Question 1

For this problem in both parts I used LINDO to solve them. In each of the part sections there is a figure that will have the data output from LINDO's solver, and the code that I ran in order to get that output. Also the the orientation of the figure is like so, on the left there is the LINDO code used to represent the problem at hand, and the on the right is the output data that shows the optimal solution.

(a) In this problem we are trying to find the shortest path from G to C, below in figure 1, the objective function is the first line of code and the constraints that the problem is subjected to are all the lines of code in between ST and end, which includes lines 2.

Objective Function:

Stretched_out(maximize) distance between G and C; finds shortest distance from G $\rightarrow$ C.

Constraints:

$dg = 0$
$dg - de \leq 7$
$dh - dg \leq 3$
$da - dh \leq 4$
$da - df \leq 5$
$df - da \leq 10$

$db - da \leq 8$
$db - dh \leq 9$
$de - db \leq 10$
$dd - de \leq 9$
$de - dd \leq 25$
$dd - dc \leq 3$

$dc - db \leq 4$
$db - df \leq 7$
$dd - dg \leq 2$
$df - dd \leq 18$
$de - df \leq 2$

The optimal solution is shown below in figure 1 of the LINDO solver program. The shortest path between G and C is 16.

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

```
LP OPTIMUM FOUND AT STEP      6

       OBJECTIVE FUNCTION VALUE

       1)      16.00000

 VARIABLE        VALUE          REDUCED COST
       DC        16.000000          0.000000
       DG         0.000000          0.000000
       DE         0.000000          0.000000
       DH         3.000000          0.000000
       DA         4.000000          0.000000
       DF         5.000000          0.000000
       DB        12.000000          0.000000
       DD         0.000000          0.000000


      ROW   SLACK OR SURPLUS     DUAL PRICES
       2)         0.000000          1.000000
       3)         7.000000          0.000000
       4)         0.000000          1.000000
       5)         3.000000          0.000000
       6)         6.000000          0.000000
       7)         9.000000          0.000000
       8)         0.000000          0.000000
       9)         0.000000          1.000000
      10)        22.000000          0.000000
      11)         9.000000          0.000000
      12)        25.000000          0.000000
      13)        19.000000          0.000000
      14)         0.000000          1.000000
      15)         0.000000          0.000000
      16)         2.000000          0.000000
      17)        13.000000          0.000000
      18)         7.000000          0.000000

NO. ITERATIONS=       6
```

Figure 1: shortest path from G to C, with LINDO code and data output.

(b) This problem asks for the shortest path from G to all other vertices on the graph. Below in figure 2, the objective function is the first line of LINDO code and the constraints that the problem is subjected to are all the lines of code in between ST and end, which includes lines 2.

Objective Function:

stretched_out (maximize) sum of the distances from G to other all other vertices. Finds shortest distance between G to all other vertices.

Constraints:

$dg = 0$

$dg - de \leq 7$

$dh - dg \leq 3$

$da - dh \leq 4$

$da - df \leq 5$

$df - da \leq 10$

$db - da \leq 8$

$db - dh \leq 9$

$de - db \leq 10$

$dd - de \leq 9$

$de - dd \leq 25$

$dd - dc \leq 3$

$dc - db \leq 4$

$db - df \leq 7$

$dd - dg \leq 2$

$df - dd \leq 18$

$de - df \leq 2$

From the data output in figure 2; the shortest paths from G to all other vertices are as follows:

G $\rightarrow$ A = 7

G $\rightarrow$ B = 12

G $\rightarrow$ C = 16

G $\rightarrow$ D = 2

G $\rightarrow$ E = 19

G $\rightarrow$ F = 17

G $\rightarrow$ G = 0

G $\rightarrow$ H = 3

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

```
 LP OPTIMUM FOUND AT STEP      9

        OBJECTIVE FUNCTION VALUE

        1)      76.00000

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


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

 NO. ITERATIONS=       9
```

Figure 2: Shortest path from G to all other vertices, with LINDO code and data output.

Question 2

This problem is asking for the maximized profits from product mixing, which means the optimal solution will be the amount of money that can be made from specific combinations of product mixes that can be manufactured.
I will use LINDO to solve this problem and produce the optimal solution that the question is asking for. In Figure 3 below I included the data output and the code that I used to produce the output, the output will include the the optimal solution that the question is asking for. The orientation of the figure is as follows: on the left side is the code used to represent the the problem, and on the right is the data output from the LINDO solver which displays all of the answers to the question in problem 2.

