Trevor Stahl

[stahltr@oregonstate.edu](mailto: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

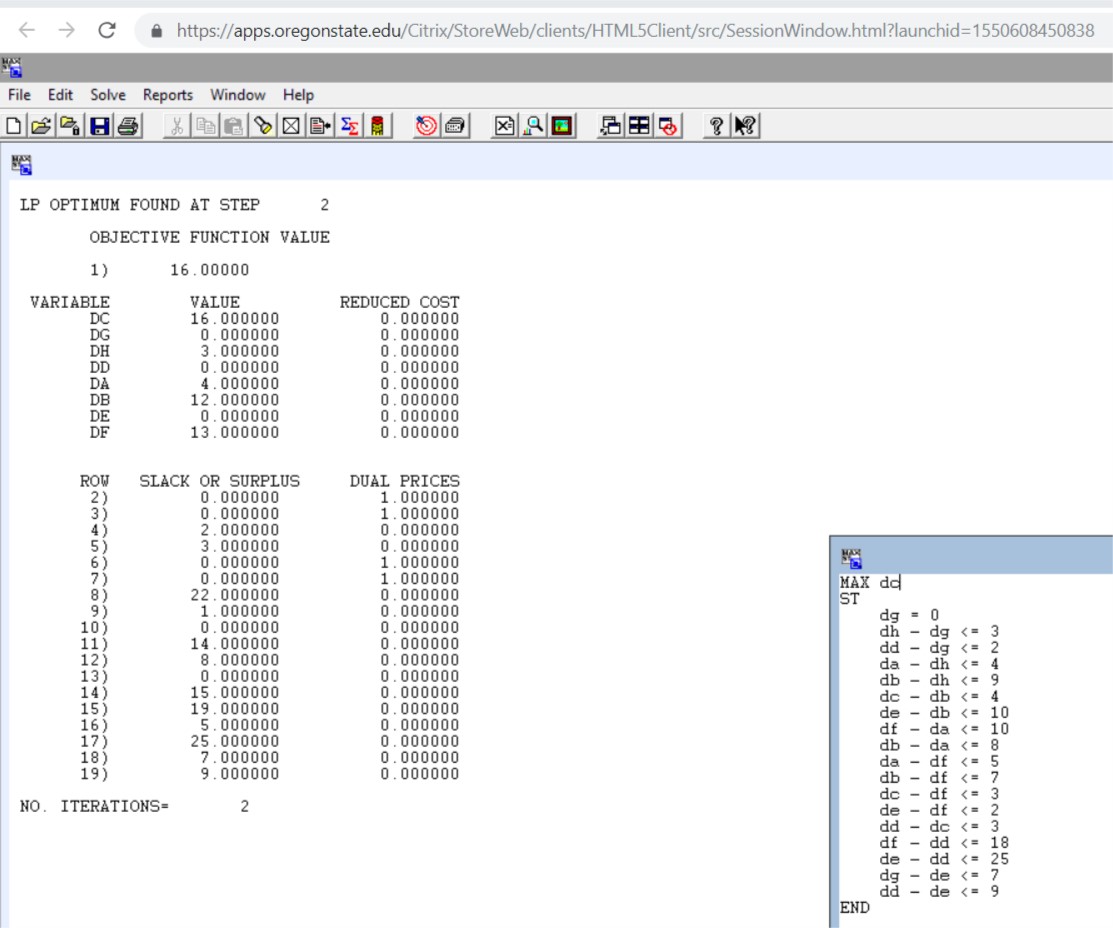
df – dd <= 18

de – dd <= 25

dg – de <= 7

dd – de <= 9

END



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

