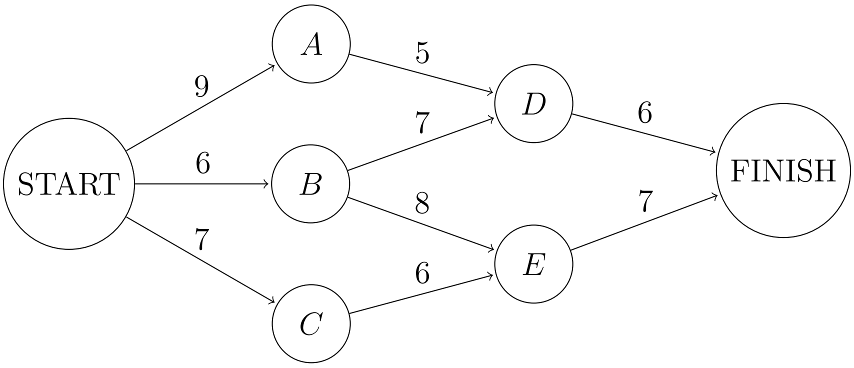
Operations Research

Final Exam

December 17, 2016

1. Minimize
$$z = x^2 - 2\,x + y^4 - 2\,y^2 + 2$$
subject to
$$\begin{aligned} 2\,x &\leq 4\,, \\ x + 2\,y &\leq 6\,, \\ x, y &\geq 0\,. \end{aligned}$$
(Be aware that for nonlinear programs, the location of the minimum may be anywhere on an edge or in the interior the feasible region!) (20)

2. Consider the following network where the number on each arc represents the actual distance.



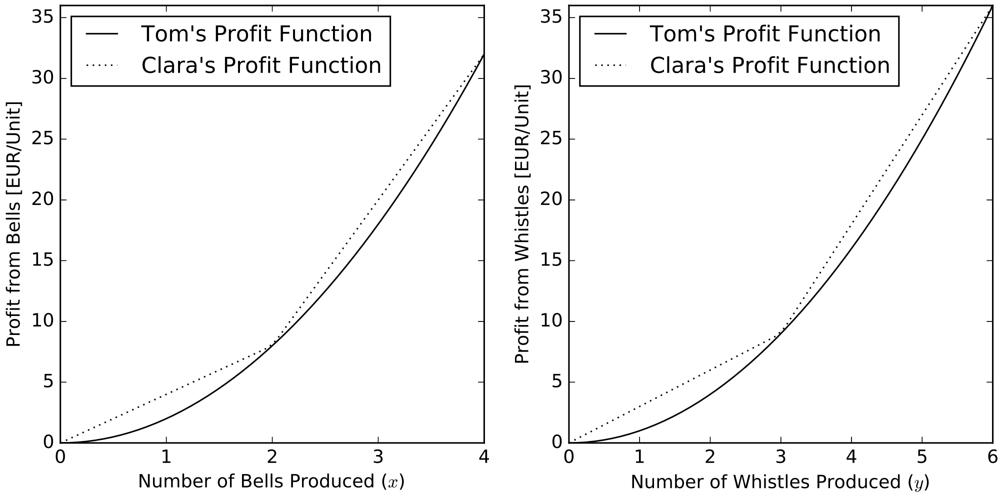
   (a) Solve the shortest path problem for this network. Use dynamic programming and show the details of your work.

   (b) Formulate the shortest path problem as an equivalent linear programming problem. (You can use the concrete network above or, alternatively, consider an abstract shortest path problem.)

   (c) Which solver should be used when solving shortest path problems in their LP formulation? Explain!

(10+5+5)

3. *Gadgets & More Inc.* produces bells and whistles. Their products are so popular that the more they sell, the more customers are willing to pay. Tom and Clara both work in production planning and each, independently, derived a function for the profit from each product given the number of units produced.



As the factory has limited production capacity, they need to solve an optimization problem for the total profit as a function of the number of bells x and whistles y produced.

Tom's and Clara's Pyomo implementations are attached.

(a) State Tom's model in mathematical notation. Make sure you list all constraints.

(b) State Clara's model in mathematical notation. Again, make sure you list all constraints.