Type of Tie	Selling Price ($)	Labor Cost($)	Material Costs ($)	Profit Per Tie ($)
Silk (s)	6.70	0.75	$0.125 \times 20 = 2.5$	3.45
Polyester (p)	3.55	0.75	$0.08 \times 6 = 0.48$	2.32
Blend 1 (b)	4.31	0.75	$(0.05 \times 6) + (0.05 \times 9) = 0.75$	2.81
Blend 2 (c)	4.81	0.75	$(0.03 \times 6) + (0.07 \times 9) = 0.81$	3.25

Table 1: Profit per tie calculation table; using selling price, labor costs, and material costs. $profit\ per\ tie = selling\ price - labor\ cost - material\ cost$

Objective Function:

maximize $3.45s + 2.32p + 2.81b + 3.25c$, this maximizes the profits that can be gained from product mixing.

Constraints:

$0.125s \leq 1000$
$0.08p + 0.05b + 0.03c \leq 2000$
$0.03b + 0.07c \leq 2000$
$6000 \leq s \leq 7000$
$1000 \leq p \leq 14000$
$13000 \leq b \leq 16000$
$6000 \leq c \leq 8500$

Optimal solution (max profit): $120,196.00. Each type of tie needed to be made to maximize profits:

Silk: 7,000 units
Polyester: 13,625 units
Blend 1: 13,100 units
Blend 2: 8,500 units

```
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 >= 1000
        p <= 14000
        b >= 13000
        b <= 16000
        c >= 6000
        c <= 8500
end
```

```
 P 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)     12625.000000          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
```

Figure 3: Optimal product mix to maximum profit, with LINDO code and data output.

Question 3

For parts a-c I will be using LINDO to running linear programming optimizations to find the optimal solutions that each part asks for. The figures that I include in each section follows this layout; on the left is the LINDO code and on the right is the data output from the code that is solved, the objective function's output and answers to the question are displayed.[1]

(a) Minimize Shipping Costs of Refrigerators:

Objective Function:

minimize the shipping cost of the refrigerators ($10s1\_1 + 15s1\_2 + 11s2\_1 + 8s2\_2 + 13s3\_1 + 8s3\_2 + 9s3\_3 + 14s4\_2 + 8s4\_3 + 5d1\_1 + 6d1\_2 + 7d1\_3 + 10d1\_4 + 12d2\_3 + 8d2\_4 + 10d2\_5 + 14d2\_6 + 12d3\_4 + 12d3\_5 + 12d3\_6 + 6d3\_7$)

Constraints:

$s1\_1 + s1\_2 \leq 150$

$s2\_1 + s2\_2 \leq 450$

$s3\_1 + s3\_2 + s3\_3 \leq 250$

$s4\_2 + s4\_3 \leq 150$

$d1\_1 \geq 100$

$d1\_2 \geq 150$

$d1\_3 + d2\_3 \geq 100$

$d1\_4 + d2\_4 + d3\_4 \geq 200$

$d2\_5 + d3\_5 \geq 200$

$d2\_6 + d3\_6 \geq 150$

$d3\_7 \geq 100$

$s1\_1 + s2\_1 + s3\_1 - d1\_1 - d1\_2 - d1\_3 - d1\_4 = 0$

$s1\_2 + s2\_2 + s3\_2 + s4\_2 - d2\_3 - d2\_4 - d2\_5 - d2\_6 = 0$

$s3\_3 + s4\_3 - d3\_4 - d3\_5 - d3\_6 - d3\_7 = 0$

$s1\_1 \geq 0$	$s3\_3 \geq 0$	$d1\_4 \geq 0$	$d3\_5 \geq 0$
$s1\_2 \geq 0$	$s4\_2 \geq 0$	$d2\_3 \geq 0$	$d3\_6 \geq 0$
$s2\_1 \geq 0$	$s4\_3 \geq 0$	$d2\_4 \geq 0$	$d3\_7 \geq 0$
$s2\_2 \geq 0$	$d1\_1 \geq 0$	$d2\_5 \geq 0$	
$s3\_1 \geq 0$	$d1\_2 \geq 0$	$d2\_6 \geq 0$	
$s3\_2 \geq 0$	$d1\_3 \geq 0$	$d3\_4 \geq 0$	

