Trevor Stahl

stahltr@oregonstate.edu

HW6

Problem 1:

USING LINDO

D* is distance from dg to vertex d*, such as dh, da etc.

a)

17 edges so 18 constraints with dg=0

MAX dc

ST

dg = 0

dh – dg <= 3

dd -dg <= 2

da – dh <= 4

db -dh <= 9

dc – db <= 4

de – db <= 10

df – da <= 10

db – da <= 8

da – df <= 5

db – df <= 7

dc – df <= 3

de – df <= 2

dd – dc <= 3

df – dd <= 18

de – dd <= 25

dg – de <= 7

dd – de <= 9

END

https://apps.oregonstate.edu/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1550608450838

File Edit Solve Reports Window Help

```
LP OPTIMUM FOUND AT STEP      2

       OBJECTIVE FUNCTION VALUE

       1)      16.00000

 VARIABLE        VALUE          REDUCED COST
       DC        16.000000          0.000000
       DG         0.000000          0.000000
       DH         3.000000          0.000000
       DD         0.000000          0.000000
       DA         4.000000          0.000000
       DB        12.000000          0.000000
       DE         0.000000          0.000000
       DF        13.000000          0.000000


      ROW   SLACK OR SURPLUS     DUAL PRICES
       2)         0.000000          1.000000
       3)         0.000000          1.000000
       4)         2.000000          0.000000
       5)         3.000000          0.000000
       6)         0.000000          1.000000
       7)         0.000000          1.000000
       8)        22.000000          0.000000
       9)         1.000000          0.000000
      10)         0.000000          0.000000
      11)        14.000000          0.000000
      12)         8.000000          0.000000
      13)         0.000000          0.000000
      14)        15.000000          0.000000
      15)        19.000000          0.000000
      16)         5.000000          0.000000
      17)        25.000000          0.000000
      18)         7.000000          0.000000
      19)         9.000000          0.000000

NO. ITERATIONS=       2
```

```
MAX dc
ST
    dg = 0
    dh - dg <= 3
    dd - dg <= 2
    da - dh <= 4
    db - dh <= 9
    dc - db <= 4
    de - db <= 10
    df - da <= 10
    db - da <= 8
    da - df <= 5
    db - df <= 7
    dc - df <= 3
    de - df <= 2
    dd - dc <= 3
    df - dd <= 18
    de - dd <= 25
    dg - de <= 7
    dd - de <= 9
END
```

So, the answer for a) is 16

b)

MAX dc + dh + db + da + df + dd + de

ST

dg = 0

dh – dg <= 3

dd -dg <= 2

da – dh <= 4

db -dh <= 9

dc – db <= 4

de – db <= 10

df – da <= 10

db – da <= 8

da – df <= 5

db – df <= 7

dc – df <= 3

de – df <= 2

dd – dc <= 3

df – dd <= 18

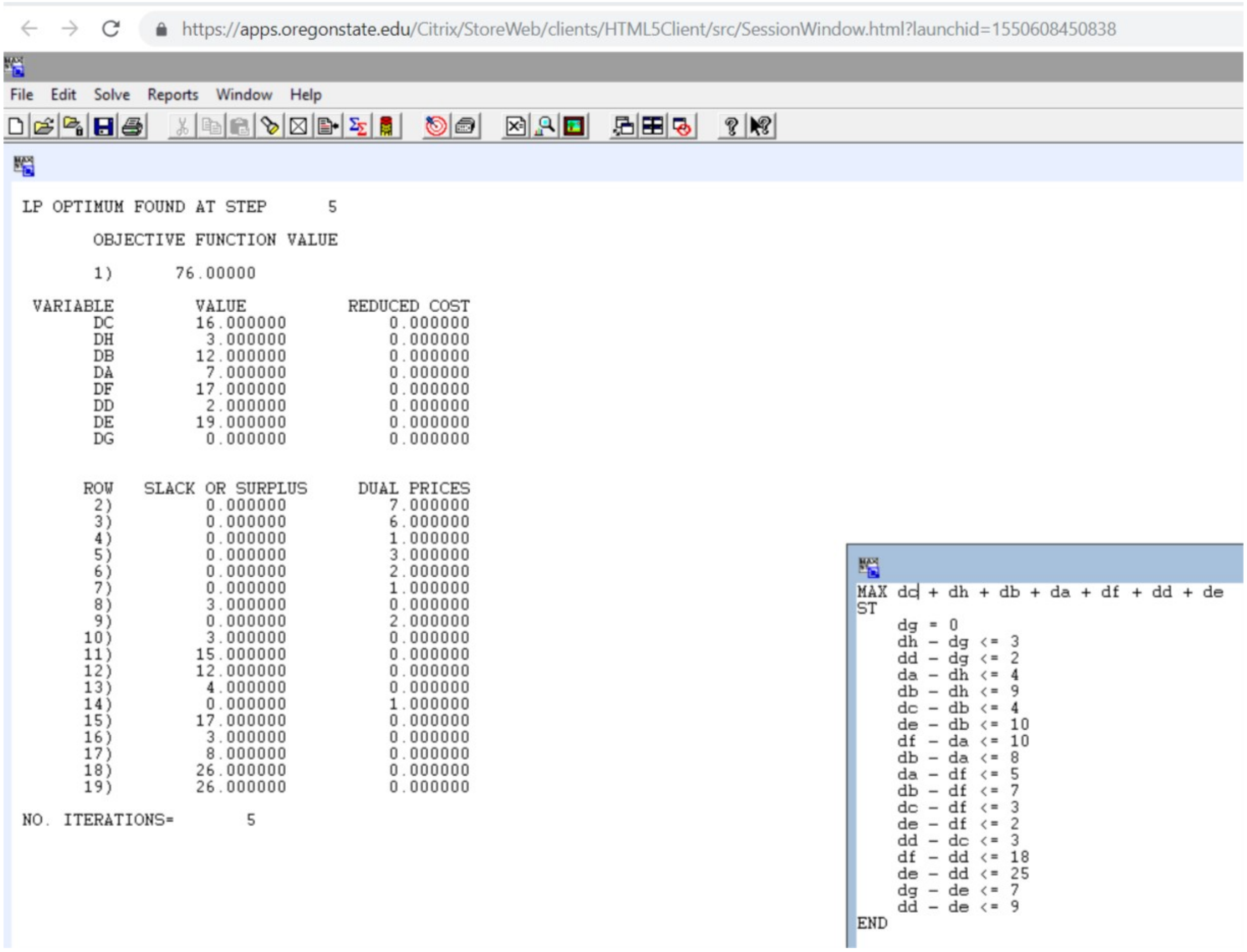
de – dd <= 25

dg – de <= 7

dd – de <= 9

END

Picture of LINDO code and output is below

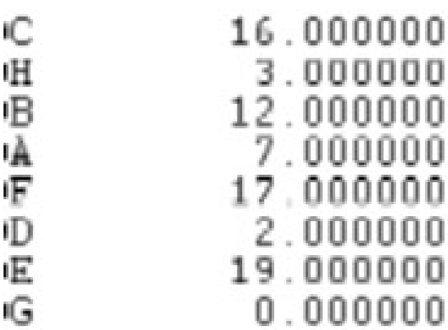

https://apps.oregonstate.edu/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1550608450838

