1. Minimize

$$z = x^2 - 2x + y^4 - 2y^2 + 2$$

subject to

$$2x \leq 4,$$
$$x + 2y \leq 6.$$
$$x, y \geq 0.$$

(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)

We first look for an interior minimum:

$$\frac{\partial z}{\partial x} = 2x - 2 = 0 \quad \Rightarrow \quad x = 1$$

$$\frac{\partial z}{\partial y} = 4y^3 - 4y = 4y(y^2 - 1) \quad \Rightarrow \quad y = 0 \text{ or } y = \pm 1$$

@ $(x,y) = (1,0)$: This point is clearly in the feasible region and

$z = 1 - 2 + 0 - 0 + 2 = 1$

@ $(x,y) = (1,1)$: This point is also clearly in the feasible region and

$z = 1 - 2 + 1 - 2 + 2 = 0$ (*)

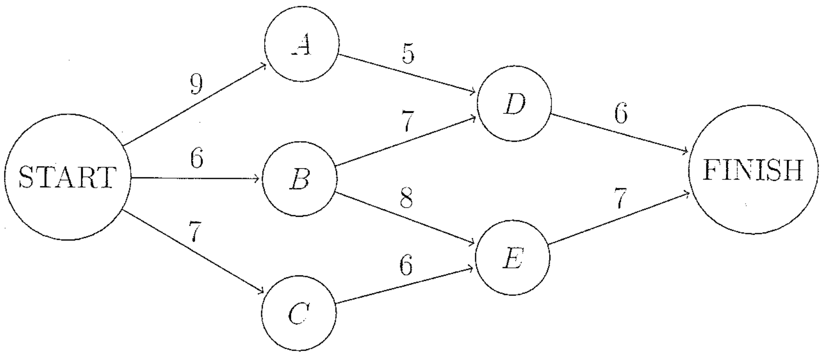
@ $(x,y) = (1,-1)$: This point is not in the feasible region as $y < 0$;

$z = 1 - 2 + 1 - 2 + 2 = 0$ (*)

As $z \to \infty$ as $x, y \to \infty$, the points (*) are a global minimum for the unconstrained problem. As the first of these is within the feasible region, this point is also a global minimum for the constrained problem.

Alternative solution: Write $z = (x-1)^2 + (y^2-1)^2$ and argue from there...

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)

(a) Stage 3: $f_3^*(D) = 6$, $f_3^*(E) = 7$

Stage 2:

x_2	$f_2(x_2, D)$	$f_2(x_2, E)$	f_2^*	x_3^*
A	5+6	-	11	D
B	7+6	8+7	13	D
C	-	6+7	13	E

Stage 1:

f_1(START, A)	f_1(START, B)	f_1(START, C)	f_1^*	x_2^*
9+11	6+13	7+13	19	B

So shortest path is START - B - D - FINISH with length 19.

(b) minimize $\sum_{(i,j)\text{ edge}} f_{ij}\, d_{ij}$

subject to $\sum_{(i,k)\text{ edge}} f_{ik} = \sum_{(k,j)\text{ edge}} f_{kj}$

for every transshipment node k,

$$\sum_{(\text{START},j)\text{ edge}} f_{\text{START},j} = 1 ,$$

$$\sum_{(j,\text{FINISH})\text{ edge}} f_{j,\text{FINISH}} = 1 ,$$

$$f_{ij} \geq 0$$

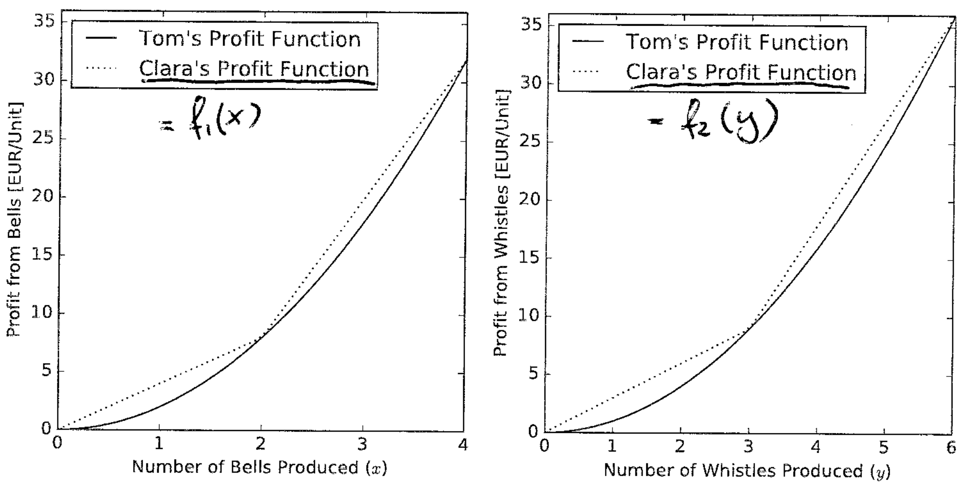
Remark: Here the transshipment nodes are $\{A, B, C, D, E\}$,

edges are $\{(\text{START}, A), (\text{START}, B), (\text{START}, C), (A,D), (B,D), (B,E), (C,E), (D,\text{FINISH}), (E,\text{FINISH})\}$

$d_{\text{START},A} = 9$, $d_{\text{START},B} = 6$, $d_{\text{START},C} = 7$, $d_{AD} = 5$, $d_{BD} = 7$, $d_{BE} = 8$, $d_{CE} = 6$, $d_{D,\text{FINISH}} = 6$, $d_{E,\text{FINISH}} = 7$.