From figure 4, we get the optimal solution (minimized shipping cost is $17100.00 dollars, and the optimal shipping routes with the number of refrigerators that should be shipped on each route are as follows:

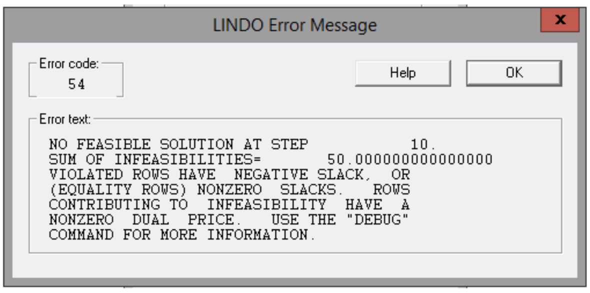
P1 → W1 = 150	P4 → W2 = 0	W2 → R4 = 200
P1 → W2 = 0	P4 → W3 = 150	W2 → R5 = 200
P2 → W1 = 200	W1 → R1 = 100	W2 → R6 = 0
P2 → W2 = 250	W1 → R2 = 150	W3 → R4 = 0
P3 → W1 = 0	W1 → R3 = 100	W3 → R5 = 0
P3 → W2 = 150	W1 → R4 = 0	W3 → R6 = 150
P3 → W3 = 100	W2 → R3 = 0	W3 → R7 = 100

```
min     10s1_1 + 15s1_2 + 11s2_1 + 8s2_2 + 12s3_1 + 8s3_2 +
        9s3_3 + 14s4_2 +8s4_3 +5d1_1 + 6d1_2 + 7d1_3 +
        10d1_4 + 12d2_3 + 8d2_4 + 10d2_5 + 14d2_6 +12d3_4 +
        12d3_5 + 12d3_6 +6d3_7
ST
        s1_1 + s1_2 <= 150
        s2_1 + s2_2 <= 450
        s3_1 + s3_2 + s3_3 <= 250
        s4_2 + s4_3 <= 150

        d1_1 >= 100
        d1_2 >= 150
        d1_3 + d2_3 >= 100
        d1_4 + d2_4 + d3_4 >= 200
        d2_5 + d3_5 >= 200
        d2_6 + d3_6 >= 150
        d3_7 >= 100

        s1_1 + s2_1 + s3_1 - d1_1 - d1_2 - d1_3 - d1_4 = 0
        s1_2 + s2_2 + s3_2 + s4_2 - d2_3 - d2_4 -d2_5 - d2_6 = 0
        s3_3 + s4_3 - d3_4 - d3_5 - d3_6 - d3_7 = 0

        s1_1 >= 0
        s1_2 >= 0
        s2_1 >= 0
        s2_2 >= 0
        s3_1 >= 0
        s3_2 >= 0
        s3_3 >= 0
        s4_2 >= 0
        s4_3 >= 0

        d1_1 >= 0
        d1_2 >= 0
        d1_3 >= 0
        d1_4 >= 0
        d2_3 >= 0
        d2_4 >= 0
        d2_5 >= 0
        d2_6 >= 0
        d3_4 >= 0
        d3_5 >= 0
        d3_6 >= 0
        d3_7 >= 0
end
```

```
 LP OPTIMUM FOUND AT STEP     13

        OBJECTIVE FUNCTION VALUE

        1)      17100.00

  VARIABLE        VALUE          REDUCED COST
      S1_1       150.000000          0.000000
      S1_2         0.000000          8.000000
      S2_1       200.000000          0.000000
      S2_2       250.000000          0.000000
      S3_1         0.000000          1.000000
      S3_2       150.000000          0.000000
      S3_3       100.000000          0.000000
      S4_2         0.000000          7.000000
      S4_3       150.000000          0.000000
      D1_1       100.000000          0.000000
      D1_2       150.000000          0.000000
      D1_3       100.000000          0.000000
      D1_4         0.000000          5.000000
      D2_3         0.000000          2.000000
      D2_4       200.000000          0.000000
      D2_5       200.000000          0.000000
      D2_6         0.000000          1.000000
      D3_4         0.000000          5.000000
      D3_5         0.000000          3.000000
      D3_6       150.000000          0.000000
      D3_7       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)         0.000000        -11.000000
       14)         0.000000         -8.000000
       15)         0.000000         -9.000000
       16)       150.000000          0.000000
       17)         0.000000          0.000000
       18)       200.000000          0.000000
       19)       250.000000          0.000000
       20)         0.000000          0.000000
       21)       150.000000          0.000000
       22)       100.000000          0.000000
       23)         0.000000          0.000000
       24)       150.000000          0.000000
       25)       100.000000          0.000000
       26)       150.000000          0.000000
       27)       100.000000          0.000000
       28)         0.000000          0.000000
       29)         0.000000          0.000000
       30)       200.000000          0.000000
       31)       200.000000          0.000000
       32)         0.000000          0.000000
       33)         0.000000          0.000000
       34)         0.000000          0.000000
       35)       150.000000          0.000000
       36)       100.000000          0.000000

 NO. ITERATIONS=      13
```