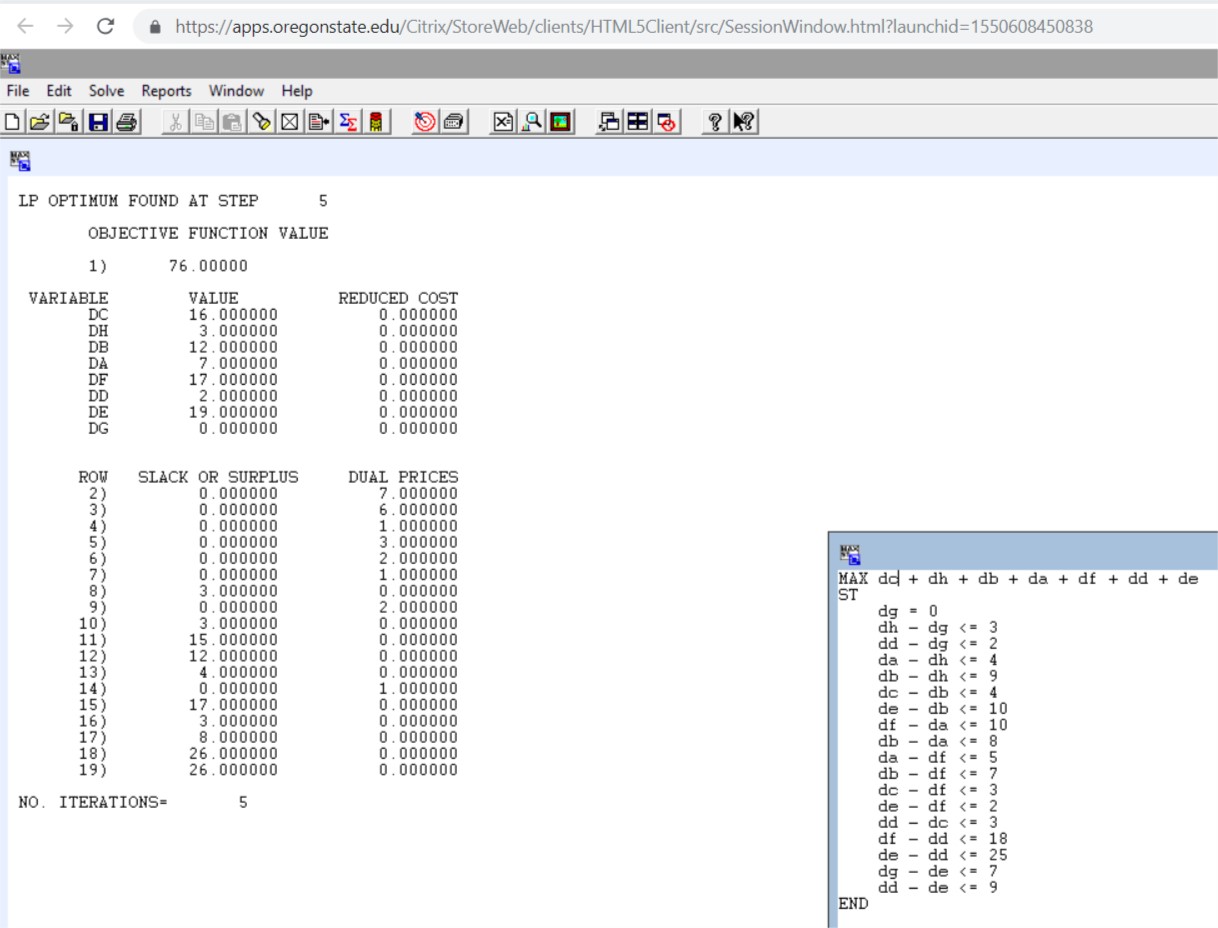
de – dd <= 25

dg – de <= 7

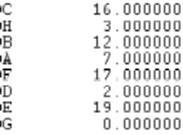
dd – de <= 9

END

\*Picture of LINDO code and output is below\*



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



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