File Edit Solve Reports Window Help

```
LP OPTIMUM FOUND AT STEP      5

       OBJECTIVE FUNCTION VALUE

       1)      76.00000

 VARIABLE        VALUE          REDUCED COST
       DC        16.000000          0.000000
       DH         3.000000          0.000000
       DB        12.000000          0.000000
       DA         7.000000          0.000000
       DF        17.000000          0.000000
       DD         2.000000          0.000000
       DE        19.000000          0.000000
       DG         0.000000          0.000000


      ROW   SLACK OR SURPLUS     DUAL PRICES
       2)         0.000000          7.000000
       3)         0.000000          6.000000
       4)         0.000000          1.000000
       5)         0.000000          3.000000
       6)         0.000000          2.000000
       7)         0.000000          1.000000
       8)         3.000000          0.000000
       9)         0.000000          2.000000
      10)         3.000000          0.000000
      11)        15.000000          0.000000
      12)        12.000000          0.000000
      13)         4.000000          0.000000
      14)         0.000000          1.000000
      15)        17.000000          0.000000
      16)         3.000000          0.000000
      17)         8.000000          0.000000
      18)        26.000000          0.000000
      19)        26.000000          0.000000

NO. ITERATIONS=       5
```

```
MAX dc + dh + db + da + df + dd + de
ST
    dg = 0
    dh - dg <= 3
    dd - dg <= 2
    da - dh <= 4
    db - dh <= 9
    dc - db <= 4
    de - db <= 10
    df - da <= 10
    db - da <= 8
    da - df <= 5
    db - df <= 7
    dc - df <= 3
    de - df <= 2
    dd - dc <= 3
    df - dd <= 18
    de - dd <= 25
    dg - de <= 7
    dd - de <= 9
END
```

The answer to b) is the picture below with vertices and shortest path distances:



C	16.000000
H	3.000000
B	12.000000
A	7.000000
F	17.000000
D	2.000000
E	19.000000
G	0.000000

G is 0 as expected. C remains 16 as expected. Adjacent vertices have expected values as well. And all others have their correct values.

Problem 2 – USING LINDO AGAIN

s = number of silk ties, p = number of polyester ties, b = number of blend1 ties, c = number of blend2 tie

~~ms = raw silk material in yards, mp = raw polyester material in yards, mc = raw cotton material in yards~~

material Cost for each tie in dollars:

s = 0.125*20 + 0 + 0 = 2.5

p= 0 + 0.08*6 + 0 = 0.48

b= 0 + 0.05*6 + 0.05*9 = 0.3 + 0.45 = 0.75

c= 0 + 0.03*6 + 0.07*9 = 0.18 + 0.63 = 0.81

Labor cost for each tie is 0.75 so TOTAL costs for each tie is:

s= 3.25

p= 1.23

b= 1.5

c= 1.56

Profits for each tie are:

s= 6.7 - 3.25 = 3.45

p= 3.55 - 1.23 = 2.32

b= 4.31 - 1.5 = 2.81

c= 4.81 - 1.56 = 3.25

This means we are maximizing the following:

Maximize: 3.45s + 2.32p + 2.81b + 3.25c

The constraints are the following:

Cannot use more than 1,000 yards of silk

0.125*s <= 1000

Cannot use more than 2,000 yards of Polyester

0.08*p + 0.05*b + 0.03*c <= 2000

Cannot use more than 1,250 yards of cotton

0.05b + 0.07c <= 1250

Cannot sell less than 6000 s ties

s >= 6000

Cannot sell more than 7000 s ties

s <= 7000

Cannot sell less than 10,000 p ties

p >= 10000

Cannot sell more than 14,000 p ties

p <= 14000

Cannot sell less than 13,000 b ties

b >= 13000

Cannot sell more than 16,000 b ties

b <= 16000

Cannot sell less than 6,000 c ties

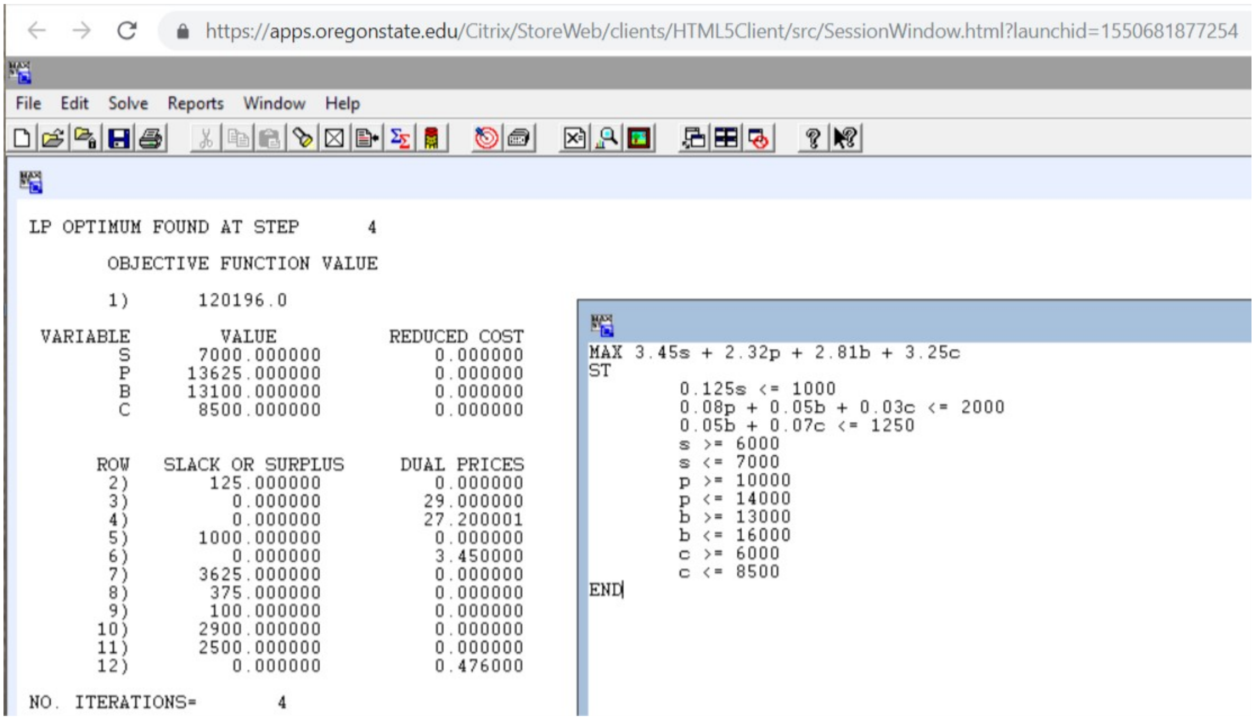
c >= 6000

Cannot sell more than 8,500 c ties

c <= 8500

So we have for the objective function and constraints:

```
MAX 3.45s + 2.32p + 2.81b + 3.25c
ST
        0.125*s <= 1000
        0.08*p + 0.05*b + 0.03*c <= 2000
        0.05b + 0.07c <= 1250
        s >= 6000
        s <= 7000
        p >= 10000
        p <= 14000
        b >= 13000
        b <= 16000
        c >= 6000
        c <= 8500
END
```



```
LP OPTIMUM FOUND AT STEP       4

       OBJECTIVE FUNCTION VALUE

       1)       120196.0

 VARIABLE        VALUE          REDUCED COST
        S        7000.000000          0.000000
        P       13625.000000          0.000000
        B       13100.000000          0.000000
        C        8500.000000          0.000000


      ROW   SLACK OR SURPLUS     DUAL PRICES
       2)         125.000000          0.000000
       3)           0.000000         29.000000
       4)           0.000000         27.200001
       5)        1000.000000          0.000000
       6)           0.000000          3.450000
       7)        3625.000000          0.000000
       8)         375.000000          0.000000
       9)         100.000000          0.000000
      10)        2900.000000          0.000000
      11)        2500.000000          0.000000
      12)           0.000000          0.476000

NO. ITERATIONS=       4
```

```
MAX 3.45s + 2.32p + 2.81b + 3.25c
ST
        0.125s <= 1000
        0.08p + 0.05b + 0.03c <= 2000
        0.05b + 0.07c <= 1250
        s >= 6000
        s <= 7000
        p >= 10000
        p <= 14000
        b >= 13000
        b <= 16000
        c >= 6000
        c <= 8500
END
```

This means the optimal profit is 120196 dollars.

And the number of ties for each type are as follows:

S= 7000

P= 13625

B= 13100

C= 8500

This makes sense to me because the only limit on silk ties is the maximum number which is 7000. C ties are also maxed out because there is enough material and they are the most profitable out of those that use polyester and cotton. P ties and b ties are such that profit is max using up the resources needed.

Problem 3

Part A

So, we are minimizing because we are trying to reduce cost. We have a variable for every edge or route. I see 21 edges in graph. I also see 21 costs in the tables. We will have 21 variables to MIN.

Some constraints will be that supply is not exceeded and demand is met.

Another constraint is that values are non-negative because we are minimizing.

Also, what is going into a warehouse minus what is going out must = 0

Variables will be that p11 signifies shipping from plant 1 to ware house 1, etc. And w11 will be shipping from warehouse 1 to retail 1, etc.

So the objective function with constraints in near LINDO format will be:

MIN 10*p11 + 15*p12 + 11*p21 + 8*p22 + 13*p31 + 8 *p32 + 9*p33 + 14*p42 + 8*p43 + 5*w11 + 6*w12 + 7*w13 + 10*w14 + 12*w23 + 8*w24 + 10*w25 + 14*w26 + 14*w34 + 12*w35 + 12*w36 + 6*w37

ST

P11 + p12 <= 150

P21 + p22 <= 450

P31 + p32 + p33 <= 250

P42 + p43 <= 150

W11 >= 100

W12 >= 150

W13 + w23 >= 100

W14 + w24 + w34 >= 200

W25 + w35 >= 200

W26 + w36 >= 150

W37 >= 100

P11, p12, p21, p22, p31,p32,p33,p42,p43,w11,w12,w13,w14,w23,w24,w25,w26,w34,w35,w36,w37 >= 0

P11 + p21 + p31 - w11 – w12 – w13 – w14 >= 0

P12 + p22 + p32 + p42 – w23 -w24 – w25 – w26 >= 0

P33 + p43 – w34 – w35 – w36 – w37 >= 0

END

USING LINDO

Full picture of code and results

```
LP OPTIMUM FOUND AT STEP      13

        OBJECTIVE FUNCTION VALUE

        1)      17100.00

  VARIABLE        VALUE          REDUCED COST
       P11       150.000000          0.000000
       P12         0.000000          8.000000
       P21       200.000000          0.000000
       P22       250.000000          0.000000
       P31         0.000000          2.000000
       P32       150.000000          0.000000
       P33       100.000000          0.000000
       P42         0.000000          7.000000
       P43       150.000000          0.000000
       W11       100.000000          0.000000
       W12       150.000000          0.000000
       W13       100.000000          0.000000
       W14         0.000000          5.000000
       W23         0.000000          2.000000
       W24       200.000000          0.000000
       W25       200.000000          0.000000
       W26         0.000000          1.000000
       W34         0.000000          7.000000
       W35         0.000000          3.000000
       W36       150.000000          0.000000
       W37       100.000000          0.000000


       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)         0.000000          1.000000
        3)         0.000000          0.000000
        4)         0.000000          0.000000
        5)         0.000000          1.000000
        6)         0.000000        -16.000000
        7)         0.000000        -17.000000
        8)         0.000000        -18.000000
        9)         0.000000        -16.000000
       10)         0.000000        -18.000000
       11)         0.000000        -21.000000
       12)         0.000000        -15.000000
       13)       150.000000          0.000000
       14)         0.000000          0.000000
       15)       200.000000          0.000000
       16)       250.000000          0.000000
       17)         0.000000          0.000000
       18)         0.000000          0.000000
       19)       100.000000          0.000000
       20)         0.000000          0.000000
       21)       150.000000          0.000000
       22)       100.000000          0.000000
       23)       150.000000          0.000000
       24)       100.000000          0.000000
       25)         0.000000          0.000000
       26)         0.000000          0.000000
       27)       200.000000          0.000000
       28)       200.000000          0.000000
       29)         0.000000          0.000000
       30)         0.000000          0.000000
       31)         0.000000          0.000000
       32)       150.000000          0.000000
       33)       100.000000          0.000000
       34)         0.000000        -11.000000
       35)         0.000000         -8.000000
       36)         0.000000         -9.000000

 NO. ITERATIONS=      13
```

```
<untitled>
MIN 10p11 + 15p12 + 11p21 + 8p22 + 13p31 + 8p32 + 9p33 + 14p42 + 8p43 + 5w11 + 6w12 + 7w13 + 10w14 + 12w23 + 8w24 + 10w25 + 14w26 + 14w34 + 12w35 + 12w36 + 6w37
ST
        p11 + p12 <= 150
        p21 + p22 <= 450
        p31 + p32 + p33 <= 250
        p42 + p43 <= 150
        w11 >= 100
        w12 >= 150
        w13 + w23 >= 100
        w14 + w24 + w34 >= 200
        w25 + w35 >= 200
        w26 + w36 >= 150
        w37 >= 100
        p11 >= 0
        p12 >= 0
        p21 >= 0
        p22 >= 0
        p31 >= 0
        p31 >= 0
        p33 >= 0
        p42 >= 0
        p43 >= 0
        w11 >= 0
        w12 >= 0
        w13 >= 0
        w14 >= 0
        w23 >= 0
        w24 >= 0
        w25 >= 0
        w26 >= 0
        w34 >= 0
        w35 >= 0
        w36 >= 0
        w37 >= 0
        p11 + p21 + p31 - w11 - w12 - w13 - w14 >= 0
        p12 + p22 + p32 + p42 - w23 - w24 - w25 - w26 >= 0
        p33 + p43 - w34 - w35 - w36 - w37 >= 0
END
```

