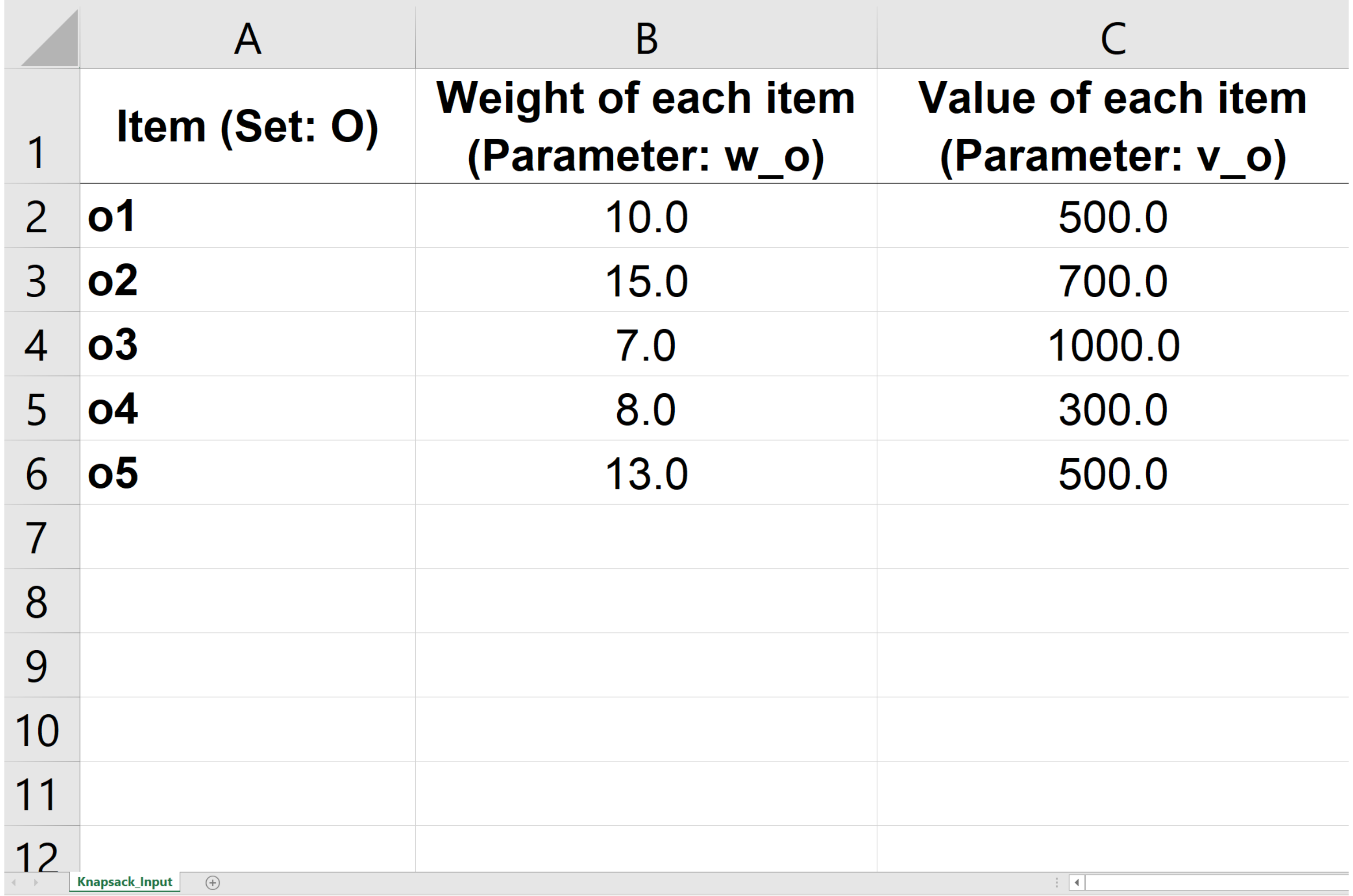
$$\min \sum_{o \in O} v_o x_o$$

Constraints:

$$\sum_{o \in O} w_o x_o \leq k$$

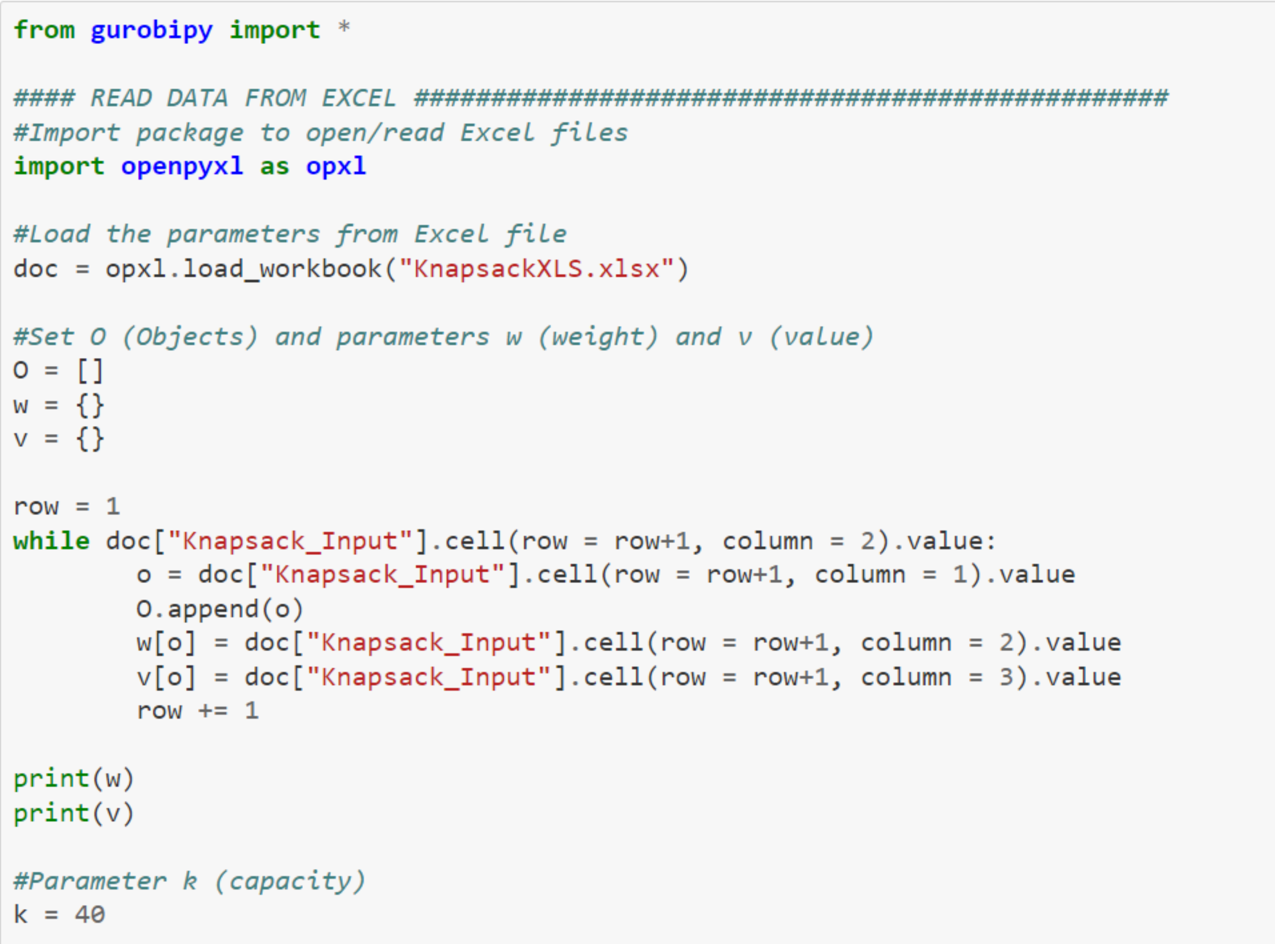
$$x_o \in \{0,1\} \quad ,\forall o \in O$$