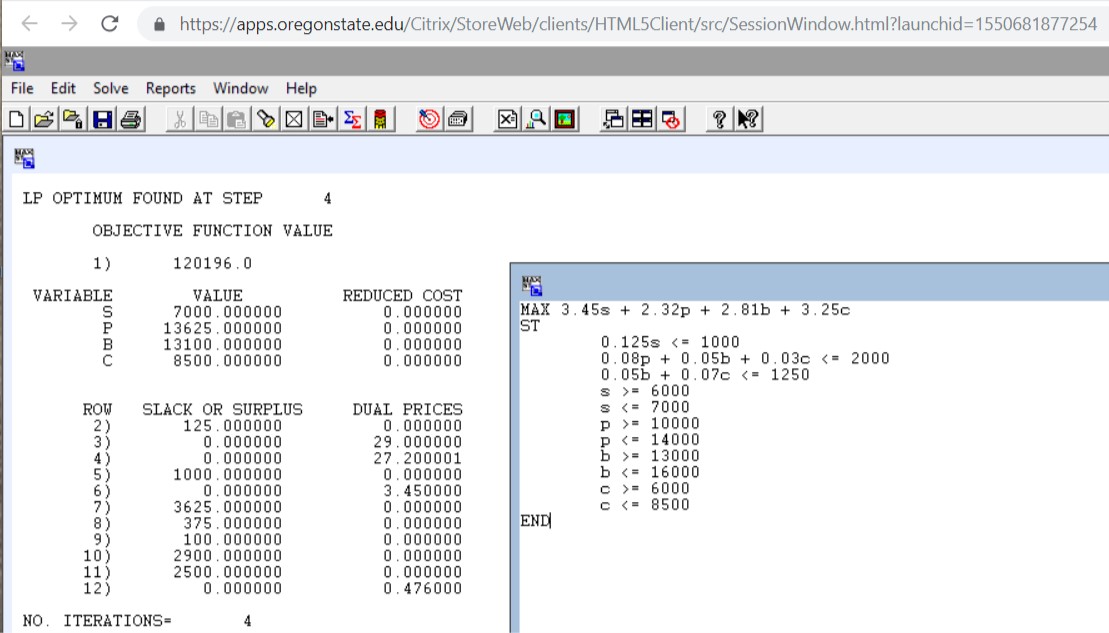
b >= 13000

b <= 16000

c >= 6000

c <= 8500

END



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

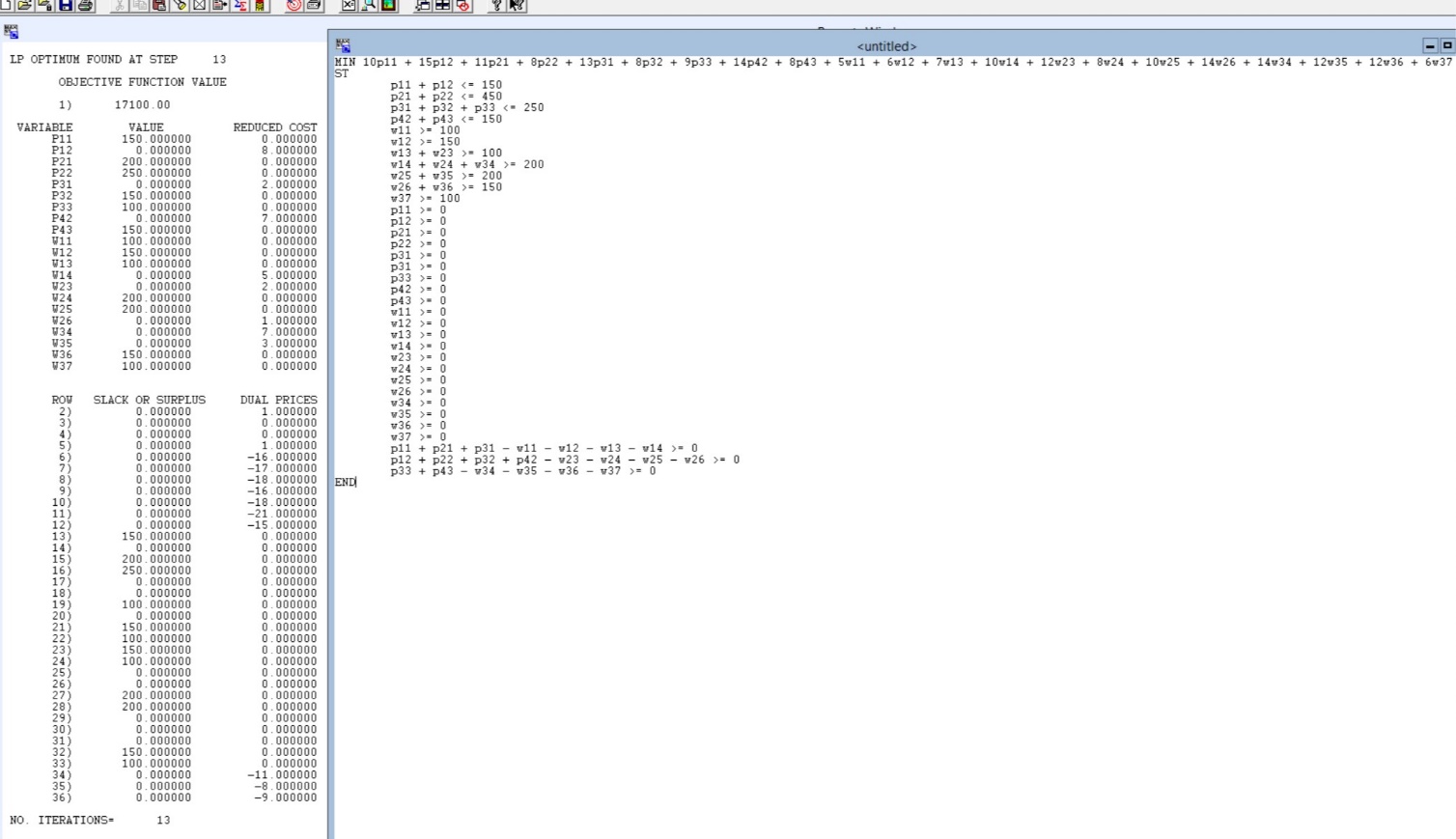
P12 + p22 + p32 + p42 – w23 -w24 – w25 – w26 >= 0

