ED1

February 5, 2017

```
In [1]: from pyomo.environ import *
        import os
        model = AbstractModel()
        solverexe = "gurobi"
        dirsolver = r"C:\Users\ch9fod\Documents\GitHub\ED\solvers"
        datafile = "data1.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)
        #added just to calculate emissions
        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 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)
        #instance.solutions.load_from(results)
        #print(results)
```

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

```
Model unknown

  Variables:
    P : Size=3, Index=G
        Key : Lower : Value          : Upper : Fixed : Stale : Domain
          1 :  None : 399.999999999 :  None : False : False :  Reals
          2 :  None : 169.999999997 :  None : False : False :  Reals
          3 :  None : 30.0000000039 :  None : False : False :  Reals

  Objectives:
    OBJ : Size=1, Index=None, Active=True
        Key  : Active : Value
        None :   True : 18985.0000000134

  Constraints:
    maxprod : Size=3
        Key : Lower : Body          : Upper
          1 :  None : 399.999999999 : 400.0
          2 :  None : 169.999999997 : 300.0
          3 :  None : 30.0000000039 : 250.0
    minprod : Size=3
        Key : Lower : Body          : Upper
          1 :  20.0 : 399.999999999 :  None
          2 :  20.0 : 169.999999997 :  None
          3 :  30.0 : 30.0000000039 :  None
    demand : Size=3
        Key : Lower : Body              : Upper
          1 :   600 : 599.9999999999001 :   600
          2 :   600 : 599.9999999999001 :   600
          3 :   600 : 599.9999999999001 :   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 81899.999999792
2 34588.9999993898
3 3104.50000039117
Total 119593.49999957297
```

```
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 -1.9999999997
       2 -0.0
       3 -0.0
   Constraint minprod
       1 0.0
       2 0.0
       3 4.0000000011
   Constraint demand
       1 41.9999999997
       2 -0.0
       3 -0.0
```