(c) You should use a simplex-based solver (e.g. glpk) or a special network solver, but not an interior point method as it will not, in general, guarantee integrality of the answer. (The shortest path length will still be correct within error bounds, but you will not generally get a shortest path as those edges where $f_{ij} = 1$.)

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)

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
```

(a) maximize $z = 2x^2 + y^2$

subject to $6x + 4y \leq 24$,

$x, y \geq 0$.

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
```

(b) maximize $z = 4x_1 + 12x_2 + 3y_1 + 9y_2$

subject to $6(x_1+x_2) + 4(y_1+y_2) \le 24$

$x_1 \le 2 \qquad y_1 \le 3$

$2b_1 \le x_1 \qquad 3b_2 \le y_1$

$x_2 \le 2b_1 \qquad y_2 \le 3b_2$

$b_1, b_2 \in \{0,1\}$

x_1, x_2, y_1, y_2 are non-negative integers

(c) maximize $z = f_1(x) + f_2(y)$ where f_1, f_2 are the dashed functions shown in the graphs above

subject to $6x + 4y \le 24$

x, y are non-negative integers.

(d) Tom obtains $x=4$, $y=0$, $z=32$

Clara obtains $x=0$, $y=6$, $z=36$

Both have the same feasible region. So you can plug Clara's values for x and y into Tom's objective function:

$$z = 2 \cdot 0^2 + 6^2 = 36 \quad !$$

So Tom has NOT found the global maximum.

Explanation: Tom is using an interior point algorithm which can get stuck in a local maximum (and it does here as is plain obvious!).

In particular, the difference between the solutions cannot be explained by the slightly different profit functions.

(e) Try different starting values for the interior point optimizer, or randomize the starting value and run an ensemble of solver runs. That does not guarantee finding the global maximum, but makes it more likely.

(f) Tom's advantage: Simple, will run fast even for bigger problems.

Tom's disadvantage: Does not guarantee global optimum.

Clara's advantage: Global optimum guaranteed by solver.

Clara's disadvantage: It's a combinatorial optimisation problem where computation time can increase badly as the problem size gets larger; coding more complex. Preference depends on problem scale, but Clara's approach is more robust.

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)

To simplify notation, let

B = event that batch is bad

G = event that batch is good

D = event that device is defective

F = event that device is functional

(a) $P(D) = P(D|G)\,P(G) + P(D|B)\,P(B)$

$= \frac{1}{10}\frac{2}{3} + \frac{8}{10}\frac{1}{3} = \frac{1}{3}$ $\Rightarrow P(F) = \frac{2}{3}$