Smaller picture of variable values

```
P11       150.000000
P12         0.000000
P21       200.000000
P22       250.000000
P31         0.000000
P32       150.000000
P33       100.000000
P42         0.000000
P43       150.000000
W11       100.000000
W12       150.000000
W13       100.000000
W14         0.000000
W23         0.000000
W24       200.000000
W25       200.000000
W26         0.000000
W34         0.000000
W35         0.000000
W36       150.000000
W37       100.000000
```

This means that the minimum cost to ship is 17100

The smaller above graph details exactly how to ship to achieve this optimal. That is P11 = 150 and p12 = 0 means plant 1 will only ship to warehouse one and will ship 150 units. Similarly, plant two will ship 200 to warehouse one and 250 to warehouse 2 and so on for the rest of the plants. W11 = 100, w12= 150, w13 = 100 and w14 = 0 means that warehouse one will not ship to retail 4 but will ship 100, 150 and 100 units to retails 1 2 and 3 respectively. This is the same for the rest of the warehouses shown above in the smaller pciture.

below is additional picture for part A of only code for easier viewing

```
MIN 10p11 + 15p12 + 11p21 + 8p22 + 13p31 + 8p32 + 9p33 + 14p42 + 8p43 + 5w11 + 6w12 + 7w13 + 10w14 + 12w23 + 8w24 + 10w25 + 14w26 + 14w34 + 12w35 + 12w36 + 6w37
ST
        p11 + p12 <= 150
        p21 + p22 <= 450
        p31 + p32 + p33 <= 250
        p42 + p43 <= 150
        w11 >= 100
        w12 >= 150
        w13 + w23 >= 100
        w14 + w24 + w34 >= 200
        w25 + w35 >= 200
        w26 + w36 >= 150
        w37 >= 100
        p11 >= 0
        p12 >= 0
        p21 >= 0
        p22 >= 0
        p31 >= 0
        p31 >= 0
        p33 >= 0
        p42 >= 0
        p43 >= 0
        w11 >= 0
        w12 >= 0
        w13 >= 0
        w14 >= 0
        w23 >= 0
        w24 >= 0
        w25 >= 0
        w26 >= 0
        w34 >= 0
        w35 >= 0
        w36 >= 0
        w37 >= 0
        p11 + p21 + p31 - w11 - w12 - w13 - w14 >= 0
        p12 + p22 + p32 + p42 - w23 - w24 - w25 - w26 >= 0
        p33 + p43 - w34 - w35 - w36 - w37 >= 0
END
```

Part B

If we eliminate warehouse 2 entirely, starting from code for part A the only changes that need to be made are:

Removing variables p12, p22,p32, p42, w23, w24, w25, w26 from objective function

And from constraints.

~~The constraints should not change much outside of the removal of these variables and the objective function is the same~~. By removing these variables one entire constraint will be removed and many will be modified So now the near-LINDO code is the following:

MIN 10*p11 + 11*p21 + 13*p31 + 9*p33 + 8*p43 + 5*w11 + 6*w12 + 7*w13 + 10*w14 + 14*w34 + 12*w35 + 12*w36 + 6*w37

ST

P11 <= 150

P21 <= 450

P31 + p33 <= 250

p43 <= 150

W11 >= 100

W12 >= 150

W13 >= 100

W14 + w34 >= 200

w35 >= 200

w36 >= 150

W37 >= 100

P11, p21, p31, p33, p43,w11,w12,w13,w14 ,w34,w35,w36,w37 >= 0

P11 + p21 + p31 - w11 – w12 – w13 – w14 >= 0

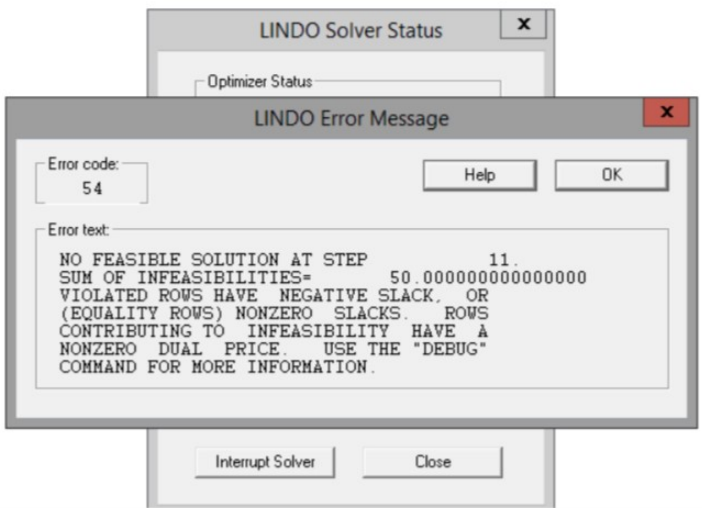
P33 + p43 – w34 – w35 – w36 – w37 >= 0

END

USING LINDO

RESULTS BELOW IN PICTURE

```
MIN 10p11 + 11p21 + 13p31 + 9p33 + 8p43 + 5w11 + 6w12 + 7w13 + 10w14 + 14w34 + 12w35 + 12w36 + 6w37
ST
        p11 <= 150
        p21 <= 450
        p31 + p33 <= 250
        p43 <= 150
        w11 >= 100
        w12 >= 150
        w13 >= 100
        w14 + w34 >= 200
        w35 >= 200
        w36 >= 150
        w37 >= 100
        p11 >= 0
        p21 >= 0
        p31 >= 0
        p33 >= 0
        p43 >= 0
        w11 >= 0
        w12 >= 0
        w13 >= 0
        w14 >= 0
        w34 >= 0
        w35 >= 0
        w36 >= 0
        w37 >= 0
        p11 + p21 + p31 - w11 - w12 - w13 - w14 >= 0
        p33 + p43 - w34 - w35 - w36 - w37 >= 0
END
```



The result was NO FEASIBLE SOLUTION

Specifically, at step 11.

The reason why being that in this scenario given the routes, disregarding prices, the most that can pass through warehouse 3 is from plant 3 and plant 4 for a max of 400 however, the retails stores than can only receive from warehouse 3 in this scenario require 450. So even if warehouse three ships nothing to retail 4 which is the only retail that can receive from either warehouse 1 or 3, there will not be enough supply coming from plant 3 and 4 to meet this demand. Thus in this scenario retails 7, 5 and 6 will no matter what not receive enough supply for their demand. So the result should be infeasible as was determined by LINDO and my code.

Part C

I am doing part C before part B because it will be easier to make the changes in that order.