Input data



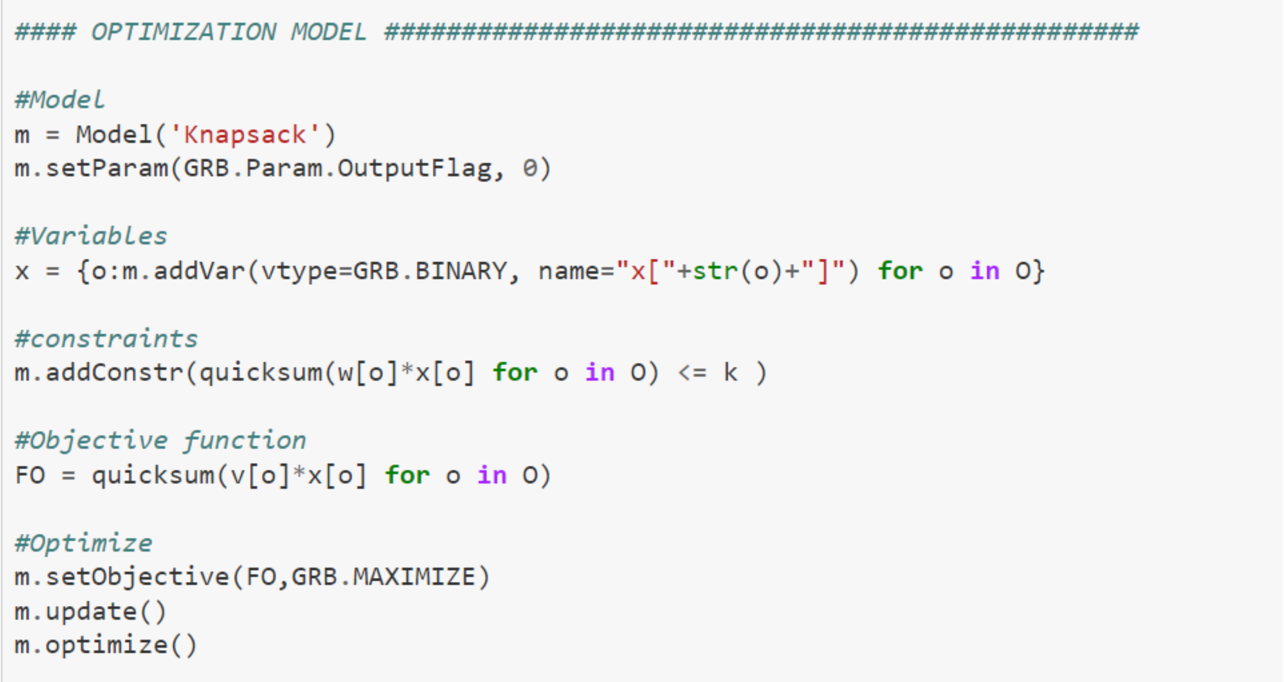
	A	B	C
1	Item (Set: O)	Weight of each item (Parameter: w_o)	Value of each item (Parameter: v_o)
2	o1	10.0	500.0
3	o2	15.0	700.0
4	o3	7.0	1000.0
5	o4	8.0	300.0
6	o5	13.0	500.0
7			
8			
9			
10			
11			
12			