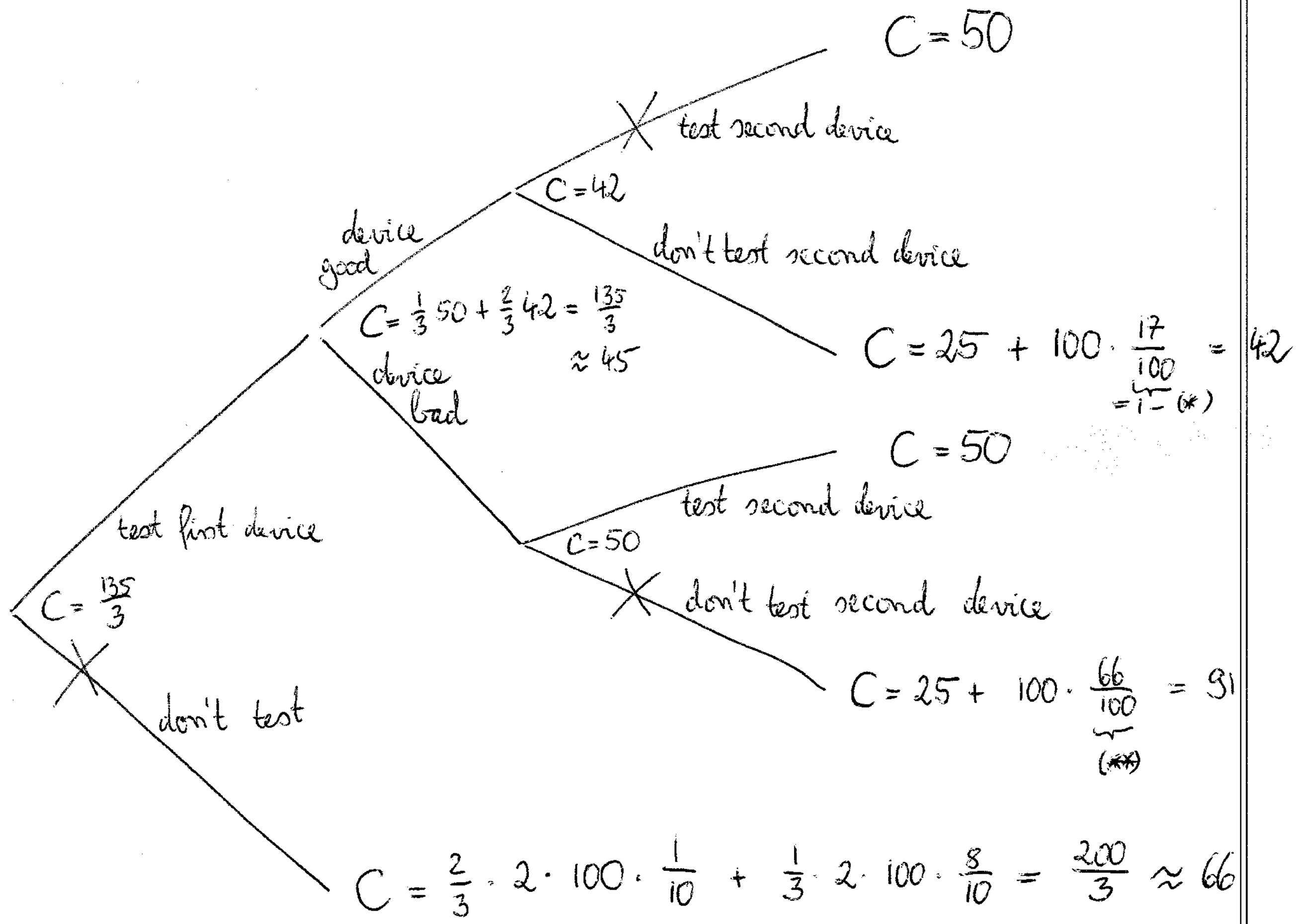
(b) $P(G|F) = \dfrac{P(F|G)\,P(G)}{P(F)}$ (Bayes' rule)

$= \dfrac{\frac{9}{10}\frac{2}{3}}{\frac{2}{3}} = \frac{9}{10}$

So for the second device from the same batch:

$P(F) = P(F|G)\,P(G) + P(F|B)P(B) = \frac{9}{10}\frac{9}{10} + \frac{2}{10}\frac{1}{10} = \frac{83}{100}$ (*)

(c)



Note: as in (b), after testing the first device,

$$P(B|D) = \frac{P(D|B)\ P(B)}{P(D)} = \frac{\frac{8}{10}\frac{1}{3}}{\frac{1}{3}} = \frac{8}{10}$$

Then for the second device of the same batch:

$$P(D) = P(D|G)\,P(G) + P(D|B)\,P(B)$$
$$= \frac{1}{10}\frac{2}{10} + \frac{8}{10} \cdot \frac{8}{10} = \frac{66}{100} \qquad (**)$$

Thus, it is most economical to proceed as follows: Test the first device; if good, use second device without testing. If bad, also test the second device.

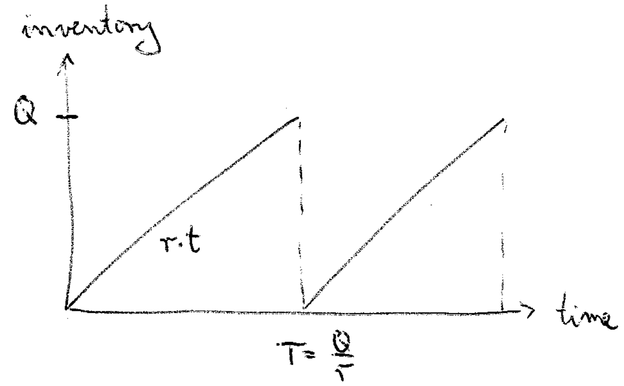
<u>Note:</u> The implicit assumption here is that the devices are <u>valuable</u>, i.e. no good device is to be tossed away. If the devices are sufficiently cheap, it will be more economical to discard the batch if one device fails the test.

$r = 100$ $\quad$ $K = 500$

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 $= h$ per unit per day until they are shipped. Derive a formula for the most economical shipping batch size Q. (10)

Note: The situation is precisely that of the simple EOQ model without planned shortages. Here, the items are put into storage rather than used at a continuous rate, but that will not change the formulas.

Solution:



Cost per cycle: $C_{cycle} = K + \frac{1}{2} h Q T$

Cost per time: $C = \dfrac{C_{cycle}}{T} = \dfrac{Kr}{Q} + \frac{1}{2} h Q$

$$\frac{dC}{dQ} = -\frac{Kr}{Q^2} + \frac{1}{2} h$$

Setting this to zero gives $\dfrac{Kr}{(Q^*)^2} = \frac{1}{2} h$

$$\Rightarrow Q^* = \sqrt{\frac{2Kr}{h}}$$

$$= \sqrt{\frac{2 \cdot 500 \cdot 100}{\frac{1}{10}}} = 1000$$