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

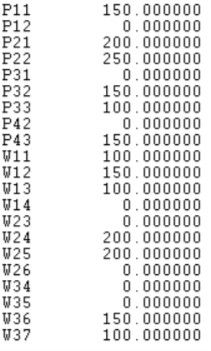
END

USING LINDO

Full picture of code and results



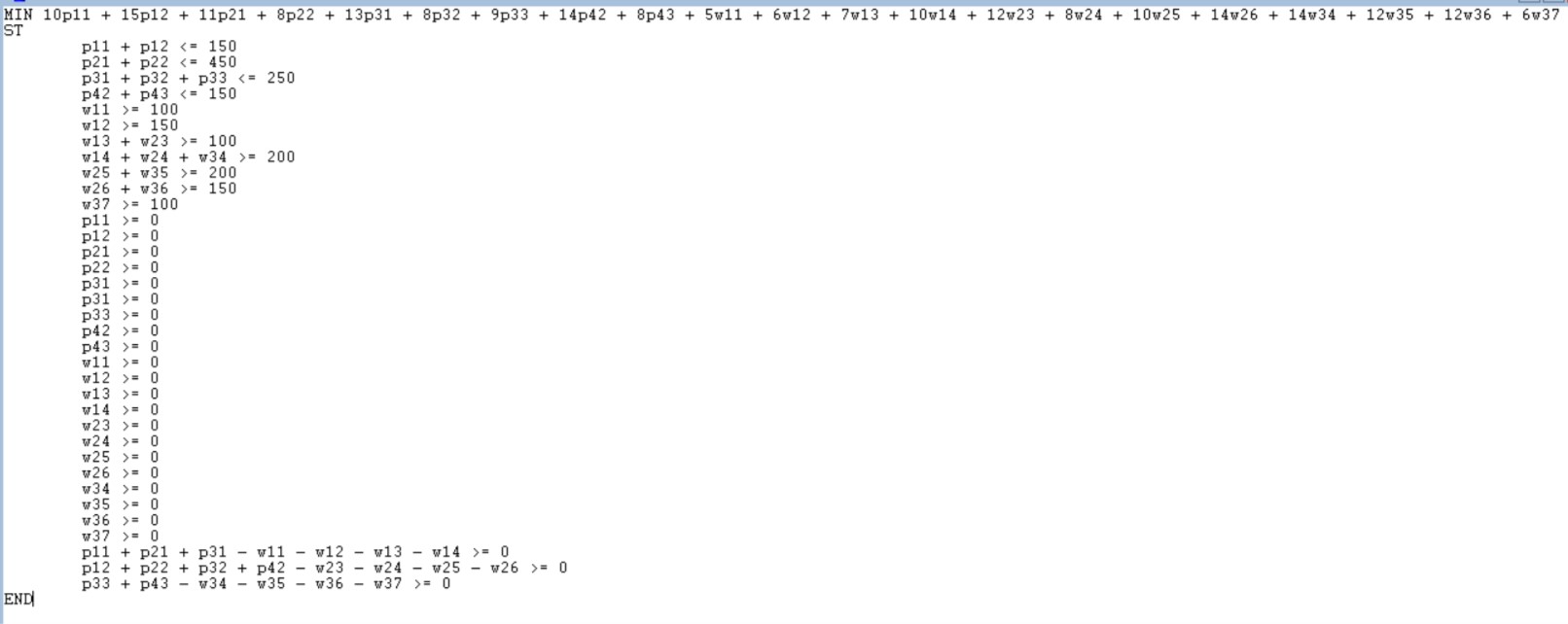
Smaller picture of variable values



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\*



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

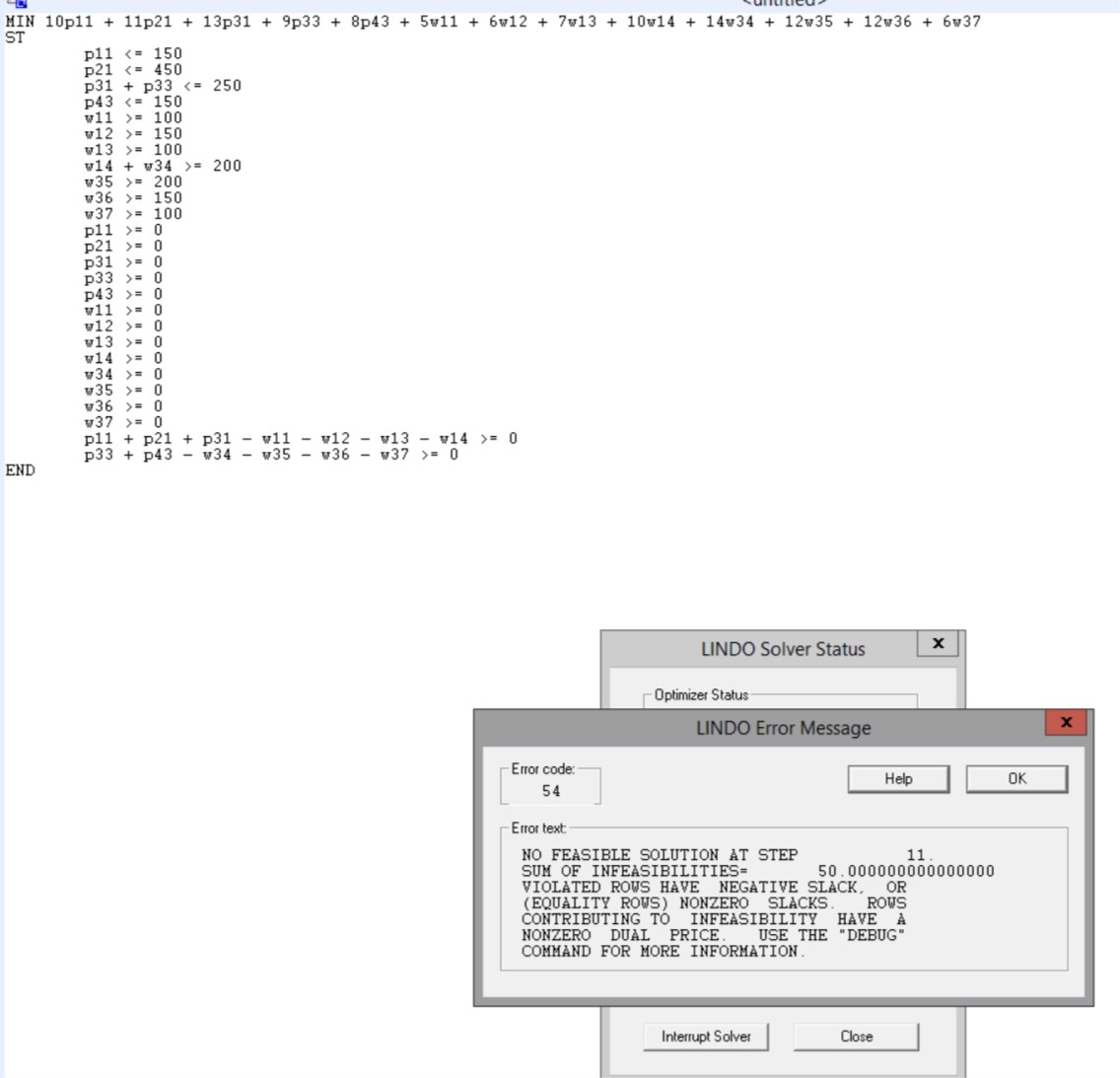
P11 + p21 + p31 - w11 – w12 – w13 – w14 >= 0