If we limit warehouse 2 to having only 100 incoming and 100 outgoing shipments the only change to our code needed is extra constraints saying so. These will be the following:

P12 + p22 + p32 + p42 <= 100

W23 + w24 + w25 + w26 <= 100

Maybe we could use only one but I will put in both to be safe

So using same code as before but with these two extra constrains we get:

USING LINDO STILL

```
LP OPTIMUM FOUND AT STEP      15

        OBJECTIVE FUNCTION VALUE

        1)      18300.00

  VARIABLE        VALUE          REDUCED COST
       P11       150.000000          0.000000
       P12         0.000000          8.000000
       P21       350.000000          0.000000
       P22       100.000000          0.000000
       P31         0.000000          4.000000
       P32         0.000000          2.000000
       P33       250.000000          0.000000
       P42         0.000000          9.000000
       P43       150.000000          0.000000
       W11       100.000000          0.000000
       W12       150.000000          0.000000
       W13       100.000000          0.000000
       W14       150.000000          0.000000
       W23         0.000000          7.000000
       W24        50.000000          0.000000
       W25        50.000000          0.000000
       W26         0.000000          4.000000
       W34         0.000000          4.000000
       W35       150.000000          0.000000
       W36       150.000000          0.000000
       W37       100.000000          0.000000

       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)         0.000000          1.000000
        3)         0.000000          0.000000
        4)         0.000000          2.000000
        5)         0.000000          3.000000
        6)         0.000000        -16.000000
        7)         0.000000        -17.000000
        8)         0.000000        -18.000000
        9)         0.000000        -21.000000
       10)         0.000000        -23.000000
       11)         0.000000        -23.000000
       12)         0.000000        -17.000000
       13)       150.000000          0.000000
       14)         0.000000          0.000000
       15)       350.000000          0.000000
       16)       100.000000          0.000000
       17)         0.000000          0.000000
       18)         0.000000          0.000000
       19)       250.000000          0.000000
       20)         0.000000          0.000000
       21)       150.000000          0.000000
       22)       100.000000          0.000000
       23)       150.000000          0.000000
       24)       100.000000          0.000000
       25)       150.000000          0.000000
       26)         0.000000          0.000000
       27)        50.000000          0.000000
       28)        50.000000          0.000000
       29)         0.000000          0.000000
       30)         0.000000          0.000000
       31)       150.000000          0.000000
       32)       150.000000          0.000000
       33)       100.000000          0.000000
       34)         0.000000        -11.000000
       35)         0.000000         -8.000000
       36)         0.000000        -11.000000
       37)         0.000000          0.000000
       38)         0.000000          5.000000

 NO. ITERATIONS=      15
```

```
<untitled>
MIN 10p11 + 15p12 + 11p21 + 8p22 + 13p31 + 8p32 + 9p33 + 14p42 + 8p43 + 5w11 + 6w12 + 7w13 + 10w14 + 12w23 + 8w24 + 10w25 + 14w26 + 14w34 + 12w35 + 12w36 + 6w37
ST
        p11 + p12 <= 150
        p21 + p22 <= 450
        p31 + p32 + p33 <= 250
        p42 + p43 <= 150
        w11 >= 100
        w12 >= 150
        w13 + w23 >= 100
        w14 + w24 + w34 >= 200
        w25 + w35 >= 200
        w26 + w36 >= 150
        w37 >= 100
        p11 >= 0
        p12 >= 0
        p21 >= 0
        p22 >= 0
        p31 >= 0
        p31 >= 0
        p33 >= 0
        p42 >= 0
        p43 >= 0
        w11 >= 0
        w12 >= 0
        w13 >= 0
        w14 >= 0
        w23 >= 0
        w24 >= 0
        w25 >= 0
        w26 >= 0
        w34 >= 0
        w35 >= 0
        w36 >= 0
        w37 >= 0
        p11 + p21 + p31 - w11 - w12 - w13 - w14 >= 0
        p12 + p22 + p32 + p42 - w23 - w24 - w25 - w26 >= 0
        p33 + p43 - w34 - w35 - w36 - w37 >= 0
        p12 + p22 + p32 + p42 <= 100
        w23 + w24 + w25 + w26 <= 100
END
```

This means that this is feasible if we limit warehouse 2.

However, now the objective function optimal value is larger at 18300.

And these are the exact routes and shipment amounts for each as was presented for part A:

Below

P11	150.000000
P12	0.000000
P21	350.000000
P22	100.000000
P31	0.000000
P32	0.000000
P33	250.000000
P42	0.000000
P43	150.000000
W11	100.000000
W12	150.000000
W13	100.000000
W14	150.000000
W23	0.000000
W24	50.000000
W25	50.000000
W26	0.000000
W34	0.000000
W35	150.000000
W36	150.000000
W37	100.000000

And again, here is close up of code:

```
MIN 10p11 + 15p12 + 11p21 + 8p22 + 13p31 + 8p32 + 9p33 + 14p42 + 8p43 + 5w11 + 6w12 + 7w13 + 10w14 + 12w23 + 8w24 + 10w25 + 14w26 + 14w34 + 12w35 + 12w36 + 6w37
ST
        p11 + p12 <= 150
        p21 + p22 <= 450
        p31 + p32 + p33 <= 250
        p42 + p43 <= 150
        w11 >= 100
        w12 >= 150
        w13 + w23 >= 100
        w14 + w24 + w34 >= 200
        w25 + w35 >= 200
        w26 + w36 >= 150
        w37 >= 100
        p11 >= 0
        p12 >= 0
        p21 >= 0
        p22 >= 0
        p31 >= 0
        p31 >= 0
        p33 >= 0
        p42 >= 0
        p43 >= 0
        w11 >= 0
        w12 >= 0
        w13 >= 0
        w14 >= 0
        w23 >= 0
        w24 >= 0
        w25 >= 0
        w26 >= 0
        w34 >= 0
        w35 >= 0
        w36 >= 0
        w37 >= 0
        p11 + p21 + p31 - w11 - w12 - w13 - w14 >= 0
        p12 + p22 + p32 + p42 - w23 - w24 - w25 - w26 >= 0
        p33 + p43 - w34 - w35 - w36 - w37 >= 0
        p12 + p22 + p32 + p42 <= 100
        w23 + w24 + w25 + w26 <= 100
END
```

Problem 4

4. Making Change**(6 points)**

Given coins of denominations (value) $1 = v_1 < v_2 < \dots < v_n$, we wish to make change for an amount A using as few coins as possible. Assume that v_i's and A are integers. Since $v_1 = 1$ there will always be a solution. Solve the coin change using integer programming. For each the following denomination sets and amounts formulate the problem as an integer program with an objective function and constraints, determine the optimal solution. What is the minimum number of coins used in each case and how many of each coin is used? Include a copy of your code.

a) V = [1, 5, 10, 25] and A = 202.
b) V = [1, 3, 7, 12, 27] and A = 293

V1 for both a) and b) is coin of value 1. V2 for a is 5 and 3 for b. V3 for a is coin of value 10 and for b a coin of value 7. V4 is a coin of 25 for a and 12 for b. And v5 is used only in b) which represents the coin of value 27. And used GIN on v1 – v4 for a and GIN on v1 – v5.

