9681 (a) x = (5, 15/2, 0, 0), optimal solution is 25/2

1 (b) (i)

Any tableau that at optimal has a BV = 0; e.g.

| x\_1 x\_2 x\_3 |

Z 0 -1 0 1

x\_1 1 0 0 1

x\_3 0 1 1 0

so optimality can be at either I = {1, 2} or I = {1, 3}, with solution (1, 0, 0) with x\_1 = 1 enforced

1 (b) (ii)

Let leaving var be xp and entering xq. This implies ypq > 0 & \betaq > 0.  
ypp is 1 by definition.  
(All prime vars denote after leaving)  
\beta\_p’ = \beta\_p - \beta\_q \* (y\_{pp} / y\_{pq}) = \beta\_p - \beta\_q / y\_{pq}

Now we need to show \beta\_p’ < 0.   
This is straightforward as \beta\_p = 0 because x\_p was basic before, in addition to \beta\_q > 0 & y\_{pq} > 0.

In words:

We choose an entering variable only if the coeff in **the** objective row is +ve (so as to reduce the objective value by increasing the value of the variable from 0 to >0).  
When a variable leaves, the coeff in the objective row of that var becomes negative from zero -> can’t reenter.

2

(a)

x1: # of tables produced in period 1

x2: # of consoles produced in period 1

x3: # of tables sold in period 1

x4: # of consoles sold in period 1

x5: # of tables sold in period 2

x6: # of consoles sold in period 2

Max 300(x3+x5) + 400(x4+x6) // maximize the total profit across period 1 and 2

x3 <= 100 \\ table sales for period 1 is less than demand

x4 <= 200 \\ console sales for period 1 is less than demand

x1 >= x3 \\ we produce enough table for our sales at period 1

x2 >= x4 \\ we produce enough console for our sales at period 1

x1 - x3 <= 100 \\ we have enough storage space for table

x2 - x4 <= 200 \\ we have enough storage space for console

40x1 + 60x2 <= 20000 \\ we have enough production capacity for period 1

40(x5 - (x1 - x3)) + 60(x6 - (x2 - x4)) <= 20000 \\ we have enough production capacity for period 2 where the difference between sales at period 2 and inventory from period 1 is what we produced at period 2

x5 <= 300 \\ table sales shouldn’t exceed demand at period 2

x6 <= 150 \\ console sales shouldn’t exceed demand at period 2

x5 >= x1 - x3 \\ table storage is empty

x6 >= x2 - x4 \\ console storage is empty

x1, x2, x3, x4, x5, x6 >= 0

2 (b)// **<NB Out of syllabus, but good practice> (Edit by someone else: is it really out of syllabus?)**

//Minimise radii for all satellites

Min sum r\_i m\_i over all i

r\_i >= d\_i for all i

/\*

now consider

| r\_i – r\_j | >= 10 for all i,j in [1,n]

if r\_i < r\_j we need r\_i – r\_j <= - 10

so we can reformulate the constraint as

(r\_i – r\_j >= 10 && r\_i >= r\_j) || (r\_i – r\_j <= -10 && r\_i < r\_j)

\*/lu

r\_i – r\_j >= 10 – M(\delta\_{ij}) for all i, j in [1,n], j > i

r\_i – r\_j >= - M(\delta\_{ij}) for all i, j in [1,n] , j > i

r\_i – r\_j <= -10 + M(1 – \delta\_{ij}) for all i,j in [1,n] , j > i

r\_i – r\_j < M(1 – \delta\_{ij}) for all i,j in [1,n] , j > i

r\_i >= 0, \delta\_{ij} in {0, 1} for all i,j in [1,n] , j > i

// \delta\_{ij} = 0 if r\_i >= r\_j

M is a large int, M > max(d\_i) for all i in [1, n]

3(a) r3c3, 1.35

ii) Yes.

3(b)

(i)

max z’ = b^T y

s.t. A^T y <= c

y free

(ii) m x 1

(iii) the kth entry of y\_D\* will be divided by \mju

because the corresponding column in A^T will be multiplied by mju and in effect the kth entry of y needs to be divided by mju to counter this change

Mathematically:

Let c\_i be column i in A^T

We can see the problem as

(... c\_k …) y\_D = ( … \mju\*c\_k … ) y\_D’  
Where y\_D = ( … y\_k …)T and y\_D’ = ( … y\_k’ … )T

Comparing coeffs of c\_k in the equality, we get  
y\_k = \mju y\_k’ and finally  
y\_k’ = y\_k / \mju.

(iv)

so in A\_T now, the rth col is rth col added by \mju times the kth col

Let c\_i be column i in A^T

We can see the problem as  
  
( … c\_k … c\_r … ) y\_D = ( … c\_k ... c\_r+\mju\*c\_k … ) y\_D’

Where y\_D = ( … y\_k … y\_r … )T and y\_D’ = ( … y\_k’ … y\_r’ …)T  
We can “compare constants”, as c\_k and c\_r are equivalent, in other words we equate the coeff of c\_k and c\_r in the equality above

c\_k: y\_k = y\_k’ +\mju y\_r’ : (1)  
c\_r: y\_r = y\_r’ : (2)

So from (2) the rth value is same, but kth value is now, from (1),  
y\_k’ = y\_k - \mju y\_r

4 (a)

x\_1 = 30y\_1 + 60y\_2, y\_1 + y\_2 <= 1, y\_1 & y\_2 in {0, 1}

x\_2 = 15y\_3 + 30y\_4, y\_3 + y\_4 <= 1, y\_3 & y\_4 in {0, 1}

(b)

First write down all the minimal cover cut0s

x\_1 + x\_3 <= 1

x\_2 + x\_3 <= 1

Solving P\_0 immediately gives optimal at 25, (1, 1, 0).

(c)

(i) no – because z\*\_{IP} is also feasible in LP

(ii) yes – some decision variables might not be able to take integer values

(iii) yes – if z\*\_{LP} also int (i think it’s if X\* also int for all Xi element of X\* in the LP relaxation)

(iv) yes – LP can give better soon

(v) no as in iv

(d) IP tutorial

IP is not NP- hard. Having totally unimodular matrices as constraint allow us to solve the problem in polynomial time.