(c) Clara's model is a linear (integer) programming problem. Can you provide an equivalent nonlinear programming formulation that looks more like Tom's?

(d) Compare the two solutions. How are they different? Do you think the difference is caused by the use of slightly different profit functions? Explain!

(e) Can you indicate a fix for the approach which gives the inferior solution?

(f) State their advantages and disadvantages of the two approaches and express a preference. Explain your choice.

(5+5+5+5+5+5)

4. The Hit-and-Miss Manufacturing Company produces integrated electronic circuits in batches of 2. It is known that there are "good" batches, where each circuit has an independent 10% chance of being defective and there are "bad" batches where each circuit has an independent 80% chance of being defective. Good batches are produced $\frac{2}{3}$ of the time, bad batches are produced $\frac{1}{3}$ of the time.

(a) What is the probability that a randomly selected circuit is defective?

(b) Given that one circuit is tested and found functional, what is the probability that the other circuit *from the same batch* is also functional?

(c) Testing is expensive at €25 per circuit. The test is 100% reliable. If a defective unit is passed on to the next stage in the manufacturing process, a loss of €100 is incurred. It is possible to forgo testing completely, to test only one unit from each batch, or, possibly depending on the outcome of the first test, also test the other unit from each batch. Determine the most economical testing schedule. As part of your answer, draw a complete decision tree for this problem.

(5+5+10)

5. A factory produces items continuously at a rate of 100 units per day. From time to time, a truck takes finished units to a major distributor at a cost of €500 independent of the number of items shipped. Finished units incur a holding cost of €0.10 per unit per day until they are shipped. Derive a formula for the most economical shipping batch size Q. (10)

Tom's solution for the *Gadgets & More Inc.* problem:

```
In [1]: from pyomo.environ import *
        from pyomo.opt import *
        opt = solvers.SolverFactory("ipopt")
        model = ConcreteModel()
```

```
In [2]: model.x = Var(within=NonNegativeReals)
        model.y = Var(within=NonNegativeReals)

        model.c = Constraint(expr = 4*model.y + 6*model.x <= 24)

        model.z = Objective(expr = 2*model.x**2 + model.y**2, sense=maximize)
```

```
In [3]: results = opt.solve(model)
        model.x.get_values()[None]

Out[3]: 4.000000046353438
```

```
In [4]: model.y.get_values()[None]

Out[4]: 0.0
```

```
In [5]: model.z.expr()

Out[5]: 32.00000074165502
```

Clara's solution for the *Gadgets & More Inc.* problem:

In [1]:

```
from pyomo.environ import *
from pyomo.opt import *
opt=solvers.SolverFactory("glpk")
model=ConcreteModel()
```

In [2]:

```
model.x1=Var(within=NonNegativeIntegers)
model.x2=Var(within=NonNegativeIntegers)
model.y1=Var(within=NonNegativeIntegers)
model.y2=Var(within=NonNegativeIntegers)
model.b1=Var(within=Boolean)
model.b2=Var(within=Boolean)

model.c1 = Constraint(expr = 6*model.x1 + 6*model.x2 +
                             4*model.y1 + 4*model.y2 <= 24)
model.c2 = Constraint(expr = model.x1 <= 2)
model.c3 = Constraint(expr = model.y1 <= 3)
model.c4 = Constraint(expr = model.b1*2 <= model.x1)
model.c5 = Constraint(expr = model.b2*3 <= model.y1)
model.c6 = Constraint(expr = model.x2 <= model.b1*2)
model.c7 = Constraint(expr = model.y2 <= model.b2*3)

model.z = Objective(expr = 4*model.x1 + 12*model.x2 +
                           3*model.y1 + 9*model.y2,
                    sense=maximize)
```

In [3]:

```
results = opt.solve(model)
model.x1.get_values()[None] + model.x2.get_values()[None]
```

Out[3]:

```
0.0
```

In [4]:

```
model.y1.get_values()[None] + model.y2.get_values()[None]
```

Out[4]:

```
6.0
```

In [5]:

```
model.z.expr()
```

Out[5]:

```
36.0
```