USING LINDO WITH GIN

a)

Before any LP, I will use brute force to find a guess of a solution. This is essentially assigning largest coin that will not force sum over A. Add 25 values 8 times to get 200, 8*250. Obvious I cannot get 200 with fewer coins of smaller values. The one's place value of 2 can only be obtained from 2 1coins. This means the fewest coins to be used is 8+2 or 10. Exchanging the 2 ones for anything else is impossible to achieve 2 in the ones place of the resulting A. And exchanging any of the 25coins for 5 5coins or 2 10coins and a 5 result in a larger number of coins so I am aiming to achieve 10.

```
MIN v1 + v2 + v3 + v4

ST

        V1 + 5v2 + 10v3 + 25v4 = 202

END

GIN v1

GIN v2

GIN v3

GIN v4
```

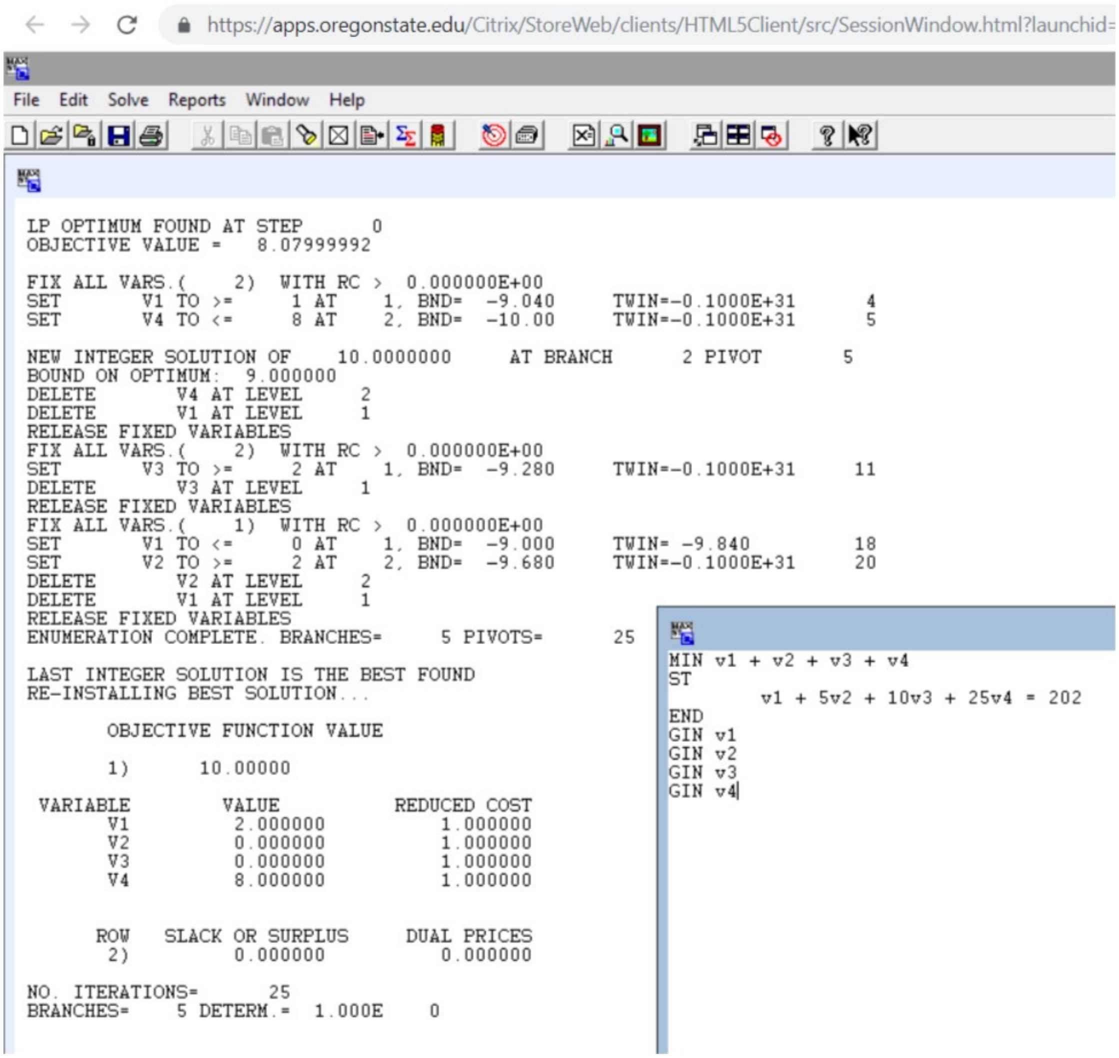
Picture of results and input code below:

Objective function value = 10.0

With 2 v1 coins

And 8 v4 coins

As predicted from my brute force.



https://apps.oregonstate.edu/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launchid=

File Edit Solve Reports Window Help

```
LP OPTIMUM FOUND AT STEP      0
OBJECTIVE VALUE =   8.07999992

FIX ALL VARS.(     2)  WITH RC >  0.000000E+00
SET       V1 TO >=      1 AT     1, BND=  -9.040     TWIN=-0.1000E+31      4
SET       V4 TO <=      8 AT     2, BND=  -10.00     TWIN=-0.1000E+31      5

NEW INTEGER SOLUTION OF    10.0000000     AT BRANCH      2 PIVOT       5
BOUND ON OPTIMUM:  9.000000
DELETE       V4 AT LEVEL     2
DELETE       V1 AT LEVEL     1
RELEASE FIXED VARIABLES
FIX ALL VARS.(     2)  WITH RC >  0.000000E+00
SET       V3 TO >=      2 AT     1, BND=  -9.280     TWIN=-0.1000E+31     11
DELETE       V3 AT LEVEL     1
RELEASE FIXED VARIABLES
FIX ALL VARS.(     1)  WITH RC >  0.000000E+00
SET       V1 TO <=      0 AT     1, BND=  -9.000     TWIN= -9.840        18
SET       V2 TO >=      2 AT     2, BND=  -9.680     TWIN=-0.1000E+31     20
DELETE       V2 AT LEVEL     2
DELETE       V1 AT LEVEL     1
RELEASE FIXED VARIABLES
ENUMERATION COMPLETE. BRANCHES=     5 PIVOTS=      25

LAST INTEGER SOLUTION IS THE BEST FOUND
RE-INSTALLING BEST SOLUTION...

       OBJECTIVE FUNCTION VALUE

       1)      10.00000

 VARIABLE        VALUE          REDUCED COST
       V1         2.000000          1.000000
       V2         0.000000          1.000000
       V3         0.000000          1.000000
       V4         8.000000          1.000000


      ROW   SLACK OR SURPLUS     DUAL PRICES
       2)         0.000000          0.000000

NO. ITERATIONS=      25
BRANCHES=     5 DETERM.=   1.000E    0
```

```
MIN v1 + v2 + v3 + v4
ST
       v1 + 5v2 + 10v3 + 25v4 = 202
END
GIN v1
GIN v2
GIN v3
GIN v4
```

b) is below

b)

V= 1,3,7,12,27

A=293

Starting with brute force/greedy algorithm, I will first remove the max number of 27coins before getting a negative A leftover. So 277*11= 297 but 27*10=270. So, 10 27coins with a left over of 23. One 12coin for a leftover of 11. One 7coin with a leftover of 4, and then one 3coin and one onecoin for 4. So we have 10+1+1+1+1=14. Instead of 23 being made with 4 coins we could have 3 7coins and 2 onecoins but this would be 5. I feel confident for now, that the objective function optimal solution is 14.

