

ED2

February 5, 2017

```
In [1]: from pyomo.environ import *
import os
model = AbstractModel()
solverexe = "gurobi"
dirlsolver = r"C:\Users\ch9fod\Documents\GitHub\ED\solvers"
datafile = "data2.dat"

In [2]: #set
model.G = Set()

In [3]: #parameters
model.a = Param(model.G)
model.b = Param(model.G)
model.Pmin = Param(model.G)
model.Pmax = Param(model.G)
model.d = Param(model.G)
model.e = Param(model.G)
model.f = Param(model.G)
#lone parameter
model.D = Param()

In [4]: #variables
model.P = Var(model.G)

In [5]: #constraints
def maxp(model,i):
    return model.P[i] <= model.Pmax[i]
model.maxprod = Constraint(model.G, rule = maxp)

def minp(model,i):
    return model.P[i] >= model.Pmin[i]
model.minprod = Constraint(model.G, rule = minp)

def demand_r(model,i):
    return model.D == sum(model.P[i] for i in model.G)
model.demand = Constraint(model.G, rule = demand_r)

In [6]: #objective
def cost_rule(model):
```

```

        return sum(model.a[i]*model.P[i] +
                    0.5*model.b[i]*model.P[i]**2 +
                    10*model.d[i] + 10*model.e[i]*model.P[i] +
                    10*model.f[i]*model.P[i]**2 for i in model.G)
#default is to minimize
model.OBJ = Objective(rule=cost_rule)

In [7]: if solverexe == "gurobi":
        solver = SolverFactory(solverexe)
    else:
        solver = SolverFactory(solverexe,
                                executable=os.path.join(dirsolver, solverexe))
    instance = model.create_instance(datafile)
    instance.dual = Suffix(direction=Suffix.IMPORT)
    results = solver.solve(instance)

```

```
In [8]: instance.display()
```

Model unknown

Variables:

```

P : Size=3, Index=G
   Key : Lower : Value : Upper : Fixed : Stale : Domain
   1 :  None : 200.0 :  None : False : False : Reals
   2 :  None : 150.0 :  None : False : False : Reals
   3 :  None : 250.0 :  None : False : False : Reals

```

Objectives:

```

OBJ : Size=1, Index=None, Active=True
     Key : Active : Value
     None : True : 992500.0

```

Constraints:

```

maxprod : Size=3
         Key : Lower : Body : Upper
         1 :  None : 200.0 : 400.0
         2 :  None : 150.0 : 300.0
         3 :  None : 250.0 : 250.0
minprod : Size=3
         Key : Lower : Body : Upper
         1 :  20.0 : 200.0 :  None
         2 :  20.0 : 150.0 :  None
         3 :  30.0 : 250.0 :  None
demand  : Size=3
         Key : Lower : Body : Upper
         1 :   600 : 600.0 :   600
         2 :   600 : 600.0 :   600
         3 :   600 : 600.0 :   600

```

```
In [9]: p = [0, 0, 0, 0]
        for i in range(3):
            p[i+1] = value(instance.P[i+1])
```

```
In [10]: print ("Total Emissions (tonCO2/MWh)")
          x = y = 0
          for i in range(3):
              y = (instance.d[i+1] + instance.e[i+1]*p[i+1] +
                  instance.f[i+1]*p[i+1]**2)
              x = x + y
              print(i+1, y)
          print ("Total", x)
```

```
Total Emissions (tonCO2/MWh)
1 40700.0
2 30525.0
3 25412.5
Total 96637.5
```

```
In [11]: print ("Duals")
          from pyomo.core import Constraint
          for c in instance.component_objects(Constraint, active=True):
              print ("    Constraint", c)
              cobject = getattr(instance, str(c))
              for index in cobject:
                  print ("        ", index, instance.dual[cobject[index]])
```

```
Duals
Constraint maxprod
  1 -0.0
  2 -0.0
  3 -955.0
Constraint minprod
  1 0.0
  2 0.0
  3 0.0
Constraint demand
  1 2070.0
  2 -0.0
  3 -0.0
```