Reading data from Excel



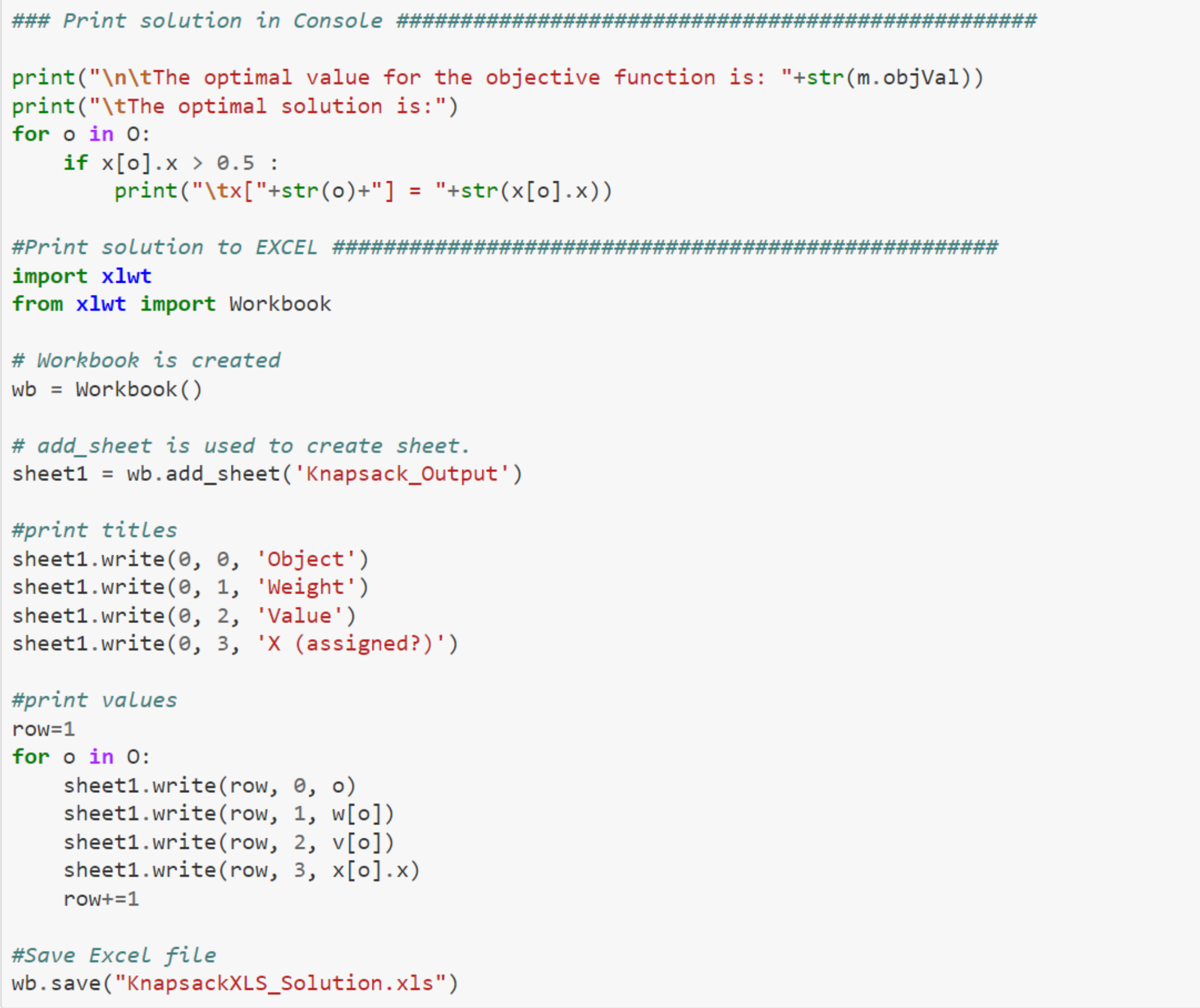
```
from gurobipy import *

#### READ DATA FROM EXCEL #################################################
#Import package to open/read Excel files
import openpyxl as opxl

#Load the parameters from Excel file
doc = opxl.load_workbook("KnapsackXLS.xlsx")

#Set O (Objects) and parameters w (weight) and v (value)
O = []
w = {}
v = {}

row = 1
while doc["Knapsack_Input"].cell(row = row+1, column = 2).value:
        o = doc["Knapsack_Input"].cell(row = row+1, column = 1).value
        O.append(o)
        w[o] = doc["Knapsack_Input"].cell(row = row+1, column = 2).value
        v[o] = doc["Knapsack_Input"].cell(row = row+1, column = 3).value
        row += 1

print(w)
print(v)

#Parameter k (capacity)
k = 40
```