*Code and result below as two pictures from LINDO used in browser *

Two pics to provide all of the result page https://apps.oregonstate.edu/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1550608450838

File Edit Solve Reports Window Help

```
LP OPTIMUM FOUND AT STEP      1
OBJECTIVE VALUE =   10.8518515

FIX ALL VARS.(     2)  WITH RC >  0.000000E+00
SET       V1 TO <=      0 AT    1, BND=  -12.33     TWIN= -12.83          8
SET       V5 TO <=      9 AT    2, BND=  -13.17     TWIN=-0.1000E+31     10
SET       V4 TO >=      5 AT    3, BND=  -13.63     TWIN=-0.1000E+31     12
SET       V5 TO <=      8 AT    4, BND=  -13.63     TWIN=-0.1000E+31     12
SET       V4 TO >=      7 AT    5, BND=  -14.74     TWIN=-0.1000E+31     16
SET       V5 TO <=      7 AT    6, BND=  -15.67     TWIN=-0.1000E+31     18
SET       V4 TO >=      9 AT    7, BND=  -15.85     TWIN=-0.1000E+31     20
SET       V5 TO <=      6 AT    8, BND=  -16.92     TWIN=-0.1000E+31     22
SET       V4 TO >=     11 AT    9, BND=  -16.96     TWIN=-0.1000E+31     24
SET       V5 TO <=      5 AT   10, BND=  -18.17     TWIN=-0.1000E+31     26
SET       V4 TO >=     14 AT   11, BND=  -18.63     TWIN=-0.1000E+31     28
SET       V5 TO <=      4 AT   12, BND=  -19.42     TWIN=-0.1000E+31     30
SET       V4 TO >=     16 AT   13, BND=  -19.74     TWIN=-0.1000E+31     32
SET       V5 TO <=      3 AT   14, BND=  -20.67     TWIN=-0.1000E+31     34
SET       V4 TO >=     18 AT   15, BND=  -20.85     TWIN=-0.1000E+31     36
SET       V5 TO <=      2 AT   16, BND=  -21.92     TWIN=-0.1000E+31     38
SET       V4 TO >=     20 AT   17, BND=  -21.96     TWIN=-0.1000E+31     40
SET       V5 TO <=      1 AT   18, BND=  -23.17     TWIN=-0.1000E+31     42
SET       V4 TO >=     23 AT   19, BND=  -23.63     TWIN=-0.1000E+31     44
SET       V5 TO <=      0 AT   20, BND=  -24.42     TWIN=-0.1000E+31     45
DELETE       V4 AT LEVEL    21
DELETE       V5 AT LEVEL    20
DELETE       V4 AT LEVEL    19
DELETE       V5 AT LEVEL    18
DELETE       V4 AT LEVEL    17
DELETE       V5 AT LEVEL    16
DELETE       V4 AT LEVEL    15
DELETE       V5 AT LEVEL    14
DELETE       V4 AT LEVEL    13
DELETE       V5 AT LEVEL    12
DELETE       V4 AT LEVEL    11
DELETE       V5 AT LEVEL    10
DELETE       V4 AT LEVEL     9
DELETE       V5 AT LEVEL     8
DELETE       V4 AT LEVEL     7
DELETE       V5 AT LEVEL     6
DELETE       V4 AT LEVEL     5
DELETE       V5 AT LEVEL     4
DELETE       V4 AT LEVEL     3
DELETE       V5 AT LEVEL     2
FLIP        V1 TO >=          1 AT    1 WITH BND=   -12.833333
SET       V4 TO <=      1 AT    2, BND=  -22.00     TWIN= -12.93         48

NEW INTEGER SOLUTION OF    22.0000000     AT BRANCH     21 PIVOT      48
BOUND ON OPTIMUM:  12.00000
FLIP        V4 TO >=          2 AT    2 WITH BND=   -12.925926
SET       V5 TO <=      9 AT    3, BND=  -14.08     TWIN=-0.1000E+31     50
SET       V4 TO <=      4 AT    4, BND=  -15.00     TWIN= -14.59         53

NEW INTEGER SOLUTION OF    15.0000000     AT BRANCH     23 PIVOT      53
BOUND ON OPTIMUM:  12.00000
DELETE       V4 AT LEVEL     4
DELETE       V5 AT LEVEL     3
DELETE       V4 AT LEVEL     2
DELETE       V1 AT LEVEL     1
RELEASE FIXED VARIABLES
FIX ALL VARS.(     2)  WITH RC >  0.000000E+00
SET       V4 TO <=      1 AT    1, BND=  -12.57     TWIN= -12.70         63
SET       V3 TO >=      2 AT    2, BND=  -12.75     TWIN=-0.1000E+31     65
SET       V4 TO >=      1 AT    3, BND=  -12.89     TWIN= -13.29         67
SET       V5 TO <=      9 AT    4, BND=  -12.89     TWIN=-0.1000E+31     67
SET       V5 TO >=      9 AT    5, BND=  -15.43     TWIN=-0.1000E+31     69
DELETE       V3 AT LEVEL     6
DELETE       V5 AT LEVEL     5
DELETE       V5 AT LEVEL     4
FLIP        V4 TO <=          0 AT    3 WITH BND=   -13.285714
SET       V3 TO >=      4 AT    4, BND=  -13.81     TWIN=-0.1000E+31     71
SET       V3 TO <=      4 AT    5, BND=  -13.81     TWIN=-0.1000E+31     71
DELETE       V5 AT LEVEL     6
DELETE       V3 AT LEVEL     5
DELETE       V3 AT LEVEL     4
DELETE       V4 AT LEVEL     3
DELETE       V3 AT LEVEL     2
FLIP        V4 TO >=          2 AT    1 WITH BND=   -12.703704
SET       V5 TO <=      9 AT    2, BND=  -13.58     TWIN=-0.1000E+31     73
SET       V5 TO >=      9 AT    3, BND=  -13.58     TWIN=-0.1000E+31     73
SET       V4 TO <=      3 AT    4, BND=  -14.00     TWIN=-0.1000E+31     74

NEW INTEGER SOLUTION OF    14.0000000     AT BRANCH     31 PIVOT      74
BOUND ON OPTIMUM:  12.33333
DELETE       V4 AT LEVEL     4
DELETE       V5 AT LEVEL     3
DELETE       V5 AT LEVEL     2
DELETE       V4 AT LEVEL     1
RELEASE FIXED VARIABLES
FIX ALL VARS.(     2)  WITH RC >  0.000000E+00
SET       V4 TO >=      2 AT    1, BND=  -12.85     TWIN= -14.67         83
SET       V4 TO <=      2 AT    2, BND=  -12.85     TWIN=-0.1000E+31     83