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

END

USING LINDO

RESULTS BELOW IN PICTURE



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:

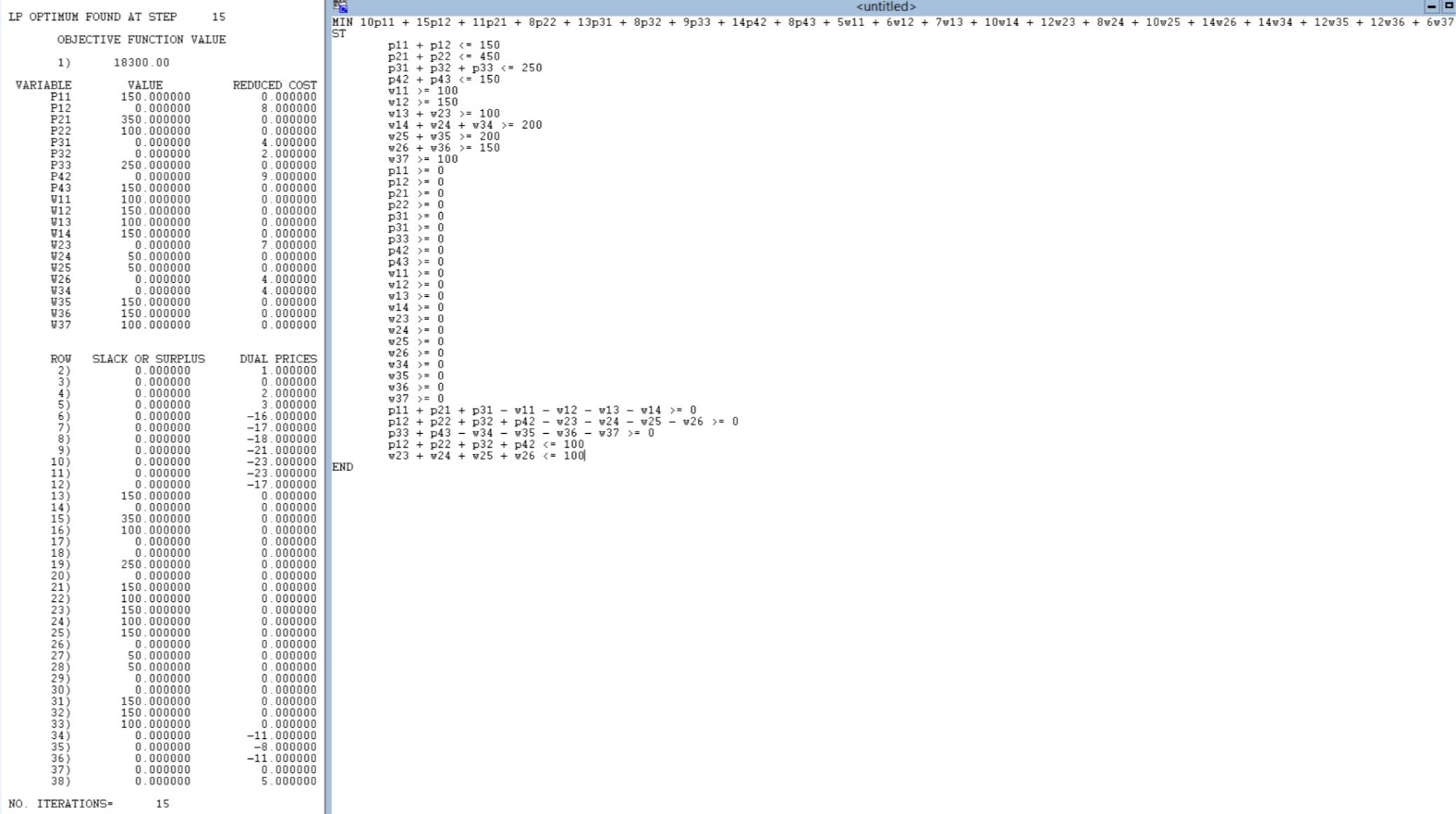
P12 + p22 + p32 + p42 <= 100

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

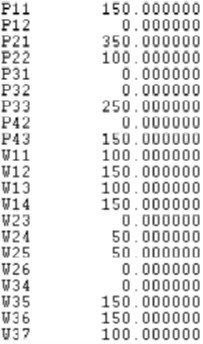


This means that this is feasible if we limit warehouse 2.