Building and solving the optimization model in Python/Gurobi



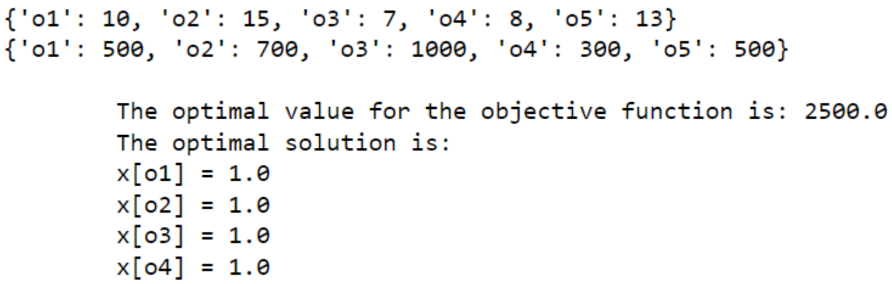
```
#### OPTIMIZATION MODEL #################################################

#Model
m = Model('Knapsack')
m.setParam(GRB.Param.OutputFlag, 0)

#Variables
x = {o:m.addVar(vtype=GRB.BINARY, name="x["+str(o)+"]") for o in O}

#constraints
m.addConstr(quicksum(w[o]*x[o] for o in O) <= k )

#Objective function
FO = quicksum(v[o]*x[o] for o in O)

#Optimize
m.setObjective(FO,GRB.MAXIMIZE)
m.update()
m.optimize()
```