SET       V5 TO <=      9 AT    3, BND=  -19.67     TWIN=-0.1000E+31     85
DELETE       V5 AT LEVEL     3
DELETE       V4 AT LEVEL     2
DELETE       V4 AT LEVEL     1
RELEASE FIXED VARIABLES
SET       V3 TO <=      0 AT    1, BND=  -13.58     TWIN= -13.08         97
DELETE       V3 AT LEVEL     1
ENUMERATION COMPLETE. BRANCHES=    34 PIVOTS=      97

LAST INTEGER SOLUTION IS THE BEST FOUND
RE-INSTALLING BEST SOLUTION...

       OBJECTIVE FUNCTION VALUE

       1)      14.00000
```

```
MIN v1 + v2 + v3 + v4 + v5
ST
       v1 + 3v2 + 7v3 + 12v4 + 27v5= 293
END
GIN v1
GIN v2
GIN v3
GIN v4
GIN v5
```

```
DELETE        V5 AT LEVEL    16
DELETE        V4 AT LEVEL    15
DELETE        V5 AT LEVEL    14
DELETE        V4 AT LEVEL    13
DELETE        V5 AT LEVEL    12
DELETE        V4 AT LEVEL    11
DELETE        V5 AT LEVEL    10
DELETE        V4 AT LEVEL     9
DELETE        V5 AT LEVEL     8
DELETE        V4 AT LEVEL     7
DELETE        V5 AT LEVEL     6
DELETE        V4 AT LEVEL     5
DELETE        V5 AT LEVEL     4
DELETE        V4 AT LEVEL     3
DELETE        V5 AT LEVEL     2
FLIP        V1 TO >=          1 AT     1 WITH BND=    -12.833333
SET        V4 TO <=      1 AT     2, BND=  -22.00     TWIN= -12.93         48

NEW INTEGER SOLUTION OF    22.0000000     AT BRANCH     21 PIVOT      48
BOUND ON OPTIMUM:  12.00000
FLIP        V4 TO >=          2 AT     2 WITH BND=    -12.925926
SET        V5 TO <=      9 AT     3, BND=  -14.08     TWIN=-0.1000E+31     50
SET        V4 TO <=      4 AT     4, BND=  -15.00     TWIN= -14.59         53

NEW INTEGER SOLUTION OF    15.0000000     AT BRANCH     23 PIVOT      53
BOUND ON OPTIMUM:  12.00000
DELETE        V4 AT LEVEL     4
DELETE        V5 AT LEVEL     3
DELETE        V4 AT LEVEL     2
DELETE        V1 AT LEVEL     1
RELEASE FIXED VARIABLES
FIX ALL VARS.(     2)  WITH RC >  0.000000E+00
SET        V4 TO <=      1 AT     1, BND=  -12.57     TWIN= -12.70         63
SET        V3 TO <=      2 AT     2, BND=  -12.75     TWIN=-0.1000E+31     65
SET        V4 TO >=      1 AT     3, BND=  -12.89     TWIN= -13.29         67
SET        V5 TO <=      9 AT     4, BND=  -12.89     TWIN=-0.1000E+31     67
SET        V5 TO >=      9 AT     5, BND=  -15.43     TWIN=-0.1000E+31     69
DELETE        V3 AT LEVEL     6
DELETE        V5 AT LEVEL     5
DELETE        V5 AT LEVEL     4
FLIP        V4 TO <=          0 AT     3 WITH BND=    -13.285714
SET        V3 TO >=      4 AT     4, BND=  -13.81     TWIN=-0.1000E+31     71
SET        V3 TO <=      4 AT     5, BND=  -13.81     TWIN=-0.1000E+31     71
DELETE        V5 AT LEVEL     6
DELETE        V3 AT LEVEL     5
DELETE        V3 AT LEVEL     4
DELETE        V4 AT LEVEL     3
DELETE        V3 AT LEVEL     2
FLIP        V4 TO >=          2 AT     1 WITH BND=    -12.703704
SET        V5 TO <=      9 AT     2, BND=  -13.58     TWIN=-0.1000E+31     73
SET        V5 TO >=      9 AT     3, BND=  -13.58     TWIN=-0.1000E+31     73
SET        V4 TO <=      3 AT     4, BND=  -14.00     TWIN=-0.1000E+31     74

NEW INTEGER SOLUTION OF    14.0000000     AT BRANCH     31 PIVOT      74
BOUND ON OPTIMUM:  12.33333
DELETE        V4 AT LEVEL     4
DELETE        V5 AT LEVEL     3
DELETE        V5 AT LEVEL     2
DELETE        V4 AT LEVEL     1
RELEASE FIXED VARIABLES
FIX ALL VARS.(     2)  WITH RC >  0.000000E+00
SET        V4 TO >=      2 AT     1, BND=  -12.85     TWIN= -14.67         83
SET        V4 TO <=      2 AT     2, BND=  -12.85     TWIN=-0.1000E+31     83
SET        V5 TO <=      9 AT     3, BND=  -19.67     TWIN=-0.1000E+31     85
DELETE        V5 AT LEVEL     3
DELETE        V4 AT LEVEL     2
DELETE        V4 AT LEVEL     1
RELEASE FIXED VARIABLES
SET        V3 TO <=      0 AT     1, BND=  -13.58     TWIN= -13.08         97
DELETE        V3 AT LEVEL     1
ENUMERATION COMPLETE. BRANCHES=    34 PIVOTS=      97

LAST INTEGER SOLUTION IS THE BEST FOUND
RE-INSTALLING BEST SOLUTION...

       OBJECTIVE FUNCTION VALUE

       1)      14.00000

 VARIABLE        VALUE          REDUCED COST
       V1         0.000000          1.000000
       V2         0.000000          1.000000
       V3         2.000000          1.000000
       V4         3.000000          1.000000
       V5         9.000000          1.000000


      ROW   SLACK OR SURPLUS     DUAL PRICES
       2)         0.000000          0.000000

NO. ITERATIONS=       97
BRANCHES=    34 DETERM.=  1.000E    0
```

```
MIN v1 + v2 + v3 + v4 + v5
ST
        v1 + 3v2 + 7v3 + 12v4 + 27v5= 293
END
GIN v1
GIN v2
GIN v3
GIN v4
GIN v5
```

LINDO with GIN provided my expected total of 14 coins but with different coins used. This is due partly to the nature of coin values in relation to this specific A=293 and the algorithm used in LINDO.

The coins provided in the LINDO result answer are: 9 27coins, 3 12coins, and 2 7coins. 9 + 3 + 2 = 14