Figure 4: Transshipment Optimal Solution, with LINDO code and data output.

(b) If we take out warehouse 2 from the transshipment graph, the following LINDO code will represent the scenario where there will only be warehouses 1 and 3 that are used to process shipments. This situation is non feasible since some of the constraints that are bounding the problem can not be met; for example if retailers 5, 6 and 7 together needed 450 units of refrigerators, which would mean that only warehouse 3 can supply them. We also know that the only plants 3 and 4 can supply warehouse 3. So if we pull all of the units supplied by plants 3 and 4 we get only 400 units which would not be enough to meet the combined demand from retails 5, 6, and 7's 450 units of refrigerators, hence a non feasible solution will result from LINDO solver (error message). Below in figure 5 we can see that the error message on the left confirms my conclusions about this situation. The code that is used to represent the situation in on the right of the error message.




```
min     10s1_1 + 11s2_1 + 13s3_1 + 9s3_3 +
        8s4_3 +5d1_1 + 6d1_2 + 7d1_3 + 10d1_4 +
        14d3_4 + 12d3_5 + 12d3_6 +6d3_7
ST
        s1_1 <= 150
        s2_1 <= 450
        s3_1 + s3_3 <= 250
        s4_3 <= 150

        d1_1 >= 100
        d1_2 >= 150
        d1_3 >= 100
        d1_4 + d3_4 >= 200
        d3_5 >= 200
        d3_6 >= 150
        d3_7 >= 100

        s1_1 + s2_1 + s3_1 - d1_1 - d1_2 - d1_3 - d1_4 = 0
        s3_3 + s4_3 - d3_4 - d3_5 - d3_6 - d3_7 = 0

        s1_1 >= 0
        s2_1 >= 0
        s3_1 >= 0
        s3_3 >= 0
        s4_3 >= 0

        d1_1 >= 0
        d1_2 >= 0
        d1_3 >= 0
        d1_4 >= 0
        d3_4 >= 0
        d3_5 >= 0
        d3_6 >= 0
        d3_7 >= 0
end
```

Figure 5: Transshipment optimal solution w/o warehouse 2, with LINDO code and error message.

(c) If the warehouse 2 was to be limited to 100 shipments a week then there will be feasible solution that can be found. Limiting warehouse 2 to 100 shipments a week will definitely increase the shipping costs compared to the original problem. Consider the following linear program, such that:

Objective Function:

minimize the shipping cost of the refrigerators ($10s1\_1 + 15s1\_2 + 11s2\_1 + 8s2\_2 + 13s3\_1 + 8s3\_2 + 9s3\_3 + 14s4\_2 + 8s4\_3 + 5d1\_1 + 6d1\_2 + 7d1\_3 + 10d1\_4 + 12d2\_3 + 8d2\_4 + 10d2\_5 + 14d2\_6 + 12d3\_4 + 12d3\_5 + 12d3\_6 + 6d3\_7$)

Constraints:

$s1\_1 + s1\_2 \leq 150$

$s2\_1 + s2\_2 \leq 450$

$s3\_1 + s3\_2 + s3\_3 \leq 250$

$s4\_2 + s4\_3 \leq 150$

$s1\_2 + s2\_2 + s3\_2 + s4\_2 \leq 100$

$d1\_1 \geq 100$

$d1\_2 \geq 150$

$d1\_3 + d2\_3 \geq 100$

$d1\_4 + d2\_4 + d3\_4 \geq 200$

$d2\_5 + d3\_5 \geq 200$

$d2\_6 + d3\_6 \geq 150$

$d3\_7 \geq 100$

$s1\_1 + s2\_1 + s3\_1 - d1\_1 - d1\_2 - d1\_3 - d1\_4 = 0$

$s1\_2 + s2\_2 + s3\_2 + s4\_2 - d2\_3 - d2\_4 - d2\_5 - d2\_6 = 0$

$s3\_3 + s4\_3 - d3\_4 - d3\_5 - d3\_6 - d3\_7 = 0$

$s1\_1 \geq 0$	$s3\_1 \geq 0$	$s4\_3 \geq 0$	$d1\_4 \geq 0$
$s1\_2 \geq 0$	$s3\_2 \geq 0$	$d1\_1 \geq 0$	$d2\_3 \geq 0$
$s2\_1 \geq 0$	$s3\_3 \geq 0$	$d1\_2 \geq 0$	$d2\_4 \geq 0$
$s2\_2 \geq 0$	$s4\_2 \geq 0$	$d1\_3 \geq 0$	$d2\_5 \geq 0$

$d2\_6 \geq 0$ $\quad d3\_5 \geq 0$ $\quad d3\_7 \geq 0$
$d3\_4 \geq 0$ $\quad d3\_6 \geq 0$

From figure 6, we get that the optimal solution (minimized shipping costs) is \$18300.00 dollars, with the optimal shipping routes and the optimal number of refrigerators per route is as follows:

P1 → W1 = 150	P4 → W2 = 0	W2 → R4 = 50
P1 → W2 = 0	P4 → W3 = 150	W2 → R5 = 50
P2 → W1 = 350	W1 → R1 = 100	W2 → R6 = 0
P2 → W2 = 100	W1 → R2 = 150	W3 → R4 = 0
P3 → W1 = 0	W1 → R3 = 100	W3 → R5 = 150
P3 → W2 = 0	W1 → R4 = 150	W3 → R6 = 150
P3 → W3 = 250	W2 → R3 = 0	W3 → R7 = 100

```
min     10s1_1 + 15s1_2 + 11s2_1 + 8s2_2 + 13s3_1 + 8s3_2 +
        9s3_3 + 14s4_2 +8s4_3 +5d1_1 + 6d1_2 + 7d1_3 +
        10d1_4 + 12d2_3 + 8d2_4 + 10d2_5 + 14d2_6 +12d3_4 +
        12d3_5 + 12d3_6 +6d3_7
ST
        s1_1 + s1_2 <= 150
        s2_1 + s2_2 <= 450
        s3_1 + s3_2 + s3_3 <= 250
        s4_2 + s4_3 <= 150
        s1_2 + s2_2 + s3_2 + s4_2 <= 100

        d1_1 >= 100
        d1_2 >= 150
        d1_3 + d2_3 >= 100
        d1_4 + d2_4 + d3_4 >= 200
        d2_5 + d3_5 >= 200
        d2_6 + d3_6 >= 150
        d3_7 >= 100

        s1_1 + s2_1 + s3_1 - d1_1 - d1_2 - d1_3 - d1_4 = 0
        s1_2 + s2_2 + s3_2 + s4_2 - d2_3 - d2_4 -d2_5 - d2_6 = 0
        s3_3 + s4_3 - d3_4 - d3_5 - d3_6 - d3_7 = 0

        s1_1 >= 0
        s1_2 >= 0
        s2_1 >= 0
        s2_2 >= 0
        s3_1 >= 0
        s3_2 >= 0
        s3_3 >= 0
        s4_2 >= 0
        s4_3 >= 0

        d1_1 >= 0
        d1_2 >= 0
        d1_3 >= 0
        d1_4 >= 0
        d2_3 >= 0
        d2_4 >= 0
        d2_5 >= 0
        d2_6 >= 0
        d3_4 >= 0
        d3_5 >= 0
        d3_6 >= 0
        d3_7 >= 0
end
```

```
        OBJECTIVE FUNCTION VALUE

        1)      18300.00

  VARIABLE        VALUE          REDUCED COST
      S1_1       150.000000          0.000000
      S1_2         0.000000          8.000000
      S2_1       350.000000          0.000000
      S2_2       100.000000          0.000000
      S3_1         0.000000          4.000000
      S3_2         0.000000          2.000000
      S3_3       250.000000          0.000000
      S4_2         0.000000          9.000000
      S4_3       150.000000          0.000000
      D1_1       100.000000          0.000000
      D1_2       150.000000          0.000000
      D1_3       100.000000          0.000000
      D1_4       150.000000          0.000000
      D2_3         0.000000          7.000000
      D2_4        50.000000          0.000000
      D2_5        50.000000          0.000000
      D2_6         0.000000          4.000000
      D3_4         0.000000          2.000000
      D3_5       150.000000          0.000000
      D3_6       150.000000          0.000000
      D3_7       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          5.000000
        7)         0.000000        -16.000000
        8)         0.000000        -17.000000
        9)         0.000000        -18.000000
       10)         0.000000        -21.000000
       11)         0.000000        -23.000000
       12)         0.000000        -23.000000
       13)         0.000000        -17.000000
       14)         0.000000        -11.000000
       15)         0.000000        -13.000000
       16)         0.000000        -11.000000
       17)       150.000000          0.000000
       18)         0.000000          0.000000
       19)       350.000000          0.000000
       20)       100.000000          0.000000
       21)         0.000000          0.000000
       22)         0.000000          0.000000
       23)       250.000000          0.000000
       24)         0.000000          0.000000
       25)       150.000000          0.000000
       26)       100.000000          0.000000
       27)       150.000000          0.000000
       28)       100.000000          0.000000
       29)       150.000000          0.000000
       30)         0.000000          0.000000
       31)        50.000000          0.000000
       32)        50.000000          0.000000
       33)         0.000000          0.000000
       34)         0.000000          0.000000
       35)       150.000000          0.000000
       36)       150.000000          0.000000
       37)       100.000000          0.000000

 NO. ITERATIONS=      15
```