Printing the solution to console and Excel

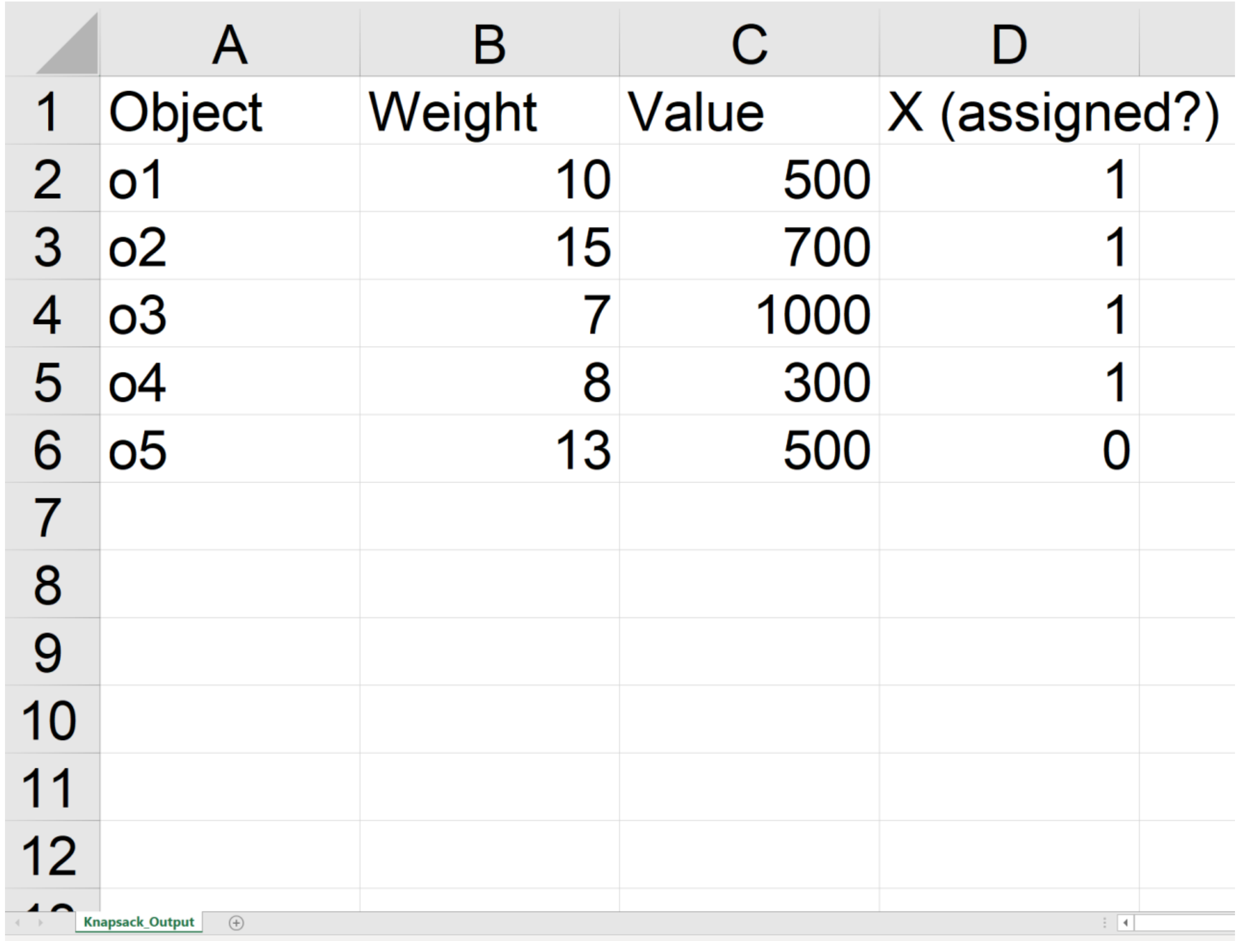


```
### Print solution in Console ##################################################

print("\n\tThe optimal value for the objective function is: "+str(m.objVal))
print("\tThe optimal solution is:")
for o in O:
    if x[o].x > 0.5 :
        print("\tx["+str(o)+"] = "+str(x[o].x))

#Print solution to EXCEL ####################################################
import xlwt
from xlwt import Workbook

# Workbook is created
wb = Workbook()

# add_sheet is used to create sheet.
sheet1 = wb.add_sheet('Knapsack_Output')

#print titles
sheet1.write(0, 0, 'Object')
sheet1.write(0, 1, 'Weight')
sheet1.write(0, 2, 'Value')
sheet1.write(0, 3, 'X (assigned?)')

#print values
row=1
for o in O:
    sheet1.write(row, 0, o)
    sheet1.write(row, 1, w[o])
    sheet1.write(row, 2, v[o])
    sheet1.write(row, 3, x[o].x)
    row+=1

#Save Excel file
wb.save("KnapsackXLS_Solution.xls")
```

Output from console and Excel



```
{'o1': 10, 'o2': 15, 'o3': 7, 'o4': 8, 'o5': 13}
{'o1': 500, 'o2': 700, 'o3': 1000, 'o4': 300, 'o5': 500}

        The optimal value for the objective function is: 2500.0
        The optimal solution is:
        x[o1] = 1.0
        x[o2] = 1.0
        x[o3] = 1.0
        x[o4] = 1.0
```



	A	B	C	D
1	Object	Weight	Value	X (assigned?)
2	o1	10	500	1
3	o2	15	700	1
4	o3	7	1000	1
5	o4	8	300	1
6	o5	13	500	0
7				
8				
9				
10				
11				
12				

THANK YOU

QUESTIONS?

Andrés D. González | andres.gonzalez@ou.edu