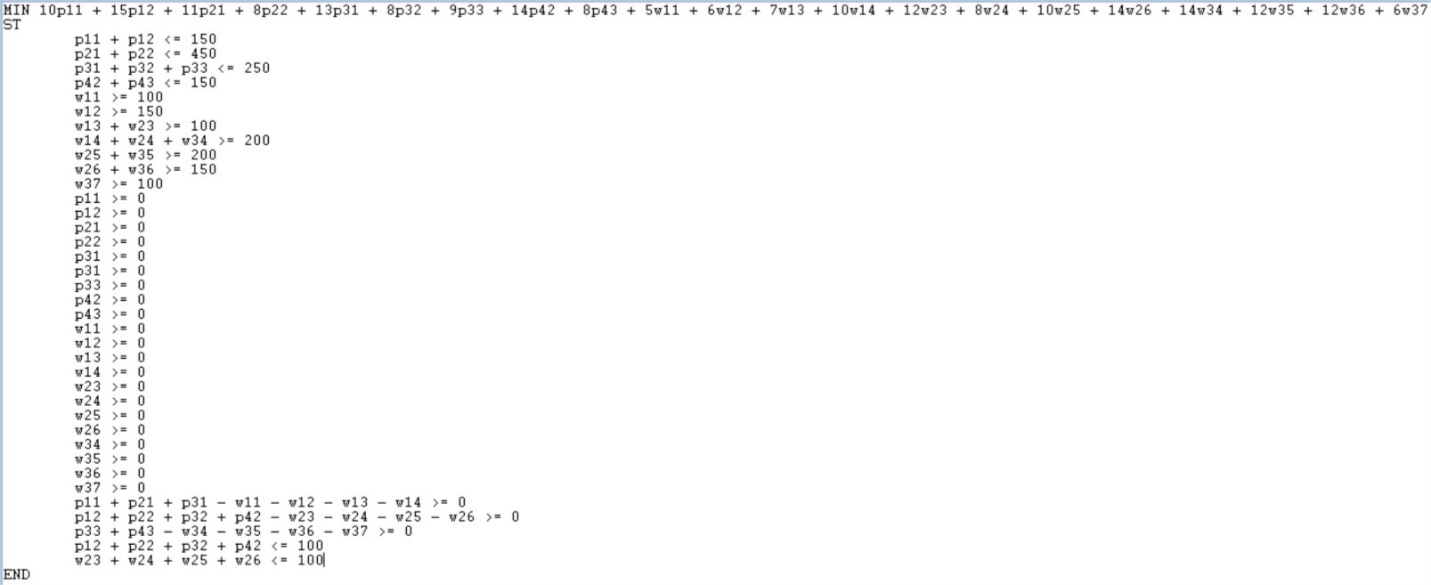
However, now the objective function optimal value is larger at 18300.

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

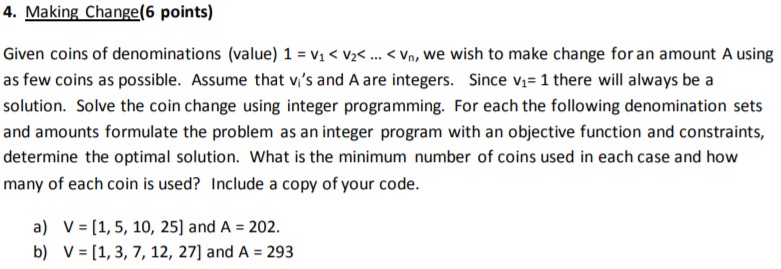
\*Below\*



And again, here is close up of code:



Problem 4



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

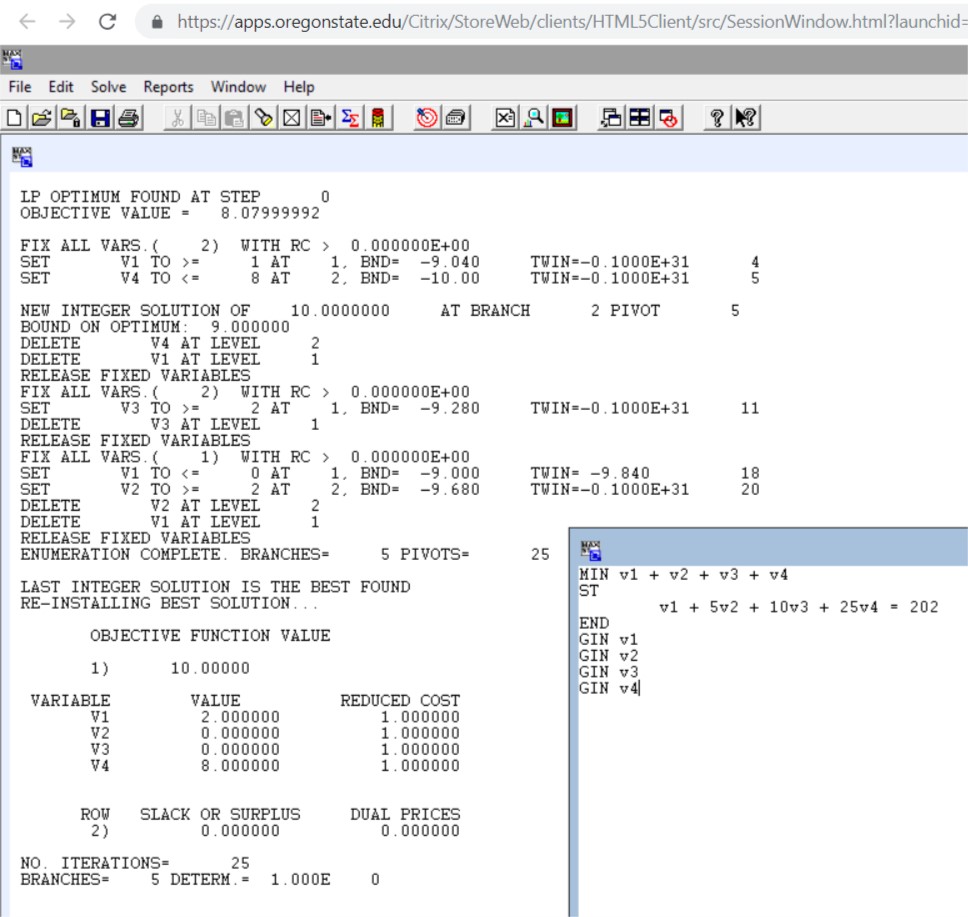
Picture of results and input code below:

Objective function value = 10.0

With 2 v1 coins

And 8 v4 coins

As predicted from my brute force.



\*b) is below\*

b)