Figure 6: Transshipment optimal solution with limited warehouse 2, with LINDO code and data output.

Question 4

For the following coining problem sets a) and b), I will be using Excel Solver to come up with the minimum amount of coin used to make the exact change. For each of the problem sets I will include the the decision variables, objective function, the constraints to that problem, and the optimal solution. I will also include a figure that will show the equations used for the objection function and constraints used to the solve the problem in each coin set.[2]

(a) $V = [1, 5, 10, 25]$ where $A = 202$.

A will also be known as "Desired Amount".

Objective Function and Decision Variables:

Minimize: $w + x + y + z$ coins used to make A, where decision variables are:

w = amount of 1's coins used.

x = amount of 5's coins used.

y = amount of 10's coins used.

z = amount of 25's coins used.

Constraints:

$w \geq 0$

$x \geq 0$

$y \geq 0$

$z \geq 0$

w = Integers

x = Integers

y = Integers

z = Integers

$w + 5x + 10y + 25z = A$ (Desired Amount)

From figure 7 below, the optimal solution (minimum coins used to make exact desired change) is 10 coins. The solution uses 8 25's coins, and 2 1's coins.

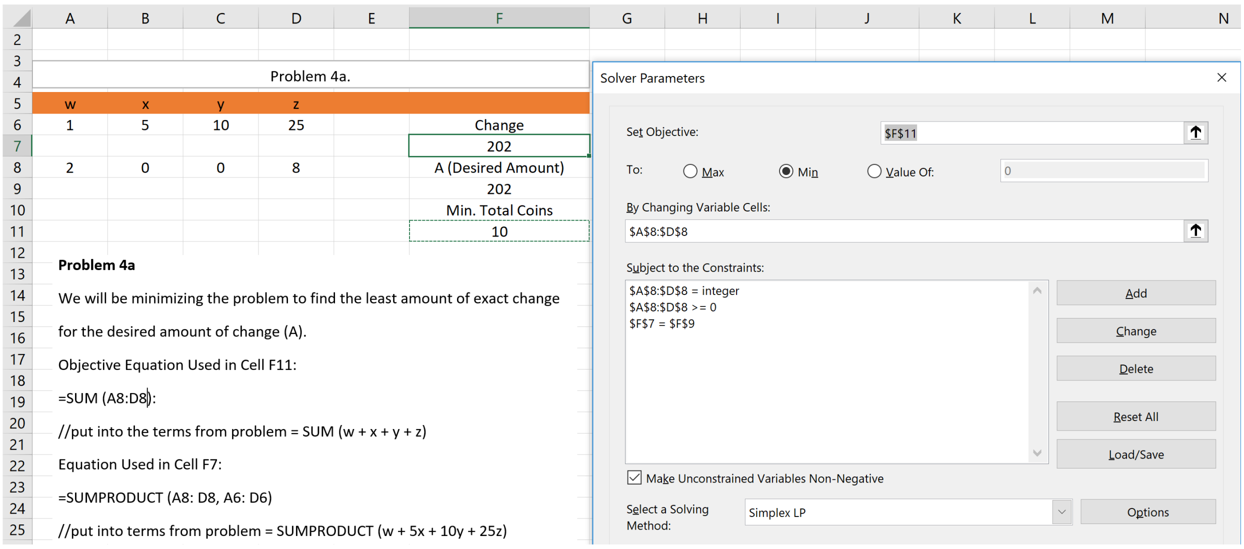


Figure 7: Coin Set 1 Optimal Output with Constraints and Objective Function using Microsoft Excel.

(b) $V = [1, 3, 7, 12, 27]$ where $A = 293$

A can also be known as "Desired Amount".

Objective Function and Decision Variables:

Minimize: $v + w + x + y + z$ coins used to make A, where decision variables are:

w = amount of 1's coins used.

w = amount of 3's coins used.

x = amount of 7's coins used.

y = amount of 12's coins used.

z = amount of 27's coins used.

Constraints:

$w \geq 0$

$x \geq 0$

$y \geq 0$

$z \geq 0$

w = Integers

x = Integers

y = Integers

z = Integers

$v+3w+7x+12y+27z = A$ (Desired Amount)

From figure 8 below, the optimal solution (minimum coins used to make exact desired change) is 14 coins. The solution uses 10 27's coins, 1 12's coin, 1 7's coin, 1 3's coin, and a 1's coin.

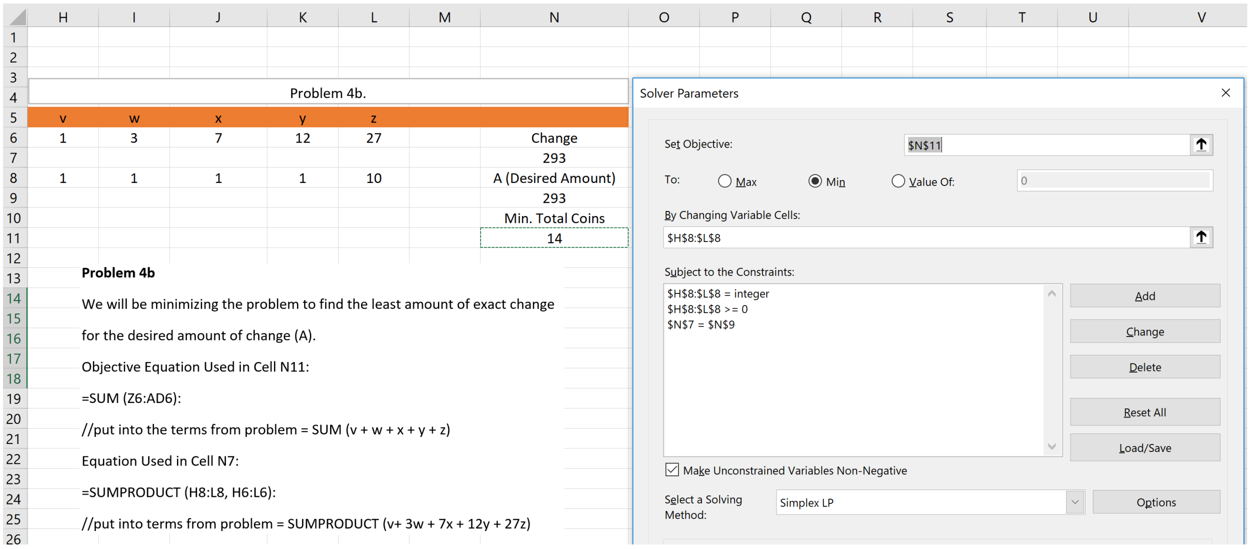


Figure 8: Coin Set 2 Optimal Output with Constraints and Objective Function using Microsoft Excel.

References

[1] B. Greve, “Transshipment.” https://www.youtube.com/watch?v=UGRIVTB9po8, 2012.

[2] M. C. Patterson, “The coin changing problem as a mathematical model.” https://oregonstate.instructure.com/courses/1683584/files/71545501/download?wrap=1, 2010.