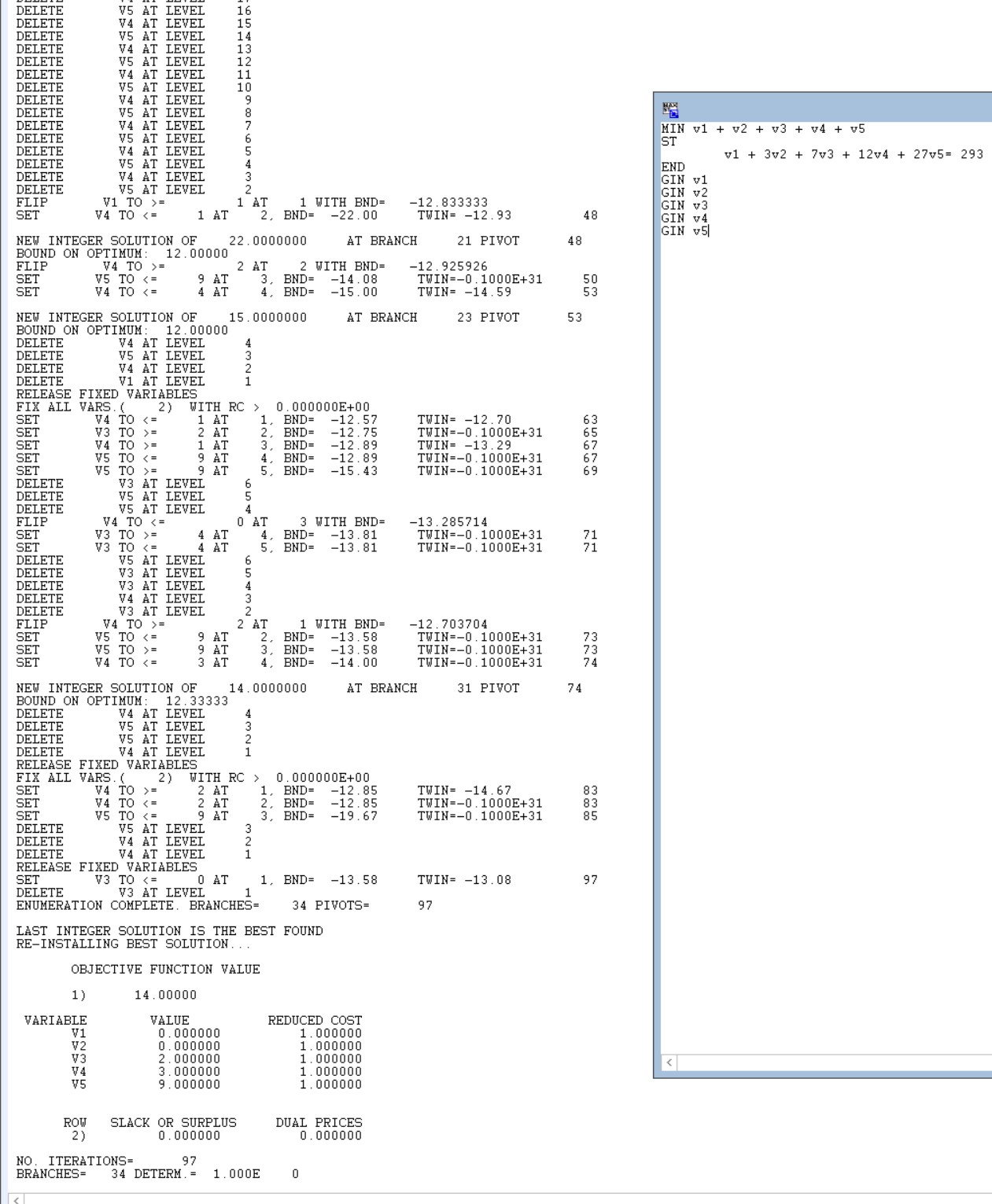
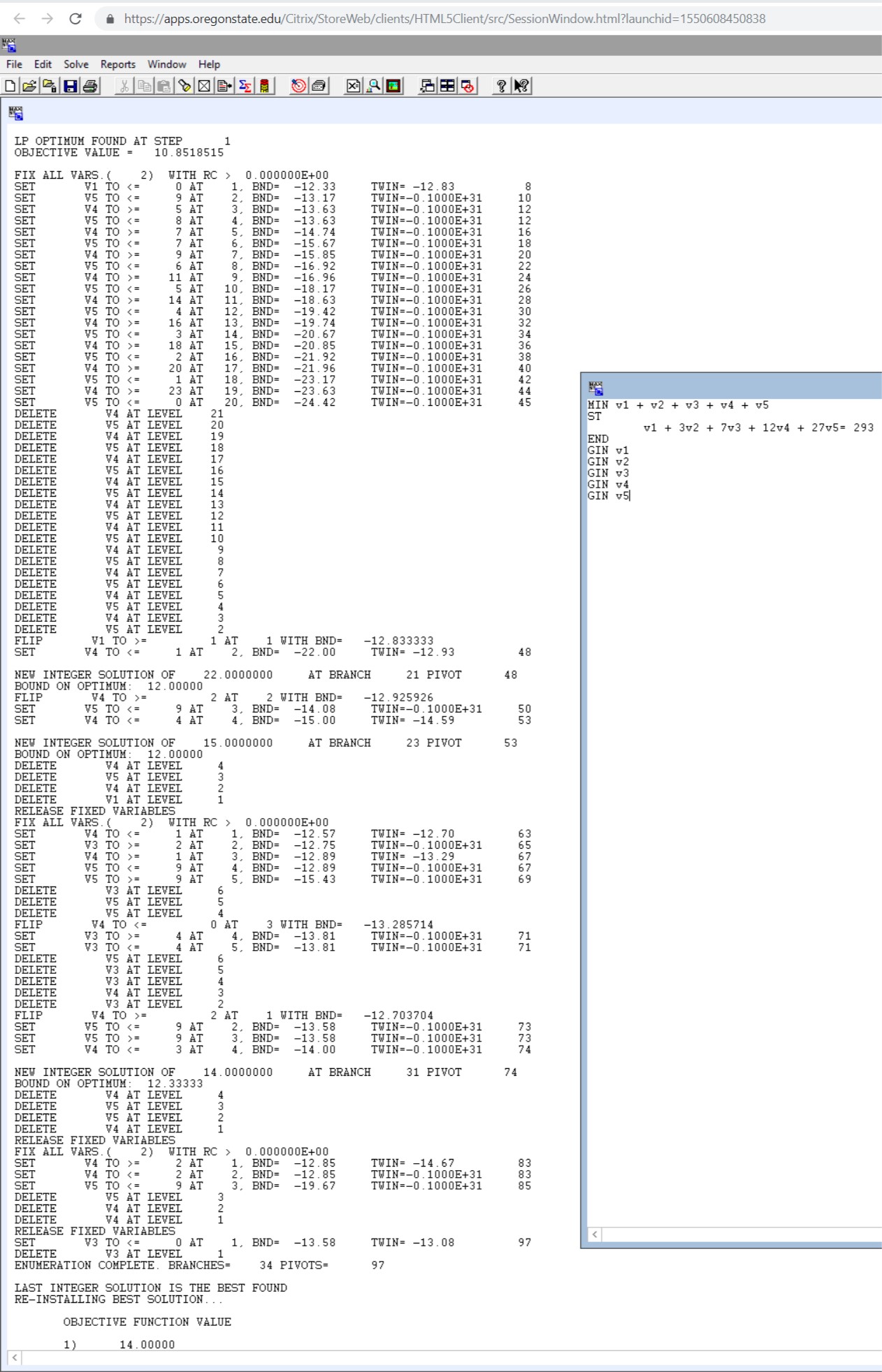
V= 1,3,7,12,27

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\*



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